Studies in Computational Intelligence 339

Halina Kwaśnicka
Lakhmi C. Jain (Eds.)

# Innovations in Intelligent Image Analysis

Springer

Halina Kwaśnicka and Lakhmi C. Jain (Eds.)

Innovations in Intelligent Image Analysis

## Studies in Computational Intelligence, Volume 339

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

Vol. 317. Roger Lee (Ed.)
*Computer and Information Science 2010*
ISBN 978-3-642-15404-1

Vol. 318. Oscar Castillo, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
*Soft Computing for Intelligent Control and Mobile Robotics,* 2010
ISBN 978-3-642-15533-8

Vol. 319. Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, Tokuro Matsuo, and Hirofumi Yamaki (Eds.)
*Innovations in Agent-Based Complex Automated Negotiations*, 2010
ISBN 978-3-642-15611-3

Vol. 320. xxx

Vol. 321. Dimitri Plemenos and Georgios Miaoulis (Eds.)
*Intelligent Computer Graphics 2010*
ISBN 978-3-642-15689-2

Vol. 322. Bruno Baruque and Emilio Corchado (Eds.)
*Fusion Methods for Unsupervised Learning Ensembles,* 2010
ISBN 978-3-642-16204-6

Vol. 323. Yingxu Wang, Du Zhang, and Witold Kinsner (Eds.)
*Advances in Cognitive Informatics,* 2010
ISBN 978-3-642-16082-0

Vol. 324. Alessandro Soro, Vargiu Eloisa, Giuliano Armano, and Gavino Paddeu (Eds.)
*Information Retrieval and Mining in Distributed Environments,* 2010
ISBN 978-3-642-16088-2

Vol. 325. Quan Bai and Naoki Fukuta (Eds.)
*Advances in Practical Multi-Agent Systems,* 2010
ISBN 978-3-642-16097-4

Vol. 326. Sheryl Brahnam and Lakhmi C. Jain (Eds.)
*Advanced Computational Intelligence Paradigms in Healthcare 5,* 2010
ISBN 978-3-642-16094-3

Vol. 327. Slawomir Wiak and Ewa Napieralska-Juszczak (Eds.)
*Computational Methods for the Innovative Design of Electrical Devices,* 2010
ISBN 978-3-642-16224-4

Vol. 328. Raoul Huys and Viktor K. Jirsa (Eds.)
*Nonlinear Dynamics in Human Behavior,* 2010
ISBN 978-3-642-16261-9

Vol. 329. Santi Caballé, Fatos Xhafa, and Ajith Abraham (Eds.)
*Intelligent Networking, Collaborative Systems and Applications,* 2010
ISBN 978-3-642-16792-8

Vol. 330. Steffen Rendle
*Context-Aware Ranking with Factorization Models,* 2010
ISBN 978-3-642-16897-0

Vol. 331. Athena Vakali and Lakhmi C. Jain (Eds.)
*New Directions in Web Data Management 1,* 2011
ISBN 978-3-642-17550-3

Vol. 332. Jianguo Zhang, Ling Shao, Lei Zhang, and Graeme A. Jones (Eds.)
*Intelligent Video Event Analysis and Understanding,* 2011
ISBN 978-3-642-17553-4

Vol. 333. Fedja Hadzic, Henry Tan, and Tharam S. Dillon
*Mining of Data with Complex Structures,* 2011
ISBN 978-3-642-17556-5

Vol. 334. Álvaro Herrero and Emilio Corchado (Eds.)
*Mobile Hybrid Intrusion Detection,* 2011
ISBN 978-3-642-18298-3

Vol. 335. Radomir S. Stankovic and Jaakko Astola
*From Boolean Logic to Switching Circuits and Automata,* 2011
ISBN 978-3-642-11681-0

Vol. 336. Paolo Remagnino, Dorothy N. Monekosso and Lakhmi C. Jain (Eds.)
*Innovations in Defence Support Systems – 3,* 2011
ISBN 978-3-642-18277-8

Vol. 337. Sheryl Brahnam and Lakhmi C. Jain (Eds.)
*Advanced Computational Intelligence Paradigms in Healthcare 6,* 2011
ISBN 978-3-642-17823-8

Vol. 338. Lakhmi C. Jain, Eugene V. Aidman, and Canicious Abeynayake (Eds.)
*Innovations in Defence Support Systems – 2,* 2011
ISBN 978-3-642-17763-7

Vol. 339. Halina Kwaśnicka and Lakhmi C. Jain (Eds.)
*Innovations in Intelligent Image Analysis,* 2011
ISBN 978-3-642-17933-4

Halina Kwaśnicka and Lakhmi C. Jain (Eds.)

# Innovations in Intelligent Image Analysis

Springer

Prof. Halina Kwaśnicka
Deputy Director for Scientific Researches
Institute of Informatics
Wroclaw University of Technology
Wyb. Wyspianskiego 27
51-370 Wroclaw
Poland

Prof. Lakhmi C. Jain
Professor of Knowledge-Based Engineering
Founding Director of the KES Centre
School of Electrical and Information Engineering
University of South Australia Adelaide
Mawson Lakes Campus
South Australia SA 5095
Australia
E-mail: Lakhmi.jain@unisa.edu.au

ISBN 978-3-642-17933-4 e-ISBN 978-3-642-17934-1

DOI 10.1007/978-3-642-17934-1

Studies in Computational Intelligence ISSN 1860-949X

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

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

# Foreword

## How Intelligent Must A System Be for Image Analysis?

Professor Dr Ryszard Tadeusiewicz
Head of the Automatic Control Department,
AGH University of Science and Technology, Krakow, Poland
email: rtad@agh.edu.pl

It is an honor to have been invited to write some words of introduction to this unique book. As a computer vision specialist I am a familiar with the problems and some of the methods of the image analysis. I have been involved for many years in artificial intelligence research. I therefore read all of the chapters in this book carefully and I find it to be a very valuable and impressive contribution to the subject. A general overview of the book's contents is done in Chapter 1 entitled Advances in Intelligent Image Analysis. The editors of the book wrote this valuable oversight. In the chapter a birds eye survey of computer vision problems and methodology is presented. This overview is one of the best I have ever read. Therefore in this introduction I am not able give characteristic of the books contents. This is because it is very well done. Therefore in this introduction I will focus my attention on the word "intelligent" which is expressed in the title of the book and also in the text of many chapters. I will now try to answer the question formulated as part of the title of foreword:

### How Intelligent Must A System Be for Image Analysis?

Apparently the answer is both simple and evident. All systems used should be as intelligent as possible. This also applies to the system used for image analysis.

The practical application is not so simple. Artificial intelligence is of course a very exciting property of contemporary IT systems. It is also very convenient and very useful in most cases where it is possible to use such a system. In my opinion every reader of this book after short reflection will be able to give many examples, where some systems are found to be too intelligent. Writing about systems which are "too intelligent" I think about such systems, which are supposed to help us when we use them. This artificial invention can go too far and particular systems are in practice not suitable for this reason.

A simple example would be the following. Contemporary word processors are "intelligent". They check spelling and try to correct mistakes. Sometimes it works, sometimes the result is worse. Most corrections are well done and useful. Sometimes correction of a simple typing error leads to the use of an unsuitable word. Thus an "intelligent correction" can change sense of the whole sentence or even the whole paragraph. In my opinion the use of "intelligence" is harmful and not useful.

Another simple example is spam filtering in email acquisition systems. Spam is a big problem and everybody loses valuable time in removing unnecessary items from an ever increasing stream of incoming information. Intelligent spam filtering never works without error and every day I must check my email inbox and my trashcan because of the many errors. Because of these "intelligent" algorithms many of the spams remain seemingly useful worthy contributions and attention worthy in the inbox. Frequently many important and valuable letters are automatically thrown into the garbage. I consequently wonder if my own self made additional filter selection for this "garbage" is not perceptive enough to sort the information and important messages can be lost in the sorting process. In my opinion such apparently "intelligent" but inaccurate mail selection machines cannot be safely used. Unfortunately it is very difficult to switch off this option. This is because the producer of such an "intelligent" tool is proud from his/her achievement and could never believe it to be harmful.

In both of the above quoted above cases we must take into account available methods of intelligent processing and analysis of the text data. This book is dedicated to the intelligent processing and the analysis of the images. But when considering its use in the computer vision area the question "how much intelligence really necessary"? The problem becomes even more important and more complicated.

For all readers of this book it is evident that the image when treated as a data structure is much more complicated than the simple text. Images must be considered as two-dimensional when grayscale used or multidimensional when color and multispectral data representation is taken into account. It is evident, that such multidimensional data which has many such important features registered. The mutual relation between the pixels or voxels is much more complicated than simple text which is treated as merely a string of letters. We must also take into account that the human ability for image interpretation, assessment and understanding are almost ideal. Therefore almost always a visual inspection of the image will be much more advanced, precise and subtle than the efficiency of any computer vision program. People will not accept mistakes in image processing, which are evident for human assessment in all situations. The problem which is to be solved by the specialized software is very difficult. Error caused by the input data and included noises. The human eye is not sensitive to these noises and the human brain is able in a very clever manner to select useful information obtained from observed data. If a computer makes mistake the system user is very perplexed. Therefore the application of intelligent methods to the image processing, analysis and interpretation must be introduced using special care.

### A Detailed Explanation Follows

The typical way in which computer vision applied to any particular use may be divided into five or possibly six steps as shown in Figure 1. The steps are listed here. The discussion of the usefulness of artificial intelligence for particular steps is done in the next paragraphs.

The first step is part of the image acquisition process. Next we come to image filtration and other image processing procedures. Next step in the process is the image segmentation. A further step leads to image analysis. After the analysis we can enter the image classification and perhaps also recognition. Lastly but not least we try to apply an automatic understanding of the image and the image-based decision making process.

The next question is: Which level of the artificial intelligence can be applied and which one should be applied in the particular steps of the computer vision procedures?

The image acquisition process is in most applications not connected with advanced artificial intelligence methods requirements. When taking photos or for video recording intelligence is not necessary. Sometimes however we may observe exceptions. For example when the object which should be represented and analyzed on the image is moving and must be found on the space before photocopy. Image acquisition needs intelligent procedures. When there are many objects visible and we must select one of interest for the image processing and interpretation – the image acquisition must be combined with intelligent scene analysis. In typical situations the image acquisition does not need intelligence.

When the correct image is registered by the computer vision system we typically need to do some processing. This may include image filtering, denoising or enhancement. In different systems we may use different image processing methods. This is because the noise filtration process as well as image enhancement strongly depend on the final destination of the image processing. Even elementary questions such as about the difference between the useful signal and the noise is not evident without a special analysis related to the main goal of the image interpretation process.

For example when considering medical images (e.g. images from magnetic resonance scanner) we can obtain (in simplest case) two types of visual information: general shapes of the internal organs and textures filling these shapes inside. Depending on the goal of the investigation we can concentrate on the forms and dimensions of the organs or we can take into account the properties of the tissue, which can be evaluated (assessed) on the basis of image texture. Therefore once the shape of large objects visible on the image is source of useful information (this is pure signal) and the texture must be treated as noise and should therefore be filtered. For another purpose filtration must be performed in the opposite direction. That is by enhancing high frequency image components (That is the elements of the texture) and removing low frequency elements (That is big objects which relate to the particular organs).

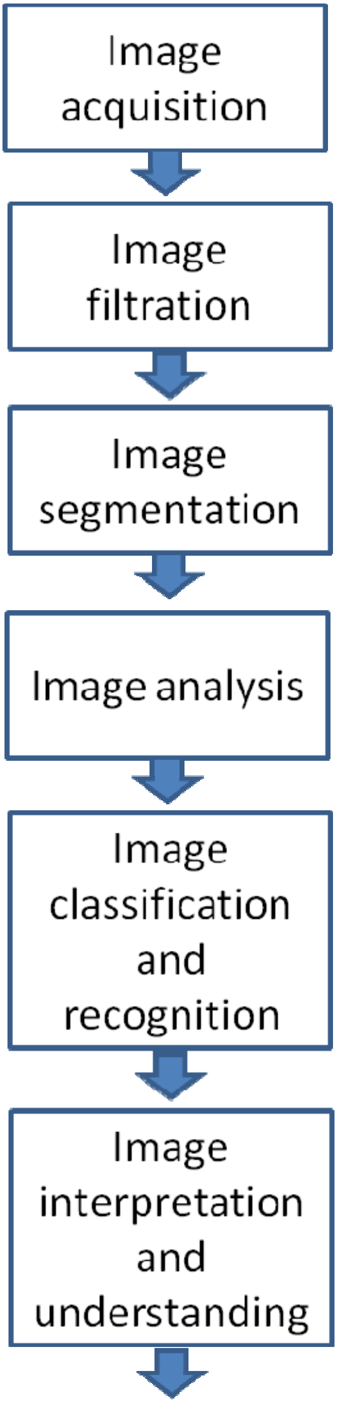


Image acquisition
Image filtration
Image segmentation
Image analysis
Image classification and recognition
Image interpretation and understanding

**Fig. 1.** Elements of the computer vision structure used for discussion

The question is, whether the selection of the proper method of image filtration and enhancement can be selected automatically during the preprocessing of registered image – or not. Another formulation of the same question is following: Are the artificial intelligence methods are applicable for image processing level and if so why?

If particular form of adaptive image enhancement can be defined and practically performed, we can then think about the intelligent image processing. Otherwise we must think about two definitely separated processes: One of the processes is related to the intelligent selection of a proper method of image processing. It must be performed by a person, who is only intelligent operator in the whole system. Another process is image processing. For example this may be filtration, enhancement, object-background differentiation. This is done using a computer. This second process typically involves a lot of mathematical operations because of millions pixels which are in the processed image. We must take also into account the context processing. For the evaluation value of one pixel on the output image need calculations taking into account the same pixel on the source image and some pixels in the neighborhood. If we imagine, that appropriate calculations related to the selected method of image filtering, must be performed for every pixel, the computational load is high for this step of image processing. The algorithms for image processing done by computer have no intelligent components. Intelligent analysis of the processing purpose and the selection of the optimal processing method is as yet done by human mind.

There is a similar situation during the image segmentation process. Before analysis, recognition and also automatic interpretation of the image – we must select the objects, whose features will be calculated during the image analysis process. A proper classification is essential for the image-based decision making processes. The differentiation between object and background is not simple when the general situation is considered. Differentiation between objects is important when considering the final destination of the whole analysis. Objects which are of only marginal importance can also very difficult problem. Here intelligent methods are indispensable.

Another area where artificial intelligence methods are necessary is that of image analysis. The main goal of image analysis is the transformation of the source image to the parametric form. Instead of huge number of pixels and voxels representing image itself, after the analysis we have sparse parameters. The information value can be equivalent or perhaps better when pursuing the principal goal of the whole system. That can be in the medical diagnosis field. This is the most goal-dependent step in the whole of the computer vision system functioning. In the general case it may be very responsible and useful to consider the use of artificial intelligence methods.

After describing the image by means of the selected method which can include artificial intelligence, we can reach the final decision making process. At this level of image interpretation artificial intelligence is indispensable. All non trivial methods of pattern clustering, classification and pattern recognition must be based on the use of artificial intelligence methods. The same situation is related to methods of automatic understanding. These are knowledge-based methods and can be used only when proper artificial intelligence algorithms have been developed.

**Concluding this discussion:** Problems and tasks described and discussed in the present book are important and lie within image processing and analysis area. Therefore everybody, who reads this book, will be more successful in both the theoretical and practical problem solving in the image processing area and computer vision based activities. In my opinion everybody reading this book will learn much as a result of the information contained in.

## References

1. Dougherty G. (2009), Digital image processing for medical application. Cambridge University Press
2. Montabone S. (2010), Beginning Digital Image Processing: Using Free Tools for Photographers, Apress, New York
3. Ogiela M. R., Tadeusiewicz R. (2008) Modern Computational Intelligence Methods for the Interpretation of Medical Images, Studies in Computational Intelligence, Volume 84, Springer-Verlag, Berlin – Heidelberg – New York
4. Tadeusiewicz R., Ogiela M. R. (2004) Medical Image Understanding Technology, Series: Studies in Fuzziness and Soft Computing, Vol. 156, Springer-Verlag, Berlin – Heidelberg – New York
5. Tadeusiewicz R. (2010), Place and Role of Intelligent Systems in Computer Science. Computer Methods in Materials Science, Vol. 10, No. 4, pp. 193-206

# Preface

For a complete historical view of the digital image processing area, we are forced to return to the 1940s and the beginning of 1950s. This is because, it is when some fundamental research on the military application of image analysis were attempted. In the 1960s such applications as character recognition, satellite imagery, image enhancement, medical image processing and videophone were studied. The rapid development of cheaper and faster computers in 1970s, enabled some particular problems such as television standards conversion, where images are processed in real time. The Massachusetts Institute of Technology (MIT) played an important role in the development of these industrial applications. Visual perception has also been used in the robotics field to locate significant objects. A system of image analysis enabled the control of a robotic arm for industrial use.

An enormous growth of the industrial applications of machine vision occured in the 1980s. This was due to the commercial availability of single board image processors, cameras for industrial applications, and the development of grayscale machine vision algorithms. The advancing computer technology in the 1990s allowed the development of very smart cameras. These were able to extract information from collected image data. This was also due to the development of the processing chip. These technical developments forced the progression of advanced studies. Consequently many methods and algorithms have since been studied and applied. In this decade the first robot vision system for navigation has been created (robot Polly, constructed by I. Horswill). The speed was obtained in the order of one meter per second.

Vision systems were developed and used in car control systems. A well known example of a car steered by computer was ALVINN. In this example, a human driver controlled the brakes and the throttle. A fully controlled by computer system car-robot later drove more than 1000 miles, in the traffic. Critical situations however were managed by the driver. Some Artificial Intelligence systems used for threat-detection include machine vision applications. They have been developed due to forced research in this field. The 2001 attacks proved a powerful incentive for this development.

Advances in medical imaging now play a very important role in medical diagnostic processes. The need for such computer assistance in detection and diagnosis fields stimulate the need for the rapid development of those methods and application of images processing and analysis. Recently, the need for image interpretation and subsequent understanding has become imperative. Such computer systems in medicine improve clinical decisions. Computers were applied to medical images from the early days. This was as early as the 1950s. Their intensive development may be observed from 1980s. Normal and pathological images are also collected and analyzed to improve the teaching of students and in the daily work of medical doctors.

Image processing and analysis techniques are intensively applied in many very different fields. The collections of images are now described. That is the images obtained should be annotated by appropriate words. Such a task is very burdensome for humans, and consequently automatic methods of image annotation have been intensively studied. Image retrieval is interesting and valuable research area.

All of the above applications of computer vision systems are extremely important both from the practical and scientific point of view. The most fascinating area is that of image understanding. We can interpret a given image and determine what it presents. Examples include expressions of fear, gladness, impression, for example. The question is: how we can make a computer 'read' the meaning of any given image? The proper interpretation of images is a very important first step. An example is that of Medical Decision Support System Development. This involves searching for similar images. Here similarity as in the context of the meaning of the images. It appears that good image interpretation of the results may require some prior knowledge.

We present here only a small introduction to new and important research in the images processing and analysis area. It is hoped that this book will be useful for scientists and students involved in many aspects of image analysis. The book does not attempt to cover all of the aspects of Computer Vision, but the chapters do present some state of the art examples. It also provides an indication to some new, original results.

We thank all authors and reviewers involved in the book preparation process. Their commitment has allowed us to complete the work. We hope that that this book will be helpful in the development of future research. Thanks are also due to Dr Neil Allen for the fruitful discussion during the development phase of this book.

Halina Kwaśnicka, Poland
Lakhmi C. Jain, Australia

# Editors

Halina Kwaśnicka received M.Sc and Ph.D. from the Wroclaw University of Technology, Poland. From the same University she received habilitation (Doctor of Science) in 2000. Nine years later she has been nominated by the president of Poland as a full professor (the highest scientific title in Poland).

She is currently the Deputy Director for Scientific Researches of Institute of Informatics, and the head of Division of Artificial Intelligence at the Faculty of Computer Science and Management, Wroclaw University of Technology.

At the beginning of her scientific career, as an assistant professor in the Futures Research Center, she has been engaged in research on forecasting methods and evolutionary computation. In 1998 she moved to the Institute of Informatics, where currently she is a professor, she teaches graduated and postgraduate students. Over time her interest has widened to nature inspired methods, data mining and knowledge based systems. During last decade the intelligent methods of images analysis becomes very important area of researches. She was or is a leader in a number of research projects on applications of artificial intelligence methods in medical images analysis in cooperation with Wroclaw Medical University. She is active in international cooperation, e.g., she is the coordinator of Polish-Singapore join research project entitled 'Framework for Visual Information Retrieval and Building Content-based Visual Search Engines' (http://www.ii.pwr.wroc.pl/~visible), it is realized in cooperation with NTU, School of Computer Engineering, Singapore.

She has published over 150 papers, five books (in Polish) and has presented a number of invited lectures. She is currently serving in editorial boards of several journals and many conferences. Besides reviewing the scientific works in Poland, she has been invited to review candidates for the award of tenure by NTU, Singapore, a candidate for the promotion to the rank of professor Applied Science Private University, Amman – Jordan, and PhD thesis (Louis Pasteur University, France; University of Melbourne, Australia; Otago University, New Zealand).

She has been a supervisor of over 50 master theses, promoted six Ph.D. students, and is a supervisor of next four Ph.D. students. She is also a mentor of Students Artificial Intelligence Club, where students are engaged in the automatic understanding of sign language.

Lakhmi C. Jain is a Director/Founder of the Knowledge-Based Intelligent Engineering Systems (KES) Centre, located in the University of South Australia. He is a fellow of the Institution of Engineers Australia.

His interests focus on the artificial intelligence paradigms and their applications in complex systems, art-science fusion, e-education, e-healthcare, unmanned air vehicles and intelligent agents.

# Contents

Chapter 9

Chapter 10

# Chapter 1
# Advances in Intelligent Image Analysis

Halina Kwaśnicka and Lakhmi C. Jain

**Abstract.** This chapter presents some recent advances in the area of computer vision. The various stages involved in the area of image processing and their interpretation are described. The first step is that of image registration. That is, the overlay two or more images of the same scene. These are taken from different viewpoints, at different times or possibly by different sensors. The next phase is image preprocessing. This mainly involves image enhancement and clearing for example. Other problem is that of image analysis. This is the extraction of important features of the image. Having obtained a description of the image, the process of object (pattern) recognition can be performed. All of these tasks are very important and useful, they still do not give a semantic interpretation of images. Image interpretation, similar image searching is still a major challenge facing researchers. The second part of this chapter summarises the remaining chapters of the book.

## 1 Introduction

One of a number of definitions of Artificial Intelligence is "the field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" [14]. The other claims that it is "the branch of computer science that

Halina Kwaśnicka
Deputy Director for Scientific Research
Institute of Informatics,
Wroclaw University of Technology
Wyb. Wyspianskiego 27
51-370 Wroclaw, Poland

Lakhmi C. Jain
School of Electrical and Information Engineering,
University of South Australia,
Adelaide,
Mawson Lakes Campus,
South Australia SA 5095,
Australia

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 1–6.
springerlink.com

attempts to emulate intelligent behavior by automation" [11]. The tremendous advances in artificial intelligence include learning possibilities, knowledge processing including imprecise knowledge for example. This work covers a very wide application area. One of the most important problems is knowledge acquisition from vast collections of data. Initially researchers considered only numerical and symbolic values of data to be of value. An important issue is that of text understanding. This includes information retrieval from text documents, automatic translators, for example. Other sources of data source are images. Human beings can easily see and interpret images. We are able to determine the emotions expressed by these images, and to correctly interpret a given image. For example whether that it presents danger or pleasure.

*Image processing* is a very vast, popular, and important field of studies. Processes arising from images are clearing, enhancing, repair of image understanding. Generally speaking, image processing is the manipulation of signals which are inherently multidimensional [8]. Often photographs (or other images) and video sequences are such signals. The main goals of image processing one can indicate: images compression; images enhancement or restoration; images recognition; images understanding; images visualization convenient for users. Different image processing techniques are useful in numerous areas which include medicine, video communication, astronomy, archeology, electronic games, and many others.

We will shortly describe a number of different problems which indicate the wide range of the image processing area.

Beginning with real objects, and commencing with objects need to be registered. Typically, *image registration* is essential in remote sensing. For example, environmental monitoring, integrating information into geographic information systems (GIS). In medicine for example, the combining of computer tomography (CT), NMR data to present a more complete information base on the patient, monitoring tumor growth, treatment verification; in cartography, and so on. Image registration is a process in which two or more images are captured of a same scene, taken from several different viewpoints, at different times. This information is captured using different sensors which may overlap. A review of recent and classic image registration methods is given in [16].

The next task in image processing procedure is *image preprocessing*. The methods used depend on the particular goal of the image processing [1], [6, 7]. The preprocessing phase is comprised of a series of operations, which are used to enhance this image. Very often a number of different filters are used in this process. The operations also suppress undesired noise data and enhances some image features that are important for further processing. The image processing methods used should be able to answer the question: of how best to increase the quality and visibility of the present image?

*Image analysis* or *image description* is another task in the image processing [4, 5]. The aim of this phase is the extraction of useful information from the processed digital image. Some methods such as detection of the edges, try to mimic human visual perception processes. These image analysis methods help us to answer the question: what are the exact values of selected features of the image? [15].

Having obtained important features of the image we can perform the *object recognition* task [13]. The main aim of this is to answer the question: which, if any, of a given set of objects (patterns) appear in the image considered or the image sequence. The object recognition problem can be defined as the problem of matching models from a given database. The representations of those models are extracted from the digital image. The representation of the object model plays very important role and a number of different approaches are studied [5], [7]. Recognition systems consist of two phases. The first is the construction of a model library from the descriptions of the given (known) objects. Then there is the recognition process. Here the system is presented together with a perspective image. This determines the location of the image and identity of any objects which may be located from a given library.

Performing all above tasks we can receive information concerning the names of all objects in that image. It is known that, often, images which are apparently similar can hide a semantically different content. The vice versa situation applies. So the very important problems arise: one is how to obtain and possess semantic knowledge from these images. It also means using a process of image understanding and interpretation.

From Tadeusiewicz [15] we can say that automatic image understanding allows us to answer a number of questions. These include: what follows from the visualized details? What is the meaning of the features extracted from the image? What follows from the existence of individual objects belonging to particular classes in the image? Proper interpretation of the images is vitally important. For example, in the problem of Medical Decision Support Systems. It seems that adequate image interpretation of the results may require some measure of prior knowledge. [15].

Two more concepts closely connected with the image processing field are: *Computer Vision* and *Machine Vision* [2]. According to the Free On-line Dictionary of Computing (http://dictionary.die.net) *Computer Vision* is the branch of Artificial Intelligence concerned computer processing of images obtained from real world. Typically it consists of low level processing and high level pattern recognition and image understanding of features present in the considered image. Narrowing, or using computer vision for application to factory automation is called *Machine Vision*. Machine vision systems can work in a similar manner to human inspectors. They visually inspect assembly lines to judge the quality of workmanship. Machine vision systems use digital cameras and image processing software to perform these inspections. A machine vision system is a computer system that can make decisions based on the analysis of digital images. A single machine vision system performs a narrowly defined task. For example, counting objects on a conveyor, reading serial numbers and searching for surface defects. Computer systems reveal many advantages in contrast to human inspectors. That is, they are liable to make fewer mistakes because they do not get sick, do not tire and work repeatedly. On the other hand, humans may display a finer perception over the short periods and have greater flexibility. They can adapt to new problems.

We realise that computers ‘see’ in another way than human beings. People use inference and prior knowledge or experience and can interpret images according to the context and the ultimate goal of the image analysis.

*Image retrieval* is another problem that has been intensively studied. It is associated with the concept of image similarity. We wish to develop a computer system which is able to find similar images to a given image. Similarity plays very important role. Viable similarity measure decides the efficacy of retrieving images having a related content. Each person can find a similar image to a given one. This image is similar in our view. Images similarity term is not precise. It is very subjective when used by various people. One can expect that such a computer system will be imprecise and, it possibly will need to be tuned for individual users. Accounting for the human perspective and expectations, the similar images retrieving is difficult, if at all possible [24]. One approach to image retrieval is application of object recognition paradigm with a similarity measure of the recognized concepts. Instead of low level image features based queries, the user is able to formulate a meaningful concept based queries [10]. This image retrieval scheme is sometimes referred as *Annotation Based Image Retrieval* [9]. This is in contrast to classic *Content Based Image Retrieval.* The key disadvantage of such approach is that the number of concepts is both predefined and is finite [12], [14]. The object recognition based paradigm is not applicable when it is faced with the infinite diversity of the surrounding world [3]. Effective image retrieval can require the continuous creation of new concepts.

## 2 Chapters Included in the Book

This book includes thirteen chapters. Chapter one gives an introduction to the image processing field and different problems connected with this area. Brief summaries of chapters included in the book are given.

Chapter 2 deals with multi-class classification using subclass problem-dependent error correcting output codes. The authors have presented a novel way in which to model complex multi-class classification problems. The technique has been validated and the improvement in performance is demonstrated.

Chapter 3 is about morphological operator design. The author gives a description of the translation invariant morphological operator learning. This uses training images supported by the sub-decomposition representation structure of these operators.

Chapter 4 is about an extended notion of salience for use in object recognition. The authors demonstrate the link between salience and cascaded processes. They show why and how these could be constructed.

Chapter 5 considers fast and efficient local features for the detection for use in image recognition. The authors presented a new method for the detection and filtering of local features. The technique is validated to prove its superiority.

Chapter 6 is about visual perception in image analysis process. The author introduces a form of perceptual image analysis using a methodology for determining the resemblance between pairs of visual tolerance spaces within the context of digital images.

Chapter 7 is about magnetic resonance (MR) imaging. An overview of the structural MR imaging is presented and a variety of applications for the diagnoses of neurological diseases are discussed.

Chapter 8 deals with the detection of similarities in the context of digital images. This approach is useful for image retrieval and in the solution of the image correspondence problem. What is important in this chapter is the near set approach to object recognition. Sets of objects X,Y are considered near each other if the sets contain objects with at least partial matching descriptions.

Chapter 9 gives a technique for the detection of near duplicate fragments in images having unknown contents. The technique is validated to demonstrate the superiority of this approach.

Chapter 10 is about feature analysis for object and scene categorization. It demonstrates that feature analysis is indispensable in constructing effective classifiers.

Chapter 11 introduces curve and edge parameterization by moments. Examples are used to demonstrate the applications in analytically-defined image functions, generated images, and real-world images.

Chapter 12 is about intelligent approaches to colour palette design. It demonstrates that intelligent approaches outperform the standard colour quantisation techniques in terms of their image quality.

Chapter 13 is about mean shift and its application in image segmentation. It is demonstrated that the application of a mean shift process can improve image segmentation algorithms.

## 3 Conclusion

This chapter presents a short overview of the use of computers for image processing and analysis. The main tasks and concepts connected with this area are described. Some new and important problems are underlined.

The book chapters do not cover the whole area of computer vision, but give an overview of research in this field. Particular chapters present the state of the art as well as the authors' researches and results.

The short presentation of problems connected with widely understood computer vision systems show that the results are far from satisfying the expectations and needs. A number of books, conferences, journal titles, and other publications dedicated to computer vision is now being developed but there is still a need for further research in the area. Maybe that close cooperation by specialists from different centers and different countries will accelerate the development of this fascinating and important field. We hope that this book will contribute and lead to closer cooperation between the various centers and researchers.

## References and Further Readings

[1] Bieniecki, W., Grabowski, S., Rozenberg, W.: Image Preprocessing for Improving OCR Accuracy. In: International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2007 (2007)

[2] Davies, E.R.: Machine Vision: Theory, Algorithms, Practicalities. Morgan Kaufmann, San Francisco (2004)

[3] Dickinson, S.J., Leonardis, A., Schiele, B., Tarr, M.J. (eds.): Object Categorization: Computer and Human Vision Perspectives. Cambridge University Press, Cambridge (2009)

[4] O'Gorman, L., Sammon, M.J., Seul, M.: Cambridge University Press, Cambridge (2008)

[5] Häder, D.P.: Image analysis: methods and applications. CRC Press, Boca Raton (2001)

[6] van der Heide, A., Urano, T., Polderdijk, F., de Haan, W., Bosiers, J.T.: IDEAL: An image pre-processing architecture for high-end professional DSC applications. In: SPIE Electronic Imaging 2009, pp. 7250–7238 (2009)

[7] Heusch, G., Rodriguez, Y., Marcel, S.: Local Binary Patterns as an Image Preprocessing for Face Authentication. In: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR 2006) (2006)

[8] Huang, T.S., Aizawa, K.: Image Processing: Some Challenging Problems. PNAS 90, 9766–9769 (1993)

[9] Inoue M.: On the need for annotation-based image retrieval. In: Proceedings of the Information Retrieval in Context (IRiX), A Workshop at SIGIR 2004, pp. 44–46 (2004)

[10] Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: Proceedings of Neural Information Processing Systems (NIPS). MIT Press, Cambridge (2003)

[11] Luger, G.F., Stubblefield, W.A.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 2nd edn. Benjamin Cummings, Palo Alto (1993)

[12] Maier, O., Stanek, M., Kwaśnicka, H.: PATSI - photo annotation through similar images with annotation length optimization. In: Kłopotek, M.A. et al. (eds.) Intelligent information systems, pp. 219–232. Publishing House of University of Podlasie (2010)

[13] Mian, A.S., Bennamoun, M., Owens, R.A.: An efficient multimodal 2D-3D hybrid approach to automatic face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11), 1927–1943 (2007)

[14] Robert, J., Schalkoff, R.J.: Artificial Intelligence: An Engineering Approach. McGraw-Hill College, New York (1990)

[15] Stanek, M., Broda, B., Kwasnicka, H.: PATSI — photo annotation through finding similar images with multivariate gaussian models. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6375, pp. 284–291. Springer, Heidelberg (2010)

[16] Tadeusiewicz, R., Ogiela, M.: Medical Image Understanding Technology. Springer-Verlag GmbH (2004)

[17] Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21, 977–1000 (2003)

# Chapter 2
# Multi-class Classification in Image Analysis via Error-Correcting Output Codes

Sergio Escalera, David M.J. Tax, Oriol Pujol, Petia Radeva, and Robert P.W. Duin

**Abstract.** A common way to model multi-class classification problems is by means of Error-Correcting Output Codes (ECOC). Given a multi-class problem, the ECOC technique designs a codeword for each class, where each position of the code identifies the membership of the class for a given binary problem. A classification decision is obtained by assigning the label of the class with the closest code. In this paper, we overview the state-of-the-art on ECOC designs and test them in real applications. Results on different multi-class data sets show the benefits of using the ensemble of classifiers when categorizing objects in images.

Sergio Escalera
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
e-mail: `sergio@maia.ub.es`

David M.J. Tax
Information and Communication Theory (ICT) Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, PO Box 5031, 2600 GA, Delft, The Netherlands
e-mail: `d.m.j.tax@gmail.com`

Oriol Pujol
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
e-mail: `oriol@maia.ub.es`

Petia Radeva
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
e-mail: `petia.ivanova@ub.edu`

Robert P.W. Duin
Information and Communication Theory (ICT) Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, PO Box 5031, 2600 GA, Delft, The Netherlands
e-mail: `r.duin@ieee.org`

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 7–29.
springerlink.com

## 1 Introduction

In the literature, one can find several powerful binary classifiers. However, when one needs to deal with multi-class classification problems, many learning techniques fail to manage this information. Instead, it is common to construct the classifiers to distinguish between just two classes, and to combine them in some way. In this sense, Error Correcting Output Codes (ECOC) were born as a general framework to combine binary problems to address the multi-class problem. The strategy was introduced by Dietterich and Bakiri [17] in 1995. Based on the error correcting principles [17] and because of its ability to correct the bias and variance errors of the base classifiers [16], ECOC has been successfully applied to a wide range of image analysis applications, such as face recognition [30], face verification [15], text recognition [11] or manuscript digit classification [32].

The ECOC technique can be broken down into two distinct stages: encoding and decoding. Given a set of classes, the coding stage designs a codeword[1] for each class based on different binary problems. The decoding stage makes a classification decision for a given test sample based on the value of the output code.

At the coding step, given a set of $N$ classes to be learnt, $n$ different bi-partitions (groups of classes) are formed, and $n$ binary problems (dichotomizers) are trained. As a result, a codeword of length $n$ is obtained for each class, where each bit of the code corresponds to the response of a given dichotomizer (coded by +1, -1, according to their class set membership). Arranging the codewords as rows of a matrix, we define a *coding matrix* $M$, where $M \in \{-1,1\}^{N \times n}$ in the binary case.

It was when Allwein et al. [1] introduced a third symbol (the zero symbol) in the coding process when the coding step received special attention. This symbol increases the number of partitions of classes to be considered in a ternary ECOC framework by allowing some classes to be ignored. Then, the ternary coding matrix becomes $M \in \{-1,0,1\}^{N \times n}$. In this case, the symbol zero means that a particular class is not considered by a certain binary classifier.

The decoding step was originally based on error-correcting principles under the assumption that the learning task can be modeled as a communication problem, in which class information is transmitted over a channel [17]. During the decoding process, applying the $n$ binary classifiers, a code is obtained for each data point in the test set. This code is compared to the base codewords of each class defined in the matrix $M$, and the data point is assigned to the class with the *closest* codeword.

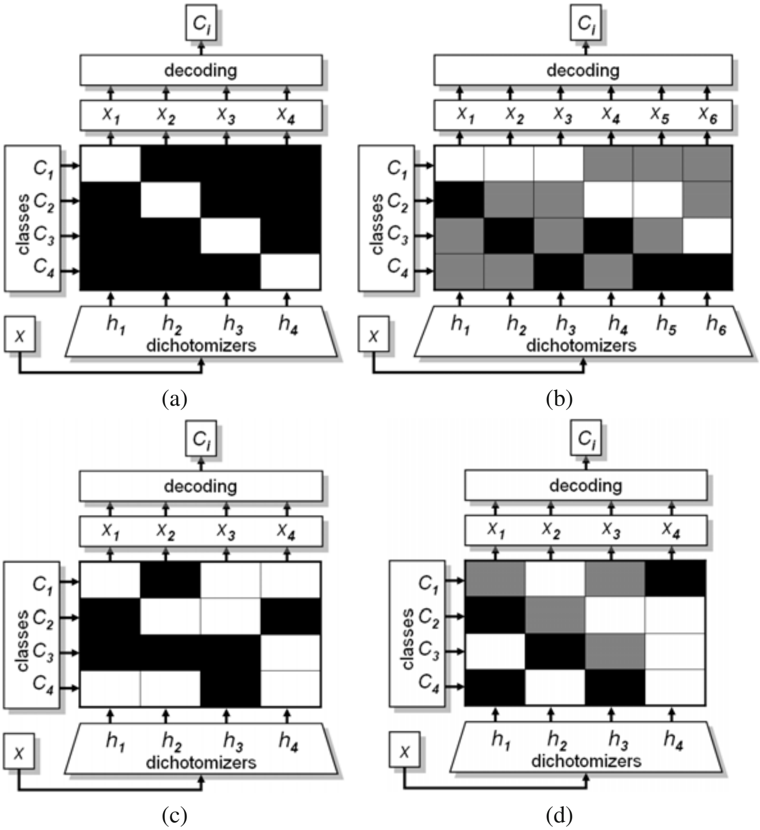
In this paper, we overview the state-of-the-art ECOC designs. We describe the different ECOC strategies available from both the binary and the ternary ECOC frameworks. We analyze the complexity in terms of the number of classifiers and test the designs on different multi-class data sets, showing the benefits of using the ensemble of classifiers when categorizing objects in images.

The paper is organized as follows: Section 2 overview the state-of-the-art ECOC designs. Section 3 shows the experimental results on two challenging image categorization problems. Finally, section 4 concludes the paper.

[1] The codeword is a sequence of bits of a code representing each class, where each bit identifies the membership of the class for a given binary classifier.

## 2 ECOC Designs

The most well-known binary coding strategies are the one-versus-all strategy [19], where each class is discriminated against the rest of classes, and the dense random strategy [1], where a random matrix $M$ is generated maximizing the rows and columns separability in terms of the Hamming distance [17]. In Fig. 1(a), the one-versus-all ECOC design for a 4-class problem is shown. The white regions of the coding matrix $M$ correspond to the positions coded by 1, and the black regions to -1. Thus, the codeword for class $C_1$ is $\{1,-1,-1,-1\}$. Each column $i$ of the coding matrix codifies a binary problem learned by its corresponding dichotomizer $h_i$. For instance, dichotomizer $h_1$ learns $C_1$ against classes $C_2, C_3$ and $C_4$, dichotomizer $h_2$ learns $C_2$ against classes $C_1, C_3$ and $C_4$, etc. An example of a dense random matrix for a 4-class problem is shown in Fig. 1(c).



(a) (b) (c) (d)

**Fig. 1** One-versus-all (a), one-versus-one (b), dense random (c), and (d) sparse random ECOC designs.

The classical coding designs in the ternary ECOC framework are the one-versus-one [13] and the random sparse coding [1]. Fig. 1(b) shows the one-versus-one ECOC configuration for a 4-class problem. In this case, the grey positions correspond to the zero symbol. A possible sparse random matrix for a 4-class problem is shown in Fig. 1(d). Note that the previous coding designs are predefined. Thus, the training data is not considered until the coding matrix $M$ is constructed. Then, each dichotomizer uses the coded positions of $M$ to train the different binary problems.

The most frequently applied decoding strategies are the Hamming ($HD$) [19] and the Euclidean ($ED$) decoding distances [13]. With the introduction of the zero symbol, Allwein et al. [1] showed the advantage of using a loss based function of the margin of the output of the base classifier. Recently, the authors of [9] proposed a Loss-Weighted strategy to decode, where a set of probabilities based on the performances of the base classifiers are used to weight the final classification decision. In Fig. 1, each ECOC codification is used to classify an input object $X$. The input data sample $X$ is tested with each dichotomizer $h_i$, obtaining an output $X_i$. The final code $\{X_1,...,X_n\}$ of the test input $X$ is used by a given decoding strategy to obtain the final classification decision. Note that in both, the binary and the ternary ECOC framework, the value of each position $X_j$ of the test codeword can not take the value zero since the output of each dichotomizer $h_j \in \{-1,+1\}$.

Recently, new improvements in the ternary ECOC coding demonstrate the suitability of the ECOC methodology to deal with multi-class classification problems [27][26][10][29][5]. These recent designs use the knowledge of the problem-domain to learn relevant binary problems from ternary codes. The basic idea of these methods is to use the training data to guide the training process, and thus, to construct the coding matrix $M$ focusing on the binary problems that better fits the decision boundaries of a given data set. The definition of new coding designs also motivated the design of novel decoding methodologies [9]. Next, we describe some of these recent designs: ECOC-ONE, DECOC, Sub-class ECOC coding, and Loss-Weighted decoding.

## 2.1 ECOC-ONE Coding

ECOC-Optimal Node Embedding defines a general procedure capable of extending any coding matrix by adding dichotomizers based on a discriminability criterion. In the case of a multiclass recognition problem, the procedure starts with a given ECOC coding matrix. Then, this ECOC matrix is increased in an iterative way, adding dichotomizers that correspond to different sub-partitions of classes. These partitions are found using greedy optimization based on the confusion matrices so that the ECOC accuracy improves on both training and validation sets. The training set guides the convergence process, and the validation set is used to avoid overfitting and to select a configuration of the learning procedure that maximizes the generalization performance [17]. Since not all problems require the same dichotomizers structure -in form of sub-partitions-, the optimal node embedding approach

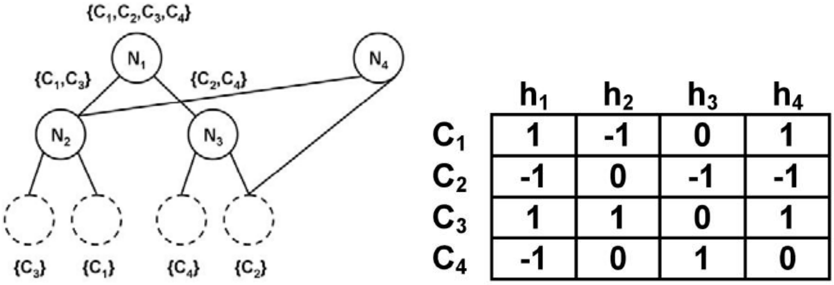
**Table 1** ECOC-ONE general algorithm

```
Given $N_c$ classes and a coding matrix $M$:

while $error > \varepsilon$ or $error_t < error_{t-1}, t \in [1,T]$:

    Compute the optimal node $t$:

    1) Test accuracy on the training and validation sets $S_t$ and $S_v$.

    2) Select the pair of classes with the highest error analyzing the confusion matrices from $S_t$ and $S_v$.

    3) Find the partition that minimizes the error rate in $S_t$ and $S_v$.

    4) Compute the weight for the dichotomy of the partition based on its classification score.

Update the matrix $M$.
```

generates an optimal ECOC-ONE matrix dependent on the hypothesis performance in a specific problem domain.

Table 1 shows the summarized steps for the ECOC-ONE approach. Note that, the process described is iterated while the error on the training subsets is greater than $\varepsilon$ or the number of iterations $i \leq T$. An example of an ECOC-ONE strategy applied to a four-class classification example can be found in Fig. 2. The initial optimal tree corresponds to the dichotomizers of optimal sub-partition of the classes. This tree has been generated using accuracy as a sub-partition splitting criterion. After testing the performance of the ensemble tree (composed by the columns $\{h_1, h_2, h_3\}$ of the ECOC matrix $M$ of Fig. 2(b)), let assume that classes $\{C_2, C_3\}$ get maximal error in the confusion matrices $\nu_t$ and $\nu_v$. We search for the sub-partition of classes using the training and validation subsets so that the error between $\{C_2, C_3\}$ and all previous misclassified samples is minimized. Suppose now that this sub-partition is $\{C_1, C_3\}$ versus $\{C_2\}$. As a result, a new node $N_4$ corresponding to dichotomy $h_4$ is created. We can observe in Fig. 2 that $N_4$ uses a class partition that is present in the tree. In this sense, this new node connects two different nodes of the tree. Note that using the previously included dichotomizers, the partition $\{C_1, C_3\}$ is solved by $N_2$. In this way, the Hamming distance between $C_2$ and $C_3$ is increased by adding the new dichotomy to the whole structure. At the same time, the distance among the rest of the classes is usually maintained or slightly modified.

One of the desirable properties of the ECOC matrix is to have maximal distance between rows. The ECOC-ONE procedure focuses on the relevant difficult partitions, increasing the distance between ”close” classes. This fact improves the robustness of the method since difficult classes are likely to have a greater number of dichotomizers centered on them. In this sense, it creates different geometrical



|  | $h_1$ | $h_2$ | $h_3$ | $h_4$ |
|---|---|---|---|---|
| $C_1$ | 1 | -1 | 0 | 1 |
| $C_2$ | -1 | 0 | -1 | -1 |
| $C_3$ | 1 | 1 | 0 | 1 |
| $C_4$ | -1 | 0 | 1 | 0 |

(a) (b)

**Fig. 2** (a) Optimal tree and first optimal node embedded, (b) ECOC-ONE code matrix $M$ for four dichotomizers.

arrangements of decision boundaries, and leads the dichotomizers to make different bias errors.

## *2.2 DECOC and Sub-class ECOC Coding*

One of the main reasons why the recent problem-dependent designs [27][26][10] attains a good performance is because of the high number of possible sub-groups of classes that is possible in the ternary ECOC framework. On the other hand, using the training data in the process of the ECOC design allows to obtain compact codewords with high classification performance. However, the final accuracy is still based on the ability of the base classifier to learn each individual problem. Difficult problems, those which the base classifier is not able to find a solution for, require the use of complex classifiers, such as Support Vector Machines with Radial Basis Function kernel [23], and expensive parameter optimizations. Look at the example of Fig. 3(a). A linear classifier is used to split two classes. In this case, the base classifier is not able to find a convex solution. On the other hand, in Fig. 3(b), one of the previous classes has been split into two sub-sets, that we call *sub-classes*. Then, the original problem is solved using two linear classifiers, and the two new sub-classes have the same original class label. Some studies in the literature tried to form sub-classes using the labels information, which is called Supervised Clustering [31][7]. In these types of systems, clusters are usually formed without taking into account the behavior of the base classifier that learns the data. In a recent work [33], the authors use the class labels to form the sub-classes that improve the performance of particular Discriminant Analysis algorithms.

From an initial set of classes $C$ of a given multi-class problem, the objective of the Sub-class ECOC strategy is to define a new set of classes $C'$, where $|C'| > |C|$, so that the new set of binary problems is easier to learn for a given base classifier. For this purpose, the approach uses a guided procedure that, in a problem-dependent way, groups classes and splits them into sub-sets if necessary.

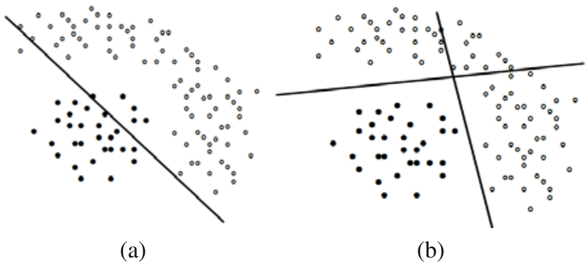


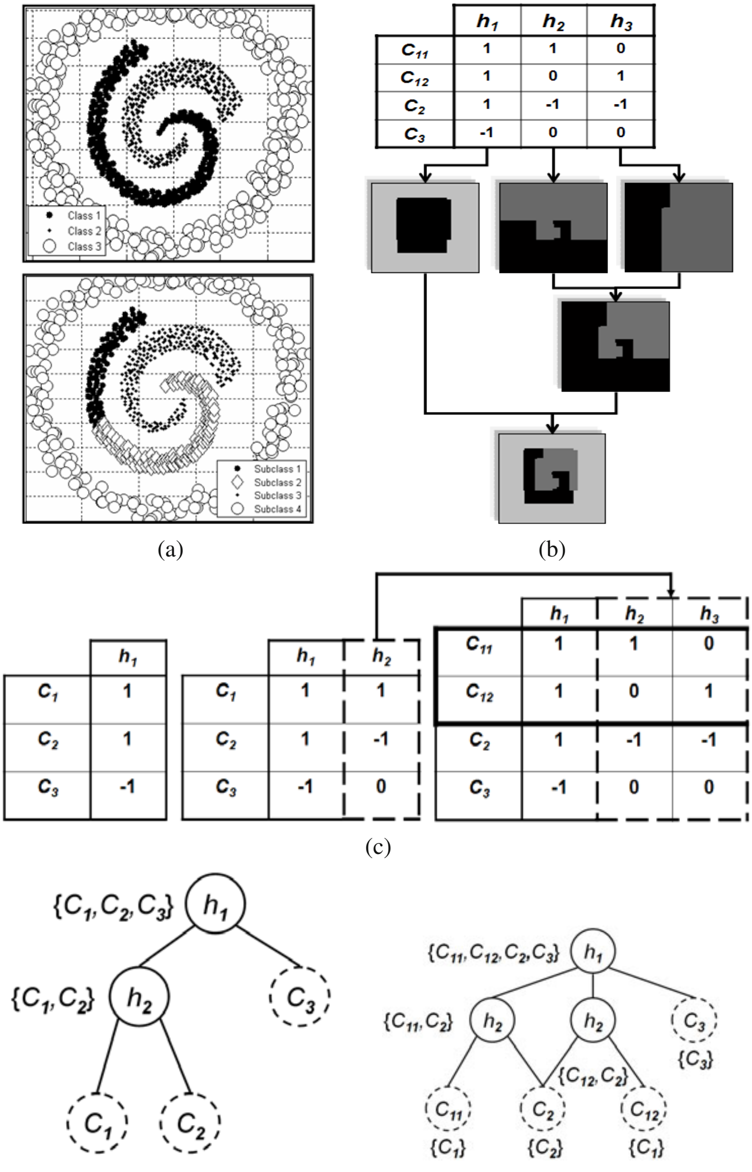
**Fig. 3** (a) Decision boundary of a linear classifier of a 2-class problem. (b) Decision boundaries of a linear classifier splitting the problem of (a) into two more simple tasks.

Recently, the authors of [27] proposed a ternary problem-dependent design of ECOC, called DECOC, where given $N$ classes, a high classification performance is achieved with only $N-1$ binary problems. The method is based on the embedding of discriminant tree structures derived from the problem domain. The binary trees are built by looking for the partition that maximizes the mutual information (*MI*) between the data and their respective class labels. Look at the 3-class problem shown on the top of Fig. 4(a). The standard DECOC algorithm considers the whole set of classes to split it into two sub-sets of classes $\wp^+$ and $\wp^-$ maximizing the *MI* criterion on a sequential forward floating search procedure (*SFFS*). In the example, the first sub-sets found correspond to $\wp^+ = \{C_1, C_2\}$ and $\wp^- = \{C_3\}$. Then, a base classifier is used to train its corresponding dichotomizer $h_1$. This classifier is shown in the node $h_1$ of the tree structure shown in Fig. 4(d). The procedure is repeated until all classes are split into separate sub-sets $\wp$. In the example, the second classifier is trained to split the sub-sets of classes $\wp^+ = C_1$ from $\wp^- = C_2$ because the classes $C_1$ and $C_2$ were still contained in a single sub-set after the first step. This second classifier is codified by the node $h_2$ of Fig. 4(d). When the tree is constructed, the coding matrix $M$ is obtained by codifying each internal node of the tree as a column of the coding matrix (see Fig. 4(c)).

In the case of Sub-class ECOC, sequential forward floating search (*SFFS*) is also applied to look for the sub-sets $\wp^+$ and $\wp^-$ that maximizes the mutual information between the data and their respective class labels [27].

Given a $N$-class problem, the whole set of classes is used to initialize the set $L$ containing the sets of labels for the classes to be learned. At the beginning of each iteration $k$ of the algorithm, the first element of $L$ is assigned to $S_k$ in the first step of the algorithm. Next, $SFFS$ [25] is used to find the optimal binary partition $BP$ of $S_k$ that maximizes the mutual information $I$ between the data and their respective class labels.

To illustrate the procedure, let us return to the example of the top of Fig. 4(a). On the first iteration of the sub-class ECOC algorithm, $SFFS$ finds the sub-set $\wp^+ = \{C_1, C_2\}$ against $\wp^- = \{C_3\}$. The encoding of this problem is shown in the first matrix of Fig. 4(c). The positions of the column corresponding to the classes of the first partition are coded by +1 and the classes corresponding to the second partition to -1, respectively. In the Sub-class procedure, the base classifier is used to test



(a) (b)

(c)

(d) (e)

**Fig. 4** (a) Top: Original 3-class problem. Bottom: 4 sub-classes found. (b) Sub-class ECOC encoding using the four sub-classes using Discrete Adaboost with 40 runs of Decision Stumps. (c) Learning evolution of the sub-class matrix *M*. (d) Original tree structure without applying sub-class. (e) New tree-based configuration using sub-classes.

if the performance obtained by the trained dichotomizers is sufficient. Observe the decision boundaries of the picture next to the first column of the matrix in Fig. 4(b). One can see that the base classifier finds a good solution for this first problem.

Then, the second classifier is trained to split $\wp^+ = C_1$ against $\wp^- = C_2$, and its performance is computed. To separate the current sub-sets is not a trivial problem, and the classification performance is poor. Therefore, the procedure tries to split the data $J_{\wp^+}$ and $J_{\wp^-}$ from the current sub-sets $\wp^+$ and $\wp^-$ into more simple sub-sets. Then, the splitting criteria $SC$ takes as input a data set $J_{\wp^+}$ or $J_{\wp^-}$ from a sub-set $\wp^+$ or $\wp^-$, and splits it into two sub-sets $J^+_{\wp^+}$ and $J^-_{\wp^+}$ or $J^+_{\wp^-}$ and $J^-_{\wp^-}$.

When two data sub-sets $\{J^+_{\wp^+}, J^-_{\wp^+}\}$ and $\{J^+_{\wp^-}, J^-_{\wp^-}\}$ are obtained, only one of both split sub-sets is used. The selected sub-sets are those that have the highest distance between the means of each cluster. Suppose that the distance between $J^+_{\wp^-}$ and $J^-_{\wp^-}$ is larger than between $J^+_{\wp^+}$ and $J^-_{\wp^+}$. Then, only $J_{\wp^+}$, $J^+_{\wp^-}$, and $J^-_{\wp^-}$ are used. If the new sub-sets improve the classification performance, new sub-classes are formed, and the process is repeated.

In the example of Fig. 4, applying the splitting criteria $SC$ over the two sub-sets, two clusters are found for $\wp^+ = C_1$ and for $\wp^- = C_2$. Then, the original encoding of the problem $C_1$ vs $C_2$ (corresponding to the second column of the matrix in the center of Fig. 4(c)) is split into two columns marked with the dashed lines in the matrix on the right. In this way, the original $C_1$ vs $C_2$ problem is transformed to two more simple problems $\{C_{11}\}$ against $\{C_2\}$ and $\{C_{12}\}$ against $\{C_2\}$. Here the first subindex of the class corresponds to the original class, and the second subindex to the number of sub-class. It implies that the class $C_1$ is split into two sub-classes (look at the bottom of Fig. 4(a)), and the original 3-class problem $C = \{C_1, C_2, C_3\}$ becomes the 4-sub-class problem $C' = \{C_{11}, C_{12}, C_2, C_3\}$. As the class $C_1$ has been decomposed by the splitting of the second problem, we need to save the information of the current sub-sets and the previous sub-sets affected by the new splitting. For this purpose, we use the object labels to define the set of sub-classes of the current partition $\wp_c$. If new sub-classes are created, the set of sub-classes $C'$ and the data for sub-classes $J'$ have to be updated. Note that when a class or a sub-class previously considered for a given binary problem is split in a future iteration of the procedure, the labels from the previous sub-sets $\{\wp^+, \wp^-\}$ need to be updated with the new information. Finally, the set of labels for the binary problems $\wp'$ is updated with the labels of the current sub-set $\wp' = \wp' \cup \wp_c$. In the example of Fig. 4, the dichotomizer $h_1$ considers the sub-sets $\wp_1^+ = \{C_1, C_2\}$ and $\wp_1^- = \{C_3\}$. Then, those positions containing class $C_1$ are replaced with $C_{11}$ and $C_{12}$. The process is repeated until the desired performance is achieved or the stopping conditions are full-filled.

The conditions that guide the learning and splitting process are defined by the set of parameters $\theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$, where $\theta_{size}$ corresponds to the minimum size of a sub-set to be clustered, $\theta_{perf}$ contains the minimum error desired for each binary problem, and $\theta_{impr}$ looks for the improvement of the split sub-sets regarding the previous ones. In the example of Fig. 4, the three dichotomizers $h_1$, $h_2$, and $h_3$ find a solution for the problem (look the trained boundaries shown in Fig. 4(b)), obtaining a classification error under $\theta_{perf}$, so, the process stops. Now, the original

**Table 2** Problem-dependent Sub-class ECOC algorithm.

```
Inputs: $J,C,\theta=\{\theta_{size},\theta_{perf},\theta_{impr}\}$ //Thresholds for the number of samples, performance, and improvement between iterations
Outputs: $C',J',\wp',M$
[Initialization:]
Create the trivial partition $\{\wp_0^+,\wp_0^-\}$ of the set of classes $\{C_i\}$: $\{\wp_0^+,\wp_0^-\}=\{\{\emptyset\},\{C_1,C_2,...,C_N\}\}$
$L_0=\{\wp_0^-\};J'=J;C'=C;\wp'=\emptyset;M=\emptyset;k=1$
Step 1 $S_k$ is the first element of $L_{k-1}$
    $L'_k=L_{k-1}\backslash\{S_k\}$
Step 2 Find the optimal binary partition $BP(S_k)$:
    $\{\wp_k^+,\wp_k^-\}=argmax_{BP(S_k)}(I(\mathbf{x},d(BP(S_k))))$
    where $I$ is the mutual information criterion, $\mathbf{x}$ is the random variable associated to the features and $d$ is the discrete random variable of the dichotomy labels[a], defined in the following terms,
    $d=d(\mathbf{x},BP(S_k))=\begin{cases}1 & \text{if } \mathbf{x}\in C_i|C_i\in\wp_k^+\\ -1 & \text{if } \mathbf{x}\in C_i|C_i\in\wp_k^-\end{cases}$
Step 3 // Look for sub-classes
    $\{C',J',\wp'\}=SPLIT(J_{p_k^+},J_{p_k^-},C',J',J,\wp',\theta)$[b]
Step 4 $L_k=\{L'_k\cup\wp_k^i\}$ if $|\wp_k^i|>1\ \forall i\in\{+,-\}$
Step 5 If $|L_k|\neq 0$
    $k=k+1$ go to Step 1
Step 6 Codify the coding matrix $M$ using each partition $\{\wp_i^+,\wp_i^-\}$ of $\wp',i\in[1,..,|\wp'|]$ and each class $C_r\in\wp_i=\{\wp_i^+\cup\wp_i^-\}$ as follows:
    $M(C_r,i)=\begin{cases}0 & \text{if } C_r\notin\wp_i\\ +1 & \text{if } C_r\in\wp_i^+\\ -1 & \text{if } C_r\in\wp_i^-\end{cases}$    (1)
```

[a] Use $SFFS$ of [25] as the maximization procedure and $MI$ of [27] to estimate $I$.

[b] Using the splitting algorithm of table 3.

tree encoding of the DECOC design shown in Fig. 4(d) can be represented by the tree structure of Fig. 4(e), where the original class associated to each sub-class is shown in the leaves. The algorithms summarizing the subclass approach and the splitting methodology are shown in tables 2 and 3, respectively.

Summarizing, when a set of objects belonging to different classes is split, object labels are not taken into account. It can be seen as a clustering in the sense that the sub-sets are split into more simple ones while the splitting constraints are satisfied. It is important to note that when one uses different base classifiers, the sub-class splitting is probably applied to different classes or sub-classes, and therefore, the final number of sub-classes and binary problems differs.

Finally, to decode the new sub-class problem-dependent design of ECOC, the authors use the recently proposed Loss-Weighted decoding design described in the next section.

**Table 3** Sub-class *SPLIT* algorithm.

```
Inputs: $J_{\wp^1}, J_{\wp^2}, C', J', J, \wp', \theta$ // $C'$ is the final set of classes, $J'$ the data for the final set of classes, and $\wp'$ is the labels for all the partitions of classes of the final set.
Outputs: $C', J', \wp'$
Step 1 Split problems:
$\{J^+_{\wp^+}, J^-_{\wp^+}\} = SC(J_{\wp^+})^a$
$\{J^+_{\wp^-}, J^-_{\wp^-}\} = SC(J_{\wp^-})$
Step 2 Select sub-classes:
if $|\overline{J^+_{\wp^+}}, \overline{J^-_{\wp^+}}| > |\overline{J^+_{\wp^-}}, \overline{J^-_{\wp^-}}|$ // find the largest distance between the means of each sub-set.
    $\{J^+_+, J^-_+\} = \{J^+_{\wp^+}, J_{\wp^-}\}$; $\{J^+_-, J^-_-\} = \{J^-_{\wp^+}, J_{\wp^-}\}$
else
    $\{J^+_+, J^-_+\} = \{J^+_{\wp^-}, J_{\wp^+}\}$; $\{J^+_-, J^-_-\} = \{J^-_{\wp^-}, J_{\wp^+}\}$
end
Step 3 Test parameters to continue splitting:
if $TEST\_PARAMETERS(J_{\wp^1}, J_{\wp^2}, J^1_1, J^2_1, J^1_2, J^2_2, \theta)$// call the function with the new sub-sets
    $\{C', J', \wp'\} = SPLIT(J^1_1, J^2_1, C', J', J, \wp', \theta)$
    $\{C', J', \wp'\} = SPLIT(J^1_2, J^2_2, C', J', J, \wp', \theta)$
end
Step 4 Save the current partition:
Update the data for the new sub-classes and previous sub-classes if intersections exists $J'$.
Update the final number of sub-classes $C'$.
Create $\wp_c = \{\wp_{c^1}, \wp_{c^2}\}$ the set of labels of the current partition.
Update the labels of the previous partitions $\wp$.
Update the set of partitions labels with the new partition $\wp' = \wp' \cup \wp_c$.
```

[a] *SC* corresponds to the splitting method of the input data into two main clusters.

To show the effect of the Sub-class ECOC strategy for different base classifiers, we used the previous toy problem of the top of Fig. 4(a). Five different base classifiers are applied: Fisher Linear Discriminant Analysis (*FLDA*), Discrete Adaboost, Nearest Mean Classifier, Linear *SVM*, and *SVM* with Radial Basis Function kernel. Using these base classifiers on the toy problem, the original DECOC strategy with the Loss-Weighted algorithm obtains the decision boundaries shown on the top row of Fig. 5. The new learned boundaries are shown on the bottom row of Fig. 5 for fixed parameters $\theta$. Depending on the flexibility of the base classifier more sub-classes are required, and thus, more binary problems. Observe that all base classifiers are able to find a solution for the problem, although with different types of decision boundaries.

## 2.3 Loss-Weighted Decoding

The Loss-Weighted decoding is defined as a combination of normalized probabilities to adapt the decoding to both binary and ternary ECOC frameworks. The

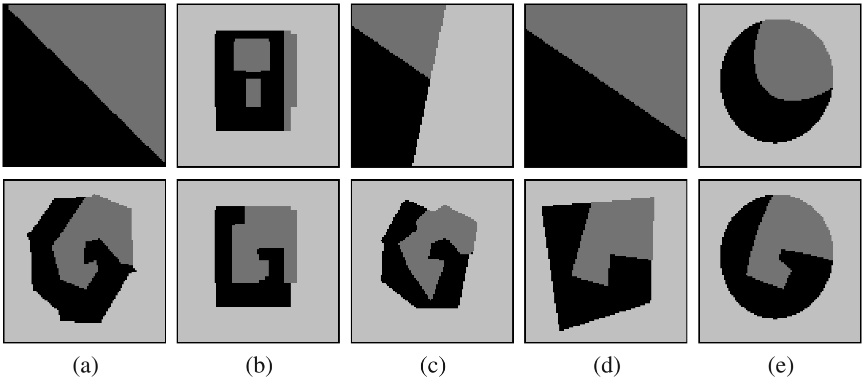


**Fig. 5** Sub-class ECOC without sub-classes (top) and including sub-classes (bottom): for *FLDA* (a), Discrete Adaboost (b), *NMC* (c), Linear *SVM* (d), and *RBF SVM* (e).

properties of the decoding strategy are encoded in a matrix that is used to weight the decoding process. Moreover, as not all the hypotheses have the same performance on learning the data samples, the accuracy of each binary problem is used to adjust the final classification decision.

A weight matrix $M_W$ is defined by assigning to each position of the codeword codified by $\{-1,+1\}$ a weight of $\frac{1}{n-z}$, where $z$ is the number of positions codified by zero. We assign to each position $(i,j)$ of a performance matrix $H$ a continuous value that corresponds to the performance of the dichotomizer $h_j$ classifying the samples of class $C_i$ as follows:

$$H(i,j) = \frac{1}{m_i}\sum_{k=1}^{m_i} \varphi(h^j(\rho_k^i),i,j), \quad \text{based on} \quad \varphi(x^j,i,j) = \begin{cases} 1, & \text{if} \quad x^j = y_i^j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $m_i$ are the number of samples of class $C_i$, $\rho$ is a test sample, and $x^i$ and $y^i$ are the $j$-th position of a test and a class codeword, respectively. Note that eq.(2) makes $H$ to have zero probability at those positions corresponding to unconsidered classes.

We normalize each row of the matrix $H$ so that $M_W$ can be considered as a discrete probability density function:

$$M_W(i,j) = \frac{H(i,j)}{\sum_{j=1}^{n} H(i,j)}, \quad \forall i \in [1,\ldots,N], \quad \forall j \in [1,\ldots,n] \tag{3}$$

In Fig. 6, a weight matrix $M_W$ for a 3-multi-class problem of four hypotheses is estimated. Figure 6(a) shows the coding matrix $M$. The matrix $H$ of Fig. 6(b) represents the accuracy of the hypotheses classifying the instances of the training set. The normalization of $H$ results in a weight matrix $M_W$ shown in Fig. 6(c).

Once we obtain the weight matrix $M_W$, we introduce the weight matrix in the Loss-based decoding. The decoding estimation is obtained by means of an *ELB* decoding model $L(\partial) = \mathbf{e}^{-\partial}$, where $\partial$ corresponds to $y_i^j \cdot f(\rho,j)$, weighted using $M_W$:

$$M=\begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & -1 \end{bmatrix} \quad H=\begin{bmatrix} 0.955 & 0.955 & 1.000 & 0.000 \\ 0.900 & 0.800 & 0.000 & 0.000 \\ 1.000 & 0.905 & 0.805 & 0.805 \end{bmatrix} \quad M_W=\begin{bmatrix} 0.328 & 0.328 & 0.344 & 0.000 \\ 0.529 & 0.471 & 0.000 & 0.000 \\ 0.285 & 0.257 & 0.229 & 0.229 \end{bmatrix}$$

(a) (b) (c)

**Fig. 6** (a) Coding matrix $M$ of four hypotheses for a 3-class problem. (b) Performance matrix $H$. (c) Weight matrix $M_W$.

$$LW(\rho,i)=\sum_{j=1}^{n} M_W(i,j)L(y_i^j\cdot f(\rho,j)) \tag{4}$$

The summarized algorithm is shown in table 4.

**Table 4** Loss-Weighted algorithm.

**Loss-Weighted strategy**: Given a coding matrix $M$,

1) Calculate the performance matrix $H$:

$$H(i,j)=\frac{1}{m_i}\sum_{k=1}^{m_i}\varphi(h^j(\rho_k^i),i,j) \text{ based on } \varphi(x^j,i,j)=\begin{cases}1, & \text{if } x^j=y_i^j,\\ 0, & \text{otherwise.}\end{cases} \tag{5}$$

2) Normalize $H$: $\sum_{j=1}^{n} M_W(i,j)=1, \quad \forall i=1,\dots,N$:

$$M_W(i,j)=\frac{H(i,j)}{\sum_{j=1}^{n}H(i,j)}, \quad \forall i\in[1,\dots,N], \quad \forall j\in[1,\dots,n] \tag{6}$$

3) Given a test data sample $\rho$, decode based on:

$$LW(\rho,i)=\sum_{j=1}^{n} M_W(i,j)L(y_i^j\cdot f(\rho,j)) \tag{7}$$

## 3 Experimental Results

In this section, we test the state-of-the-art ECOC configurations on two challenging image analysis applications: a real 9-class traffic sign classification problem from the Geomobil project of [4] and Intravascular Ultrasound Tissue Characterization.

## 3.1 Traffic Sign Categorization

For this experiment, we use the video sequences obtained from the Mobile Mapping System [4] to test a real traffic sign categorization problem. We choose the speed data set since the low resolution of the image, the non-controlled conditions, and the high similarity among classes make the image categorization a difficult task. In this system, the position and orientation of the different traffic signs are measured with fixed video cameras in a moving vehicle. The system has a stereo pair of calibrated cameras, which are synchronized with a GPS/INS system. The result of the acquisition step is a set of stereo-pairs of images with their position and orientation information. Fig. 7 shows several samples of the speed data set used for the experiments. The data set contains a total of 2500 samples divided into nine classes. Each sample is composed by 1200 pixel-based features after smoothing the image and applying a histogram equalization. From this original feature space, about 150 features are derived using a *PCA* that retained 90% of the total variance.

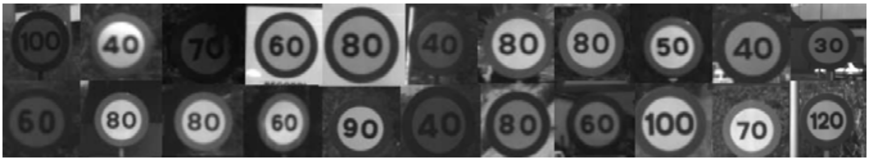


**Fig. 7** Speed data set samples.

The performance and the estimated ranks using the different ECOC strategies for the different base classifiers are shown in table 5. These results are also illustrated in the graphics of Fig. 8. The random matrices were selected from a set of 20000 randomly generated matrices, with $P(1) = P(-1) = 0.5$ for the dense random matrix and $P(1) = P(-1) = P(0) = 1/3$ for the sparse random matrix. The number of binary problems was fixed to the number of classes. Therefore, a direct comparison to the one-versus-all and DECOC designs is possible. Each strategy uses the previously mentioned Linear Loss-weighted decoding to evaluate their performances at identical conditions. To evaluate the performance of the different experiments, we apply stratified ten-fold cross-validation and test for the confidence interval at 95% with a two-tailed t-test [8].

In this particular problem, the sub-class is only required for Discrete Adaboost and *NMC*, while the rest of base classifiers are able to find a solution for the training set without the need for sub-classes. Finally, though the results do not significantly differ between the strategies, the Sub-class ECOC approach attains a better position in the global rank of table 5.

## 3.2 Intravascular Ultrasound Tissue Characterization

Cardiovascular diseases represented the first cause of sudden death in the occidental world [22]. Plaque rupture is one of the most frequent antecedent of coronary

**Table 5** Rank positions of the classification strategies for the Speed data set.

| | one-versus-one | one-versus-all | dense | sparse | DECOC | Sub-class ECOC |
|---|---|---|---|---|---|---|
| D. Adaboost | 66.1(3.1) | 56.6(3.1) | 55.2(2.8) | 52.3(3.6) | 58.6(3.2) | 60.8(3.1) |
| NMC | 60.7(3.2) | 50.65(3.7) | 47.4(3.8) | 45.1(3.8) | 51.9(3.2) | 62.8(3.1) |
| FLDA | 74.7(2.8) | 71.4(2.9) | 74.9(2.6) | 72.7(2.5) | 72.6(2.8) | 76.2(3.0) |
| Linear SVM | 74.9(2.7) | 72.3(2.1) | 71.8(2.1) | 68.2(2.9) | 78.9(2.1) | 78.9(1.9) |
| RBF SVM | 45.0(0.9) | 45.0(0.9) | 45.0(0.9) | 44.0(0.9) | 45.0(0.9) | 45.0(0.9) |
| Global rank | 1.8 | 3.6 | 3.4 | 4.6 | 2.6 | 1.2 |

pathologies. Depending on the propensity to collapse, coronary plaque can be divided into stable and vulnerable plaque [3]. According to pathological studies, the main features of a stable plaque are characterized by the presence of a large lipid core with a thin fibrous cap. This last type of plaque can rupture generating thrombi followed by an intimal hyperplasia. Therefore, an accurate detection and quantification of plaque types represents an important subject in the diagnosis in order to study the nature and the plaque evolution to predict its final effect.

One of the most widely used diagnostic procedures consists of screening the coronary vessels employing Intravascular Ultrasound Imaging (IVUS). This technique yields a detailed cross-sectional image of the vessel allowing coronary arteries and their morphology to be extensively explored. This image modality has become one of the principal tools to detect coronary plaque. An IVUS study consists of introducing a catheter which shots a given number of ultrasound beams and collect their echoes to form an image. According with these echoes, three distinguishable plaques are considered in this type of images: calcified tissue (characterized by a very high echo-reflectivity and absorbtion of the ultrasound signal), fibrous plaque (medium echo-reflectivity and good transmission coefficient), and lipidic or soft plaque (characterized with very low reflectance of the ultrasound signal).

Despite the high importance of studying the whole coronary vessel, in clinical practice, this plaque characterization is performed manually in isolated images. Moreover, due to the variability among different observers, a precise manual characterization becomes very difficult to perform. Therefore, automatic analysis of IVUS images represents a feasible way to predict and quantify the plaque composition, avoiding the subjectivity of manual region classification and diminishing the characterization time in large sequences of images. Given its clinical importance, automatic plaque classification in IVUS images has been considered in several research studies. The process can be divided into two stages: plaque characterization step which consists of extracting characteristic features in order to describe each tissue, and a classification step where a learning technique is used to train a classifier.

In this section, we present an intravascular data set based on texture-based features, RF signals, combined features, and slope-based features to characterize the different types of tissues.

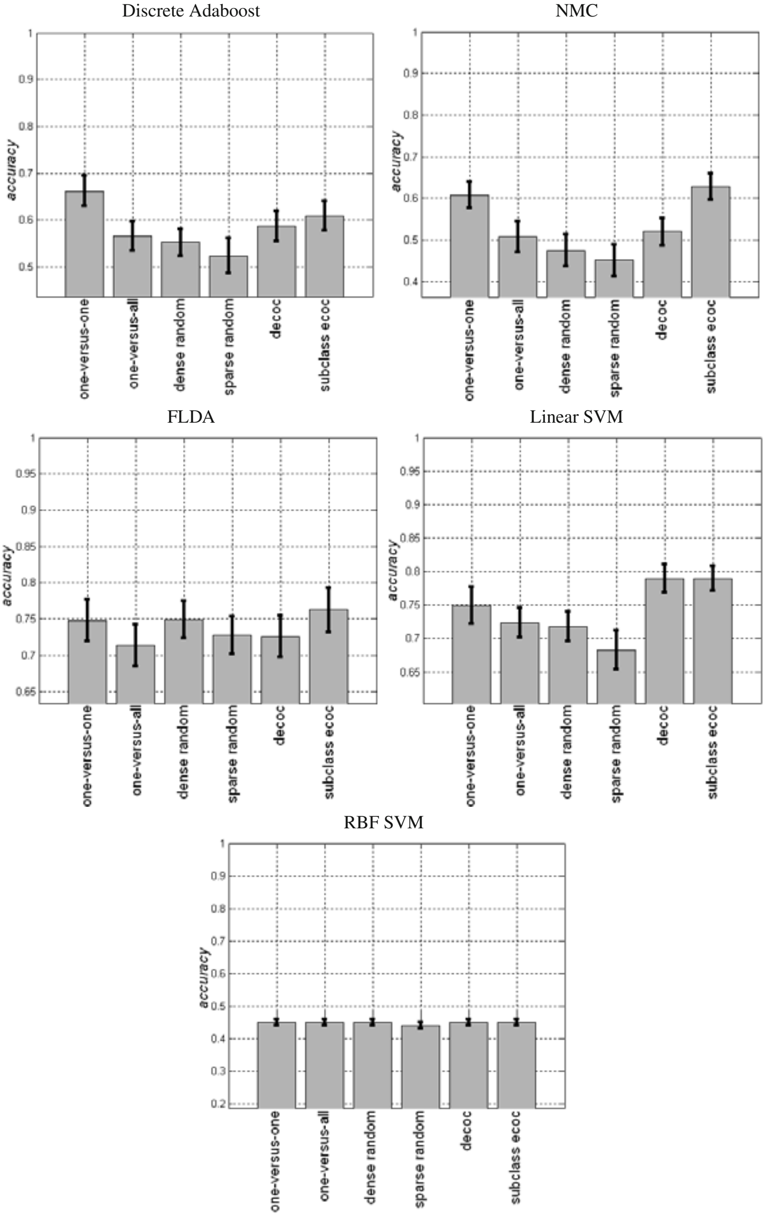


**Fig. 8** Speed data set performances.

### Feature Extraction

We consider three types of features, the first ones obtained from RF signals, the second ones based on texture-based features from reconstructed images, and finally, the slope-based features proposed in [18].

*RF Features*
In order to analyze ultrasound images, the RF signals are acquired from the IVUS equipment with a sampling rate of at least two times the transducer frequency, and filtered using a band-pass filter with 50% gain centered at the transducer frequency [14]. Then, an exponential Time Gain Compensation (TGC) is applied [14]. Once the RF signals have been acquired, filtered and exponentially compensated by the TGC, the power spectrum is obtained. Nair et al. in [18] show the modelling of the power spectrum using Autoregressive Models (ARM) as one of the most suitable and stable methods to analyze ultrasound signals [18]. It also represents an alternative to the Fourier Transform since the ARM have been proved to be more stable when small signal windows are considered.

The ARM are defined as a linear prediction equation where the output $x$ at a certain point $t$ for each A-line is equal to a linear combination of its $p$ previous outputs weighted by a set of parameters $a_p$ [24]:

$$x(t) = \sum_{k=1}^{p} a_p(k)x(t-k),$$

where $p$ is the ARM degree and the coefficients $a_p$ are calculated minimizing the error of the modelled spectrum with respect to the original using the Akaike's error prediction criterium [24].

A sliding window is formed by $n$ samples and $m$ contiguous A-lines with a displacement of $n/4$ samples and $m/3$ A-lines in order to obtain an average AR model of a region. Only one side of the obtained spectrum is used because of its symmetrical properties. This spectrum is composed of $h$ sampled frequencies ranging from 0 to $f_s/2$ [24].

In addition to the spectrum, two global measures are computed: the energy of the A-line and the energy of the window spectrum. All these features are compiled into a unique vector of $h+2$ dimensions which is used as a feature vector in the classification process.

*Texture Features Extraction*
Given that different plaques can be discriminated as regions with different grey-level distributions, it is a natural decision to use texture descriptors. In the bibliography, one can find a wide set of texture descriptors and up to our knowledge there are no optimal texture descriptors for image analysis in the general case. Our strategy is instead of trying to find out the optimal texture descriptor for our problem to gather several families of descriptors and apply multiple classifiers able to learn and extract the optimal features for the concrete problem.

Therefore, we employ three different texture descriptors: co-occurrence Matrix [20], local binary patterns [21] and Gabor filters [6, 2]. Additionally, taking into account that highly non-echogenic plaques produce significant shade in the radial direction of the vessel, we include in the feature set the presence of shading in the image as a complementary feature.

The co-occurrence matrix is defined as the estimation of the joint probability density function of gray level pairs in an image [20]. The sum of all element values is:

$$P(i,j,D,\theta) = P(I(l,m) = i \otimes I(l+Dcos(\theta), m+Dsin(\theta)) = j),$$

where $I(l,m)$ is the gray value at pixel $(l,m)$, $D$ is the distance among pixels and $\theta$ is the angle between neighbors. We have established the orientation $\theta$ to be $[0^o, 45^o, 90^o, 135^o]$ [28, 20]. After computing this matrix, Energy, Entropy, Inverse Difference Moment, Shade, Inertia and Promenance measures are extracted [20].

Local Binary Patterns (LBP) are used to detect uniform texture patterns in circular neighborhoods with any quantization of angular space and spatial resolution [21]. LBP are based on a circular symmetric neighborhood of $P$ members with radius $R$. To achieve gray level invariance, the central pixel $g_c$ is subtracted to each neighbor $g_p$, assigning the value 1 to the result if the difference is positive and 0, otherwise. LBPs are defined as follows:

$$LBP_{R,P} = \sum_{p=0}^{P} a(g_p - g_c) \cdot 2^p$$

A Gabor filter is a special case of wavelets [6] which is essentially a Gaussian modulated by a complex sinusoid $s$. In 2D, it has the following form in the spatial domain:

$$h(x,y) = \frac{1}{2\pi\sigma^2} \exp\{-\frac{1}{2}[(\frac{x^2+y^2}{\sigma^2})]\} \cdot s(x,y)$$
$$s(x,y) = \exp[-i2\pi(Ux+Vy)] \qquad \phi = \arctan V/U$$

where $\sigma$ is the standard deviation, $U$ and $V$ represent the 2D frequency of the complex sinusoid, and $\phi$ is the angle of the frequency.

According to [12], one of the main differences in the appearance of calcified tissue compared to the rest of tissue types is the shadow which is appreciated behind it. In order to detect this shadow, we perform an accumulative mean of the pixels gray values on the polar image from a pixel to the end of the column (the maximal depth considered). As a result of extracting the texture descriptors, we construct an $n$-dimensional feature vector where $n = k + l + m + 1$, k is the number of co-occurrence matrix measurements, $l$ is the number of Gabor filters, $m$ is the number of LPB and the last feature is the measure of the ”shadow” in the image.

**Intravascular data set**

In order to generate the data sets, we used the RF signals and their reconstructed images from a set of 10 different patients with Left Descent Artery pullbacks acquired in Hospital ”German Trias i Pujol” from Barcelona, Spain. All these pullbacks contain the three classes of plaque. For each one, 10 to 15 different vessel sections were selected to be analyzed. Two physicians independently segmented 50 areas of interest per pullback. From these segmentations we took 15 regions of interest (ROI) of tissue per study randomly making a total of 5000 evaluation ROIs. To build the data set, these selections were mapped in both RF signals and reconstructed images. In order to reduce the variability among different observers, the regions where both cardiologist agreed have been taken under consideration. Some samples from the data set are shown on the left of Fig. 9.

To generate the data set on texture features, the intersection between segmented images is mapped into a feature vector. Then, all the features collected are categorized by patient and each of the three possible plaques type. The image features are extracted by using the previous texture descriptors: Co-ocurrence Matrix, Local Binary Patterns, and Gabor Filters. Those features are calculated for each pixel

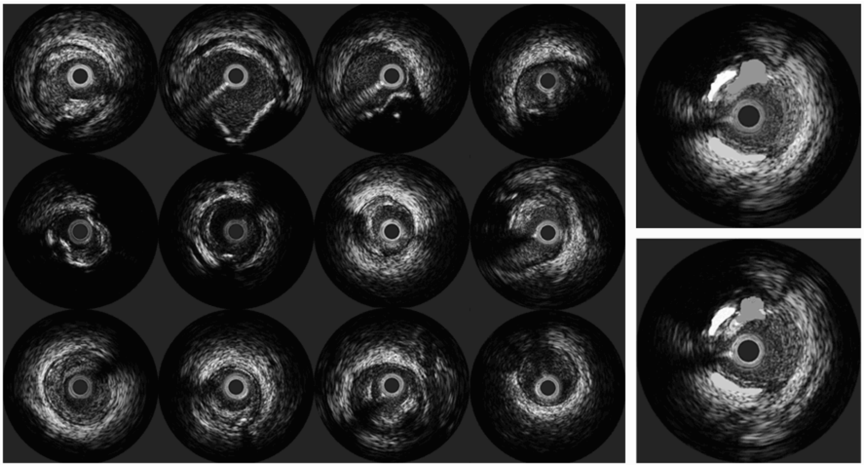


**Fig. 9** Left: IVUS data set samples. Right: (top) segmentation by a physician and (down) Automatic classification with Texture-Based Features. The white area corresponds to calcium, the light gray area to fibrosis, and the dark gray area to soft plaque.

and gathered in a feature vector of 68 dimensions. An example of a manual and automatic texture-based segmentation for the same sample is shown on the right of Fig. 9.

To generate the data set of RF features, the RF signals have been acquired using a 12-bit acquisition card with a sampling rate of $f_s = 200MHz$. The IVUS equipment used is Galaxy II from Boston Scientific with a catheter transducer frequency of $f = 40Mhz$, and it is assumed a sound speed in tissue of $1565m/s$. Each IVUS image consists of a total of 256 A-lines (ultrasound beams), with a radial distance of $r = 0.65cm$. The attenuation in tissue factor used is $\alpha = 1Db/Mhz \times cm$. To analyze the RF signals, the sliding window is composed of $n = 64$ samples of depth and $m = 12$ radial A-lines, and the displacement is fixed in 16 samples and four A-lines. The power spectrum of the window ranges from 0 to $100MHz$ and it is sampled by 100 points. Then, it is complemented with two energy measures yielding a 102 feature vector.

We also consider a third data set that concatenates the descriptors from the previous RF and texture-based features, obtaining a feature vector of length 170 features.

*Slope-based features*

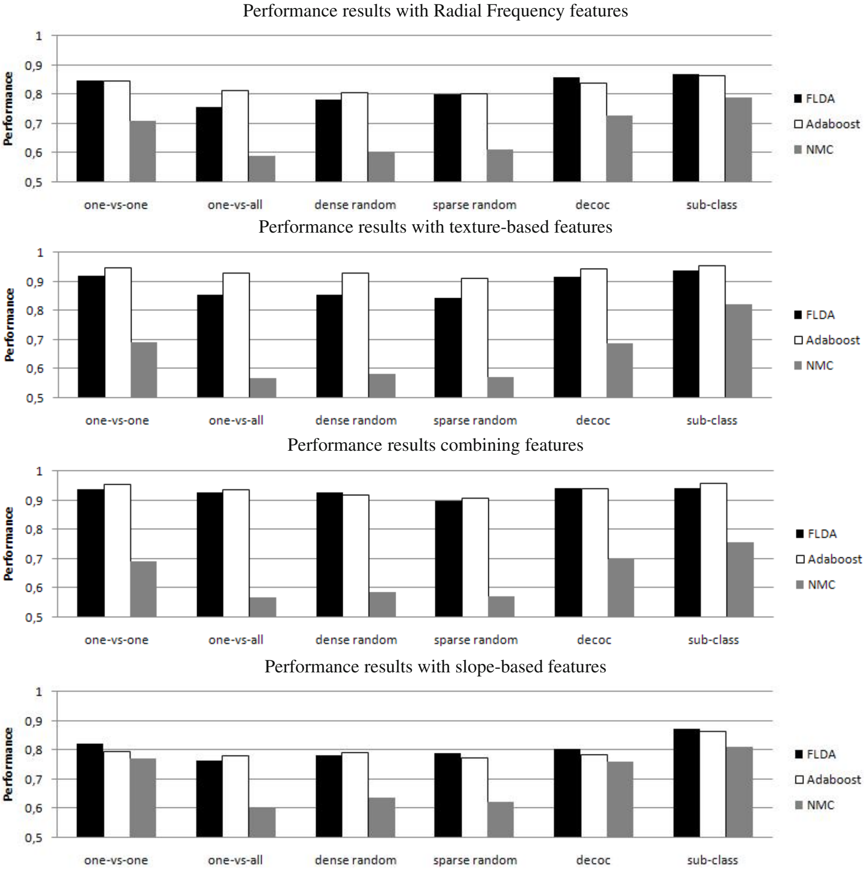
Finally, the fourth data set considers the slope-based features proposed by [18]. In particular, each sample is characterized by means of 14 slope-based features corresponding to: maximum power in DB from 20 to 60 MHz, frequency at the maximum power, negative slope in db/MHz between maximum and 60, minimum power in that slope, frequency corresponding to this negative slope, the estimated $y$ intercept of this slope, the positive slope in db/Mhz between 20 and maximum, minimum power in that slope, frequency corresponding to this negative slope, the estimated $y$

intercept of this slope, the mean power, the power at 0 MHz, power Db at 100 Mhz, and the power at the midband frequency (40 MHz) in DB [18].

To solve the problem of Intravascular tissue characterization we apply the Subclass ECOC strategy over the four previous data sets.

*IVUS characterization with sub-classes*
For this experiment, we use the four previous IVUS data sets. To measure the performances, we apply leave-one-patient-out evaluation.

Applying *NMC*, Adaboost, and *FLDA* over a set of ECOC configurations, the performance results for RF features, texture-based features, combined RF and texture-based features, and slope-based features are shown in Fig. 10. Comparing the results among the different data sets, one can see that the worst performances are obtained by the RF and slope-based features, which obtain very similar results for all the base classifiers and ECOC configurations. The texture-based features obtain in most


Performance results with Radial Frequency features

Performance results with texture-based features

Performance results combining features

Performance results with slope-based features

**Fig. 10** Performance results for different sets of features, ECOC designs and base classifiers on the IVUS data set.

cases results upon 90%. Finally, the data set of combined RF and texture-based features slightly outperform the results obtained by the texture-based feature, though the results do not significantly differ.

Concerning the classification strategies, observing the obtained performances in Fig. 10, one can see that independently of the data set and the ECOC design applied, the Sub-class ECOC approach always attains the best results. To compare these performances, the mean rank of each ECOC design considering the twelve different experiments is shown in table 6. In this case, the rankings are obtained estimating each particular ranking $r_i^j$ for each problem $i$ and each ECOC configuration $j$, and computing the mean ranking $R$ for each ECOC design as $R_j = \frac{1}{N}\sum_i r_i^j$, where $N$ is the total number of problems (3 base classifiers × 4 data sets). One can see that the Sub-class ECOC attains the best position for all experiments. To analyze if the difference between methods ranks are statistically significant, we apply the Friedman and Nemenyi tests. In order to reject the null hypothesis that the measured ranks differ from the mean rank, and that the ranks are affected by randomness in the results, we use the Friedman test. The Friedman statistic value is computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)}[\sum_j R_j^2 - \frac{k(k+1)^2}{4}] \tag{8}$$

In our case, with $k = 6$ ECOC designs to compare, $X_F^2 = 30.71$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1)-X_F^2} \tag{9}$$

Applying this correction we obtain $F_F = 11.53$. With six methods and twelve experiments, $F_F$ is distributed according to the $F$ distribution with 5 and 55 degrees of freedom. The critical value of $F(5,55)$ for 0.05 is 2.40. As the value of $F_F$ is higher than 2.45 we can reject the null hypothesis. One we have checked for the for the non-randomness of the results, we can perform a post hoc test to check if one of the techniques can be singled out. For this purpose we use the Nemenyi test - two techniques are significantly different if the corresponding average ranks differ by at least the critical difference value (CD):

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}} \tag{10}$$

where $q_\alpha$ is based on the Studentized range statistic divided by $\sqrt{2}$. In our case, when comparing six methods with a confidence value $\alpha = 0.10$, $q_{0.10} = 1.44$. Substituting in eq.10, we obtain a critical difference value of 1.09. Since the difference of any technique rank with the Sub-class rank is higher than the $CD$, we can infer that the Sub-class approach is significantly better than the rest with a confidence of 90% in the present experiments.

**Table 6** Mean rank for each ECOC design over all the experiments.

| ECOC design | one-versus-one | one-versus-all | dense random |
|---|---|---|---|
| Mean rank | 2.33 | 5.08 | 4.25 |

| ECOC design | sparse random | decoc | sub-class |
|---|---|---|---|
| Mean rank | 5.00 | 2.67 | 1.00 |

## 4 Conclusions

In this paper, we reviewed the state-of-the-art on coding and decoding designs of Error-Correcting Output Codes. We analyzed the most recent ensemble strategies, showing their benefit to deal with multi-class classification in image analysis. Moreover, the different ECOC configurations were used to solve two challenging computer vision applications: traffic sign classification and intravascular ultrasound tissue characterization, with high success.

**Acknowledgements.** This work has been partially supported by the projects TIN2006-15694-C02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

## References

1. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. JMLR 1, 113–141 (2002)
2. Bovik, A., Clark, M., Geisler, W.: Multichannel texture analysis using localized spatial filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(1), 55–73 (1990)
3. Burke, A.P., Farb, A., Malcom, G.T., Smialek, J., Virmani, R.: Coronary risk factors and plaque morphology inmen with coronary disease who died suddently. The New England Journal of Medicine 336(18), 1276–1281 (1997)
4. Casacuberta, J., Miranda, J., Pla, M., Sanchez, S., Serra, A., Talaya, J.: On the accuracy and performance of the geomobil system. In: International Society for Photogrammetry and Remote Sensing (2004)
5. Crammer, K., Singer, Y.: On the learnability and design of output codes for multi-class problems. Machine Learning 47, 201–233 (2002)
6. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Journal of the Optical Society of America 2(A), 1160–1169 (1985)
7. Daume, H., Marcu, D.: A bayesian model for supervised clustering with the dirichlet process prior. Journal of Machine Learning Research, 1551–1577 (2005)
8. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. JMLR, 1–30 (2006)
9. Escalera, S., Pujol, O., Radeva, P.: Loss-weighted decoding for error-correcting output codes. In: VISAPP (to appear)
10. Escalera, S., Pujol, O., Radeva, P.: Boosted landmarks of contextual descriptors and forest-ecoc: A novel framework to detect and classify objects in clutter scenes. Pattern Recognition Letters 28(13), 1759–1768 (2007)

11. Ghani, R.: Combining labeled and unlabeled data for text classification with a large number of categories. In: Int. conf. Data Mining, pp. 597–598 (2001)
12. Gil, D., Hernandez, A., Rodriguez, O., Mauri, F., Radeva, P.: Statistical strategy for anisotropic adventitia modelling in ivus. IEEE Trans. Medical Imaging 27, 1022–1030 (2006)
13. Hastie, T., Tibshirani, R.: Classification by pairwise grouping. NIPS 26, 451–471 (1998)
14. Caballero, K.L., Barajas, J., Pujol, O., Salvatella, N., Radeva, P.I.: In-vivo ivus tissue classification: a comparison between rf signal analysis and reconstructed images. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 137–146. Springer, Heidelberg (2006)
15. Kittler, J., Ghaderi, R., Windeatt, T., Matas, J.: Face verification using error correcting output codes. CVPR 1, 755–760 (2001)
16. Kong, E.B., Dietterich, T.G.: Error-correcting output coding corrects bias and variance. In: ICML, pp. 313–321 (1995)
17. Madala, H., Ivakhnenko, A.: Inductive Learning Algorithm for Complex Systems Modelling. CRC Press Inc., Boca Raton (1994)
18. Nair, A., Kuban, B., Obuchowski, N., Vince, G.: Assesing spectral algorithms to predict atherosclerotic plaque composition with normalized and raw intravascular ultrasound data. Ultrasound in Medicine & Biology 27, 1319–1331 (2001)
19. Nilsson, N.J.: Learning machines. McGraw-Hill, New York (1965)
20. Ohanian, P., Dubes, R.: Performance evaluation for four classes of textural features. Pattern Recognition 25, 819–833 (1992)
21. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
22. W. H. Organization, World health organization statistics (2006), `http://www.who.int/entity/healthinfo/statistics/`
23. h. OSU-SVM-TOOLBOX
24. Proakis, J., Rader, C., Ling, F., Nikias, C.: Advanced digital signal processing. Mc Millan, Basingstoke (1992)
25. Pudil, P., Ferri, F., Novovicova, J., Kittler, J.: Floating search methods for feature selection with nonmonotonic criterion functions. In: ICPR, pp. 279–283 (1994)
26. Pujol, O., Escalera, S., Radeva, P.: An incremental node embedding technique for error correcting output codes. Pattern Recognition (to appear)
27. Pujol, O., Radeva, P., Vitrià, J.: Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. PAMI 28, 1001–1007 (2006)
28. Randen, T., Husoy, J.H.: Filtering for texture classification: A comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 4, 291–310 (1999)
29. Utschick, W., Weichselberger, W.: Stochastic organization of output codes in multiclass learning problems. Neural Computation 13(5), 1065–1102 (2001)
30. Windeatt, T., Ardeshir, G.: Boosted ecoc ensembles for face recognition. In: International Conference on Visual Information Engineering, pp. 165–168 (2003)
31. Zgu, Q.: Minimum cross-entropy approximation for modeling of highly intertwining data sets at subclass levels. Journal of Intelligent Information Systems, 139–152 (1998)
32. Zhou, J., Suen, C.: Unconstrained numeral pair recognition using enhanced error correcting output coding: a holistic approach. In: Proc. in Conf. on Doc. Anal. and Rec., vol. 1, pp. 484–488 (2005)
33. Zhu, M., Martinez, A.M.: Subclass discriminant analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1274–1286 (2006)

# Chapter 3
# Morphological Operator Design from Training Data
## A State of the Art Overview

Nina S.T. Hirata

**Abstract.** Mathematical morphology offers a set of powerful tools for image processing and analysis. From a practical perspective, the expected results of many morphological operators can be intuitively explained in terms of geometrical and topological characteristics of the images. From a formal perspective, mathematical morphology is based on complete lattices, which provides a solid theoretical framework for the study of algebraic properties of the operators. Despite of these nice characteristics, designing morphological operators is not a trivial task; it requires knowledge and experience. In this chapter, a self-contained exposition on the design of translation-invariant morphological operators from training data is presented. The described training procedure relies on the canonical sup-decomposition theorem of morphological operators, which in the context of binary images states that any translation-invariant operator can be expressed uniquely in terms of two elementary operators, erosions and dilations, plus set operations. An important issue considered in this exposition is how the bias-variance tradeoff manifests within the training context and how its understanding can lead to approaches that generate better results. Several application examples that illustrate the usefulness of the described design procedure are also presented.

**Keywords:** morphological operator, translation-invariance, Boolean function, automatic design, training, bias-variance tradeoff.

## Introduction

The field of mathematical morphology emerged in the sixties in the context of binary image analysis. The works by Georges Matheron [Matheron, 1975] and

Nina S.T. Hirata
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
e-mail: nina@ime.usp.br

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 31–58.
springerlink.com

Jean Serra [Serra, 1982] are considered the main references, having inspired and influenced subsequent works in the field. In [Matheron and Serra, 2002], the reader may find an account on the birth of mathematical morphology.

Nowadays, morphological image operators are widely available in many image processing libraries and packages. Several morphological operators, such as erosion, dilation, opening, closing, morphological gradient and watershed segmentation, just to list a few of them, are commonly used. A nice characteristic of these operators is that the way they operate on an image can be intuitively explained in terms of geometrical and topological properties of the image [Serra, 1982, Soille, 2003, Dougherty and Lotufo, 2003].

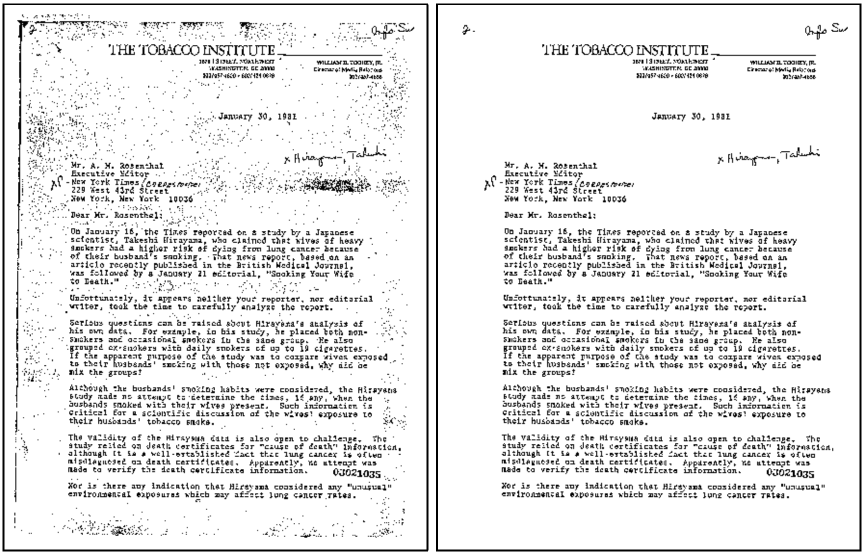
In addition, mathematical morphology is based on a sound mathematical framework. From a pure algebraic point of view, morphological operators can be modeled as mappings between complete lattices. The literature presents a number of formal studies regarding algebraic properties of the morphological operators [Serra, 1988, Heijmans and Ronse, 1990, Ronse and Heijmans, 1991, Banon and Barrera, 1991, Banon and Barrera, 1993, Heijmans, 1994].

Despite that, the use of complex operators, such as those obtained by sequentially composing simpler operators, does not seem to be a common practice. In fact, some expert level knowledge and experience seem to be key factors for successfully designing complex morphological operators. The design process is usually a trial and error based procedure, in which one needs not only to define the proper composition but also to determine the adequate parameters of each operator. In other cases, the nature of the operator requires more patience than skill. Problems that consist of fine tuning structuring elements to match several types of templates are typically the case of the latter type. Designing such type of operators is a very time-consuming and error-prone task.

A possible approach to deal with these issues is to use machine learning based techniques to design the operators. Ideally, the user should specify the desired image transformation in a high level description. Such description would then be automatically mapped to a low-level description of a suitable image operator.

In order to build such mapping, an expressive representation form for the operators must be available. A first general representation property, due to Matheron, states that every increasing morphological operator can be expressed as a union of erosions [Matheron, 1975]. Later, Maragos showed the sufficient condition for the existence of minimal representations of such form [Maragos, 1985]. These results were later extended for the non-necessarily increasing operators by Banon and Barrera [Banon and Barrera, 1991, Banon and Barrera, 1993]. According to that extension, any translation-invariant operator can be expressed in terms of four elementary operators, namely erosions, dilations, anti-erosions and anti-dilations. These results are powerful in the sense that they provide a common representation structure for the class of translation-invariant operators.

Dougherty was one of the first to exploit these representation structures in order to design morphological operators automatically from training data. Desired image transformations were specified through pairs of input-output image samples, like the ones shown in Fig. 1. The first works were restricted to the class of increasing binary morphological operators [Dougherty, 1992a], and later they have been extended to the non-necessarily increasing [Dougherty and Loce, 1993, Barrera et al., 1997] and non-binary cases [Dougherty, 1992b, Hirata Jr. et al., 2000].



Info Sv

THE TOBACCO INSTITUTE

1875 I Street, Northwest
Washington, DC 20006
202/457-4800 · 800/424-9876

William H. Toohey, Jr.
Director of Media Relations
202/457-4888

January 30, 1981

x Hirayama, Takeshi

Mr. A. M. Rosenthal
Executive Editor
New York Times
229 West 43rd Street
New York, New York 10036

Dear Mr. Rosenthal:

On January 16, the Times reported on a study by a Japanese scientist, Takeshi Hirayama, who claimed that wives of heavy smokers had a higher risk of dying from lung cancer because of their husband's smoking. That news report, based on an article recently published in the British Medical Journal, was followed by a January 21 editorial, "Smoking Your Wife to Death."

Unfortunately, it appears neither your reporter, nor editorial writer, took the time to carefully analyze the report.

Serious questions can be raised about Hirayama's analysis of his own data. For example, in his study, he placed both non-smokers and occasional smokers in the same group. He also grouped ex-smokers with daily smokers of up to 19 cigarettes. If the apparent purpose of the study was to compare wives exposed to their husbands' smoking with those not exposed, why did he mix the groups?

Although the husbands' smoking habits were considered, the Hirayama study made no attempt to determine the times, if any, when the husbands smoked with their wives present. Such information is critical for a scientific discussion of the wives' exposure to their husbands' tobacco smoke.

The validity of the Hirayama data is also open to challenge. The study relied on death certificates for "cause of death" information, although it is a well-established fact that lung cancer is often misdiagnosed on death certificates. Apparently, no attempt was made to verify the death certificate information. 03021035

Nor is there any indication that Hirayama considered any "unusual" environmental exposures which may affect lung cancer rates.

**Fig. 1** High level description of an image transformation through an input-output pair of images. Input image (left) from [Agam et al., 2006], and output image (right) obtained by manually editing the input image. Scale modified.

Prior to that, some attempts for the automatic design of morphological operators have appeared also in [Schimitt, 1989, Vogt, 1989, Joo et al., 1990]. More recent approaches such as [Harvey and Marshall, 1996, Yoda et al., 1999, Quintana et al., 2006] consider sequential or hybrid decomposition structure of the operators. For instance, the number of operators in the composition are allowed to vary up to a maximum number of erosions and/or dilations, with structuring element size varying within a predefined range.

Another class of operators, directly related to mathematical morphology, that received great attention are the stack filters. Stack filters [Wendt et al., 1986] have been introduced independently to mathematical morphology as a generalization of the median filters. They consist of filters whose results can be obtained by thresholding the input image at every level, then applying the filter to the binary images resulting at each level, and then summing up the binary results. The connection of stack filters and mathematical morphology soon became evident. From the perspective of morphological operators, stack filters are gray-scale image increasing operators, with flat structuring elements [Maragos and Schafer, 1987, Heijmans, 1994]. To date, several approaches have been proposed for the design of stack filters from training data [Coyle and Lin, 1988, Lin and Kim, 1994, Tăbuş et al., 1996, Yoo et al., 1999, Lee et al., 1999, Dellamonica Jr. et al., 2007]. Since stack filters are characterized by positive Boolean functions, their design has much in common with the design of increasing binary morphological operators, which are also characterized by Boolean functions of the same type.

This chapter is devoted to the presentation of a self-contained state of the art overview on the design of morphological operators from training data in the following scenario: the design goal is specified in high level abstraction through pairs of input-output images, which will serve as training data to a learning system. The learning system will then output a sup-decomposition structure of an operator that minimizes the mean absolute error (between the resulting image and the ideal output image). The presentation is restricted to the context of binary image operator design. However, extension of the concepts to gray-scale image mappings is straightforward, as will be pointed out in the conclusion.

The organization of this chapter is as follows. In Section 1, basic notations, terminologies and definitions that will help the understanding of which operators and representation are considered for the design problem are presented. In Section 2 a formulation of the design problem as an statistical estimation problem is first presented. Then, a machine learning based approach to solve the problem is described and the steps of the training procedure are detailed and illustrated through an example. In Section 3, a bias-variance decomposition of the design error is discussed together with approaches to mitigate these errors. In Section 4, several application examples are presented. In Section 5, the chapter is concluded with a brief summary of the main ideas and discussions on current challenges and future steps.

## 1 Binary Morphological Operators

Let $E = \mathbb{Z}^2$, the Cartesian grid, be the image domain. Each point in $E$ corresponds to a pixel location. The origin of $E$ is denoted 0. Given two points $x$ and $y$ in $E$, $x+y$ denotes the usual vector addition in $E$. Given a set $X \subseteq E$, $X^c$ denotes the complementary set with respect to $E$; $X_z = \{x + z : x \in X\}$ denotes the translate of $X$ by $z$; and $\check{X} = \{-x : x \in X\}$ denotes the transpose of $X$. The power set of a set $A$ is denoted $\mathcal{P}(A)$.

A binary image defined on $E$ can be expressed as a function $f$ from $E$ to $\{0, 1\}$. The set of all such mappings is denoted $\{0, 1\}^E$. A function $f \in \{0, 1\}^E$ can be seen as an indicator function of a set $S_f \subseteq E$, that is, for any $x \in E$, $x \in S_f \iff f(x) = 1$. The set $S_f$ (points such that $f(x) = 1$) corresponds to the foreground pixels of image $f$, while $S_f^c$ (points such that $f(x) = 0$) corresponds to the background pixels. Because of this relation between sets and binary functions, a binary image will be referred as a function or, interchangeably, as a subset, depending on the context. The collection $\mathcal{P}(E)$ is understood as the set of all possible binary images on $E$. In order to simplify notation, both functions and corresponding sets are denoted by a same letter. Thus, $x \in S$ or $S(x) = 1$ indicates that $x$ is a pixel in the foreground of image $S$.

A binary image mapping, or binary image operator, is a mapping from $\{0,1\}^E$ to $\{0,1\}^E$ or, equivalently, from $\mathcal{P}(E)$ to $\mathcal{P}(E)$. Because of the latter, binary image operators are also called set operators.

Sets form a rich algebraic structure known as Boolean lattice or Boolean algebra (see, for instance, [Heijmans, 1994] for more details). The usual set inclusion relation is a partial order relation (that is, it is reflexive, antisymmetric and transitive). The notion of partial order allows the definition of intervals. Given two sets $A, B \in \mathcal{P}(E)$, the interval with extremities $A$ and $B$ is given by $[A, B] = \{X : A \subseteq X \subseteq B\}$. If $A \nsubseteq B$ then $[A, B] = \emptyset$.

**Erosion and dilation.** Two important operators from mathematical morphology are erosion and dilation. Given a binary image $X$ and a set $B$ called structuring element, the erosion and dilation of $X$ by $B$ are defined[1], respectively, as

$$\varepsilon_B(X) = \{x \in E : B_x \subseteq X\} \tag{1}$$

$$\delta_B(X) = \{x \in E : \check{B}_x \cap X \neq \emptyset\} . \tag{2}$$

The structuring element plays the role of locally probing the input image; the output at a given location is determined by the relation between the structuring element and the image around that location. By considering structuring elements of different sizes and shapes, images can be analyzed with respect to different geometrical and topological features.

Erosions and dilations are dual operators, that is, $\delta_B(X) = (\varepsilon_{\check{B}}(X^c))^c$. This relation will be useful later.

Many operators can be defined by composing the elementary operators erosion and dilation. For instance, the opening of an image $X$ by structuring element $B$ is defined as $\gamma_B(X) = \delta_B(\varepsilon_B(X))$, while the closing is defined as $\varphi_B(X) = \varepsilon_B(\delta_B(X))$. The morphological gradient is defined as $\nabla_B(X) = \delta_B(X) \setminus \varepsilon_B(X)$, where $\setminus$ denotes the set difference.

**Hit-or-miss.** Another important operator is the so called hit-or-miss. Given two structuring elements, $A$ and $B$, hit-or-miss is defined as

[1] The definition of dilation is sometimes with $B_x$ in place of $\check{B}_x$, which may cause some confusion. See more details in [Soille, 2003]. Also, since usually a symmetric structuring element is used (which implies $\check{B} = B$), this difference ends up having no consequences in practice.

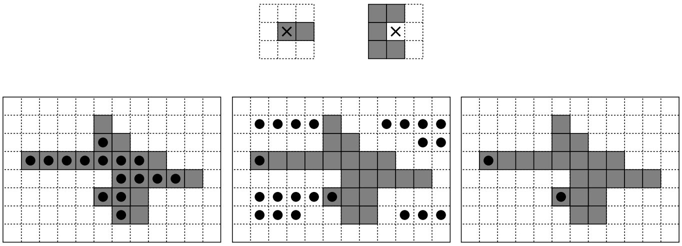


**Fig. 2** Top row: the two structuring elements, $A$ and $B$, with origin at the center. Bottom row: the set of black dots in the three images indicate, respectively, the points in which $A$ hits $X$, the points in which $B$ hits the background (or, equivalently, misses $X$), and the points detected by $H_{(A,B)}$.

$$H_{(A,B)}(X) = \{x \in E : A_x \subseteq X \text{ and } B_x \subseteq X^c\} = \varepsilon_A(X) \cap \varepsilon_B(X^c)\,. \tag{3}$$

Hit-or-miss finds locations in which the first structuring element fits into the foreground while the second one fits to the background (or, equivalently, misses the foreground). If $A \cap B \neq \emptyset$ the result is empty. Figure 2 shows an example. Dark boxes represent the foreground pixels while non-filled ones represent background pixels. This convention will be used throughout the chapter.

An operator closely related to $H$ is the sup-generating or wedge operator. It is characterized by a pair $(A, B)$ of structuring elements, such that $A \subseteq B$, and is defined as

$$\Lambda_{(A,B)}(X) = \{x \in E : A_x \subseteq X \subseteq B_x\} = \varepsilon_A(X) \cap (\delta_{\check{B}^c}(X))^c\,. \tag{4}$$

This operator verifies if the shape around $x$ is between $A$ and $B$. Since $\{x \in E : X \subseteq B_x\} = \{x \in E : B^c_x \cap X = \emptyset\} = \{x \in E : B^c_x \cap X \neq \emptyset\}^c = (\delta_{\check{B}^c}(X))^c$ and because of the duality relation $(\delta_{\check{B}}(X))^c = \varepsilon_B(X^c)$ between erosions and dilations, it follows that $\Lambda_{(A,B)}(X) = H_{(A,B^c)}(X)$.

One difference between hit-or-miss and wedge operators is that in the former both structuring elements are usually small and finite while in the latter the second structuring element is unbounded.

**Sup-decomposition of translation-invariant operators.** The kernel of an operator $\Psi$ is defined as

$$\mathcal{K}(\Psi) = \{X \subseteq E : 0 \in \Psi(X)\}\,. \tag{5}$$

An operator $\Psi : \mathcal{P}(E) \to \mathcal{P}(E)$ is translation-invariant if, for any $X \in \mathcal{P}(E)$ and $z \in E$, $[\Psi(X)]_z = \Psi(X_z)$. If $\Psi$ is translation invariant, then

$$\Psi(X) = \bigcup_{[A,B] \subseteq \mathcal{K}(\Psi)} \Lambda_{(A,B)}(X) \tag{6}$$

(see a proof in [Banon and Barrera, 1991, Heijmans, 1994]).

The set of all maximal intervals contained in $\mathcal{K}(\varPsi)$ is called the basis of $\varPsi$ and denoted $\mathcal{B}(\varPsi)$. The minimal decomposition theorem for translation-invariant operators [Banon and Barrera, 1991] states that

$$\varPsi(X) = \bigcup_{[A,B]\in\mathcal{B}(\varPsi)} \varLambda_{(A,B)}(X)\,. \tag{7}$$

If $\varPsi$ is increasing and $A$ is in the kernel of $\varPsi$, then any set $B$ such that $A \subseteq B$ is also in the kernel. Thus intervals in the basis of an increasing operator are of the form $[A, E]$. Note that if $B = E$, then $(\delta_{\check{B}^c}(X))^c = (\delta_{\check{E}^c}(X))^c = (\delta_{\emptyset}(X))^c = \emptyset^c = E$. Thus, in Eq. 7 one has $\varLambda_{(A,B)} = \varepsilon_A$, and that leads to the sup-decomposition of increasing operators due to Matheron [Matheron, 1975].

**Local definition.** Given a non-empty set $W$ in $E$, called window, an operator $\varPsi$ is locally defined within $W$ if, for any $X \subseteq E$,

$$[\varPsi(X)](x) = [\varPsi(X \cap W_x)](x)\,. \tag{8}$$

Suppose $W$ is finite and let $n = |W|$ (cardinality of $W$). Let $\psi : \{0,1\}^n \to \{0,1\}$ be a Boolean (or switching) function with $n$ variables $x_1, x_2, \ldots, x_n$. Suppose each Boolean variable $x_i$ is assigned respectively to each point $w_i \in W$. For each location $z \in E$, an assignment is defined by doing $x_i = 1 \iff w_i \in X_{-z} \cap W$ (this is equivalent to checking if $w_i + z \in X$), and $x_i = 0$ otherwise. Such assignment is denoted by $\psi(X_{-z} \cap W)$.

Locally defined operators can be characterized by local functions $\psi_x : \{0,1\}^n \to \{0,1\}$, by the following relation:

$$[\varPsi(X)](x) = \psi_x(X_{-x} \cap W)\,. \tag{9}$$

If $\varPsi$ is translation-invariant, then $\psi_x = \psi_y$ for any $x, y \in E$.

***W*-operators.** Operators that are translation-invariant and locally defined are called $W$-operators. Notice that, although $W$ seems to impose a restriction to the class of operators, it is flexible enough to include all translation-invariant operators (for that, one only needs to consider $W = E$).

Many operators are locally defined. For instance, $\varepsilon_A$ and $\delta_A$ are locally defined within $W = A$. The Boolean function that characterizes $\varepsilon_A$ is given by the logical product $x_1\, x_2\, \ldots\, x_n$, for this is the logical expression that will output 1 only if every $x_i$ equals 1. In the case of dilations, the function is $x_1 + x_2 + \ldots + x_n$ because it suffices that only one $x_i$ equals 1 in order to the function value be equal to 1. Hit-or-miss operators with parameters $(A, B)$ correspond to logical product terms in which variables related to elements that are both in $A$ and $B^c$ appear uncomplemented ($x_i$), those that are neither in $A$ nor in $B^c$ appear complemented ($\overline{x_i}$), and those that are not in $A$ but are in $B^c$ are don't cares (do not appear in the logical product term).

Since the local definition property implies $0 \in \Psi(S) \iff 0 \in \Psi(S \cap W)$, kernel elements can be constrained within $W$. Therefore, for $W$-operators the kernel definition can be rewritten as

$$\mathcal{K}(\Psi) = \{X \subseteq W : 0 \in \Psi(X)\}\,, \tag{10}$$

and the locally defined wedge operator as

$$\begin{aligned} \Lambda_{(A,B)}(S) &= \{x \in E : A_x \subseteq (S \cap W_x) \subseteq B_x\} \\ &= \{x \in E : A \subseteq (S_{-x} \cap W) \subseteq B\}\,, \end{aligned} \tag{11}$$

with $A \subseteq B \subseteq W$.

The local definition property plus translation-invariance is sufficient to establish a lattice isomorphism between $W$-operators and Boolean functions [Barrera and Salas, 1996]. The canonical decomposition of Boolean functions as a sum of products corresponds to the sup-decomposition of operators by trivial intervals of the type $[A, A]$ for each $A$ in the kernel of $\Psi$. The minimal sum of products form corresponds to the minimal sup-decomposition (each product term corresponds to a basis interval). Furthermore, the representation of increasing operators as a supremum of erosions correspond to the representation of monotone (or positive) Boolean functions as a sum of products, each one involving no complemented variables. As in the case of Boolean functions, $W$-operators also admit dual canonical representations [Banon and Barrera, 1991].

## 2 Designing Morphological Operators from Training Images

The problem formulation presented below is essentially the one stated in early works on this approach. Among them, representative ones are [Dougherty and Loce, 1994, Barrera et al., 1997].

Images to be processed and corresponding desired output images are considered to be, respectively, realizations of random sets $\mathbf{S}$ and $\mathbf{I}$, with joint distribution $P(\mathbf{S}, \mathbf{I})$. It is also supposed that they are characterized by a jointly stationary local process $(\mathbf{S} \cap W_z, \mathbf{I}(z))$, with realizations of $\mathbf{S} \cap W_z$ in $\mathcal{P}(W_z)$ and of $\mathbf{I}(z)$ in $\{0, 1\}$. Owing to stationarity, location $z$ may be dropped from notation and thus the local process is denoted $(\mathbf{X}, \mathbf{y})$. Realizations of $\mathbf{X}$ are in $\mathcal{P}(W)$ and realizations of $\mathbf{y}$ are in $\{0, 1\}$.

Under these assumptions, the optimality of $W$-operators is characterized by a probabilistic error in terms of the local process $(\mathbf{X}, \mathbf{y})$. The mean absolute error (MAE) of $\Psi$, defined by a local function $\psi$, with respect to a joint process $(\mathbf{S}, \mathbf{I})$, characterized by a local joint process $(\mathbf{X}, \mathbf{y})$, is given by

$$MAE\langle\Psi\rangle = E[|\psi(\mathbf{X}) - \mathbf{y}|]\,. \tag{12}$$

Given $\Psi$, its MAE can be written as

$$\begin{aligned} MAE\langle\Psi\rangle &= \sum_{(X,y)} |\psi(X) - y| \, P(X,y) \\ &= \sum_{(X,0)} \psi(X)P(X,0) + \sum_{(X,1)} |\psi(X) - 1|P(X,1) \\ &= \sum_{\{X:\,\psi(X)=1\}} P(X,0) + \sum_{\{X:\,\psi(X)=0\}} P(X,1)\,. \end{aligned} \tag{13}$$

The amount each $X$ contributes to MAE does not depend on how much others contribute. Thus, in order to minimize MAE, one should choose for each $X$ the value of $\psi(X)$ based on $P(X,y)$, $y \in \{0,1\}$. More precisely, the optimal operator with respect to a local joint process $(\mathbf{X},\mathbf{y})$ under the MAE criterion is given by

$$\psi(X) = \begin{cases} 1, & \text{if } P(X,0) < P(X,1), \\ 0, & \text{if } P(X,0) > P(X,1), \\ 1 \text{ or } 0, & \text{if } P(X,0) = P(X,1). \end{cases} \tag{14}$$

Equation 14 means that, if the joint probability $P(\mathbf{X},\mathbf{y})$ is known, so it is the optimal MAE operator. However, in practice $P(\mathbf{X},\mathbf{y})$ is not known for every possible pattern $X$. In the following section, a machine learning based formulation to estimate an optimal operator from training data is described.

## *Basic Training Procedure*

An approach proposed in early works (such as [Barrera et al., 1997, Barrera et al., 2000]), which is here called the basic training procedure, consists in estimating the joint probabilities $P(X,y)$, $X \in \mathcal{P}(W)$ and $y \in \{0,1\}$, from training images. The set of training images consists of pairs $(S_i, I_i)$, $i = 1,\ldots,m$, where $S_i$ corresponds to an input image (an image to be processed) and $I_i$ corresponds to its respective output (desired ideal result). These images are usually prepared by manual edition. The basic training procedure is composed with the following main steps:

**Step 1 – estimation of $P(X,y)$.** Window $W$ is slided on the input image of each training pair $(S_i, I_i)$. For each pixel $p$ of $S_i$, the pattern $X$ under $W$ ($X = (S_i)_{-p} \cap W$) and its respective output value $y = I_i(p)$ are recorded into a frequency table. If the translated window $W$ is not entirely contained in the image domain, then the corresponding pixel location is ignored. The result of this step is an estimate $\hat{P}(X,y)$ of $P(X,y)$.

**Step 2 – optimal decision.** For each observed pattern $X$, a value aiming to minimize MAE is chosen for $\psi(X)$ : if $\hat{P}(X,1) > \hat{P}(X,0)$, then $\psi(X) = 1$ and, otherwise, $\psi(X) = 0$. In the cases in which $\hat{P}(X,1) = \hat{P}(X,0)$, assignment to either 1 or 0 is equivalent in terms of MAE. By convention, $\psi(X) = 0$ is adopted. This choice is reasonable if the probability of foreground pixels being mapped to output background is larger or equal to the opposite. The results of this step are partly defined kernel and non-kernel sets.

**Step 3 – generalization.** In step 2, if all patterns $X$ are observed, the estimated optimal MAE operator will be completely defined. However, in practice, unless window $W$ is very small, not all patterns are observed in step (1), leading to an incompletely specified function. In order to complete the definition, a training algorithm is used. Specifically, a Boolean function minimization algorithm [Hirata et al., 2002] is used. Besides generalization (completion of definition), it transforms trivial intervals of the form $[X, X]$ to maximal ones, generating a more compact representation.

It should be noted that, from a computational point of view, the learning part of the procedure is a classification problem in which the goal is to discriminate kernel from non-kernel elements. Therefore, any classification algorithm could be applied on data obtained in step (2) or even in step (1). An advantage of using Boolean function minimization as a learning algorithm is the fact that the result is in sum of products form, which has a straightforward interpretation in terms of morphological operators (i.e., as a supremum of wedge operators). That allows, for instance, an easy interpretation of the morphological effects of the operator on the images. On the other hand, it is important to mention that there is an essential difference between learning classifiers and learning operators: operators are endowed with properties and structures that can be explored in the design process together with the geometrical conformation of the features (observations related to individual window points).

## *Design Example*

The steps described above are illustrated through a detailed analysis of an example. Consider the problem of detecting end-points in binary images. A training pair of images is shown in Fig. 3. At the left side is the input image and at the right side is the (ideal) output image.

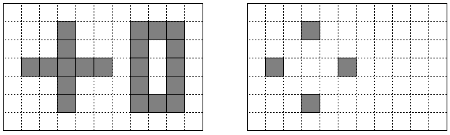


**Fig. 3** A training pair of images for (vertical and horizontal) end point detection.

Considering the 5-point elementary cross window, the patterns observed in the input image and the corresponding output frequencies are shown in Fig. 4.

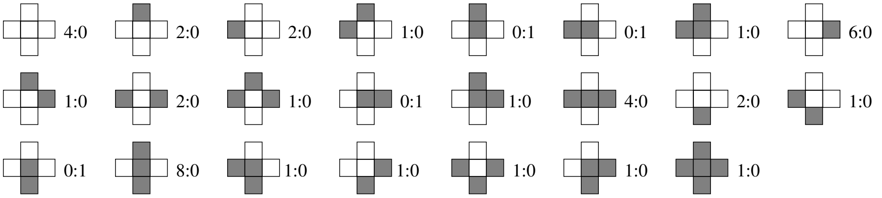


**Fig. 4** Frequency table. For each pattern $X$, two frequencies $f_0 : f_1$ are shown ($f_0$ is the frequency of occurrences of $(X, 0)$ while $f_1$ is the frequency of occurrences of $(X, 1)$).

Since $n = 5$, the number of possible patterns within $W$ is 32. However, only 23 of them have been observed in the above training data. Among these, four are such that $P(X, 1) > P(X, 0)$. After Boolean function minimization, the four intervals that define the estimated operator is shown in Fig. 5.

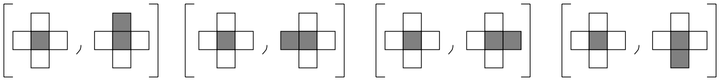


**Fig. 5** Intervals of the operator estimated from the training data shown in Fig. 4.

Figure 6 shows two test images and the respective results (black dots) generated by the trained operator, overlaid on them. The result for the image at the left side is as expected. However, in the image at the right side, four pixels in the diagonal lines that are not end points are marked. These errors are due to the fact that these points correspond to patterns that were not observed during training. Specifically, in the above example, the training image contains only 4-connected patterns, while the second test image contains 8-connected patterns.

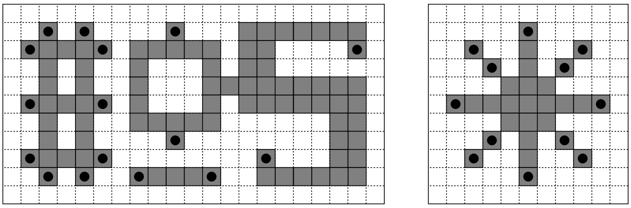


**Fig. 6** Test images and respective results (black dots).

Generalization refers to the ability of the operator to correctly classify those patterns that were not observed during training. If one wishes to obtain operators that correctly classify all patterns, then the training images should contain most of the relevant patterns.

## 3 The Bias-Variance Tradeoff

Let $\Psi_{\text{opt}}$ be the overall optimal operator in a given application domain. The underlying window $W_{\text{opt}}$ of $\Psi_{\text{opt}}$ is unknown (and possibly infinite), but its existence is admitted for purposes of error analysis. Suppose $W_{\text{opt}}$ is finite and that a $W_{\text{opt}}$-operator is designed from a set of training images, resulting in the estimated optimal operator $\hat{\Psi}_{\text{opt}}$. The difference $\Delta(\hat{\Psi}_{\text{opt}}, \Psi_{\text{opt}}) = MAE\langle\hat{\Psi}_{\text{opt}}\rangle - MAE\langle\Psi_{\text{opt}}\rangle$ corresponds to the estimation error. The expected estimation error can then be expressed as the expected error $E[\Delta(\hat{\Psi}_{\text{opt}}, \Psi_{\text{opt}})]$ with respect to all possible sets of training images.

On the other hand, let $C$ be a subspace of the space of all $W_{\text{opt}}$-operators. For instance, $C$ could be the space of $W$-operators for a window $W \subset W_{\text{opt}}$. The optimal operator in this constrained space $C$ is denoted $\Psi_C$ and the difference $\Delta(\Psi_C, \Psi_{\text{opt}}) = MAE\langle\Psi_C\rangle - MAE\langle\Psi_{\text{opt}}\rangle$ corresponds to the constraint error. Let $\hat{\Psi}_C$ be the operator designed in $C$; then its estimation error is given by $\Delta(\hat{\Psi}_C, \Psi_C) = MAE\langle\hat{\Psi}_C\rangle - MAE\langle\Psi_C\rangle$.

Relative to the globally optimal operator, the main goal in the design process is to minimize the difference $\Delta(\hat{\Psi}_C, \Psi_{\text{opt}}) = MAE\langle\hat{\Psi}_C\rangle - MAE\langle\Psi_{\text{opt}}\rangle$. The error of the estimated operator $\hat{\Psi}_C$ can be written as

$$\begin{aligned}\Delta(\hat{\Psi}_C, \Psi_{\text{opt}}) &= MAE\langle\hat{\Psi}_C\rangle - MAE\langle\Psi_{\text{opt}}\rangle \\ &= MAE\langle\hat{\Psi}_C\rangle - MAE\langle\Psi_C\rangle \\ &\quad + MAE\langle\Psi_C\rangle - MAE\langle\Psi_{\text{opt}}\rangle \\ &= \Delta(\hat{\Psi}_C, \Psi_C) + \Delta(\Psi_C, \Psi_{\text{opt}})\,.\end{aligned}$$

The expected error of $\hat{\Psi}_C$ relative to the optimal operator is

$$E[\Delta(\hat{\Psi}_C, \Psi_{\text{opt}})] = E[\Delta(\hat{\Psi}_C, \Psi_C)] + \Delta(\Psi_C, \Psi_{\text{opt}}) \tag{15}$$

and it is composed of two terms: the estimation error restricted to the subspace $C$ plus the constraint error, which is a (unknown) constant.

This error expression explains the bias-variance error decomposition [Hastie et al., 2009] in the context of $W$-operator design from training data. The second term in the summation corresponds to the bias error: if the constraint is too severe so as to restrict the operator space too much, it leads to high bias. The first term is related to the variance error: a too large space may imply a too large estimation error.

In practice, restriction is imposed by fixing a window $W$. Considering a fixed amount of training data, it is common that a plot of the MAE against window size exhibits a U-shaped curve, as the ones shown in Fig. 7. As can be seen, the MAE is very large for small windows and, as the window size increases, MAE error drops considerably until to the point where it starts to oscillate and then to slowly increase. This behavior can be explained by the bias-variance decomposition of the error: for small windows, MAE error is large due to strong bias (space constraint) whereas for large windows MAE error tends to increase due to variance (large estimation error). This type of behavior is generally observed irrespective to the amount of training data.

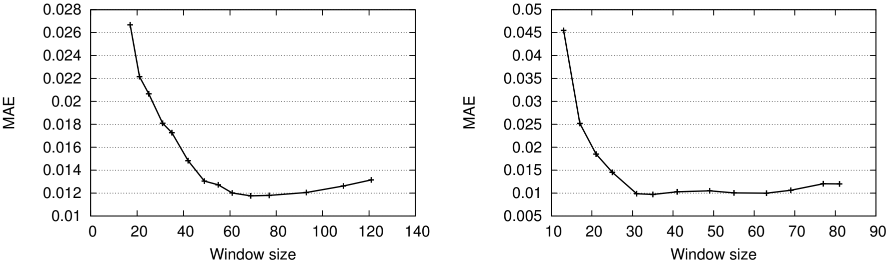


**Fig. 7** MAE curve for different window sizes, with fixed amount of training data.

Comparing Eq. 15 with $E[\Delta(\hat{\Psi}_{\text{opt}}, \Psi_{\text{opt}})]$ it is clear that constraint $C$ is beneficial if the constraint error plus estimation error in the constrained space $(E[\Delta(\hat{\Psi}_C, \Psi_C)] + \Delta(\Psi_C, \Psi_{\text{opt}}))$ is smaller than estimation error in the unconstrained space $(E[\Delta(\hat{\Psi}_{\text{opt}}, \Psi_{\text{opt}})])$.

An undoubtedly secure way to mitigate this tradeoff is to set up a large window, so as to avoid bias, and increase the amount of training data so as to reduce variance. However, in many cases, training data is not easily obtained. In a realistic scenario, the number of training images is fixed. Thus, one needs to make choices that represent the best tradeoff between bias and variance for the given amount of training data.

In the following subsections, approaches that have been proposed to tackle this issue, considering situations in which the number of training images is fixed, are presented. These approaches are grouped in two categories: one that considers constrained operator spaces (by algebraic or structural constraints) and a second one that exploits data modeling.

## *Algebraic Constraint*

One possible approach to constrain the space of operators is by considering subclasses of operators that satisfy some algebraic properties.

**Anti-extensive operators.** An operator $\Psi$ is anti-extensive if, for any $X \in \mathcal{P}(E)$, $\Psi(X) \subseteq X$. In words, they are operators such that the resulting image is a subset of the input image. Typical applications that require this type of operators are additive noise filtering and segmentation.

An immediate consequence of this property is that all elements of the kernel contains the origin. Therefore, the design algorithm can safely set $\psi(X) = 0$ for all patterns $X$ that does not contain the origin. Moreover, since no pixel in the background will be mapped to a foreground pixel in the resulting image, there is no need to analyze background pixels during joint probability estimation and application of the designed operators. This saves computational time.

**Increasing operators.** Increasing operators are those that preserve the order relation, i.e., $\Psi$ is increasing if, for any $X, Y \in \mathcal{P}(E)$, $X \subseteq Y$ implies that $\Psi(X) \subseteq \Psi(Y)$. In terms of kernel elements this means that if a given set $X$ is in the kernel so do all other sets $Y$ such that $X \subseteq Y$, and symmetrically, if a set $X$ is not in the kernel then so do not all other sets $Y$ such that $Y \subseteq X$. The characteristic Boolean functions of such operators are positive (monotone).

The optimal MAE operator defined in Eq. 14 is not necessarily increasing. In order to design an increasing operator, one has to switch the decision value on some patterns. However, any switching comes with an increase in the MAE error. More specifically, the error increase relative to the optimal operator due to switching $\psi(X)$ from $y$ to $1-y$ is $P(X, y) - P(X, 1-y)$ (see, for instance [Hirata et al., 2000a], for more details). In addition, switching the value at a given pattern may imply the need to switch the values of other patterns.

This problem can be formulated as a linear programming (LP) problem. However, since the number of variables and constraints in this formulation grows exponentially with the window size, the resulting LP problem is computationally challenging. An exact algorithm to solve this problem has been proposed in [Dellamonica Jr. et al., 2007]. By breaking the problem into subproblems, it is able to efficiently solve instances with about 25 variables in standard desktop computers at a expense of memory usage. Another algorithm to solve the problem of designing increasing operators appears in the context of stack filter design [Yoo et al., 1999]. It relies on an iterative technique that is guaranteed to converge. One drawback is that the convergence speed is not controlled.

Most interesting design application examples of increasing operators appear in the context of stack filters.

## *Structural Constraint*

Another way to constrain the class of operators is by considering a specific decomposition structure of the operators. For instance, fixing a window $W$ constrains the space to the class of $W$-operators. Finding an adequate window, that leads to minimum error, is a challenging subproblem that is equivalent to the classical feature selection problem [Guyon and Elisseeff, 2003]. However, rather than considering the canonical decomposition of operators, one can consider any other specific structure. For instance, one could think of a sequence of erosions and dilations, or sequential composition of operators from some subclasses of operators. Such structures are explored in some works that employ evolutionary approaches [Yoda et al., 1999, Quintana et al., 2006].

Recently, a multilevel design approach has been proposed as a way to mitigate the bias-variance tradeoff issue [Hirata, 2009]. It is inspired on stacked generalization, a classifier combination technique [Wolpert, 1992]. The basic idea consists in first designing several image operators according to the basic procedure described above, each one based on its own window, and then combining the resulting images in order to reach a kind of consensus. The combination rule is also determined by training. This idea of combining the results of previous level operators can be extended to an arbitrary number of training levels. The iterative training approach for designing morphological operators, previously reported in [Hirata et al., 2000b], is a particular case of the multilevel training scheme.

Iterative and multilevel approaches successively refine the result and that could be considered an explanation to the improved performance of multilevel operators relative to the performance of single level operators. While this is an intuitive explanation, a more formal explanation relies on the bias-variance decomposition of error. When two operators, based on windows $W_1$ and $W_2$, are sequentially composed, the resulting operator may depend on a neighborhood whose dimensions are up to $\delta_{W_2}(W_1)$ (the dilation of $W_1$ by $W_2$). This means that composed operators form a larger subspace of operators (smaller bias) and, since individual operators are based on not so large windows, their estimation error is kept small implying also in overall smaller estimation error (smaller variance).

Additional details as well as some experimental results of the multilevel approach can be find in [Hirata, 2009].

## *Data Model Related Issues*

While the two approaches above deal with constrained operator spaces, a third way relates to data modeling. For instance, an attempt to model the joint probabilities $P(\mathbf{X}, \mathbf{y})$ has been considered in [Dougherty and Barrera, 1997]. Another natural approach could rely on deformations of the observed patterns as a way to increase the amount of training data. However, that would require a careful selection of the deformation model, a difficult task in general. In addition, the design approach would become too content dependent.

A simple idea explored recently towards increasing the amount of training data is based on the observation that, for many image processing tasks, a relatively small resolution image is enough. For instance, many examples presented in this work, related to document image processing, consider images with spatial resolution of approximately 100dpi. Another reason to use such low resolution images is that otherwise a too large window would be required in the design process. At the same time, images are usually obtained in higher resolution (in the case of documents, 300dpi) and then its resolution is reduced by software. Rather than just reducing the resolution of the images for training, a possible approach is to consider different resolution reduction algorithms in order to obtain multiple low resolution images from each high resolution image. For instance, if four algorithms are used, then four low resolution images will be generated and they will correspond to a four times larger training data set.

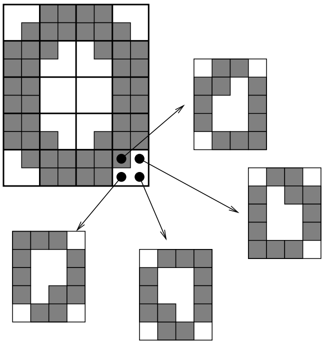


**Fig. 8** (a) Four down-samplings by choosing one pixel from each group of 4 pixels.

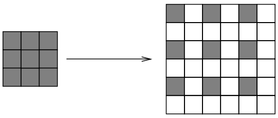


**Fig. 8** (b) A sparse window corresponding to a 3 × 3 window.

A simple resolution reduction algorithm is down-sampling by choosing equally spaced pixels from alternate rows and columns. A nice characteristic in this case is the fact that operator training can be done without changing the design procedure described above and without explicitly generating the low resolution images. For instance, the training data obtained by a 3 × 3 window from the four down-sampled images shown in Fig. 8(a) is equivalent to the training data obtained using the corresponding sparse window, shown in Fig. 8(b), directly on the original high resolution image.

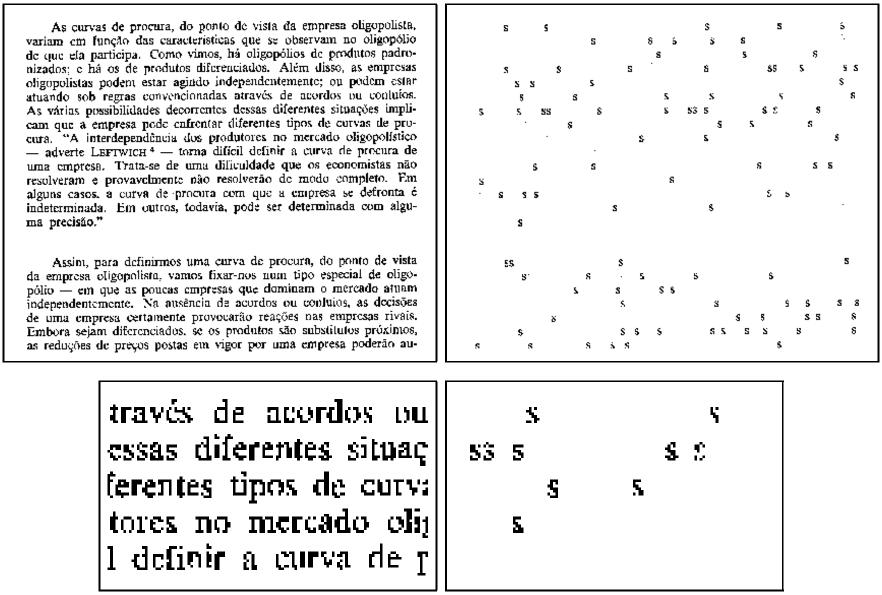
Thus, one only needs to consider high resolution training images and sparse windows that mimic the down sampling effects. This is a simple yet effective technique to increase the precision of trained operators. Preliminary results of this approach appear in [Hirata and Dornelles, 2009].

## 4 Application Examples

In this section a series of application examples are presented with the intention of illustrating the types of results that can be achieved using the training procedure described in this chapter. The pairs of images shown in this section refer to test images (independent of training images) and respective results obtained with the trained operators. For each example, the whole image shown in reduced scale provides an overall view of the result, while scale unchanged cropped region highlights details of the resulting image.

Although not exhibited here, for binary images, experimental results show that MAE agrees with visual perception: the smaller the MAE the better the resulting image according to visual inspection.

For each case, the training procedure followed a two-level scheme, using no more than a total of 10 training images. These training images were divided



As curvas de procura, do ponto de vista da empresa oligopolista, variam em função das características que se observam no oligopólio de que ela participa. Como vimos, há oligopólios de produtos padronizados; e há os de produtos diferenciados. Além disso, as empresas oligopolistas podem estar agindo independentemente; ou podem estar atuando sob regras convencionadas através de acordos ou conluios. As várias possibilidades decorrentes dessas diferentes situações implicam que a empresa pode enfrentar diferentes tipos de curvas de procura. "A interdependência dos produtores no mercado oligopolístico — adverte Leftwich [4] — torna difícil definir a curva de procura de uma empresa. Trata-se de uma dificuldade que os economistas não resolveram e provavelmente não resolverão de modo completo. Em alguns casos, a curva de procura com que a empresa se defronta é indeterminada. Em outros, todavia, pode ser determinada com alguma precisão."

Assim, para definirmos uma curva de procura, do ponto de vista da empresa oligopolista, vamos fixar-nos num tipo especial de oligopólio — em que as poucas empresas que dominam o mercado atuam independentemente. Na ausência de acordos ou conluios, as decisões de uma empresa certamente provocarão reações nas empresas rivais. Embora sejam diferenciados, se os produtos são substitutos próximos, as reduções de preços postas em vigor por uma empresa poderão au-

través de acordos ou
essas diferentes situaç
ferentes tipos de curv
tores no mercado oli
l definir a curva de p

**Fig. 9** Character segmentation: all occurrences of letter s have been identified. Some are perfectly segmented, while few are not.

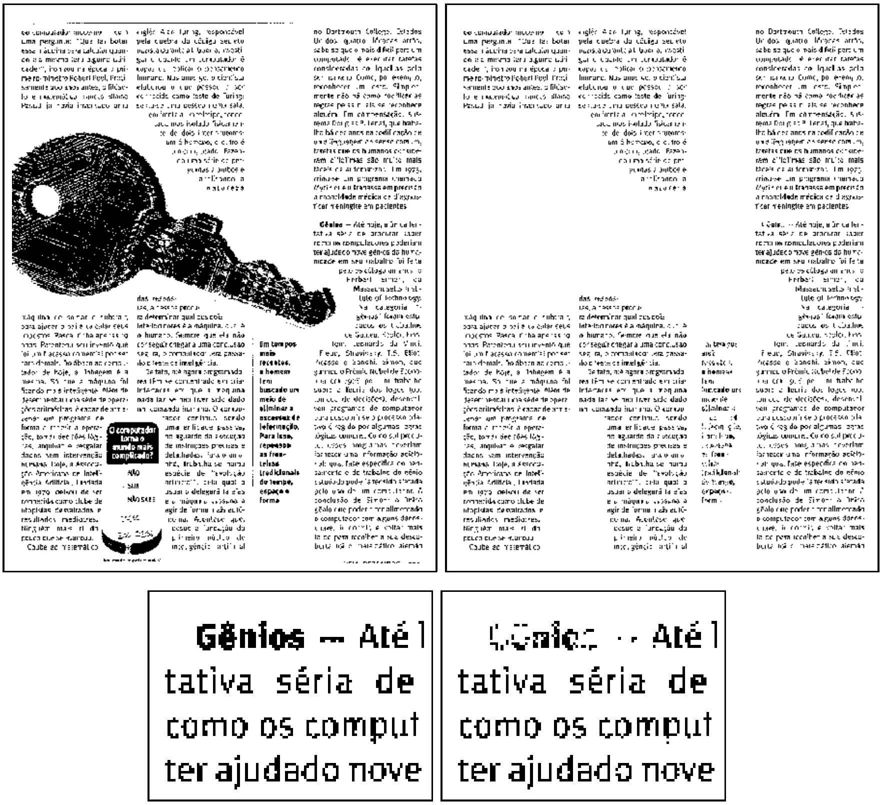
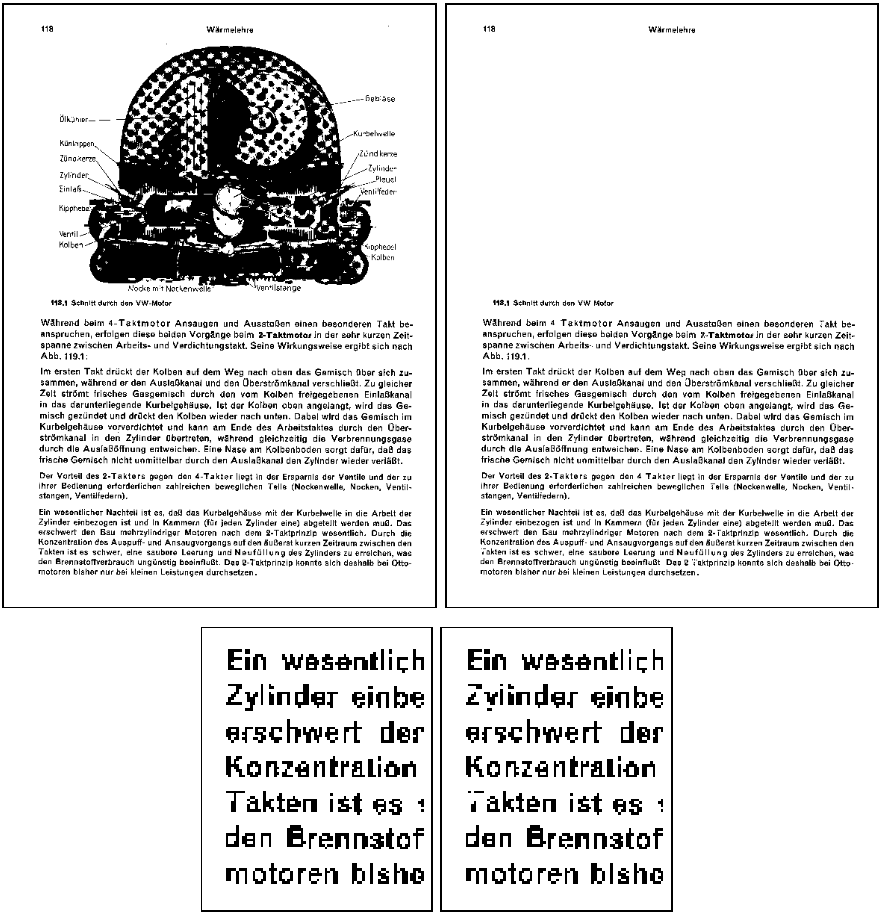


**Fig. 10** Text segmentation: boldface fonts, which are less common, are not very well segmented. Still, the segmentation result is very good.

in two parts, part of it (larger set) being used for the training of first-level operators and the rest (smaller set) for the training of second-level operators. The number of operators in the first-level training as well as their corresponding windows have been chosen manually. Low resolution images (100dpi) were used in all cases, except in the examples of noise filtering in text images (Figs. 13 and 14). For those, high resolution images in conjunction with sparse windows have been used (input images from *Tobacco800 Complex Document Image Database* [Lewis et al., 2006, Agam et al., 2006]).

## 5 Concluding Remarks

A state of the art overview on translation-invariant morphological operator ($W$-operator) learning from training images has been presented. From a machine learning perspective, the problem of learning $W$-operators can be seen as a classification problem. Specifically, in the binary case, the goal is to



118 Wärmelehre

118.1 Schnitt durch den VW-Motor

Während beim 4-Taktmotor Ansaugen und Ausstoßen einen besonderen Takt beanspruchen, erfolgen diese beiden Vorgänge beim **2-Taktmotor** in der sehr kurzen Zeitspanne zwischen Arbeits- und Verdichtungstakt. Seine Wirkungsweise ergibt sich nach Abb. 119.1:

Im ersten Takt drückt der Kolben auf dem Weg nach oben das Gemisch über sich zusammen, während er den Auslaßkanal und den Überströmkanal verschließt. Zu gleicher Zeit strömt frisches Gasgemisch durch den vom Kolben freigegebenen Einlaßkanal in das darunterliegende Kurbelgehäuse. Ist der Kolben oben angelangt, wird das Gemisch gezündet und drückt den Kolben wieder nach unten. Dabei wird das Gemisch im Kurbelgehäuse vorverdichtet und kann am Ende des Arbeitstaktes durch den Überströmkanal in den Zylinder übertreten, während gleichzeitig die Verbrennungsgase durch die Auslaßöffnung entweichen. Eine Nase am Kolbenboden sorgt dafür, daß das frische Gemisch nicht unmittelbar durch den Auslaßkanal den Zylinder wieder verläßt.

Der Vorteil des 2-Takters gegen den 4-Takter liegt in der Ersparnis der Ventile und der zu ihrer Bedienung erforderlichen zahlreichen beweglichen Teile (Nockenwelle, Nocken, Ventilstangen, Ventilfedern).

Ein wesentlicher Nachteil ist es, daß das Kurbelgehäuse mit der Kurbelwelle in die Arbeit der Zylinder einbezogen ist und in Kammern (für jeden Zylinder eine) abgeteilt werden muß. Das erschwert den Bau mehrzylindriger Motoren nach dem 2-Taktprinzip wesentlich. Durch die Konzentration des Auspuff- und Ansaugvorgangs auf den äußerst kurzen Zeitraum zwischen den Takten ist es schwer, eine saubere Leerung und Neufüllung des Zylinders zu erreichen, was den Brennstoffverbrauch ungünstig beeinflußt. Das 2-Taktprinzip konnte sich deshalb bei Ottomotoren bisher nur bei kleinen Leistungen durchsetzen.

118 Wärmelehre

118.1 Schnitt durch den VW Motor

Während beim 4 Taktmotor Ansaugen und Ausstoßen einen besonderen Takt beanspruchen, erfolgen diese beiden Vorgänge beim 2-Taktmotor in der sehr kurzen Zeitspanne zwischen Arbeits- und Verdichtungstakt. Seine Wirkungsweise ergibt sich nach Abb. 119.1.

Im ersten Takt drückt der Kolben auf dem Weg nach oben das Gemisch über sich zusammen, während er den Auslaßkanal und den Überströmkanal verschließt. Zu gleicher Zeit strömt frisches Gasgemisch durch den vom Kolben freigegebenen Einlaßkanal in das darunterliegende Kurbelgehäuse. Ist der Kolben oben angelangt, wird das Gemisch gezündet und drückt den Kolben wieder nach unten. Dabei wird das Gemisch im Kurbelgehäuse vorverdichtet und kann am Ende des Arbeitstaktes durch den Überströmkanal in den Zylinder übertreten, während gleichzeitig die Verbrennungsgase durch die Auslaßöffnung entweichen. Eine Nase am Kolbenboden sorgt dafür, daß das frische Gemisch nicht unmittelbar durch den Auslaßkanal den Zylinder wieder verläßt.

Der Vorteil des 2-Takters gegen den 4 Takter liegt in der Ersparnis der Ventile und der zu ihrer Bedienung erforderlichen zahlreichen beweglichen Teile (Nockenwelle, Nocken, Ventilstangen, Ventilfedern).

Ein wesentlicher Nachteil ist es, daß das Kurbelgehäuse mit der Kurbelwelle in die Arbeit der Zylinder einbezogen ist und in Kammern (für jeden Zylinder eine) abgeteilt werden muß. Das erschwert den Bau mehrzylindriger Motoren nach dem 2-Taktprinzip wesentlich. Durch die Konzentration des Auspuff- und Ansaugvorgangs auf den äußerst kurzen Zeitraum zwischen den Takten ist es schwer, eine saubere Leerung und Neufüllung des Zylinders zu erreichen, was den Brennstoffverbrauch ungünstig beeinflußt. Das 2 Taktprinzip konnte sich deshalb bei Ottomotoren bisher nur bei kleinen Leistungen durchsetzen.

Ein wesentlich
Zylinder einbe
erschwert der
Konzentration
Takten ist es
den Brennstof
motoren bisho

Ein wesentlich
Zylinder einbe
erschwert der
Konzentration
Takten ist es
den Brennstof
motoren bisho

**Fig. 11** Text segmentation: some less common characters such as capital T and Z are not perfectly segmented.

classify patterns defined within a given window as kernel or non-kernel elements. Because of the lattice isomorphism between $W$-operators and Boolean functions, the problem reduces to the problem of estimating a Boolean function that characterizes an operator. However, the fact that $W$-operators are endowed with properties and structures of lattice theory adds information that are not usually present in common classification problems. Such information open possibilities, for instance, to explore algebraic and structural constraints on the space of operators to mitigate the bias-variance tradeoff.

An important aspect of the learning approach described in this chapter is its flexibility with respect to applications. As shown in the provided examples, good results are obtained for a variety of image processing tasks. The results

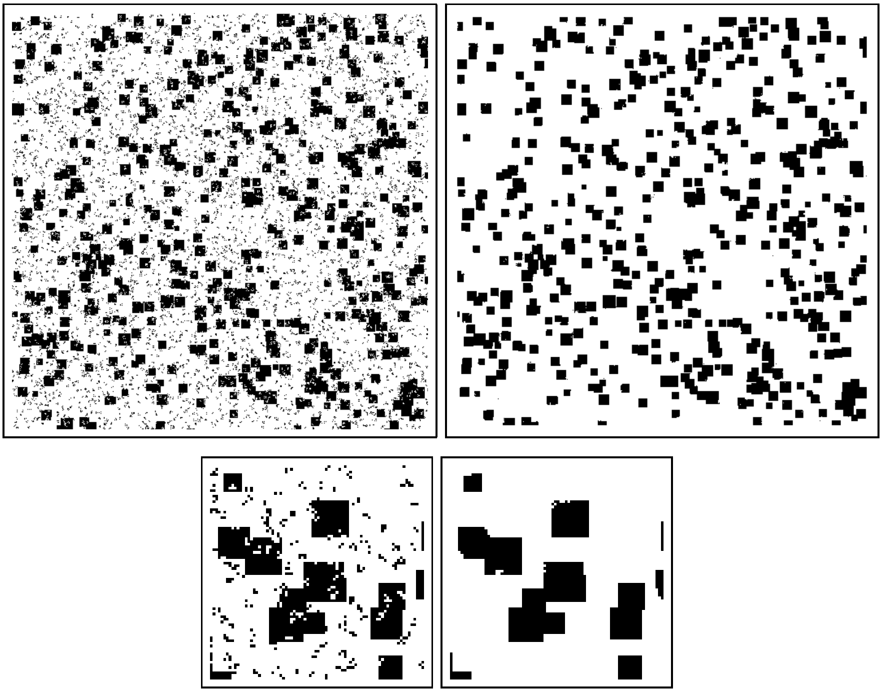
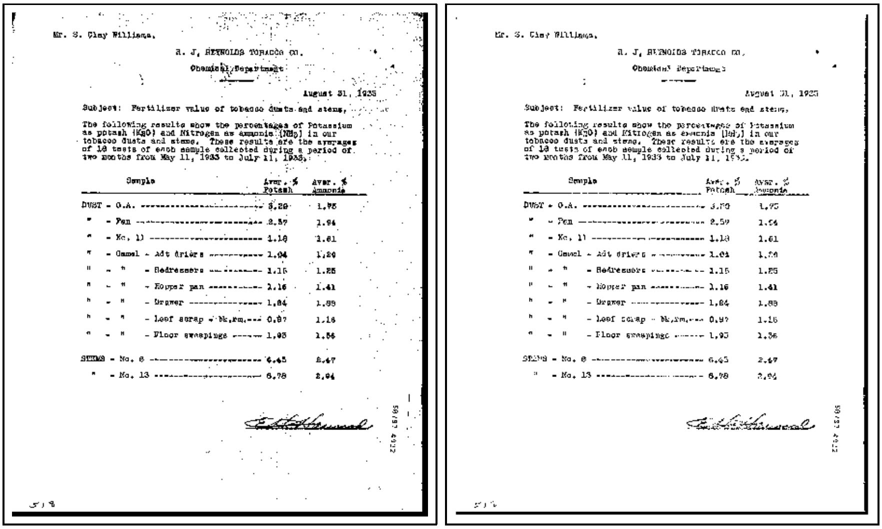


**Fig. 12** Boolean noise filtering from synthetic images.

obtained so far indicate that, as far as representative training images are provided, operators with good performance can be obtained. It should be noted that the learned operators are not rotation or scale invariant. Although this is a drawback, it can be overcome by providing training images that contemplate such cases.

Recently proposed multilevel training model offers the possibility to balance model bias and precision variance in order to obtain operators with better performance. Also, by considering the problem from a machine learning perspective, it becomes possible to consider a mixed multilevel approach. In such approach, not only geometrical patterns within a window, but also other data related to scale, orientation, or topological information, that are not easily captured within a window, could be integrated as features.

One of the challenges in the design procedure is finding an appropriate window, one that results in optimal MAE for the given number of training images. Although some attempts have been made in this direction [Martins Jr. et al., 2006], this issue needs more investigation. It corresponds to the classical feature selection problem. Furthermore, with the multilevel training model, an additional challenge is to find a proper multilevel training architecture, i.e., the number of operators in each training



Mr. S. Clay Williams,

R. J. REYNOLDS TOBACCO CO.

Chemical Department

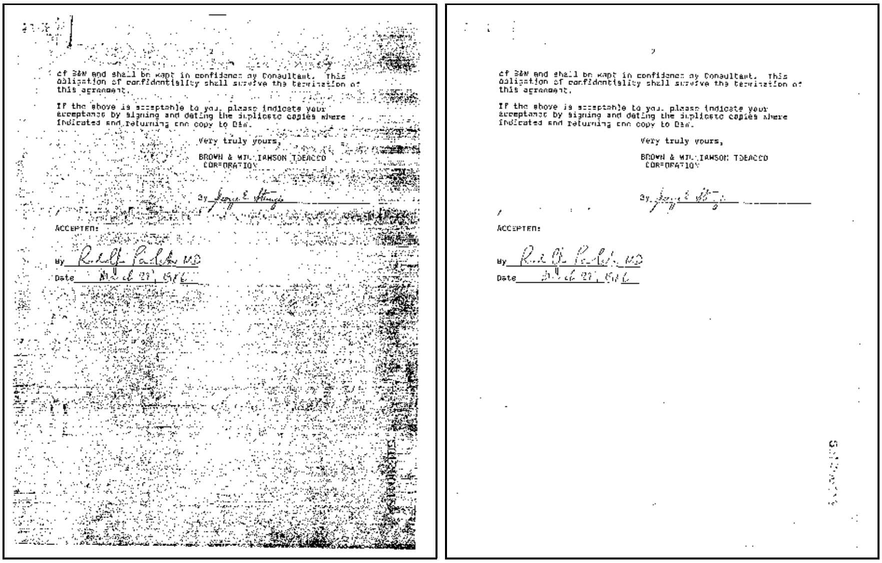
August 31, 1935

Subject: Fertilizer value of tobacco dusts and stems.

The following results show the percentages of Potassium as potash (K2O) and Nitrogen as ammonia (NH3) in our tobacco dusts and stems. These results are the averages of 18 tests of each sample collected during a period of two months from May 11, 1933 to July 11, 1933.

| Sample | Aver. % Potash | Aver. % Ammonia |
|---|---|---|
| DUST - O.A. | 3.29 | 1.75 |
| " - Fan | 2.57 | 1.94 |
| " - No. 11 | 1.18 | 1.61 |
| " - Camel - Adt driers | 1.04 | 1.20 |
| " " " - Redressers | 1.15 | 1.25 |
| " " " - Hopper pan | 1.16 | 1.41 |
| " " " - Drawer | 1.84 | 1.59 |
| " " " - Leaf scrap - Bk.Rm. | 0.87 | 1.16 |
| " " " - Floor sweepings | 1.93 | 1.36 |
| STEMS - No. 6 | 6.45 | 2.47 |
| " - No. 13 | 6.78 | 2.94 |

**Fig. 13** Noise filtering.



of B&W and shall be kept in confidence by Consultant. This obligation of confidentiality shall survive the termination of this agreement.

If the above is acceptable to you, please indicate your acceptance by signing and dating the duplicate copies where indicated and returning one copy to B&W.

Very truly yours,

BROWN & WILLIAMSON TOBACCO CORPORATION

By

ACCEPTED:

By M.D.

Date

**Fig. 14** Noise filtering.

level as well as their respective windows. Some preliminary works that use information theory related concepts [Santos et al., 2010] and genetic algorithm [Dornelles and Hirata, 2010] offer perspectives for possible approaches to this problem.

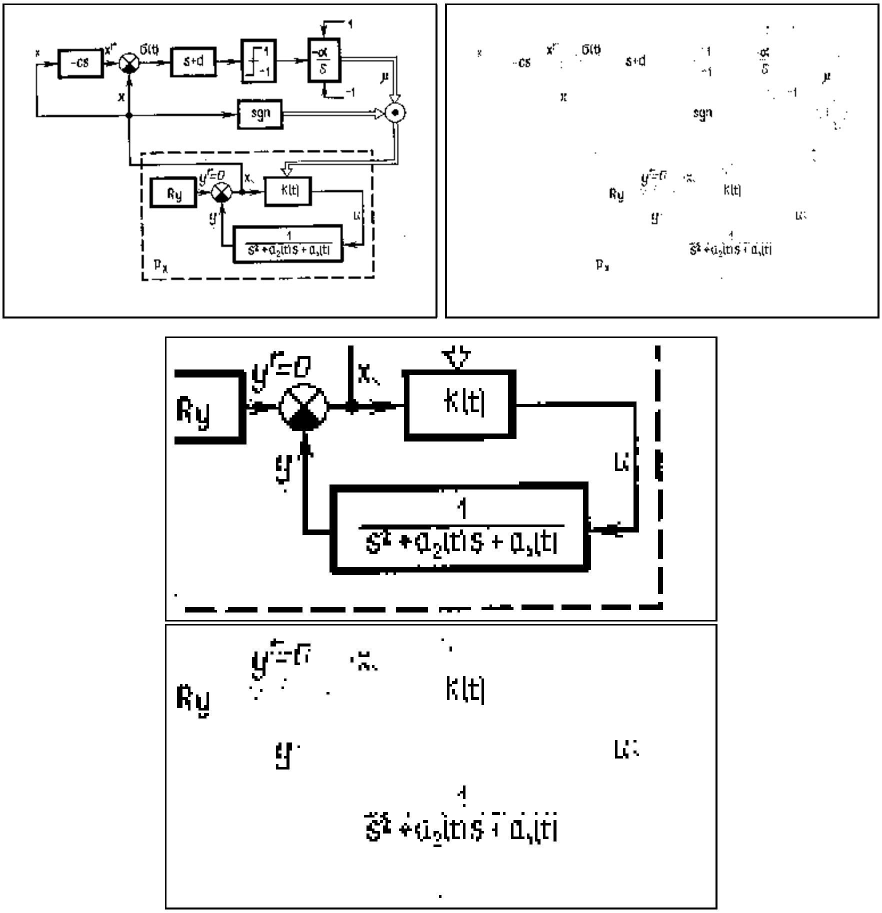


**Fig. 15** Character (text) segmentation: characters were relatively well segmented.

Several concepts as well as the design procedure described for binary images can be extended to gray scale images [Dougherty, 1992b]. A major difference is in the complexity of the local function that characterizes an operator. If the number of gray levels is $k$, then there are $k^{|W|}$ possible patterns within window $W$ and thus $k^{(k^{|W|})}$ possible functions from $W$ to $\{0, 1, \ldots, k-1\}$. From a machine learning perspective, gray-scale $W$-operators correspond to multiclass classifiers.

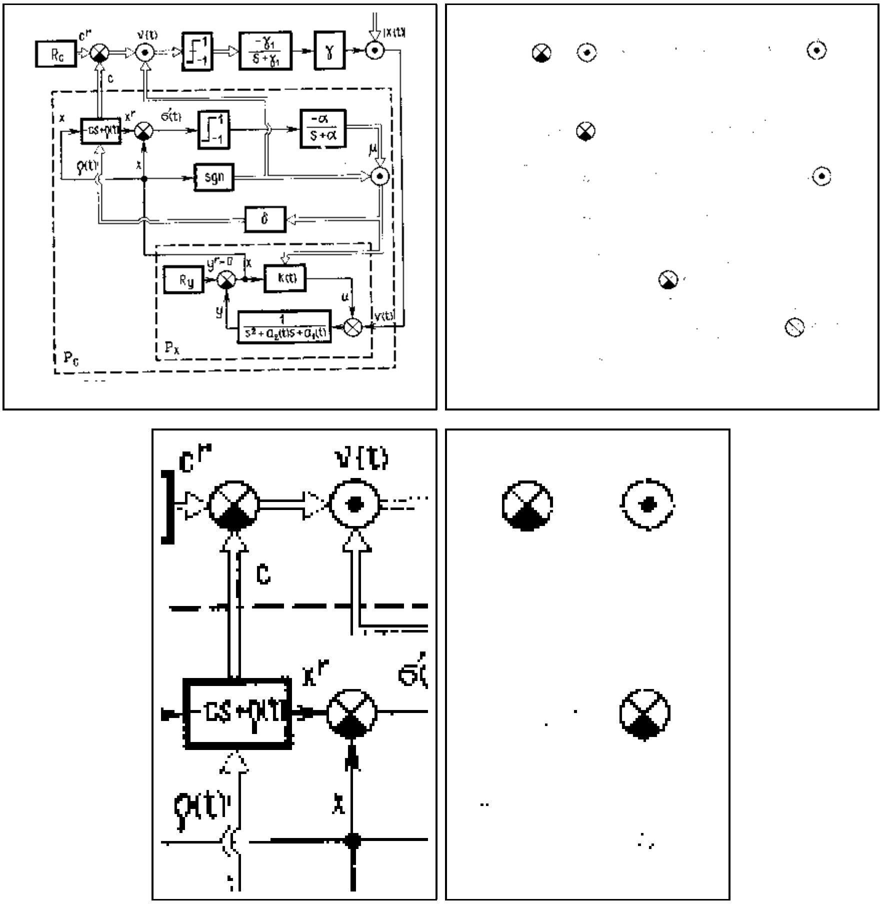


**Fig. 16** Circular object segmentation.

Designing gray-scale morphological operators is therefore expected to be both computationally and statistically much more challenging than designing binary image operators. Existing approaches are limited to subclasses of operators or very simple applications. The multilevel approach may be a way to make the extension to gray-scale operators viable for a large range of applications.

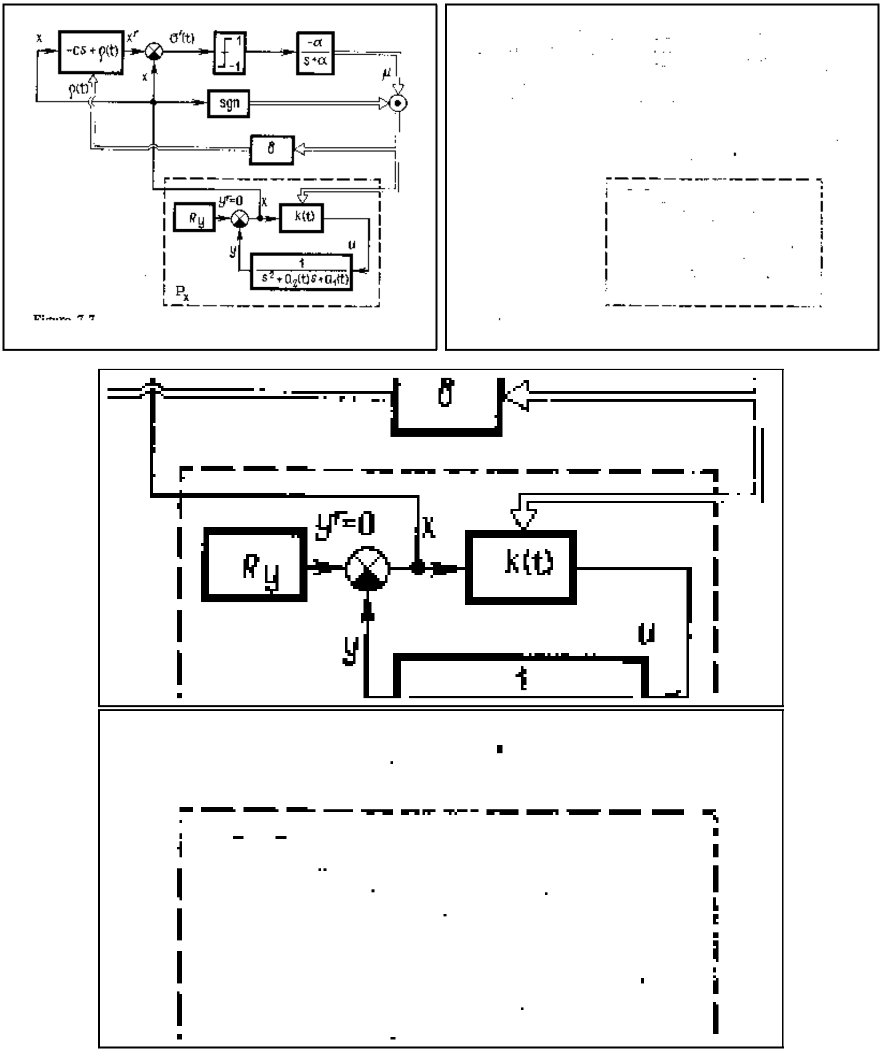


**Fig. 17** Dashed rectangle segmentation.

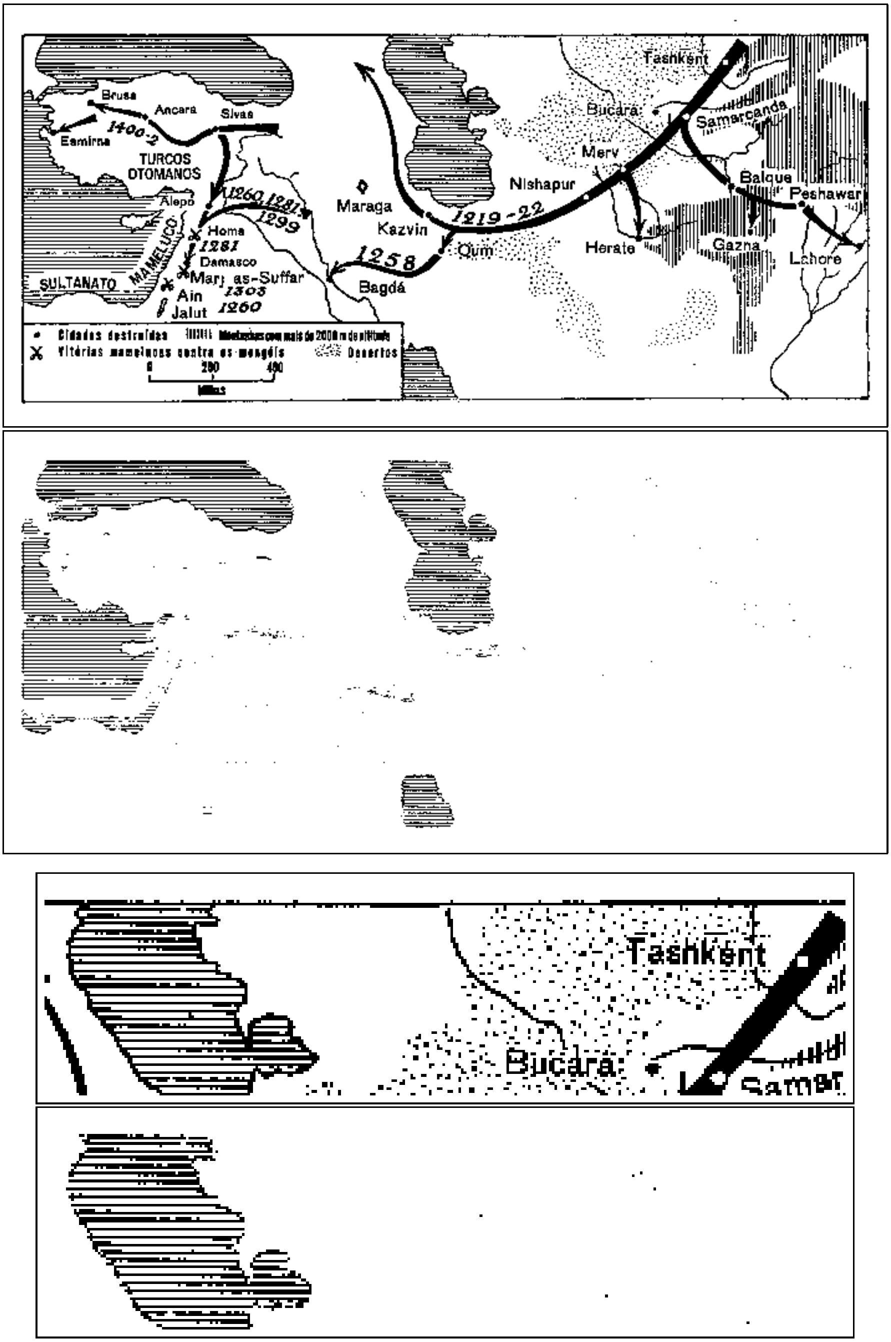


**Fig. 18** Texture segmentation.

**Acknowledgements.** The author is partly supported by CNPq, Brazil.

## References

[Agam et al., 2006] Agam, G., Argamon, S., Frieder, O., Grossman, D., Lewis, D.: The Complex Document Image Processing (CDIP) Test Collection Project. Illinois Institute of Technology (2006)

[Banon and Barrera, 1991] Banon, G.J.F., Barrera, J.: Minimal Representations for Translation-Invariant Set Mappings by Mathematical Morphology. SIAM J. Applied Mathematics 51(6), 1782–1798 (1991)

[Banon and Barrera, 1993] Banon, G.J.F., Barrera, J.: Decomposition of Mappings between Complete Lattices by Mathematical Morphology, Part I. General Lattices. Signal Processing 30, 299–327 (1993)

[Barrera et al., 1997] Barrera, J., Dougherty, E.R., Tomita, N.S.: Automatic Programming of Binary Morphological Machines by Design of Statistically Optimal Operators in the Context of Computational Learning Theory. Electronic Imaging 6(1), 54–67 (1997)

[Barrera and Salas, 1996] Barrera, J., Salas, G.P.: Set Operations on Closed Intervals and Their Applications to the Automatic Programming of Morphological Machines. Electronic Imaging 5(3), 335–352 (1996)

[Barrera et al., 2000] Barrera, J., Terada, R., Hirata Jr., R., Hirata, N.S.T.: Automatic Programming of Morphological Machines by PAC Learning. Fundamenta Informaticae 41(1-2), 229–258 (2000)

[Coyle and Lin, 1988] Coyle, E.J., Lin, J.-H.: Stack Filters and the Mean Absolute Error Criterion. IEEE Transactions on Acoustics, Speech and Signal Processing 36(8), 1244–1254 (1988)

[Dellamonica Jr. et al., 2007] Dellamonica Jr., D., Silva, P.J.S., Humes Jr., C., Hirata, N.S.T., Barrera, J.: An Exact Algorithm for Optimal MAE Stack Filter Design. IEEE Transactions on Image Processing 16(2), 453–462 (2007)

[Dornelles and Hirata, 2010] Dornelles, M.M., Hirata, N.S.T.: A genetic algorithm based method for determining two-level morphological operators. In: Proceedings of 17th International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 453–456 (2010)

[Dougherty, 1992a] Dougherty, E.R.: Optimal Mean-Square N-Observation Digital Morphological Filters I. Optimal Binary Filters. CVGIP: Image Understanding 55(1), 36–54 (1992a)

[Dougherty, 1992b] Dougherty, E.R.: Optimal Mean-Square N-Observation Digital Morphological Filters II. Optimal Gray-Scale Filters. CVGIP: Image Understanding 55(1), 55–72 (1992b)

[Dougherty and Barrera, 1997] Dougherty, E.R., Barrera, J.: Bayesian Design of Optimal Morphological Operators Based on Prior Distributions for Conditional Probabilities. Acta Stereologica 16(3), 167–174 (1997)

[Dougherty and Loce, 1993] Dougherty, E.R., Loce, R.P.: Optimal Mean-Absolute-Error Hit-or-Miss Filters: Morphological Representation and Estimation of the Binary Conditional Expectation. Optical Engineering 32(4), 815–827 (1993)

[Dougherty and Loce, 1994] Dougherty, E.R., Loce, R.P.: Precision of Morphological-Representation Estimators for Translation-invariant Binary Filters: Increasing and Nonincreasing. Signal Processing 40, 129–154 (1994)

[Dougherty and Lotufo, 2003] Dougherty, E.R., Lotufo, R.A.: Hands-on Morphological Image Processing. SPIE Press, San Jose (2003)

[Guyon and Elisseeff, 2003] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)

[Harvey and Marshall, 1996] Harvey, N.R., Marshall, S.: The Use of Genetic Algorithms in Morphological Filter Design. Signal Processing: Image Communication 8(1), 55–71 (1996)

[Hastie et al., 2009] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning – Data Mining, Inference and Prediction, 2nd edn. Springer, Heidelberg (2009)

[Heijmans, 1994] Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, Boston (1994)

[Heijmans and Ronse, 1990] Heijmans, H.J.A.M., Ronse, C.: The Algebraic Basis of Mathematical Morphology – Part I: Dilations and Erosions. Computer Vision, Graphics and Image Processing 50, 245–295 (1990)

[Hirata, 2009] Hirata, N.S.T.: Multilevel training of binary morphological operators. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(4), 707–720 (2009)

[Hirata et al., 2002] Hirata, N.S.T., Barrera, J., Terada, R., Dougherty, E.R.: The Incremental Splitting of Intervals Algorithm for the Design of Binary Image Operators. In: Talbot, H., Beare, R. (eds.) Mathematical Morphology – Proceedings of the 6th International Symposium – ISMM 2002, pp. 219–228 (2002)

[Hirata and Dornelles, 2009] Hirata, N.S.T., Dornelles, M.M.: The use of high resolution images in morphological operator learning. In: Proceedings of the XXII Brazilian Symposium on Computer Graphics and Image Processing, pp. 141–148 (2009)

[Hirata et al., 2000a] Hirata, N.S.T., Dougherty, E.R., Barrera, J.: A Switching Algorithm for Design of Optimal Increasing Binary Filters Over Large Windows. Pattern Recognition 33(6), 1059–1081 (2000a)

[Hirata et al., 2000b] Hirata, N.S.T., Dougherty, E.R., Barrera, J.: Iterative Design of Morphological Binary Image Operators. Optical Engineering 39(12), 3106–3123 (2000b)

[Hirata Jr. et al., 2000] Hirata Jr., R., Dougherty, E.R., Barrera, J.: Aperture Filters. Signal Processing 80(4), 697–721 (2000)

[Joo et al., 1990] Joo, H., Haralick, R.M., Shapiro, L.G.: Toward the Automatic Generation of Mathematical Morphology Procedures Using Predicate logic. In: Proc. of the Third International Conference on Computer Vision, Osaka, Japan, pp. 156–165 (1990)

[Lee et al., 1999] Lee, W.-L., Fan, K.-C., Chen, Z.-M.: Design of Optimal Stack Filters Under the MAE Criterion. IEEE Transactions on Signal Processing 47(12), 3345–3355 (1999)

[Lewis et al., 2006] Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a test collection for complex document information processing. In: Proc. 29th Annual Int. ACM SIGIR Conference, pp. 665–666 (2006)

[Lin and Kim, 1994] Lin, J.-H., Kim, Y.-T.: Fast Algorithms for Training Stack Filters. IEEE Transactions on Signal Processing 42(4), 772–781 (1994)

[Maragos and Schafer, 1987] Maragos, P., Schafer, R.W.: Morphological Filters: Part II: Their Relations to Median, Order Statistic, and Stack-Filters. IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-35, 1170–1184 (1987)

[Maragos, 1985] Maragos, P.A.: A Unified Theory of Translation-invariant Systems with Applications to Morphological Analysis and Coding of Images. PhD thesis, School of Electrical Engineering - Georgia Institute of Technology (1985)

[Martins Jr. et al., 2006] Martins Jr., D.C., Cesar Jr., R.M., Barrera, J.: W-operator window design by minimization of mean conditional entropy. Pattern Analysis and Applications 9, 139–153 (2006)

[Matheron, 1975] Matheron, G.: Random Sets and Integral Geometry. John Wiley, Chichester (1975)

[Matheron and Serra, 2002] Matheron, G., Serra, J.: The birth of mathematical morphology. In: Talbot, H., Beare, R. (eds.) Mathematical Morphology – Proceedings of the 6th International Symposium – ISMM 2002, Sydney, Australia, pp. 1–16. Commonwealth Scientific and Industrial Research Organisation (2002)

[Quintana et al., 2006] Quintana, M.I., Poli, R., Claridge, E.: Morphological algorithm design for binary images using genetic programming. Genetic Programming and Evolvable Machines 7(1), 81–102 (2006)

[Ronse and Heijmans, 1991] Ronse, C., Heijmans, H.J.A.M.: The Algebraic Basis of Mathematical Morphology – Part II: Openings and Closings. Computer Vision, Graphics and Image Processing: Image Understanding 54, 74–97 (1991)

[Santos et al., 2010] Santos, C.S., Hirata, N.S.T., Hirata Jr., R.: An information theory framework for two-stage binary image operator design. Pattern Recognition Letters 31(4), 297–306 (2010)

[Schimitt, 1989] Schimitt, M.: Mathematical morphology and artificial intelligence: an automatic programming system. Signal Processing 16(4), 389–401 (1989)

[Serra, 1982] Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)

[Serra, 1988] Serra, J.: Image Analysis and Mathematical Morphology. Theoretical Advances, vol. 2. Academic Press, London (1988)

[Soille, 2003] Soille, P.: Morphological Image Analysis, 2nd edn. Springer, Berlin (2003)

[Tăbuş et al., 1996] Tăbuş, I., Petrescu, D., Gabbouj, M.: A training Framework for Stack and Boolean Filtering – Fast Optimal Design Procedures and Robustness Case Study. IEEE Transactions on Image Processing 5(6), 809–826 (1996)

[Vogt, 1989] Vogt, R.: Automatic Generation of Morphological Set Recognition Algorithms. Springer, Heidelberg (1989)

[Wendt et al., 1986] Wendt, P.D., Coyle, E.J., Gallagher Jr., N.C.: Stack Filters. IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-34(4), 898–911 (1986)

[Wolpert, 1992] Wolpert, D.H.: Stacked generalization. Neural Networks 5(2), 241–259 (1992)

[Yoda et al., 1999] Yoda, I., Yamamoto, K., Yamada, H.: Automatic Acquisition of Hierarchical Mathematical Morphology Procedures by Genetic Algorithms. Image and Vision Computing 17(10), 749–760 (1999)

[Yoo et al., 1999] Yoo, J., Fong, K.L., Huang, J.-J., Coyle, E.J., Adams III, G.B.: A Fast Algorithm for Designing Stack Filters. IEEE Transactions on Image Processing 8(8), 1014–1028 (1999)

# Chapter 4
# Task-Specific Salience for Object Recognition

Jerome Revaud, Guillaume Lavoue, Yasuo Ariki, and Atilla Baskurt

**Abstract.** Object recognition is a complex and challenging problem. It involves examining many different hypothesis in terms of the object class, position, scale, pose, etc., but the main trend in computer vision systems is to lazily rely on the brute force capacity of computers, that is to explore every possibilities indifferently. Sadly, in many case this scheme is way too slow for real-time or even practical applications. By incorporating salience in the recognition process, several approaches have shown that it is possible to get several orders of speed-up. In this chapter, we demonstrate the link between salience and cascaded processes and show why and how those ones should be constructed. We illustrate the benefits that it provides, in terms of detection speed, accuracy and robustness, and how it eases the combination of heterogeneous feature types (i.e. dense and sparse features) by some innovating strategies from the state-of-the-art and a practical application.

**Keywords:** task-specific salience, cascades, feature combination, optimization.

## 1 Introduction

When it comes to understand a new image, a human being immediately "knows" which spots to focus on to get a fast understanding of the scene.

Jerome Revaud · Guillaume Lavoue · Atilla Baskurt
Universite de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621, France
e-mail: {firstname.lastname}@liris.cnrs.fr

Yasuo Ariki
CS17 Media Laboratory
Kobe University,
Japan
e-mail: ariki@kobe-u.ac.jp

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 59–85.
springerlink.com

This intuition has a lot to do with the pre-processing done automatically by the pre-cognitive system in our brain in order to predict interest areas in the image. This phenomena is materialized through the eyes by a series of extremely fast small jumps of the iris known as "saccades", in which the eyes successively stop for a few milliseconds on some spots perceived as *salient*. The main purpose of this saccadic movement is to save body resources by sensing only small parts of the scene with a greater resolution. For instance, flat areas like the sky are of smaller interest for scene understanding, so almost no time is spent analyzing them. Interestingly enough, one can draw a central parallel between the human (or animal) vision system and what we would like to introduce in this chapter. More precisely, the very notion of *salience* is worth investigating further with respect to its application in computer vision for object recognition. We mean here by object recognition the joint detection and localization of an *object* in the widest sense, i.e., from a specific object (e.g. this book) to a class of objects (e.g. cars, faces). Some recognition examples are shown in Figure 6 in this case of specific object detection.

In a general frame, salience is often defined as the state or quality of an item that stands out relatively to neighboring items. Although this definition is not inexact strictly speaking, it needs to be refined with respect to the goal of this chapter. Similarly to the way in which the saccades are generated by a neuronal mechanism that bypasses time-consuming circuits to restrain most of the work to a few salient areas, we will refer to the term *salience* as a low-level, easy to extract, property of an image region that is used in the perspective of reducing computations for a specific task of object recognition. Saliency maps have already been defined by [Itti et al., 1998] in order to extract prominent, singular image spots using different channels (luminance, color, movement, texture, and so on). Subsequent works along the same line include those of [Walther and Koch, 2006] and [Paletta and Fritz, 2008]. Those systems however focus on biologically plausible systems for general purpose detection, whereas in this chapter we put aside the biological aspect and rely on more specific definitions of salience which depend on the exact tasks to accomplish. Searching for cats or for faces, for example, would involve different definitions of salience although the general principles mentioned earlier would stay true (i.e. salience as a coarse first-pass in order to reduce information processing) and in the same kind of idea to the ones used for extracting saliency maps. The reader interested by this biological aspect of salience can found further details about cognitive approaches in the book series *Attention in Cognitive Systems*, Springer (e.g. see [Paletta and Tsotsos, 2008]).

A second major point, closely related to the utilization of salience in image analysis, is the combination of different types of image features for object recognition. Although those two themes could theoretically be discussed separately, it appears more and more obvious as one deepens the subject that the utilization of salience is of a great help for associating together different types of features. Indeed, the key insight is that by dividing the detection process into several stages (with the first one being salient region detection), one can replace a single decision on a multi-type feature set by a succession of decisions each of them concerning a single type of features. This cascade-based detection scheme is not new in itself [Viola and Jones, 2001, Fleuret and Geman, 2001, Elad et al., 2001]; however, it is only recently that some approaches have taken advantage of its potential to combine different types of features in order to increase performances [Gu et al., 2009, Fleuret and Geman, 2008]. We give a practical and explicit example in section 3 to concretely illustrate how to use salience in a cascade-based scheme in order to firstly (a) combine different types of features (e.g. sparse and dense types, see §2 and §2), possibly incompatible in many classical frameworks, and secondly (b) to increase the detection time to its utmost.

In the following of this chapter, we begin by presenting in details an extended definition of salience in section 2. This includes various aspects such as describing the global shape of a salience-oriented detection process (§2), explaining how to combine different feature types (§2) and how to build a salience detector according to the exact task to accomplish (§2). For each point, we highlight the main divergence points with mainstream works in terms of advantages or drawbacks. We then put those principles into application in a specific object recognition framework (section 3) and we provide quantitative and comparative evaluations (§3) to demonstrate the superiority of this kind of approaches on the state-of-the-art. Finally, we conclude and give some perspectives in section 4.

## 2 An Extended Definition of Salience

Roughly speaking, the main trend in object recognition is as follow: (1) to exhaustively extract features in the image or in a rectangular image window like in [Dalal and Triggs, 2005], and (2) to use the resulting feature vector to take a decision upon its content (typically: a binary decision of the type target/clutter) through a classifier (either generative or discriminative). Although this description may sound somehow simplistic, most works can roughly fit with it. Indeed, this scheme can work pretty well in most situations, but in this section we want to highlight the drawbacks of this strategy with respect to salience-oriented methods. As presented in the introduction, salience is to be seen in this study as local, low-level information of the image which can be used to save computations and enable more complex processing including combining different feature types. To sum up, this definition includes three aspects:

1. The structure of the detection process: how the processing pipeline should be organized to make the most of salience.
2. The definition of salience: how it is precisely defined with respect to the object or the class of objects to recognize.

3. The feature combination: how different types of features can be used at the same time in such a framework.

We now describe in detail each of these aspects in order to give a precise though general description of a salience oriented framework in object recognition. In the same time, a description of mainstream approaches is provided to enhance the advantages provided by the use of salience in the detection process.

## *The Structure of the Detection Process*

A critical interest of using salience lies in the fact of being able to cut off the computations as soon as possible, i.e. when a negative outcome becomes evident, one should stop spending energy into the detection process. For instance, it seems trivial for us that searching for faces or cars in uniform areas such as a blue sky is useless. Yet, the computer vision world usually do not take into account such clues. An historical reason for this fact is that many object recognition methods were actually adapted from the essentially different task of image classification [Harzallah et al., 2009]. Image classification consists of extracting global image features in a first step and taking a decision about its content in a second step (typically, does it contain a dog, buildings, etc.). Experiments on difficult datasets like Pascal VOC dataset [Everingham et al.] have shown impressive results, but this is still far from what would be expected from a human being. Indeed, it appears more intuitive to us that classifying a picture as "dog" would first imply to recognize *where* is the dog before knowing that a dog is in picture. As a result, we believe that producing better results also involves a better understanding of the image content: what is in there, where, and how, for instance.

The problem of sliding window approaches like [Dalal and Triggs, 2005] is that it involves to examine tens of thousands sub-images per image independently (according to [Gu et al., 2009]), each time applying the same feature extraction and classification process to each of them. This simple scheme do enable invariance to translation and scale (invariance to pose and noise being handled by the classifier and invariance to illumination by the features), but the same amount of computations is spent whether the considered image region is plain blue sky or not. Intuitively, one can understand that this procedure is far from the optimum computationally speaking and that many time that could be reinvested into more complex tasks, is lost. Moreover, such approach becomes dramatically costly for detecting more than one type of object provided that the window aspect ratio, the features or the classifiers used are different. Obviously, using salience here in such framework could save a large amount since it would enable to eliminate entire regions on the test image basing only on a few very simple low-level tests. For example, Vedaldi et al. [Vedaldi et al., 2009] have reported a decrease of processing time per image from 27 hours to 67 seconds when they used salience to filter the mass of possible windows. Fleuret and Geman [Fleuret and Geman, 2008] have also noted that this slowness is an important drawback for major applications using detection, such as real-time detection in video or indexing of large databases of pictures since they require very low computation times in order to process several scenes per second.

A solution for such issue is to decompose the recognition process into several successive steps. This is known in the literature as *cascade* [Viola and Jones, 2001, Elad et al., 2001] or *coarse-to-fine* recognition [Fleuret and Geman, 2001] ([Gangaputra and Geman, 2006] provides a unification of those two related concepts). In both case, the recognition process is fragmented so as to enable an early ending: instead of taking a decision on the full available knowledge, like is done typically in classification approaches (with Support Vector Machine for instance), the global decision function $F : \Re^n \rightarrow \{0, 1\}$ is divided into a set of smaller, more compact functions $f_i : \Re^{q_i} \rightarrow \{0, 1\}$ that can be evaluated sequentially:

$$F(\mathbf{x}) \equiv \bigotimes_i f_i(\mathbf{x}_i) \text{ with } \forall i \, q_i < n, \text{ i.e. } \mathbf{x}_i \subset \mathbf{x}$$

where $\mathbf{x}$ represents the full feature vector and $\bigotimes$ is a generic sequence operator that can take various forms. Since in the real world, the vast majority of input vectors are *clutter*s [Elad et al., 2001, Eveland et al., 2005], the purpose of the chain of *subclassifiers* $\{f_i\}$ is to label $\mathbf{x}$ as clutter as fast as possible. Hence, each $f_i$ is dedicated to clutter detection rather than true positive labelling: a single negative decision suffices to ensure a final negative detection and makes the classification process exits. We illustrate this process in Figure 1.(b) with a 3 layered cascade (in the following, we refer to the term *layer* as a stage in the detection process corresponding to the evaluation of a single subclassifier). Of course, in order to keep the interest of the procedure, each of those subclassifiers $f_i$ has to depend on a reduced set of features $\mathbf{x}_i \subset \mathbf{x}$ so as to correspondingly reduce their complexity. Thus, by ordering the $f_i$ in a smart way - i.e. the faster decisions are evaluated first - one can avoid a lot of computations and hence efficiently simulate the effect of salience. In fact, it is natural to let later stages use more target-class prototypes than earlier ones, thus allowing the classifier to more closely model the effective support of the distribution [Eveland et al., 2005].

**Construction of the cascade.** Intuitively, one may be afraid that the approximation of a global decision function by a sequence of simpler ones may deteriorate the performances, fortunately it is not necessarily the case. By ensuring a null false negative rate for each $f_i$, it is straightforward to see that no true detection is forgotten while in the same time a vast field of clutter is evacuated from the decision process. A nearly null false negative rate is achieved in practice by adjusting the decision surface of each $f_i$, usually, by varying a constant bias term added at the output of the classifier. This

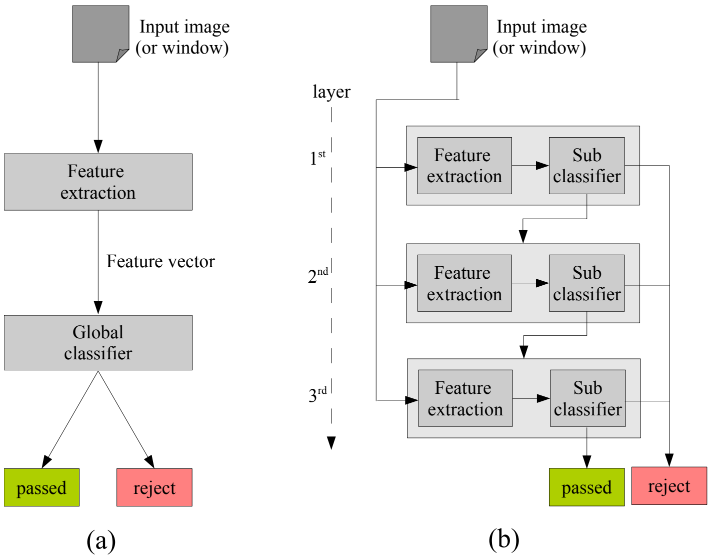


**Fig. 1** Comparison between a classical detection process and a cascaded detection process. (a) Standard process; (b) cascade with 3 layers. Each layer is activated if the previous layer returns a positive response.

operation however needs some precautions in order to avoid over-fitting issues and is therefore generally done by using a validation set disjoint of the set used to train the classifier. Since in practice a null false negative rate may lead to classify every sample as positive, the false positive rate has also to remain as low as possible; this tradeoff between these two conditions is achieved by adjusting the bias term or, in certain cases, by complexifying the subclassifiers. The final algorithm is thus to add layers iteratively, each time ensuring for the new subclassifier that both targets (i.e. maximum false negative rates and minimum true negative rate) are met. The procedure stops when the overall target rates are met [Viola and Jones, 2001].

The general procedure for training a cascade involves boostrapping [Lampert, 2010, Viola and Jones, 2004]. In the context of machine learning, boostrapping denotes the action of reusing negative samples wrongly classified as positive by the previous classifier to train the current classifier. In a cascade, it is straightforward to put in practice: it suffices to run the (incomplete) cascade detector on background images that are known not to contain the model object and to collect false positives to train the new cascade layer (in addition to true positive samples collected in the same fashion on model images). As a consequence, the same learning algorithm will yield a different subclassifier (in fact more complex) because the set of negative samples is different: they gradually become harder and harder to classify as the layer depth increases.

Finally, a promising track seems to be connecting the cascade layers together. Instead of disjoint layers trained independently as in the precursory work of Viola and Jones [Viola and Jones, 2001], some works have recently investigated the possibility of reusing the subclassifier real-valued results (i.e. before binary thresholding). Huang et al. [Huang et al., 2004], for instance, have proposed to reinject the real-valued result of the previous subclassifier $f_{i-1}$ in the current subclassifier $f_i$ to simulate the integration of the features used for $f_{i-1}$ in $f_i$. This have been shown to significantly improve the results as well as decreasing the complexity of the decision functions. In a different fashion, Felzenszwalb et al. [Felzenszwalb et al., 2010] summed the real-valued subclassifiers responses across the layers. Since this sum actually symbolizes the distance to the model of what is found instead in the image (i.e., a perfect model match would yield a null distance), the detection process exits as soon as the sum exceeds a fixed threshold, different for each layer. It enables more flexibility and robustness as even if one important model part was occluded it would not necessarily stop the detection process.

To conclude this section, cascades are the ideal way of making the most of salience: the idea is to smartly order the succession of tests in the cascade such that the first test would also be the least expensive one. This can be seen as a generalization of the single-level of salience used by our eyes, in which the saccades are generated by a neuronal mechanism that bypasses time-consuming circuits to directly activate the eye muscles. In a cascaded approach, salience operates at several levels although its “core activity” only concerns the first layer. For instance, in the face detector of Viola and Jones [Viola and Jones, 2001], the full cascade contains 38 layers and more than 6000 individual features (i.e. 158 features per layer, on average), but the detector of the first layer only takes its decision based on a single feature. In sum, it acts as a salience detector (for faces) by pre-selecting a set of interest regions at an extremely low computational price, that is, only considering low-level pixel information. Subsequent layers iteratively refines those detections by increasing the number of tests/features and hence the complexity level of the analysis. There also exists intermediary works between hard cascades like [Viola and Jones, 2001, Elad et al., 2001] and soft cascades in which the decision functions are more connected [Huang et al., 2004, Felzenszwalb et al., 2010].

## *Optimal Salience Detector for a Given Task*

Now that a global framework has been set up for object recognition, we need to dwell on the definition of salience that is going to be used practically for detection. In fact, there exist two possibilities: either define an ad hoc, task-specific salience, either use an existing generic detector. Empirically, class object recognition often deals with task-specific salience definitions while single object recognition usually takes advantage of existing generic detectors.

We now give a review of the advantages and drawbacks of both techniques and when or why each one should be used.

**Generic salience detectors.** To begin with, generic salience detectors propose to select a subset of spots from all possible spots within an image. Here, we mean by "spot" a set of connected pixels (e.g. lines or regions) although it can reduce to single points in the oriented image scale-space (e.g. see interest point detectors, or *keypoints*, like SIFT [Lowe, 2004]). The reduction of possibilities due to this selection is generally massive, hence it can be thought as equivalent to the first layer of a cascade: indeed, only those spots selected by the generic detector are further analyzed in the rest of the detection process.

The extraction generally follows a low-level rule, as salience implies, defined in term of mathematical invariance to a subset of expected transformations: namely, lighting change, rescaling, in-plane rotation and even affine transformations [Rothganger et al., 2006] as an approximation of perspective effects. Edges or regions, for instance, are robust to most usual transformations. In the literature, those spots are often addressed as "sparse features" or "local features" due to their limited number and localized aspect. In the following of the paragraph, we drop the term "sparse" for simplicity. Existing detectors has been developed from almost the very beginning of image processing and a non-exhaustive list includes edge detectors (Canny [Canny, 1986]), keypoint detectors (SIFT [Lowe, 2004], MSER [Matas et al., 2002], Hessian-Harris corners [Harris and Stephens, 1988]) and region detectors like[Arbelaez et al., 2009].

Since the properties used for extracting these features are invariant to real-world transformation, a widely used technique for *specific* object detection is to describe the model object by a set of these local features in the training stage; and to search those features in the detection stage basing firstly on their individual appearance (i.e. pairwise matches between keypoints, lines or regions) and secondly on their geometrical consistency (i.e. relative positions of the features) [Lowe, 2004, Rothganger et al., 2006]. In this case, the feature extraction step is equivalent to a first cascade layer based on salience, at the difference that all features are extracted at once before further processing (i.e. visual and geometrical matching, here corresponding to a second cacade layer) are applied. To sum up, salience is used in this process to initially simplify the huge search space that is an image into a much smaller set of features, easier to handle separately.

Using those salience detectors could also appear to be an excellent solution for recognizing class of objects; however, there are some drawbacks. The main problem lies in the definition of the features themselves which are too much specific: their low-level nature makes them not invariant to semantic transformations that occur in the case of intra-class variations. Indeed in the case of specific object detection, the robustness of the system is usually expected to be higher than for detection of class of objects (e.g. invariance to rotation is added) in order to compensate for the extra-simplicity of the task. To solve this issue, several solutions exist. Chum and Zisserman have proposed an original voting system which elects a subset of image windows in which the model is likely to be found [Chum and Zisserman, 2007]. For that purpose, pairs of a keypoint and an associated rectangle are learned from training images based on the most discriminative keypoints with respect to the model objects. During test, votes in the form of rectangular windows are generated from the learned pairs and are aggregated using mean-shift clustering, resulting in a small set of windows left for more complex examination [Vedaldi et al., 2009]. A similar principle was proposed using regions instead of keypoints by Gu et al [Gu et al., 2009]. Finally, bag-of-features (BOF) is also a popular way to bypass the keypoint specificity: by clustering keypoints, a dictionary is formed to generate keypoint histograms which ignore the spatial relationships among the points and thus give more flexibility [Csurka et al., 2004]. Such techniques are however not always sufficient for all tasks. Indeed, a single generic salience detector considers only one mathematical property and thus may not be adapted to every possible object classes. Typically, keypoint detectors works well for textured objects like motos or books; but they become quite inefficient when it comes to plain object like coffee mugs or swans, as well as their derivatives like bag-of-features. As a result, more dedicated salience detectors have to be derived in an ad hoc manner, usually resulting in more than one cascade layers contrary to the case of generic salience detectors.

**Ad hoc salience.** Higher-level objects like faces or pedestrians offer few invariant low-level features because of the important amount of intra-class variations. As a result, generic detectors cannot spot them reliably. For such difficult cases, it is often necessary to build an ad hoc detector, directly derived from the training sample set. A single layer of salience, like in the previous paragraph, is often not sufficient due to the difficulty of the task; instead, a cascade is build where each layer gradually filter out more and more negative samples. Somehow, this strategy can be simpler to implement because generally, the same learning algorithm is used to learn every layer, i.e. from the salience level (first layer) to the ultimate layer.

The usual strategy to build the cascade has been already described in Subsection 2. Historically, AdaBoost [Freund and Schapire, 1995] have been a preferred classifier in this context. AdaBoost is a meta algorithm for machine learning that builds a strong classifier from a linear combination of weak learners, their exact implementation being left to the user. One reason for the success of AdaBoost lies in the simplicity of training, efficiency and rapidity that can be achieved through boosting. Another reason is more related to this chapter and is that the boosting process can be viewed as a feature selection process [Torralba et al., 2007]. As a result, boosting is perfectly suited for salience issues since it produces a compact classifier that is fast to evaluate and that needs a small number of features to take a decision. The constraint in this case is to use *stumps* as weak learners. In the literature [Fleuret and Geman, 2008, Torralba et al., 2007], a decision stump is a degenerate decision tree with a single node. Thus it is an overly simplified learner which depends on a single feature, of the form

$$h_m(\mathbf{x}) = \begin{cases} a & \text{if } \mathbf{x}_f < \theta \\ b & \text{otherwise} \end{cases}$$

where $\mathbf{x}_f$ denotes the $f$'th component (dimension) of the feature vector $\mathbf{x}$. Since the algorithm of AdaBoost iteratively picks the weak learner owning the minimal classification error with respect to the sample weights (the weights of misclassified samples increase throughout the iterations), the final strong classifier results in fact in a selection of the most discriminant features. To give a short example of the interest of using boosting to build cascade, a time ratio of several order of magnitudes was reached by Liu et al. [Liu et al., 2005] with respect to a Support Vector Machine (SVM): they have obtained a comparable accuracy from boosting with an extremely small training model (2000 times smaller than the SVM model) and with a very high speed in classification (54 times faster than a SVM). This is due to the fact that a SVM uses every available features, building extraordinary complex decision surfaces, while boosting intrinsically minimizes the number of regression stumps used. Another famous example that used AdaBoost is the face detector of Viola et al. [Viola and Jones, 2001]. The classifier of the first cascade layer takes its decision according to a single Haar feature, capitalizing on the fact that the eye area is often darker than the cheeks. Recently, some cascades have been developed using SVMs as subclassifiers. The trick in this case relies on selecting kernels of gradually increasing complexity across the layers [Harzallah et al., 2009, Vedaldi et al., 2009]: a fast linear kernel for the first layer, a quasi-linear kernel for the second layer and a non-linear kernel in the third layer. Other examples of ad hoc salience detectors include more elaborate face detectors [Huang et al., 2005] or the template based detector of Felzenszwalb et al. [Felzenszwalb et al., 2010]. In this last example, the salience part is defined as a template matching of a low-resolution version of the root model part. Once the root has been found in the image, the other parts are searched around in a greedy fashion. When too many other parts have not been successfully found, the search stops. The authors have reached similar level of state-of-the-art performances than with a full template search achieving a speed-up factor of 10 to 20 times faster.

## *The Feature Combination Problem*

Now that we have detailed how salience is practically used in object recognition to reduce and optimize the search, we now explicitely extend to the multi-feature case (i.e. when more than one type, or channel, of features is available) where computational issues are even more justified and show the numerous benefits that salience can bring in this case.

The combination, or fusion, of different types of features in a same approach is usually a non-trivial matter in computer vision and especially for object recognition. It often raises several well-known issues, such as the normalization problem, the increase of computational complexity due to feature extractions and the inherent difficulties to combine heterogeneous types of features: global/dense features (e.g. textures) with local/sparse features (e.g. keypoints). In contrast, systems that are using salience, by their cascaded nature, can easily bypass all those problems at once.

**Normalization issues.** Different types of features involve different ranges of values and it gets generally bothersome when such heterogeneous values are gathered in a same feature vector. In the literature, normalization is generally achieved by assuming that each component of the global feature vector follows a Gaussian distribution (meaning, subtracting the mean and dividing by the standard deviation) or a chi-square distribution in the case of histograms [Vedaldi et al., 2009]. In practice however, such hypotheses are not always realistic. Recent works on Multiple Kernel Learning (MKL) have contributed to partially solve some of these issues (combining heterogeneous types by using a linear combination of dedicated kernels), but the results can be still disappointing compared to a simple averaging for instance [Gehler and Nowozin, 2009, Vedaldi et al., 2009].

A first benefit in a salience-oriented framework is that in a cascade, the different subclassifiers $\{f_i\}$ use different subsets $\varphi_i$ of the whole feature set $\varphi$: $f_i : \varphi_i \to \{0, 1\}$. Let's first assume that each $T_i$ only contains features picked out from a particular type (e.g. a texture or an edge descriptor). In this case each decision function combines comparable features, which shrugs off most of the problems: there are no need for normalization and no combination of heterogeneous types. We illustrate such a method in section 3 in which heterogeneous types (namely, dense textures, sparse keypoints and semi-sparse edges) are used alternately in the subclassifiers. Nonetheless, this strong constraint (one feature type per subclassifier) does not always fit with reality but note that in the particular case of using AdaBoost to build the cascade, normalization is solved thanks to the way in which subclassifiers are built using weak learners (i.e. stumps, see §2) intrinsically acting as single feature normalizers.

**Computational issues.** Feature extraction is a time-consuming process which can even become a bottleneck in a standard object detection application. For instance, Vedaldi et al. [Vedaldi et al., 2009] have evaluated that, in the perspective of a classical approach (see Figure 1.(a)), just computing the input for all possible windows is prohibitively slow. On the contrary, a salience-oriented framework offers efficient ways to reduce the computational burden. As the decisions are taken temporally (i.e. one after the others), it becomes possible to prune every unnecessary feature extraction work. Ideally, a cascaded system is expected to extract the features at run-time, i.e., just before they are required for evaluation by a subclassifier (see Figure 1.(b)). This way, only a few spots in the image get closely examined, saving important amounts of computational power as demonstrated by [Felzenszwalb et al., 2010] for instance.

In particular, it can be interesting to limit the number of feature types used in the first cascade layers (i.e. the part of the cascade which is evaluated the most frequently). Since feature types are generally independent, each type requires its own machinery to be extracted from the image. By retaining a subset or even single feature type to feed the subclassifier of the first layer, the time spent to extract all the other types will be saved. Such a strategy was used independently by Harzallah et al. [Harzallah et al., 2009] and Vedaldi et al. [Vedaldi et al., 2009]. In the first case, only the feature type the the least expensive to compute (namely, histogram of oriented gradient optimized through integral images) was used for the initial cascade layer, without significant loss of performance compared to using all types. In the second case, jumping window technique relying again on a single type (namely SIFT keypoints) was used to generate candidate windows sent to the second cascade layer. Finally, other techniques like dynamic programming can be combined with cascade as done in [Felzenszwalb et al., 2010] (here, indexing response maps in order not to recompute them later) to further reduce the computational burden.

## 3 A Practical Example for Specific Object Recognition

We illustrate in this section the principles stated above in the form of a practical application of object detection. The objective here is to recognize a single model object (for simplicity and clarity) in images. Although well-known systems like Lowe's [Lowe, 2004] already exists for that purpose, it remains challenging because of the drop in performances of generic detectors in difficult conditions: either in the case of noisy images, either because of poorly textured model objects (see the experiments in section 3).

To solve these issues, we develop a salience-oriented framework that takes advantage of different feature types, some of which being sparse, some being dense. In the following, we begin by presenting mainstream state-of-the-art works about specific detection to highlight the differences with our method (§3). Then, we briefly present the different feature types used in our framework and their indexing in §3. Afterward, we describe the construction of the cascade (here, a lattice) and the detection mechanism (§3). Finally, we quantitatively demonstrate the quality of this approach both in terms of detection performances and recognition time in Subsection 3. More specifically, we show that standard state-of-the-art method are outperformed by this cascade and multi-features detector.

## *Previous Works*

Specific object recognition has been a challenging problem since at least 30 years. Recently, the apparition of *interest points* (or *keypoints*) as discussed in §2 has enabled a considerable simplification of the problem. Indeed, the search of a specific object immediately translates to the problem of finding a geometrically consistent correspondence between two sets of keypoints: one set originating from the model image, and the other one from the scene image. Since each keypoint is supported by a descriptor (most often expressed in terms of histograms of oriented gradient measured in the surrounding of the point) specific enough to evacuate most pairwise match ambiguities, the problem becomes tractable. In addition, the generic nature of keypoints allows detecting a large variety of model objects (e.g. books, toys, bottles, etc.), whereas in the case of class object recognition template features specific to the model class have typically to be learned during training [Vidal-Naquet and Ullman, 2003, Epshtein and Ullman, 2007].

In short, the usual strategy in the literature is to decompose the model and test images into a discrete set of invariant (salient) keypoints, each of them depicted by a local descriptor. Then, the matching is performed in two steps: (a) descriptors are compared independently in a pairwise fashion in order to elect a set a credible matches, and (b) geometric constraints are used to reject inconsistent assortments of points. For this last step, either RANSAC [Fischler and Bolles, 1981] or Hough/voting strategies [Lowe, 2004] can be used to retrieve the position of the object in the scene despite important clutter and/or viewpoint change. The problem in this scheme lies in the features: since salience is a compulsory prerequisite for all the remaining processing, any problem affecting their extraction can prevent the recognition from working properly. Typically, keypoints repeatability (i.e. robustness) is not so good when it comes to noisy conditions or moderate 3D viewpoint changes (more than 25-30° according to [Moreels and Perona, 2007]). Different papers have studied how to improve the probabilistic model used in step (b) like [Moreels and Perona, 2005], but to our knowledge none of them tackles the feature reliability problem.

A solution to this issue would then be to integrate different types of image features (including dense features) in the same framework, but this is generally difficult due to the heterogeneity of the feature types and to the aberration of using a slow feature type with respect to real-time constraint generally required in such applications. We show in the proposed approach how to combine with a cascade model salient and non-salient features without reducing the detection time while in the same time significantly increasing the performances on a noisy dataset. The scheme we adopt is to represent the model image as a graph of local features and then applying ideas developed by Messmer and Bunke [Messmer and Bunke, 1998] for matching subgraphs and Viola and Jones [Viola and Jones, 2001] for the cascade process. In the end, our specific object recognition system can deal with noise and occlusion while still being minimalist in terms of computations.

More generally, our purpose is to illustrate that both specific or class object recognition categories should borrow ideas and techniques from the other category. While it was not possible in the past (dense features for class object recognition, discrete salient features for single object recognition), cascaded approaches made this feasible by decomposing the process into several step where different feature types can be used. More generally, our purpose is to illustrate that both specific or class object recognition categories should borrow ideas and techniques from the other category. While it was not possible in the past (dense features for class object recognition, discrete salient features for single object recognition), cascaded approaches made this feasible by decomposing the process into several step where different feature types can be used (see §2).

## *Used Features*

We now lean onto the three different types of features used in our recognition system:

- Keypoints, denoted by $\varphi_K$.
- Edges, denoted by $\varphi_E$.
- Textures, denoted by $\varphi_T$.

For each type $\varphi_i \in \varphi$ with $\varphi = \{\varphi_K, \varphi_E, \varphi_T\}$, we outline its properties below and define a pairwise distance between local features. This distance is referred to as *local kernel* $K : \varphi_i \times \varphi_i \rightarrow \mathbb{R}$ since it takes into account both the positions $\mathbf{p} = \{x, y, \sigma, \theta\}$ of the two features (respectively, their center, scale and orientation) and their visual descriptors $\mathbf{z}$, in contrast with standard kernels as in MKL which act at a global scale. This distance is used in the recognition process to evaluate local similarities between the model and the scene image in the cascade subclassifiers.

**Keypoints.** We used SIFT keypoints since it has been shown to be well suited for single object recognition [Lowe, 2004]. SIFT keypoints are points in the oriented scale-space of the image, the invariant property used here is the extrema in the difference of Gaussian space. This property especially fits textured objects as strong textures provide the most stable keypoints under usual transformations (rotation, scaling, translation, etc.). The descriptor used for the keypoints is also the SIFT descriptor: it consists of a local patch around the interest point described in terms of oriented gradient histogram (thus handling invariance to illumination).

In our framework, the SIFT detector serves as a salience detector in the image. Namely, only regions where a SIFT keypoint was found are further analyzed, of course at the extra-condition that the keypoint descriptor matches a model one. Each image keypoint $\phi_k \in \varphi_K$ is formally defined by a center $\mathbf{c}_k = (x, y)$, a radial vector $\mathbf{h}_k = (\sigma cos\,\theta, \sigma sin\,\theta)$ and a descriptor $\mathbf{z}_k$ of 128 dimensions. The local kernel between two keypoints $\phi_k$, $\phi_l$ is defined as:

$$K_K(\phi_k, \phi_l) \quad = \quad \begin{cases} \|\mathbf{c}_l - \mathbf{c}_k\|^2 + \alpha_K^2 \|\mathbf{h}_l - \mathbf{h}_k\|^2 & \text{if } \|\mathbf{z}_l - \mathbf{z}_k\| \leq \eta_K, \\ \infty & \text{otherwise.} \end{cases}$$

where $\eta_K$ is a constant threshold that defines the acceptable amount of noise to match two keypoints (see §3). Since the system will need in the following to quickly compare a given (model) keypoint at a given location versus all keypoints present in the scene image, we index the scene keypoints in a k-d tree. Contrary to [Lowe, 2004], this indexing is based on the position information $\mathbf{c}$ and $\mathbf{h}$ instead of the descriptor.

**Edges.** We used the Canny edge detector [Canny, 1986] followed by a step of polygonization to obtain a bunch of line segments. A segment $\phi_e \in \varphi_E$ is only defined by its center $\mathbf{c}_e$ and its radial vector $\mathbf{h}_e$ such that the boundaries of the segments are $\mathbf{c}_e + \mathbf{h}_e$ and $\mathbf{c}_e - \mathbf{h}_e$. The local kernel $K_E$ between two edges $\phi_e$ and $\phi_f$ is the maximum of the minimum distance between each pair of pixels lying on both edges:

$$K_E(\phi_e, \phi_f) = \begin{cases} \max_{p\in[-1,1]} \min_{q\in[-1,1]} \|(\mathbf{c}_e + p\mathbf{h}_e) - (\mathbf{c}_f + q\mathbf{h}_f)\| & \text{if } |\theta_f - \theta_e| \leq \eta_E, \\ \infty & \text{otherwise.} \end{cases}$$

(since no visual descriptor comes with a line, we simply check the orientation). Again, we used 6 distance maps (one for each orientation, so that $\eta_E = 30°$ actually) to reduce the search time of a given line segment against all existing segments in the scene image. This technique is robust to a noisy segmentation since the distance does not vary if the existing line undergoes cuts or oversize. When the algorithm needs to retrieve the closest segment in the image corresponding to the request, we back-project the query onto the closest image edges. Thanks to this operation, the set of possible retrieved segments is infinite. One can thus think of the edge feature type as a flexible feature in the sense that it can adapt to the current search (i.e. this behavior is clearly impossible to implement in a classical graph matching application where the set of feature is finite and well defined before proceeding to the matching). Therefore we call edges semi-salient features in our framework.

**Textures.** We derived a new texture descriptor from the work of Tola and Lepetit about the DAISY feature [Tola et al., 2008]. Since textures are dense features, they exist for every pixel of the image scale-space. In our case, the descriptor $\mathbf{z}_u$ of a texture feature $\phi_u \in \varphi_T$ is defined as the concatenation of three subdescriptors extracted at the same location $\mathbf{c}_u$ but at different scales $\sigma_u/1.54$, $\sigma_u$ and $1.26\sigma_u$ (according to a similar construction in [Kruizinga and Petkov, 1999]). Each subdescriptor is an 8-bins histogram of the gradient extracted at the corresponding position, scale and orientation (see [Tola et al., 2008] for details). The local kernel is simply defined as the Euclidean distance between the two descriptors, provided the two locations are not too far away in the scale-space:

$$K_T(\phi_u, \phi_v) = \begin{cases} \|\mathbf{z}_u - \mathbf{z}_v\| & \text{if } \|\mathbf{c}_u - \mathbf{c}_v\|^2 + \alpha_T^2 \|\mathbf{h}_u - \mathbf{h}_v\|^2 \le \eta_T, \\ \infty & \text{otherwise.} \end{cases}$$

As in the original paper, we precomputed eight gradient maps (one for each orientation) at the finest scale and spanned the rest of the scale-space with a pyramid of Gaussian. In our case, we found it faster than extracting features at run-time in an independent fashion (the pre-computation step takes less than 60 ms for a 720x480 image).

Finally, we introduce here the notation $\min_I K_t(\phi) \equiv \min_{\psi \in \varphi_t} K_t(\phi, \psi)$ to mean that a given "request" feature $\phi$ is searched against all possible image features of the same type $t$, hence the notation of the minimum of $K_t$ on the whole image $I$.

### *Description of the Algorithm*

We used a cascade-based detector constructed in the form of an incomplete lattice. A lattice $\mathcal{L} = \{N, E, X, Y\}$ is a structure comparable to a tree at the difference that two different paths starting from the root can meet up later (but still excluding cycles, since edges are oriented). Here, $N$ denotes

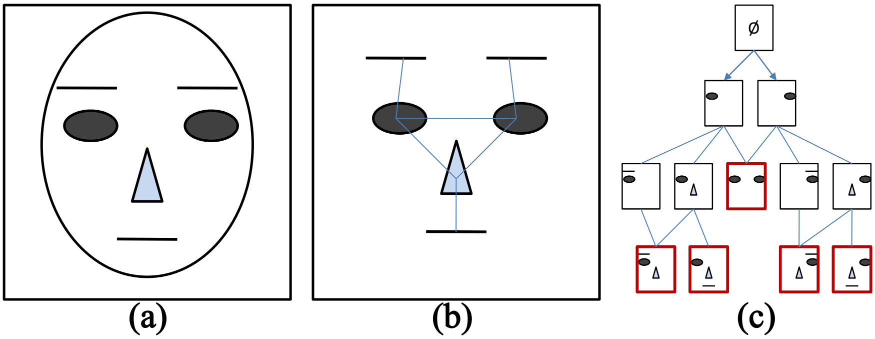


**Fig. 2** (a) Idealized model images. (b) Lines, triangles and ellipses represents different types of local features, connected if they are close enough. (c) Example of a lattice built with this idealized model.

the set of nodes, $X$ their associated content (see below), $E \subset N \times N$ the set of oriented edges and $Y$ their content. A toy example of such an incomplete lattice is presented in Figure 2.(c).

In this perspective, each lattice node $n \in N$ represents a collection of local model *parts*, where we mean by a part a single local feature: namely, a keypoint, an edge or a texture as defined in Subsection 3. Formally, $x_n \subseteq \varphi$ where $x_n \in X$ denotes the content of node $n$. Then, each lattice edge $e = (n \rightarrow m) \in E$ corresponds to the addition of a single feature $\phi_e \in \varphi$ to the starting node (hence $x_m = x_n \cup \phi_e$) at a given location $\mathbf{p}_e$, defined relatively to the locations of the features already present in that node:

$$y_e = (\phi_e = (\mathbf{p}_e, \mathbf{z}_{\phi_e}),\ f_e) \in Y.$$

As in usual cascades, a subclassifier $f_e$ is associated with this addition in order to evaluate its (binary) outcome during the detection stage (assuming that the collection of parts in node $n$ has already been successfully detected). It specifically consists of taking a decision upon whether or not the new part $\phi_e$ can be found in the scene image at a position relative to the set of already detected parts. In the case of a negative answer, the current path is abandoned, otherwise the end node $m$ is reached and the algorithm reiterates the procedure with the edges starting from that node. When a leaf node is reached, a vote is cast for the corresponding extrapolated model position in the scene image. The final step is a classical clustering of the votes.

To detect an object in the image at a given pose, one must feed the lattice with an initial local feature. In our case, we used the SIFT keypoints as starting points for the algorithm (which thus imposes that the features added between the root and the first lattice layer are necessarily keypoints). To completely scan the space of model poses, the procedure is iterated for every scene keypoint. The pseudo-code for the detection algorithm is listed in Algorithm 1. In our framework, the subclassifiers $f_i : \mathbb{R} \rightarrow \{0, 1\}$ are simplified to the uttermost since they only take as input a single real parameter returned by the local kernel of the type corresponding to the feature $\phi_e$ being tested. Of course, the position of the searched feature is adjusted in the scene image so as to be equivalent relatively to the positions of the already detected parts $x'_n$. By a slight abuse of notation, we denoted by $\{\mathbf{p}'_n\}$ the positions of the features present in $x_n$'. To compute this alignment, we simply used a 2D similarity transform.

**Construction of the lattice.** The lattice is constructed in a greedy fashion. Nodes are added to the lattice until a sufficient detection rate is achieved.

To begin with, the first lattice layer is built using the most reliable model keypoints. This information is estimated from a few model images (in our experiments, no more than 4) in which the model object is shot in various lighting conditions. The purpose of this strategy is to enable a nearly null

**Algorithm 1** Pseudo-code for the detection lattice.

```
Require: input image $I$
Require: keypoint set $\varphi_K$ of the image $I$
Require: detection lattice $\mathcal{L}$
Output: List of votes $V$
MAIN:
  $V := \emptyset$
  $n_0$ :=root node of $\mathcal{L}$
  for every $k \in \varphi_K$:
    for every edge $e = (n_0 \to m)$:
      if $\|\mathbf{z}_k - \mathbf{z}_{\phi_e}\| < \eta'_K$ then
        $V := V \cup$ RUNLATTICE$(m, \{k\}, I)$
      end if
    end for
  end for
  return $V$

RUNLATTICE$(n, x_u, I)$:
  // $n$ is a model lattice node containing model parts $x_n$
  // $x_u$ is the corresponding set of parts detected in the scene image
  if $n$ is a leaf node then
    $\mathbf{p}_{model}$ := extrapolated model position from $\mathbf{p}_u$
    return $\mathbf{p}_{model}$
  end if
  $V := \emptyset$
  for every edge $e = (n \to m)$:
    $\mathbf{p}'_e$ := relative position to $\{\mathbf{p}_u\}$ with respect to $\mathbf{p}_e$
    $\phi'_e := (\mathbf{p}'_e, \mathbf{z}_{\phi_e})$ // update feature position
    if $f_e(\min_I K(\phi'_e))$ then
      $\phi_{new} := \arg\min_I K(\phi'_e)$
      $V := V \cup$ RUNLATTICE$(m, u \cup \phi_{new}, I)$
    end if
  end for
  return $V$
```

false negative rate in the first lattice layer, while still discriminating a large part of clutter.

Then, subsequent lattice layers are built using a greedy algorithm similar to the one of [Vidal-Naquet and Ullman, 2003]: the layers are added one after the others. Specifically, a large set of candidate nodes is proposed for the new layer (the one being added). Each of those nodes consists of an upgrade of an existing node in the previous layer, i.e. a new feature is added to this node with the associated edge. The new feature has a random type and is randomly sampled in the vicinity of the other node features. This last condition comes from the fact that we want to build compact aggregates of features both to enable robustness to occlusion and to accelerate the training. Then, we evaluate each of those candidate nodes using Shannon's theory as did [Vidal-Naquet and Ullman, 2003]. This enables both to select the optimal threshold for the subclassifiers and to evaluate the joint efficiency of the nodes thanks to the addition in mutual information. The best candidates are kept, the others are eliminated. As in usual cascade, we use bootstrapping for learning: only true and false positives that reach the candidate nodes are used for learning. True positives are found on model images while false negatives are collected by launching the detection on background images. When a lattice node generates a sufficiently small proportion of false positives (typically, less than 5%), then its growth is stopped and it becomes a leaf node.

All in all, the lattice is robust to occlusion thanks to the variety of leaf nodes available for detection, each of them taking care of a reduced area of the whole model. Low run-time complexity is also achieved because of the cascaded structure: indeed, most tests (i.e. evaluations of a local kernel) are pruned and thus never carried out during detection. In our framework, we considered each such tests to be equally costly independently of the kernel type, but we could have integrated that information as in [Gangaputra and Geman, 2006] to favor cheap tests.

### *Relationship with Others Cascaded Approaches*

Although our detection system relies on a cascade, there exists significant differences with the approaches presented in Section 2:

- Contrary to cascades for class object detection like the ones of [Harzallah et al., 2009, Vedaldi et al., 2009], the subclassifiers in our system do not become more complex as the layer increases. Each subclassifier remains focused on thresholding the result of a single local kernel whatever its layer. This is due to the nature of our lattice which originates from a model graph (see Section 3).
- For the same reason, our approach only deals with a subpart of the object's surface whereas this limitation does not exist in classical cascaded approach like [Viola and Jones, 2004]. The reason behind this choice is to enable robustness to occlusion. Interestingly, [Paletta and Fritz, 2008] have developed a similar approach but for a different reason. In their system, they learn saccadic patterns of features in order to reduce the amount of information to process, hence enabling an efficient detection.
- Our system used the mutual information theory in order to build the subclassifiers. We believe this choice to be uncommon, as most often boosting techniques (see [Viola and Jones, 2004, Huang et al., 2005, Fleuret and Geman, 2008]) or SVMs (see [Vedaldi et al., 2009, Harzallah et al., 2009]) are preferred. Those mainstream classifiers are well suited for dense features, but do not fit well our rotation and scale invariant system that uses sparse features.

## *Quantitative Evaluation*

We present in this section an evaluation of our method with respect to other existing techniques for specific object detection.

### Parameter Settings

**Kernel parameters.** A few parameters have to be fixed *a priori* to tune the degree of freedom of the kernels. Each parameter define a sort of bounding box on the local area where the on-line search takes place, so they can be used to balance a long search time against a good robustness.

The threshold $\eta_K$ was set to $\eta'_K = 0.5$ for the root subclassifiers (see Algorithm 1) and to $\eta_K = 0.7$ for the subsequent levels. In the first case, it corresponds to a virtual vocabulary of about 1000 visual words: indeed, the probability that two random descriptors matches $p(\|\mathbf{z}_1 - \mathbf{z}_0\| \leq \eta_K)$ is worth about 1/1000 according to our tests on natural images. This vocabulary size is a common setting in bag-of-feature systems [Csurka et al., 2004]. We relaxed this threshold for the subclassifiers in the subsequent layers to enable more robustness, which empirically resulted in improved performances. Also, we empirically set $\alpha_K^2 = 8$ to balance x/y position against scale and orientation. Since textures descriptors vary smoothly along the image pixels, they are robust to slight positioning errors so we considered a unique test as sufficient, i.e. we set $\alpha_T^2 = 1$ and $\eta_T = 0$. This intuition was confirmed by the experiments (see Subsection 3).

**Lattice parameters.** The lattice is parametrized by several different constants:

- $N_1$: the number of node in the first lattice layer. Since in this layer each node only contains a single model keypoint, at least one of those $N_1$ model keypoints must be present in the scene for the object to be detected. We tried different values in the range $[8, 24]$ and intermediate values of $N_1 = 16$ and 20 gave the best results (see Figure 4).
- $p_{min}$ : the stopping probability to assert that a node becomes terminal in the lattice (i.e. a leaf node). Formally, the precision, or positive prediction value, is empirically evaluated for each node during training based on positive and negative image sets. When the precision exceeds $p_{min}$, the node is not grown anymore (note that this occurs independently of the layer: a node may terminate sooner than an other one if the features that it contains are more model-specific). We tried different values and $p_{min} = 0.95$ appears to be an good choice.
- $N_{children}^{max}$: the maximum number of children per nodes was set to 4 without noticeable differences with neighboring values.

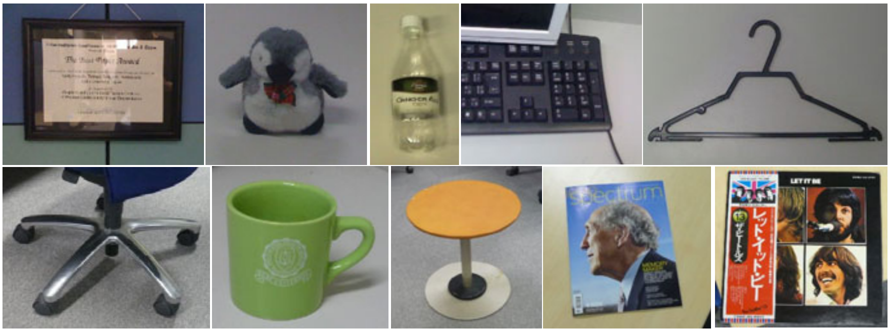


**Fig. 3** The ten model objects used in the experiments.

- $L^{max}$: the maximum width of the lattice. This value is important so that the lattice do not explode in term of node count and thus detection time. We empirically set $L^{max} = 4N_1$.

In our experiments, the parameters that appeared to really matters were $N_1$ in the first place and $p_{min}$ in the second place. The other ones did not change fundamentally the outcome of the experiments. As a consequence, we only varied $N_1$ in the experiments after having fixed the other parameters to their optimal value (Subsection 3).

**Experimental settings.** In order to highlight the increase of robustness due to the addition of semi-salient and non-salient types of features (i.e. edges, textures), we compare our approach to existing keypoint-based systems on a realistic dataset for robotic vision. The variety of represented noises includes a poor/garish luminosity due to the indoor lighting, various camera noises (captor noise, movement blur, video interlace). Moreovover, the model objects themselves are not always heavily textured, which makes the detection harder for keypoint-based systems.

The dataset consists of 2838 independant images where the ground truth (ten model objects) was manually labeled. The images come from videos manually shot with a standard SONY Handycam camera sampled at 10 fps. The model objects are visible in Figure 3 and were chosen so as to cover a large range of possible indoor objects: some are textured, some are not; some have complex 3D shapes; some are more prone to specular reflections; etc. Each object is visible in a few hundreds of frames and all objects were searched in the whole set of frames (multiple instances per frame are allowed). In order to conform ourselves to a realistic case of use, we only used 19 photos as negative training set (shared to train all models) and less than 4 positive images per model.

**Comparison with existing systems.** We compared our system against widely used specific object detection methods from the state-of-the-art:

- The baseline RANSAC [Fischler and Bolles, 1981] with a k-d tree matching scheme followed by an homography.
- The locally optimized RANSAC (LO-RANSAC) by Chum et al. [Chum et al., 2003] adapted by Philbin et al. [Philbin et al., 2007].
- The object recognition system by Lowe [Lowe, 2004].

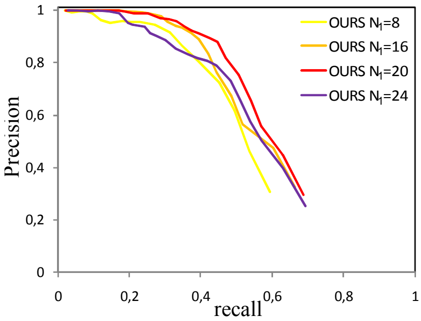


**Fig. 4** Global Recall-Precision curves on the robotic dataset for the proposed lattice with $N_1 \in [8, 24]$.

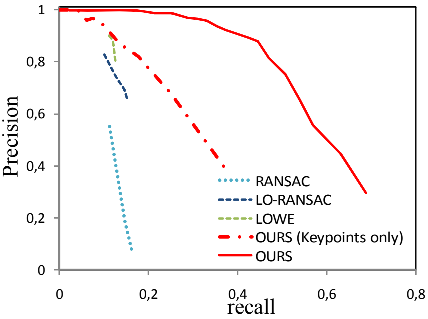


**Fig. 5** Comparison with existing systems in terms of recall-precision curves on the robotic dataset ($N_1 = 20$).

**Quantitative results.** Results are presented in Figure 4 and Figure 5 in terms of recall-precision curves. Recall and precision are defined as $N_c/N_g$ and $N_c/N_d$ respectively, where $N_c$ is the number of correct detections, $N_g$ the number of ground truth boxes and $N_d$ the total number of detections (the higher is the curve, the better). Some detection examples are shown in fig. 4.

To begin with, we investigated the influcence of the parameter $N_1$ which represents the number of nodes in the first lattice layer. Since the detection speed is roughly proportional to $N_1$, it is important to keep its value as small as possible. Interestingly enough, the curves presented in Figure 4 show that an intermediary value of $N_1 = 20$ yields the best detection performance. Practically, it implies that the detection can be efficiently initiated by a small number of features for each model (the most stable ones). Moreover, it justifies our choice for the algorithm design to rely on a generic salient detector as a virtual first cascade layer, since a nearly null false negative rate is indeed reached as shown by the extremly high values of recall (more than 70%, to relate to only 20% at most with existing detectors).

We also compared our lattice with existing specific object detectors and with the same method without using edges or textures. Globally, our system outperforms the keypoint-based methods for every value of $N_1$, including the smallest $N_1 = 8$ (only the curve with $N_1 = 20$ is drawn in Figure 5 for clarity). The contribution of semi- and nonsalient features (namely, edges and textures) is clearly important as shown by the difference of recall and precision between the proposed method with $N_1 = 20$ (curve "OURS" in Figure 5) and the same method without using these features (curve "OURS (keypoints only)" in Figure 5). Nonsalient features are indeed less disturbed by noise than sparse features. Note that our method already outperforms existing systems even without using extra features; this comes from the fact that (1) Lowe's method and LO-RANSAC requires respectively at least 3 and 4 correctly matched keypoints to assert a single detection, which is a rather difficult pre-requisite in noisy conditions, whereas we require only 2 keypoints; and (2) the proximity constraint that we used to build the lattice nodes helps to filter out more false positives, whereas the other methods are global and hence more sensitive to noise. To conclude with, we prove here that a cascade structure can enable more recall than non-cascaded systems if the null false negative rate constraint is effectively enforced without sacrifying detection performances at all.

**Detection speed.** The proposed method is also faster for the detection than the other methods: 54 ms/image on average for 10 objects ($N_1 = 20$), while Lowe's method and LO-RANSAC requires ˜70 ms and RANSAC about 224 ms. This result straightly follows from the cascaded lattice structure.

Our method however requires some additional pre-computations (done once per image) with respect to the edge and texture features (see §3). The times spent for these operations are respectively 155 ms and 60 ms for pre-computing the oriented edge maps and the pyramids of texture, respectively. We believe it to be small regarding the time required by the SIFT detector (1648 ms per image, on average, using the executable provided by Lowe [Lowe, 2004]) and the increase of recognition performance due to the extra feature types.

## 4 Conclusion and Perspectives

An extended vision of how salience can be used for task-specific object recognition was given in this chapter. Salience efficiently reduces the computational burden, which is currently a major worrying for most practical implementation, thanks to the cascaded shape resulting from its implementation. Salience also helps to combine different types of features together, again without proportionally increasing the computations since most work is pruned. In practice, it requires to select one of the two following techniques: either a generic salience detector, like a keypoint extractor for instance, which acts as the virtual initial layer of the cascade; or either an ad hoc detector in order to decompose the problem of detecting salient areas in many subclassifiers of gradually increasing complexity in the case of important intra-class variations. To sum-up, the way in which we take advantage of salience strongly depends on the task to accomplish, i.e., on the object or class of objects we want to recognize.

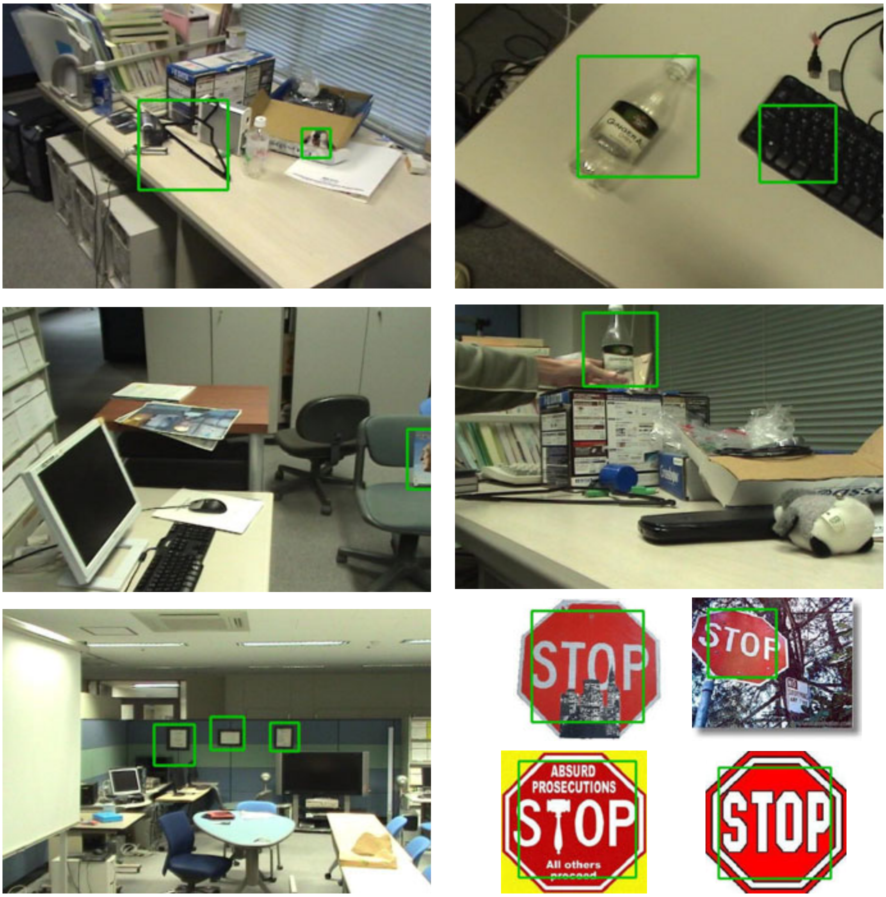


**Fig. 6** Examples of correct detections from the robotic dataset (2800 test images) and the Caltech-101 “stop sign” class [Fei-Fei et al., 2006] (only one training image was used in that case).

A practical application was presented to illustrate how to implement the principles discussed above for the case of specific object recognition. The proposed system outperforms in every points classical approaches from the state-of-the-art that do not take advantage of a cascaded structure: both the detection time and performances are better. Moreover, the cascade provides a straightforward way to combine heterogeneous types of features which was shown to result in a huge increase of detection performances.

Finally, a promising track seems to connect the cascade layers together to enable more robustness and flexibility, as in the recent work of [Felzenszwalb et al., 2010]. This way, the progression in the cascade does not depend on absolutely finding *every* model parts but on the contrary can afford some digression with the perfect model match.

## References

Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: Computer Vision and Pattern Recognition (2009)

Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)

Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: Computer Vision and Pattern Recognition (2007)

Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. Pattern Recognition, 236–243 (2003)

Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision (2004)

Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Schmid, C., Soatto, S., Tomasi, C. (eds.) International Conference on Computer Vision & Pattern Recognition, vol. 2, pp. 886–893 (2005); INRIA Rhône-Alpes,ZIRST-655, av. de l'Europe, Montbonnot-38334

Elad, M., Hel-Or, Y., Keshet, R.: Pattern detection using a maximal rejection classifier. Pattern Recognition Letters 23, 1459–1471 (2001)

Epshtein, B., Ullman, S.: Semantic hierarchies for recognizing objects and parts. In: Computer Vision and Pattern Recognition, CVPR 2007 (2007)

Eveland, C.K., Socolinsky, D.A., Priebe, C.E., Marchette, D.J.: A hierarchical methodology for class detection problems with skewed priors. J. Classif. 22(1), 17–48 (2005)

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results, `http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop`

Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. 28(4), 594 (2006)

Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: ComputerVision and Pattern Recognition (2010)

Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)

Fleuret, F., Geman, D.: Coarse-to-fine face detection. Int. J. Comput. Vision 41(1-2), 85–107 (2001)

Fleuret, F., Geman, D.: Stationary features and cat detection. Journal of Machine Learning Research 9, 2549–2578 (2008)

Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Proceedings of the Second European Conference on Computational Learning Theory, pp. 23–37. Springer, London (1995)

Gangaputra, S., Geman, D.: A design principle for coarse-to-fine classification. In: CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1877–1884. IEEE ComputerSociety, Washington, DC (2006)

Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Proceedings of the Twelfth IEEE International Conference on Computer Vision, ICCV (2009)

Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: Computer Vision and Pattern Recognition (2009)

Harris, C., Stephens, M.: A combined corner and edge detection. In: Proceedings of The Fourth Alvey Vision Conference, pp. 147–151 (1988)

Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: International Conference on Computer Vision (2009)

Huang, C., Ai, H., Li, Y., Lao, S.: Vector boosting for rotation invariant multi-view face detection. In: International Conference on Computer Vision (ICCV 2005), pp. 446–453. IEEE Computer Society, Los Alamitos (2005)

Huang, C., Ai, H., Wu, B., Lao, S.: Boosting nested cascade detector for multi-view face detection. In: ICPR 2004: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), vol. 2, pp. 415–418. IEEE Computer Society, Washington, DC (2004)

Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans.Pattern Anal. Mach. Intell. 20(11), 1254–1259 (1998)

Kruizinga, P., Petkov, N.: Nonlinear operator for oriented texture. IEEE Transactions on Image Processing 8(10), 1395–1407 (1999)

Lampert, C.: An efficient divide-and-conquer cascade for nonlinear object detection. In: Computer Vision and Pattern Recognition (2010)

Liu, X., Zhang, L., Li, M., Zhang, H., Wang, D.: Boosting image classification with lda-based feature combination for digital photograph management. Pattern Recognition 38(6), 887–901 (2005)

Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)

Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference, vol. 1, pp. 384–393 (2002)

Messmer, B.T., Bunke, H.: A new algorithm for errortolerant subgraph isomorphism detection. IEEE Trans. Pattern Anal. Mach. Intell. 20(5), 493–504 (1998)

Moreels, P., Perona, P.: Probabilistic coarse-to-fine object recognition. Technical report, California Institute of Technology (2005)

Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. International Journal of Computer Vision 73(3), 263–284 (2007)

Paletta, L., Fritz, G.: Reinforcement learning for decision making in sequential visual attention, pp. 293-306 (2008)

Paletta, L., Tsotsos, J.K.: Attention in Cognitive Systems. LNCS. Springer, Heidelberg (2008)

Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition, pp. 1–8 (2007)

Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using local affneinvariant image descriptors and multi-view spatial constraints. International Journal of Computer Vision 66(3), 231–259 (2006)

Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)

Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5), 854–869 (2007)

Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: International Conference on ComputerVision (2009)

Vidal-Naquet, M., Ullman, S.: Object recognition with informative features and linear classification. In: ICCV 2003: Proceedings of the Ninth IEEE International Conference on Computer Vision, p. 281. IEEE Computer Society, Washington, DC (2003)

Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, pp. 511–518 (2001)

Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)

Walther, D., Koch, C.: Modeling attention to salient proto-objects. Neural Networks 19(9), 1395–1407 (2006); Brain and Attention, Brain and Attention

# Chapter 5
# Fast and Efficient Local Features Detection for Building Recognition

G.P. Nguyen and H.J. Andersen

**Abstract.** The vast growth of image databases creates many challenges for computer vision applications, for instance image retrieval and object recognition. Large variation in imaging conditions such as illumination and geometrical properties (including scale, rotation, and viewpoint) gives rise to the need for invariant features; i.e. image features should have minimal differences under these conditions. Local image features in the form of key points are widely used because of their invariant properties. In this chapter, we analyze different issues relating to existing local feature detectors. Based on this analysis, we present a new approach for detecting and filtering local features. The proposed approach is tested in a real-life application which supports navigation in urban environments based on visual information. The study shows that our approach performs as well as existing methods but with a significantly lower number of features.

## 1 Introduction

The vast growth of images creates many challenges for computer vision applications in general, and image recognition in particular. Large variation in imaging conditions such as geometrical properties (including scale, orientation, and viewpoint) and illumination lead to very high demands being placed on the effectiveness of image features. Specifically, image features should be invariant (i.e. have minimal changes) under these divergent imaging conditions.

Early on in the development of image recognition, the use of global features was the most common approach. Global features are features capturing

G.P. Nguyen · H.J. Andersen
Department of Architecture, Design and Media Technology,
Aalborg University, Denmark
e-mail: {gnp,hja}@create.aau.dk

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 87–104.
springerlink.com

information from the whole image, and representing each image by a single feature vector such as color histograms. Global features are unable to distinguish foreground from background as this information is mixed together, which means their application is usually limited to cases with uniform backgrounds. In addition, image clutter and occlusions pose major problems in the use of global features. To overcome this limitation, local features were later developed, which are computed at different areas within a given image, for example points, edges, or image patches. Information extracted from these areas are called descriptors. An image is then represented by a set of local feature descriptors.

Local features have been applied successfully because of their stability under different imaging conditions [1, 2, 3, 4, 5, 6]. Commonly used local features include those extracted from interest points at different types of junctions, on contrast areas, or on texture areas [2]. Despite their wide utilization, however, some features of local descriptors have not been fully investigated.

Image recognition under outdoor conditions constitutes an extreme case in which many factors interfere with the appearance of the scene. For building recognition, most existing systems carry out the evaluation with images taken from different viewpoints and with different scales and orientations [7, 8, 9], but seasonal, daytime, and weather variations are generally not considered. For example, an image of a building in summer will be significantly more illuminated than the same image during winter. Moreover, seasonal changes also affect the appearance of the scene: in winter, buildings can be partly covered by snow; and at Christmas time and on other special occasions, buildings may be decorated. Other factors like shadow, shading, reflection, or scene occlusion caused by non-static elements such as people or cars may further interfere with the recognition process. Yet despite these obvious influences of temporal variation on recognition performance, such factors have not been fully analyzed. In this chapter, we address this challenge in an effort to contribute to the development of robust landmark detection systems.

Another major issue in approaches using local detectors is that they usually produce a large number of local features. On the one hand, this presents very rich information with which to analyze the image content, but on the other hand, it raises issues of computational processing time, storage, and performance efficiency. For example, SIFT detectors on average create $\sim$ 2000 features for an image of 500x500 pixels [3]. This number can increase significantly when the image contains many details. Since the computational cost of matching is positively correlated with the number of features extracted, solving this issue is essential.

Recently, more attention has been paid to recognition speed in application systems [10, 11] such as robot tracking [12, 13], and to real-time recognition with mounted devices [14]. To meet this speed requirement, different approaches for improving local detectors have been proposed. In [15], the authors present a method for reducing the dimensionality of SIFT descriptors using the PCA dimensional reduction method which projects the original

SIFT feature space from 128 dimensions to 20 dimensions. This PCA-SIFT method results in significant space benefits, and requires a third of the time in the matching phase compared to the original SIFT.

A different approach is put forward in [11], where a vocabulary tree is used to index features. The k-means algorithm is used to cluster all features and place them in the correct branch. For each query image, extracted features are traced down the tree, a score list is given for all leaves, and the one with the highest score is returned as the best match. This approach has proven to be very fast and scalable to a very large number of features.

Neither of the approaches mentioned above alter the original number of features; however, not all features are equally important. Some detected features, for example, are irrelevant in the recognition phase. In such cases, having too many features often reduces the recognition rate. For this reason, attention should be focused only on those features that are informative.

In summary, in this chapter we address the following two problems that can arise when using local feature descriptors for image recognition:

(i) The influence of significant temporal variation, especially in outdoor environments, such as different times of day, or from one day to another.

(ii) The provision of an efficient approach to deal with the large number of local features by selecting a subset of informative features that should meet two essential requirements. The first requirement is processing speed. The second is system performance; i.e. while reducing information from the original set, the system should perform as well as or even better than existing systems.

The chapter is organized as follows. Section 2 provides a short review of existing methods for detecting local feature points. This is followed in section 3 by our comparison of existing methods in the case of large temporal variation. We then present a new approach that provides an efficient solution to deal with the large number of local features in section 4. Subsequently, we setup a complete system for image recognition using a combination of the above techniques. And finally, to evaluate the proposed system, we implement a real-life application using mobile phones (section 5.2) for navigation based on visual information of nearby buildings.

## 2 Local Feature Detectors

A number of local descriptors are proposed in the literature, including several comprehensive reviews [1, 2]. Various authors compare a number of existing local descriptors, such as the scale-invariant feature transform (SIFT) [3], shape context [16], and moment invariants [17]. The evaluations demonstrate that the SIFT-based descriptors, such as the original SIFT features by Lowe [3] and PCA-SIFT by Ke and Sukthankar [15], outperform other transforms.

Therefore, these descriptors seem to be favorable choices for image recognition systems [18, 19, 9, 7].

SIFT's method for detecting local features consists of three main steps. Given an image set $\mathcal{I} = \{I_i(x, y)\}$: first, each input image $I_i(x, y)$ is represented at different scales, i.e. scale-space representation, using difference-of-Gaussian (DoG) function $D(x, y, \sigma)$

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I_i(x, y) \tag{1}$$

where $G(x, y, \sigma)$ is a variable scale Gaussian, and $*$ is the convolution operation in $x$ and $y$. $k$ is set to $2^{1/s}$, and $s$ is an integer number . Figure 1 shows an example of a scale-space representation. This step is used to identify potential key points that are invariant to scale and orientation.

Next, key points are extracted at each scale level using local maxima detection; i.e. by comparing a pixel to its neighbors in the scale-space. Points along edges are eliminated. After that, orientation is assigned to each key point using local image gradient directions. This is done by computing a histogram of local gradient directions, and selecting a peak corresponding to the dominant direction of the local gradient. Each key point now contains information about its location in the image, the scale where the key point is found, and orientation calculated from the previous step. Finally, descriptors are computed for each key point. A patch size of 16 × 16 sample is drawn, centered at the key point and having the orientation of that point. The patch is divided into 4 × 4 sub patches. Within each sub patch, an 8-dimensional orientation histogram is calculated using the same technique as described in the orientation assignment step. In total, this creates a 4 × 4 × 8 = 128-dimensional feature vector as a descriptor for a key point.

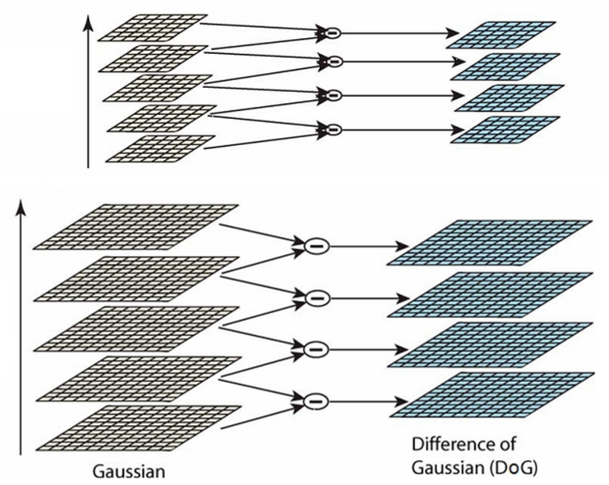


**Fig. 1** Application of DoG for scale-space representation.

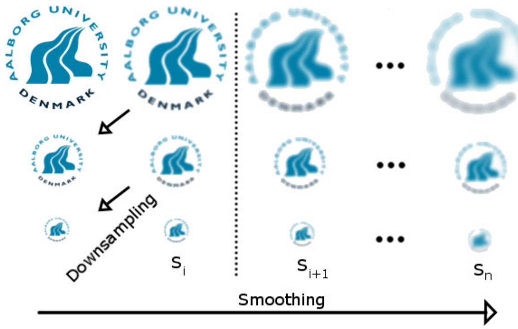
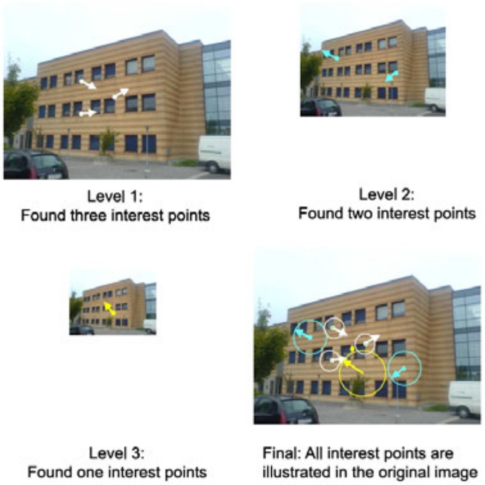


**Fig. 2** An image represented in a scale-space.

More recently, Brown et al. [20] have developed a new feature-based descriptor for matching and recognition, known as multi-scale oriented patches (MOPS). The authors discuss comparisons between SIFT and MOPS, and conclude that their recognition performances are comparable. The method can be briefly described as follows:

Each input image $I_i(x, y) \in \mathcal{I}$ is incrementally smoothed with a Gaussian kernel $\{\sigma_i\}_{i=1..n}$. An image pyramid is then constructed by down-sampling at rate $r$ of the image, as illustrated in figure 2.

In the second step, interest points are extracted using the Harris corners detector at each pyramid level (figure 3). This step returns a set of points located where the corner strength is a local maximum of a $3 \times 3$ neighborhood


Level 1:
Found three interest points

Level 2:
Found two interest points

Level 3:
Found one interest points

Final: All interest points are
illustrated in the original image

**Fig. 3** Examples of MOPS features. The size of the circles represent the pyramid level where the features are extracted, and the inner lines represent the orientations.

and above a threshold of 10. Third, a Taylor expansion is used to find the sub-pixel precision (up to the quadric term) at those extreme points. Each extreme point is described by its orientation in a window of size 28 × 28 (corresponding to a Gaussian kernel with $\sigma = 4.5$), and sampling of gray-level values in a $40 \times 40$ neighborhood, in a grid with a spacing of 5 pixels rotated according to the orientation. This gives a feature vector for each landmark consisting of 8 × 8 gray-level values. The feature vector is standardized by subtracting the mean and dividing by its standard deviation. Finally, a Haar wavelet transform is applied to the 8 × 8 descriptor patch to form a feature vector of 64-dimensions $F_{I_i(x,y)}$.

## 3 Local Feature Detectors under Large Temporal Variation

In this section, we concentrate on the influence of large temporal variation on local feature detectors. The two detectors described above provide input for our investigation.

As mentioned in section 1, image recognition in outdoor environments is an extreme case where the large temporal variation is common. To compare the performance of the two local feature detectors under large temporal variation, we apply them to a building recognition application. Figure 4 shows an example of images of the same building taken at different settings.

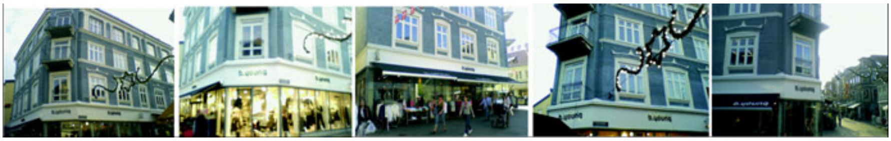


**Fig. 4** An example of large temporal variation, where a building is captured at different times of the day and season.

The test set consists of 442 images of 19 different buildings. Images of the buildings were captured by different people to achieve natural variation in viewpoint and style of capturing images. This brings the testing system much closer to a real application, where buildings are usually captured in different styles. To create large temporal variation in the dataset, the images were also taken at different times of the day and year, in varying weather conditions such as very cloudy or very sunny. Building decorations were taken into account as well, to ensure large variation in the buildings' appearance. The images were captured at the following times:

1. 16/11/2006: during daytime (a cloudy day).
2. 17/11/2006: in the evening (with electrical lights on).
3. 28/11/2006: during daytime (with Christmas decoration).

4. 29/11/2006: during daytime (with Christmas decoration).
5. 05/12/2006: in the evening (with Christmas decoration).
6. 03/05/2007: summer time (a sunny day), during daytime (many buildings decorated with Danish flags).

For the evaluation, features from all images were calculated and stored as off-line data. Then, each image was chosen from the database in sequence and used as a query image. Features of the selected query were compared to those of all other 441 images in the database. This features matching step was similar to that performed by Brown et al. [20]. The system returned a ranked list of matched results, and the top rank $m$ images, where $m = [1 \ldots 5]$, were then reported. Precision was computed as the top $m$ for each query image, and finally averaged over all 442 query images to obtain the average precision performance.

Figure 5 shows the average precision result with $m = [1 \ldots 5]$. For comparison, the MOPS approach is compared with the commonly used SIFT extractor. In the implementation, the default parameters for the SIFT extractor as indicated by Lowe [3] were used. Top ranked lists and the average precision over all queries were returned, and are also illustrated in figure 5.

Our experimental results show that the MOPS features adapt much better when it comes to recognizing buildings under large temporal variation as well as different viewpoints, scales, and orientations. On average, the system performance with MOPS is considerably improved over that with the SIFT features. In figure 6, we show the recognition results of two example queries of building recognition under large temporal variation using MOPS. In the first example, the query building is captured on a sunny summer day, where

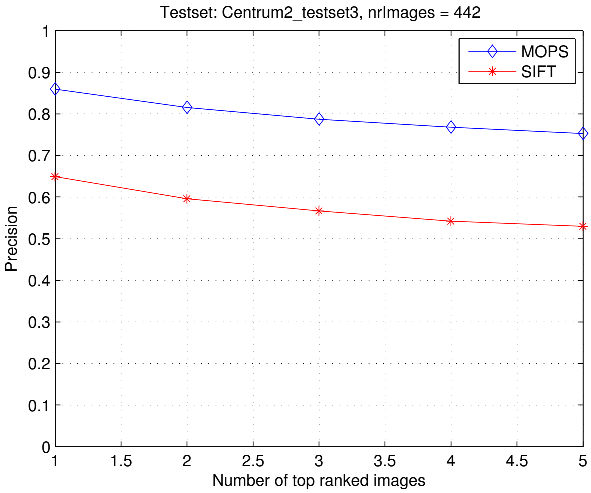
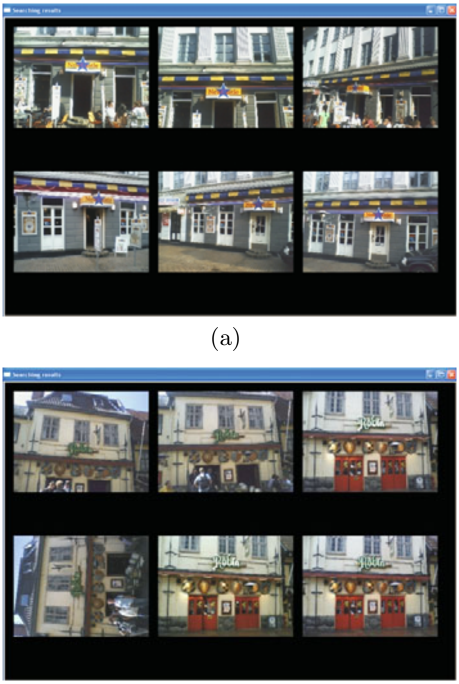


**Fig. 5** Precision vs. number of top ranked images. Results show performances using SIFT and MOPS as features detectors for building recognition.



(a)

(b)

**Fig. 6** Two examples using MOPS for recognition of buildings under large temporal variation. The top-left image is the query image. Next, from left-to-right and top-bottom is the ranked list of matched images.

the bar is open and many people are sitting in front of the building. The MOPS method is able to return images of the same building at winter time when the bar is closed, with no decoration, and very different illumination.

Using MOPS, we were able to achieve a high performance with 0.85 precision in the top match. We, therefore, chose MOPS as the basis for further investigation.

# 4 Selection of Informative Local Features

## *4.1 Uniqueness Filtering for Local Features*

When developing techniques for selecting features, it is generally assumed that certain features are more important than others. The terms "discriminative" and "informative" are usually used to describe those features. Li and Kosecka [21] observe that certain features are more stable and thus better able to handle variations in scale and viewpoint. It is these feature that they therefore aim to select. For each feature extracted by means of the SIFT detector from each image at each location, they calculate a posterior probability. The probability values are used as ranking criteria. Brown et at. [20] present an adaptive non-maximal suppression (ANMS) algorithm that selects a subset of interest points based on their corner strength. The general idea of this algorithm is that for each point extracted through the process described above, they calculate the corner strength, then select strongest points within a neighborhood of radius $k$ pixels. In all these experiments, the authors select a maximum of 500 points for each image, meaning that a set of 500 features is used to describe the content of an image. Another technique for selecting informative (i-SIFT) features using the SIFT detector is put forward by Fritz et al. [19]. In this approach, informative features are defined as those that appear in discriminative regions. These regions are detected on the basis of an entropy-coded image derived by calculating posterior distribution.

In this section, we propose a different method for defining the discriminative descriptors that identify the most salient features in a given image. One salient property is rarity, as defined by Kadir and Brady [22], and Schiele and Crowley [23], which identifies those discriminative descriptors that are almost unique, i.e. which maximize discrimination between objects. We thus propose to select descriptors on the basis of their uniqueness i.e. their rarity within a descriptor set.

**Definition:** *A unique feature has an identifiable property that distinguishes it from other features in the image.*

In other words, in a feature space where all descriptors are located, a unique feature is the one with the fewest features within its $\epsilon$-neighborhood. Starting from this definition, our method of selecting unique features is as follows.

Given an image $I$, assume that $I$ has $k$ descriptors or feature vectors $\mathcal{F}_I = \{F_1, F_2, \ldots, F_k\}$. To calculate the uniqueness of each feature, we first compute the dissimilarity values between feature vectors. For the MOPS descriptors, L2 distance is used as the dissimilarity function. We have $S_{ij} = \sqrt{\sum_{l=1}^{t}(f_l^i - f_l^j)^2}$, where $f_l^i$ and $f_l^j$ are components of feature vectors $F_i$ and $F_j$ respectively, and $t = 64$. Each feature vector $F_i$ is compared to the others $\{F_1, \ldots, F_{i-1}, F_{i+1}, \ldots, F_k\}$. We then obtain a set of dissimilarity values $\{S_{i1}, S_{i2}, \ldots, S_{i,i-1}, S_{i,i+1}, \ldots, S_{ik}\}$. To decide whether two feature vectors are similar or not, an $\epsilon$ neighborhood is established. If a feature point $F_j$ in the given feature space falls within the $\epsilon$-neighborhood of $F_i$, it is considered similar to $F_i$, i.e.

If $S_{ij} < \epsilon$ then $F_i$ and $F_j$ are similar,
otherwise $F_i$ and $F_j$ are dissimilar.

As indicated in our definition of a unique feature, we select features that have the smallest number of similar features within their $\epsilon$-neighborhood. This means that the greater the number of neighbors, the less unique the feature is. Hence, the uniqueness of a feature vector $F_i$ in image $I$ is formulated as:

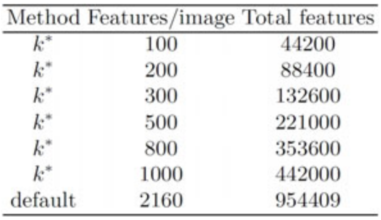
$$\mathcal{U}_{F_i} = \|\{S_{ij} < \epsilon\}\|_{j=1..k,j\neq i}$$

## 4.2 Evaluation

A number of experiments to evaluate the proposed technique for filtering local features were performed using the data-set as described in section 3. The MOPS detector was applied to all images with default parameters and all extracted features were stored. Next, the method for selecting the unique features of each image was applied. These unique features were stored separately. As the extraction of unique features from the images was carried out off-line, the processing time was not important. Instead, our focus was on comparing the performance with that achieved by other methods.

To evaluate performance during the matching process, precision values were reported. Each image in the data-set was sequentially used as a query. The query was then compared to all other images in the corresponding data-set. The top five best matches were returned, and the precision values calculated for each of the images. The baseline was the performance achieved by using the default MOPS detector on all extracted features. We also applied the adaptive non-maximal suppression (ANMS) selection method as described by Brown et al. [20], which selects the strongest features on the basis of corner strength. To ensure a fair comparison, experiments with different numbers of selected features were performed, namely 100, 200, 300, 500, 800, and 1000. In all experiments, the unique features were identified using $\epsilon = 0.3$.

First, we present the results from different cases in which varying numbers of descriptors were used, as shown in figure 7. $k^*$ represents our uniqueness filtering method. In this table, the last row is the default MOPS with all



| Method | Features/image | Total features |
|---|---|---|
| $k^*$ | 100 | 44200 |
| $k^*$ | 200 | 88400 |
| $k^*$ | 300 | 132600 |
| $k^*$ | 500 | 221000 |
| $k^*$ | 800 | 353600 |
| $k^*$ | 1000 | 442000 |
| default | 2160 | 954409 |

**Fig. 7** Experiments with different numbers of unique features per image vs. the default MOPS detector.

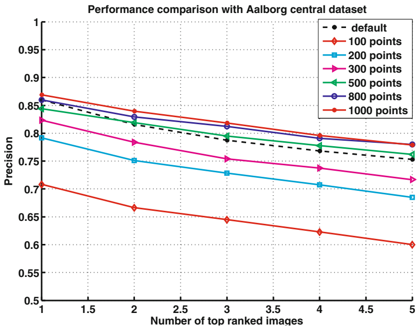
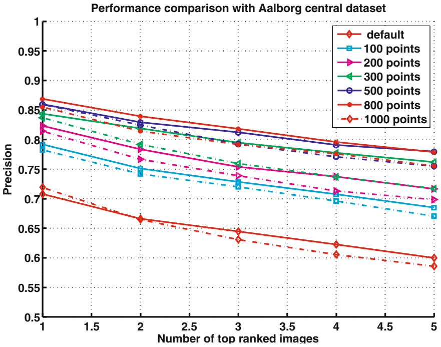


**Fig. 8** Precision vs. number of top ranked images. Results show the performance of the default MOPS with all extracted features vs. our approach focusing only on unique features.



(a)

**Fig. 9** Precision vs. number of top ranked images. Results show performance using our approach (full lines) vs. ANMS (dotted lines) to select descriptors.

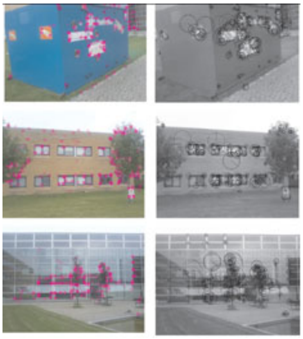


**Fig. 10** Examples of buildings with 500 selected descriptors using uniqueness criteria vs. ANMS.

features taken into account. The second column reports the average number of descriptors per image, while the third column reports the total descriptors of each data-set. They show that, on average, the number of features extracted in the default cases is significantly larger.

Figure 8 presents the recognition results, where we compare performance when different numbers of unique features are used, versus default performance when all extracted features are employed. The figure shows that although the default MOPS has the highest number of descriptors, it performs less accurately than our method, which uses smaller numbers of descriptors. The results achieved by using a certain number of unique features are significantly better. A selection of 300 features enables the reliable recognition of a given building, although even with fewer features we still obtained a relatively high recognition rate of 70% in the best match. With 500 unique features, i.e. fewer than $\sim \frac{1}{4}$ of the total descriptors, we achieved the same performance as the default approach. Further improvement is shown with 800 and 1000 unique features. We can also see from the figure that there is a saturation point between 800 and 1000 descriptors, at which there is little improvement in performance. This means more descriptors are unnecessary, and may even reduce performance by creating disturbance.

In the next experiment, we compared the performance achieved by our approach with that achieved when using ANMS to select descriptors. Figure 9 shows the result of the two approaches with different numbers of selected descriptors. The dotted lines represent the results for the ANMS approach, and the solid lines represent the results for our uniqueness method. With 100 or 200 selected descriptors, the performance of the two approaches was comparable. Where a higher number of descriptors was used, our proposed approach achieved better results.

In figure 10, we present some examples of buildings with 500 selected descriptors. All features are first extracted, then two different filtering approaches are used. The left column shows the results produced with the proposed approach, and the right column the results produced with ANMS. In general, the two methods select very similar descriptors. ANMS selects features on the basis of their corner strength, which means the features selected are mainly corners. In our case, we do not consider corners the most important features, as they will recur in other similar areas. The features selected should be those that are least like other features in the same image: i.e. where they are resembled by the smallest number of features in each image. Since corners are not our first priority in selection, other salient details are chosen instead.

The foregoing has demonstrated that the proposed approach for selecting informative features both reduces the number of local features extracted, and improves the recognition performance.

# 5 System and Application

## *5.1 System Scheme*

As our investigation has answered the two questions stated in section 1, we are now able to draw up a complete scheme for an image recognition system. Figure 11 shows an overview of the proposed scheme, made up of two stages. In the off-line stage, database images provide input for the feature extraction step using MOPS, after which uniqueness filtering is applied to select the informative features. In the on-line stage, the user provides a query image, and the system extracts and filters the features of that image. These query features are then compared to the off-line data and the system returns a list of matching images.

## *5.2 Application*

The ultimate test of our system is in a real-life application. For that purpose, we have developed a scenario where we provide user information to help individuals navigate in unfamiliar areas using visual information from the surrounding environment, i.e. nearby buildings. Assume that you have to go somewhere but you don't know how to get there. Traditionally, you would first have to locate yourself on a paper map, then spend time figuring out the best route. Integrated software to instruct users via mobile devices such as PDAs or mobile phones is now replacing this traditional approach. The currently available products in this field are supported by systems such as GoogleStreet for the geo-referencing of information. However, the occlusion

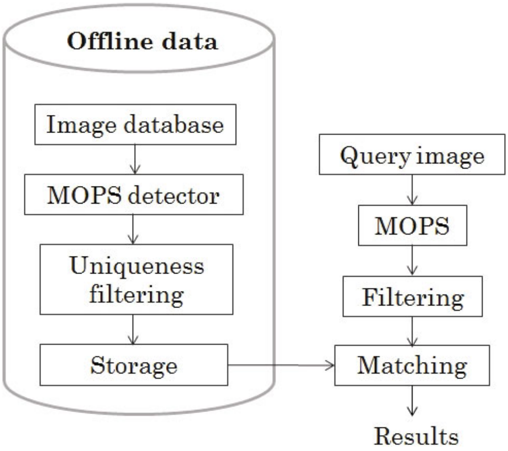


**Fig. 11** The scheme of our image recognition system.

of satellites and significant drifts especially apparent in urban environments serve to limit GPS service. Moreover, visual information is a rich source of information for localization and navigation that is in accordance with the way we as humans navigate, and may thus give rise to a versatile and robust method easily adapted by users. In our application, when the user needs a route description, he or she need only take a picture of a nearby building in order to retrieve feedback regarding the location as well as guiding instructions.

## *5.3 Scenario*

In this section, we describe our mobile phone application to help users navigate in urban environments. We set up a scenario in which users are placed in an unfamiliar area, and instructed to make their way from one location to another. Each user is provided with a mobile phone with a built-in camera. If a user needs navigation assistance, he or she will capture an image of a nearby building, and send it to the server for guidance. At the server, the system processes the image and searches through the database to find possible matches. Matching images and associated information are then sent back to the user. This process is illustrated in figure 12.

At the starting point, the user sees a welcome screen with different destinations. By selecting a certain image, the user chooses the destination and starts the application. Figure 13 shows the mobile phone interface of this step; information on the selected destination is displayed at the bottom of the screen. Then, the user sees an overview map with a route drawn from the starting point to the destination. In this way, the interface gives the user an overall view of the journey as well as the estimated walking time.

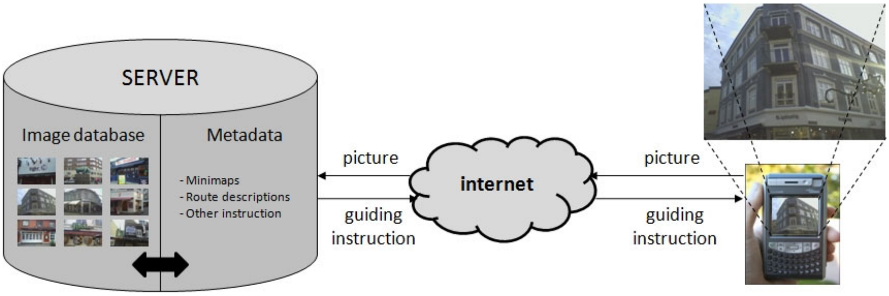
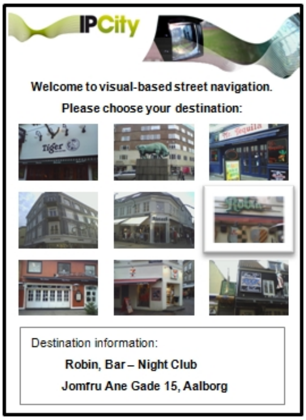
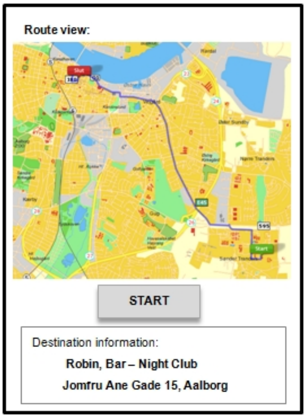


**Fig. 12** Overview of client-server system for visual-based navigation in urban environments.



Welcome to visual-based street navigation.
Please choose your destination:

Destination information:
Robin, Bar – Night Club
Jomfru Ane Gade 15, Aalborg



Route view:

START

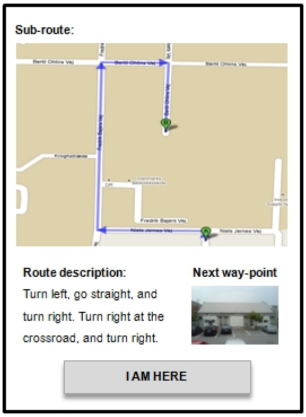
Destination information:
Robin, Bar – Night Club
Jomfru Ane Gade 15, Aalborg

**Fig. 13** Welcome screen

**Fig. 14** Overview route

Along the route, we mark a number of buildings as way-points depending on how far away the destination is. This number is determined before-hand and stored for each route. In our application, the distance between two way-points is approximately 300 meters. This means that the overview route is divided into several sub-routes, with the user navigating only on "sub-route level" at any given time. This makes it easier for the user to remember the map, especially when the destination is far away. Figure 15 shows an interface with the sub-route map, which we call a mini map. This interface shows the route from the current location to the nearest way-point, with a route description. It also displays a picture of the next way-point, so that the user knows what to look out for. When the user reaches to the next way-point, he or she simply clicks on the button "I AM HERE" to get further instructions.

To ensure that the user actually reaches the right way-point, we ask the user to take a picture of the building and send it to the server. The server then performs the building recognition process to find matching images. A

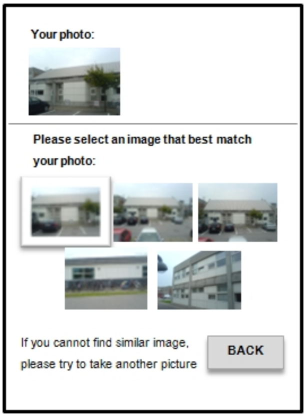

Sub-route:

Route description:

Turn left, go straight, and turn right. Turn right at the crossroad, and turn right.

Next way-point

**Fig. 15** Sub-route


Your photo:

Please select an image that best match your photo:

If you cannot find similar image, please try to take another picture

**Fig. 16** Retrieved information

list of the top 5 best matches is returned, and the user chooses the one that he or she thinks is the most similar (figure 16). If the system fails to find at least one correct match, the user is prompted to take another image, as the failure may be caused by the quality of the query image (e.g. if it is too blurred). If the user is now able to find an image from the matching list, the next sub-route interface is shown with a new mini map and route description. These steps are repeated until the user reaches the destination.

## *5.4 Evaluation*

To test the application, we simulated user actions. We set up a number of routes from one location to another in the test area, where images had been captured (section 3). Out of 19 buildings, we randomly created 50 different routes. Each route contained three way-points, i.e. simulated users had to pass through those way-points to reach the destination. This meant that for each route, we ran the recognition system at least four times. Figure 17 shows an example of a route from building 1 to building 14 which passes through 3 way-points, namely buildings 18, 13, and 15.

At each way-point, an image of that building was randomly selected from the query set. If there was a correct match, it was returned from the server and the simulation continued with the next way-point. If there was no correct match, another query image was randomly chosen, which simulated a user's action of taking a new picture of the building. We allowed this step to repeat a maximum of three times. If there was still no correct match after three tries, the simulation stopped and we reported it as a failed case. If the simulation reached the destination, we reported it as a successful case. Finally, we computed the percentage for these cases, which revealed that our experiment yielded a 99% success rate when allowed three trials per building.

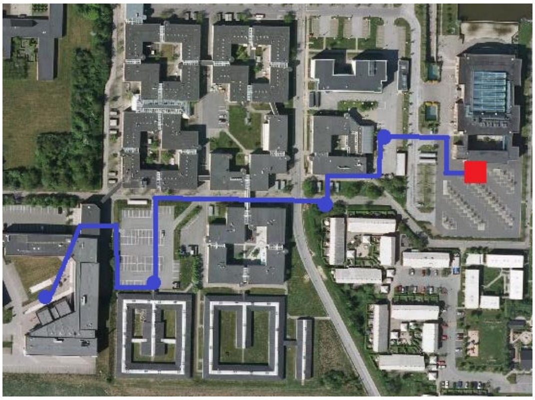


**Fig. 17** An example of a route passing through 3 way-points.

As a real-time application, time is also an essential factor. We recorded both the recognition processing time and the waiting time. The former is the time needed to complete the recognition process, i.e. the feature extraction and matching time. The latter refers to the total time the user has to wait after sending a query image to the server, and before receiving the information back. This means that after the user takes a picture of a building, a timer starts when he or she presses the "Get information" button, and finishes when all information is received from the server and displayed to the user.

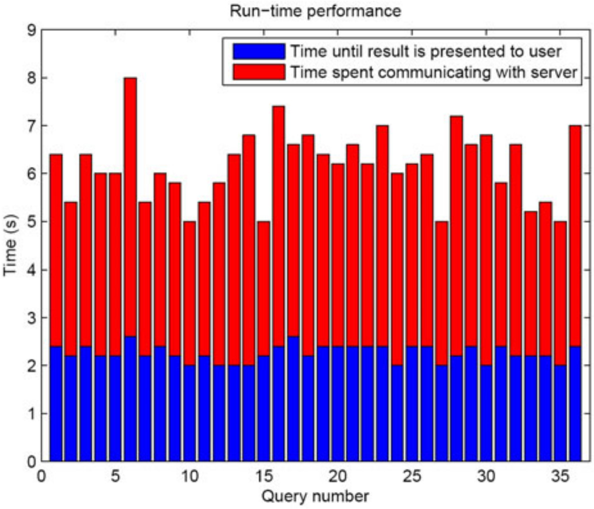


**Fig. 18** Run-time performance of the navigation system. The blue bar is the time taken to upload and download information to and from the server. The red bar is the time needed for feature extraction and matching.

The response time for each query image has been averaged, and figure 18 shows the total response time as a bar plot. The top bar is the time it took to upload the image to the server and receive a response, while the bottom bar is the time it took to display the result to the user. The two bars are stacked to indicate the total time, which is what the user would experience. On average, the recognition processing time was 3.9 seconds and the display time 2.3 seconds, giving a total time of 6.2 seconds.

## 6 Conclusion

In this chapter we have indicated how the local feature image recognition method MOPS may be filtered to a subset of the original local features while still maintaining performance. Filtering of features is of particular interest for applications running on resource-limited devices such as mobile phones. Our application aims to help users to navigate in urban environments by providing feedback with map and guidance directions when the user sends an image of a nearby building. The simulation and tests reported on here demonstrate the potential of the proposed application in a real-life navigation setting.

**Acknowledgements.** This research is supported by the IPCity project (FP-2004-IST-4-27571), a EU-funded Sixth Framework program Integrated Project on Interaction and Presence in Urban Environments.

## References

1. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis & Machine Intelligence 27(10), 1615–1630 (2005)
2. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2008)
3. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
4. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. International Journal of Computer Vision 87(3), 284–303 (2010)
5. Datta, R., Joshi, D., Li, J., Wang, J.: Image retrieval: ideas, influences, and trends of the new age. ACM Computing Surveys 40(2), 1–60 (2008)
6. Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. Information Retrieval 11(2), 77–107 (2008)
7. Khan, S., Rafi, F., Shah, M.: Where was the picture taken: Image localization in route panoramas using epipolar geometry. In: IEEE International Conference of Multimedia and Expo., pp. 249–252 (2006)
8. Rajashekhar, Chaudhuri, S., Namboodiri, V.: Retrieval of images of man-made structures based on projective invariance. Pattern Recognition 40(1), 296–308 (2007)
9. Zhang, W., Kosecka, J.: Hierarchical building recognition. Journal of Image and Vision Computing 25(5), 704–716 (2007)

10. Calonder, M., Lepetit, V., Mihelich, K., Bowman, J., Fua, P.: Compact signatures for high-speed interest point description and matching. In: International Conference on Computer Vision (September 2009)
11. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2161–2168 (2006)
12. Royer, E., Lhuillier, M., Dhome, M., Chateau, T.: Localization in urban environments: monocular vision compared to a differential gps sensor. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 114–121 (2005)
13. Kosecka, J., Li, F., Yang, X.: Global localization and relative positioning based on scale-invariant keypoints. Robotics and Autonomous Systems 52(1), 27–38 (2005)
14. Robertson, D., Cipolla, R.: An image based system for urban navigation. In: Proceedings of the British Machine Vision Conference (2004)
15. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 506–513 (2004)
16. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 2(4), 509–522 (2002)
17. Gool, L.V., Moons, T., Ungureanu, D.: Affine/ photometric invariants for planar intensity patterns. In: Proceedings of the 4th European Conference on Computer Vision, vol. 1, pp. 642–651 (1996)
18. Zhang, W., Kosecka, J.: Localization based on building recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 21 (2005)
19. Fritz, G., Seifert, C., Paletta, L.: Urban object recognition from informative local features. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 132–138 (2005)
20. Brown, M., Szeliski, R., Winder, S.: Multi-image matching using multi-scale oriented patches. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 510–517 (2005)
21. Li, F., Kosecka, J.: Probabilistic location recognition using reduced feature set. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3405–3410 (2006)
22. Kadir, T., Brady, M.: Saliency, scale and image description. International Journal of Computer Vision 2(45), 83–105 (2001)
23. Schiele, B., Crowley, J.: Probabilistic object recognition using multidimensional receptive field histograms. In: Proceedings of the International Conference on Pattern Recognition (1996)

# Chapter 6
# Visual Perception in Image Analysis[⋆]
## Digital Image Content via Tolerance Near Sets

James F. Peters

**Abstract.** This chapter considers how visual perception can be used to advantage in image analysis. The key to the solution to this problem was first pointed out by J.H. Poincaré in 1893 in his representation of the results of G.T. Fechner's 1860 psychophysics experiments with sensation sensitivity in lifting small weights. The focus of Fechner's experiments was on sensation sensitivity. By contrast, the focus of Poincaré rendition of Fechner's experiments was on determining sets of similar sensations that serve as a model for a physical continuum. In what he later called a representative space (*aka*, tolerance space), Poincar'e informally discerned tolerance relations in determining tolerance classes containing perceptually indistinguishable sensations. A formal view of tolerance spaces was first introduced by E.C. Zeeman in 1962 (nearly 70 years after Poincaré's work on representative spaces). Unlike Poincaré, Zeeman focused on visual acuity in formulating the idea of a tolerance space. By defining a tolerance relation, one provides a basis for a rigorous study of resemblance between perceptual objects such as digital images or observed

James F. Peters
Computational Intelligence Laboratory
University of Manitoba,
Department of Electrical & Computer Engineering,
75A Chancellor's Circle, EITC Room E1-526
Winnipeg, Manitoba, R3T 3E2, Canada
e-mail: `jfpeters@ee.umanitoba.ca`

[⋆] Many thanks to Som Naimpally, Sheela Ramanna, Andrzej Skowron, Jarek Stepaniuk, Amir H. Meghdadi, Christopher Henry, Homa Fashandi, Shabnam Shahfar, and Piotr Wasilewski for the suggestions and insights concerning topics in this paper. I also want to Christopher Henry for the work he has done in putting together the NEAR system, a complete implementation of near sets tailored to the needs of image analysis that is reported in this chapter. This research has been supported by the Natural Sciences & Engineering Research Council of Canada, *i.e.*, *Conseil de recherches en sciences naturelles et en génie du Canada* (NSERC) grant 185986, Manitoba Centre of Excellence Fund (MCEF) grant, and the Canadian Arthritis Network grant SRI-BIO-05.

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 105–125.
springerlink.com

behaviour patterns of collections of social robots. Eventually, the study of the resemblance of disjoint sets by Z. Pawlak and J.F. Peters, starting in 2002, led to the discovery of a formal basis for measuring the degree of nearness between distinct tolerance spaces. The main contribution of this paper is the introduction of a form of perceptual image analysis in terms of a methodology for determining the resemblance between pairs of visual tolerance spaces defined within the context of digital images.

**Keywords:** Image correspondence, metric space, nearness, near sets, perception, perceptual image analysis, resemblance, tolerance space.

## 1 Introduction

This chapter considers how visual perception can be used to solve image analysis problems such as discerning the extent of correspondence between images. The solution to the image correspondence utilizes image matching strategies to establish affinities between two or more images. This is one of the central tasks in photogrammetry and computer vision. Recently, it has been shown that tolerance near sets can be used in a perception-based approach to discovering correspondences between images (see, *e.g.*, [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]). Disjoint sets that resemble each other are called near sets [15]. The study of near sets is directly related to more recent work on similarity, tolerance, covering systems, and resemblance [16, 17, 18] and a tolerance space view of what we see [3].

1.1: Walking dog on stilts [1]

1.2: cover, p=5, $\varepsilon = 0.1$

1.3: cover, p=10, $\varepsilon = 0.1$

1.4: cover, p=15, $\varepsilon = 0.1$

1.5: cover, p=20, $\varepsilon = 0.1$

**Fig. 1** Sample Image Coverings Determined by Tolerance Relation $\simeq_{gr,\varepsilon}$; see (1)

## 2 Related Works

Work on a basis for near sets began in 2002, motivated by image analysis and inspired by a study of the perception of the nearness of physical objects carried out in cooperation with Zdzisław Pawlak in [19]. This initial work led to the introduction of near sets [20], elaborated in [21, 13]. The introduction of tolerance near sets leads to a perception-based approach to discovering resemblances between digital images. This approach to the study of perceptual resemblances also leads to the introduction of perceptual representative spaces that are a generalization of the notion of a representative space introduced by J.H. Poincaré in [22] in a study of the contrast between the properties of physical and mathematical continua.

The proposed approach to measuring similarities between images in a visual space has been inspired by J.H. Poincaré's approach to organising visual sensations in sets of similar sensations [22, 23, 24] that later became known as tolerance classes, work by E.C. Zeeman on tolerance spaces inspired by human visual perception [25], and is directly related to the design of robotic vision systems (see, *e.g.*, [26, 27, 28, 29]). A vision system mimics the power and capability of the human sense of sight (*i.e.* the ability to detect light) combined with some type of cognition, perception, or interpretation of the stimulus. Even though a complete survey of vision systems is outside the scope of this chapter, the following examples are presented to give an idea as to the various types of vision systems. [30] present a vision system with the goal to position multiple cameras to identify and track multiple objects of interest in dynamic multiobject environments. [31] use 3-D time of flight (rather than stereo vision) to control a robot in a simulation of loading a container ship. The visual system generates range data to the objects that need to be loaded onto a ship, and performs segmentation of an image generated from range date to identify the centre of gravity and the rotation angle (information necessary to grab the simulated containers). Finally, another example of a vision system is the CogV system presented in [32] which mimics saccade and vergence movements in a binocular camera system to identify objects of interest in the field of view.

## 3 Tolerance Spaces and Visual Perception

The term *tolerance space* was coined by E.C. Zeeman in 1961 in modelling visual perception with tolerances [33, 34]. A tolerance space is a set $X$ supplied with a binary relation $\simeq$ (*i.e.*, a subset $\simeq\, \subset X \times X$) that is reflexive (for all $x \in X$, $x \simeq x$) and symmetric (for all $x, y \in X$, $x \simeq y$ and $y \simeq x$) but transitivity of $\simeq$ is not required. Sets of similar elements in a tolerance relation are called preclasses, introduced by M. Schroeder and M. Wright [35]. A set $A \subset\simeq$ is a preclass if, and only if $\forall x, y \in A,\ x \simeq y$. A tolerance class is a maximal preclass in a tolerance relation. Let $\mathfrak{O} \subset X$ denote a family of subsets of $X$. A family $\mathfrak{O}$ is a cover of a set $X$ if, and only if every element of $X$ belongs to some subset of $\mathfrak{O}$ [36]. A *covering* of $X$ is determined by $\simeq$.

For example, it is possible to define a tolerance space relative to sets of images. This is made possible by assuming that each image is a set of fixed points. Put $p \in [1, n], n \in \mathbb{N}$ (natural numbers). Let $O$ denote a set of perceptual objects (*e.g.*, $p \times p$ greyscale subimages) and let $gr(x)$ = average grey level of subimage $x$. Define the tolerance relation $\simeq_{gr,\varepsilon}$, where

$$\simeq_{gr,\varepsilon} = \{(x, y) \in O \times O \ : \ |gr(x) - gr(y)| \leq \varepsilon\}, \tag{1}$$

for some tolerance $\varepsilon \in \Re$ (reals). Then $(O, \simeq_{gr,\varepsilon})$ is a sample tolerance space. A tolerance $\simeq_{gr,\varepsilon}$ is directly related to the exact idea of closeness or resemblance (*i.e.*, perception being within some $\varepsilon$) in comparing object descriptions. The basic idea is to find objects such as digital images that resemble each other with a tolerable level of error. The choice of $p$ (for $p \times p$ subimages) determines the granularity of an image covering and influences the time required to construct image covers. Sample choices of $p \in \{5, 10, 15, 20\}$ with $\varepsilon = 0.1$ are shown in Figures 1.2-1.5 for covers of the drawing in the Fig. 1.1.

Tolerance relations provide a basis for representative spaces introduced by J.H. Poincaré [22, 23, 24]. A *representative space* is denoted by $\langle X, \simeq \rangle$ for a finite, non-empty set $X$ and a tolerance relation $\simeq$. For Poincaré, a representative space (*aka*, tolerance space) is a model for a physical continuum (pc). The elements of a pc are sets of similar sensations implicitly determined by a similarity relation. Such a similarity relation is perception-based and is characterized by the term *perceptually indistinguishable*. Let $W$ denote a set of sensations that result from lifting small, hand-held weights and let $\simeq_\varepsilon$ denote a perceptual similarity relation.

For example, let $w10, w11, w12 \in W$ denote 10, 11, and 12 gm weights. respectively. Put $\varepsilon = 1$. Assume that $X, Y \subset W$ and $X = \{w11, w11\}, Y = \{w11, w12\}$ are sets of perceptually indistinguishable weights determined by $\simeq_\varepsilon$, where each hand holds one of the weights. In effect, $X, Y$ are examples of preclasses belong to the relation $\simeq_\varepsilon$. By way of illustration, the preclass shown in Fig. 2 represents a set of perceptually indistinguishable weight-lifting sensations. The pair $(W, \simeq_\varepsilon)$ is an example of a Poincaré form of representative space that represents human weight-lifting perceptions, *i.e.*, $w1 \simeq_\varepsilon w2$, since $|w1 - w2| \leq 1$ and $w2 \simeq_\varepsilon w3$, since $|w2 - w3| \leq 1$. Poincaré's introduction of representative spaces in [22] stems from his interpretation of G. Fechner's 1860 notion of psychophysics, sensory circles and measurement of sensitivity to changes in external stimuli [37, 38]. Recent work on representative spaces has led to the introduction of perceptual representative spaces (PRSs) that provide useful frameworks for images analysis and image retrieval [3].

In a pc, almost solutions are common and a given equation has no exact solution. An *almost solution* of an equation (or a system of equations) is an object which, when substituted into the equation, transforms it into a numerical 'almost identity', *i.e.*, a relation between numbers which is true only approximately (within a prescribed tolerance) [39]. Equality in the physical world is meaningless, since it can never be verified either in practice or in theory. Hence, the basic idea in a tolerance space view of digital images, for example, is to replace the indiscernibility relation in rough sets with a perceptual tolerance relation in covering images with homologous regions where there is a high likelihood of overlaps, *i.e.*,

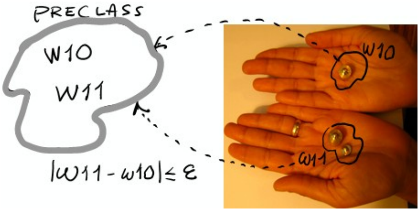


**Fig. 2** Set of Sensations

non-empty intersections between tolerance classes that are sets of subimages with similar descriptions.

This is the main idea underlying representative spaces introduced by Henri Poincaré [22, 23, 24]. A representative space (*aka*, tolerance space) is a model for a physical continuum (pc). The elements of a pc are sets of similar sensations implicitly determined by a similarity relation. Such a similarity relation is perception-based and is characterized by the term *perceptually indistinguishable*. Let $W$ denote a set of sensations that result from lifting small, hand-held weights and let $\simeq_{\varepsilon}$ denote a perceptual similarity relation. For example, let $w1, w2, w3 \in W$ denote 10, 11, and 12 gm weights and let $\varepsilon = 1$. Assume that $X, Y \subset W$ and $X = \{w1, w2\}, Y = \{w2, w3\}$ are sets of perceptually indistinguishable weights determined by $\simeq_{\varepsilon}$, where each hand holds one of the weights.

This paper has the following organization. A brief history of near sets that provide a foundation for perceptual image analysis is given in Sect. 4. This history includes a consideration of both spatially near sets defined relative to the F. Hausdorff lower distance between points and perceptually near sets defined relative to a metric space for a set of n-dimensional feature vectors of real numbers representing features of perceived objects and a distance function that measures the closeness or apartness of pairs of feature vectors. Visual tolerance spaces are introduced in Sect. 5. A visual tolerance space for digital images is introduced in Sect. 6. Measuring the resemblance between tolerance spaces is considered in Sec. 7. The main contribution of this paper is the introduction of a form of perceptual image analysis in terms of a methodology for determining the resemblance between pairs of visual tolerance spaces defined within the context of digital images.

## 4 History of Near Sets Underlying Perception in Image Analysis

The notion of nearness in mathematics and the notion of resemblance in the perception of physical objects such as digital images can be traced back to J.H. Poincaré, who introduced sets of similar sensations (nascent tolerance classes) to represent the results of G.T. Fechner's sensation sensitivity experiments [38] and a framework for the study of resemblance in representative spaces as models of what he termed physical continua [40, 22, 24]. The elements of a physical continuum (pc) are sets of sensations. The notion of a pc and various representative spaces (tactile, visual, motor spaces) were introduced by Poincaré in an 1894 article on the mathematical continuum [40], an 1895 article on space and geometry [22] and a compendious 1902 book on science and hypothesis [24] followed by a number of elaborations, *e.g.*, [41]. The 1893 and 1895 articles on continua (Pt. 1, ch. II) as well as representative spaces and geometry (Pt. 2, ch IV) are included as chapters in [24]. Later, F. Riesz introduced the concept of proximity or nearness of pairs of sets at ICM in 1908 [42]. During the 1960s, E.C. Zeeman introduced tolerance spaces in modeling visual perception [25]. A.B. Sossinsky observed in 1986 [39] that the main idea underlying tolerance space theory comes from Poincaré, especially [22] (Poincaré was not mentioned by Zeeman). In 2002, Z. Pawlak and J. Peters considered an informal approach to the perception of the nearness of physical objects such as snowflakes that was not limited to spatial nearness [43]. In 2006, a formal approach to the nearness of objects was considered by J. Peters, A. Skowron and J. Stepaniuk [44] in the context of proximity spaces [45, 46, 47, 48, 49]. In 2007, near sets were introduced by J. Peters [20, 21], followed by the introduction of tolerance near sets [50, 3].

In 1906, M. Fréchet introduced the idea of a metric spacemetric space in connection with a study of function spaces [51]. Fréchet observed that a distance function $\rho : X \times X \rightarrow \Re$ can be defined on any non-empty set $X$ and called it a metric. Thus was born the concept of a metric space.

**Definition 1. Pseudometric Space**
The pair $\langle X, \rho \rangle$ denote a metric space that consists of non-empty set $X$ and function $\rho$ defined on the set $X$, assuming non-negative values and satisfying the following conditions for all $x, y, z \in X$.
(M.1) $\rho(x,y) = 0 \iff x = y$,
(M.2) $\rho(x,y) = \rho(y,x)$ (symmetry),
(M.3) $\rho(x,z) \leq \rho(x,y) + \rho(y,z)$ (triangle inequality).

The notion of a pseudometric space formalizes the notion of the relative nearness of points. For example, a point $x$ is absolutely near a set $A$ if, and only if $\rho(x,A) = 0$ [52]. The study of metric spaces depends upon the fundamental concept of the limit point of a set that can be described in terms of nearness of a point to a set which was first suggested by F. Riesz [42].

**Remark 1. Point**
Let $\mathfrak{O}$ denote a family of subsets of a set $X$. The term *point* is interpreted to mean either a point is a tuple of real numbers in n-dimensional Euclidean space denoted by $\Re^n$ [36] or elements of a non-empty set $X$ in topological space $\langle X, \mathfrak{O} \rangle$ [53] or elements of $X$ in a metric space $\langle X, \rho \rangle$ [53]. In our case, a point is an n-dimensional feature vector in $\Re^n$ representing a description (see (6)) of a perceptual object such as a pixel or a subimage (collection of pixels) or an image patch (collection of subimages) in a digital image. ■

For a point $x$ and a non-empty set $B$, define a lower distance

$$\rho(x,B) = \inf_{y \in B} xy, \tag{2}$$

*i.e.*, the greatest lower bound of the distances of $x$ from points $y \in B$. F. Hausdorff introduced lower distance in his 1914 work on the elements of set theory [54], later translated by J.R. Aumann for the American Mathematical Society [55]. The Hausdorff lower distance $\rho(x,B) = \inf\{\rho(x,b) : b \in B\}$ is a continuous function of $x$ [55].

In a metric space $\langle X, \rho \rangle$, the *gap* between two non-empty sets $A, B \subset X$ is denoted by $D_\rho(A,B)$ in [56], *i.e.*,

$$D_\rho(A,B) = \inf\{\rho(a,b) : a \in A, b \in B\}. \tag{3}$$

Another formulation of the the gap distance (3) between a pair of non-empty sets $A, B \subset X$ in a metric space $\langle X, \rho \rangle$ appears in N. Bourbaki, written as

$$\rho(A,B) = \inf_{x \in A, y \in B} \rho(x,y). \tag{4}$$

F. Riesz introduced the concept of nearness of pairs of sets at the ICM in Roma in 1908 [42]. Two types of near sets can be identified, namely, spatially near sets based on spatial separation between points in pairs of sets and perceptually near sets based on the extent of the separation between object descriptions (n-dimensional feature vectors of real values) defined within a feature space. The nearness between sets $A, B$ was originally restricted to a zero lower distance between $A$ and $B$, *i.e.*, $D_\rho(A,B)$ [57] and applied in the context of digital images in [58]. By relaxing the zero distance requirement, a non-spatial nearness between sets is given in Def. 2.

**Definition 2. Near Sets of Perceptual Objects**
Put $\varepsilon \in [0, \infty)$ and let $\mathscr{B}$ denote a countable set of probe functions representing perceptual object features. Let $\langle X, \rho_{\mathscr{B}} \rangle$ denote a pseudometric space with a non-empty set of perceptual objects $X$ and distance function $\rho_{\mathscr{B}}$. Disjoint sets $A, B \subset X$ are near sets if, and only if the gap distance $D_{\rho_{\mathscr{B}}}(A,B) \leq \varepsilon$.

In other words, the nearness of disjoint sets of perceptual objects is defined relative to the Hausdorff lower distance between corresponding sets of descriptions of the objects. That is, disjoint sets of objects with descriptions that are close enough to each other are viewed as near sets. Def. 2 relaxes the requirement that $D_\rho(A,B) = 0$ for the nearness of sets in [45]. A non-spatial view of near sets appears in [59] and, more recently, nearness of sets based on resemblance (similar features of objects in sets) in [21], [48]. In this work, the distance function $\rho_{\mathscr{B}}$ is defined using the $L_p$ norm distance. With this in mind, notice that (5) can be rewritten using $\rho_{\mathscr{B}}$. Various forms of $\rho_{\mathscr{B}}$ considered in the context of pseudometric spaces are given in [60].

Put $\varepsilon \in [0, \infty)$. Put $A, B \subset X, x \in A, y \in B, \mathscr{B} = \{\phi : \phi : X \rightarrow \Re\}, \phi_i \in \mathscr{B}$ and $\tau_{\mathscr{B},\varepsilon}$ is a tolerance relation in a perceptual representative space $\langle X, \tau_{\mathscr{B},\varepsilon} \rangle$ introduced in [3], *i.e.*,

$$\tau_{\mathscr{B},\varepsilon} = \{(x,y) \in X \times X : \parallel \phi(x) - \phi(y) \parallel_p \leq \varepsilon\}, \tag{5}$$

where $\parallel \cdot \parallel_p = (\sum_{i=1}^{k} \cdot_i^p)^{\frac{1}{p}}$ ($L_p$ norm distance).

A function $\phi \in \mathscr{B}$ denotes a probe function representing a feature of a perceptual object. The notion of a probe function was introduced by M. Pavel [61] in a study of image registration viewed in the context of general topology. A *perceptual representative space* denoted by $\langle X, \tau_{\mathscr{B},\varepsilon} \rangle$ is a recent generalization of a Poincaré representative space [3]. This form a representative space is considered perceptual, since it is defined in terms of a finite, non-empty set of perceptual objects such as digital images and a tolerance relation $\tau_{\mathscr{B},\varepsilon}$. The relation $\tau_{\mathscr{B},\varepsilon}$ is itself considered perceptual, since it determines a covering of a set $X$ relative to a countable set of probe functions representing perceived features of objects in $X$.

**Proposition 1.** [3]
*A perceptual representative space $\langle X, \tau_{\mathscr{B},\varepsilon} \rangle$ is a generalization of a Poincaré representative space.*

Let

$$\phi_{\mathscr{B}}(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_k(x)) \tag{6}$$

denote a feature vector description of $x$ in $\Re^n$.

**Remark 2. Perceptual Metric Space**
If we let $X$ denote a finite, non-empty set of perceptual object descriptions (feature vectors), then $\langle X, \| \cdot \|_p \rangle$ is a perceptual pseudometric space. ■

**Definition 3. Perceptually Near Sets** [4, 62]
Sets $A, B$ are perceptually near (denoted by $A \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, B$) if, and only if, there are $x \in A, y \in B$ such that

$$\| \ \phi_{\mathscr{B}}(x) - \phi_{\mathscr{B}}(y) \ \|_p \leq \varepsilon.$$

Perceptually near sets can also be can be viewed more globally in terms of preclasses. Let $X, Y\mathscr{B}$ denote finite, non-empty, disjoint sets and countable set of probe functions. The tolerance relation in (5) determines a covering for $X$ and $Y$. Let $\mathrm{H}^{\varepsilon}_{\mathscr{B}}(X)$ denote the set of preclasses in the covering of $X$ determined by $\tau_{\mathscr{B},\varepsilon}$. Similarly, $\mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$ denotes the set of preclasses in the covering of $Y$ determined by $\tau_{\mathscr{B},\varepsilon}$.

**Proposition 2.** *Sets X and Y are perceptually near sets ($X \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, Y$) if, and only if there are preclasses $A \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X), B \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$ such that $A \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, B$.*

*Proof.* Consider
$\Rightarrow$ Assume $X \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, Y$. Then, from Def. 3, there are $x \in X, y \in Y$ such that $x \ \tau_{\mathscr{B},\varepsilon} \ y$, *i.e.*, $\| \ \phi_{\mathscr{B}}(x) - \phi_{\mathscr{B}}(y) \ \|_p \leq \varepsilon$. Put $x \in A \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X)$ and $y \in B \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$. Again, from Def. 3, $A \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, B$.
$\Leftarrow$ Assume $A \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X), B \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$ such that $A \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, B$. Then from Def. 3, there are $x \in A, y \in B$ such that $x \ \tau_{\mathscr{B},\varepsilon} \ y$. In other words, $X \,\underline{\bowtie}_{\tau_{\mathscr{B},\varepsilon}}\, Y$. □

Interest in nearness in digital images is fairly widespread. The concept of *nearness* enters as soon as one starts studying digital images (see, *e.g.*, [58, 63, 64, 2, 4, 3]). The digital image of a photograph should resemble, as accurately as possible, the original subject, *i. e.*, an image should be globally close to its source. Since proximity deals with global properties, it is appropriate for this study. The quality of a digital image depends on proximity and this proximity is more general than the one obtained from a metric. We note here that a digital image of a landscape is made up of a very limited number of points (depending on the sensory array of a camera), whereas the original landscape in a visual field contains many more points than its corresponding digital image. However, from the point of view of *perception*, they are near, depending on the tolerance we choose rather crudely in comparing visual field segments of a real scene with digital image patches (sets of scattered pixels). To make such comparisons work, the requirement that the image should appear as precise as possible as the original is relaxed. And the *precise-match* requirement is replaced by a *similarity* requirement so that a digital image should only remind us (within some tolerance) of the original scene. Then, for example, a cartoon in a newspaper of a person may be considered near, if parts of a cartoon are similar or if it resembles an original scene. Spatially near digital images were considered [58]. Descriptively near digital images are considered in [47, 64, 2, 4, 3].

## 5 Visual Tolerance Spaces and What We See

E.C. Zeeman pointed out that we do not perceive Euclidean 2D space with one eye because the Euclidean plane has an infinite number of points and the brain contains only a finite number of atoms [25]. Zeeman also illustrates tolerance spaces relative to visual sensation, *e.g.*, consider Zeeman's proposed visual acuity tolerance space.

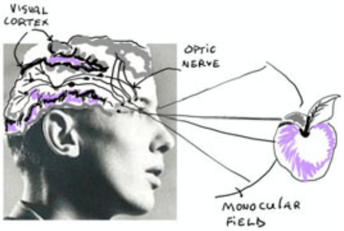


**Fig. 3** Retina-to-visual cortex sensory signals path

**Example 1. Visual Acuity Tolerance**
Let $S^2$ denote a large sphere concentric with the right eyeball and let $\langle V, \xi \rangle$ denote a visual acuity tolerance space (from [25]), where $V \subseteq S^2$ is the visual field of the right eye[1] while $\xi$ is the visual acuity tolerance on $V$ consisting of all pairs of elements of $V$ which are indistinguishable: its distance is less than $\lambda \in (0, +\infty)$ according to some metric from $S^2$ (Zeeman uses angular distance), *i.e.*, $\xi$ is a distance tolerance.

The notion of a sensation in Poincaré [41] and a physical model for a probe function[2] from near set theory [13] is implicitly explained by Zeeman [25] in terms of mappings of sense inputs from sensory units in the retina of the eye to visual cortex cells of the brain (see, *e.g.*, Fig. 3). In this case, Zeeman considers a new tolerance space $\langle Y, \eta \rangle$ in which $Y$ is a right visual lobe[3] and $\eta$ is a tolerance defined to be

[1] The visual field is everything in the physical world that causes light to fall on the retina of the eyeball.

[2] The term *probe* function was introduced by M. Pavel in 1993 as part of a study of image registration and classical topology in discerning patterns in digital images [61].

[3] Zeeman considers the case where the set of nerve cells in the lateral geniculate bodies are stimulated by steady points of light in a visual field.

$\eta = \alpha^{-1} \circ \alpha$, where a relation $\alpha \subseteq V \times Y$ is interpreted in terms of a light at $x$ that stimulates $y$ and two points $y_1, y_2 \in Y$ are in the relation $\eta$ when they are stimulated by at least one $x \in V$.

A sense input can be represented by a number representing the intensity of the light from the visual field impacting on the retina. The intensity of light from the visual field will determine the level of stimulation of a cortex cell from retina sensory input. Over time, varying cortex cell stimulation has the appearance of an electrical signal that travels along the optical path from retina to visual cortex (see, *e.g.* [65]). The magnitude of cortex cell stimulation is a real-value. The combination of retina impulses sent to cortex cells (visual stimulation) is likened to what Poincaré calls sets of sensations. This model for sensation underlies what is known as a probe function in near set theory [21, 13].

**Example 2. Indistinguishability Relation Between Sets**
Zeeman also points out that a tolerance $\xi$ on $V$ induces another tolerance $\Xi$ on the subsets of $V$: for any $X_1, X_2 \subseteq V$, $X_1 \Xi X_2$ iff $X_1 \subseteq \xi(X_2)$ and $X_2 \subseteq \xi(X_1)$. Zeeman calls the relation $\Xi$ an indistinguishability relation (between sets) [25]. Notice that in image processing, $X_1, X_2$ can be interpreted as digital images, where the relation $\Xi$ is an example of a relation between images (one of the main tools in a near set-based approach in image analysis, *e.g.*, [2, 64, 13, 66]).

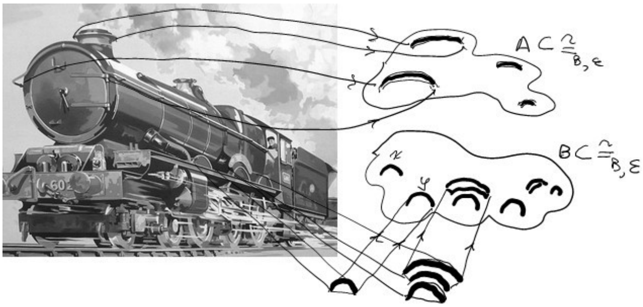

$A \subset \cong_{B,\varepsilon}$
$B \subset \cong_{B,\varepsilon}$
$x$
$y$

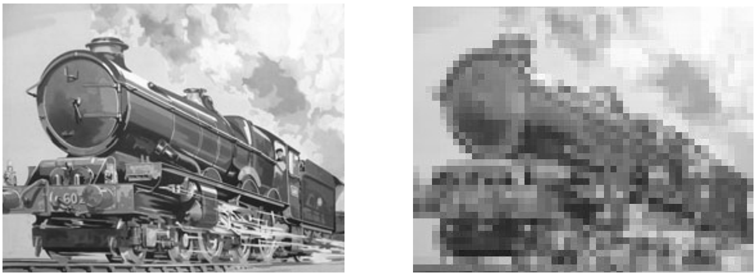
**Fig. 4** King George class Locomotive [67] Patch Preclasses

## 6 Visual Space

Visual space is considered both by Poincaré [24] and Zeeman [25]. A visual space is an outcome of perception consisting of representation of sensations resulting from light reflected from objects in a visual field[4], *i.e.*, representations of visual sensations taken collectively (following Poincare) create a visual space.

Representations of sensations are presented in near set theory in a perceptual representative space $\langle\, O, \tau_{\mathscr{B},\varepsilon} \,\rangle$ consisting of a finite, non-empty set $O$ of perceptual

[4] The visual field is everything in the physical world that causes light to fall on the retina of the eyeball.



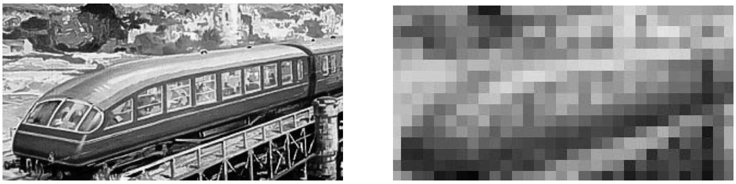
5.1: King George engine [67]

5.2: Engine cover, $p = 5$

**Fig. 5** Visual Space

objects such as patches in digital images and a tolerance relation $\tau_{\mathscr{B},\varepsilon}$ defined in terms of a countable family $\mathbb{F}$ of probe functions and a tolerance $\varepsilon$[3]. In this work, the term *visual representative space* (vis-à-vis J.H. Poincaré) is used interchangeably with the term *visual tolerance space* (vis-à-vis E.C. Zeeman). In digital image processing, perceptual objects can be interpreted in various ways [13], *e.g.* as pixels or image patches (an *image patch* is a $p \times p$[5] array of pixels in a digital image, where $p$ is the number of pixels). Visual images (as well as results of perceptions of other types) in near set theory are presented as subsets of a set of perceptual objects [13]. Thus, a visual sense input can be represented by image patches containing picture elements (pixels) with varying light intensities. In such cases, preclasses consist of image patches (see *e.g.* a preclass representing very similar various shadows with apparently similar grey level intensities in the sketch of a locomotive in Fig. 4). The intensity of light from the visual field will determine the level of stimulation of a cortex cell from retina sensory input. Over time, varying cortex cell stimulation has the appearance of an electrical signal that travels along the optical path from retina to visual cortex (see, *e.g.*, [65]). The magnitude of cortex cell stimulation is a real-value. Multiple retina impulses sent to cortex cells (visual stimulation) are likened to what Poincaré calls sets of sensations. Poincaré confines his comments on visual space to its lack of homogeneity, which is consistent with the contemporary view of visual space (*i.e.*, all points of the retina do not play the same role (points of light at the edge of the retina are less distinct than points of light impacting on the centre of the retina) and non-isotropic (impressions made by perception of depth of points of light are not the same in all directions) [24, II.4]. In keeping with Poincaré's approach in assembling sets of similar visual sensations, one can imagine the assembly of sets of similar visual sensations (preclasses) resulting from the perception of similar light intensities (see, *e.g.*, bounded region representing a sample hat preclass containing 5 × 5 image patches with similar average grey levels in Figs 5.2 and 6.2). Scattered shaded boxes in image patches with the same average greylevel intensity represent a single preclass in an image cover.

[5] Here we follow the notational convention widely used in image processing (*e.g.*, [68]), where the term *array* of pixels is synonymous with a matrix of pixels.



6.1: Coach train [69]

6.2: Train cover

**Fig. 6** Another Visual Space

**Example 3.** Visual Tolerance Space for Digital Images
There is an obvious practical application of Poincaré's approach to defining a representative space for vision. That is, it is possible to extract from image regions[6] clusters of preclasses containing digital image patches with similar average intensities. Let $O$ denote a set of grey level image patches and let $\phi_{gr} : O \rightarrow \Re$. For $x \in O$, $\phi_{gr}(x)$ is an average grey level of patch $x$. For the tolerance relation in (5), put $\mathscr{B} = \{\phi_{gr}\}$ and write $\tau_{\phi_{gr},\varepsilon}$ to denote a single function relation. Consider the visual tolerance space $\langle O, \tau_{\phi_{gr},\varepsilon}\rangle$ with a relation $\tau_{\phi_{gr},\varepsilon}$ defined in (7).

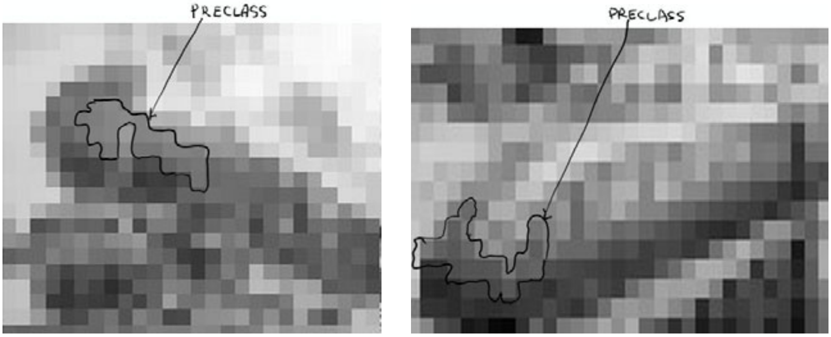
$$\tau_{\phi_{gr},\varepsilon} = \{(x,y) \in O \times O : |\phi_{gr}(x) - \phi_{gr}(y)| \leq \varepsilon\}. \tag{7}$$

For example, the image patches covered with ■ shaded boxes represent a maximal locomotive preclass in Fig. 5.2). Similarly, one can identify a coach train preclass in Fig. 6.2 that is part of the cover for the train shown in Fig. 6. Since both covers have image patches containing ■ shaded boxes, then Fig. 5.2 $\underline{\underline{\bowtie}}_{\tau_{\phi_{gr},\varepsilon}}$ Fig. 6.2, *i.e.*, the locomotive in Fig. 5.2 resembles the coach train in Fig. 6.2 for subimage description based on average greylevel, $p = 5, \varepsilon = 0.1$. Sample preclasses are shown in Fig. 7.1 and 7.2.

## 7 Resemblance between Tolerance Spaces

Poincaré and Zeeman presage the introduction of near sets [20, 21] and research on similarity relations, *e.g.*, [16, 39, 18]. Distance tolerance relations are directly related to the idea of closeness or resemblance (*i.e.*, being within some tolerance) in comparing objects such as the digital images. By way of application of Poincarś approach in defining visual spaces and Zeeman's approach to tolerance relations, the basic idea in this section is to compare objects such as image patches in the interior of digital images (*e.g.*, the $10 \times 10$ patches in Fig. 7.1 and Fig. 7.2) and discern image patches that resemble each other relative to one or more features with a tolerable level of error. By restricting the comparison of image patches to one feature (*e.g.*, average grey level of an image patch), the study of resemblance between patches in digital images is analogous to Poincaré's approach to discovering sets of sensations

[6] An image region is a set of contiguous image patches and more than one region can contain very similar image patches (see, *e.g.*, [13]).



7.1: Engine preclass

7.2: Train Preclass

**Fig. 7** Sample Image Preclasses, $p = 10, \varepsilon = 0.1$

in a representative space such as the weight-lifting space in Example 4 or the visual space in Example 3 (see, in particular, the realization of a set of indistinguishable sensations in comparing hand-held weights in Fig. 2). Considering more than one feature in the study of Poincaré's representative spaces is outside the scope of this article (see, *e.g.*, [2, 64] for tolerance spaces defined relative to multiple features such as colour, texture, edge intensity, edge-orientation).

**Example 4. Poincaré's Representative Weight-Lifting Space**
Poincaré repeatedly refers to Fechner's weight-lifting experiment. In Poincaré's hands, there is a shift from Fechner's search for a measure of sensation to the introduction of a representative space[7] containing representations of our sensations [24, II.4]. Let $W$ denote a set containing sensations resulting from weight lifting, *e.g.*, sensations w1,w2,w3 recorded by Fechner for 10, 11, and 12 gm weights, respectively. From a perceptual point of view, Poincaré observes that w1 is indistinguishable from w2, w2 is indistinguishable form w3, but w1 is distinguishable from w3.

This view of weight-lifting sensations leads to sets of indistinguishable sensations (*i.e.*, preclasses such as the one shown in Fig 2). Let $\varepsilon$ denote a threshold on 'perceived difference in weights' and let $\phi : W \rightarrow \Re$. For $x \in W (\phi(x)$ is the perceived weight, the sensation one experiences when lifting $x$). Poincaré *implicitly* defines a tolerance space $\langle W, \simeq_{\phi,\varepsilon} \rangle$ (he calls it a *representative space* [24]). *Impicit* in Poincaré's reasoning about a represenative weight-lifting space is the tolerance relation defined in (8).

$$\simeq_{\phi,\varepsilon} = \{(x,y) \in W \times W : |\phi(x) - \phi(y)| \leq \varepsilon\}. \tag{8}$$

Let the family of preclasses of a perceptual representative space $\langle W, \simeq_{\phi,\varepsilon} \rangle$ be denoted by $\mathrm{PH}^{\varepsilon}_{\phi}(W)$. It follows from definitions that a set $A \subset W$ is a preclass of the relation $\simeq_{\phi,\varepsilon}$, $A \in \mathrm{PH}^{\varepsilon}_{\phi}(W)$, if $|\phi(x) - \phi(y)| \leq \varepsilon$ for every $x, y \in A$. Then taking Fechner's example and restricting $W = \{w1, w2, w3\}$, the weight-lifting sensations

[7] *l'espace représentatif* [22, p.2], [24, I.4,77].

separate into overlapping preclasses $A, B \in \mathrm{PH}^{\varepsilon}_{\phi}$, $A = \{w_1, w_2\}$ and $B = \{w_2, w_3\}$. These preclasses are elements in what Poincaré calls a physical continuum [24, I.2, 51]. Poincaré did not consider the resemblance between representative spaces. Zeeman [33, p. 242] does consider pairs of tolerance spaces $\langle X, \xi \rangle$ and $\langle X, \eta \rangle$ that are related to each other if there exists a relation $\alpha \subset X \times Y$ such that $\xi = \alpha \cdot \alpha^{-1}$ and $\eta = \alpha^{-1} \cdot \alpha$ (see Example 1). However, Zeeman did not consider the similarity between tolerance spaces. That is the central topic in this section (an approach to measuring the resemblance between pairs of visual tolerance spaces is given here).

**Example 5. Tolerance Space-Based Nearness Measure**
Let $O$ denote a set of image patches and let $X, Y \subset O$ denote sets of image patches in individual digital images. After the manner in Example 3, consider a tolerance relation $\tau_{\phi,\varepsilon}$ on $X, Y$ defined relative to a probe function $\phi$ and $\varepsilon \in (0, +\infty)$, *i.e.*, for the relation defined in (5), put $\mathscr{B} = \{\phi\}$. Then consider the covering of tolerance classes on the set $X \cup Y$ for $X, Y \subset O$ determined by $\tau_{\phi,\varepsilon}$.

Let $\mathrm{H}^{\varepsilon}_{\phi}(X \cup Y)$ denote the family of all tolerance classes of relation $\tau_{\phi,\varepsilon}$ on the set $X \cup Y$. Notice that $\tau_{\phi,\varepsilon}$ is defined over a perceptual pseudometric space $\langle X \cup Y, \| \cdot \|_p \rangle$ (see Remark 2). The tNM nearness measure (introduced in [70], elaborated in [71, 2]) estimates the degree of resemblance between $X$ and $Y$. This measure is defined in (9) as the weighted average of the closeness between the cardinality (size) of sets $A \cap X$ and $A \cap Y$ where $A \in \mathrm{H}^{\varepsilon}_{\phi}(X \cup Y)$ and the cardinality of tolerance class $A$ is used as the weighting factor.

$$tNM(X,Y) = \frac{\sum\limits_{A \in \mathrm{H}^{\varepsilon}_{\phi}(X \cup Y)} \left( \frac{\min\{\,|A \cap X|, |A \cap Y|\,\}}{\max\{\,|A \cap X|, |A \cap Y|\,\}} \cdot |A| \right)}{\sum\limits_{A \in \mathrm{H}^{\varepsilon}_{\phi}(X \cup Y)} |A|}. \tag{9}$$

**Example 6. Resemblance Between a Pair of Visual Tolerance Spaces**
Start with a pair of visual tolerance spaces $\langle X, \tau_{\phi,\varepsilon} \rangle$ and $\langle Y, \tau_{\phi,\varepsilon} \rangle$ for particular choices of the probe function $\phi$ (in this Example, two choices of $\phi$ are given). First, the digital images in Fig. 5.1 and Fig. 6.1 are transformed into sets of $p \times p$ image patches. In this example, $p = 5, \varepsilon = 0.01$. The transformation of digital images into sets of image patches for particular $p, \varepsilon$ and the use of tNM in (9) to measure the resemblance between images has recently been made possible in a public-domain NEAR system toolset [72].

Put $X, Y$ in Example 5 equal to the images viewed as sets of image patches in Fig. 7.1 and Fig. 7.2, respectively. After the manner in Example 3, consider a tolerance $\tau_{\phi_{gr},\varepsilon}$ on $X, Y$. That is, consider a covering of tolerance classes on a set $X \cup Y$ determined by $\tau_{\phi_{gr},\varepsilon}$ given in (7) and $\phi_{gr}$ as defined in Example 3 with $\varepsilon \in (0, +\infty)$. Let $\mathrm{H}^{\varepsilon}_{\phi_{gr}}(X \cup Y)$ denote the family of all tolerance classes of relation $\tau_{\phi_{gr},\varepsilon}$ on the set $X \cup Y$.

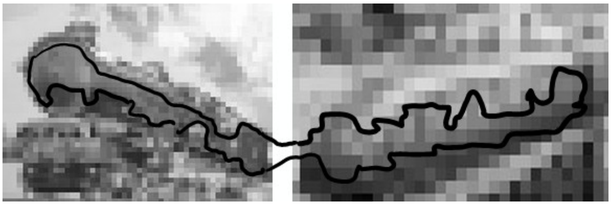


**Fig. 8** Preclass Consisting of Similar Image Patches from Figs. 7.1 & 7.2

A sample preclass constructed with similar image patches (some in the set of image patches in Fig. 7.1 and some image patches in Fig 7.2) is shown in Fig. 8. Table 1 gives some sample results for various choices of $p$ and $\varepsilon$.

**Table 1** Greyscale Image Nearness Estimates Table.

| Relation | Edge Length $p$ | Tolerance $\varepsilon$ | $tNM(X,Y)$ |
|---|---|---|---|
| $\tau_{\phi_{gr},0.1}$ | 20 | 0.1 | 0.2409 |
| $\tau_{\phi_{gr},0.15}$ | 20 | 0.1 | 0.2499 |
| $\tau_{\phi_{gr},0.2}$ | 20 | 0.2 | 0.2540 |
| $\tau_{\phi_{gr},0.1}$ | 15 | 0.1 | 0.2242 |
| $\tau_{\phi_{gr},0.15}$ | 15 | 0.15 | 0.2276 |
| $\tau_{\phi_{gr},0.2}$ | 15 | 0.2 | 0.2410 |

From Table 1, it is apparent that the original images in Fig. 5.1 and Fig. 6.1 are not very near (have little resemblance). For the experiments recorded in Table 1, the best result is 0.2540 for $p = 20, \varepsilon = 0.2$.

However, the measure of image resemblance using tNM will change (probably increase), if one were to consider other features (either singly or in various combinations). For example, if we introduce a probe function (denoted by $\phi_{eo}$) to compute the average edge orientation of subimages in the locomotive and touring train, one can expect higher nearness measurements, since one can observe that there are many edges that are similar in both images (see Table 2).

The results in Table 1 corroborate our perception concerning the abundance of similar edges in both the locomotive and touring train images. That is, from 0.5151 with $p = 20, \varepsilon = 0.01$, and from 0.5161 with $p = 15, \varepsilon = 0.01$, it can be concluded that the images in Fig. 5.1 and Fig. 6.1 are moderately near each other relative to edge orientation. Many other results with various nearness measures have been reported in [71, 64, 5, 6, 7, 8, 73] and numerous other studies of image resemblance in [2].

**Table 2** Image Edge Orientation Nearness Estimates Table.

| Relation | Edge Length $p$ | Tolerance $\varepsilon$ | $tNM(X,Y)$ |
|---|---|---|---|
| $\tau_{\phi_{eo},0.01}$ | 20 | 0.01 | 0.5151 |
| $\tau_{\phi_{eo},0.1}$ | 20 | 0.1 | 0.4670 |
| $\tau_{\phi_{eo},0.2}$ | 20 | 0.2 | 0.4465 |
| | | | |
| $\tau_{\phi_{eo},0.01}$ | 15 | 0.01 | 0.5161 |
| $\tau_{\phi_{eo},0.1}$ | 15 | 0.1 | 0.4528 |
| $\tau_{\phi_{eo},0.2}$ | 15 | 0.2 | 0.4529 |

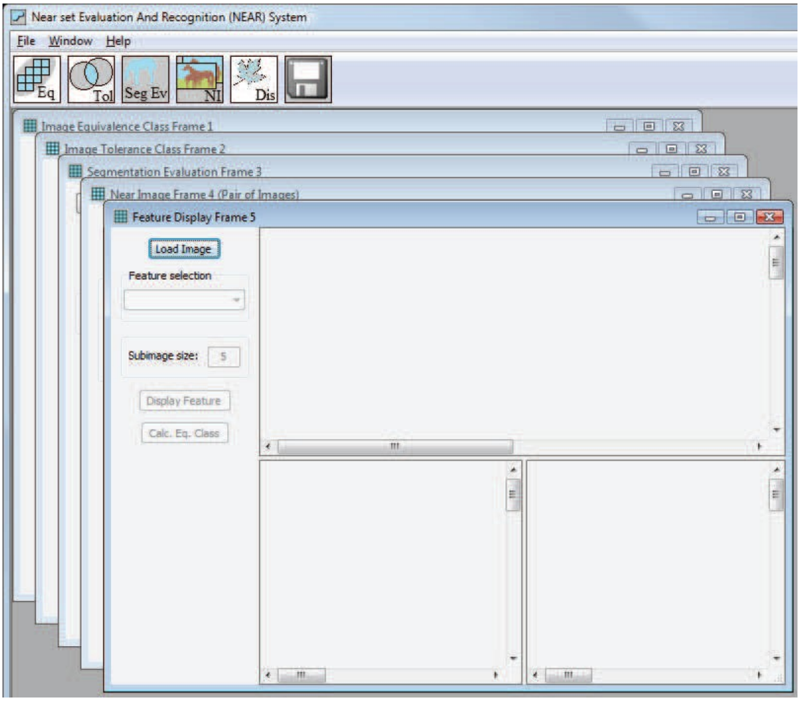

Near set Evaluation And Recognition (NEAR) System
File
Window
Help
Eq
Tol
Seg Ev
NI
Dis
Image Equivalence Class Frame 1
Image Tolerance Class Frame 2
Segmentation Evaluation Frame 3
Near Image Frame 4 (Pair of Images)
Feature Display Frame 5
Load Image
Feature selection
Subimage size: 5
Display Feature
Calc. Eq. Class

**Fig. 9** NEAR system GUI [74]

## 8 Implementation of a Nearness System

The goal of the NEAR system is to demonstrate applications of the near set theory. A complete tutorial and the NEAR system itself is available for downloading at [74]. The system implements a Multiple Document Interface (MDI) (see, *e*.g., Fig. 9) where each separate processing task is performed in its own child frame. The objects (in the near set sense) in this system are subimages of the images being processed and the probe functions (features) are image processing functions defined on the subimages. The system was written in C++ and was designed to facilitate the addition of new processing tasks and probe functions[8]. Currently, the system performs a number of major tasks, namely, displaying equivalence and tolerance classes for an image, segmentation evaluation, measuring the nearness of pairs of images, content-based image retrieval (CBIR) on a selected image database, displaying the output of processing an image using an individual probe functions, and storing the results of an image analysis session in a location selected by a researcher. This report is organized as follows:

topic.1 Brief introduction to near set theory implemented in the NEAR system,
topic.2 Implemented distance functions: tNM, Hausdorff, Hamming,
topic.3 Perceptual image processing

- Probe functions,
- Average greyscale value,
- Normalized RGB,
- Shannon entropy,
- Pal entropy,
- Edge-based probe functions (Mallat's method),
- Grey level co-occurrence matrices,
- Zernike moments
- CIELUV colour space,

topic.4 Approximate nearest neighbours
topic.5 Equivalence class frame
topic.6 Tolerance class frame
topic.7 Segmentation evaluation frame
topic.8 Near image frame
topic.9 Feature display frame

## 9 Conclusion

This chapter introduces a visual perception approach in image analysis (briefly, *perceptual image analysis*). This approach has been motivated by a need to solve the image correspondence problem in terms of perceived resemblances between digital images. What the eye sees should correspond, to some extent, to measures of nearness between pairs of images. Pointers on how to go about establishing a perceptual image analysis can be found in studies of representative space models of physical continua by Poincaré toward the end of the 19th century and the connection between visual acuity and tolerance spaces introduced by Zeeman during the 1960s. In addition, the parallel discoveries about spatially near sets that began with F. Riesz in 1908 and continued with the introduction of proximity spaces in a seminal work by S.A. Naimpally in 1970, amplified and extended by others, led to the

[8] Parts of the Graphical User Interface (GUI) were inspired by the GUI reported in [75] and the wxWidgets example in [76].

recent discovery of tolerance near sets that are perception-based and not limited to spatial nearness. The important thing here is the need to arrive at an understanding of the meaning of a *point* in perceptual representative spaces and the formulation of description-based pseudometric spaces suitable for image analysis. This chapter is attentive to both research streams (*i.e.*, tolerance space stream and near set stream) in presenting a viable approach to solving the image correspondence problem and in arriving at a satisfactory approach to perceptual image analysis.

## References

1. Morrow, G.: Dog on stilts. Punch, or the London Charivari CXXXV, 168–183 (1908)
2. Pal, S., Peters, J.: Rough Fuzzy Image Analysis. Foundations and Methodologies. CRC Press, Taylor & Francis Group (September 2010); ISBN 13: 9781439803295, ISBN 10: 1439803293
3. Peters, J.: Corrigenda and addenda: Tolerance near sets and image correspondence. Int. J. Bio-Inspired Computation 2(5), 310–318 (2010)
4. Peters, J.F.: Tolerance near sets and image correspondence. International Journal of Bio-Inspired Computation 1(4), 239–245 (2009)
5. Ramanna, S., Meghdadi, A.H.: Measuring resemblances between swarm behaviours: A perceptual tolerance near set approach. Fundamenta Informatica 95(4), 533–552 (2009); ISSN:0169-2968.
6. Ramanna, S.: Discovering image similarities: Tolerance near set approach. In: Pal, S., Peters, J. (eds.) Rough Fuzzy Image Analysis, pp. 12.1–12.15. CRC Press, Boca Raton (2010)
7. Ramanna, S.: Perceptually near pawlak partitions. In: Peters, J.F., Skowron, A., Słowiński, R., Lingras, P., Miao, D., Tsumoto, S. (eds.) Transactions on Rough Sets XII. LNCS, vol. 6190, pp. 170–192. Springer, Heidelberg (2010)
8. Wasilewski, P., Peters, J., Ramanna, S.: Perceptual tolerance intersection. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 277–286. Springer, Heidelberg (2010)
9. Meghdadi, A., Peters, J., Ramanna, S.: Tolerance classes in measuring image resemblance. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009. LNCS (LNAI), vol. 5712, pp. 127–134. Springer, Heidelberg (2009)
10. Gupta, S., Patnaik, K.: Enhancing performance of face recognition systems by using near set approach for selecting facial features. Journal of Theoretical and Applied Information Technology 4(5), 433–441 (2008)
11. Henry, C., Peters, J.: Image pattern recognition using approximation spaces and near sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 475–482. Springer, Heidelberg (2007)
12. Peters, J., Ramanna, S.: Affinities between perceptual granules: Foundations and perspectives. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling. SCI, vol. 182, pp. 49–66. Springer, Heidelberg (2009)
13. Peters, J.F., Wasilewski, P.: Foundations of near sets. Elsevier Science 179(1), 3091–3109 (2009)
14. Hassanien, A., Abraham, A., Peters, J., Schaefer, G., Henry, C.: Rough sets and near sets in medical imaging: A review. IEEE Trans. Info. Tech. in Biomedicine 13(6), 955–968 (2009), doi:10.1109/TITB.2009.2017017

15. Henry, C., Peters, J.: Near sets. Wikipedia (2010), http://en.wikipedia.org/wiki/Near_sets
16. Pogonowski, J.: Tolerance Spaces with Applications to Linguistics. University of Adam Mickiewicz Press, Poznań (1981)
17. Grätzer, G., Wenzel, G.: Tolerances, covering systems, and the axiom of choice. Archivum Mathematicum 25(1-2), 27–34 (1989)
18. Wasilewski, P.: On selected similarity relations and their applications into cognitive science. PhD thesis, Department of Logic, Cracow (2004) (in Polish)
19. Pawlak, Z., Peters, J.: Jak blisko (how near). Systemy Wspomagania Decyzji I, 57, 109 (2002, 2007) ISBN 83-920730-4-5
20. Peters, J.: Near sets. special theory about nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
21. Peters, J.: Near sets. general theory about nearness of objects. Applied Mathematical Sciences 1(53), 2009–2029 (2007)
22. Poincaré, J.: L'espace et la géomètrie. Revue de m'etaphysique et de morale 3, 631–646 (1895)
23. Poincaré, J.: Le continu mathématique. Revue de méaphysique et de morale 1, 26–34 (1893)
24. Poincaré, J.: Sur certaines surfaces algébriques; troisième complément 'a l'analysis situs. Bulletin de la Société de France 30, 49–70 (1902)
25. Zeeman, E.: The topology of the brain and visual perception. In: Fort Jr., M.K. (ed.) University of Georgia Institute Conference Proceedings, Topology of 3-Manifolds and Related Topics, pp. 240–256. Prentice-Hall, Inc., Englewood Cliffs (1962)
26. Peters, J.F., Ramanna, S.: Modeling timed behavior in real-time systems with temporal logic. Cybernetics and Systems: An Int. J. 22, 583–608 (1991)
27. Peters, J.F.: Typed timed input/output automata in real-time cybernetic explanation. Cybernetics and Systems: An Int. J. 24, 115–137 (1993)
28. Peters, J.: Reasoning about real-time systems. Australian Computer Journal 25(4), 135–148 (1993)
29. Peters, J.F., Borkowski, M., Henry, C., Lockery, D., Gunderson, D.S.: Line-crawling bots that inspect electric power transmission line equipment. In: Proceedings of the third International Conference on Autonomous Robots and Agents (ICARA 2006), Palmerston North, New Zealand, pp. 39–45 (2006)
30. Bakhtari, A., Benhabib, B.: An active vision system of multitarget surveillance in dynamic environments. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 37(1), 190–198 (2007)
31. Hussmann, S., Liepert, T.: Three-dimensional tof robot vision system. IEEE Transactions on Instrumentation and Measurement 58(1), 141–146 (2009)
32. Zhang, X., Tay, A.L.P.: A physical system for binocular vision through saccade generation and vergence control. Cybernetics and Systems 40, 549–568 (2009)
33. Zeeman, E.: The topology of the brain and the visual perception. In: Fort, K.M. (ed.) Topology of 3-manifolds and Selected Topics, pp. 240–256. Prentice Hall, New Jersey (1962)
34. Zeeman, E.C., Buneman, O.P.: Tolerance spaces and the brain. In: Waddingto, C.H. (ed.) Towards a Theoretical Biology, pp. 140–151. The Origin of Life, Aldine Pub. Co., Aldine (1968)
35. Schroeder, M., Wright, M.: Tolerance and weak tolerance relations. Journal of Combinatorial Mathematics and Combinatorial Computing 11, 123–160 (1992)
36. Kelley, J.: General Topology. Springer, Berlin (1955)

37. Fechner, G.T.: Elements of Psychophysics, vol. I. Holt, Rinehart & Winstonton, London (1966) H. E. Adler's trans. of Elemente der Psychophysik (1860)
38. Fechner, G.: Elemente der Psychophysik, vol. 2. E.J. Bonset, Amsterdam (1860)
39. Sossinsky, A.B.: Tolerance space theory and some applications. Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications 5(2), 137–167 (1986)
40. Poincaré, J.: Sur la nature du raisonnement mathématique. Revue de méaphysique et de morale 2, 371–384 (1894)
41. Poincaré, H.: Mathematics and Science: Last Essays. Kessinger Publishing, N. Y (1963), Bolduc's, J.W. trans. of Dernières Pensées (1913)
42. Riesz, F.: Stetigkeitsbegriff und abstrakte mengenlehre. m IV Congresso Internazionale dei Matematici II, 18–24 (1908)
43. Pawlak, Z., Peters, J.: Jak blisko. Systemy Wspomagania Decyzji I, 57 (2007)
44. Peters, J., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
45. Naimpally, S.A., Warrack, B.D.: Proximity spaces. In: Cambridge Tract in Mathematics, vol. (59). Cambridge University Press, Cambridge (1970)
46. Mozzochi, C., Naimpally, S.: Uniformity and proximity. In: Allahabad Mathematical Society Lecture Note Series, vol. 2, pp. xii+153. The Allahabad Math. Soc., Allahabad (2009); ISBN 978-81-908159-1-8
47. Naimpally, S.A.: Near and far. A centennial tribute to Frigyes Riesz. Siberian Electronic Mathematical Reports 6, A.1–A.10 (2009)
48. Naimpally, S.: Near and far. A centennial tribute to Frigyes Riesz. Siberian Electronic Mathematical Reports 2, 144–153 (2009)
49. Hocking, J., Naimpally, S.: Nearness—a better approach to continuity and limits. In: Allahabad Mathematical Society Lecture Note Series, vol. 3, pp. iv+66. The Allahabad Math. Soc., Allahabad (2009); ISBN 978-81-908159-1-8
50. Peters, J.: Tolerance near sets and image correspondence. Int. J. Bio-Inspired Computation 1(4), 239–245 (2009)
51. Fréchet, M.: Sur quelques points du calcul fonctionnel. Rend. Circ. Mat. Palermo 22, 1–74 (1906)
52. Hazelwinkel, M.E.: Encyclopaedia of Mathematics, vol. 5. Kluwer Academic Publishers, Dordrecht (1995)
53. Engelking, R.: General Topology, Revised & completed edition. Heldermann Verlag, Berlin (1989)
54. Hausdorff, F.: Grundzüge der mengenlehre. Verlag Von Veit & Comp., Leipzig (1914)
55. Hausdorff, F.: Set Theory. AMS Chelsea Publishing, Providence (1957); Aumann, J.R., et al. (trans.) Mengenlehre (1937)
56. Beer, G., Lucchetti, R.: Weak topologies for the closed subsets of a metrizable space. Trans. Am. Math. Soc. 335(2), 805–822 (1993)
57. Naimpally, S., Warrack, B.: Proximity Spaces. Cambridge Tract in Mathematics, vol. (59). Cambridge University Press, Cambridge (1970)
58. Pták, P., Kropatsch, W.: Nearness in digital images and proximity spaces. In: Nyström, I., Sanniti di Baja, G., Borgefors, G. (eds.) DGCI 2000. LNCS, vol. 1953, pp. 69–77. Springer, Heidelberg (2000)
59. Mozzochi, C., Gagrat, M., Naimpally, S.: Symmetric Generalized Topological Structures, p. i+73. Exposition Press, Hicksville (1976)
60. Peters, J., Meghdadi, A., Ramanna, S.: Fuzzy metrics for near rough sets. Theoretical Computer Science (2010) (submitted)
61. Pavel, M.: Fundamentals of Pattern Recognition. Marcel Dekker, Inc., New York (1993)

62. Peters, J., Ramanna, S.: Affinities between perceptual granules: Foundations and perspectives. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling. SCI, vol. 182, pp. 49–66. Springer, Berlin (2009)
63. Naimpally, S.: Near and far. A centennial tribute to Frigyes Riesz. Siberian Electronic Mathematical Reports 2, 144–153 (2009)
64. Peters, J.F., Puzio, L., Szturm, T.: Measuring nearness of rehabilitation hand images with finely-tuned anisotropic wavelets. In: Choraś, R.S., Zabludowski, A. (eds.) Image Processing & Communication Challenges, pp. 342–349. Academy Publishing House, Warsaw (2009)
65. Schatz, C.: The developing brain. Scientific American 267(3), 60–67 (1992)
66. Hassanien, A.E., Abraham, A., Peters, J.F., Schaefer, G., Henry, C.: Rough sets and near sets in medical imaging: A review. IEEE Transactions on Information Technology in Biomedicine 3(6), 955–968 (2009)
67. Gordon, W.: Our Home Railways. Frederick Warne and Co., London (1910)
68. Gonzales, R., Woods, R.: Digital Image Processing, 3rd edn. Pearson Prentice Hall, Upper Saddle River (2008)
69. MacDermot, E.: History of the Great Western Railway. The Great Western Railway, Paddington (1927)
70. Henry, C., Peters, J.: Near set image in an objective image segmentation evaluation framework. In: GEOBIA 2008 Pixels, Objects, Intelligence. GEOgraphic Object Based Image Analysis for the 21st Century, pp. 1–6. University of Calgary, Alberta (2008)
71. Henry, C.: Near set evaluation and recognition (near) system. In: Pal, S., Peters, J.F. (eds.) Rough Fuzzy Image Analysis. Foundations and Methodologies, ch.7, pp. 7.1–7.22. CRC Press, Boca Raton (2010)
72. Henry, C., Peters, J.F.: Near set evaluation and recognition (near) system. Technical report, Computational Intelligence Laboratory, University of Manitoba, UM CI Laboratory Technical Report No. TR-2009-015 (2009)
73. Meghdadi, A.H., Peters, J.F., Ramanna, S.: Tolerance classes in measuring image resemblance. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009. LNCS, vol. 5712, pp. 127–134. Springer, Heidelberg (2009)
74. Henry, C., Peters, J.F.: Near set evaluation and recognition (near) system. Technical report, Computational Intelligence Laboratory, University of Manitoba, UM CI Laboratory Technical Report No. TR-2010-017 (2010), http://wren.ee.umanitoba.ca
75. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low level vision. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec City, vol. 4, pp. 150–156 (2002)
76. wxWidgets: wxwidgets cross-platform gui library v2.8.9 (2009), http://www.wxwidgets.org

# Chapter 7
# An Introduction to Magnetic Resonance Imaging: From Image Acquisition to Clinical Diagnosis

Kenneth Revett

**Abstract.** Magnetic resonance imaging provides a comprehensive and non-invasive view of the structural features of living tissue at very high resolution (typically on the 1-2 mm scale). A variety of pulse sequences have been developed that provide quantitative information regarding the structural features of a variety of tissue classes, providing details that are extremely beneficial in a clinical setting. Unlike positron emission tomography (PET), MRI as it does not deploy the use of radioactive isotopes, and hence can be performed repeatedly. Modern day MRI scanners can provide extremely high resolution images in a relatively short period of time (approximately 20 minutes) on average in a typical diagnostic scan. A variety of measurements can be made in a single scanning session through the application of serial pulse sequences. These pulse sequences are computer programmes that control the scanner parameters, which in turn control factors such as tissue contrast. By deploying the appropriate pulse sequence, One can obtain detailed information about the vasculature of a region of the body (magnetic resonance angiogram), deep tissue injury, and more recently one can obtain information regarding the microstructural features of the brain. Indeed, MRI is routinely used to identify and/or confirm the diagnosis of a variety of brain parenchyma or vasculature diseases such as multiple sclerosis and stroke respectively. With further improvements in the electronics and pulse sequences, more detailed and accurate imaging techniques may provide medical science with the opportunity to automate the diagnosis of a variety of diseases which present ultastructural changes.

**Keywords:** diffusion weighted imaging, fibre tractography, image segmentation, magnetic resonance imaging, multiple sclerosis, pulse sequence.

Kenneth Revett
Faculty of Informatics & Computer Science
British University in Egypt
El Sherouk City
Egypt
e-mail: Ken.revett@bue.edu.eg

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 127–161.
springerlink.com

# 1 Introduction to MRI

Magnetic resonance imaging is a technology that provides high quality images utilising the intrinsic magnetic properties of matter. MRI is utilised as a diagnostic tool in medicine, which will be the focus of this chapter. More specifically, MRI is a non-invasive method for acquiring high resolution images of the internal structures of the body non-invasively. This ability has placed MRI as an indispensable tool for clinical diagnosis of a number of diseases, ranging from minor sports injuries to multiple sclerosis. The purpose of this chapter is to provide an overview of structural MR imaging and describe how MRI can be used to diagnose a variety of neurological diseases. The first section of the chapter will describe how an MRI is obtained – covering the basic aspects of image acquisition. This is followed by a discussion of imaging modalities such as T1, T2, and proton density (PD) imaging, with an emphasis on enhancing tissue class contrast. This is followed by a section on the application of machine learning techniques for processing MRI images to enhance the information content in the context of providing additional information for clinical diagnosis. The last section discusses two classes of conditions that have been studied extensively using MRI technology.

## *1.1 The Historical Development of MRI*

MRI is based on a technology that was established in the mid part of the 20th century - nuclear magnetic resonance (NMR), the discovery of which resulted in several Noble Prizes (notably Felix Bloch and Edward Purcell in Physics, 1952, see http://nobelprize.org/nobel_prizes/physics/laureates/1952/purcell-lecture.pdf for details). NMR relies on the observation that when certain nuclei were placed in a strong static magnetic field, and then exposed to an additional oscillating magnetic field, certain atoms will absorb energy from the oscillating magnetic field, provided the energy level is correct for the particular atom. Once the secondary oscillating magnetic field is turn off, the atoms in the sample will release a packet of energy, which is detected using an appropriately designed receiver. In this way, the sample is probed to determine if a particular atom exists in a sample of material, and in addition, the quantity can be determined as well. This is a very brief overview of NMR, as it s applied to chemical spectroscopy – the determination of the chemical composition of matter. The trick to NMR is to ensure the secondary oscillating magnetic field impart the correct amount of energy into the sample. Typically, the energy is in the radio frequency (RF) part of the spectrum. In order for NMR to work, the energy of the incoming magnetic field must correspond to the precessional frequency of the atoms being probed. Precession – which is a process whereby a rotating charged particle such as a proton rotates on its axis as the Earth does. Each type of proton has its own precessional frequency – termed the Larmor frequency (after the Irish physicist, Sir Joseph Larmor), whom had discovered that there is a quantitative relationship between an applied magnetic field and the precessional frequency of protons.

More specifically, the Larmor frequency quantifies the angular frequency of precession of nuclear spins as a function of the applied magnetic field. Hence, NMR relies on the properties of the nucleus, and this fact is reflected in the name of the technology: NMR – nuclear magnetic Resonance.

The 'nuclear' component was applied because only the nuclei of certain atoms would resonate at a particular frequency when exposed to an externally applied magnetic field. It turns out that atoms with an odd number of protons in their nucleus are able to produce a signal under the conditions deployed in NMR. This property is well known in the Quantum Physics world – and they have assigned a quantum label to it – called S for spin. The term 'magnetic' was used because the phenomenon requires static and oscillating magnetic fields, and lastly, 'resonance' was used because of the frequency of precession (also termed resonance) is dependent partially on the magnitude of the magnetic fields. NMR was initially deployed as an analytical chemistry technique, principally to investigate the atomic composition of compounds. NMR became *the* method of choice for quantitative chemistry, but in addition, served as the basis for a new technology that was named MRI.

In 1973, Paul Lauterbur published a paper entitled 'image formation by induced local interactions; examples employing magnetic resonance,' in the journal Nature (Lauterbur, 1973). This paper initiated the new field of MRI, which Lauterbur termed 'zeugmatography,' from the Greek (*zeugmo*), meaning to join together. He was referring to joining together weak and strong magnetic fields for producing a 2D image, thus extending spectral analysis from 1D (as in NMR spectroscopy) to 2D (and thus earning him the title 'father of MRI'). Although Lauterbur's initial study did not involve living tissue (he imaged 2 test tubes - see Figure 1), his results clearly provided the impetus to apply this technology to other domains - such as imaging the structure of complex materials.

In the 1970's, a number of laboratories were experimenting and refining a separate technology termed computed tomography, sometimes referred to as Computed Axial Tomography (CT or CAT). Tomography, (from the Greek term *tomos*) means to "cut" and computed refers to the way the image was produced using computer based technology. CT as we currently know it was developed by the British engineer Godfrey Hounsfield in 1972, for which he won the Nobel Prize for Physiology or Medicine, 1979. CT is itself a technology that was derived from X-ray technology, first discovered by Wilhelm Roentgen in 1895 (yet another Nobel Prize winner!). CT utilises the imaging power of X-rays, but to acquire more depth in the image required the X-ray device to move (or likewise the object must be rotated) so that the object is imaged from different angles. The pictures from various angles are then combined to form a single image of the object (see Figure 2 for one of the original images from the early 1970's). The significant contribution from CT to the burgeoning MRI technology was the concept of forming a tomogram - a 2D projection of the object in the scanner onto a film. This advance provided greater in plane resolution, which is a requirement for modern day MRI technology. MRI then can be attributed to the confluence of two very different - yet complimentary technologies: NMR provided the analytical power and CT provided the tomographic component. In addition, a number of

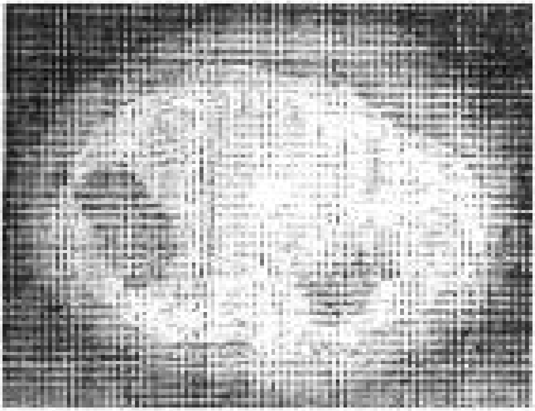


**Fig. 1** This image was one of the first MRI images produced - it was taken by Prof Lauterbur, the father of MRI' - and depicts 2 test tube taken from a superior (top-down) view.

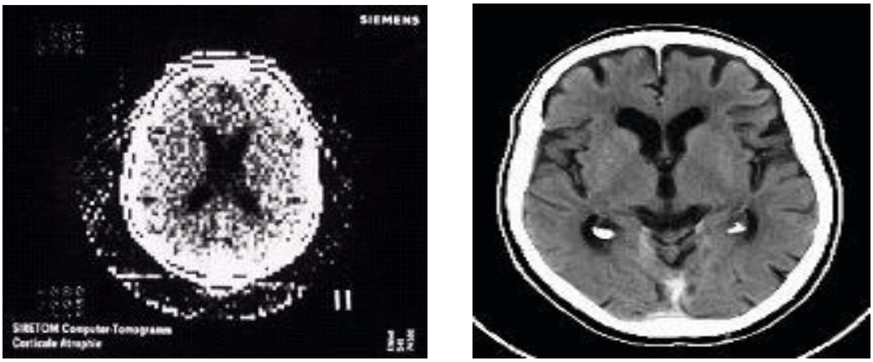


**Fig. 2** An early and modern CT image of an axial slice through a human brain, using state-of-the-art technology at the time, ca. 1975 for the original (on the left, and 2005 for a modern image on the right).

scientists have provided valuable insights and refinements along the course of the development of MRI. The next section presents an overview of MRI technology in its current form.

## 2 The Basics of MRI

MRI provides a high quality image in terms of spatial resolution of an object placed within the scanner (see Figure 3 for a typical medical grade MRI scanner). It is a *tomographic* technique (like CT) that creates a set of images (each essentially in 2D) of a subject when placed in the scanner, in the form of a collection of slices, much like a loaf of bread. When the slices are placed next to each other, and viewed using specialised computer software, the slices form a continuous 3D image of the object. The images contain details about the structural features of the material - in this chapter, the discussion will focus exclusively on imaging the brain.

The brain is a complex structure, containing a variety of tissue types such as cerebrospinal fluid (CSF), grey matter (GM), white matter (WM), vasculature, along with other tissue types that are found in the skull cavity such as muscle, fat, and other minor components. The task of producing a clinically relevant MRI image is to generate an image that provides contrast between the various tissue types of interest, without compromising spatial resolution – clarity. As you will soon discover, there is a delicate balance between the clarity of an image – which is characterised by the amount of signal acquired given some inherent noise level –termed the signal-to-noise (SNR) ratio. Each tissue type will generate a signal when exposed to an appropriate MRI procedure (if for instance we are using the Hydrogen atom as our probe) – the magnitude of which is dependent upon the number of atoms and their unique chemical environment. This is the basis of contrast between tissue types the difficulty is to create an image with maximal contrast between the various tissue types, without compromising SNR. A high resolution image will provide information that will greatly assist a physician in diagnosing and/or determining the extent or the level of progression of a given medical condition.

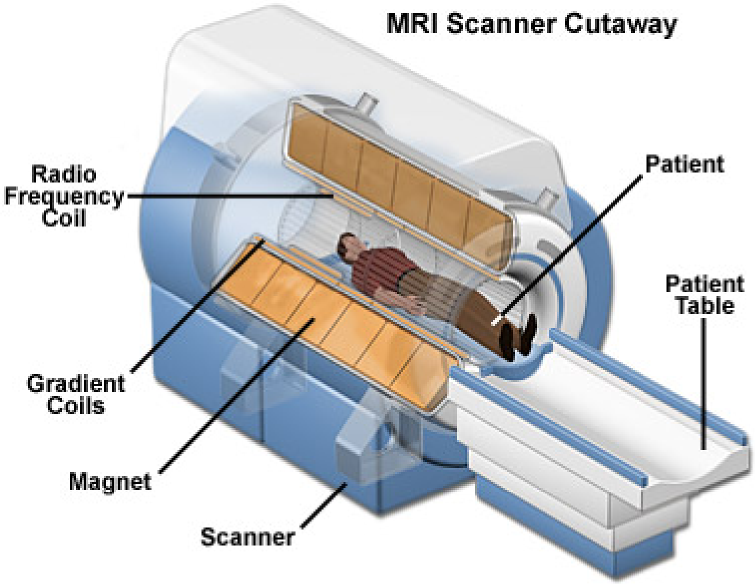


**Fig. 3** A cut-away view of a scanner with a subject inside the scanner bore, illustrating the major features of a medical grade, closed-MRI scanner. Image source: http://www.magnet.fsu.edu/education/tutorials/magnetacademy/mri/images/mri-scanner.jpg

Diseases and medical conditions typically produce structural changes across one or more tissue types in the brain - a classic example is a stroke. A stroke is caused by a reduction of the blood supply to a region of the brain. When this happens, tissue dies, which in turn changes the properties of the tissue in the effected region(s) of the brain. The altered tissue properties can be detected and even quantified with an appropriate type of high resolution MRI scan. As another example, multiple sclerosis (MS), a debilitating disease which causes a reduction in nerve conduction and plaques, can be detected using MRI at two different levels. At one level, lesions can be detected using typical high resolution imaging protocols and in addition, the subtle changes in the structure of white matter can be detected through diffusion weighted imaging techniques (Schmierer et al., 2010). Moreover, the stage of the disease can be quantified fairly accurately, and is typically correlated with overt cognitive and physiological symptoms. There are a wide range of diseases and medical conditions that produce alterations in brain anatomy - both at the macro and micro-structural levels. The role of MRI in medicine is to highlight these changes - how this is done is addressed in the next section.

## *2.1 Overview of the Image Acquisition Process*

There are several components to an MRI system: the scanner hardware itself, with a collection of supra paramagnetic magnets, a radio frequency (RF) receivers/transmitters (also a type of magnet, but one that oscillating), a computer based control center, and a pulse sequence. A pulse sequence is a software base system that allows the radiologists to create very specific types of image scans based on a set of parameters that control the operation of the hardware components. A typical scan takes approximately 10-20 minutes, depending on the purpose of the scan - that is what condition/disease is being investigated. Note that the subject is perfectly safe during this process - no harmful radiation is used, either in the form of radioisotopes or ionizing radiation (as is used in PET/CT scans). Instead, MRI influences the natural behaviour of certain atomic nuclei through the application of magnetic fields to produce images. The material is placed within the scanner, and a large static magnetic field is applied which magnetises susceptible material contained within the scanner. For the discussion in this chapter, the focus will be on using the Hydrogen atom as the probe, as it occupies 60% of the total atoms in a human body (principally in the form of water). The atoms in the subject respond to the static magnetic field (which is on the order of 1.5-3.0 T) by aligning either with or against the applied magnetic field, acting as a collection of tiny magnets themselves. The protons remain in this configuration while the externally applied magnetic field is on – or until another magnetic field is applied. Then a radio frequency (RF) pulse of electromagnetic radiation (this is an oscillating magnetic field from the RF transmitter) is applied to the subject in the scanner very briefly. If the energy is correct (i.e. at the right frequency for Hydrogen atoms), this RF pulse imparts energy into the hydrogen protons, and in the process begin to move against the direction of the externally

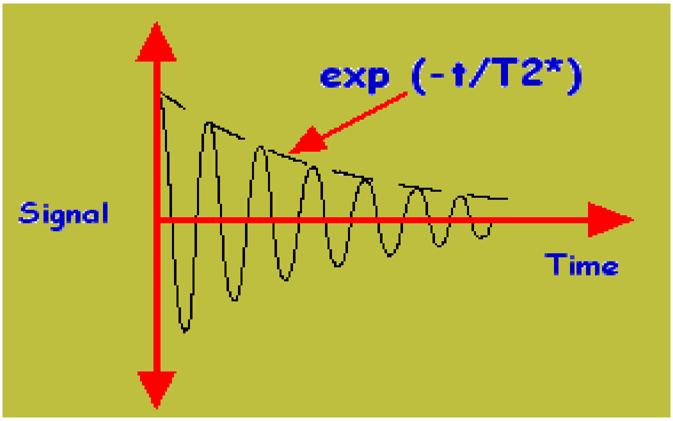


**Fig. 4** An example of a signal termed a free Induction decay (FID), emitted when the RF pulse has been turned off.

applied magnetic field for the duration of the RF pulse. Once the RF pulse is turned off, the displaced magnetised material returns back to alignment with the externally applied magnetic field. When this happens, energy is released which is detected by a receiver (the RF receiver component) placed around the tissue of interest (i.e. the brain). The profile of a signal is illustrated in Figure 4, which depicts an oscillating signal whose amplitude decays with time – that is a periodic signal encased in an exponentially decaying envelope. The energy is in the form of an oscillating magnetic field - which induces an alternating current (AC) in the recording circuitry. The AC current is sampled in order to digitise the input for storage within a computer system. This data is used to form an image, which is typically collected as a series of 'slices' - each representing a section through the region of the body scanned. The numeric data associated with the scan is stored as part of the patient record, which can be viewed off-line using a variety of applications such as MRIcron (see http://www.cabiatl.com/mricro/mricron/install.html). The resulting images are used by medical practitioners to evaluate the status of the patient for diagnostic purposes. Note that these 'films' are essentially copies of the slices printed on specialised paper - available for visual inspection immediately after the scan - a very nice feature. In the next section, this process is described in more detail, so the reader will gain a fuller understanding of the process of image formation.

## *2.2 The Physics behind MRI*

At the heart of any MRI system is the scanner itself, which houses a large magnet. Typical MRI grade magnetic field strengths are measured in units called Tesla, named after the Serbian born scientist Nikola Tesla. The Tesla (abbreviated *T*) is equivalent to 10,000 Gauss (G), the SI unit for magnetic flux density. For perspective, the Earth's gravitational field has a magnetic flux density of 0.5 G, and a typical refrigerator magnet has a value of approximately 50 G. In contrast, a typical medical grade MRI scanner houses a 1.5T magnet - and some research grade scanners can go up to 8T. The magnitude of the magnet (in terms of T) has a significant impact on the quality (in terms of resolution) of the resulting images - much like the quality of a digital camera (resolution) depends on the number of pixels it stores for each image.

There are two basic forms of scanners: open and closed forms. Open form scanners are typically flat bed scanners, where the subject lies down on, much like a bed. This configuration is typically used for subjects that are not comfortable entering into a closed, whole body scanner, which has a tunnel like configuration. For instance, anyone whom is claustrophobic might find the closed configuration uncomfortable, as well as young children, or obese individuals might find it difficult to fit into the bore of the scanner. A drawback of the open configuration is the inability to produce a strong magnetic field around the subject. Therefore, open configuration systems tend to have low field strengths (015-0.5 T), which reduces the resolution of the resulting images. On the other hand, closed forms provide a tube into which the subject is placed, and are typically of sufficient length to house the entire length of an individual. The subject is placed on a gantry which is then moved into the bore of the scanner electronically. This configuration provides a much stronger and a constant magnetic field strength, and hence a higher resolution image set can be obtained.

Magnets found in closed forms are typically superconducting magnets, which need to be cooled to very low temperatures in order to produce a high quality and powerful magnetic flux. These are electromagnets, which are built from coils of superconducting wire. Essentially, a large solenoid, the superconducting wire (typically made from a substance such as niobium-titanium) is wrapped around a core, and the entire housing must be kept at extremely low temperatures (@ 10K) in order for the material to be in a superconducting state. Typically, a substance such as liquid helium is used to maintain the temperature below the critical temperature, which is one of the reasons for the expensive price tag of a medical grade MRI scanner (1-1.5 Millions $).

When a substance (such as a person) is placed inside an MRI scanner, the material elements respond to the magnetic field according to its magnetic susceptibility, which is a measure of how magnetised it becomes. There are three basic classes of objects with respect to magnetic susceptibility: diamagnetic, paramagnetic, and ferromagnetic. Briefly, diamagnetic materials respond by producing a small magnetic field in the opposite direction of the applied magnetic field. In effect, these materials reduce the overall magnet field - these substances are not magnetisable at all (i.e. plastic). Paramagnetic materials are magnetisable, but only for the duration of the magnetic field. Notably, certain metabolic forms of haemoglobin, such as de-oxyhemoglobin are very paramagnetic. Hemosiderin, a breakdown product of blood (as occurs at the end-stage of a hemmorhage) is supra magnetic - much like niobium-titanium, used in the magnetic core of an MRI scanner. Ferromagnetic substances are strongly attracted to a magnetic field - and are permanently magnetise - such as load-stone. Typical hobby magnets are examples of substances belonging to this category. The human body is generally

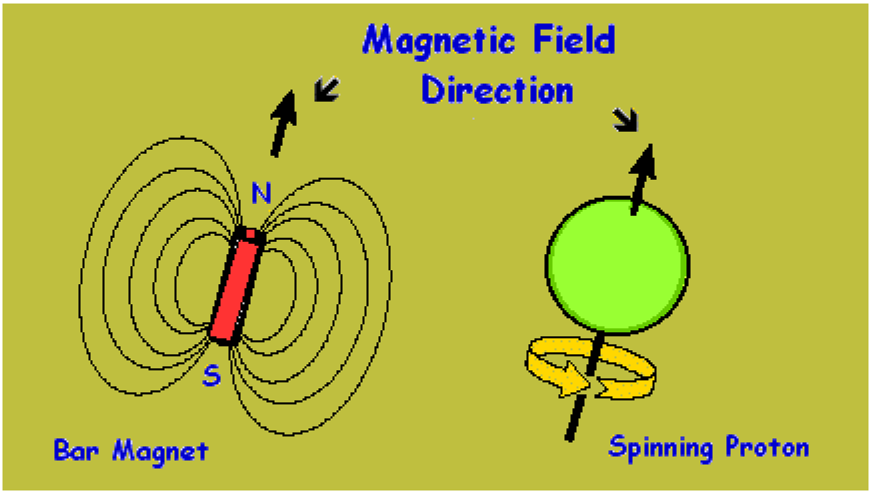


**Fig. 5** The alignment of protons in an external magnetic field.

diamagnetic - for example bulk water found in tissues is diamagnetic. A natural question to ask is: What happens when a body is placed in the scanner?

Since MRI is based on NMR technology, one would expect any response to the injection of energy would be due to interactions within the nucleus, which is indeed the case. As one will remember from chemistry, water, the most abundant substance in the body, consists of 2 H atoms and 1 Oxygen atom. The Hydrogen nucleus consists of a single proton, a spinning particle with a positive charge. It is well known that spinning charges generate a small but measurable magnetic field (as proposed by Felix Bloch). When the magnet is turned on, the protons in water respond by aligning with the applied external magnetic field. From quantum mechanics, it is known that atomic nuclei have a property termed 'spin' - or more accurately a spin quantum number *S*. The S for the hydrogen nucleus - a single proton is 1/2. There are 2S + 1 possible spin states for a given atom, and hence there are 2 spin states for a hydrogen atom. The energy states for the hydrogen atom are designated ± ½. The result of this is that there are two energy states for the hydrogen atom. On state, which we can designate as the '+' energy level is energetically favorable relative to the other possible state (denoted as '-' for example), which is energetically less stable. If there were equal number of protons in each of these two states, they would not produce a net magnetic field, because there would not be any net polarity. In atoms with an odd number of protons, it is not possible to have an equal number of protons populating each energy level – there will always be 1 more in one of the two states. This response to an externally applied magnetic field results in the production of a net magnetic field – termed a magnetic dipole. Each magnet (which is really just a spinning aligned proton is extremely small, that is why you need to use probes that are extremely abundant – such as Hydrogen atoms), The magnitude of the impressed magnetic dipole moment depends on the number of protons that are not paired with protons in the opposite direction. Typically, one finds that there will be approximately one in approximately $10^6$ protons that will align in the higher energy state direction, yielding a very small magnetic dipole. This may not sound like a lot (1 in $10^6$), but one has to remember that in 1 mole of water (18 grams), there are 6.02 x $10^{23}$ atoms - so there is the potential for a lot of protons to align in favor of the higher energy state, producing a large magnetic moment (on the order of $10^{15}$). This process is illustrated in Figure 6, which highlights the imposed net magnetic dipole induced in the material due to the application of the external magnetic field. These protons remain in their configuration as long as the magnetic field is applied (they are diamagnetic), and will revert back to their ground state when the field is removed.

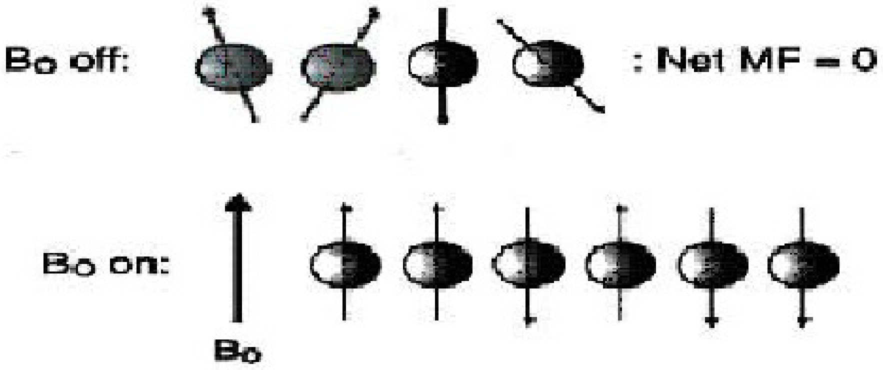


**Fig. 6** Net magnetisation vector when the external magnetic field is turned on.

In addition to forming a magnetic dipole moment aligned in the direction of the applied magnetic field, the protons will precess – revolve around the principle axis of the main external magnetic field, much like a top does when it is set into motion and begins to slow down. Precession is characterised by an angular frequency, which is related to the normal linear frequency according to the following: $\omega = 2\pi f$, where f is the linear frequency. There is a relationship between the angular frequency of precession and the magnitude of the magnetic field - the Larmor equation - $\omega = \gamma B$, where B is the magnitude of the magnetic field, and $\gamma$ is the gyromagnetic constant (42.57 MHz/T for the hydrogen nucleus). The gyromagnetic ratio is a proportionality factor that is unique for each nucleus that contains an odd number of protons: for instance $^{19}F$ has a value for $\gamma$ of 40.0 MHz/T. It must be re-iterated that each atom that is magnetically susceptible has its own unique gyromagnetic constant. This is a fundamental property of behaviour of matter at the atomic level – and provides a way to image different types of atoms with extreme selectivity.

At this point, with the subject in the scanner, we are now ready to extract a collection of signals that will be used to generate the MRI image of the tissue of interest (i.e. the brain). Remember, the magnetic dipole induced by the external field is aligned along the Z-axis, with protons precessing at their Larmor frequency. Everything will remain this way unless the magnetic field experienced by the subject changes. If the magnetic field is changed in a particular way – then a signal can be generated from the subject. This is the subject of the next section.

## *2.3 Generation of an MRI Image*

The MRI signal that is recorded is produced by disturbing protons that have become aligned in the Z-axis - along the direction of the applied magnetic field, termed $B_{0.}$ More specifically, as in NMR, if we impart energy at the right strength, then we can induce resonance within the material (i.e. Hydrogen protons). To do this requires the use of energy in the radio frequency (RF) range - which is extremely low energy, well below that of visible light. The RF transmitter is aligned orthogonally to the external magnetic field – in the X-Y plane. The application of an RF pulse, at the right frequency (the Larmor frequency), causes protons to flip (typically $90^o$) from the Z-axis to the transverse plane (along the X-Y axis) for the duration of the pulse. The protons will only 'flip' into the transverse plane if the imparted energy (in the form of an oscillating magnetic field) is at the correct frequency – note that energy and frequency are related to one anther – so one could speak of the right energy level as well. It is like a child on a swing wish to be pushed to move higher. The effort of pushing will only be effective if you push the child at the right point along their trajectory. Only at this frequency will energy be imparted into the system. When the right level of energy is imparted in MRI, to atoms aligned with an external field, they flip and begin to resonate in the transverse plane. So the protons now have absorbed energy. When this energy is released, the precessing protons begin to fall back down into the longitudinal plane. As they do, they release energy – again in the RF range. Within approximately 1 second, after turning off the RF pulse, all the protons will become realigned with the externally applied magnet field – precessing again in the Z-axis. But, while the protons are relaxing back down to their ground state, they emit energy which can be detected by a piece of hardware called an RF receiver, which is oriented in a certain direction (usually the X-axis). This receiver records the magnetic dipoles when they are aligned in the direction of the receiver coil – in this case the X-axis. This will produce an oscillating magnetic field, which creates an electrical signal (see Faraday's Law of Induction for more details). It is this current that is produced by the protons that form the signal that is used to produce an MRI image. The magnitude of the signal is dependent upon several factors - the principle one being the number of protons in the sample Up to this point, the sample is the subject in the scanner – under the RF transmitter/receiver coils – which is the entire head. Sp

Acquiring high resolution images is not part of the NMR technology itself – the NMR part of MRI only tells us how to acquire a signal from the material substance. To acquire the signal in such a way as to ensure one has sufficient spatial resolution requires a carefully designed scanning protocol. The process requires first of all, flipping the protons in the transverse plane thousands or more times during the course of the scan. This ensures that we have thousands of signals

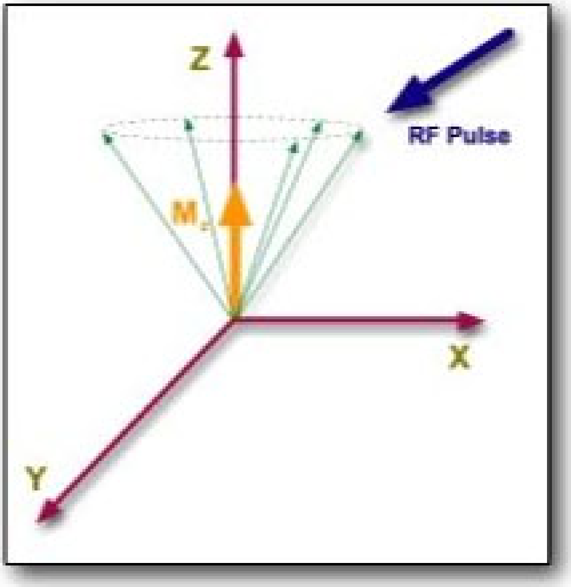


**Fig. 7** Depiction of the precession around the externally applied magnetic field, oriented along the Z-axis, just prior to turning on the RF pulse.

from the subject – and the next step is to ensure that these signals are recorded in a spatially varying manner. The subject matter is partitioned logically into regions termed voxels – essentially a 3D volume of tissue. Each voxel is described by 3 coordinates or axes: x,y,z. The Z-axis is along the axial or the longitudinal plane – along the long part of the body. Perpendicular to this axis is the x-y plane, also termed the transverse plane. In MRI, we typically first take the subject and partition it along with longitudinal plane (along the Z-axis) – must like slicing a loaf of bread. The next stage entails cutting each slice in the X-Y directions – so now we will have a collection of cubes – these are the voxels. What we would like it to do is acquire a signal from each of these voxels – and then if we can put all of the voxels back together the way they were created in the first place, we have a signal map of the entire subject that preserves the spatial relations of the source of the signals – the brain. How this is done – is described in the next sections, starting with the slice selection process.

## *2.4 Slice Selection*

In order to partition the brain (or any object placed in the scanner) into a series of slices, an additional magnetic field is required. This additional magnetic field is typically termed a head-coil when imaging the brain. The head coil provides an additional magnetic field that can be superimposed upon the longitudinal magnetic field. The head coil creates a linear magnetic gradient, which is applied along the

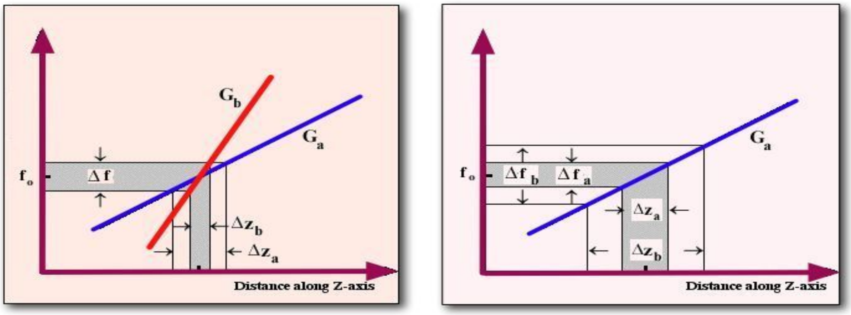


**Fig. 8** The slice select gradient, superimposed on the externally applied magnetic field.

longitudinal plane for axial imaging. The effect of the slice selection gradient as it is called, it to vary the magnetic field in a linear way. For instance, Figure 7 depicts the net magnetic field when the slice selection gradient field has been turned on. At one end of the head, the magnetic field is 1.4 T, and the other end it is 1.6 T, and the central position experiences a magnetic field of 1.5 T. (the static external magnetic field strength). Why is this useful? Well if you remember the Larmor equation, it states that the energy required to produce resonance is proportional to the magnitude of the magnetic field. The energy for a position that experiences a 1.5 T magnetic field strength will require a differing amount of energy for it to resonate then a position that requires 1.51 T and so forth along the edge of the slice select gradient. The gradient is designed to vary linearly along the entire region to be scanned. The energy imparted into the system is termed an RF pulse, which can be made to vary in energy by selecting appropriate imaging parameter values – this is part of the task of the pulse sequence.

Figure 8 depicts the effect of superimposing the slice gradient onto the $B_0$ gradient. The slope of the gradient influences how quickly the gradient changes spatially. What we would like to have is control over how thick the slice is. This is in part determined by the slope of the slice select gradient – if it is shallow, then we can have thicker slices – so the slice thickness is inversely proportional to the slope of the slice select gradient. There is another way to affect the slice thickness – this has to do with the frequency range of the RF pulse that we inject into the sample. For details, see the excellent text by Hashemi and Colleagues (Hashemi et al., 2003) – but very briefly, when we turn on the RF pulse – it can consist of a range of frequencies – all centered on any particular frequency of interest. Typically, the centre frequency corresponds to the central frequency of the slice that we are trying to excite. Therefore, the range of frequencies in the RF pulse – termed the bandwidth will ultimately determine the slice thickness, given a constant slice select gradient. That is, an RF pulse with a wide bandwidth will excite more of the linear space along the gradient and hence produce a larger slice.

Now, varying the RF pulse centre frequency allows us to move along the slice select gradient, each time flipping on a range of tissue into the transverse plane. This is effect has produced our slices – and has extracted one of the dimensions for our voxel – the Z-axis. If we record from a slice, we will receive signals from the entire slice – which though better than the entire brain, is not enough yet. What we need to do is to also partition each slice into the X-Y planes – obtain a signal from each of these sections – at each slice – and then we have a true 3D measurement. This process of extracting information from a slice is termed *spatial encoding*, and is described in the next section.

## 2.5 Spatial Encoding

Spatial encoding refers to the process of extracting signals from specific locations within the subject. Ultimately, what is expected from MRI is a high resolution map of the various tissue components of the subject. To produce a 3D map of the subject, the material is first divided into a series of slices - much like a loaf of bread. Typically, when imaging the brain, the slices are arranged in the axial (longitudinal) plane - that is along the long axis of the body. Typically, the slices are approximately 1-5 mm thick, covering the entire head. Each slice in turn can be examined in both the X-Y directions, yielding a 3D volumetric picture of the proton distribution within the brain. This process effectively partitions the region being scanned (e.g. the brain) into a collection of 3D volume elements, termed voxels. The final image produced is the magnitude of the signal, reported as a single scalar value, for each voxel. As long as the voxel topology is preserved during the process of creating the image, the final image will reflect the signal distribution of the scanned material - i.e. the subject's brain.

After the application of the slice select gradient, the subject (e.g. the brain) has been partitioned into a series of slices, thereby providing spatial information along the axial direction - along the z-axis. This step will produce a series of images (equal to the number of slices), each of which will contain the signal from all of the protons in the slice. Remember, the Larmor frequency of the incoming RF pulse must match the Larmor frequency of the tissue. The MRI scan manipulates the Larmor frequency of tissue by the careful placement and setting of magnetic field gradients. With the application of slice selection, the MRI signal will reflect the properties of the entire slice - which, though better than a signal from the entire brain, still does not provide the required spatial resolution suitable for clinical diagnosis. The last stage towards this goal is termed *spatial encoding*, which provides in-plane (within a slice) details along the X-Y directions. In combination with the slice select stage, the tissue is now imaged at the highest level of resolution, at the voxel level (with 3D coordinates: X,Y,Z).

Spatial encoding is produced by the application of two additional gradients: a phase and a frequency encoding gradient. The phase encoding gradient is used to alter the phase of voxels in the Y-direction, and the frequency encoding gradient is used to change the precessing frequency in the X-direction. The process of spatial encoding is depicted n Figure 9. Without spatial encoding, the signal acquired after slice selection will be the sum of the signals from the entire slice. As an example, consider a 4x4 matrix representation of a slice, and the associated magnitude of the signal in each region wave. What we would like to do is partition the slice (in this example into 4 columns and 4 rows), and be able to acquire a signal from each of the 16 subsections of the slice. Briefly, this is accomplished by applying two additional gradients - a phase encoding gradient which is applied along the Y-direction, and a frequency encoding gradient, applied in the X-direction. The purpose of the phase encoding gradient is to alter the phase of the protons along the Y-direction. This is accomplished by applying a gradient (magnetic field) along the Y-direction, in a manner very similar to that for slice selection. Each row will be exposed to a different magnetic field (except for the central row which will remain unchanged) for the same reason as in the slice select gradient – it is the zero centre of the applied gradient. This is turn alters the Larmor frequency of the material in each of the rows. Now the protons will change phase across the rows, because they will each be disturbed by a slightly different magnetic field, which causes the precession to change its phase. The last stage requires encoding along the X-axis, which is across the columns. This is done after the phase encoding phase, and is implemented using a frequency shifting operation. That is a frequency encoding gradient (sometimes called the 'Read Out' gradient – because t is turned on when we record a signal) is applied, again in a manner identical to the slice select and the phase encoding gradients. Now the additional applied gradient induces a change in the magnetic field experience by the slice along the X-axis. Again, those columns to the left of the central column will experience a lower gradient and those to the right will experience higher gradient. The thickness along each dimension is again dependent upon the bandwidth and the corresponding Larmor frequency present n the applied gradients. The frequency encoding gradient changes the angular frequency of the spinning protons because each column will be experiencing a different externally applied magnetic field – and hence will spin at a different rate. The recording hardware records signals with a particular phase and a particular angular frequency – so each element in the grid will be detected as a separate signal – this is the concept of spatial encoding.

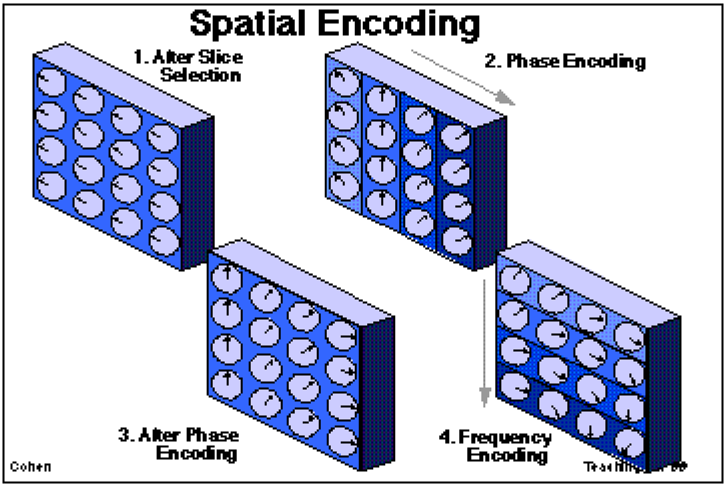


**Fig. 9** A depiction of the process of spatial encoding, highlighting the phase (along the Y-axis) and the frequency encoding steps.

Note that to implement spatial encoding requires that we apply additional magnets (RF pulses) at certain energies and in certain temporal and arrangements. The sequence of gradients is: slice select, then turn of the phase encoding gradient briefly, turn it off, and then turn on the frequency encoding gradient – then record the signal. This is the basic idea behind a pulse sequence, which defines which gradients are turned on, at what energies, and at what times. We now have the concept of a voxel – a 3D volume of tissue from which we can extract a signal from. Note that typical matrices are much larger than the example provided here - typical values are 256 x 256 (that is 256 phase and frequency encoding gradients are applied). The size of the matrix ultimately determines the resolution of the object, for a given volume. Typical sizes of voxels in high resolution are 2x2x2 mm - providing a very high level of resolution that is suitable for medical diagnosis. Figure 10 presents a more realistic image of a slice with respect to spatial encoding and the corresponding axes labelled in the imaging domain.

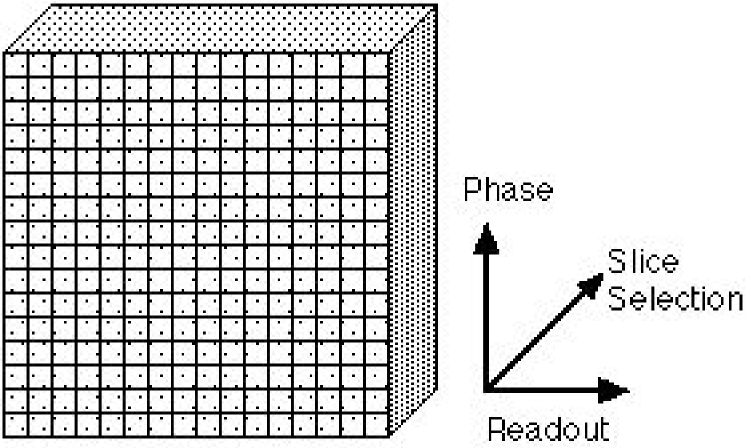


**Fig. 10** A hypothetical slice that has been partitioned into volxels and the corresponding axes which has been labelled with their corresponding MRI encoding labels.

We now have a mechanism for partitioning a volume of tissue into a series of small voxels - each producing a signal that is recorded by the MRI scanner, and used to construct a 3D map of the tissue for visual inspection. The question remains - how does the neuroanatomy map onto a signal? Why do different parts of the brain yield voxels with differing intensity values? It is well known that the brain is composed of a variety of tissue types: the principle cerebral spinal fluid (CSF), white matter (WM), grey matter (GM), vasculature, surrounded by fat, muscle, and bone predominantly. Each of these tissues types contains a large number of protons, but they exists in different chemical configurations. For instance, the CSF contains a large amount of bulk water, and so the protons are able to move freely in the CSF. White matter on the other hand has protons associated primarily in the hydrocarbon chains that make up the myelin sheath. Different tissue classes therefore present protons in slightly differing configurations. These subtle differences will produce characteristic but unique signals which are detectable using MRI. Provided the spatial resolution is sufficient, a voxel may contain purely white matter, grey matter, or other tissue classes. This will result in an image that contains fine structural detail of the brain. But in many cases, the tissue contrast is less then desirable - for instance a voxel may contain white matter and grey matter, yielding a mixture that is different from either tissue class. Tissue boundaries often present difficulties with respect to image clarity because they tend to present abrupt changes in proton densities. These issues are partially addressed by a small number of scanning parameters, which will be discussed next.

## 3 Tissue Contrast

There are several imaging modalities that were developed to enhance tissue contrast - the most popular ones are termed T1, T2, and PD. These modalities, termed weightings in the literature, reflect the physical properties of protons with respect to how they respond to an applied magnetic field. For instance, T1 is a measure of the time it takes for protons to relax back down to the $B_0$ gradient when the RF pulse is turned off. Different tissue classes will respond with their own characteristic T1 value - which is typically on the order of 500 ms for white matter (see table 1 for full details). Each tissue class has a different T1 value, and therefore, the pulse sequence can be designed to highlight the differences based on the unique set of T1 values each tissue class exhibits. When one deploys such a pulse sequence, it is termed a T1-weighted image. Likewise, each tissue class has a unique T2 value, which represents the loss of signal in the transverse plane after protons have been flipped into that plane. Almost immediately, protons begin to lose phase coherence and the signal decays in proportion to the extent of phase coherence. As for the T1 time, each tissue class has a characteristic T2 value as well. The pulse sequence can be designed to highlight differences in T2 values for different tissue classes, resulting in a T2-weighted image. Lastly, a proton density (PD) image can be formed that *reduces* the T1 and T2 weighting of a voxel, relying solely on the number of protons contained within the voxel. This is termed a PD-weighted image. each imaging modality provides a slightly different picture of the brain, as is illustrated in Figure 13.

The T1 of a tissue reflects the rate at which it reverts back to the ground state after being flipped into the transverse plane via an RF pulse at the required Larmor frequency. Figure 11 depicts in graphical form the trajectory that occurs during this process, which can be described by the following equation:

$$S = (1\text{-}e^{-t/T1}) \qquad (1)$$

where 'S' is the recorded signal, 't' is time, and 'T1' is the longitudinal relaxation time.

The trajectory is an increasing exponential, which ranges from zero (0) immediately when the RF pulse is turned off, up to a maximum of 1 after some

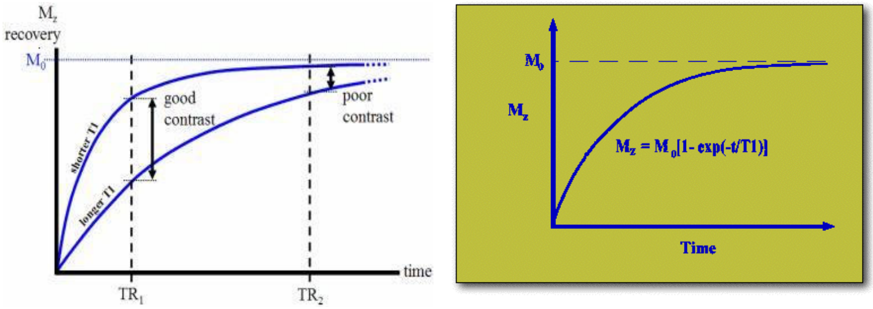


**Fig. 11** The left panel indicates the temporal profile of tow tissues with differing T1 time constants. The right hand panel illustrates the genera process of longitudinal relaxation, the basis for T1-weighted imaging.

time. More specifically, 63% of the signal will be recovered in the z-axis after a time T1. By 4-5 T1 times, the signal is essentially completely recovered along the z-axis. In figure 12, we see the T1 profile of 2 hypothetical tissues. In order to weight an image for T1, what we want is to choose a time that maximises the signal differences between the two tissues classes. In figure 12, we see that classes A and B are most distinguishable at short time periods - this region provides the greatest contrast between these two tissues classes. As the time increases, beyond several t1 times, the signal between the two classes is very similar, and we would lose contrast as a result. So the T1 is a time - and is calculated as the time taken for 63% of the signal to recover in the z-axis, and is characteristic for each tissue class. By exploiting these differences, we can weight an image to produce enhanced contrast between tissues classes based on this feature. This is termed a T1-weighted image.

Another popular way to weight an image, termed a T2 weighting, relies on the time over which protons lose their magnetisation in the transverse plane. When protons are flipped by the RF pulse, protons are precessing about the x-y plane, with no signal in the z-axis. After the RF pulse is turned off, the protons, while still in the x-y plane begin to interact with one another, reducing the strength of the signal they initially produced in the transverse plane. Very quickly, the protons become desynchronised, resulting in an exponential loss of signal in the transverse plane.

$$S = e^{-t/T2} \tag{2}$$

where 'S' is the recorded signal, 't' is time, and 'T2' is the transverse relaxation time.

The time at which the signal in the transverse plane decreases to 63% of its original value is termed the T2 relaxation time. Its trajectory is depicted in Figure 12. Typically, T2 times tend to be much shorter than T1 times, with typical values somewhere around 100 ms. Again, just like the T1, major tissue classes have characteristic T2 values, which can be used to create a T2-weighted image.

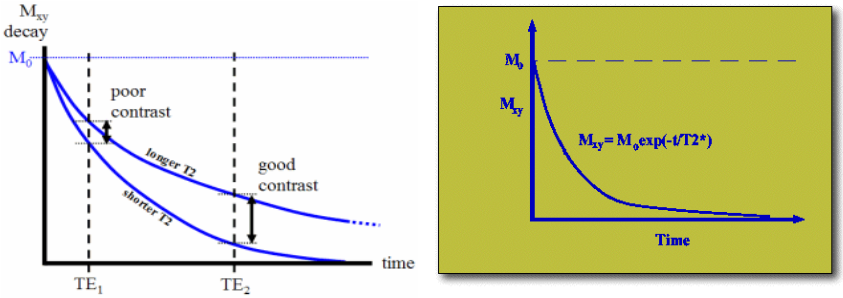


**Fig. 12** The left panel indicates the temporal profile of tow tissues with differing T2 time constants. The right hand panel illustrates the general process of transverse (or spin-spin interactions) relaxation, the basis for T2-weighted imaging.

A proton density (PD) weighted image is designed to minimise the T1 and T2 weightings of tissues. A long T1 and a short T2 value will eliminate the T1 and T2 contributions to the image, yielding a purely proton density weighted image. The equation that describes PD is the following:

$$S = N(H)*(1-e^{-t/T1})*e^{-t/T2} \tag{3}$$

where N(H) is the proton density, and the other terms are for T1 and T2 as per above.

That is, if you look at the equations for T1 and T2, you will note that as time (t) is very long, the T1 term approaches one, effectively removing the T1 contribution to the signal. If the T2 value is short (i.e. small), the T2 term also disappears (goes to 1), and we are only left with the N(H) term, which is a measure of the number of protons in a voxel - the proton density. Figure 13 displays a set of three images from different axial positions using the three principal imaging modalities of T1, T2, and PD. One should note that differences in the intensity/brightness of the three major tissue classes (CSF, WM, and GM) across each modality.

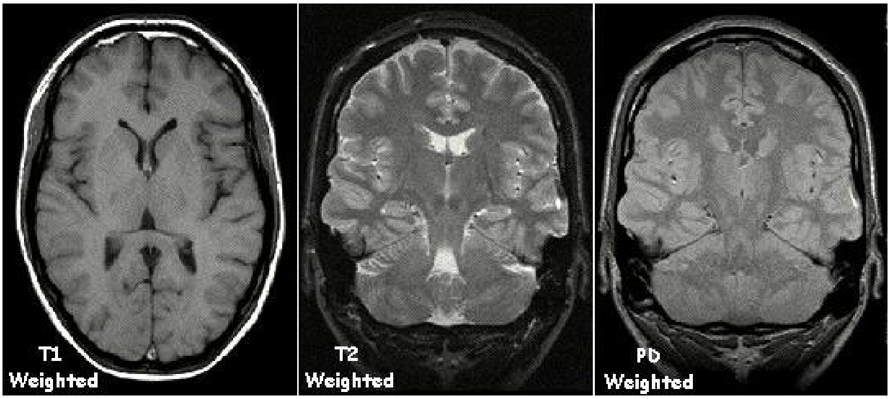


**Fig. 13** A set of T1, T2, and PD-weighted images taken from the same scanner, at differing axial locations from a healthy subject.

These terms are intrinsic factors of tissue - how can they be used to enhance tissue contrast? There are two scanner parameters - TR and TE that can be set up by the scanning protocol. TR stands for repetition time, which is a measure of time between turning on successive RF pulses. when the RF pulse is turned on, we are ready to record a signal. If the TR is set at approximately the T1 value, then we will have a T1-weighted image. So the T1 values associated with the intrinsic tissue properties are used by the scanning protocol to produce the appropriately weighted image. Likewise, the TE - which is the time to echo, influences the T2 property of tissues. The TE is the time after the RF pulse has been turned off and the signal is recorded. Figure 12 depicts two theoretical tissue classes, A and B, and their decay profile as a function of time (that is, their T2 profiles). One can see that the differences between the two tissue classes increase over time. Therefore, if one wishes to T2-weight an image, the time of measurement (the 't') in equation 2 should be fairly long - longer than the T2 value for the tissue of interest. Table 1 summarises the values of TR and TE required to produce images of various weightings (T1, T2, and PD). It should be noted that there are a variety of additional parameters that can be tuned during the scanning process to create a wide variety of pulse sequences, which in turn elicit very detailed and specific types of contrast. It is beyond the scope of this chapter to discuss these issues, but the interested reader is directed to the following sources (Hashemi et al, 2003).

## 4 Image Processing of MRI

After an image volume has been collected, one can view the volume to determine if there are any abnormalities which can be used for diagnostic purposes directly. In addition to such a qualitative analysis, one would like to have a quantitative estimate of the extent of the changes, in terms of the number of voxels involved, across all tissue classes. Other measures, such as the diffusion coefficient of water in various directions can be acquired, providing details on the underlying micro structure of brain tissue can be acquired fairly routinely. The processing stages applied to MRI volume sets is quite extensive, in what follows is a very brief survey of some typical approaches that have been applied. The ultimate goal of these approaches is to acquire detailed information that may not be obvious from viewing a series of slices from a brain volume - which can be a tedious and error prone task. In addition, certain features of brain injury, such as diffusion changes can not be viewed on classical imaging modalities (T1, T2, PD) as they are not designed for estimating the relevant parameters. The text by Tofts provides a very comprehensive survey of this interesting approach to MRI analysis (Tofts, 2003).

**Table 1** Summary of the relative durations of TR (repetition time) and Te (echo time) on the MRI weighting.

| | T1-weighted | T2-weighted | PD |
|---|---|---|---|
| TR | Short | Long | Long |
| TE | Short | Long | Short |

## *4.1 Edge Detection*

The ability to demarcate the borders of tissues and other structures is one of the first image processing steps. For instance, separating the head from an image is a necessary step in performing many quantitative tasks - see Figure 15 for an example of segmentation of the skull from the background. Detecting edges typically relies on the continuity of voxels with respect to their magnitude. For instance, an abrupt change in the magnitude of a voxel may indicate an edge - to be certain, one then searches locally, looking to see if there are other voxels with a similar magnitude. These voxels are then examined to see if there is continuity between them - that they are more or less contiguous. If so, then it is fairly certain that an edge has been detected. This approach can be used to separate the skull from the background in an image.

There are a number of edge detecting operators that have been applied to MRI image datasets. Roberts introduced one of the earliest edge detection algorithms (1965), named after him, in which he convolved the image with two spatial operators, which were designed to enhance any edges that were orthogonal to one another. This approach was susceptible to noise, and so other approaches have been advanced since this time. The Prewitt and Sobel edge detection algorithms are similar in their simplicity to the Robert's approach, but are less susceptible to noise, and produce superior edge detection. The Sobel and Prewitt methods solved the issues associated with noise, but did not consider issues of scale This is largely the result of the convolution operators applied - which were typically 3x3 matrices, suitable for features that are on the voxel level, but fail to take into account edges that exist over variable widths. Two approaches have been proposed to handle the issue of scale: the Marr-Hildreth and the Canny approaches. Essentially, these two approaches are similar: they both performed a smoothing operation using a Gaussian template which includes a scale parameter. In the Canny approach, the derivative was used to find the zero crossing point, which would indicate and edge. The Marr-Hildreth approach took the double derivative to find an edge. The interested reader should consult any text on image processing (see Morris, 2004).

Edge detection has largely been subsumed under the need for image segmentation - which not only includes edges - but also the material within an closed edge border, which may differ from the voxels at the edges. In the next section, a few examples of image segmentation are presented to give you a feel for the process and show you what sort of results can be obtained.

## *4.2 Image Segmentation*

We now have discussed how to acquire a high resolution, 3D volumetric image of the brain, with maximal contrast between tissue classes. How is the data to be used for diagnostic purposes? In one simple sense, the images could be viewed manually by a trained radiologists/clinician, visually inspecting the image for subtle cues which may provide the required details for a diagnosis - or to confirm

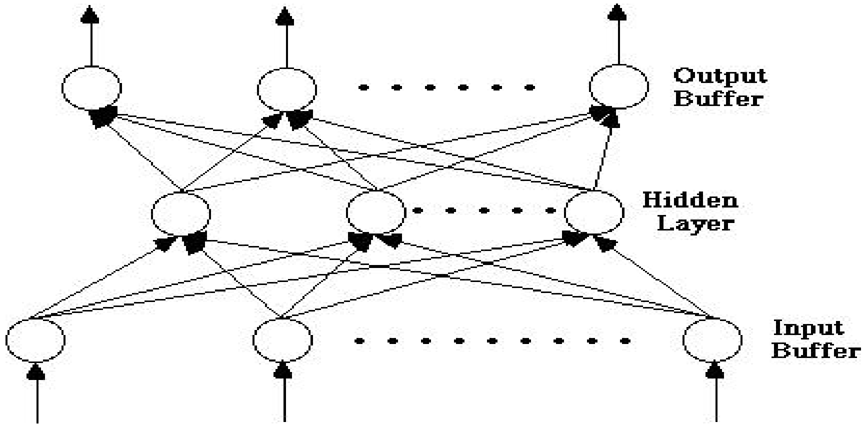


**Fig. 14** An example of a multi-layer feedforward neural network architecture, consisting of 2 layers (in addition to the inputs which many do not consider to be a proper layer).

a suspected diagnosis. Many conditions, such as stroke, or Alzheimer's yield very obvious structural changes in the brain and are easy to detect. Other conditions, such as multiple sclerosis (MS), produce more subtle changes that are not so obvious. In addition to the qualitative aspects a disease presents on the structural integrity of tissue, one may also wish to quantify the extent of the changes. This typically can not be performed by visual inspection, and may require the use of purpose built computer based applications.

The ability to quantify tissue classes is a very important and by now routine step towards quantitative analysis of MRI data. The relative amounts of CSF, WM, and GM is fairly constant across individuals. Any significant deviation from standard values may indicate a pathological condition. For instance, various dementias reveal significant reductions in the amount of GM. Further, an increase in the number of non-standard tissue class voxels may indicate a lesion - as lesion voxels may yield values that are not consistent with those associated with typical tissue class values. In addition, it would be very useful to have a quantitative measure of the extent of a lesion, whether it was produced by MS or a tumour. A first step towards this quantitative analysis relies on the use of image segmentation techniques.

Image segmentation refers to labelling each voxel with its corresponding tissue class - it is essentially a classification task. If every voxel can be accurately labelled, then one can simply simply count or otherwise record the entire volume occupied by that class. But typically, life is not so easy - many voxels may contain signals that were generated by two classes - and hence the recorded value is a linear sum of the relative contributions from each class. This is especially true for boundary regions - such as might occur with CS and white matter or white matter and grey matter. This effect is termed the 'partial volume effect' - PVE, and will reduce the classification accuracy - or at the very least, make it somewhat more difficult than it could be. Barring the PVE effect, how can one approach image segmentation?

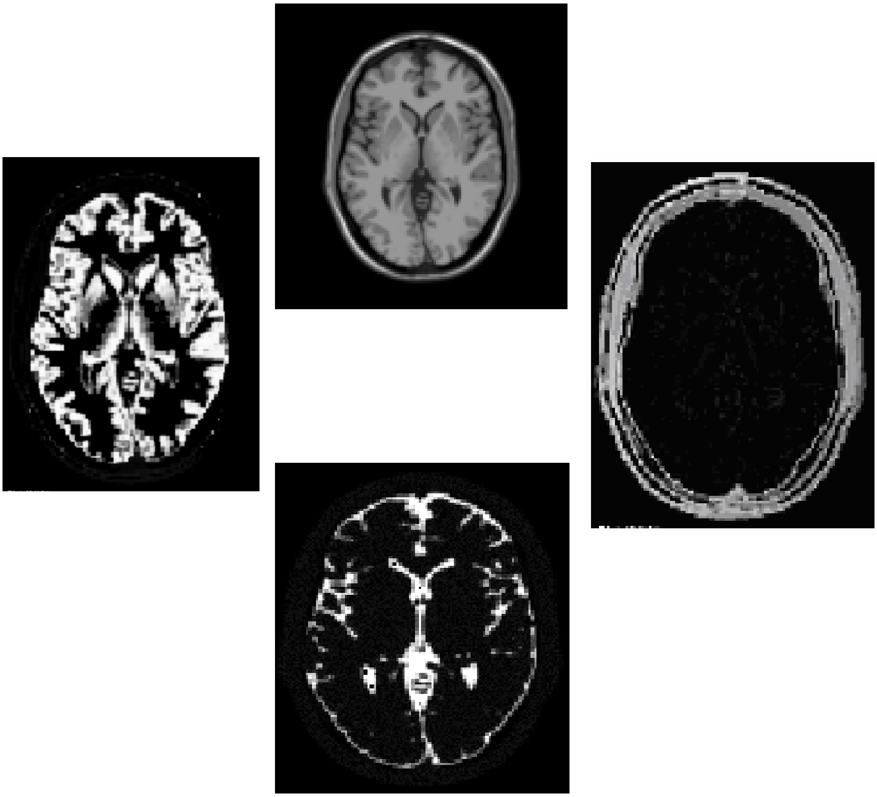


**Fig. 15** A example of tissue segmentation, where the top image is the original T2-weighted image, the left most image is GM, the bottom image is CSF, and the right-most image is the skull. From Li & Chi, 2005.

There are two basic methods for image segmentation - one termed supervised and the other unsupervised. In the supervised method, one must select a seed voxel - that is, if one wishes to quantify grey matter, then one must highlight a voxel (or a collection of voxels) that are definitely GM according to some known criteria. The criteria is typically expert neuroanatomical/medical knowledge, but could be further validated by the magnitude of the signal, which is typically characteristic for a scanning modality (i.e. T1 or T2). Care must be taken with this approach though, as different scanners may yield different magnitudes - via differences in scanner gain for example. One may therefore have to normalise the data - re-scale it across the range of values that are present in the image set to solve this problem. Once a seed has been selected, the software will then deploy a variety of methods to find other voxels with the same properties, and label them accordingly. This process is repeated for all tissue types of interest.

Segmentation using an unsupervised learning approach by definition is achieved with no or minimal human intervention. There are no labels attached to a training set - instead the algorithm itself provides labels to the entire set of voxels in the images. A majority of unsupervised approaches used in image segmentation utilise some form of data clustering, where the information content of voxels is used in the classification process. As an example, the C-means algorithm has been applied with considerable success.

An example of a popular supervised technique deployed in image segmentation relies on the use of neural networks. A neural network is a computational device that was inspired by the computational capacities of neurons - it is a biologically inspired model of computation. The historical developments of neural networks started in 1943, with the formulation of the McCulloch and Pitts neuron. Since that time, the field of neural networks has grown extensively, with applications in engineering, aviation, and medicine to name a few. Neural networks can be used in many ways - but for image segmentation, it is typically utilised as a classification device. Neural networks contain two principle elements: an architecture and a learning rule. The architecture consists of one or more processing nodes, which are connected to one another in a particular way. The nodes are connected to each other via a weight - which quantifies the impact one node has on another node. Much like neurons in biological systems, the weights reflect the gain of the response to the inputs from other neurons. The learning rule is an algorithm that sets the weights between neurons in a manner that facilitates the classification task. The way the system works can be summarised in the following way: examples of mappings are presented to the network - in the form of an input-output relationship. When a particular input is presented - the network should produce a specific output. In order to facilitate this mapping, the network is trained using a subset of the objects it is to be trained to classify - this is termed the training set. Each element in the training set is presented one at a time, usually drawn randomly from a sample, and the output of the network is produced. The output is compared with what the actual output value should be - this is what makes this a supervised approach. If the output doesn't match the true output, then the network needs to adjust itself. This self-adjustment is accomplished through alterations of the weights - the connections between processing nodes. The weights are adjusted in proportion to the errors associated with the produced output, relative to the desired output. This process is repeated until the errors produced after presenting the entire training set is below a threshold, such as 5%, then the network is considered properly trained. Once the network has been trained, it then can be used to classify unknown examples. How well it does this is a measure of how well the network was able to generalise. In the current context - samples of known voxels from all tissue classes would be used for training purposes. Once the network has been trained, it can be used to classify the rest of the voxels.

A clustering type algorithm is a classic example of an unsupervised segmentation approach. Clustering is an approach that examines each voxel in the image - or in the region of interest (ROI) selected by a person - and assign it to a particular class. The class labels are initially seeded by a human operator - there will be a class label for each tissue class one wishes to classify (these are called the cluster centers). The algorithm will then go through and look at each voxel, one at a time, and decides which cluster center it is closest to. The measure of

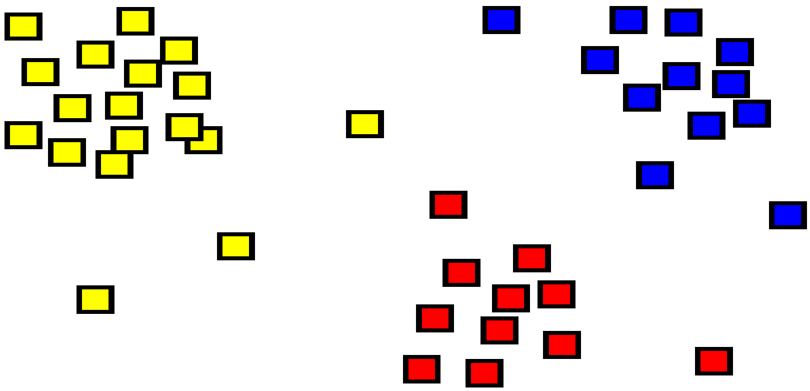


**Fig. 16** An example of the concept of clustering, which contains 3 separate clusters, which are separated spatially by the values of the features.

similarity is typically based on some measure in magnitude space. - for instance if one class label has a magnitude of 650 and another 1,300, then if an unlabelled voxel has a magnitude of 1,250, then one would calculate the distance to each of the classes and assign it the label of the closest matching class (the one corresponding to the value of 1300). Typically the Euclidean distance is used for the distance metric, though there are many other metrics that have been deployed (see 4.3 for details). Then, one re-calculates the new cluster centers for each class based on the average of the voxels with a given label. In effect, what we have done is to calculate a mean value for each class after each voxel has been associated with a cluster. This step is performed for each class, until all of the voxels have been assigned a class label, or until some other convergence criteria has been met (such as the change in the position of the cluster centres). This approach is termed *c-means*, where the 'c' refers to the number of clusters expected in the data.

Regardless of the segmentation approach deployed, the examples used for training (or the seed values for a clustering approach) are extremely important in terms of the overall classification accuracy. Even when seeds are properly chosen, there will be a significant number of voxels that occur at tissue boundaries, which may exhibit a significant partial volume effect. The extent of PVE will depend on the resolution of the image - which is reflected in the size of the voxels. Large voxels tend to enhance the signal to noise ratio (SNR), because more protons exist within the voxel. Unfortunately, large voxels reduce the spatial resolution - one needs to find a balance between SNR and resolution - which depends on the task at hand. One way to estimate the extent of this effect is to produce a simple histogram of the data. Most image viewers provide a function that will display a histogram of a complete slice. The histogram should display a series of peaks that correspond to the various tissue classes in the slice. The overlap between peaks is the area of concern - the larger this region is, the more difficult the segmentation task will be (the overlapping regions are indicative of the extent of PVE contribution to the data). To date, there is no definitive method for removing the PVE - at best one can try to model it using some sort of fuzzy approach, such as fuzzy c-means. The interested reader is directed to Ballester et al., 2002, for more details. Another difficulty associated with segmentation is the effect of intensity nonuniformity (INU), which causes a spatial blurring over a region of a slice (potentially across all slices). It is typically caused by imperfections in the gradient coils, and generates a systematic error across slices. If the segmentation depends on the absolute magnitude of the voxels - which the above mentioned approaches do, then care must be taken when selecting seeds or training exemplars. Typically, a histogram of a large ROI will provide some measure of the stability of the signal - indicating the existence of an INU effect. An approach to reducing its effect if present is to perform a smoothing operation, which will blend the subtle changes across the tissue, reducing the likelihood that a new class will be produced from the region containing INU effects.

It should be noted that most segmentation approaches operate on a per slice basis. Once completed, the slices can be collated and registered together to form a true 3D image (a volume). Depending on the software available, the number of voxels per class of interest can be calculated, providing an estimate of the percentage of the image space it occupies. Further, some applications are then able to 'remove' a tissue class or any arbitrarily selected ROI from the image set (slice/volume). These ROIs can then be displayed and manipulated by magnification and/or rotation operations to provide a very interactive capability for the user.

## *4.3 Voxel-Based Morphometry*

Many investigators have proposed the idea of detecting changes in brain structure by attempting to directly compare a brain with suspected structural pathology against a normal brain. The basic idea is to map the two brains onto a common frame of reference - referred to as a common stereotactic space, which requires the brains are matched up (registered) along all of the major brain landmarks. This is typically followed by a segmentation algorithm, which could be used to extract voxels according to tissue classes. Then direct comparisons between normal and suspected pathological brains can be made at the voxel level. This approach is designed to provide an unbiased and comprehensive assessment of anatomical differences throughout the brain (Ashburner & Friston, 2000). There are a variety of software packages that implement VBM, arguably the most popular being the SPM (Statistical Parametric Mapping) package, available as an add-on toolbox for Matlab (see http://www.fil.ion.ac.uk for the SPM homepage). From a more practical perspective, the steps involved in VBM analysis can are: spatial normalisation of the images (across all slices) and subjects to a common stereotactic space, extracting the tissue class of interest (i.e. grey matter) from the normalised images, smoothing, and then a statistical analysis to compare the brains from the subjects.

The spatial normalisation step is required to account for differences in brain shapes, sizes, and volumes. The average human brain has a typical volume of approximately 1,300cc, but this is highly dependent on gender, age, and ethnic origin. A study of 46 adults of European descent yielded a mean brain volume of 1274 cc for men (range 1053-1499cc) and a mean of 1131 cc (range 975-1398cc) for women (Cosgrove et al., 2007). This inherent variability requires some form of registration and scaling so that the brains are compared on an absolute as opposed to a relative basis. To perform the normalisation process requires the use of a template - which is considered to be an 'ideal' brain. A widely used brain template is available for use produced by the Montreal Neurological Institute (MNI) brain Imaging Centre, which has produced a high resolution brain (segmented at 1cc sized voxels), which was derived from the average of 152 healthy individuals scanned multiple times using high resolution MRI across a variety of pulse sequences (T1, T2, PD). Note the SPM package incorporates some version of the MNI brain (as their gold standard) for registration purposes.

In addition, there is a commonly used stereotactic brain atlas, developed by Talairach and Tournoux - typically referred to as mapping into the Tailarach-space (Talairach & Tournoux, 1988) The Talairach-space refers to stereotactic coordinates (3D) of structures within the brain, that can be used for registration of images - and allows for a common point of reference. A set of potentially invariant reference points are utilised - the anterior and posterior commissure lines are used - and a horizontal line is drawn through them. This line traverses the midline of the brain, and hence defines a coordinate system. Distances for a given brain structure is reported in terms of the distance (Talairach distance) to the anterior commissure point. This coordinate system facilitates the image registration process by providing landmarks that much be matched up during the process of mapping two brains together - the purpose of registration.

In order to match up two different brains, regions must be shifted through a variety of spatial transformations, until the overlap between the two brains is maximal (or the error is minimal). A variety of translation protocols have been deployed, but typically can be classed into either a linear transformation approach or a non-rigid or elastic approach. Very generally, the linear transformations include translations, rotations, scaling, and other affine transformation to register two volumes together. Non-rigid registration deploys a local warping of structure in order to produce the required alignment. Typical approaches utilise various localised geometrical operations such as surface splines and deformation models. There are a variety of techniques that have been utilised for this process, the details of which would take up a single tome in its own right, but the interested reader should consult (Ashburner & Friston, 2000, Mechelli et al., 2005) for more details.

Once the spatial normalisation process has been completed, the volumes are segmented using a variety of techniques – as mentioned in section 4.2 for some examples. The ultimate purpose here is to extract either the argy matter or white matter from the volumes for subsequent quantitative analysis. Typically, the tissue segments are smoothed, which helps reduce the magnitude of outlier voxels - by shifting their values towards the mean for the tissue class, which in turn adds statistical power to the subsequent stage of voxel labelling. In addition, smoothing may help compensate for the inevitable errors that may result from the registration process. Once this step has been completed, the volumes are ready to be analysed quantitatively, on a voxel-by-voxel basis.

Typically, a statistical approach is deployed to perform the voxel-based analysis - yielding statistical parametric maps as an example approach. The basic approach deploys standard statistical measures such as t-tests and F tests between subjects across various ROIs or volumes of interest (VOIs). The final result is a quantitative methodology for comparing subjects belonging to different categories - i.e. normal versus disease. with this technique, one can identify and quantify differences in anatomical structure between the two different groups, providing a basis for further examination of the subject. In addition, this approach can be used to map disease progression or the effectiveness of treatment for a given medical condition that can be mapped using MRI. Note that there are several other approaches besides VBM for quantifying tissue classes, and the interested reader is directed to (Mechelli et al., 2005) for details. There is simply not enough space to discuss all the approaches in a single book chapter. The next section provides a survey of a sample of medical conditions that yield MRI signatures, highlighting the imaging modalities deployed and the key results.

## *4.4 Fiber Tractography*

A relatively recent MRI modality was developed that examines the diffusivity of water - that is it measures the diffusion coefficient of water along multiple directions. Diffusion refers to the movement of molecules in some medium - in this case, a liquid medium such as the interstitial space that surrounds tissues, or the intracellular space. In bulk fluids, such as the CSF, water molecules are free to diffuse in any direction without running into physical barriers, which would tend to impede its flow in a particular direction. Such tissues are describes as isotropic - in that they do not influence the directional translation of molecules engaged in pure Brownian motion. In densely packed tissues, such as white matter, there are barriers to diffusion, as cellular membranes may impede the net movement of molecules - this type of tissue is termed anisotropic. White matter induces anisotropy - and this fact can be used to measure the direction of fiber tracts within the brain - this process results in what is termed fiber tractography.

Fiber tractography is an MRI technique that provides detailed information regarding the structural arrangements of tissue - and has been applied predominantly to white matter tracts. These tracts provide information about how axons in the CNS are connected together. A special imaging modality, termed diffusion tensor imaging (DTI) is used for this purpose. The pulse sequence is designed to detect the translational motion of molecules by recording across a selection of coordinates. Typically 9 directions are recorded, as is illustrated in Figure 17, which is termed the diffusion tensor (Bozzali & Cherunini, 2007, Basser et al., 2000).

The diffusion tensor provides directional information regarding the net diffusion of water molecules, and the principal axis is used to describe an ellipse, the principal direction being along the fiber tracks. The voxels will then consists of principal diffusion directions, and hence map out the direction of the fibers. An example of a DTI map is presented in Figure 18. DTI imaging can be used to

$$D = \begin{matrix} D_{XX} & D_{XY} & D_{XZ} \\ D_{YX} & D_{YY} & D_{YZ} \\ D_{ZX} & D_{ZY} & D_{ZZ} \end{matrix}$$

**Fig. 17** An example of a diffusion tensor with 9 separate recording directions. Note the principle axes are along the main diagonal of this tensor matrix.

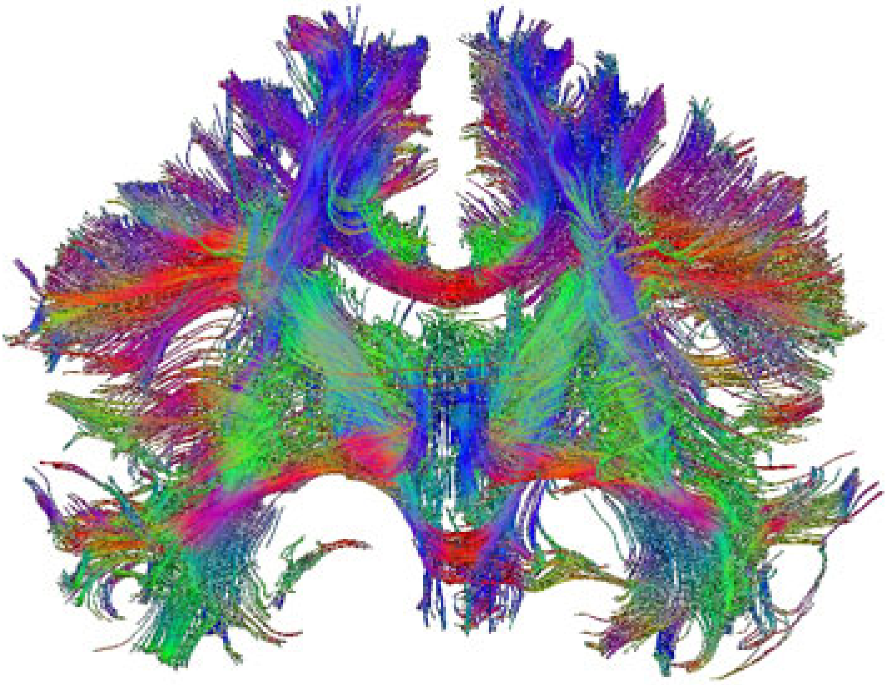


**Fig. 18** An example of fiber tractography, highlighting most of the major white matter tracts. Taken from http://www.biomed.ee.ethz.ch/research/bioimaging/brain/diffusion_fiber_tracking

examine quantitatively processes that disturb the microstructure of tissues. For instance, Multiple Sclerosis is a disease where myelin is destroyed through an auto-immune response. The extent of demyelination can be quantified using DTI imaging. This form of imaging has also been deployed to examine tissue damage resulting from stroke, and a variety of other neurological disorders (Ciccarelli et al., 2003, DaSilva et al., 2003).

# 5 Medical Applications

MRI has been used as a clinical tool for well over 30 years now - essentially since its invention. It was immediately recognised that the ability to examine structural information below the skin across the entire body was an unparalleled event in clinical diagnosis. With the appropriate hardware, pulse sequences, and contrast enhancement techniques, MRI can be used to investigate virtual any disease that yields a change in anatomical structure. This chapter has intentionally omitted another facet of MRI - functional imaging, designed to elicit information regarding functional activity within the brain. The interested reader is encouraged to consult (Ogawa et al., 1990, Logothetis, 2001) for a comprehensive exposition on this fascinating topic. Typical diseases examined using MRI include stroke, multiple sclerosis, Alzheimer's disease, and various forms of dementia. The diseases described in this section form a continuum with respect to the ease of detecting the disease both in terms of the resulting structural changes and the level of sophistication of the scanning protocol. Strokes tend to be quite evident on an MRI, visible with a variety of different scanning protocols. Dementias on the other hand may present subtle, but diffuse effects which are more difficult to detect and quantify. Issues such as quantifying the extent of the lesions may require specialised pulse sequences to enhance contrast, increasing the quantitative estimate of its extent and volume. In addition, it is very important to understand which regions of the brain have become infiltrated by the lesion - so the reliance on accurate segmentation approaches is critical in this regard. A variety of dementias - such as Alzheimer's (AD) and frontotemporal dementia (FTD) present very similar structural changes and require careful and very selective strategies for differentiation (Weiner, 2007, Wolf & Detre, 2007 ).

## 5.1 Stroke

Conditions such as stroke produce tissue damage through reductions in blood supply - either through an embolism or a hemorrhage (see Figure 18). The extent of the damage is primarily determined by the location of the occlusion - if it occurs at the start of the vessel - the damage can be widespread as the vascular territory is very large. Likewise, if the damage occurs at the distal end of a vessel, the damage can be quite focal and much harder to detect and quantify. The resulting tissue damage will occur across all tissue types - the condition is not restricted to white or gray matter for instance. When examining a stroke, issues such as the extent of the damage and the underlying structures that are involved is clinically important. A high resolution T1-weighted image is often used to provide a high resolution image for processing. The extent of the damage is typically quantified using a segmentation process - designed to demarcate and quantify the extent of the damage. In addition, the determination of which brain structures are included in the lesion is vitally important for prognosis and rational therapeutic strategies. This process can be facilitated using a Talairach mapping, which can provide information about which structures have become infarcted.

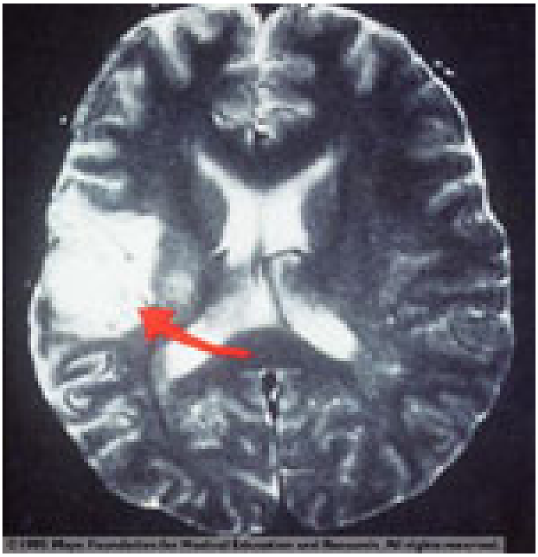


**Fig. 19** A T2-weighted axial image of a patient that has presented with symptoms of a stroke. Note the hyperintense region n the middle of the left hemisphere – this indicates that there has been an alteration of tissue structure – indicative of acute focal brain trauma.

T1or T2 -weighted images of stroke provide a snapshot of the damage caused by a stroke. In many cases though, a stroke evolves over time - expanding over the course of hours and even a few days after the event. A significant part of the expansion of a stroke is due to the effect of cerebral edema - a build-up of fluid within the brain that is generated by the stroke event. There are two types of edema - cytotoxic and vasogenic. Either can caused the extent of the damage to increase due to enhanced pressure on the underlying tissue. Cytotoxic edema refers to an increase in cellular water content - which will damage tissue due to disruption of cellular structures such as cell membranes (Loubinoux et al., 1997, Desmond et al., 2001). Vasogenic edema is typically caused by a breakdown of the blood brain barrier (BBB), which provides a barrier between the between the brain and the rest of the body. When the BBB integrity has been breached, which can occur as a result of a brain hemmorhage - the fluid balance within the brain is disrupted - but this time from the extracellular space - the region of the brain that doesn't contain cellular components. Vasogenic edema induces additional tissue damage due to hydrostatic pressure effects. The extent of the final tissue damage is related to the extent of the edema - and hence if one could estimate its extent in time - this would provide an estimate of the expected extent of the infarct expansion.

An imaging protocol called diffusion weighted imaging (DWI), detects subtle structural changes that reflect the extent of edema. This imaging technique measures the diffusion of water in a directional fashion. The diffusion coefficient reflects the mobility of water in a particular in 3D space - and is measured in units of area per unit time ($\mu m^2/s$). Cytotoxic edema tends to reflect the current physiological status of the tissue and does not appear to evolve over time.

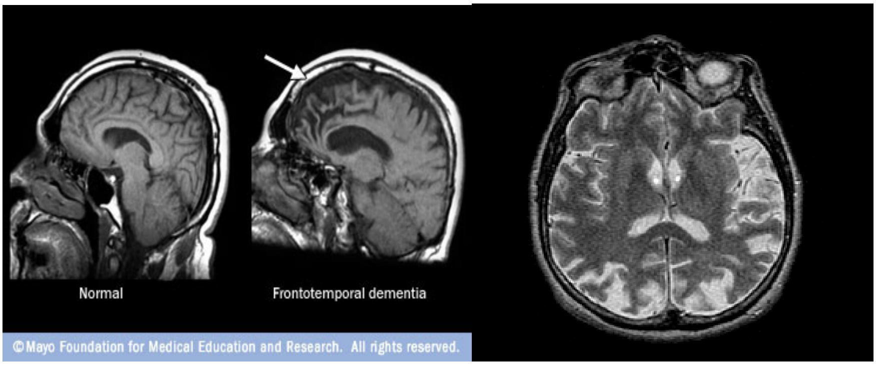
vasogenic edema on the other hand, reflects the potential for damage - in that a high level of vasogenic edema at the early stage of a stroke indicates a significant, though variable progression of tissue damage. it is therefore important to be able to differentiate cytotoxic from vasogenic edema, and to be able to quantify its extent. This ability is provided by pulse sequencing, imaging modalities that are able to detect the diffusion coefficient of water. This information can only be obtained through neuroimaging techniques based on MRI.

## *5.2 Dementias*

Dementia is a progressive neurodegenerative disease that is typically diagnosed through behavioral deficits that tend to occur in later stages in life. By the time the effects are noticeable from behavioral evidence, the diagnosis is fairly clear. What would be useful, is to have some sort of signature that could be used to predict its occurrence. In addition, it would be very informative to know which brain structures were altered during the course of the disease progression. In addition, there are a variety of dementias, all producing debilitating alterations in cognitive ability. For instance, two of the principle dementias, Alzheimer's Disease (AD) and frontotemporal dementia (FTD) produce very similar responses from a clinical perspective, and FTD is often mistaken for AD.

Frontotemporal dementia is a neurodegenerative condition involving the frontal aspects of the brain. It is the second-most common dementia after Alzheimer's disease, which principally affects the hippocampus and the temporal lobe. In their early stages, the two diseases present similar symptoms, making accurate diagnosis difficult. There are two principle mechanisms involving MRI that can be used to distinguish between these two different candidate diagnoses: one relies on measuring cortical thickness, and the other relies on an MRI modality termed arterial spin labelling (ASL). More specifically, Alzheimer's disease is associated with cortical thinning across all brain lobular regions (frontal, parietal, temporal and occipital lobes), while a different regional pattern of cortical thinning is found in FTD, involving primarily the frontal and temporal lobes. Cortical thickness measures can be established using voxel-based morphometry (VBM), which provides a quantitative estimate of cortical thickness across the entire aspect of the cortex (Ashburner & Friston, 2000). Therefore, cortical thinning may be diagnostic for differentiating FTD from AD - there is regional cortical thinning in both, but the regions tend to differ significantly. The best discriminator between Alzheimer's disease and FTD was parietal lobe atrophy in Alzheimer's disease. The latest research findings indicate dementia severity is negatively correlated with cortical thickness in Alzheimer's disease, while comparable correlations in FTD were not significant (Weiner, 2007, Wolf & Detre, 2007).

ASL is an imaging technique that measures perfusion levels in the tissue of interest. Cerebral blood flow is reduced (hypoperfusion) in patients diagnosed with dementia - whether it is a cause or effect remains to be elucidated (Detre, 2008). Not withstanding the issue of causality, patients with dementia are diagnosed with considerable hypoperfusion - the pattern of which may be



©Mayo Foundation for Medical Education and Research. All rights reserved.

**Fig. 20** A T2-weighted mid-sagittal image of a patient suspected of frontotemporal dementia (FTD) s on the left, and an axial image of a patient diagnosed with Alzheimer's disease on the right Both present a reduction in tissue volume, but in differing areas and to differing degrees.

diagnostic for a particular form of dementia. DTF and AD can be differentiated based on the spatial pattern of hypoperfusion - with the later producing hypoperfusion in the right frontal lobe regions, relative to patients with AD, which presented a more diffuse pattern of hypoperfusion (Du et al., 2006).

There are a variety of other forms of dementias - many are of vascular origin that can be detected using some form of perfusion based MRI. As new imaging techniques emerge - additional features associated with a variety of dementias will become available - but they must be matched with cognitive observations in order to provide the necessary correlations. As convenient and harmless as MRI scans are - not everyone will routinely undergo an MRI scan unless it is indicated by more traditional and routine observations and clinical measurements.

## 6 Conclusion

The ability to acquire high resolution images of internal structure of living organisms produced by MRI is unparalleled in the imaging world. Fine anatomical details can be brought into focus with high contrast across and in a high definition format - with voxels at mm dimensions. MRI provides access to structural deficits from direction measurements of their proton content - and can also examine and quantify cerebral blood flow - which provides a measure of the life expectancy of tissue challenged with alterations in cerebral blood flow. Software has been developed that allows a clinician to segment, extract, and quantify images at the voxel level in many cases. This technology is afforded to patients without any possibility of damaging repercussions - there is no radioactive materials nor ionizing radiation involved - the procedure is relatively quick and can be repeated without any significant side-effects.

The future of MRI is bright (even on a PD-weighted image!) - new pulse sequences are being developed at a rapid pace - each providing a unique window into the structural (no mention has been made regarding functional MRI in this chapter) effects produced by disease. Currently, the principle features that are recordable using MRI rely on perfusion, diffusion, and proton density of tissue. These features provide a wealth of information regarding the structural integrity of tissue - both in cases of disease and the normal ageing process. The quantitative aspects of MRI is an area that will continue to develop - in hand with novel pulse sequences. In conjunction with data from clinical and other venues, the applicability of MRI in the medical domain will continue to flourish.

## References

Ashburner, J., Friston, K.J.: Voxel-based morphometry - the methods. NeuroImage 11, 805–821 (2000)

Ballester, G., Zisserman, A.P., Brady, M.: Estimation of the partial volume effect in MRI. Med. Image Anal. 6(4), 389–405 (2002)

Basser, P.J., Pajevic, S., Pierpaoli, C., Duda, J., Aldroubi, A.: In vivo fiber tractography using DT-MRI data. Magnetic Resonance in Medicine 44, 625–632 (2000)

Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)

Bozzali, M., Cherubini, A.: Diffusion tensor MRI to investigate dementias: a brief review. Magnetic Resonance Imaging 25(6), 969–977 (2007)

Choi, S.H., Na, D.L., Chung, C.S., Lee, K.H., Na, D.G., Adair, J.C.: Diffusion-weighted MRI in vascular dementia. Neurology 54, 83–89 (2000)

Choi, S.H., Na, D.L., Chung, C.S., Lee, K.H., Na, D.G., Adair, J.C.: Diffusion-weighted MRI in vascular dementia. Neurology 54, 83–90 (2000)

Ciccarelli, O., Toosy, A.T., Parker, G.J.M., Wheeler-Kingshott, C.A.M., Brker, G.J., Miller, D.H., Thompson, A.J.: Diffusion tractography based group mapping of major white-matter pathways in the human brain. NeuroImage 19, 1545–1555 (2003)

DaSilva, A.F.M., Tuch, D.S., Wiegell, M.R., Hadjikhani, N.: A primer on diffusion tensor imaging of anatomical substractures. Neurosurg. Focus 15(1), 1–4 (2003)

Desmond, P.M., Lovell, A.C., Rawlinson, A.A., Parsons, M.W., Barber, P.A., Yang, Q., Li, T., Darby, D.G., Gerraty, R.P., Davis, S.M., Tress, B.M.: The Value of Apparent Diffusion Coefficient Maps in Early Cerebral Ischemia. AJNR Am. J. Neuroradiol. 22, 1260–1267 (2001)

Detre, J.A.: Arterial Spin Labeled Perfusion MRI. Clinical Neurology (2008)

Cosgrove, K.P., Mazure, C.M., Staley, J.K.: Evolving knowledge of sex differences in brain structure, function, and chemistry. Biol. Psychiat. 62(8), 847–855 (2007)

Hashemi, R.H., Bradley Jr., W.O., Lisanti, C.J.: MRI-The Basics, Lippincott Williams and Wilkins, 2nd Revised edn., October 1 (2003) ISBN: 0781741572

Hsu, Y.Y., Du, A.T., Schuff, N., Weiner, M.: Magnetic Resonance Imaging and Magnetic Resonance Spectroscopy in Dementias. J. Geriatr. Psychiatry Neurol. 14, 145–166 (2001)

Lauterbur, P.C.: Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance. Nature 242, 190–191 (1973)

Lee, S.K., Kim, D.I., Kim, J., Kim, D.J., Kim, H.D., Kim, D.S., Mori, S.: Diffusion-Tensor MR Imaging and Fiber Tractography: A New Method of Describing Aberrant Fiber Connections in Developmental CNS Anomalies. Radiographics 25, 53–65 (2005)

Li, Y., Chi, Z.: MR Brain Image Segmentation Based on Self-Organizing Map Network. International Journal of Information Technology 11(8), 45–53 (2005)

Logothetis, N.K.: Neurophysiological investigation of the basis of the fMRI signal. Nature 412, 150–157 (2001)

Loubinoux, I., Volk, A., Borredon, J., Guirimand, S., Tiffon, B., Seylaz, J., Meric, P.: Spreading of Vasogenic Edema and Cytotoxic Edema Assessed by Quantitative Diffusion and T2 Magnetic Resonance Imaging. Stroke 28, 419–427 (1997)

Mechellli, A., Ashburner, J., Friston, K.J.: Voxel-based morphometry of the human brain: methods and applications. Current Medical Imaging Reviews 1(13), 1–7 (2005)

Morri, T. (ed.): Computer Vision and Image Processing. Cornerstones of Computing Series. Palgrave MacMillan, New York (2004)

Ogawa, S., Lee, T.M., Nayak, A.S., Glynn, P.: Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. Magnetic Resonance in Medicine 14, 68–78 (1990)

O'Sullivan, M., Rich, P.M., Barrick, T.R., Clark, C.A., Markus, H.S.: Frequency of Subclinical Lacunar Infarcts in Ischemic Leukoaraiosis and Cerebral Autosomal Dominant Arteriopathy with Subcortical Infarcts and Leukoencephalopathy. AJNR Am. J. Neuroradiol. 24(7), 1348–1354 (2003)

Parker, G.J.M.: Analysis of MR diffusion weighted images. Br. J. Radiol. 77(suppl_2), S176–S185 (2004)

Schmierer, K., Parkes, H.G., So, P.W., An, S.F., Brandner, S., Ordidge, R.J., Yousry, T.A., Miller, D.H.: High field (9.4 Tesla) magnetic resonance imaging of cortical grey matter lesions in multiple sclerosis. Brain 133, 858–867 (2010)

Talairach, J., Tournoux, P.: Co-planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System - an Approach to Cerebral Imaging. Thieme Medical Publishers, New York (1988)

Tofts, P. (ed.): Quantitative MRI of the Brain. John Wiley & Sons, West Sussex (2003)

Weiner, M.W.: Different regional patterns of cortical thinning in Alzheimer's disease and frontotemporal dementia. Brain 130(pt. 4), 1159–1166 (2007)

Wolf, R.L., Detre, J.A.: Clinical Neuroimaging Using Arterial Spin-Labeled Perfusion Magnetic Resonance Imaging. Neurotherapeutics 4(3), 346–359 (2007)

# Chapter 8
# Image Analysis in Poincaré-Peters Perceptual Representative Spaces*
## A Near Set Approach

Sheela Ramanna

**Abstract.** The problem considered in this paper is how to detect similarities in the content of digital images, useful in image retrieval and in the solution of the image correspondence problem, *i.e.*, to what extent does the content of one digital image correspond to content of other digital images. The solution to this problem stems from a recent extension of J.H. Poincaré's representative spaces from 1895 introduced by J.F. Peters in 2010 and near sets introduced by J.F. Peters in 2007. Elements of a perceptual representative space are sets of perceptions arising from n-dimensional image patch feature vector comparisons. An image patch is a set of subimages. In comparing digital images, partitions of images determined by a particular form of indiscernibility relation $\sim_{\mathscr{B}}$ is used. The $L_1$ (taxicab distance) norm in measuring the distance between feature vectors for objects in either a perceptual indiscernibility or a perceptual tolerance. These relations combined with finite, non-empty sets of perceptual objects constitute various representative spaces that provide frameworks for image analysis and image retrieval. An application of representative spaces and near sets is given in this chapter in terms of a new form of content-based image retrieval (CBIR). This chapter investigates the efficacy of perceptual CBIR using Hausdorff, Mahalanobis as well as tolerance relation-based distance measures to determine the degree of correspondence between pairs of digital image. The contribution of this chapter is the introduction of a form of image analysis defined within the context of Poincare-Peters perceptual representative spaces and near sets.

Sheela Ramanna
University of Winnipeg,
Dept. Applied Computer Science,
515 Portage Ave., Winnipeg, Manitoba, R3B 2E9, Canada
e-mail: s.ramanna@uwinnipeg.ca

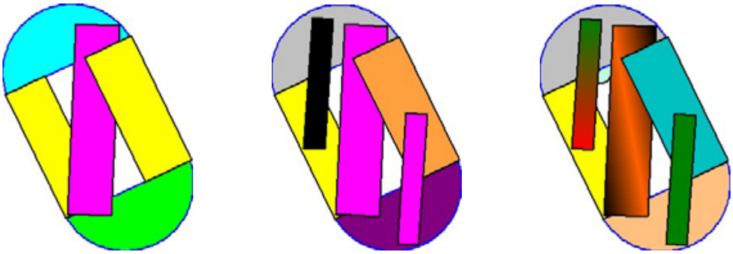
* Many thanks to James F. Peters, Som Naimpally, Amir H. Meghdadi and Christopher Henry for the suggestions and insights concerning topics in this paper. We especially want to thank Christopher Henry and Amir H. Meghdadi for the use of their implementations of near sets. This research has been supported by the Natural Science & Engineering Research Council of Canada (NSERC) grant 194376.

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 163–194.
springerlink.com

**Keywords:** Content-based image retrieval, image analysis, near sets, perception, perceptually indiscernible partitions, Poincaré, representative space.

## 1 Introduction

This chapter considers a solution to the problem of detecting similarities in digital images from based on a recent extension of J.H. Poincaré's representative spaces [1, 2] introduced by J.F. Peters in 2010 [3] and near sets introduced by J.F. Peters in 2007 [4, 5] and elaborated in [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. Briefly, near sets are disjoint sets that resemble each other. Resemblance between objects such image patches in digital image is represented by object description. A description is defined by an n-dimensional feature vector of real values, each value representing an observable feature such graylevel intensity or colour brightness. For example, the cmyk image in Fig. 1.1 resembles (is near) the image in Fig. 1.2, since both images have ■ (magenta) colour patches. Similarly, Fig. 1.2 is near Fig. 1.3, since both images have ■ (grey) colour patches.



1.1: cmyk image patches

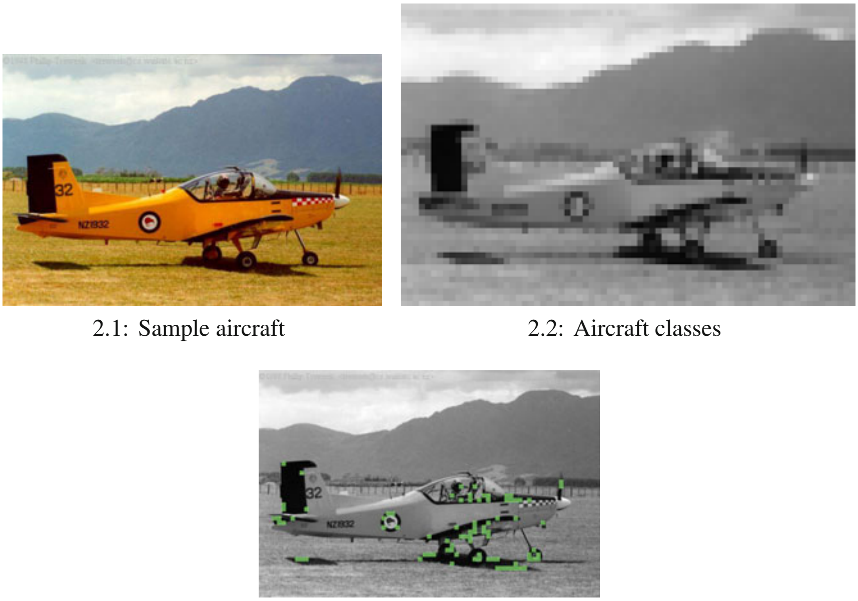
1.2: near cmyk colour patches, *i.e.*, Patches in 1.1 visually near Patches 1.2

1.3: least near cmyk colour patches, *i.e.*, Patches in 1.1 less visually near Patches 1.3

**Fig. 1** Sample Near Colour Sets

In Poincaré, a representative space [1] models a physical continuum that is neither homogeneous nor isotropic (the same in all directions) and contrasts with a mathematical continuum such as the familiar one in Euclidean geometry [**?**]. The elements of a physical continuum are sets of similar (perceptually indistinguishable) sensations. A set of similar sensations is determined by an implicit tolerance relation $\simeq_{\phi,\varepsilon}$, where $\phi$ is perceptible feature of sensation such as intensity of pinpoint pressure on the skin and $\varepsilon \in [0,+\infty)$ is a tolerance used to compare sensation feature values. Let $X,\Re$ denote a set of sensations and set of real numbers, respectively. Put $x,y \in X$. Let $\phi : X \to \Re^+$ denote a probe that maps sensations to real values. Then a simple tolerance relation is defined by

$$\simeq_{\phi,\varepsilon}= \{(x,y) \in X \times X : |\phi(x) - \phi(y)| \le \varepsilon\}.$$



2.1: Sample aircraft

2.2: Aircraft classes

2.3: Sample aircraft class

**Fig. 2** Sample Covering

A sample representative space is denoted by $\langle X, \simeq_{\phi,\varepsilon} \rangle$. For example, let $X$ denote a non-empty set of greylevel intensities in a visual image. In Poincaré's view, a pair of intensities $x, y$ belong to $A \subset X$ in case where $x \simeq_{\phi,\varepsilon} y$. For simplicity, let 1.1 = Fig. 1.1, for example. Then the following two sets can be identified, namely, $A = \{$🟪$_{1.1}$, 🟪$_{1.2}\}$, $B = \{$▨$_{1.2}$, ▨$_{1.3}\}$, indicating the visual "intersection" of three separate images. For an in-depth study of non-empty perceptual description-based intersection between disjoint sets, see [10].

A sample covering (all tolerance classes , each containing8 × 8 subimages of Fig. 2.1) is shown in Fig. 2.2. The relation $\simeq_{\phi,\varepsilon}, \phi : X \longrightarrow \Re$ (greylevel intensity) ,$\varepsilon = 0$ determines the classes denoted with a mixture ▨(grey) image patches in Fig. 2.2. The relation $\simeq_{\{\phi_1,\phi_2\},\varepsilon}, \phi_1 : X \longrightarrow \Re$ (greylevel intensity), and $\phi_2 : X \longrightarrow \Re$ (edge orientation) determines the class denoted with 🟩 (green) image patches in Fig. 2.3, *i.e.*, the image patches in this sample class each have the same average greylevel intensities and average edge orientation of the pixels in each of the patches shown, starting with the lower edge of the cockpit (this can be verified by starting with one of the ■ (black) subimages in the cockpit region, working round the parts of the aircraft in Fig. 2.3, comparing other subimage greylevels the greylevel of the subimage that you start with.

This chapter includes an application of the proposed approach to solving the image correspondence problem in terms of a new form of content-based image retrieval (CBIR). In a CBIR system, image retrieval from large image databases is based on some similarity measure of the actual contents of images rather than metadata such as captions or keywords. Image content can include colors, shapes, textures, or any other information that can be derived from an image. Most CBIR systems have one thing in common: images are represented by numeric values that signify properties of the images such as features or descriptors that facilitates meaningful retrieval [19]. In general, there are two approaches (i) the discrete approach is inspired by textual information retrieval and uses text retrieval metrics. This approach requires all features to be mapped to binary features; the presence of a certain image feature is treated like the presence of a word in a text document, (ii) the continuous approach is similar to nearest neighbor classification. Each image is represented by a feature vector and features are compared and subsequently ranked using various distance measures. Image patches (*i.e.*, sub-images of images) or features derived from image patches offer very promising approaches to CBIR [19]. Different types of local features can then be extracted for each subimage and used in the retrieval process. Our approach to perceptual CBIR can be viewed as an *image patch* approach where local feature values are first extracted from subimages. In the proposed approach to CBIR, image classes in an image covering determined by a tolerance relation provide the content used in the form of CBIR introduced in this chapter. This approach stems from a number of recent studies of the use of tolerance spaces in solving the image correspondence problem [16, 20, 11, 21, 3, 12, 22].

This paper has the following organization. The basic notation used in this chapter as well as an introduction to object description, perceptual systems and perceptual Pawlak partitions (PIPs) are introduced in Sect. 3. Then, in Sect. 4, a brief introduction to near sets is given. Probe functions for image features investigated in this chapter are given in Sect. 5. The distance measures are briefly explained in Sect. 6. Extensive set experiments to illustrate the new approach to CBIR are given in Sect. 7 and Sect. 8.

## 2 Related Works

An extensive survey of a broad range of image retrieval systems which includes a comparison of selected features, querying schemes and matching methods can be found in [23, 24]. Similarity between pairs of images can be defined with respect to either the image content or with respect to concepts that is derived from the visual content of an image [25, 26]. Designing a complete image retrieval system would require an image indexing scheme, user interface design, possibly relevance feedback or active learning procedures to facilitate interactive searches [27, 28, 29]. Although features such as color, shape, texture are considered to be helpful in human judgment of image similarity, human perception is not fully considered at the algorithmic stage of feature extraction. In [30], a cognitive discriminative biplot, is used to capture perceptual similarity. This method is structural where point diagrams capture the objects, their relationships and data variations. The evaluation of the biplot is conducted using a standard nearest neighbour algorithm.

Our solution to the image correspondence problem stems from an approach to pairwise comparison of images that is similar to G. Fechner's 1860 approach to comparing perceptions in psychophysics experiments [31]. For Fechner, a pair of perceptions are indistinguishable if there is no perceptible difference in a particular feature of the perceived objects, *e.g.*, perception resulting from lifting small objects where the feature is weight.

Of interest here is a solution to establishing a correspondence between regions of pairs of images using image comparison strategies. The particular form of partitions of sets considered here are named after Z. Pawlak because the partitions are defined by an equivalence relation inspired by the original indiscernibility relation introduced by Z. Pawlak in 1981 [32] and elaborated in [33, 34]. The indiscernibility relation is a basic building block in defining rough sets [33].

Rough set-based approach to image analysis dates back to the early 1990s. The application of rough sets in image analysis was launched in a seminal paper published by A. Mrózek and L. Plonka [35]. The early work on the use of rough sets in image analysis can be found in [36, 37, 38, 39, 40, 41, 42]. A review of rough sets and near sets in medical imaging can be found in [14]. More recently, D. Sen and S.K. Pal [43] introduce an entropy based, average image ambiguity measure to quantify greyness and spatial ambiguities in images. This measure has been used for standard image processing tasks such as enhancement, segmentation and edge detection. Forms of rough entropy have also been used in a clustering approach to image segmentation [44, 16]. Papers related to the foundations and applications of fuzzy sets, rough sets and near sets approaches in image analysis can be found in S.K. Pal and J.F. Peters [16].

In solving the image correspondence problem, each image is viewed as set of points. One often used approach is to assign weight to point sets and to define distance functions that incorporates not only the position but also the weight of points [45]. A well-known distance function is the Earth Movers Distance(EMD) [46]. A Proportional Transportational Distance (PTD) was introduced by Giannopolus and Veltkamp in [45]. PTD is a pseudo-metric that is invariant under rigid motion, respects scaling and obeys triangle inequality. This is in contrast to EMD which does not obey triangle inequality for sets of unequal total weight.

Our approach is to consider partitions of images that are defined by a perceptual indiscernibility relation introduced by J.F. Peters and S. Ramanna in [6] and elaborated in [12, 3, 16], where the descriptions of regions of images are compared. Let $x,y$ denote a pair of subimages in an image $X$ and let $\mathscr{B}$ denote a set of real-valued functions that represent subimage features. In the simplest form of a perceptual indiscernibility relation $\sim_{\mathscr{B}}$, put $\sim_{\mathscr{B}} = \{(x,y) \in X \times X \mid \forall \phi \in \mathscr{B}, \phi(x) = \phi(y)\}$. In comparing a pair of images $X,Y$, a partition of each image is defined by $\sim_{\mathscr{B}}$. Then pairs of digital images are near sets to the extent that partitions of the images resemble each other, *i.e.*, image resemblance is present in the case where pairs of classes $A \subset X, B \subset Y$ contain subimages $x \in A, y \in B$ where $x \sim_{\mathscr{B}} y$. In this paper, the degree of image resemblance is determined by specialized forms of distance measures: Hausdorff [47, 48, 49], Mahalanobis [50, 51] and tolerance nearness measures.

# 3 Basic Notions

This section briefly presents the basic notions underlying perceptually near Pawlak partitions used in this work.

## *3.1 Description and Perceptual Systems*

An object description in rough set theory is defined by vectors of attribute values in an information table. However, in near set theory, description of an object $x$ is defined by means of a vector of real-valued function values $\phi(x)$, named *probe functions*, that gives some measurable (or perceivable features) of an object in the physical world.

$$\phi_{\mathscr{B}}(x) = (\phi_1(x), \phi_2(x), ..., \phi_i(x), ..., \phi_l(x)), \qquad (1)$$

where $\phi_i : X \rightarrow \Re$ (reals). The object description representation in (1), was first introduced in [52] based on the intuition that object descriptions are analogous to recorded measurements from sensors and hence the name probe function. Near set theory is defined in the context of a perceptual system [17]. A perceptual system is defined as a set of perceptual objects $O$ along with a set of probe functions $\mathbb{F} = \{\phi_1, \phi_2, ..., \phi_L\}$ and is denoted by $\langle O, \mathbb{F} \rangle$. Nearness relation is then defined between sets of perceptual objects from the perceptual system relative to the probe functions defined in the perceptual system. Sets $X, Y \subseteq O$ are *weakly near* to each other if, and only if there are $x \in X$ and $y \in Y$ and there is $\mathscr{B} \subseteq \mathbb{F}$ such that $x$ and $y$ are indiscernible to each other (*i.e*, $\forall \phi_i \in \mathscr{B}, \phi_i(x) = \phi_i(y)$).

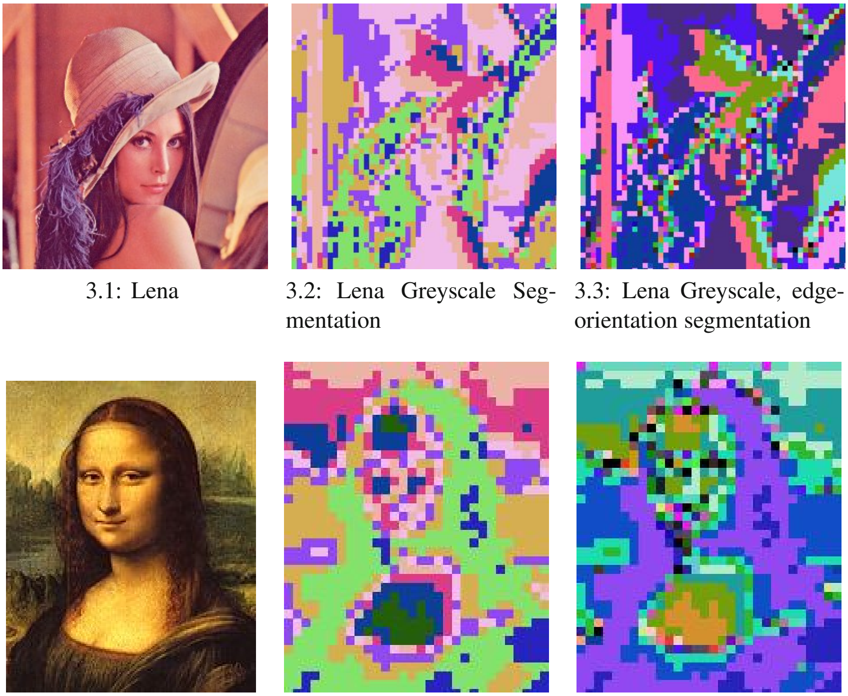
**Definition 1. Perceptual System**
A perceptual system $\langle O, \mathbb{F} \rangle$ consists of a sample space $O$ containing a finite, non-empty set of sensed sample objects and a countable, non-empty set $\mathbb{F}$ containing probe functions representing object features.

The perception of physical objects and their description within a perceptual system facilitates pattern recognition and the discovery of sets of similar objects.

## *3.2 Perceptual Pawlak Partitions*

It is possible to extend the original idea of an indiscernibility relation between objects [32, 34, 53] to what is known as perceptual indiscernibility (introduced in [12]) that befits the perception of objects in the physical world, especially in science and engineering applications where there is an interest in describing, classifying and measuring device signals and various forms of digital images.

**Definition 2. Perceptual Indiscernibility Relation** [12]
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. Let $\mathscr{B} \subseteq \mathbb{F}$ and let $\phi_{\mathscr{B}}(x)$ denote a description of an object $x \in O$ of the form $(\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_k(x))$. A perceptual indiscernibility relation $\sim_{\mathscr{B}}$ is defined relative to $\mathscr{B}$, *i.e.*,



3.1: Lena

3.2: Lena Greyscale Segmentation

3.3: Lena Greyscale, edge-orientation segmentation

3.4: Mona Lisa (ML)

3.5: ML Greyscale Segmentation

3.6: ML Greyscale, edge-orientation segmentation

**Fig. 3** Sample Segmentations

$$\sim_{\mathscr{B}} = \{(x,y) \in O \times O \mid \| \phi_{\mathscr{B}}(x) - \phi_{\mathscr{B}}(y) \|_1 = 0\},$$

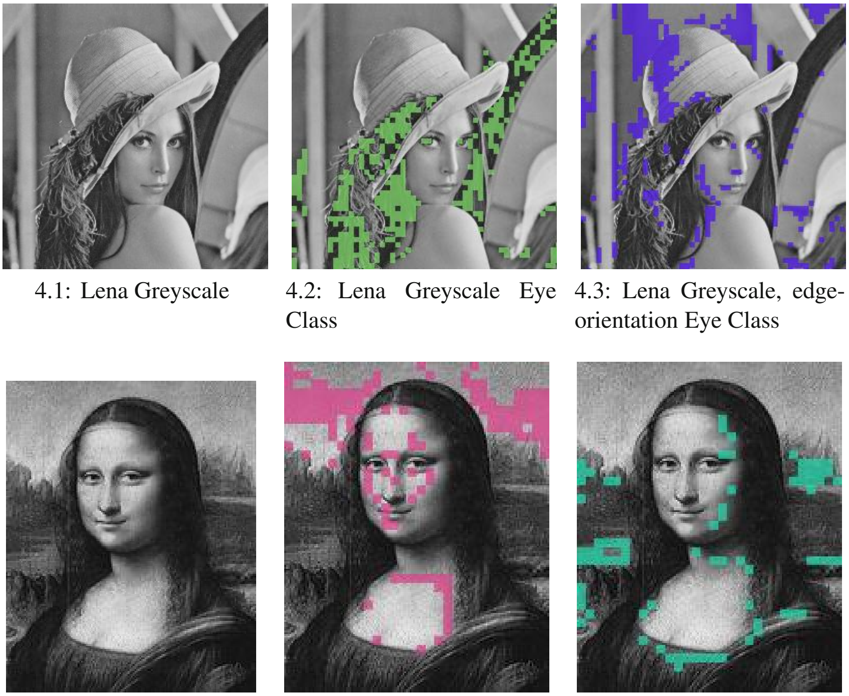
where $\| \cdot \|_1 = \Sigma_{i=1}^{k} | \cdot | (L_1$ norm).

In this section, classes are defined in the context of perceptual partitions of digital images.

**Definition 3. Perceptual Pawlak Partitions**
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. Let $X \subseteq O, \mathscr{B} \subseteq \mathbb{F}$. A perceptual Pawlak partition (PIP) of $X$ defined by $\sim_{\mathscr{B}}$ is denoted by $X_{/\sim_{\mathscr{B}}}$, *i.e.*,

$$X_{/\sim_{\mathscr{B}}} = \bigcup_{x \in X} x_{/\sim_{\mathscr{B}}}.$$

A perceptual Pawlak partition is a cover of $X$ [54], *i.e.*, a separation of the elements of $X$ into non-overlapping classes where $x_{/\sim_{\mathscr{B}}}$ denotes an equivalence class containing $x$.



4.1: Lena Greyscale

4.2: Lena Greyscale Eye Class

4.3: Lena Greyscale, edge-orientation Eye Class

4.4: Mona Lisa Greyscale

4.5: Mona Lisa Greyscale Eye Class

4.6: Mona Lisa Greyscale, edge-orientation Eye Class

**Fig. 4** Sample Classes

**Example 1. Sample Greyscale Classes**
Sample feature-based segmentations are shown in Fig. 4. The ■ boxes in Fig. 3.2 and Fig. 3.5 represent greyscale classes. These classes reveal the contrast between Lena and Mona Lisa, *i.e.*, the shading over Lena's eyes, bridge of her nose tends, hatbrim and hatfrong tend to be uniformly spread over Mona Lisa's face and appear prominently on either side of Mona Lisa's head.

**Example 2. Sample Perceptual Edge Classes in Pawlak Partition**
Fig. 3 presents two examples of perceptual Pawlak partitions. The boxes in Fig. 4.3 and Fig. 4.6 represent edge orientation classes. Each class contains subimages with the matching edge orientations.

The different coloured regions in Fig. 4.3 and Fig. 4.6 represent different equivalence classes based on $\mathscr{B} = \{e_o\}$. It is worth noting that the size (and the nature) of the equivalence classes are much smaller for the $e_o$ feature when compared with $\overline{gs}$ feature.

# 4 Near Sets and Perceptually Near Pawlak Partitions

> The basic idea in the near set approach to object recognition is to compare object descriptions. Sets of objects $X,Y$ are considered near each other if the sets contain objects with at least partial matching descriptions.
>
> –Near sets. General theory about nearness of objects,
> –J.F. Peters, 2007.

## *4.1 Near Sets*

Near sets are disjoint sets that resemble each other [15]. Resemblance between disjoint sets occurs whenever there are observable similarities between the objects in the sets. Similarity is determined by comparing lists of object feature values. Each list of feature values defines an object's description. Comparison of object descriptions provides a basis for determining the extent that disjoint sets resemble each other. Objects that are perceived as similar based on their descriptions are grouped together. These groups of similar objects can provide information and reveal patterns about objects of interest in the disjoint sets.

Near set theory provides methods that can be used to extract resemblance information from objects contained in disjoint sets, i.e., it provides a formal basis for the observation, comparison, and classification of objects. The discovery of near sets begins with choosing the appropriate method to describe observed objects. This is accomplished by the selection of probe functions representing observable object features. A basic model for a probe function was introduced by M. Pavel [55] in the text of image registration and image classification. In near set theory, a probe function is a mapping from an object to a real number representing an observable feature value [5]. For example, when comparing objects in digital images, the texture feature (observed object) can be described by a probe function representing contrast, and the output of the probe function is a number representing the degree of contrast between a pixel and its neighbour.

Probe functions provide a basis for describing and discerning affinities between objects as well as between groups of similar objects [6]. Objects that have, in some degree, affinities are considered near each other. Similarly, groups of objects (i.e. sets) that have, in some degree, affinities are also considered near each other.

**Definition 4. Near Sets**
Let $\langle O,\mathbb{F}\rangle$ be a perceptual system. Let $X,Y \subset O$ be disjoint sets in $O$ and $\mathscr{B} \subseteq \mathbb{F}$. $X \underline{\bowtie}_{\mathscr{B}} Y$ ($X$ and $Y$ are near each other) if, and only if there are $x \in X, y \in Y$, where $x \sim_{\mathscr{B}} y$, *i.e.*, $x$ and $y$ have matching descriptions.

Preclasses in tolerance relations were introduced by M. Schroeder and M. Wright [56].

**Definition 5. Perceptual Preclass** [16]
Let $\big\langle O,\mathbb{F}\big\rangle$ be a perceptual system, $A \subset \cong_{\mathscr{B},\varepsilon}$. $A$ is a preclass in $\cong_{\mathscr{B},\varepsilon}$ if, and only if $\forall x,y \in A,\ x \ \cong_{\mathscr{B},\varepsilon}\ y$.

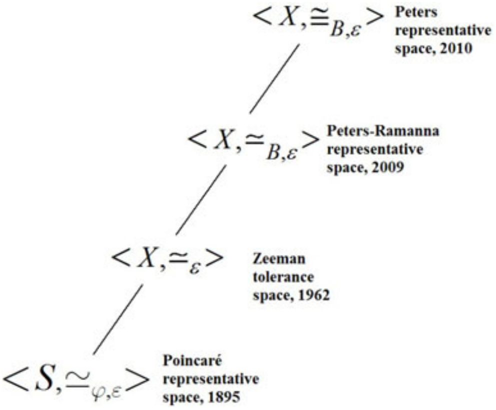

$<X, \cong_{B,\varepsilon}>$
Peters representative space, 2010
$<X, \simeq_{B,\varepsilon}>$
Peters-Ramanna representative space, 2009
$<X, \simeq_{\varepsilon}>$
Zeeman tolerance space, 1962
$<S, \simeq_{\varphi,\varepsilon}>$
Poincaré representative space, 1895

**Fig. 5** Growth of Representative Spaces

A tolerance class is a maximal preclass in a tolerance relation. Considered in the context of perceptual tolerance relations, we obtain

**Definition 6. Perceptual Class** [16]
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system, $A \subset \cong_{\mathscr{B},\varepsilon}$. $A$ is a perceptual tolerance class in $\cong_{\mathscr{B},\varepsilon}$ if, and only if $A$ is a maximal preclass in $\cong_{\mathscr{B},\varepsilon}$. Unless otherwise specified, $\mathbb{C}_{\cong_{\mathscr{B},\varepsilon}}$ denotes a perceptual tolerance class.

Every pair of objects $x, y$ in a perceptual tolerance class $\mathbb{C}_{\cong_{\mathscr{B},\varepsilon}}$ must satisfy the condition $\| \phi_{\mathscr{B}}(x) - \phi_{\mathscr{B}}(y) \|_2 \leq \varepsilon$, *i.e.*, $x, y$ have similar descriptions. Perceptual systems and tolerance near sets provide a feature-based solution of the image correspondence problem. The basic idea is to discover tolerance classes containing images with descriptions that differ from each other within a preset tolerance. Let $\mathscr{B} \subseteq \mathbb{F}$ denote a set of probe functions representing object features. Pairs of images $X, Y$ with coverings defined by a tolerance relation resemble each other in the case where $X \underset{\cong_{\mathscr{B},\varepsilon}}{\bowtie} Y$ for some tolerance $\varepsilon$.

**Definition 7. Tolerance Near Sets**
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. Put $\varepsilon \in \Re, \mathscr{B} \subset \mathbb{F}$. Let $X, Y \subset O$ denote disjoint sets with coverings determined by a tolerance relation $\cong_{\mathscr{B},\varepsilon}$. Sets $X, Y$ are tolerance near sets if, and only if there are preclasses $A \subset X, B \subset Y$ such that $A \underset{\cong_{\mathscr{B},\varepsilon}}{\bowtie} B$.

Notice that $\underset{\cong_{\mathbb{B},\varepsilon}}{\bowtie}$ is a specialized form of the weak nearness relation $\underset{\cong_{\mathbb{F}}}{\bowtie}$ in Def. 7 and shown to be a tolerance relation in Corollary 5.5 [17].

## 4.2 *Representative Spaces*

One the simplest forms of a representative space is a direct result of the perceptual indiscernibility relation in Def. 2.

**Definition 8. Perceptual Representative Space** [6, 3]
A *perceptual representative space* is denoted by $\langle O, \sim_{\mathscr{B}} \rangle$ where $O$ is a non-empty set of *perceptual objects*, $\mathscr{B}$ a countable set of *probe functions*, and $\sim_{\mathscr{B}}$ is a perceptual indiscernibility relation.

Even with its failure to take into account the problem of varying similarity between image patches in digital images , this form of representative space has proven to be quite useful (see, *e.g.*, [9]).

**Definition 9. Poincaré Representative Space** [1]
A *Poincaré representative space* is denoted by $\langle O, \simeq_{\phi,\varepsilon} \rangle$ where $O$ is a non-empty set of sensations, $\phi$ a single *probe function* $\phi \rightarrow \mathfrak{R}$, and $\simeq_{\phi,\varepsilon}$ is a tolerance relation

$$\simeq_{\phi,\varepsilon} = \{(x,y) \in O \times O : |\phi(x) - \phi(y)| \leq \varepsilon\}.$$

A.B. Sossinsky observed in 1986 [57] that the main idea underlying tolerance space theory comes from Poincaré, especially [1] (Poincaré work on representative spaces (*aka* tolerance spaces) was not mentioned by Zeeman). In 2002, Z. Pawlak and J. Peters considered an informal approach to the perception of the nearness of physical objects such as snowflakes that was not limited to spatial nearness [58]. In 2006, a formal approach to the nearness of objects was considered by J. Peters, A. Skowron and J. Stepaniuk [13] in the context of proximity spaces [59, 60, 61, 62]. The term *tolerance space* was coined by E.C. Zeeman in 1961 in modelling visual perception with tolerances (Zeeman, 1962; Zeeman and Buneman, 1968). A tolerance space is a set $X$ supplied with a binary relation $\simeq$ (*i.e.*, a subset $\simeq \subset X \times X$) that is reflexive (for all $x \in X, x \simeq x$) and symmetric (for all $x, y \in X, x \simeq y$ and $y \simeq x$) but transitivity of $\simeq$ is not required. For example, it is possible to define a tolerance space relative to sets of images. This is made possible by assuming that each image is a set of fixed points.

This leads to what are known as perceptual representative spaces introduced in [3]. Consider, now, a countable set $\mathscr{B}$ containing probe functions in a perceptual system and a more general tolerance relation $\simeq_{\mathscr{B},\varepsilon}$. This leads to what are known as perceptual representative spaces introduced in [3].

**Definition 10. Perceptual Tolerance Representative Space** [3]
A *perceptual representative space* is denoted by $\langle O, \simeq_{\mathscr{B}} \rangle$ where $O$ is a non-empty set of *perceptual objects*, $\mathscr{B}$ a countable set of *probe functions*, and $\simeq_{\mathscr{B}}$ tolerance relation

$$\simeq_{\mathscr{B},\varepsilon} = \{(x,y) \in O \times O : \| \phi_{\mathscr{B}}(x) - \phi_{\mathscr{B}}(y) \|_p \leq \varepsilon\},$$

where $\| \cdot \|_p = (\sum_{i=1}^{n}( \cdot_i^p)^{\frac{1}{p}}$ ($L_p$ norm). Usually, either $p = 1$ (taxicab distance) or $p = 2$ (Euclidean distance). In this chapter, $p = 1$.

Perceptual representative spaces have close affinity with proximity spaces [63, 64, 65, 66, 67, 68], especially in terms of recent work on nearness in digital images [69]. Originally, the notion of nearness between sets (proximity relation between sets) was derived from a spatial meaning of distance by V. Efremovič [67, 68]. Later, the notion of proximity between sets became more general and closer to the notion of nearness underlying near sets, *i.e.*, proximity not limited to a spatial interpretation was introduced by S.A. Naimpally in 1970 [63] (see, *e.g.*, [64, 65]). This later form of proximity relation permits either a quantitative (spatial) or qualitative (non-spatial) interpretation.

Perceptual representative spaces are patterned after the original notion of a representative space introduced by J.H. Poincaré during the 1890s [1]. Poincaré's form of representative space was used to specify a physical continuum associated with a source of sensation, *e.g.*, visual, tactile, motor. A perceptual representative space $\mathscr{P} = \langle O, \sim_{\mathscr{B}} \rangle$ is a generalization of Poincaré's representative space inasmuch as $\mathscr{P}$ represents sets of similar perceptual objects in a covering of $O$ determined by a relation such as $\sim_{\mathscr{B}}$.

For example, a digital image $X$ can be viewed as a set of points represented by image pixels (picture elements). Pairs of digital images containing pixels with matching descriptions are near sets. Now consider the partition pairs of disjoint sets.

**Proposition 1.** Let $\langle O, \sim_{\mathscr{B}} \rangle$ denote a perceptual representative space and let $X, Y \subset O$ denote disjoint sets in $O$ and $\mathscr{B}$ a set of probe functions representing object features. Then $X \underline{\bowtie}_{\mathscr{B}} Y$ if, and only if $X_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} Y_{/\sim_{\mathscr{B}}}$.

*Proof.* Let $\langle O, \sim_{\mathscr{B}} \rangle$ denote a perceptual representative space, where $O$ is a non-empty set of perceptual objects, $\mathscr{B}$ a set of probe functions representing object features. And let $X, Y \subset O$ denote a pair of disjoint sets. A perceptual indiscernibility relation $\sim_{\mathscr{B}}$ determines, for example, a partition of $X$, separation of $X$ into disjoint subsets. Consider quotient sets $X_{/\sim_{\mathscr{B}}}, Y_{/\sim_{\mathscr{B}}}$ determined by $\sim_{\mathscr{B}}$.
$\Rightarrow$ Assume $X \underline{\bowtie}_{\mathscr{B}} Y$. Then, from Def. 4, there are $x \in X, y \in Y$ such that $x \sim_{\mathscr{B}} y$. That is, $x \in X_{/\sim_{\mathscr{B}}}, y \in Y_{/\sim_{\mathscr{B}}}$ have matching descriptions. Then, again from Def. 4, $X_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} Y_{/\sim_{\mathscr{B}}}$.
$\Leftarrow$ Assume $X_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} Y_{/\sim_{\mathscr{B}}}$. The proof that therefore $X \underline{\bowtie}_{\mathscr{B}} Y$ again follows from Def. 4 and the approach in the proof is symmetric with the proof of $\Rightarrow$. □

**Corollary 1.** Let $\langle O, \sim_{\mathscr{B}} \rangle$ denote a perceptual representative space, where $O$ is a non-empty set of digital images, $\mathscr{B}$ a set of probe functions representing subimage features, and $\sim_{\mathscr{B}}$ is a perceptual indiscernibility relation. Also, let $X, Y \subset O$ denote digital images in $O$. Then $X \underline{\bowtie}_{\mathscr{B}} Y$ if, and only if there are $x_{/\sim_{\mathscr{B}}} \in X_{/\sim_{\mathscr{B}}}$ and $y_{/\sim_{\mathscr{B}}} \in Y_{/\sim_{\mathscr{B}}}$ such that $x_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} y_{/\sim_{\mathscr{B}}}$.

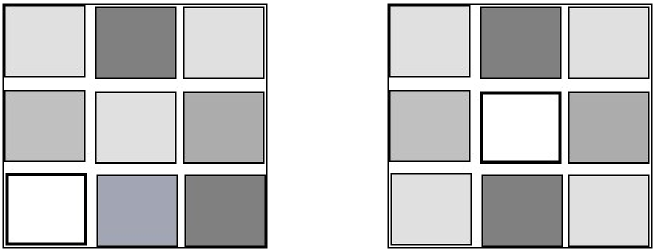
It is now possible to specialize Cor. 1 by considering $O$ to be a non-empty set of sample digital images (sets of points). The partition of a digital image $X$ (set of points) entails the selection of a subimage size, usually containing $p \times p$ pixels. In picture terms, $p$ is the number of points on a subimage edge. Let $x \in X$ denote a subimage in $X$ and let $x_{/\sim_{\mathscr{B}}}$ denote a class containing $x$, where every other subimage in $x_{/\sim_{\mathscr{B}}}$ has description similar to the description of $x$. In other words, a class in a partition of $X$ consists of one more more subimages with similar descriptions. In image processing terms, such a partition determines an image segmentation consisting of non-overlapping regions that are identified with classes.

**Corollary 2.** Let $\langle O, \sim_{\mathscr{B}} \rangle$ denote a perceptual representative space, where $O$ is a non-empty set of digital images, $\mathscr{B}$ a set of probe functions representing subimage features, and $\sim_{\mathscr{B}}$ is a perceptual indiscernibility relation. Also, let $X, Y \subset O$ denote digital images in $O$. Then $X \underline{\bowtie}_{\mathscr{B}} Y$ if, and only if $X_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} Y_{/\sim_{\mathscr{B}}}$.

An obvious extension of Prop. 1 is given in Prop. 2. Let $O$ denote a set of finite, non-empty set image patches and let $X, Y \subset O$ denote disjoint sets of image patches in individual digital images. Also, let $\mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)$ denote the family of all tolerance classes of relation $\cong_{\mathscr{B}}$ on the set $X \cup Y$.

**Proposition 2.** Let $\langle O, \cong_{\mathscr{B},\varepsilon} \rangle$ denote a perceptual representative space and let $X, Y \subset O$ denote disjoint sets in $O$ and $\mathscr{B}$ a set of probe functions representing object features. Let $A \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X), B \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$. Then $X \underline{\underline{\bowtie}}_{\mathscr{B},\varepsilon} Y$ if, and only if $A \underline{\underline{\bowtie}}_{\mathscr{B},\varepsilon} B$

*Proof.* Symmetric with the proof of Prop. 1. □



6.1: Image im1 partition

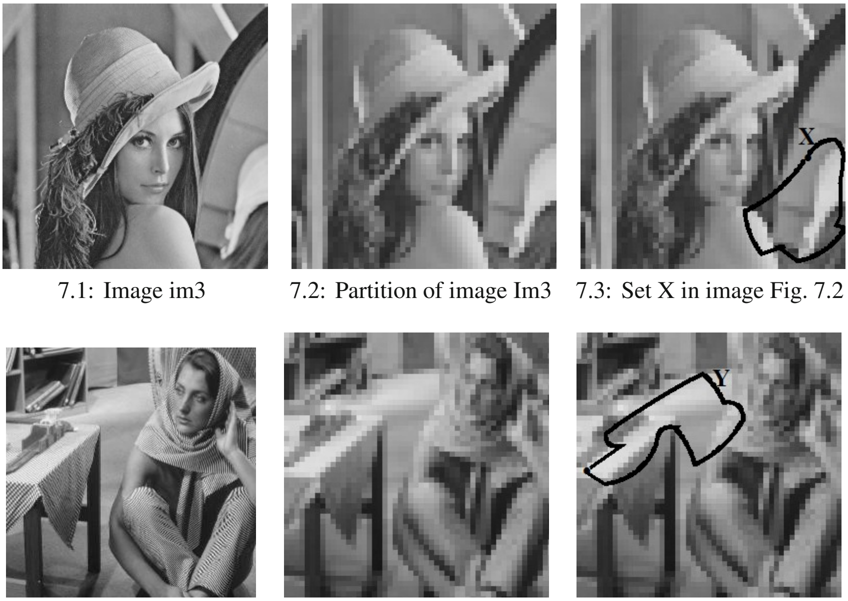
6.2: Image im2 partition

**Fig. 6** Sample Perceptually Near Image Partitions

**Example 3. Perceptually Near Digital Image Partitions**
Let $\langle O, \sim_{\mathscr{B}} \rangle$ denote a perceptual representative space, where $O$ is a non-empty set of digital images, $\mathscr{B}$ a set of probe functions representing subimage features, and $\sim_{\mathscr{B}}$ is a perceptual indiscernibility relation. Consider the images $Im1, Im2 \subset O$ in Fig. 6.1 and Fig. 6.2, respectively. Let $\overline{gs}$ denote a function that returns the average greyscale value for the pixels in a subimage and assume $\mathscr{B} = \{\overline{gs}\}$. Let $x_{/\sim_{\{\overline{gs}\}}} \in X_{/\sim_{\{\overline{gs}\}}}, y_{/\sim_{\{\overline{gs}\}}} \in Y_{/\sim_{\{\overline{gs}\}}}$ denote classes represented by a grey-shaded box ■ in the partition of the images in Fig. 6.1 and Fig. 6.2. In effect, the classes with grey-shaded boxes ■ represent subimages that have matching descriptions.

Since these images contain classes (represented by shaded boxes) with matching shades of grey determined by $\sim_{\mathscr{B}}$, such classes are examples of near sets. That is, since there are classes in the segmentations in Fig 6 that are near sets, we know from Def. 4 that $Im1_{/\sim_{\{\overline{gs}\}}} \underline{\bowtie}_{\mathscr{B}} Im2_{/\sim_{\{\overline{gs}\}}}$. Then, from Cor. 2, $Im1 \underline{\bowtie}_{\mathscr{B}} Im2$.

**Example 4. Tolerance Space-Based Near Digital Images**
Let $O$ denote a set of finite, non-empty set image patches and let $X, Y \subset O$ denote disjoint sets of image patches in individual digital images. Also, let $\mathrm{H}^{\varepsilon}_{\mathscr{B}}(X)$



7.1: Image im3

7.2: Partition of image Im3

7.3: Set X in image Fig. 7.2

7.4: Image im4

7.5: Partition of image Im4

7.6: Set Y in image Fig. 7.5

**Fig. 7** Sample Perceptually Near Partitions

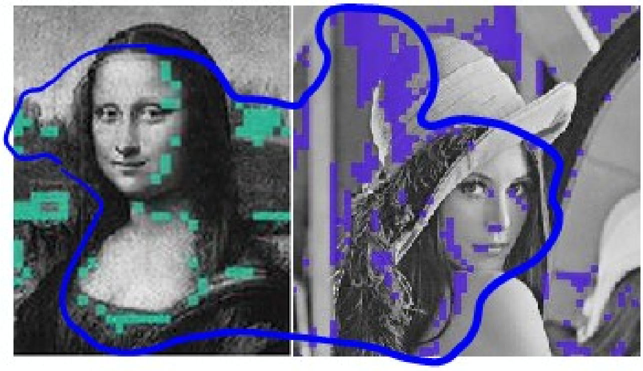


**Fig. 8** Sample Perceptually Near Image Tolerance Classes

denote the family of all tolerance classes of relation $\cong_{\mathscr{B}}$ on image $X$ and let $\mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$ denote the family of all tolerance classes of relation $\cong_{\mathscr{B}}$ on image $Y$. Let $\langle O, \cong_{\mathscr{B},\varepsilon}\rangle$ denote a perceptual tolerance representative space and let $X, Y \subset O$

denote disjoint sets in $O$ and $\mathscr{B}$ a set of probe functions representing object features. Let $A \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X), B \subset \mathrm{H}^{\varepsilon}_{\mathscr{B}}(Y)$. From Prop. 2, we know that $X \underline{\underline{\bowtie}}_{\mathscr{B},\varepsilon} Y$ if, and only if $A \underline{\underline{\bowtie}}_{\mathscr{B},\varepsilon} B$. For example, in Fig. 8, a pair of tolerance classes are shown, one class on the Mona Lisa image in Fig. 4.4 and another class on Lena in Fig. 4.1. Notice the ■ subimage in the left hand corner of Lena's right eye. All of the 🟪 and 🟦 image patches shown in the two images correspond to subimages that have average greylevels and edge orientations that are similar (within tolerance $\varepsilon = 0.1$) to the average greylevel and edge orientation of the ■ right eye subimage.

## 4.3 *Perceptually Near Digital Images*

The partition of an image defined by $\sim_{\mathscr{B}}$ results in the separation of the parts of the image into equivalence classes, *i.e.*, results in an image segmentation. A $\sim_{\mathscr{B}}$-based segmentation is called a *perceptually indiscernible partition* (PIP). Notice that, by definition, a class in the partition of an image is set of subimages with matching descriptions.

In the partition in Fig. 7.2, for example, a single class is represented by the ■ (dark grey) shaded boxes scattered throughout the partition. Depending on the features chosen, PIPs will be more or less perceptually near each other. The notion of perceptually near partitions was introduced in [4, 70] and elaborated in [12, 17, 15, 18, 71].

**Definition 11. Perceptually Near Pawlak Partitions**
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system, where $O$ is a non-empty set of picture elements (points) and $X, Y \subset O$, *i.e.*, $X, Y$ are digital images, and $\mathscr{B} \subseteq \mathbb{F}$. $X_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} Y_{/\sim_{\mathscr{B}}}$ if, and only if there are $x_{/\sim_{\mathscr{B}}} \in X_{/\sim_{\mathscr{B}}}, y_{/\sim_{\mathscr{B}}} \in Y_{/\sim_{\mathscr{B}}}$ such that $x_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} y_{/\sim_{\mathscr{B}}}$.

**Example 5. Subimages That Resemble Each Other**
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. Assume that $O$ is a set of images (each image is a set of points). Let $I1, I2 \subset O$ and let $I1_{/\sim_{\mathscr{B}}}, I2_{/\sim_{\mathscr{B}}}$ be partitions defined by $\sim_{\mathscr{B}}$. $I1_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} I2_{/\sim_{\mathscr{B}}}$ if, and only if there are subimages $X \subseteq I1_{/\sim_{\mathscr{B}}}, Y \subseteq I2_{/\sim_{\mathscr{B}}}$, where $X \underline{\bowtie}_{\mathscr{B}} Y$. That is, there are subimages $X, Y$ that resemble each other ($X, Y$ are near sets). Notice that $I1$ and $I2$ can either be the same image or two different images.

**Example 6. Sample Perceptually Near Image Partitions**
Consider images $Im3, Im4$ in Fig. 7.1, 7.4, respectively. Each shaded box in Figures 7.2, 7.5 have a uniform greylevel representing the greylevel of the pixels inside the shaded area. That is, since both $X, Y$ contain subimages represented by, for example, □ (light grey) shaded areas ($X, Y$ contain subimages with matching descriptions, namely, □).

From Def. 11, the partitions $Im3_{/\sim_{\mathscr{B}}}, Im4_{/\sim_{\mathscr{B}}}$ in Figures 7.2, 7.5 are perceptually near each other ($Im3_{/\sim_{\mathscr{B}}} \underline{\bowtie}_{\mathscr{B}} Im4_{/\sim_{\mathscr{B}}}$), since the subimages $X \subset Im1_{/\sim_{\mathscr{B}}}$ in Fig. 7.3 and $Y \subset Im2_{/\sim_{\mathscr{B}}}$ in Fig. 7.6 are examples of perceptually near sets, *i.e.* from Def. 4 and the presence of subimages with matching grey levels in both $X$ and $Y$, we conclude that $X \underline{\bowtie}_{\mathscr{B}} Y$.

**Example 7. Sample Near Image Coverings**
Consider images $X$ (Mona Lisa), $Y$ (Lena) in Fig. 7. Each shaded box in these images have average greylevels and average edge orientation for the pixels inside the parts of the images patches shown in both images. For this reason, $X$ is near (resembles) $Y$, *i.e.*, $X \underline{\underline{\bowtie}}_{\mathscr{B},\varepsilon} Y$ (this follows from Prop. 2).

## 5 Probe Functions for Image Features

This section briefly introduces probe functions for image features used in the CIBR experiments reported in this chapter.

### 5.1 *Edge Features*

Edge features include edge orientation ($e_o$) and edge intensity ($e_i$). Edge features are extracted using a wavelet-based multiscale edge detection method from [72]. This method involves calculation of the gradient of a smoothed image using wavelets, and defines edge pixels as those that have locally maximal gradient magnitudes in the direction of the gradient.

### 5.2 *Region Features*

Region features include both texture and colour from [73]. Hue and Saturation colour characteristics taken together are called Chromaticity. The texture features carry information about the relative position of pixels in an image rather than just the histogram of the intensities. Let G be the co-occurrence matrix whose element $g_{ij}$ is the number of times that pixel pairs with intensities $z_i$ and $z_j$ occur in image $f$ specified by operator $Q$ with $L$ possible intensity levels where $1 \leq i,\ j \leq L$. $Q$ defines position of two pixels relative to each other. $p_{ij}$ is the probability that a pair of points satisfying Q with intensities $z_i$ and $z_j$ defined as follows:

$$p_{ij} = \frac{g_{ij}}{n}, \text{where n is the total number of pixel pairs}$$

$m_r$ and $m_c$ are means computed along rows and columns respectively and $\sigma_r$ and $\sigma_c$ are standard deviations computed along rows and columns respectively of the normalized $G$. The following probe function for texture features are considered in this paper:

$$\phi_C = \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r \dot{\sigma}_c}, \text{Correlation}, \tag{2}$$

$$\phi_{Ct} = \sum_{i=1}^{K}\sum_{j=1}^{K} (i-j)^2 \dot{p}_{ij}, \text{Contrast}, \tag{3}$$

$$\phi_E = \sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij}^2, \text{Energy}, \tag{4}$$

$$\phi_H = \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{p_{ij}}{1+|i-j|}, \text{Homogeneity}, \tag{5}$$

$$\phi_h = \sum_{i=1}^{K}\sum_{j=1}^{K} \dot{p}_{ij}\mathrm{l\dot{o}g}_2 \dot{p}_{ij}, \text{Entropy}. \tag{6}$$

## 6 Nearness Measures

This section briefly presents the four nearness measures used to quantify the degree of resemblance between images. It should also be observed that $O$ in a perceptually representative space is viewed as a metric space with a metric defined in obvious ways by means of, for example, the $L_1$ norm from Def. 2. The approach to defining nearness methods in this section was first introduced in [11] and elaborated in [21].

The nearness methods introduced in this section are defined relative to some distance $D$ between images ($0 < D < D_{max}$). That is, a nearness measure $NM$ is defined as a function of distance, $NM = \nu(D)$, such that the $\nu(.)$ is monotonically decreasing (higher distance means lower nearness), $\nu(D_{max}) = 0$ and $\nu(0) = 1$. The following function is used when there is a finite maximum value ($D_{max}$) for $D$.

$$NM = \nu(D) = 1 - \left(\frac{D}{D_{max}}\right)^{\gamma}, \tag{7}$$

where $D/D_{max}$ is the normalized distance between 0 and 1 and $\gamma$ is a scaling factor. The scaling factor $\gamma$ is introduced here to modify distribution of distances around zero ($\gamma < 1$) or around one ($\gamma > 1$). A value of $\gamma = 0.5$ is considered to prevent the vales of $NM$ from being very close to 1. In cases where there is no maximum finite distance or when $D_{max} >> 0$, the following function is used to obtain nearness measures between 0 ($D \longrightarrow \infty$) and 1 ($D = 0$).

$$NM = \nu(D) = \frac{1}{1+D} \tag{8}$$

## 6.1 Tolerance Representative Space-Based Nearness Measures

*Tolerance nearness measure* (tNM) and *tolerance covering nearness measure* (tcNM) are based on the idea that if one considers the union of two images as the set of perceptual objects, tolerance classes should contain almost equal number of subimages from each image. In other words, a tolerance class which is defined over the union of two images should cover both images equally. The name tolerance covering nearness measure comes from this interpretation. The tolerance nearness measure (tNM) was first introduced by Henry and Peters [14]. In this paper, a more simple interpretation of tNM is introduced and also a modified version of the measure leads to a tolerance covering nearness measure (tcNM) that introduced by A.-H. Meghdadi [21].

Let $O$ denote a set of finite, non-empty set image patches and let $X,Y \subset O$ denote disjoint sets of image patches in individual digital images. Consider perceptual tolerance representative space $\langle O, \cong_{\mathscr{B},\varepsilon} \rangle$ defined with a tolerance relation $\cong_{\phi,\varepsilon}$ on $X,Y$ relative to a set of probe functions $\mathscr{B}$ and $\varepsilon \in (0,+\infty)$ and consider the covering of tolerance classes on a set $X \cup Y$ for $X,Y \subset O$ determined by $\cong_{\phi,\varepsilon}$. Let $\mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)$ denote the family of all tolerance classes of relation $\cong_{\mathscr{B}}$ on the set $X \cup Y$. The $tNM_{\cong_{\mathscr{B},\varepsilon}}$ nearness measure (introduced in [74], elaborated in [75]) estimates the degree of resemblance between $X$ and $Y$. This measure is defined in (9) as the weighted average of the closeness between the cardinality (size) of sets $A \cap X$ and $A \cap Y$ where $A \in \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)$ and the cardinality of tolerance class $A$ is used as the weighting factor.

$$tNM_{\cong_{\mathscr{B},\varepsilon}}(X,Y) = \frac{\sum\limits_{A \in \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)} \mathscr{T} \cdot |A|}{\sum\limits_{A \in \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)} |A|}, \tag{9}$$

$$\mathscr{T} = \frac{\min\{\,|A \cap X|, |A \cap Y|\,\}}{\max\{\,|A \cap X|, |A \cap Y|\,\}}. \tag{10}$$

The tcNM nearness measure, on the other hand, is defined in two steps. First, a tolerance covering distance measure (tcDM) is defined in (11) as a measure of difference between the intersection of tolerance classes with either images. Then, tcNM in (12) is defined by converting the tcDM distance to a measurement of nearness using (7).

$$tcDM = \sum_{A \in \mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)} \frac{|A \cap X| \; - \; |A \cap Y|}{|A \cap X| \; + \; |A \cap Y|}, \tag{11}$$

$$tcNM = 1 - \sqrt{\frac{tcDM}{|\mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)|}}, \tag{12}$$

where $|\mathrm{H}^{\varepsilon}_{\mathscr{B}}(X \cup Y)|$ is the total number of tolerance classes defined over the covering of $X \cup Y$ using $\cong_{\mathscr{B},\varepsilon}$.

## 6.2 Hausdorff Nearness Measure ($H_dNM$)

Hausdorff distance [47, 48, 49, 54] is defined between two finite point sets in a metric space. Assume $d(x,y)$ is a distance defined between points $x$ and $y$ in a metric space. Let $X$ and $Y$ be sets of points in the space. The Hausdorff distance $\rho_H(X,Y)$ between sets $X$ and $Y$ [54] is defined in (13). Then to such a space, we apply the Hausdorff distance between sets in measuring the resemblance between sets.

$$\rho_H(X,Y) = \max\{d_H(X,Y), d_H(Y,X)\}, \tag{13}$$

where

$$d_H(X,Y) = \max_{x\in X}\{\min_{y\in Y}\{d(x,y)\}\}, \tag{14}$$

$$d_H(Y,X) = \max_{y\in Y}\{\min_{x\in X}\{d(x,y)\}\}. \tag{15}$$

$d_H(X,Y)$ and $d_H(Y,X)$ are directed Hausdorff distances from $X$ to $Y$ and from $Y$ to $X$, respectively. In comparing images $X,Y$, a Hausdorff nearness measure $H_dNM(X,Y) = \frac{1}{1+\rho_H(X,Y)}$. The Hausdorff nearness measure $H_dNM$ is typically used to find a part of a test image that matches a given query or template image.

## 6.3 Generalized Mahalanobis Nearness Measure (gMNM)

The generalized Mahalanobis nearness measure (gMNM) was first introduced in [76]. The Mahalanobis distance [51] is a form of distance between two points in the feature space with respect to the variance of the distribution of points. The original Mahalanobis distance is usually defined between two sample multivariate vectors **x** and **y** as follows [77].

$$D_M(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T \Sigma^{-1}(\mathbf{x}-\mathbf{y}), \tag{16}$$

where the vectors are assumed to have a normal multivariate distribution with the covariance matrix $\Sigma$. This formula is usually used to measure the distance $D_M(\mathbf{x},\mathbf{m})$ between a vector **x** and the mean of the distribution **m**. Following the same approach, the Mahalanobis distance can be used to define a distance measure between two separate distributions. Let us assume $\chi_1 = (\Sigma_1, \mathbf{m_1})$ and $\chi_2 = (\Sigma_2, \mathbf{m_2})$ are two normal multivariate distributions with means $\mathbf{m_1}$,$\mathbf{m_1}$ and covariance matrices $\Sigma_1, \Sigma_2$. Moreover, assume that $P(\omega_1)$ and $P(\omega_2)$ represent prior probabilities of the given distributions. A generalized Mahalanobis distance between the two distributions is defined as follows [78, 77].

$$gMD = \sqrt{(\mathbf{m_1}-\mathbf{m_2})^T \Sigma_W^{-1}(\mathbf{m_1}-\mathbf{m_2})}, \tag{17}$$

where $\Sigma_W^{-1}$ refers to the within-class covariance matrix defined as in equation 18

$$\Sigma_W = \sum_{i=1,2} \left( P(\omega_i) \sum_{x \in \chi_i} \frac{(x-m_i)(x-m_i)^T}{n_i} \right). \tag{18}$$

Therefore, a generalized Mahalanobis distance-based nearness measure (gMNM) between two images is defined as follows. Let $X$ and $Y$ denote sets of perceptual objects (images). Let $\bar{\Phi}_X$ and $\bar{\Phi}_Y$ represent the mean feature vector for all the perceptual objects $x \in X$ and $y \in Y$, respectively. Also, let $\Sigma_X$ and $\Sigma_Y$ be the covariance matrices of the multivariate distributions of $\Phi_X$ and $\Phi_Y$ (feature values), respectively. Then

$$gMD(X,Y) = \sqrt{(\bar{\Phi}_X - \bar{\Phi}_Y)^T \Sigma_{X,Y}^{-1} (\bar{\Phi}_X - \bar{\Phi}_Y)}, \tag{19}$$

$$gMNM(X,Y) = \frac{1}{1+gMD(X,Y)}, \tag{20}$$

where

$$\Sigma_{X,Y} = \frac{1}{2}\left(\Sigma_X + \Sigma_Y\right). \tag{21}$$

## 7 Illustration: Image Nearness Measures

We illustrate the application of the image nearness measures in an image retrieval context with images drawn from the SIMPLIcity database [79] and using the Meghdadi toolset [11]. Table 1 includes pair-wise comparison of the query image numbered *420* with sample test images drawn from different categories such as dinosaurs, buildings and so on (see section 7.4 for a complete list). The experiments include a number of different image features for various subimage sizes ($p$). The notation for image features used in experiments are given next.

$$\begin{aligned}
\mathscr{T} &= \text{texture features denoted by CCtEH from Sec. 5.2,}\\
h &= \text{entropy,}\\
e_i &= \text{edge intensity,}\\
e_o &= \text{edge orientation,}\\
e_{io} &= \text{edge intensity and edge orientation taken together,}\\
\mathscr{C} &= \text{chromaticity feature with hue and saturation taken together.}
\end{aligned}$$

For conciseness (column compression), symbols represented feature are concatenated in Table 1. For example, $\mathscr{T}e_{io}h$ denotes the fact that the texture features (correlation, contrast, energy, homogeneity, entropy) and edge features (edge intensity and edge orientation) are used to compute nearness measures for a pair of images. The measurements reflect various subimage sizes from pixel matrices ranging in size from 20 × 20 to 5 × 5. Our choice of an 8 × 8 matrix was influenced by the retrieval quality of the nearness measurements and also the computation time. Image retrieval typically involves comparison of a large number of images and the issue of

**Table 1** Nearness Measurements

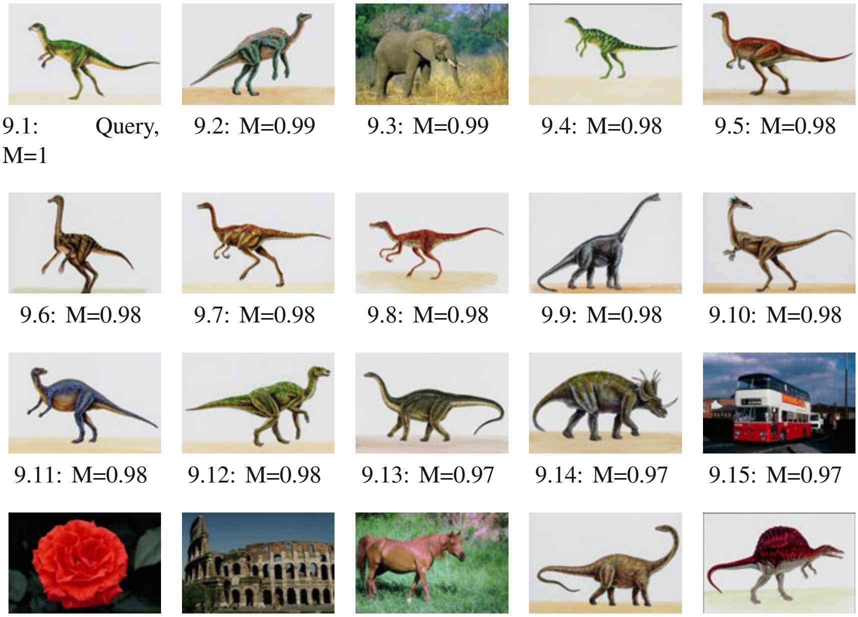
| p | Feature | Images | $gMNM$ | $H_dNM$ | p | Feature | Images | $gMNM$ | $H_dNM$ |
|---|---|---|---|---|---|---|---|---|---|
| **20** | $\mathscr{T}e_o$ | **420,423** | 0.61 | 0.80 | 8 | grey$e_{io}$ | 420,304 | 0.10 | 0.75 |
| 5 | $\mathscr{T}e_o$ | 420,423 | 0.82 | 0.90 | 8 | grey | 420,304 | 0.10 | 0.86 |
| **5** | $\mathscr{T}e_{io}$ | **420,423** | 0.81 | 0.87 | 10 | $\mathscr{T}e_{io}$h | 420,474 | 0.89 | 0.81 |
| 10 | $\mathscr{T}e_{io}$ | 420,474 | 0.97 | 0.83 | 5 | $\mathscr{T}e_{io}$h | **420,423** | 0.75 | 0.86 |
| 5 | $\mathscr{T}$ | 420,474 | 0.99 | 0.96 | 5 | $\mathscr{T}e_{io}$h$\mathscr{C}$ | **420,423** | 0.4 | 0.76 |
| 5 | $\mathscr{T}e_{io}$ | 420,474 | 0.95 | 0.75 | 5 | $\mathscr{T}e_{io}$h$\mathscr{C}$ | 420,460 | 0.90 | 0.75 |
| 5 | $\mathscr{T}e_o$ | 420,432 | 0.90 | 0.85 | 5 | $\mathscr{T}e_{io}$h | 420,460 | 0.90 | 0.75 |
| 5 | $\mathscr{T}e_{io}$ | 420,432 | 0.90 | 0.85 | 5 | $\mathscr{T}e_{io}$ | 420,460 | 0.90 | 0.76 |
| 5 | $\mathscr{T}e_{io}$h | 420,432 | 0.79 | 0.83 | 5 | $\mathscr{T}e_o$ | 420,460 | 0.91 | 0.79 |
| 5 | $\mathscr{T}e_{io}$h$\mathscr{C}$ | 420,432 | 0.60 | 0.715 | 5 | $\mathscr{T}e_{io}$h | 420,456 | 0.24 | 0.74 |
| 5 | $\mathscr{T}e_o$ | 420,436 | 0.85 | 0.87 | 5 | $\mathscr{T}e_{io}$ | 420,456 | 0.24 | 0.74 |
| 5 | $\mathscr{T}e_{io}$ | 420,436 | 0.85 | 0.83 | 5 | $\mathscr{T}e_o$ | 420,456 | 0.24 | 0.74 |
| 5 | $\mathscr{T}e_{io}$h | 420,436 | 0.78 | 0.80 | 5 | $\mathscr{T}e_{io}$h$\mathscr{C}$ | 420,436 | 0.36 | 0.68 |
| 8 | $\mathscr{T}e_{io}$ | 420,478 | 0.77 | 0.83 | 8 | $e_{io}$ | 420,478 | 0.79 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,484 | 0.94 | 0.86 | 8 | $e_{io}$ | 420,484 | 0.95 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,424 | 0.73 | 0.81 | 8 | $e_{io}$ | 420,424 | 0.74 | 0.96 |
| 8 | $\mathscr{T}e_{io}$ | 420,473 | 0.94 | 0.82 | 8 | $e_{io}$ | 420,473 | 0.99 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,490 | 0.87 | 0.79 | 8 | $e_{io}$ | 420,490 | 0.95 | 0.99 |
| 8 | $\mathscr{T}e_{io}$ | 420,514 | 0.128 | 0.72 | 8 | $e_{io}$ | 420,514 | 0.42 | 0.99 |
| 8 | $\mathscr{T}e_{io}$ | 420,703 | 0.07 | 0.76 | 8 | $e_{io}$ | 420,703 | 0.34 | 0.97 |
| 8 | $\mathscr{T}e_{io}$ | 420,600 | 0.32 | 0.72 | 8 | $e_{io}$ | 420,600 | 0.48 | 0.97 |
| 8 | $\mathscr{T}e_{io}$ | 420,200 | 0.29 | 0.77 | 8 | $e_{io}$ | 420,200 | 0.37 | 0.97 |
| 8 | $\mathscr{T}e_{io}$ | 420,304 | 0.34 | 0.74 | 8 | $e_{io}$ | 420,304 | 0.48 | 0.97 |
| 8 | $\mathscr{T}e_{io}$ | 420,499 | 0.93 | 0.85 | 8 | $e_{io}$ | 420,499 | 0.95 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,408 | 0.61 | 0.78 | 8 | $e_{io}$ | 420,408 | 0.73 | 0.97 |
| 8 | $\mathscr{T}e_{io}$ | 420,410 | 0.71 | 0.75 | 8 | $e_{io}$ | 420,410 | 0.77 | 0.96 |
| **8** | te$_{io}$ | **420,423** | 0.84 | 0.85 | 8 | $e_{io}$ | **420,423** | 0.93 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,432 | 0.88 | 0.81 | 8 | $e_{io}$ | 420,432 | 0.95 | 0.9 |
| 8 | $\mathscr{T}e_{io}$ | 420,436 | 0.79 | 0.78 | 8 | $e_{io}$ | 420,436 | 0.94 | 0.97 |
| 8 | $\mathscr{T}e_{io}$ | 420,456 | 0.93 | 0.84 | 8 | $e_{io}$ | 420,456 | 0.98 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,460 | 0.96 | 0.86 | 8 | $e_{io}$ | 420,460 | 0.99 | 0.98 |
| 8 | $\mathscr{T}e_{io}$ | 420,474 | 0.95 | 0.85 | 8 | $e_{io}$ | 420,474 | 0.95 | 0.98 |

computation time is important. It can also be observed that adding entropy and chromaticity features does not add to the discriminatory power of nearness measures.

## *7.1 Analysis of Hausdorff Nearness Measure: Image Retrieval Experiments*

In this section, we discuss the significance of the Hausdorff nearness measure values drawn from Table 1.

Fig. 9 shows 20 images (1 query and 19 test) ordered by their $H_dNM$ values using only the $e_{io}$ edge features. It can be observed that i) the difference between the measure values of the nearest and furthest images is very small (0.99 vs 0.96) and ii) images 9.3, 9.19 and 9.20 are also out of sequence. Fig. 10 shows the same 20 images ordered by their $H_dNM$ values using both edge features and texture features $\mathscr{T}e_{io}$. It is noteworthy that all images belonging to the same category as the query



9.1: Query, M=1 9.2: M=0.99 9.3: M=0.99 9.4: M=0.98 9.5: M=0.98

9.6: M=0.98 9.7: M=0.98 9.8: M=0.98 9.9: M=0.98 9.10: M=0.98

9.11: M=0.98 9.12: M=0.98 9.13: M=0.97 9.14: M=0.97 9.15: M=0.97

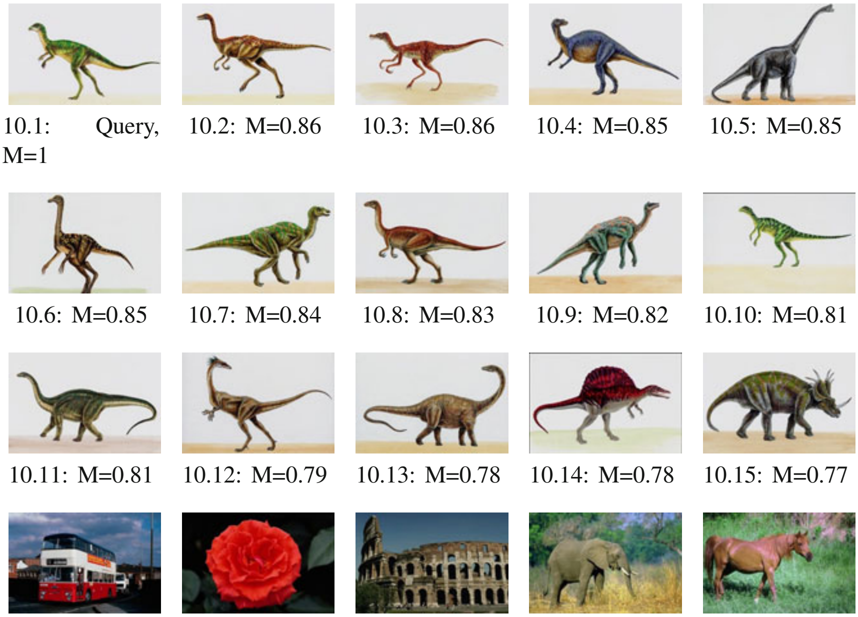
9.16: M=0.97 9.17: M=0.97 9.18: M=0.97 9.19: M=0.96 9.20: M=0.96

**Fig. 9** Images ordered by $H_dNM$ measure values with Edge Features

image are now retrieved in the proper sequence. Also the difference between the measure values of the nearest and furthest images shows some improvement(0.86 vs 0.72).

## 7.2 *Analysis of Generalized Mahalanobis Nearness Measure: Image Retrieval Experiments*

In this section, we discuss the significance of the generalized Mahalanobis nearness measure values drawn from Table 1. Fig. 11 shows 20 images (1 query and 19 test images) ordered by their $gMNM$ values using only the edge features $e_{io}$. It can be observed that i) the difference between the measure values of the nearest and furthest images is fairly large (0.99 vs 0.34) compared with $H_dNM$ measure ii) all images belonging to the same category as the query image are retrieved in the proper sequence. Fig. 12 shows the same 20 images ordered by their $gNM$ values using both edge features and texture features $\mathscr{T}e_{io}$. It can be observed that the difference between the measure values of the nearest and furthest images (0.96 vs. 0.07) is considerable, when additional features are added and that the texture features contribute significantly to the quality of the image retrieval.



10.1: Query, M=1

10.2: M=0.86

10.3: M=0.86

10.4: M=0.85

10.5: M=0.85

10.6: M=0.85

10.7: M=0.84

10.8: M=0.83

10.9: M=0.82

10.10: M=0.81

10.11: M=0.81

10.12: M=0.79

10.13: M=0.78

10.14: M=0.78

10.15: M=0.77

10.16: M=0.76

10.17: M=0.75

10.18: M=0.74

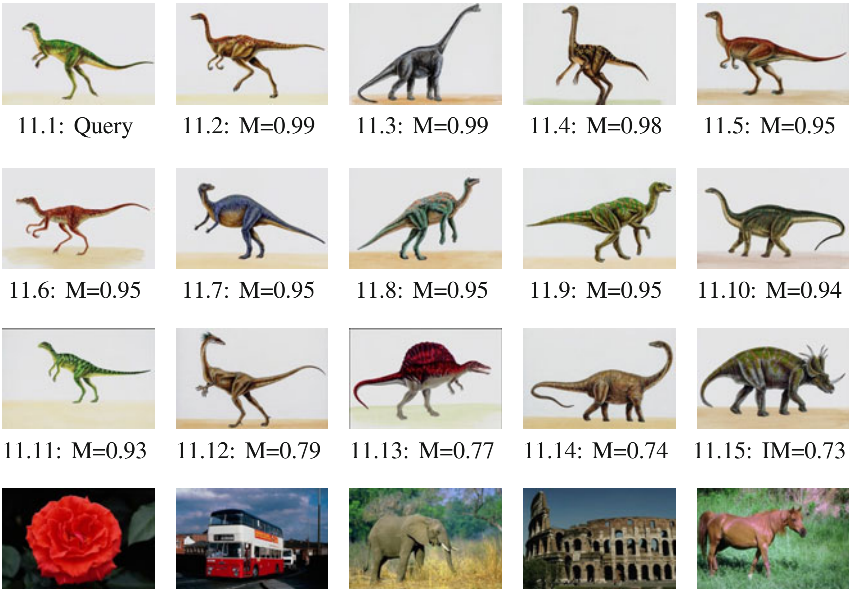
10.19: M=0.72

10.20: M=0.72

**Fig. 10** Images ordered by $H_dNM$ measure values with Edge and Texture Features

## *7.3 Remarks on Quality of Retrieval*

The observations in sections 7.2 and 7.1 reveal that the generalized Mahalanobis nearness measure is a better measure in terms of its discriminatory power and its stability. The Hausdorff measure calculates distance between two sets of feature-valued vectors extracted from subimages in two images using the $L_1$ norm. In addition, the edge intensity and edge orientation features are not sufficient to get a good match between the query image and the test images, even though the measure is modified to compensate for the effect of outliers (see eqns. 14 and 15). The Mahalanobis measure calculates distance between feature vectors with respect to the covariance of the multivariate distribution of mean feature values extracted from subimages in two images. It can be seen that with just the edge intensity and edge orientation features, the Mahalanobis nearness measure is able to find a good match for the query image. This is because this measure takes into account not only in-class co-variance, but also prior probabilities.



11.1: Query 11.2: M=0.99 11.3: M=0.99 11.4: M=0.98 11.5: M=0.95

11.6: M=0.95 11.7: M=0.95 11.8: M=0.95 11.9: M=0.95 11.10: M=0.94

11.11: M=0.93 11.12: M=0.79 11.13: M=0.77 11.14: M=0.74 11.15: IM=0.73

11.16: M=0.48 11.17: M=0.48 11.18: M=0.42 11.19: M=0.37 11.20: M=0.34

**Fig. 11** Ordered by gMNM-measure values with Edge Features

## *7.4 Performance Measure*

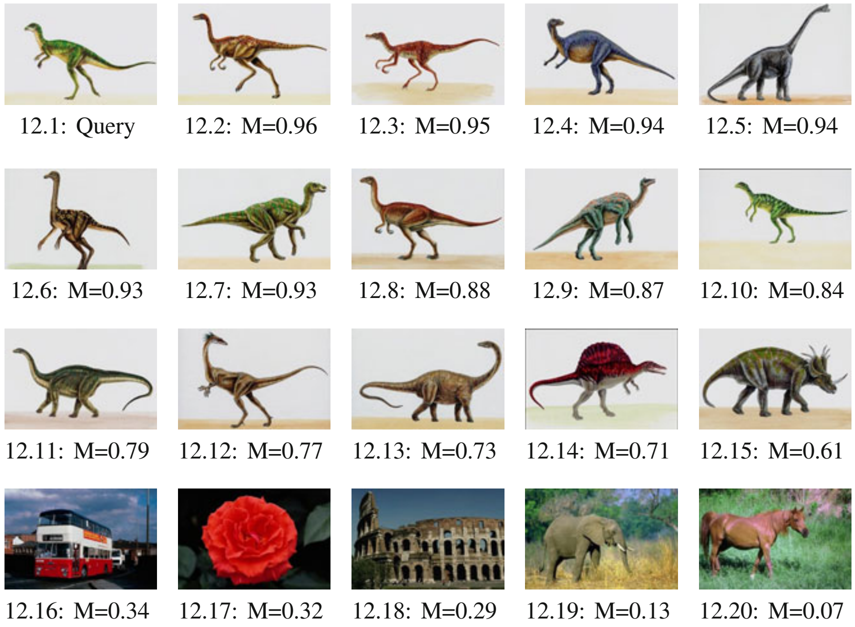
Several performance evaluation measures have been proposed based on the well-known precision(P) and the recall(R) to evaluate the performance of CBIR methods [80]. We use the following definitions for P and R:

$$P = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}}, \tag{22}$$

$$R = \frac{\text{Number of relevant images retrieved}}{\text{Total number of \ relevant images}}. \tag{23}$$

We have used a total of 578 images drawn from the SIMPLIcity database in the six categories (see Fig.9). Table 2 shows the average measure values for images in each category.

However, the individual pair-wise measure values vary within each category. In order to establish relevancy of the test images, a threshold $th$ for the nearness value which acts as a cut-off needs to be determined. Notice that the total number of relevant images should not be $> 100$ (size of the query image category set). Using this approach, we obtain the following results for $P$ and $R$ with $th = 0.6$:



12.1: Query 12.2: M=0.96 12.3: M=0.95 12.4: M=0.94 12.5: M=0.94

12.6: M=0.93 12.7: M=0.93 12.8: M=0.88 12.9: M=0.87 12.10: M=0.84

12.11: M=0.79 12.12: M=0.77 12.13: M=0.73 12.14: M=0.71 12.15: M=0.61

12.16: M=0.34 12.17: M=0.32 12.18: M=0.29 12.19: M=0.13 12.20: M=0.07

**Fig. 12** Images ordered by gMNM-measure values with Edge and Texture Features

$$P_{\mathscr{T}e_{io}} = \frac{100}{578} = 0.173 \text{ using Hausdorff } H_dNM, \tag{24}$$

$$R_{\mathscr{T}e_{io}} = \frac{100}{100} = 1.0 \text{ using Hausdorff } H_dNM, \tag{25}$$

$$P_{\mathscr{T}e_{io}} = \frac{99}{99} = 1.0 \text{ using Mahalanobis gMNM}, \tag{26}$$

$$R_{\mathscr{T}e_{io}} = \frac{99}{100} = 0.99 \text{ using Mahalanobis gMNM}. \tag{27}$$

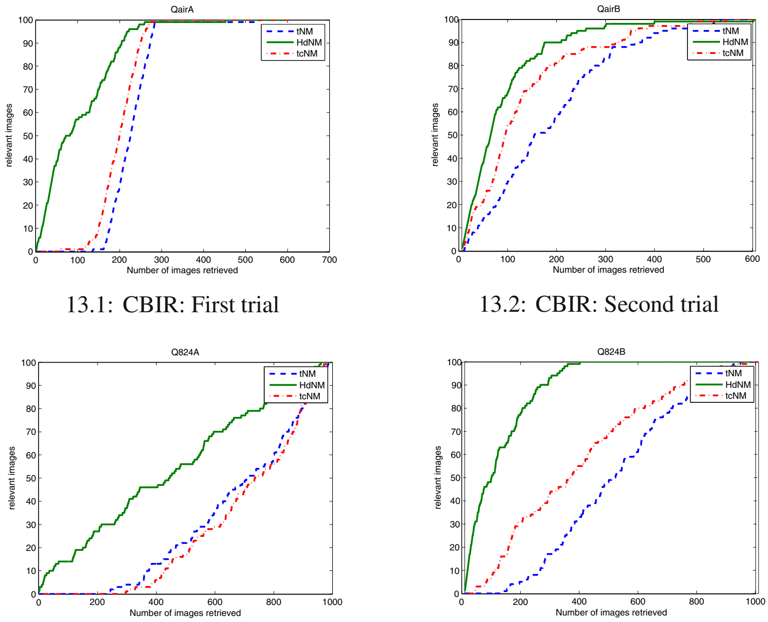
It is noteworthy that on a large sample of images, Mahalanobis *gMNM* distance measure is more precise than the Hausdorff $H_dNM$ distance measure for the same threshold value(see, *eqn.* (24)).

**Table 2** Nearness Measurements

| Category | Number of Images | *Average gMNM* | *Average$H_dNM$* |
|---|---|---|---|
| Building | 100 | 0.27 | 0.77 |
| Bus | 94 | 0.24 | 0.75 |
| Dinosaurs | 100 | 0.84 | 0.82 |
| Elephant | 100 | 0.19 | 0.74 |
| Flower | 84 | 0.31 | 0.73 |
| Horse | 100 | 0.12 | 0.74 |

**Table 3** Set of probe functions: $\mathscr{B} = \{\phi_1, \phi_2, ..., \phi_{11}\}$

| Probe function | Feature | Description |
|---|---|---|
| $\phi_1$ | Colour | Average grey level of pixels |
| $\phi_3, \phi_4, \phi_5$ | Colour | Red, Green and Blue colour components |
| $\phi_6$ | Shape | Average edge intensity |
| $\phi_7$ | Shape | Dominant edge orientation |
| $\phi_2$ | Texture | Entropy of the greylevel values |
| $\phi_8$ | Texture | Contrast |
| $\phi_9$ | Texture | Correlation |
| $\phi_{10}$ | Texture | Uniformity |
| $\phi_{11}$ | Texture | Homogeneity |



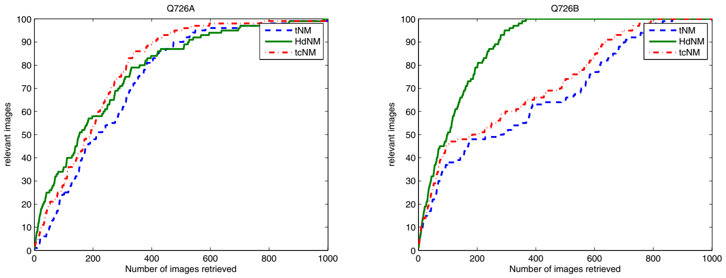
13.1: CBIR: First trial

13.2: CBIR: Second trial

13.3: CBIR: First trial

13.4: CBIR: Second trial

**Fig. 13** Comparison of CBIR results on Caltech & Simplicity



14.1: CBIR: First trial

14.2: CBIR: Second trial

**Fig. 14** Comparison of CBIR results on Simplicity

## 8 CBIR Experiments on Two Image Archives

In this section we report results of experiments on more substantial image archives. Two different types of trials were performed on all of the images in two separate image archives, with 600 images from the Caltech archive [81] and the 1000 images from the Simplicity archive [79]. Each query image is compared with all of the images in an image archive. The number of the most relevant images (*i.e.*, those images that most closely resemble the query image) in the first 100 images retrieved from each image archive are reported. The relevancy of a retrieved test image $Y$ in relation to a query image $X$ is determined by the degree of nearness of $X,Y$ computed using the $tNM, tcNM, H_dNM$ measures.

**Description of CBIR Retrieval Trials**

T.1 In this first trial, only shape features (edge intensity and edge orientation) with $\varepsilon = 0.2$ are used to determine coverings in each query image and test image pair. The set of probe functions $\mathscr{B} = \{\phi_6, \phi_7\}$ are listed in Table 3. In this first trial, the $tNM, tcNM, H_dNM$ measures are applied to each pair of image of image coverings.

T.2 In the second trial, colour, texture and shape features represented by the 11 different probe functions in Table 3 and $\varepsilon = 1.1$ are used to determine coverings in each query image and test image pair. In this second trial, the $tNM, tcNM, H_dNM$ measures are also applied to each pair of image of image coverings.

The trials reported in this section represent an application of Prop. 2, where the nearness of each pair of images is determined by a comparison of classes found in the coverings of the query and test images. The coverings themselves are viewed in the context of perceptual representative spaces that are specialized forms of Zeeman tolerance spaces and represent an application of Prop. 1.

The results of two trials are shown in Figures 13 and 14. With one exception, it can be observed that the Hausdorff nearness measure $H_dNM$ is more accurate in measuring the correspondence of each query image to the test images in both the Caltech and Simplicity image archives. In other words, for the results in five of the trials reported in Figures 13.1, 13.2, 13.3, 13.4, 14.2, the $H_dNM$ measure more accurately determines the extent that a query image image resembles a test image. The one exception in these results of Trial T.1 (comparison of shape features) can be observed in Fig. 14.1, where the $tNM, tcNM$ measures outperform the Hausdorff $H_dNM$ measure in determining the correspondence between an ROI query image and each of the test images.

## 9 Conclusion

This chapter introduces a form of image analysis defined within the context of Poincaré-Peters perceptual representative spaces and near sets. To preserve the representative heritage that began with J.H. Poincaré's representation of physical continua as set of similar sensations (*e.g.*, visual, tactile, audio), the term representative space (RS) has been used as a synonym for various forms of tolerance spaces. The Poincaré-Peters RS generalises Poincaré's original RS because it *represents* one or more features of sensation in determining the similarity of sensations. The focus in this chapter is on visual space containing points that are subimages. The new form of RS provides a framework for vision systems and CBIR.

It should also be mentioned that there is a clear distinction between a Zeeman tolerance space and a perceptual tolerance space. A Zeeman tolerance space does not take into account the features of objects in a visual space. By contrast, the focus of perceptual tolerance spaces is on tolerance relations defined in terms of one or more features of points in a visual space. Each set of image features leads to a particular perceptual tolerance space. Notice that Pawlak indiscernibility relations are specialized forms of tolerance relations inasmuch as an indiscernibility relation is reflexive and symmetric as well as transitive. Remarkably, a perceptual space $(X, \simeq_{\mathscr{B}})$ with a set of perceptual objects $X$ and perceptual indiscernible relation $\simeq_{\mathscr{B}}$ that determines a partition of $X$ that is highly useful in image analysis. This has been demonstrated in [8]. In general, however, a perceptual tolerance space $(X, \tau_{\mathscr{B}})$ with a set of perceptual objects $X$ and perceptual tolerance relation $\tau_{\mathscr{B}}$ has proved to be more useful in image analysis(see, *e.g.*, [9, 16]). This is the case because elements of one tolerance class often are also members of other tolerance classes in a cover determined by $\tau_{\mathscr{B}}$. This fact tends to make it easier to discern similarities between, for example, regions of a pair of digital images.

In the proposed approach to CBIR, image classes determined by a tolerance relation provide the content useful in image retrieval. The degree of image resemblance is determined by specialized forms of distance measures Hausdorff, Mahalanobis and tolerance nearness measures. Sample experiments reported in this chapter suggest that the proposed approach to CBIR is quite promising. Future work will include near set-based object recognition and more advanced forms of CBIR.

## References

1. Poincaré, J.: L'espace et la géomètrie. Revue de m'etaphysique et de morale 3, 631–646 (1895)
2. Poincaré, J.: Sur certaines surfaces algébriques; troisième complément 'a l'analysis situs. Bulletin de la Société de France 30, 49–70 (1902)
3. Peters, J.: Corrigenda and addenda: Tolerance near sets and image correspondence. Int. J. Bio-Inspired Computation 2(5), 310–318 (2010)
4. Peters, J.: Near sets. special theory about nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
5. Peters, J.: Near sets. general theory about nearness of objects. Applied Mathematical Sciences 1(53), 2029–2069 (2007)
6. Peters, J., Ramanna, S.: Affinities between perceptual granules: Foundations and perspectives. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling. SCI, vol. 182, pp. 49–66. Springer, Heidelberg (2009)
7. Ramanna, S., Meghdadi, A.H.: Measuring resemblances between swarm behaviours: A perceptual tolerance near set approach. Fundamenta Informatica 95(4), 533–552 (2009); ISSN: 0169-2968
8. Ramanna, S.: Discovering image similarities: Tolerance near set approach. In: Pal, S., Peters, J. (eds.) Rough Fuzzy Image Analysis, pp. 12.1–12.15. CRC Press, Boca Raton (2010)
9. Ramanna, S.: Perceptually near pawlak partitions. In: Peters, J.F., Skowron, A., Słowiński, R., Lingras, P., Miao, D., Tsumoto, S. (eds.) Transactions on Rough Sets XII. LNCS, vol. 6190, pp. 170–192. Springer, Heidelberg (2010)
10. Wasilewski, P., Peters, J.F., Ramanna, S.: Perceptual tolerance intersection. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 277–286. Springer, Heidelberg (2010)
11. Meghdadi, A.H., Peters, J.F., Ramanna, S.: Tolerance classes in measuring image resemblance. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009. LNCS, vol. 5712, pp. 127–134. Springer, Heidelberg (2009)
12. Peters, J.: Tolerance near sets and image correspondence. Int. J. of Bio-Inspired Computation 4(1), 239–445 (2009)
13. Peters, J., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
14. Hassanien, A., Abraham, A., Peters, J., Schaefer, G., Henry, C.: Rough sets and near sets in medical imaging: A review. IEEE Trans. Info. Tech. in Biomedicine 13(6), 955–968 (2009), doi:10.1109/TITB.2009.2017017
15. Peters, J. C.: Near sets. Wikipedia (2009), `http://en.wikipedia.org/wiki/Near_sets`
16. Pal, S., Peters, J.: Rough Fuzzy Image Analysis: Foundations and Methodologies. CRC Press, Boca Raton (2010) ISBN 13: 9781439803295, ISBN 10:1439803293
17. Peters, J., Wasilewski, P.: Foundations of near sets. An International Journal Information Sciences 179(18), 3091–3109 (2009), doi:10.1016/j.ins.2009.04.018.
18. Henry, C., Peters, J.: Perception-based image analysis. Int. J. of Bio-Inspired Computation 2(2) (2010) (in press)
19. Deselaers, T.: Image Retrieval, Object Recognition, and Discriminative Models. Ph.d. thesis, RWTH Aachen University (2008)
20. Henry, C., Peters, J.F.: Perceptual image analysis. International Journal of Bio-Inspired Computation 2(3), 271–281 (2010)

21. Meghdadi, A.H., Peters, J.: Perceptual systems approach to measuring image resemblance. In: Pal, S., Peters, J. (eds.) Rough Fuzzy Image Analysis, pp. 8.1–8.23. CRC Press, Boca Raton (2010)
22. Henry, C., Peters, J.F.: Perception based image classification. Technical Report TR-2009-016, Computational Intelligence Laboratory, University of Manitoba, UM CI Laboratory Technical Report No. TR-2009-016 (2009)
23. Veltkamp, R.C.: State-of-the-Art in Content-Based Image and Video Retrieval. In: Computational Imaging and Vision. Kluwer Academic Publishers, Dordrecht (2001)
24. Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: An experimental comparison. Information Retrieval 11(1), 77–107 (2008d)
25. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)
26. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. Pattern Recognition 40(1), 262–282 (2007), doi:10.1016/j.patcog.2006.04.045
27. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology 8(5), 644–655 (1998)
28. Su, Z., Zhang, H., Li, S.: Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. IEEE Transactions on Image Processing 12(8), 924–937 (2003)
29. Matthieu, C., Philippe, H.G., Sylvie, P.-F.: Stochastic exploration and active learning for image retrieval. Image and Vision Computing 25(1), 14–23 (2007)
30. Christos, T., Nikolaos, A.L., George, E., Spiros, F.: On the perceptual organization of image databases using cognitive discriminative biplots. EURASIP Journal on Advances in Signal Processing, doi:10.1155/2007/68165
31. Fechner, G.: Elemente der Psychophysik. Elements of Psychophysics, Adler, H.E (trans.). Holt, Rinehart & Winston, London, UK (1860)
32. Pawlak, Z.: Classification of objects by means of attributes. Polish Academy of Sciences 429 (1981)
33. Pawlak, Z.: Rough sets. International J. Comp. Inform. Science 11, 341–356 (1982)
34. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
35. Mrózek, A., Plonka, L.: Rough sets in image analysis. Foundations of Computing and Decision Sciences F18(3-4), 268–273 (1993)
36. Pal, S., Mitra, P.: Multispectral image segmentation using rough set initialized em algorithm. IEEE Transactions on Geoscience and Remote Sensing 11, 2495–2501 (2002)
37. Peters, J., Borkowski, M.: k-means indiscernibility over pixels. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 580–585. Springer, Heidelberg (2004)
38. Pal, S., Shankar, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 401–416 (2005)
39. Borkowski, M., Peters, J.: Matching 2d image segments with genetic algorithms and approximation spaces. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS (LNAI), vol. 4100, pp. 63–101. Springer, Heidelberg (2006)
40. Borkowski, M.: 2D to 3D Conversion with Direct Geometrical Search and Approximation Spaces. PhD thesis, Dept. Elec. Comp. Engg. (2007), http://wren.ee.umanitoba.ca/

41. Maji, P., Pal, S.: Maximum class separability for rough-fuzzy c-means based brain mr image segmentation. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 114–134. Springer, Heidelberg (2008)
42. Mushrif, M., Ray, A.: Color image segmentation: Rough-set theoretic approach. Pattern Recognition Letters 29(4), 483–493 (2008)
43. Sen, D., Pal, S.: Generalized rough sets, entropy, and image ambiguity measures. IEEE Transactions on Systems, Man, and Cybernetics–PART B 39(1), 117–128 (2009)
44. Malyszko, D., Stepaniuk, J.: Standard and fuzzy rough entropy clustering algorithms in image segmentation. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 409–418. Springer, Heidelberg (2008)
45. Giannopoulos, P., Veltkamp, R.: A pseudo-metric for weighted point sets. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 715–730. Springer, Heidelberg (2002)
46. Rubner, Y.: Perceptual Metrics for Image Database Navigation. PhD thesis, Stanford University (1999)
47. Hausdorff, F.: Grundzüge der mengenlehre. Verlag Von Veit & Comp., Leipzig (1914)
48. Hausdorff, F.: Set theory. Chelsea Publishing Company, New York (1962)
49. Rogers, C.: Hausdorff Measures. Cambridge U Press, Cambridge (1970)
50. Mahalanobis, P.: On tests and measures of group divergence i. theoretical formulae. J. and Proc. Asiat. Soc. of Bengal 26, 541–588 (1930)
51. Mahalanobis, P.: On the generalized distance in statistics. Proc. Nat. Institute of Science (Calcutta) 2, 49–55 (1936)
52. Peters, J.F.: Classification of objects by means of features. In: Proc. IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SCCI 2007), Honolulu, Hawaii, pp. 1–8 (2007)
53. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
54. Engelking, R.: General topology. Sigma series in pure mathematics. Heldermann Verlag, Berlin (1989)
55. Pavel, M.: Fundamentals of Pattern Recognition, 2nd edn. Marcel Dekker, Inc., N.Y (1993)
56. Schroeder, M., Wright, M.: Tolerance and weak tolerance relations. Journal of Combinatorial Mathematics and Combinatorial Computing 11, 123–160 (1992)
57. Sossinsky, A.: Tolerance space theory and some applications. Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications 5(2), 137–167 (1986)
58. Pawlak, Z., Peters, J.: Jak blisko (how near). Systemy Wspomagania Decyzji I, 57, 109 (2002, 2007); ISBN 83-920730-4-5
59. Naimpally, S., Warrack, B.: Proximity Spaces. In: Cambridge Tract in Mathematics, vol. (59). Cambridge Univiversity Press, Cambridge (1970)
60. Mozzochi, C., Naimpally, S.: Uniformity and proximity, Allahabad, India. Allahabad Mathematical Society Lecture Note Series, vol. 2 p. xii+153 (2009); ISBN 978-81-908159-1-8
61. Naimpally, S.: Proximity approach to problems in topology and analysis, pp. xiv+206. Oldenbourg Verlag, Munich (2009); ISBN 978-3-486-58917-7
62. Duntsch, I., Orlowska, E.: A discrete duality between apartness algebras and apartness frames. Technical Report Technical Report # CS-08-02, Computer Science Department, Brock University (2008)
63. Naimpally, S., Warrack, B.: Proximity Spaces. Cambridge Tract in Mathematics, vol. (59). Cambridge University Press, Cambridge (1970)

64. DiMaio, G., Naimpally, S.: D-proximity spaces. Czech. Math. J. 41(116), 232–248 (1991)
65. DiMaio, G., Naimpally, S.: Proximity approach to semi-metric and developable spaces. Pacific J. Math. 44, 93–105 (1973)
66. Efremovič, V.: Infinitesimal spaces. Dokl. Akad. Nauk SSSR 76, 341–343 (1951)
67. Efremovič, V.: The geometry of proximity. Mat. Sb. 31, 189–200 (1952)
68. Efremovič, V., Švarc, A.: A new definition of uniform spaces. metrization of proximity spaces. Dokl. Akad. Nauk SSSR 89, 393–396 (1953)
69. Pták, P., Kropatsch, W.: Nearness in digital images and proximity spaces. In: Nyström, I., Sanniti di Baja, G., Borgefors, G. (eds.) DGCI 2000. LNCS, vol. 1953, pp. 69–77. Springer, Heidelberg (2000)
70. Henry, C., Peters, J.: Image pattern recognition using approximation spaces and near sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 475–482. Springer, Heidelberg (2007)
71. Henry, C., Peters, J.: Near set evaluation and recognition (near) system. Technical report, Computationa Intelligence Laboratory, University of Manitoba, UM CI Laboratory Technical Report No. TR-2009-015 (2009)
72. Mallat, S., Zhong, S.: Characterization of signals from multiscale edges. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(7), 710–732 (1992)
73. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice-Hall, Upper Saddle Rv (2002); NJ 07458, ISBN 0-20-118075-8
74. Henry, C., Peters, J.F.: Near set index in an objective image segmentation evaluation framework. In: GEOgraphic Object Based Image Analysis: Pixels, Objects, Intelligence, pp. 1–6. University of Calgary, Alberta (2008)
75. Henry, C.: Near set evaluation and recognition (near) system. In: Pal, S., Peters, J.F. (eds.) Rough Fuzzy Image Analysis. Foundations and Methodologies, ch. 7, pp. 7.1–7.22. CRC Press, Boca Raton (2010)
76. Meghdadi, A., Peters, J.: Content-based image retrieval using a tolerance near set approach to image similarity. Image and Vision Computing (2009) (under review)
77. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)
78. Arai, R., Watanabe, S.: A quantitative method for comparing multi-agent-based simulations in feature space. In: David, N., Sichman, J.S. (eds.) MAPS 2008. LNCS, vol. 5269, pp. 154–166. Springer, Heidelberg (2009)
79. Wang, J.Z.: Simplicity-content-based image search engine. Content Based Image Retrieval Project (1995-2001), `http://wang.ist.psu.edu/IMAGE`
80. Muller, H., Muller, W., Squire, D., Marchand-Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: Overview and proposals. Pattern Recognition Letters 22(5), 593–601 (2001)
81. Caltech, C.V.G.: Image archives of computational vision group (2005), `http://www.vision.caltech.edu/html-files/archive.html`

# Chapter 9
# Local Keypoints and Global Affine Geometry: Triangles and Ellipses for Image Fragment Matching

Mariusz Paradowski and Andrzej Śluzek

**Abstract.** Image matching and retrieval is one of the most important areas of computer vision. The key objective of image matching is detection of near-duplicate images. This chapter discusses an extension of this concept, namely, the retrieval of near-duplicate image fragments. We assume no *a'priori* information about visual contents of those fragments. The number of such fragments in an image is also unknown. Therefore, we address the problem and propose the solution based purely on visual characteristics of image fragments The method combines two techniques: a local image analysis and a global geometry synthesis. In the former stage, we analyze low-level image characteristics, such as local intensity gradients or local shape approximations. In the latter stage, we formulate global geometrical hypotheses about the image contents and verify them using a probabilistic framework.

## 1 Introduction

Local image features have always been one of the fundamental mechanisms in machine vision. Although a wide selection of diversified local features exist, we are particularly interested in *keypoints* (also referred to as *interest points*). The concept of keypoints appeared almost 30 years ago (e.g. [7, 19]) but their main idea has remained unchanged until today. In general, keypoints indicate image fragments with distinctive visual characteristics. It is assumed that the characteristics are so prominent that whenever the same objects/scenes appear in another image, most of such fragments would be again detected as keypoints. Therefore, by matching keypoints with similar characteristics, the local similarities between images can be established.

Mariusz Paradowski
School of Computer Engineering, Nanyang Technological University, Singapore
Institute of Informatics, Wroclaw University of Technology, Poland

Andrzej Śluzek
School of Computer Engineering, Nanyang Technological University, Singapore
Nicolaus Copernicus University, Faculty of Physics, Astronomy and Informatics, Poland

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 195–224.
springerlink.com

Originally, keypoints were proposed primarily for stereovision, [19], where the problem of local correspondences was critical for a proper depth estimation. Those keypoints were relatively simple *corner points* detected over small areas of fixed size (e.g. [7, 19]). In the following years, however, the importance of keypoints in a more general task of image matching (including image search and retrieval) has been identified (e.g. [23]). For such applications, more sophisticated types of keypoints are needed. These keypoints, apart from being insensitive to illumination and contrast variations, should remain invariant (at least approximately) under shape deformations typically encountered in 2D visualization of 3D scenes. Thus, scale-invariant keypoint detectors have been proposed (e.g. [2, 13]) followed by affine-invariant (approximately or fully) detectors (e.g. [16, 20]). Such keypoints are usually represented by circles or ellipses indicating the keypoint's scale/shape so that the term *key regions* seems more appropriate. There also exist keypoint detectors that are directly based on the local shapes (e.g. MSER proposed in [15]) for which the elliptical approximations are more intuitive.

Keypoints are almost universally represented by $n$-dimensional descriptors (characterizing selected local properties of image intensities or colours) so that visual similarities between keypoints can be measured as distances between points in $n$-dimensional spaces. Various keypoint descriptors have been proposed (e.g. [1, 2, 9, 11, 14, 18, 28]) and benchmarked (e.g. [17]).

In this work, we are not particularly concerned about the type of keypoint detectors and descriptors used, although certain preferences are highlighted in Section 2. We just exploit the fact that hundreds (or even thousands) of keypoints typically detected in a single image provide a large amount of (mostly) stable and (usually) useful visual data. Therefore, keypoint-based image matching, search and retrieval have recently gained popularity in highly diversified applications (ranging from video browsing, e.g. [32], satellite image processing, e.g. [27], to urban navigation systems, e.g. [29]).

The major difficulties is such applications arise from the lack of contextual factors in the individual keypoint matching. In other words, visually similar keypoints can represent a similar/identical object only if the similarity is consistent within a wider context. The consistency is usually modeled by a geometric distortion uniformly transforming a group of matching keypoints from one image to another. Typically considered mappings include scaling+rotation and affine transforms, as most distortions presents in natural-world images can be sufficiently accurately approximated (at least locally) by such mappings. Thus, the objective of keypoint-based image matching can be specified as the identification of groups of matched keypoints that satisfy the geometric constraints of the underlying transformation.

For such problems, the RANSAC paradigm, e.g. [4], is the standard solution from which techniques used in other works have been directly or indirectly developed. This paradigm assumes that local feature matches might be incorrect due to measurement errors (incorrect detection and localization by a detector) or classification errors (similar descriptors representing unrelated visual contents). If correct matches constitute at least a significant percentage of all matches, the outliers can be (more or less effectively) rejected.

As mentioned above, affine transforms are considered the distortion model for images matching (even though RANSAC can be applied to more complex transformations) since perspective distortions are locally approximated well enough by affine functions. However, non-linear models on top of affine transforms are also sometimes considered (e.g. [28]). Such an approach works quite well for *near-duplicate image* matching/retrieval (where images contain the same scene though possibly captured under different viewing setting or photometric conditions). This is the most popular problem, widely used to illustrate or benchmark performances of various keypoint detectors and descriptors (e.g. [2, 3, 14, 16, 17, 28, 32], etc.). A similar (though a more general) problem is *near-duplicate sub-image* matching/retrieval where we attempt to identify (database) images containing the image of interest (a query image) as a fragment (e.g. [9, 12]).

Neither *near-duplicate image* nor *near-duplicate sub-image* matching techniques address the issue of detecting unspecified similar fragments in images of unpredictable, possibly unrelated contents. It can be noticed that the RANSAC paradigm is not applicable to such a problem, since correct keypoint matches can constitute an insignificant (or even statistically negligible) percentage of all matches. Recently, a partial solution has been proposed in [31], where only similarities modeled by rotations and scalings are considered. Another study, [8], proposed a solution that seems feasible only for the pre-selected scenarios since visual *bags-of-words* are precomputed from representative images (video-frames) for each scenario. Both method use keypoints as the underlying local features.

In this chapter we present the most general currently existing (to our best knowledge) solution. Formally, we can specify it as follows:

**Given two random images (i.e. with no prior knowledge about their contents) I and J, identify pairs of near-duplicate image fragments that are related by affine transformations. The term ”near-duplicate fragments” refers to fragments depicting (almost) identical objects. However, the visual appearances of those fragments may differ, because of different scene and camera settings, photometric conditions, digitization parameters and possibly because of certain deformation of the objects.**

The method is also general in a sense that we do not restrict our method to particular types of keypoint detectors or descriptors. However, the recommended typical choices are briefly overviewed in Section 2. Section 3 is the central part of the chapter. There, we discuss strategies of keypoint matching, details of affine transformation building from two types of geometric structures, two decomposition techniques for affine transformations and, finally, the methodology of affine histogram building. The affine-related near-duplicate fragments are eventually identified as prominent spikes of such histograms. Performance evaluation, the parameter-tuning issues, and prospective applications of the method are overview in Section 4.

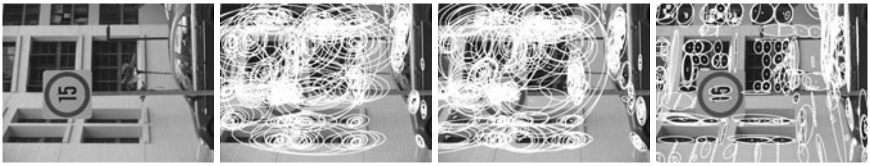
This chapter summarizes and updates results of our recent research papers (some of them not published yet). For example, one of the papers ([21]) discusses the image fragment matching using triangles of keypoint, while another ([22]) focuses on elliptical approximations of key regions surrounding the keypoints. The chapter, together with the program executables and an exemplary visual database available at *http://www.ii.pwr.wroc.pl/∼visible*, presents a *ready-to-use* technology for image fragment matching in various machine vision applications.

## 2 Keypoints: Detectors and Descriptors

Calculation of local image characteristics is the first and elementary step in many image analysis approaches. It provides the necessary data for all following steps. First, we will address the problem of keypoint detection. Later, we discuss the problem of feature calculation describing these keypoints.

### *2.1 Keypoint Detectors*

The method proposed in this chapter can prospectively work with any types of keypoint detectors. The only assumption is that the corresponding keypoint descriptors are available (preferably in a form of *n*-dimensional vectors) so that similarities between keypoint can be established. However, we focus on three popular detectors known to be reliable performers under a wide range of photometric and geometric distortions (including affine transformations). Moreover, all these detectors return keypoints in a form of elliptical key regions (which are important in the second variant of the presented work) with the keypoint coordinates at the centre of the corresponding ellipse. The detectors differ, however, in the underlying mathematics. To illustrate the concept of elliptical keypoints, in Fig. 1 we show generated keypoints for an exemplary image.



(a) Original image (b) Harris-Affine (c) Hessian-Affine (d) MSER

**Fig. 1** Examples of detected elliptical key regions, near 500 most prominent regions are chosen.

#### 2.1.1 Harris-Affine Keypoint Detector

Harris-Affine detector has been derived from the *corner detector* proposed in [7]. This corner detector is based on the autocorrelation matrix of the image intensity function $I(x,y)$ averaged over a small area (usually the averaging is modeled by an isotrophic Gaussian filter $g(\sigma_I)$ determining the *integration area*)

$$A(x,y,\sigma_I) = g(\sigma_I) \otimes \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}, \quad where \quad I_x = \frac{\partial I(x,y)}{\partial x} \quad and \quad I_y = \frac{\partial I(x,y)}{\partial y}. \tag{1}$$

The proposed measure of "cornerness" (related to the eigenvalues of matrix $A$) is

$$R(x,y) = det(A) - \alpha trace^2(A) \tag{2}$$

where typical values of $\alpha$ are in 0.04-0.08 range.

A keypoint (corner point) is detected at $(x,y)$ locations where $R(x,y)$ exceeds a predefined threshold and reaches a local maximum.

If the image is rescaled, the detection should be performed over the correspondingly changed area, i.e. the detector's Gaussain kernel should be modified.

Image rescaling is usually modeled by the convolution of the intensity function $I(x,y)$ with another isotrophic Gaussian filter $g(\sigma_D)$ so that Eq. 1 converts to:

$$A(x,y,\sigma_I,\sigma_D) = \sigma_D^2 g(\sigma_I) \otimes \begin{bmatrix} ID_x^2 & ID_xID_y \\ ID_xI_y & ID_y^2 \end{bmatrix}, \quad where \quad ID_{x/y} = \frac{\partial I(x,y,\sigma_D)}{\partial x/y}. \tag{3}$$

Both Gaussian filters should change proportionally (typically $\sigma_D = 0.7\sigma_I$) so that Eq. 2 has three variables, if the image is analyzed in multiple scales.

$$R(x,y,\sigma) = det(A) - \alpha trace^2(A). \tag{4}$$

Harris-Laplace detector identifies the local maxima of Eq. 4 in three dimensions and, thus, returns not only the coordinates of each keypoint but also its optimum scale (the size of the integration area is visualized by a circle of the radius proportional to the scale).

Harris-Affine detector is a generalization of Harris-Laplace. The difference is that we assume affine distortions of the underlying images (instead of a simple scaling) so that both Gaussian kernels should change differently to preserve the local visual characteristics of images.

Formally, we create another matrix $A$

$$A(x,y,\Sigma_I,\Sigma_D) = det(\Sigma_D)g(\Sigma_I) \otimes (\nabla I(x,y,\Sigma_D))\,(\nabla I(x,y,\Sigma_D))^T \tag{5}$$

where $\Sigma_I$ and $\Sigma_D$ are the covariance matrices for the Gaussian kernels representing affine deformations of the image ($\Sigma_D$) and the integration area ($\Sigma_I$). Again, the image deformation and the integration area deformation should be proportional so that the problem of keypoint detection consists in local maxima detection in a 5-dimensional space (two coordinates and three parameters of the covariance matrix).

In [16], the problem is decomposed and simplified. First, the coordinates of keypoints are found using Harris-Laplace detector. Subsequently, the optimum affine deformation (i.e. the optimum covariance matrix and the shape of corresponding ellipse) is iteratively found for each keypoint.

### 2.1.2 Hessian-Affine Keypoint Detector

Hessian-Affine detector is very similar to Harris-Affine. The only major difference is the measure of ”cornerness”. Instead of $A$ matrix of Eq. 1, the Hessian matrix of second derivatives of the intensity function $I(x,y)$ is used:

$$H(x,y) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}, \quad where \quad I_{xy} = \frac{\partial^2 I(x,y)}{\partial x \partial y}, \quad etc. \tag{6}$$

Keypoints are detected at coordinates where both the determinant and trace of $H$ matrix reach the local extremum.

$$det(H) = I_{xx}I_{yy} - I_{xy}^2, \quad trace(H) = I_{xx} + I_{yy} \tag{7}$$

If the image is rescaled (i.e. its intensity is convolved with an isotrophic Gaussian filter $g(\sigma)$, the extrema are found over three dimensions, i.e. $x, y, \sigma$, and instead of Eq. 7 we use

$$det(H) = \sigma^2 \left(I_{xx}I_{yy} - I_{xy}^2\right), \quad trace(H) = \sigma\left(I_{xx} + I_{yy}\right) \tag{8}$$

Other details of the detector are identical to Harris-Affine. In particular, it also returns ellipses representing the scales (and affine distortions) of keypoints.

### 2.1.3 Maximally Stable Extremal Regions

Unlike the previous two keypoint detectors, the MSER (*maximally stable extremal regions*) detector, proposed in [15], is not based on differential properties of image intensities. Instead, it is based on a sequence of binary images (starting from *all white* and terminating in *all black*) obtained by thresholding the image of interest with a gradually increasing threshold. The set of all connected regions (some of them are white and some are black) in such a sequence of images forms the set of *extremal regions*.

Extremal regions of the same color are nested. For example, a black extremal region $Q_t$ obtained for $t$ threshold is placed within another black extremal region $Q_{t+\Delta t}$ obtained with $t + \Delta t$ threshold. At the same time, by reducing the threshold to $t - \Delta t$, we obtain an extremal region (or several regions) $Q_{t-\Delta t}$ that is within $Q_t$. For white extremal region the situation is reversed (a white region is within another white region created by a lower threshold).

*Maximally stable extremal regions* (MSER) are those extremal regions that are least sensitive (in terms of their area) to the threshold changes. Formally, MSERs corresponds to the local minima of the following expression:

$$\frac{area(Q_{t+\Delta t} \backslash Q_{t-\Delta t})}{area(Q_t)}. \tag{9}$$

MSER detector obviously extracts regions of diversified sizes and shapes. However, they are usually converted into ”proper” key regions. The MSER’s centre of mass becomes the keypoint, while the moment-based elliptical approximation (e.g. [24]) of the MSER's shape forms the key region.

### *2.2 Keypoint Descriptors*

As already mentioned, we are not particularly concerned about the type of keypoint descriptors used in the proposed algorithms (as long as they accurately match visually similar key regions). However, a brief survey of the most popular descriptors (based on different principles and having different dimensionalities) is included for clarity and better understanding of the background.

The most straightforward (but also the least discriminative) keypoint descriptors are color or intensity characteristics of their key regions.

More popular are differential keypoint descriptors that use image derivatives in several ingenious ways. Typical examples of such descriptors include:

**SIFT** ([14]: This is a 128-dimensional descriptor representing (normalized) gradient magnitudes in 8 major directions over a $4 \times 4$ grid superimposed on a circle (or an ellipse normalized to a circle). Several modification and improvements of SIFT exist, e.g. PCA-SIFT ([11]), CSIFT ([1]), ASIFT ([20]), etc.

**GLOH** ([17]): It is similar to SIFT but differs mainly in the shape of grid. Its $17 \times 16$ grid in log-polar coordinates provides 272-dimensional vectors of local gradients.

***Gradient moments*** ([17]): This descriptor consists of moments (up to second order) of the image derivatives (up to second degree) computed over the key region. Thus, the descriptor's dimensionality is 25.

Other keypoint descriptors are more directly based on the image intensities. In this category the following descriptors, among others, can be listed:

**SURF** ([2]): The image intensity is decomposed (within $4 \times 4$ grid superimposed on a circle) into Haar wavelets. Each grid square is sub-divided into a $2 \times 2$ pattern to calculate its Haar wavelets so 64 descriptor coefficients exist altogether. A similar descriptor (on $3 \times 3$ grids) using another set of Haar-like functions is proposed in [28]. Its dimensionality is 36.

***Color moment invariants*** ([18]): Generalized color moments (in three color channels) combine shape and color information. 18 moment invariants for affine geometric and photometric transformations (using moments up to first order and second degree) are defined. Thus, this descriptor (computed on the normalized key regions) is 18-dimensional.

## 3 Matching Image Fragments

In this section we presents two approaches to the detection of affine-related, near-duplicate fragments in two images $\mathbf{I}$ and $\mathbf{J}$. The approaches differ in one aspect only. Affine transformations are reconstructed either by using three pairs of matched keypoints (i.e. triangles) or by using two pairs of matched ellipses (key regions). All other steps of the proposed method are basically identical.

The main methodological challenge is to avoid prohibitively high computational complexities (which can often happen when the large amounts of data are processed). Thus, the ellipse-based approach with $O(n^2)$ complexity (where $n$ estimates the number of keypoints in an image) is superior to the triangle-based method ($O(n^3)$ complexity). However, the accuracy of the less complex ellipse-based approach is slightly compromised.

Once near-duplicate image fragments are matched, we can compare image contents at a higher level. Instead of correspondences between hundreds/thousands of keypoints, we can address the issue from the perspective of similar regions usually representing fragments of real objects.

In general, the proposed method consists in the following steps:

1: Find keypoints in input images and their descriptors (off-line step),
2: Find credible keypoint matches,
3: Build geometric structure for reconstruction of affine transformations,
4: Reconstruct affine transformations,
5: Decompose affine transformations into elementary transformations,
6: Build 6D parameter histograms of decomposed transformations,
7: Find high density areas (peaks) in 6D histogram,
8: Group found high density areas into objects.

All these steps (except the first one already explained in Section 2.1) are discussed in the following sections.

### *3.1 Keypoint Matching*

Even though performances of keypoint detectors are not perfect (in terms of the actual robustness against viewpoint and illumination changes) they provide large numbers of fairly stable features that can be matched with similarly large numbers of features from other images. This strategy is applied in almost all current image matching algorithms.

Generally, three keypoints matching schemes exist, [32]: one-to-one (O2O), one-to-many (O2M) and many-to-many (M2M). In our work, only two of these schemes are used: O2O and M2M. One-to-many is by definition asymmetric and, thus, it does not fit into the general idea of our method.

O2O is much faster and sufficiently reliable unless images contain multiple copies of the same object (although simple cases of such scenarios can be handled as well). Alternatively, M2M usually generates much more matching pairs and is, therefore, slower. However, keypoints are connected *many-to-many* and detection of multiple copies of objects is easier. Otherwise, both methods perform similarly well in the stated matching problem.

In our work we have implemented the following variants of the schemes:

- As O2O we use the *mutual nearest neighbor* method, i.e. keypoints are paired *if and only if* they are mutual nearest neighbors.
- As M2M we use the *nearest neighbor* method. All nearest neighbors are used; mutual nearest neighbors are represented only once (no duplicates).

However, matched keypoint pairs are often duplicated. Some combinations of detectors and descriptors (e.g. Harris-Affine and SIFT) tend to generate multiple keypoints very close to each other or exactly at the same locations. Since our solution is, in fact, density based such multiple keypoints may generate false density spikes and thus produce artifacts. As a remedy, we detect such clusters of keypoint pairs present in both images by measuring distances between them. The experimentally selected threshold of keypoint distances is equal to 0.5% of the image resolution. Once such a cluster is detected, each keypoint is weighted $1/c$, where $c$ is the number of keypoints in the cluster.

In the next step of the method, we create elementary geometric structures on top of the keypoint pairs. These geometric structures have to be weighted accordingly. Their weights are calculated by multiplying weights of all keypoint pairs contributing to a structure.

## 3.2 Affine Transformation Reconstruction

An affine transformation is a linear mapping followed by a translation. In case of images, we are interested in affine transformations in two-dimensional spaces, i.e. transformations are defined by six parameters. Four of the parameters form the linear part, while the remaining two specify the translation vector.

If $(x,y)$ are the source image $\mathbf{I}$ coordinates, and $(u,v)$ represent the destination image $\mathbf{J}$ coordinates, the affine transformation is represented by a homogeneous matrix $\mathbf{A}$:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad where \quad \mathbf{A} = \begin{bmatrix} A & B & C \\ D & E & F \\ 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

where: $A,B,D,E$ is the linear part and $C,F$ is the translation vector.

Affine transformations have several useful properties (exploited in the later part of this chapter):

1. co-linear points remain co-linear,
2. distance ratios of co-linear points are not changed,
3. triangles are mapped into triangles (possible in a degenerate form, i.e. a line or a point),
4. second degree curves are mapped into second degree curves (possible in a degenerate form),
5. parallel lines remain parallel.

Reconstruction of affine transformations from matched keypoints is the first step of the global analysis of the image geometry. However, such a reconstruction can be done in several different ways. In this chapter we use two elementary geometric structures to reconstruct affine transformation, i.e. matched keypoints and matched elliptical key regions. Our discussion starts from the former approach, as it is conceptually simpler.

### 3.2.1 Triples of Keypoints (Triangles)

A 2D affine transformation has six parameters so that we need a non-singular system of six linear equations to reconstruct the transformation. Using the third principle (triangles are mapped into triangles) we can form such a system of equations from three matched pairs of non-colinear keypoints. Consider three points $(x_1,y_1)$, $(x_2,y_2)$ and $(x_3,y_3)$ in the source image **I**, and their counterparts $(u_1,v_1)$, $(u_2,v_2)$ and $(u_3,v_3)$ in the destination image **J**. To reconstruct the affine transformation from these triangles, we solve the following system of equations:

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix}^{-1} . \tag{11}$$

The triangle-based approach has two major advantages and one disadvantage. First, it uses a non-singular system of linear equations which can be easily solved. Triangle points (i.e. actually keypoints) are usually very precisely localized so that the transformation is reconstructed with a high accuracy. However, the computational complexity is a serious disadvantage. With $n$ pairs of matched keypoint, the total number of triangles can possibly reach $n^3$, i.e. we have to deal with the memory and computational complexity of $O(n^3)$. Considering that the number of matched pairs can be higher than 1000, this is a serious inconvenience.

### 3.2.2 Ellipses and Points

The major reason to use the ellipse-based approach is $O(n^3)$ complexity of the triangle-based technique. Using the ellipse-based approach, we can reduce the computational and memory complexity to $O(n^2)$. Apart from this significant advantage of the ellipse-based approach, a number of problems also exist, as we follow. Recreation of a single affine transformation using the ellipse-based approach is conceptually and numerically more complex.

In this approach we assume one pair of matched elliptical key regions (with matched keypoint in their centers) and another pair of matched keypoints. It should be highlighted that the elliptical key regions (regardless of their type) are not actual objects of the matched images. They are just the local estimates of the orientation, anisotropy and/or shapes of the image intensities (see Section 2.1 for more details on keypoints). This observation plays an important role in our subsequent analysis.

Assume an elliptical key region in image **I** with the center $(x_i,y_i)$, two size-related parameters $P_i,R_i$ and a rotation parameter $Q_i$. Then, the equation of the ellipse in image **I** is:

$$P_i\,(x-x_i)^2 + Q_i\,(x-x_i)\,(y-y_i) + R_i\,(y-y_i)^2 = 1. \tag{12}$$

The corresponding ellipse (key regions) in image **J** would have a center point $(u_i, v_i)$ and three parameters $A_i, B_i$ and $C_i$. The equation of this ellipse is:

$$A_i (u - u_i)^2 + B_i (u - u_i)(v - v_i) + C_i (v - v_i)^2 = 1. \tag{13}$$

We cannot reconstruct the affine transformation from such a pair of ellipses because a system of six equations with six unknowns cannot be formed from five parameters of matched ellipses. On the other hand, two pairs of matched ellipses would provide ten parameters, and the resulting system of equations would be overdetermined and strongly contradictory.

Our next choice is a matched pair of ellipses and, additionally, a matched pair of keypoints. This means seven equations altogether, i.e. system of equations is still overdetermined. However, with only equation redundant, we can attempt some kind of regularization that would provide a reliable reconstruction of affine transformations. The following problems should be addressed:

- What ellipse-related data should be eliminated to reduce the number of degrees of freedom from seven to six?
- How to analytically calculate the affine transformation from the remaining data?
- How to select a unique affine transformation if multiple solutions are obtained?

Regularization of ellipses

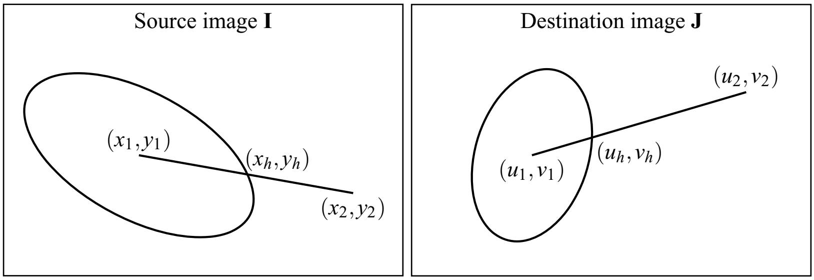
The second listed affine property (see Section. 3.2) says that distance ratios of colinear points are not changed. In case of a pair of matched ellipses and another pair of matched keypoints, this property should be satisfied for the points indicated in Fig. 2. In fact, it is practically always violated.

We, therefore, propose to either resize one of the ellipses (or to reposition one of the keypoints) so that the system of seven equations (five from the ellipse parameters and two from the keypoint coordinates) with six unknowns (affine transformation parameters) would be regularized into a system with seven unknowns.

The locations of keypoints are more credible data than the sizes of ellipses since the latter ones are just the visual estimates of the local direction and anisotropy of images, while their sizes are often determined by the keypoint detector (e.g. *Harris–Affine*) in a finite number of iterations using discretized scales. Even if the ellipses more directly represent the actual shapes, their sizes depend on the parameter settings (e.g. the step size of image thresholding in *MSER* detector) and, thus, are only roughly approximated. Keypoints, on the other hand, are usually quite accurately localized by the detectors.

Therefore, in order to satisfy the affine properties in the ellipse+keypoint structures, we propose to resize one of the ellipses, and this technique is referred to as *ellipse regularization*. A similar concept of scale omission has been presented for *homography transformation* reconstruction using a pair of ellipses [10], however the details are different.

First, ellipse and line intersections are calculated, as shown in Fig. 2, and the intersection point distance ratios $\delta_I$ and $\delta_J$ are calculated for both matched ellipses.



$$\frac{|(x_1,y_1)(x_2,y_2)|}{|(x_h,y_h)(x_2,y_2)|} = \frac{|(u_1,v_1)(u_2,v_2)|}{|(u_h,v_h)(u_2,v_2)|}$$

**Fig. 2** Distance ratios between corresponding points co-linear remain unchanged under affine transformations. In case of an arbitrary pair of matched ellipses and an arbitrary pair of matched keypoint this condition is usually violated.

Then, the ellipse in the destination image **J** is resized according to the $\Delta$ coefficient calculated as

$$\Delta = \frac{\delta_J}{\delta_I}, \quad \textit{where} \quad \delta_I = \frac{|(x_1,y_1)(x_2,y_2)|}{|(x_h,y_h)(x_2,y_2)|}, \quad \delta_J = \frac{|(u_1,v_1)(u_2,v_2)|}{|(u_h,v_h)(u_2,v_2)|}. \tag{14}$$

Once the ellipse is resized, the affine constraints are satisfied for the ellipse+keypoint configurations. From the mathematical perspective, the seventh parameter $\Delta$ (that constraints the existing variables) effectively reduces the number of degrees of freedom from seven to six.

### Regularized ellipses and keypoints – the tangent line method

Once the ellipse is regularized, the affine transformation can be analytically reconstructed form the ellipse+keypoint configurations. We exploit the fifth property of affine transformations, i.e. the preservation of line parallelism.

Actually, we convert the problem into the previously discussed (in Section 3.2.1) simple technique of affine transformation reconstruction from three pairs of points.

The first and the second pair of points are the centers of matched ellipses and the pair of matched keypoints ($(x_1,y_1),(u_1,v_1)$ and $(x_2,y_2),(u_2,v_2)$ in Fig. 3, correspondingly). The third pair of points would be determined using the idea shown in the same figure.

Assume the ellipse-line intersection point $(x_h,y_h)$ located between $(x_1,y_1)$ and $(x_2,y_2)$, and its corresponding point $(u_h,v_h)$ point between $(u_1,v_1)$ and $(u_2,v_2)$ in the second ellipse.

Lines tangent to the ellipses can be found for $(x_h,y_h)$ and $(u_h,v_h)$, respectively. These tangent lines are also related the same affine transformation. Subsequently, the tangent lines are translated to the centres of the corresponding ellipses. The equations of the translated tangent lines are:

$$\begin{aligned} y &= a_I x + b_I, & a_I &= -\tfrac{2P(x_h-x_1)+Q(y_h-y_1)}{2R(y_h-y_1)+Q(x_h-x_1)}, & b_I &= y_1 - a_I x_1, \\ v &= a_J u + b_J, & a_J &= -\tfrac{2A(u_h-u_1)+B(v_h-v_1)}{2C(v_h-v_1)+B(u_h-u_1)}, & b_J &= v_1 - a_J u_1, \end{aligned} \qquad (15)$$

where $a_I$, $b_I$, $a_J$, $b_J$ are the respective tangent line parameters in images **I** and **J**, while $P,Q,R,A,B,C$ represent the ellipses specified in Eqs 12 and 13.

The intersection points of the translated tangent lines and the ellipses (there are two such points for each ellipse) are the third points in triangles needed to reconstruct the underlying affine transformation. Therefore, we can use e.g. $(x_3,y_3)$ point to form (together with $(x_1,y_1)$ and $(x_2,y_2)$) a triangle in image **I**, while its counterpart $(u_3,v_3)$ would form the corresponding triangle (together with $(u_1,v_1)$ and $(u_2,v_2)$) in image **J**. The affine transformation would be easily retrieved from such a pair of triangles.

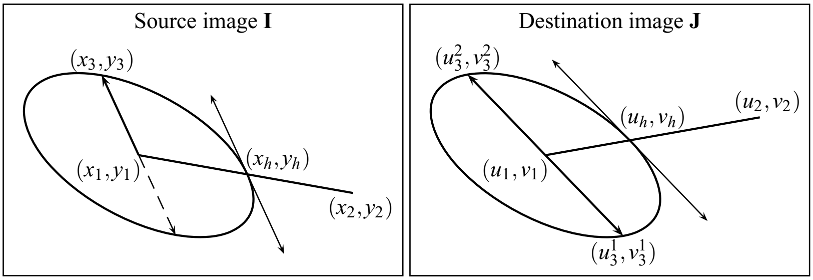


**Fig. 3** The third pair of points to reconstruct an affine transformation is found from intersections of regularized ellipses and shifted tangent lines.

However, as and shown in Fig. 3, two options exist for $(u_3,v_3)$ so that two alternative affine transformations can be found using ellipse+keypoint configurations.

This ambiguity can be solved in two ways, depending on the generally assumed definition of "near-duplicate planar objects". If we exclude mirror reflections, $(u_3^2,v_3^2)$ point is the only choice for $(u_3,v_3)$ because the affine transformation should not change the direction of vectors' cross-products. However, if we accept mirror reflections as legitimate cases of "near-identicities", $(x_3,y_3)$ can be mapped into either $(u_3^1,v_3^1)$ or $(u_3^2,v_3^2)$, and the ambiguity would be solved differently.

Although we use only $(x_2,y_2)$ and $(u_2,v_2)$ as the matched keypoints, their corresponding elliptical key regions also exist. It can be, therefore, estimated under which of the alternative affine transformations these key ellipses are more similar. The affine transformation providing a higher similarity for this second pair of ellipses is chosen as the final one.

### *3.3 Affine Transformation Decomposition*

Affine transformations describe distortions of planar surfaces. However, the information about the geometry of such distortions cannot be easily extracted from the algebraic representation of transformations (see Eq.10). The parameters of such a representation are not too meaningful and readable. Thus, in order to retrieve the geometry of distortions involved in affine transformations, we decompose them into sequences of basic operations. There are several ways of such a decomposition, some of them more meaningful, some of them much less. In our work, we focus on two decomposition techniques. For our needs, we have found these two techniques meaningful and accurate enough . In particular, they provide a satisfactory level of parameter stability in the generated decompositions of affine transformations.

### 3.3.1 Singular Value Decomposition

The first approach to affine transformation decomposition is based on *singular value decomposition* (SVD) [21, 26]. In this decomposition we consider only 2D projections (into the camera plane) of 3D planar surfaces. In fact, the decomposition does not have any three dimensional meaning. It is meaningful only in terms of elementary transformations of 2D image.

The elementary transformations used in this decomposition are: two 2D rotations and two scale changes. The translation components of the transformations remain the same as in the algebraic form of Eq.10 (see below).

Let us split an affine transformation it into the linear part **K** and the translation **P**:

$$\mathbf{A} = \begin{bmatrix} A & B & C \\ D & E & F \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{0}^T & 1 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} A & B \\ D & E \end{bmatrix}, \mathbf{P} = \begin{bmatrix} p_X \\ p_Y \end{bmatrix} = \begin{bmatrix} C \\ F \end{bmatrix}. \qquad (16)$$

The linear transformation sub-matrix **K** is decomposed into elementary operations as follows:

$$\mathbf{K} = \mathbf{Rot}(\gamma) \cdot \mathbf{Rot}(-\theta) \cdot \mathbf{N} \cdot \mathbf{S} \cdot \mathbf{Rot}(\theta). \qquad (17)$$

In such a decomposition, **Rot** represents a 2D rotation matrix, **S** is a positive defined diagonal scaling matrix and **N** is either an identity matrix or a mirror reflection matrix, i.e.

$$\mathbf{Rot}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}, \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & \pm 1 \end{bmatrix}. \qquad (18)$$

The presented decomposition can be modeled using a singular value decomposition. SVD decomposes **K** matrix into a multiplication of three matrices **U**, **D** and **V**. Matrices **U** and **V** are orthonormal and **D** is a positive defined diagonal matrix. Matrix **V** is composed of eigenvectors of $\mathbf{K^TK}$, while eigenvalues of $\mathbf{K^TK}$ form (in the decreasing order) the diagonal of matrix **D** in . Matrix **U** is obtained by a matrix multiplication from the other two. Thus, it is possible to express the linear transformation matrix **K** by the matrices of SVD:

$$\mathbf{K} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V^T} = \mathbf{U} \cdot \mathbf{V^T} \cdot \mathbf{V} \cdot \mathbf{D} \cdot \mathbf{V^T}. \qquad (19)$$

After the singular value decomposition is performed, the parameters of elementary operations can be obtained as follows:

$$\mathbf{Rot}(\gamma) = \mathbf{U} \cdot \mathbf{V}^{\mathbf{T}}, \quad \mathbf{Rot}(\theta) = \mathbf{V}^{\mathbf{T}}, \quad \mathbf{N} \cdot \mathbf{S} = \mathbf{D}. \tag{20}$$

It should be noted that the scaling factors have to be assigned according to the order of eigenvalues ($s_x = \lambda_1$ and $s_y = \lambda_2$) which are always positive. However, in case of mirror reflections a mirror reflection matrix **N** is used (see Eq. 18). Additionally, we improve the processing of scaling factors in the later stages by using *linearized scales* $z_x$ and $z_y$:

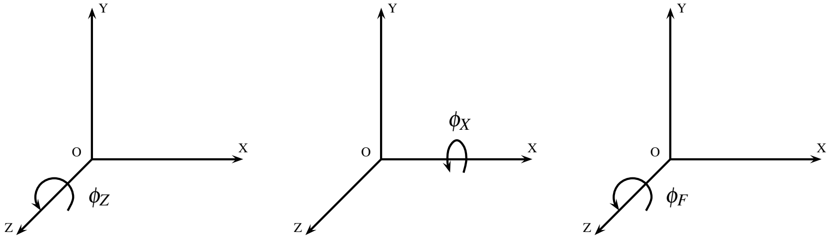
$$z = \begin{cases} s-1 & \text{if } s \geq 1 \\ 1 - 1/s & \text{otherwise} \end{cases} . \tag{21}$$

The range of angle $\theta$ (that defines shearing of affine transformations) is $(0, \pi)$ while the range of $\gamma$ angle (representing the actual 2D rotation) is $(0, 2\pi)$. Two sets of orthonormal eigenvectors (of the opposite directions) exist from which we take only one. In case of identical eigenvalues, a singularity exists and the angles $\gamma$ and $\theta$ are indistinguishable.

### 3.3.2 3D Geometric Decomposition

Alternatively, we can decompose affine transformations assuming that shape distortions of the same planar object in two images result from a 3D motion of the object relatively to the camera system of coordinates, which is followed by a perspective projection (approximated by an orthogonal projection and a scale change).

Any 3D motion is defined by a 3D rotation (i.e. three planar rotations) and a 3D translation vector. Decompositions of 3D rotations into planar rotations are usually based on Euler angles (e.g. [2, 5, 6]) and we propose to use one of such notations. As shown in Fig. 4, we represent a 3D rotation as a rotation about OZ axis ($\phi_Z$ angle), followed by a rotation about OX axis ($\phi_X$ angle) and another rotation about OZ axis ($\phi_F$ angle). All rotations are about axes of a motionless frame OXYZ.



(a) 1st rotation – Z axis (b) 2nd rotation – X axis (c) 3rd rotation – Z axis

**Fig. 4** Decomposition of 3D rotations into a sequence of planar rotations.

When a 3D translation by a vector $[p_X, p_Y, p_Z]$ is introduced, the homogeneous transformation matrix **HT** of a 3D motion is represented by Eq. 22. The image coordinates of the transformed object are found by applying the perspective projection which is approximated by the following homogeneous matrix **PR**:

$$\mathbf{HT} = \begin{bmatrix} c_F c_Z - s_F c_X s_Z & -c_F s_Z - s_F c_X c_Z & s_F s_X & p_X \\ s_F c_Z + c_F c_X s_Z & -s_F s_Z + c_F c_X c_Z & -c_F s_X & p_Y \\ -s_X s_Z & s_X c_Z & c_X & p_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{PR} = \begin{bmatrix} K & 0 & 0 & 0 \\ 0 & K & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (22)$$

Note that $c_F$ and $s_F$ indicate correspondingly $\cos\phi_F$ and $\sin\phi_F$, etc.

The superposition of Eq. 22 matrices is a 2D affine transformation matrix in a form containing parameters of the underlying 3D motion (the value of $p_Z$ translation does not appear directly but it is "hidden" in the scaling factor $K$):

$$\mathbf{PR} \cdot \mathbf{HT} = \begin{bmatrix} K(c_F c_Z - s_F c_X s_Z) & -K(c_F s_Z + s_F c_X c_Z) & 0 & Kp_X \\ K(s_F c_Z + c_F c_X s_Z) & K(-s_F s_Z + c_F c_X c_Z) & 0 & Kp_Y \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (23)$$

The elements of the Eq. 23 matrix are compared to the algebraic representation of affine transformations (Eq. 10) to form the following system of equations:

$$\begin{cases} K(c_F c_Z - s_F c_X s_Z) & = A \\ -K(c_F s_Z + s_F c_X c_Z) & = B \\ K(s_F c_Z + c_F c_X s_Z) & = D \\ K(-s_F s_Z + c_F c_X c_Z) & = E \\ Kp_X & = C \\ Kp_Y & = F \end{cases} \quad (24)$$

from which the motion parameters would be computed.

Eq. 24 is a non-linear system of equations that has multiple solutions for $\phi_Z$ and for $\phi_F$:

$$\phi_Z = \frac{1}{2} atan2\left(AB + DE, \frac{B^2 + E^2 - A^2 - D^2}{2}\right) + \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}, \quad (25)$$

$$\phi_F = \frac{1}{2} atan2\left(AD + BE, \frac{A^2 + B^2 - D^2 - E^2}{2}\right) + \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}. \quad (26)$$

However, only two solutions exist altogether, from which the following values of $\phi_Z$ and $K$ can be found:

$$K = c_Z\left(Ac_F + Ds_X\right) - s_Z\left(Bc_F + Es_X\right), \quad (27)$$

$$\phi_X = \arccos\left(\frac{c_F(Ds_Z + Ec_Z) - s_F(As_Z + Bc_Z)}{K}\right). \quad (28)$$

Finally, the translational parameters $p_X$ and $p_Y$ can be found as

$$p_X = \frac{C}{K}, \quad p_Y = \frac{F}{K}. \tag{29}$$

It should be noted that the scaling factor $K$ is also linearized using Eq. 21 (i.e. similarly to SVD decomposition).

The solution becomes singular when both arguments of $atan2$ function in Eqs 25 and 26 are zeros. This happens for similarity transformations (no rotation about OX axis – see Fig. 4) when the rotations by $\phi_Z$ and $\phi_F$ are indistinguishable.

## *3.4 Histograms of Affine Transformations*

Parameters of decomposed transformations more meaningfully represent deformations of objects than algebraic parameters of affine transformations. Two proposed decompositions provide values of rotations $\gamma \in \Gamma, \theta \in \Theta$ (or $\phi_Z \in \Phi_Z, \phi_X \in \Phi_X, \phi_F \in \Phi_F$), translations $p_X \in P_X, p_Y \in P_Y$ and scalings $z_X \in S_X, z_Y \in S_Y$ (or $k \in K$) between image fragments.

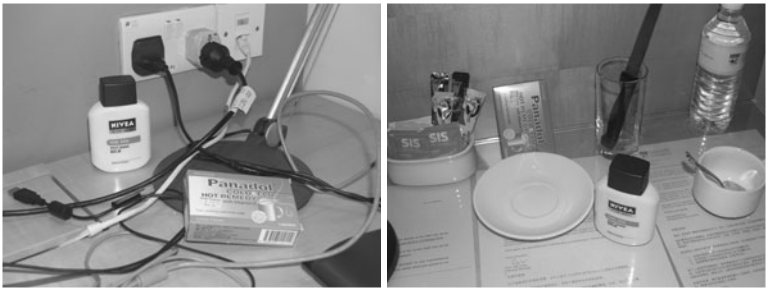
Usually there are tens or even hundreds of thousands of decomposed transformations representing relations between two images. This large volume of data should be processed in a meaningful and computationally efficient way. To do this, a probabilistic approach is proposed.

Let each affine transformation be a probabilistic event with six continuous variables. Such an event belongs to a six dimensional probabilistic space representing all affine transformations. Depending on the selected decomposition, there are two such density functions: $P_{svd}(\Gamma, \Theta, P_X, P_Y, S_X, S_Y)$ and $P_{3d}(\Phi_Z, \Phi_X, \Phi_F, P_X, P_Y, K)$.

Because the content of the processed images is unknown, the shape of distributions is unpredictable either. We can only predict that groups of similar affine transformations would form peaks in the density function. Since similar planar fragments of images are mapped by the same affine transformations, the density functions should have exactly as many density peaks as the number of similar fragments in a pair of images, but the actual numbers of such fragments should be determined from the density function itself.

As a simple example, we consider two images given in Fig. 5. Both images share two identical objects (image fragments), i.e. a *white bottle* and a *flu remedy pack*. Backgrounds of the images are completely different, containing many unrelated objects. As a further complication, the *white bottle* object is not actually planar. However, even for a such complex scene, the density peaks in the probability density function are very distinctive.

The probability density histogram created for these two images corresponds to the SVD decomposition of affine transformations. This is an arbitrary choice, and the results for the 3D decomposition method are qualitatively identical. The histogram is shown as three two-dimensional density functions because a meaningful visualization of a 6D density function is very difficult.



(a) source image (b) target image

**Fig. 5** Two scenes containing two identical objects – a white bottle and green flu remedy pack. We demonstrate histograms using these two scenes.

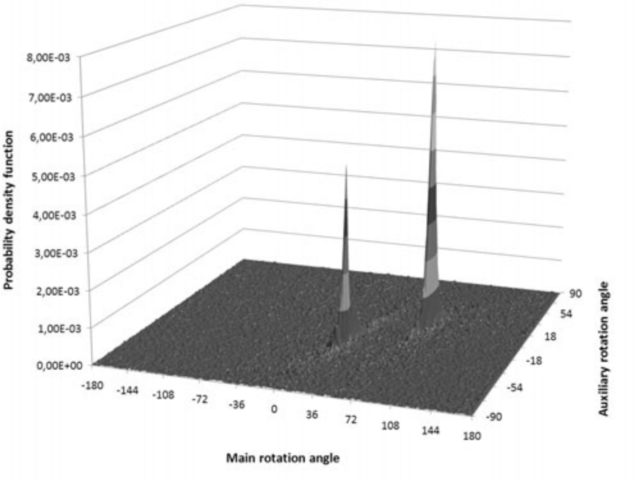


**Fig. 6** 2D rotation probability density function generated by SVD decomposition. Two peaks represent two objects located on the images. Each object is rotated differently.

The histogram in Fig. 6 presents the probability density function of two rotation angles of SVD method, i.e. $P_{svd}(\Gamma,\Theta)$ distribution. Two distinctive peaks are visible. Each of these peaks represents a single object shared by both scenes. From the histogram, we can estimate how these objects how been rotated. Looking at the *main rotation angle* $\gamma$, we judge that one of objects (represented by the left peak) has not rotated. When we compare input images (see Fig. 5), the *white bottle* object is, in fact, not rotated. The *flu remedy pack* object has been rotated by approx. 72°. Regarding the second angle, i.e. the *auxiliary rotation angle*$\theta$, both objects are rotated by about 18°. These rotations exist due to a small perspective distortion caused by different object localizations and different camera settings in both images.

We can also notice different heights of the density peaks (the peak representing the *white bottle* object is much lower). This can be attributed to different numbers of keypoints in both objects. The surface of the *flu remedy pack* object is more textured, which results in more keypoints. Additionally, the object is naturally larger than the *white bottle* objects. Altogether, larger numbers of matched keypoint pairs generate more triangles so that a larger number of affine transformations contributes to the histogram peaks. We can observe a similar effect on all presented histograms.

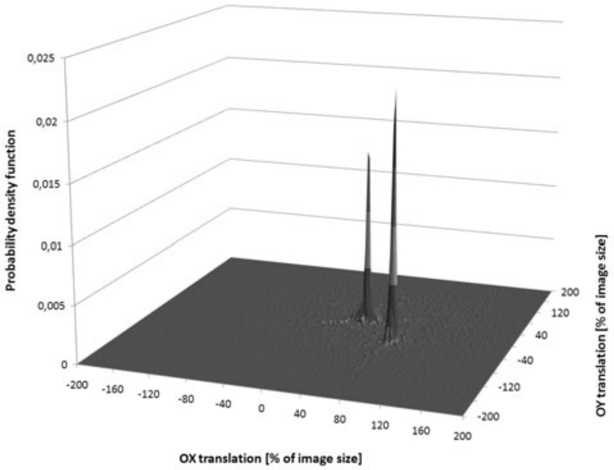


**Fig. 7** 2D translation probability density function. Similarly to the rotation histogram, there are two peaks representing two objects.

Next, we analyze the probability density function of 2D translations (which are not affected by SVD decompositions). The histogram shown in Fig. 7 represents $P_{svd}(P_X, P_Y)$ distribution. The range of modeled translations stretches from $-200\%$ to $200\%$ of image size. Larger translations are considered invalid and are cropped. Translations modeled up to $200\%$ of image size seem surprising but such a range can be justified by the properties of affine transformations (see Eq. 10). It should be noted that the linear component of the transformation may also introduce indirect translations (depending on the center of coordinates). The actual translation (applied *after* the linear mapping) compensates the effects of the indirect translations, and sometimes more than $100\%$ of image size has to be added to correct those indirect translations.

The analysis of Fig. 7 histogram provides illustrative examples. The *white bottle* objects has moved slightly right and downwards. Such a movement can be observed for the lower histogram peak in the figure. The *flu remedy pack* object is a more interesting case. The physical movement of the object between two images is slightly upwards and leftwards. If we locate the center of coordinates in the upper left corner of the image, both $x$ and $y$ coordinates of the object are reduced. However, we need

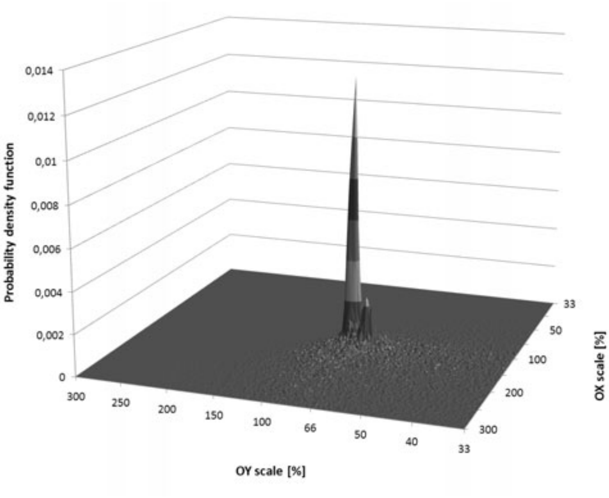


**Fig. 8** Scales (two dimensions) probability density function generated using SVD decomposition. There are two peaks, but one is much less distinctive.

to take into account a near 90° rotation, i.e. $u \approx -y + C$ and $v \approx x + F$. In such a case, *OX* translation values (*C*) would be positive, while *OY* translation values (*F*) – negative. We can observe such a result in the presented histogram.

The last presented histogram contains the scaling data. Fig. 8 shows the probability density function $P_{svd}(S_X, S_Y)$. As expected, its upper half is empty, because in the SVD decomposition we assign the larger *eigenvalue* to *OX* scale and the smaller eigenvalue becomes *OY* scale. Thus, no events can be generated $s_x < s_y$.

Similarly to the previous histograms, two peaks exist. However, one of the peaks is much less distinctive. There are two reasons for such a large difference in peak heights. The first reason (already mentioned) is a different number of triangles in both objects. In fact, the *flu remedy pack* is represented by over 4000 triangles, while *white bottle* contains less than 1000 triangles. The second reason is related to the distribution of these triangles. The *white bottle* object is not planar so that the perspective distortion is slightly different for various parts of the object. Thus, small differences in the scales exist and the triangles are distributed among several neighboring bins of the histogram. For the *flu remedy pack* object, most of the triangles fall into the same bin, and it results in a very high peak.

Altogether, histograms of decomposed affine transformations (either SVD or 3D decompositions) provide highly distinctive data about transformations relating similar fragments of two images. These data, when handled properly, can be used to detect such image fragments and, in fact, to detect objects or groups of objects present in both scenes. However, if these fragments do not change their relative positions, they will be detected as a single histogram peak because all of them are related by the same transformation.

### 3.5 Image Fragment Matching by Histogram Analysis

In the previous section we have shown how six dimensional probability density functions of decomposed affine transformations can be used to detect similar fragments (which usually represent the same objects) in images. To perform the detection, we just need to find high-density areas of the probability density function. We are actually interested not only in the density peaks, but also in their neighborhoods. The neighborhoods represent small distortions of the object (e.g. non-planar surfaces) or effects of perspective distortions.

Assume arbitrarily that we discuss only $P_{svd}$ density functions. However, exactly the same conclusions can be drawn for $P_{3D}$ density functions. In general, we have to model all dependencies of the probability density function. In particular, we cannot assume independent variables, i.e. $P_{svd}(\Gamma,\Theta,P_X,P_Y,S_X,S_Y) \neq P_{svd}(\Gamma)P_{svd}(\Theta)P_{svd}(P_X)P_{svd}(P_Y)P_{svd}(S_X)P_{svd}(S_Y)$. If scales are assumed independent of rotations and/or independent of translations, two or more objects the same value of just one parameter could not be properly differentiated and analyzed. Thus, we have to model $P_{svd}(\Gamma,\Theta,P_X,P_Y,S_X,S_Y)$ with all its complexities.

The largest problem with six dimensional densities is the efficient representation, affecting both for memory constraints and processing time. A classic histogram has $O(r^6)$ complexity, where $r$ is the resolution of the histogram dimensions (typical $r$ should be larger than 100). Such a histogram could not be even memorized in a modern personal computer. We need to consider other approaches to probability density function construction and analysis. Thus, we propose to build the histogram using hash-tables so that a linear complexity (both computational and memory) is achieved. Formally, the complexity is $O(p)$ where $p$ is the number of affine transformations, which linearly depends on the number of elementary geometrical objects – triangles or ellipses. Such an approach is similar to a classic kernel based non-parametric method (*Parzen window*) in probability density modeling. Both techniques have to iterate through all the data during the process of construction and analysis of the density function.

The first step in the histogram building is discretization of all processed data (a continuous probability density function converted into a discrete one). Discretization is performed uniformly, and its parameters are presented in Section 4.1. Each element of the hash-table is, in fact, a set of identical discretized transforms. Such a set can be uniquely addressed by its all six parameters. Since all available decomposed transformations are placed into the hash-table, we can easily access bins of the complete histogram bins by iterating over all data.

For each decomposed transformation we get all other transformations having the same parameters to calculate the discretized probability value. The probability value is used decide whether a histogram bin contains enough decomposed transformations to form an object. However, each transformations contributes to the density only as much as it weights, where the weight is determined by weights of geometric structures defining the transformation (see Section 3.1 for weighting details).

The decision on whether an object should be formed is made using thresholding, where the threshold value $t$ is arbitrarily set up. All histogram bins accumulating more than $t$ of weighted contributions from decomposed transformations are considered similar fragments of two images. Such a simple approach detects not only density peaks but also their neighboring areas (which are necessary to handle minor distortions of objects).

### *3.6 Histogram Bins Grouping*

Histogram bins grouping is the last operation in the proposed approach, in which only bins with sufficiently high contents are processed. Each histogram bin contains (apart from the affine parameters) a set of matched elementary geometric objects, which we have discussed in Section 3.2. Grouping is based on two types of data: neighborhood in histogram and similar locations of geometrical objects.

Histogram bin grouping consists of two separate steps. In the first step, we perform grouping according to neighborhood in the histogram. A simple *flood-fill* algorithm is used for this purpose. In terms of the *graph* model, two histogram bins belong to a single group *if and only if* there exists at least one path between them. To keep the computational complexity low, we use only axial neighborhood with up to 12 neighbors (in a 6D space). To effectively manage groups, we employ classic *union-find* algorithm.

After the neighborhood-based grouping, the localization-based grouping is performed. To do it effectively, we represent shapes of all generated groups (sets) as convex-hulls. We iterate through all pairs of convex-hulls and determine their intersection areas. Afterward, we calculate the relative intersection area, as a ratio between the intersection area to the total area of both convex-hulls. Two convex-hulls are grouped *if and only if* in both images the relative intersection areas are greater than a given threshold. In the ellipse based approach we also have to remove ”needle” shaped convex-hulls (they are artifacts) by applying Malinowska shape coefficient threshold.

## 4 Parameters, Evaluation and Applications

In this section we discuss practical aspects of the proposed method, including a limited experimental evaluation. The full evaluations are presented in our research papers, e.g. [21, 22]. We also discuss the parameter setup and the intuitive meaning of parameters. Finally, we highlight an exemplary application of our method.

### *4.1 Method Parameters*

Two types of the parameters have been used: general parameters and histogram resolution parameters (the full analysis can be found in [21]). The most important general parameters are histogram thresholds, i.e. $t_t$ for the triangle method and $t_e$ for the ellipse method. The higher values of these parameters, the less chance for false positives, but also a lower chance of object detection. In the triangle-based method, we also have a parameter $m$ responsible for the number of neighbor keypoints used in triangle building. A correct balance of these parameters has to be found. There are also additional parameters related to object construction and bin grouping. However, they much less affect the performances. All parameters of the method (and their recommended) values are shown in Tab. 1.

**Table 1** Default values of the method's main parameters.

| *Parameter* | *Meaning* | *Value* |
|---|---|---|
| $m$ | Number of keypoint neighbors | 60 |
| $t_t$ | Min. no. of triangles in histogram bins | 10 |
| $t_e$ | Min. no. of ellipses in histogram bins | 3 |
| $d_{min}$ | Min. triangle side, rel. to image size | 2% |
| $d_{max}$ | Max. triangle side, rel. to image size | 40% |
| $\alpha_{min}$ | Min. angle of a triangle | $3^o$ |
| $a_{thr}$ | Min. intersection area of convex hulls | 50% |
| $s_r$ | Max. value of Malinowska shape coefficient | 2 |

Parameters of the second group define size of the histogram bins (resolution). These parameters are directly responsible for geometric aspects of the near-duplicate retrieval. The higher the resolution, the lower margin for errors and more strict detection of planar surfaces. On the other hand, lower resolutions enable detection of slightly non-planar surfaces, introducing, however, a higher risk of false detections. Additionally, the issue of numerical errors in the affine transformation reconstruction and decomposition should be taken into account. Some parameters of the decomposed transformations are reconstructed with high quality, and others are significantly noised. Because the parameters of decomposed transformations are meaningful, it is possible to setup the resolution in such a way that the important details captured while the noise is filtered.

In the SVD decomposition, $\theta$ angle is the most sensitive parameter. It is responsible for modeling skew deformations, which are very difficult to capture. A similar level of sensitivity is noticed for the $\phi_f$ angle in the 3D decomposition. As a result, we lower the resolutions of these two parameters, and thus minimize the effects of inaccuracies. The main rotation angle $\gamma$ in the SVD decomposition is reconstructed with a high accuracy and can be modeled with a very high resolution. In case of 3D decompositions, $\phi_x$ and $\phi_z$ rotation angles are also sensitive because the inaccuracies, which exist in a single dimension in the SVD approach, are distributed among three dimensions. This is the major weakness of the 3D decomposition approach. However, we cannot reduce the dimensionality of these two variables, because we would also loose important geometric data.

Scalings in both decompositions are sensitive to perspective distortions, because perspective projections cannot be modeled within the affine transformation framework. To be able to compensate for such distortions to some extent, we deliberately lower the scale resolution. Translations are the least affected parameters and

**Table 2** Histogram resolutions (bin sizes) for various dimensions in SVD and 3D decompositions

| Triangles | | | | Ellipses | |
|---|---|---|---|---|---|
| SVD decomposition | | 3D decomposition | | SVD decomposition | |
| Dimension | Bin size | Dimension | Bin size | Dimension | Bin size |
| x transl. | 2% | x tr. (x res) | 4% | x transl. | 2% |
| y transl. | 2% | y tr. (y res) | 4% | y transl. | 2% |
| $\gamma$ angle | $2^o$ | $\phi_x$ angle | $1^o$ | $\gamma$ angle | $2^o$ |
| $\theta$ angle | $10^o$ | $\phi_z$ angle | $1^o$ | $\theta$ angle | $20^o$ |
| $z_x$ lin. scale | 10% | $\phi_f$ angle | $4^o$ | $z_x$ lin. scale | 10% |
| $z_y$ lin. scale | 10% | lin. scale | 10% | $z_y$ lin. scale | 10% |

we model them with high resolution. Resolutions of all histogram parameters (bin sizes) are shown in Tab. 2.

## 4.2 Method Evaluation

Both the triangle-based and the ellipse-based algorithm have been evaluated on several detectors and descriptors presented in Section 2 to show that the method is able to work effectively with various approaches to keypoint extraction. The presented results are a summary of our previous works [21, 22].

The triangle-based method uses only the location of keypoints. Shape of the key region is irrelevant, thus both circular and elliptical detectors may be used. In our evaluation we have used a range of detectors and descriptors (see Section 2). The circular detectors are Hessian-Laplace and SURF, while the elliptical ones are Harris-Affine and MSER. The tested descriptors are SIFT, GLOH, *moment invariants* and SURF. However, not all combinations of detectors and descriptors have been tested.

**Table 3** Averaged matching quality results for the triangle based approach using various combinations of local detectors (both circles and ellipses) and descriptors.

| *Score* | *unit* | HarAff<br>SIFT | HarAff<br>GLOH | HarAff<br>Mom | HesLap<br>SIFT | MSER<br>SIFT | SURF |
|---|---|---|---|---|---|---|---|
| | | O2O matching, SVD decomposition | | | | | |
| Precision | area | 0.96 | 0.96 | **0.97** | 0.94 | 0.95 | 0.90 |
| Recall | area | **0.64** | 0.51 | 0.47 | 0.51 | 0.53 | 0.49 |
| Precision | obj. | 0.96 | **0.97** | **0.97** | 0.95 | 0.94 | **0.97** |
| Recall | obj. | **0.81** | 0.71 | 0.70 | 0.69 | 0.69 | 0.62 |
| | | O2O matching, 3D decomposition | | | | | |
| Precision | area | 0.90 | 0.88 | 0.91 | 0.85 | 0.86 | 0.79 |
| Recall | area | 0.54 | 0.41 | 0.34 | 0.42 | 0.47 | 0.49 |
| Precision | obj. | 0.90 | 0.88 | 0.91 | 0.86 | 0.87 | 0.84 |
| Recall | obj. | 0.74 | 0.64 | 0.56 | 0.64 | 0.66 | 0.65 |

The experimental results are summarized in Tab. 3. It can be noted that in almost all cases the achieved precision is very high, both in terms of image area and object matching. However, Harris-Affine detector usually provides the best results. It reaches 97% (in a combination with moment invariants descriptor) for both types of measurement. This is a very satisfactory result, because it means almost no false positives. Apart from a very high precision, we also get high recall values. In case of area matching, it equals 64%, while in terms object matching it reaches 81%. However, *Harris-Affine* produces higher numbers of keypoints compared to *SURF* and *MSER* detectors so that the matching is slower. We would like to emphasize that our objective is not to compare or benchmark the keypoint detectors. On the contrary, we want to demonstrate that the proposed approach can work effectively with various kinds of detectors. For a comprehensive evaluation of keypoint detectors, we recommend further reading, e.g. [17].

We have also compared the proposed decompositions of affine transformations. Altogether, SVD decomposition performs slightly better. In particular, it better separates accurate and inaccurate parts of affine transformation reconstruction. The most difficult and inaccurate part of such reconstructions are skews. Thus, by reducing the skew angle dimensionality in the histograms, we are able to minimize these inaccuracies. The resolution of other dimensions can be kept high to avoid false positives.

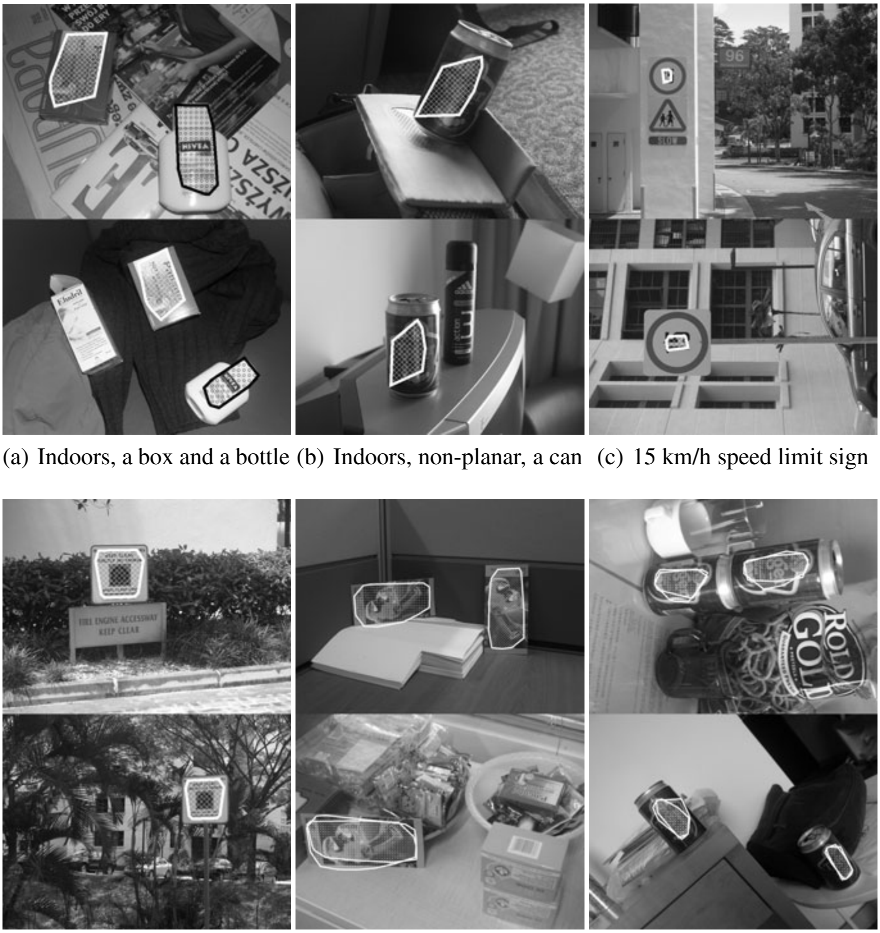
Such an approach is not possible in the 3D decomposition. Skews are entangled in all three elementary rotations and we cannot minimize inaccuracies without loosing important data. On the other hand, 3D decomposition follows the natural rules of 3D motion modeling and could be an asset for a structural analysis of scenes. Moreover, if we use high resolutions for all histogram dimensions, performances of both approaches are much more similar.

**Table 4** Averaged matching quality results for the ellipse based approach using various combinations of local elliptic based detectors and descriptors.

| *Score* | *unit* | MSER | MSER-RGB | HarAff | HarAff | HarAff |
|---|---|---|---|---|---|---|
| | | SIFT | SIFT | SIFT | GLOH | Mom |
| Precision | area | **0.92** | 0.91 | 0.84 | 0.86 | 0.80 |
| Recall | area | **0.50** | 0.48 | 0.48 | 0.39 | 0.40 |
| Precision | obj. | 0.93 | 0.93 | 0.95 | **0.96** | 0.92 |
| Recall | obj. | **0.68** | 0.65 | 0.64 | 0.55 | 0.58 |

A similar evaluation has been performed for the ellipse-based approach. However, in this case we could use only elliptical detectors: Harris-Affine and MSER. Apart from the classic MSER, we have also used a modified version of MSER, presented in [22]. We name this detector *MSER-RGB*. Its main idea is to improve time performances. Thus, we first calculate MSER regions in three RGB channels, and take only at most 333 largest ellipses from each channel. Such an approach provides the immense speedup over a classic solution. In general, the ellipse-based solution works faster than the triangle approach and we consider it a candidate for a real-time implementation, even though the results are slightly worse (the maximum performance drop for MSER is 3%). Detailed results are presented in Tab. 4.

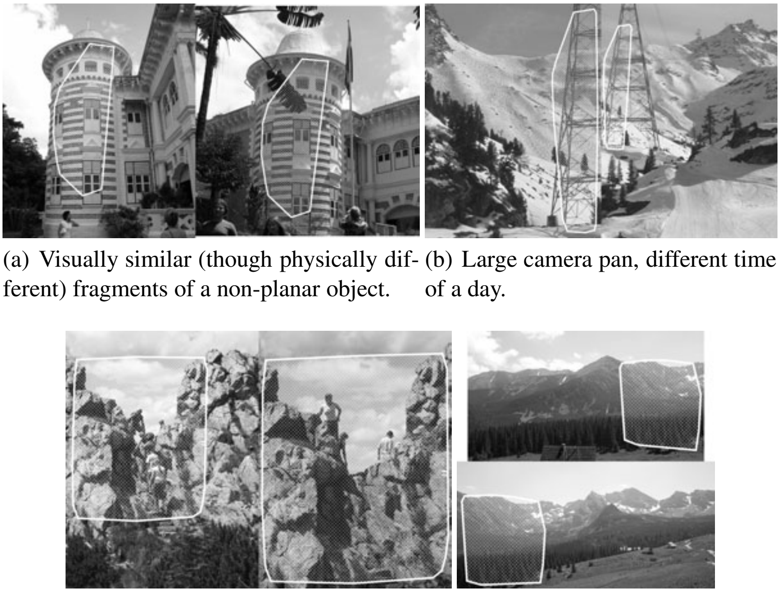
As expected, performances are more significantly affected by the detector selection than in the triangle approach. The best quality is achieved for MSER detector, because it provides the highest quality of ellipse parameters (moment-based estimation on pixel-based regions). Harris-Affine is much less effective, because of its iterative procedure of ellipse estimation. Thus, for the ellipse-based approach we clearly recommend MSER-based features.



(a) Indoors, a box and a bottle (b) Indoors, non-planar, a can (c) 15 km/h speed limit sign

(d) No-parking sign (e) Multiple object copies on one image, O2O (f) Multiple copies on both images, M2M. All four combinations are detected.

**Fig. 9** Examples of successfully matched objects. Various kinds of image transformations are present.



(a) Visually similar (though physically different) fragments of a non-planar object.

(b) Large camera pan, different time of a day.

(c) Camera zoom, some details changed.

(d) Large camera pan.

**Fig. 10** Complex scenes containing many non-planar objects. Objects are far away and appear as near-planar surfaces.

Apart from the above evaluation we also present a series of matching examples, containing both indoor and outdoor scenes. Fig. 9 demonstrates successful matches for both planar and non-planar objects. Matching of non-planar objects is possible due to effective grouping of neighboring bins. The proposed method is also able to handle multiple copies of the same objects. The multiple copies can be in only one image or in both images, but the second case is more difficult. Our experiments have shown that only M2M approach is able to match such images. We have also experimented with scenes containing multiple non-planar objects photographed from a different viewpoint and from a longer distance (they appear planar). The matching areas are almost always successfully detected, as shown in Fig. 10.

## *4.3 Forming Visual Classes – An Intended Application*

Finally, we briefly discuss an intended application of the proposed near-duplicate detection techniques [25]. Given a database of images, a binary relation can be established between near-duplicates from various images. Using such a relation, we propose visual clustering, which main goal would be to form visual classes. A single visual class would consist of mutually near-duplicate fragments. Such classes have (initially) no semantically meaningful names because near-duplicates are detected without any prior knowledge about the image contents. When the database accumulates more images of the explored environment, more classes can be formed and we can gradually replace the database images by the class models.

The idea can be presented using graphs, so that we first define the concepts of *graph cliques* and *node n-consistency*. A graph (or a subgraph of a graph) is a *clique* if it is fully connected, i.e. any two nodes are connected. A graph node is *n-consistent* if it forms a *clique* with $n-1$ other nodes of the graph. We propose that the visual class is formed when an $n$-consistent clique is found in the graph of near-duplicate fragments. Another fragment can be added to the existing class if:

- It is *n-consistent* with other nodes of the class, i.e. it is similar to $n$ already memorized fragments.
- It is *not m-consistent ($m > n$)* with other nodes of the visual class, i.e. it is not similar to too many memorized fragments.

If $n$ and $m$ parameters are properly defined, such an approach would prevent unnecessary class growth. When an image fragment which is similar to most ($\geq m$) of class representatives, it means that its visual form is already well represented within the class. On the other hand, if the new image fragment is only *n-consistent*, it may represent some novel and previously unseen properties of the class.

Of course, this is just an approach to the automatic class formation, a survey on the topic can be found in [30].

## 5 Summary

In this chapter we have presented an approach to detecting near-duplicate fragments in images of unknown contents, i.e. the concept of similarity is defined purely in the visual sense. We detect near-identical objects captured under various conditions, e.g. illumination changes, diversified background, camera settings, etc. No semantics is added to the processed fragments and, thus, our approach is general and applicable to various classes of images. The presented solution employs affine transformations to capture near-identical planar surfaces. However, we have enabled the method to detect slightly non-planar objects, e.g. cans, bottles. Detection of near-duplicate planar surfaces is based on the local approach. We first detect visually meaningful keypoints for the purpose of geometric reconstruction. During the reconstruction of image geometry, statistical methods are employed. To make the processed data meaningful, we decompose affine transformations into elementary transformations such as translations, rotations and scaling. As a result, we can identify meaningful peaks in probability density functions, representing physical transformations of fragments between two scenes.

Together with the detailed description of the methods, we also show a short experimental verification, and specify an exemplary application. Other prospective applications include: navigation in unknown and changing environments and assisting visually impaired people [25]. The basic and key concept in all these applications is the complete lack of any prior knowledge about the observed world. In fact, we assume that such systems should learn just by observing the environment.

**Acknowledgements.** The research presented in this paper is a part of a joint Poland–Singapore research program. The financial support of A*STAR Science & Engineering Research Council (grant 072 134 0052) and the Ministry of Science and Higher Education Republic of Poland (project 65/N-SINGAPORE/2007/0) is gratefully acknowledged.

## References

1. Abdel-Hakim, A., Farag, A.: Csift: A sift descriptor with color invariant characteristics. In: Proc. IEEE Conf. CVPR 2006, New York, vol. 2, pp. 1978–1983 (2006)
2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. Computer Vision and Image Understanding 110(3), 346–359 (2008)
3. Cheng, X., Hu, Y., Chia, L.-T.: Image near-duplicate retrieval using local dependencies in spatial-scale space. In: Proc. 16th ACM Int. Conf. on Multimedia, pp. 627–630 (2008)
4. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Proc. 4th European Conference on Computer Vision (ECCV 1996), Cambridge, UK, pp. 683–695 (1996)
5. Goldstein, H.: The euler angles. In: Classical Mechanics, 2nd edn., pp. 143–148. Addison-Wesley, Reading (1980)
6. Goldstein, H.: Euler angles in alternate conventions. In: Classical Mechanics, 2nd edn., pp. 606–610. Addison-Wesley, Reading (1980)
7. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conference, Manchester, pp. 147–151 (1981)
8. Heritier, M., Foucher, S., Gagnon, L.: Key-places detection and clustering in movies using latent aspects. In: Proc. 14th IEEE Int. Conf. Image Processing 2, pp. II.225–II.228 (2007)
9. Islam, M.S., Śluzek, A.: Relative scale method to locate an object in cluttered environment. Image and Vision Computing 26(2), 259–274 (2008)
10. Kannala, J., Salo, M., Heikkila, J.: Algorithms for computing a planar homography from conics in correspondence. In: British Machine Vision Conference (2006)
11. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for a local image descriptors. In: Proc. IEEE Conf. CVPR 2004, Washington, DC, pp. 506–513 (2004)
12. Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and sub-image retrieval. In: Proc. ACM Multimedia Conf., pp. 869–876 (2004)
13. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. 7th IEEE Int. Conf. Computer Vision, vol. 2, pp. 1150–1157 (1999)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
15. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. British Machine Vision Conference, Cardiff, pp. 384–393 (2002)
16. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision 60(2), 63–86 (2004)
17. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. PAMI 27, 1615–1630 (2005)
18. Mindru, F., Tuytelaars, T., van Gool, L., Moons, T.: Moment invariants for recognition under changing viewpoint and illumination. Computer Vision and Image Understanding 94(1-3), 2–27 (2004)
19. Moravec, H.: Rover visual obstacle avoidance. In: Proc. Int. Joint Conf. on Artificial Intelligence, Vancouver, pp. 785–790 (1981)

20. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences 2(2), 438–469 (2009)
21. Paradowski, M., Śluzek, A.: Matching Planar Image Fragments using Histograms of Decomposed Affine Transforms (2010), (Under second review in IEEE TPAMI)
22. Paradowski, M., Śluzek, A.: Detection of image fragments related by affine transforms: Matching triangles and ellipses. In: Proc. of ICISA 2010, Seoul, Korea, pp. 189–196 (2010)
23. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Trans. PAMI 19(5), 530–534 (1997)
24. Śluzek, A.: Zastosowanie metod momentowych do identyfikacji obiektów w cyfrowych systemach wizyjnych. WPW, Warszawa (1990)
25. Śluzek, A., Paradowski, M.: A vision-based technique for assisting visually impaired people and autonomous agents. In: Proc. of HSI 2010, Rzeszów, Poland (2010)
26. Xiao, J., Shah, M.: Two-frame wide baseline matching. In: Proc. 9th IEEE Int. Conf. on Computer Vision, pp. 603–609 (2003)
27. Xiong, Z., Zhang, Y.: A novel interest-point-matching algorithm for high–resolution satellite images. IEEE Transactions on Geoscience and Remote Sensing 47, 4189–4200 (2009)
28. Yang, D., Śluzek, A.: A low-dimensional local descriptor incorporating tps warping for image matching. Image and Vision Computing 28(8), 1184–1195 (2010)
29. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: Proc. 3rd Int. Symp. 3D Data Proc., Visualization and Transmission (3DPVT 2006), pp. 33–40 (2006)
30. Zhang, Y.-J.: Semantic-based visual information retrieval. IRM Press, Hershey (2007)
31. Zhao, W.-L., Ngo, C.-W.: Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. IEEE Trans. on Image Processing 2, 412–423 (2009)
32. Zhao, W.-L., Ngo, C.-W., Tan, H.-K., Wu, X.: Near-duplicate keyframe identification with interest point matching and pattern learning. IEEE Transactions on Multimedia 9(5), 1037–1048 (2007)

# Chapter 10
# Feature Analysis for Object and Scene Categorization

Jeremiah D. Deng

**Abstract.** Feature extraction and selection has always been an interesting issue for pattern recognition tasks. There have been numerous feature schemes proposed and empirically validated for image scene and object categorization problems, no matter it is for general-purposed applications such as image retrieval, or for specific domains such as medical image analysis. On the other hand, there are few attempts in assessing the effectiveness of these features using machine learning methods of feature analysis. We review some recent advances in feature selection and investigate the use of feature analysis and selection in two case studies. Our aim is to demonstrate that feature selection is indispensable in providing clues for finding good feature combination schemes and building compact and effective classifiers that produce much improved performance.

**Keywords:** object categorization, scene categorization, feature analysis.

## 1 Introduction

It is obvious that we use a multitude of visual clues to conduct day-to-day visual tasks such as object detection and scene classification. Over the last few decades there has been intensive research work carried out on related areas such as content-based image retrieval (CBIR), scene categorization, and object recognition. There have been many feature schemes proposed and widely applied in various studies. These include global or local feature descriptors derived from the colour, texture, and shape information within the image. Local descriptors such as SIFT and the bag-of-feature approach has found success in some recent work (Fei-Fei and Perona, 2005).

Jeremiah D. Deng
Department of Information Science
University of Otago, PO Box 56, Dunedin 9054, New Zealand
e-mail: jeremiah.deng@otago.ac.nz

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 225–244.
springerlink.com

A notably fast developing research direction is medical image annotation, enabled by recent advances of CBIR research. Various types of feature schemes are proposed (Mojsilovic and Gomes, 2002; Lehmann et al., 2004) and their performance compared using real-world datasets (Deselaers et al., 2008). Despite these promising results so far obtained, there is still a lack of feature analysis in the literature. Feature schemes are mostly borrowed from object recognition and CBIR experiences in general, and their effectiveness is assessed mostly empirically through trial-and-fail. There are few attempts in utilizing and assessing combined feature schemes for scene classification.

On the other hand, feature selection has received increasing attention in machine learning research. Although the ever-upgrading memory and CPU configurations have made nowadays computers more and more powerful, the amount of information, and sometimes the large dimensionality of datasets, still challenge the efficiency and even the effectiveness of data mining algorithms.

Feature selection as an important dimension reduction method can therefore make contribution from several aspects. First, with fewer features included in classifiers, training can be done less costly (both in time and memory). Indeed, with redundant and noisy data components removed, it is possible that better classification performance can be achieved. Furthermore, feature analysis can help us rank or select the features, and also give us clues about finding feature combinations that may produce potentially better performance.

In this chapter, it is not our intention to give a thorough survey on feature selection and analysis. Nor are we aiming at conducting a comprehensive comparison study to examine various features proposed in the scene and object categorization literature. Rather, we intend to use two case studies to demonstrate that feature selection can be useful in handling complicated image analysis tasks. In the following sections, first we will briefly review some relevant work. In Section 3, we take a look of a few existing feature selection algorithms. Section 4 follows to give two case studies in medical image categorization and wildlife scene categorization respectively, where the utilization of feature selection is explored. Some relevant feature schemes will be introduced and in some cases modifications are introduced. The result of feature selection and its implications are discussed. Finally, the paper is concluded along with a discussion on future directions.

## 2 Relevant Work

During recent years, there has been a growing interest in studying the statistics of natural images (Torralba and Oliva, 2003; Srivastava et al., 2003). Although most of the efforts are focused on using global image information such as Fourier spectral envelops (Oliva and Torralba, 2001), recent results questioned the effectiveness of using low-level power spectral information in classifying indoor and outdoor scenes (Fei-Fei et al., 2007).

From the perspective of natural and man-made scene discrimination, one of the early studies (Vailaya et al., 1998) employed a number of low level image features such as edge direction histograms, colour histograms, and DCT moments. (Luo and Boutell, 2005) modified the feature extraction of frequency spectra, using overcomplete ICA instead of PCA (Oliva and Torralba, 2001) and achieved improved performance. Global features constructed for modelling spatial layout properties within a scene has been found to be useful for rapid scene classification and for guidance of local focus needed for object recognition (Oliva and Torralba, 2006). The fractal properties of natural scenes have also been explored for constructing a feature scheme for natural vs man-made scene discrimination along with other texture features (Deng et al., 2008a). A relevant but slightly different direction is indoor-outdoor scene classification, also employing various colour and texture features (Szummer and Picard, 1998; Serrano et al., 2004; Payne and Singh, 2005).

Medical image analysis has become ever increasingly important. There have been quite a few studies made on automatic medical image categorization in recent years. A number of feature schemes have been developed or employed, including blob size (Mojsilovic and Gomes, 2002), grayscale co-occurrence and colour layout (Deselaers et al., 2008), Gabor filtering, edge histogram (Tian et al., 2008), local binary patterns (Tian et al., 2008), SIFT descriptor (Lowe, 2004; Deselaers et al., 2008), and fractal dimensions (Lehmann et al., 2004) etc.

In general, local features have found popularity in recent literature (Zhang et al., 2007). It is found that these local features are very effective in classifying textures and objects. A comprehensive survey of local features currently used in computer vision research has been given by (Li and Allinson, 2008). Apart from advanced machine learning techniques such as the kernel methods (Zhang et al., 2007), there is another notable trend that contributes to the success of local features. Under the influence of modern information retrieval research where a bag-of-words approach became very effective, a bag-of-features counterpart has become popular in visual scene and object recognition (Fei-Fei and Perona, 2005; Greenspan and Pinhas, 2007). This approach adopts local visual features such as SIFT descriptors and colour features, and then conducts clustering to construct the ‘visual words’. Images are then represented by a histogram of the occurrence of the visual words. New local feature schemes are found to be complementary and have formed a number of combinations (Zhang et al., 2007). We need, however, to be aware of the potential drawbacks of this approach. First, the computing cost becomes more challenging when more features need to be extracted and clustered. Secondly, with the high dimensionality of the feature data, the classifiers are difficult to train and their performance becomes subject to the ‘curse-of-dimensionality’ and may deteriorate rapidly. Another side-effect is that, due to the complexity of feature space, it is often hard to interpret the classification outcome. In particular, a lack of ability to 'explain' the outcome does not encourage the wide adoption such computer-based diagnosis systems in medical practice.

On the other hand, feature selection has been found to be indispensable to minimize classification error in many pattern recognition problems (Guyon and Elisseeff, 2003; Peng et al., 2005; Deng et al., 2008b). Using machine learning techniques, features can be ranked by information entropy-based indices, and an optimal subset can be searched out. For medical image categorization, there are some studies on empirically exploring the effectiveness of selected feature schemes, and it is found that some features are highly redundant to each other (Deselaers et al., 2008). A relevant study based on mutual information (Xu and Zhang, 2007) is conducted for general-purpose image categorization, but the feature pool for selection is very limited. To our knowledge, no one has attempted a comprehensive feature analysis for the problem of medical image categorization. This work, despite of its limited scope, is an attempt to fill this gap by assessing a few feature schemes using a machine learning approach.

## 3 Feature Selection

With various feature schemes proposed for object and scene categorization tasks, the sheer high-dimensionality becomes a challenge for building efficient classifiers. In fact, not only there may exist strong redundancy between these feature schemes, there may be some redundancy *within* individual feature schemes that can be explored. It is therefore necessary to conduct feature analysis and achieve dimension reduction through feature selection. In this study, we employ a few filter-based methods to rank or select potentially good features that allow for effective and efficient classification. The rest of this section presents a brief introduction to these methods.

### ReliefF

The ReliefF evaluator (Kira and Rendell, 1992) assesses the quality of attributes for classification. The difference between two values of an attribute $A_k$ in instances $i$ and $j$ is defined as:

$$\mathrm{diff}(A_k, i, j) = \frac{|A_k(i) - A_k(j)|}{\mathrm{max}A_k - \mathrm{min}A_k}. \tag{1}$$

The quality of $A_k$ for classification can therefore be estimated by checking the distance between instances found in the set of nearest neighbours $\Omega$. The estimate, denoted as $W(A_k)$, is defined as the difference between the probabilities of neighbours belonging either to the same class, or to different classes:

$$\begin{aligned} W(A_k) = {} & P(\text{diff}(A_k)|(i,j) \in \Omega, \text{class}(i) \neq \text{class}(j)) \\ & -P(\text{diff}(A_k)|(i,j) \in \Omega, \text{class}(i) = \text{class}(j)). \end{aligned} \tag{2}$$

Clearly, the bigger the $W(A_k)$ value is, the better attribute $A_k$ is for classification purpose.

## Information Gain

On the other hand, to assess the quality of a feature used for classification, a ranking approach based on statistical correlation can be adopted (Guyon and Elisseeff, 2003). In general, if a feature is relevant to the class label but is not redundant given the inclusion of other features, it is then a good feature.

Given a set of features $F = \{f_1, f_2, ..., f_n\}$, it is often the case that the features can be redundant and noisy at the same time. Hence it makes sense to evaluate the features and obtain a reduced feature set $S$ in order to leverage further data analysis processes such as classification. We can define the mutual information of two variables $x$ and $y$ as follows:

$$I(x,y) = \int\int p(x,y)\frac{p(x,y)}{p(x)p(y)}dxdy. \tag{3}$$

To rank a feature $f_i$, we need to calculate $I(f_i, c)$, i.e., the mutual information between the feature and the corresponding class label $c$. In fact $I(f_i, c)$ is often referred to as 'information gain' (IG). Based on information theory, a number of indicators can be developed to rank the features by their correlation to the class label. These include the information gain (IG) and symmetric uncertainty (SU) measures etc. (Guyon and Elisseeff, 2003). Combined with search algorithms, ranked feature sets can lead to effective feature schemes that give satisfactory and even improved classification performance.

## Minimal-Redundancy-Maximal-Relevance

Simply choosing the top-ranked features using the IG measures would be however sub-optimal, since not only these selected features can be rather relevant to each other, making the selection redundant in itself, but also other features, which are less relevant but still useful for classification, have less chance to be included in a selection of limited dimensionality.

As an improvement, in the minimal-redundancy-maximal-relevance (mRMR) approach (Peng et al., 2005), both the features average relevance to the class label

$$D = \frac{1}{|S|} \sum_{f_i \in S} I(f_i, c), \tag{4}$$

and the average mutual relevance between features

$$R = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i, f_j), \tag{5}$$

are taken into consideration to assess the feature selection, and the optimization criterion for the search process is to find a subset $S$ that gives $\max(D - R)$ or $\max(D/R)$. A incremental forward search method is usually applied to work out a near-optimal subset selection $S$. Consequently, the selected feature subset may not give the optimal classification performance.

## 4 Case Studies

We conducted two experiments as our case studies on feature analysis for image scene and object categorization:

1. Medical image categorization, whose aim is to automatically identify the following information by analyzing a radiographic image: modality, body orientation, body part, and biological system examined;
2. Wildlife scene categorization and novelty detection. This is to analyze the semantic content of a wildlife image and detect potential novelty.

In the following subsections we will introduce the experiment settings and feature schemes examined, and present the feature analysis outcome and the relevant classification performance.

### Medical Image Categorization

**Experiment settings.** We used the IRMA database (Deselaers et al., 2008) in this study. The database consists of 10,000 fully annotated radiographic images collected from medical routine at the RWTH Aachen University Hospital. The dataset has been widely used in a number of publications. Different from the past usage (Deselaers et al., 2008), here we consider much finer classification, with 115 classes to model. Also, cross validation is used so as to better assess the quality of feature schemes. Some example images are shown in Fig. 1.

**Feature schemes.** Due to the particularities of medical imaging, medical image analysis and annotation remain challenging tasks. Often generic feature descriptors inherited from CBIR research are employed to model the image categories. It is not our intention to evaluate all the feature schemes in literature. Especially, colour image features proposed in generic image

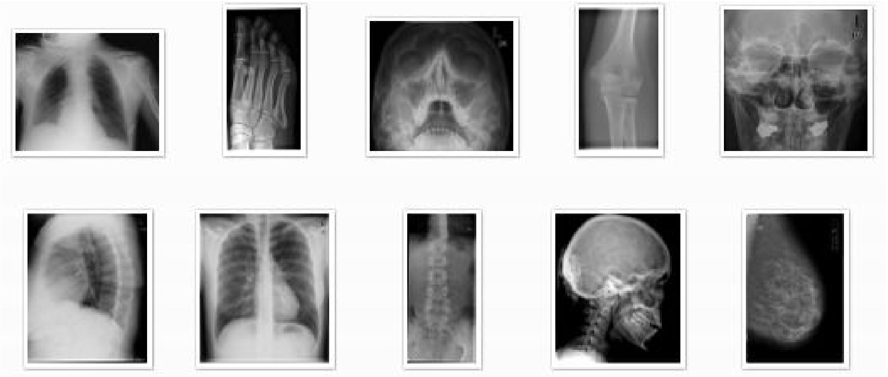


**Fig. 1** Example images in the IRMA database.

retrieval settings are not so relevant to radiographic images taken in grayscale. Furthermore, we limit our scope to histogram descriptors derived from global or regional image statistics.

**MPEG-7 edge histogram descriptor (EHD).** The EHD is a feature descriptor widely used in CBIR research (Manjunath et al., 2001). It is found to be effective especially when being combined with other descriptors (Deselaers et al., 2008). EHD captures the spatial distribution of edges on different orientations. There are six types of edges to be detected: vertical, horizontal, 45° diagonal, 135° diagonal, non-directed, and no edge. Edge detection is conducted over $2 \times 2$ macro-blocks split from the image, whose masks are shown in Fig.2. Edge statistics, collected into a 6-bin histogram over different edge types, are aggregated from these macro-blocks. Normally, an image is partitioned into $4 \times 4$ sub-images, each generating a sub-image edge histogram. Concatenating the edge histogram vectors results in a global edge histogram of 96 dimensions. Semi-global and global statistics can also be aggregated from the relevant macro-blocks and included in the final descriptor. We consider the global EHD only in this study.

$$\begin{array}{|c|c|}\hline 1 & -1 \\ \hline 1 & -1 \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline 1 & 1 \\ \hline -1 & -1 \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline 1 & -1 \\ \hline -1 & 1 \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline \sqrt{2} & 0 \\ \hline 0 & -\sqrt{2} \\ \hline\end{array} \qquad \begin{array}{|c|c|}\hline 0 & \sqrt{2} \\ \hline -\sqrt{2} & 0 \\ \hline\end{array}$$

**Fig. 2** Macroblock masks used for edge detection in EHD. Edge orientations are, clock-wise: vertical, horizontal, isotropic, 45 and 135 degrees.

**Canny Edge Descriptor (CED).** The MPEG-7 edge histogram does not take into consideration of edge intensity. Medical images, since without using colours, often bear very weak contrast and edge intensity may vary. For this reason, we also consider using the Canny edge detector (Canny, 1986).

The edge detection algorithm in Canny's algorithm undergoes a number of steps. First, the image is convolved with a Gaussian filter for noise reduction. Edge intensity is then decided by taking gradient on the smoothed image. The edge intensity can be defined on the image gradients on X- and Y-direction:

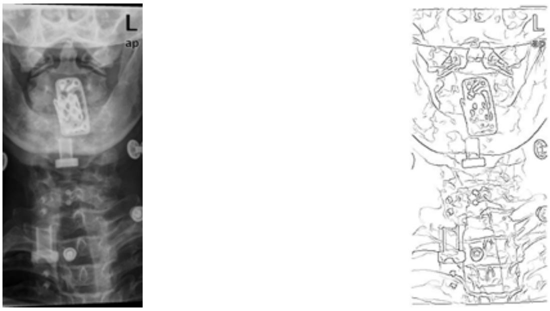
$$|G| = |G_x| + |G_y|. \tag{6}$$

And the edge direction can also be derived as:

$$\theta = \text{atan}(\frac{|G_y|}{|G_x|}). \tag{7}$$

This is then followed by a tracking process that extracts pixels with the maximal gradient magnitude. Unlike other edge detectors (including the MPEG-7 EHD) that rely on a single threshold for edge detection, Canny's algorithm employs a hysteresis process with two thresholds to suppress noises.

With edge extracted by the Canny algorithm, a better edge descriptor can be developed. For each image, we extract the the proportion of pixels marked as 'edge', as well as the proportions of edges on 10 orientations and 10 intensity levels. This results in a feature code of 21 dimensions, denoted as 'CED1'.



(a) (b)

**Fig. 3** (a) An example image; (b) Edge detected with orientation and intensity information.

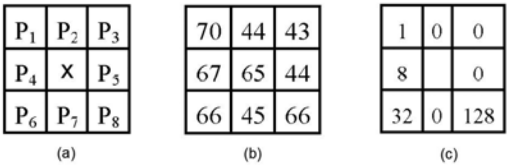
To assess the spatial difference of edge information, we cut the image horizontally into 3 equal parts, and concatenate the edgy pixel proportions and the orientation histograms. The edge intensity histogram, however remains to be extracted globally. This gives us a second descriptor, noted as 'CED2', of 43 dimensions.

**Local binary patterns (LBP).** LBPs have been shown to be a simple but effective method of texture discrimination based on a simplification of directional grayscale difference histogram (Ojala et al., 1996). Consider a local

$3 \times 3$ window centered at the current pixel, as shown in Fig. 4. We denote the central pixel as $p_c = (x, y)$, and the set of its neighbour pixels within a radius of $R$ as $\Omega = p_1, p_2, ..., p_N$. A binary weight is assigned to the neighbour cell using the given index, if the corresponding grayscale intensity is greater than that of the central pixel. The LBP code can be constructed from the following weighted sum, after comparing the intensity of $p$ and its $N$ neighbours:

$$LBP(p) = \sum_{n=1}^{N} [\mathrm{sgn}(I(p_c) - I(p_n)) + 1] 2^{n-2}, \tag{8}$$

where $I(p_i)$ denotes the intensity of Pixel $p_i$. As an example, the pattern shown in Fig.4 will give a LBP value of $1 + 8 + 32 + 128 = 169$. The LBP values of all image pixels, ranging between 0 and 255, form a histogram for the image. In a standard implementation, non-uniform values (i.e., with odd number of binary transitions) are combined into one unit, joining the rest 58 uniform values to form a LBP histogram of 59 dimensions.



| | | |
|---|---|---|
| $P_1$ | $P_2$ | $P_3$ |
| $P_4$ | X | $P_5$ |
| $P_6$ | $P_7$ | $P_8$ |

(a)

| | | |
|---|---|---|
| 70 | 44 | 43 |
| 67 | 65 | 44 |
| 66 | 45 | 66 |

(b)

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 8 | | 0 |
| 32 | 0 | 128 |

(c)

**Fig. 4** Calculation of LBP. (a) The $3 \times 3$ window; (b) Example grayscale values in the window; (c) the window thresholded by the central pixel and weighted by binary weights.

**MPEG-7 colour layout descriptor (CLD).** The CLD captures the spatial distribution of colour of an image and has been found to be useful in CBIR queries such as query-by-sketch and query-by-examples. CLD is defined as representative colours in YCrCb space on an $8 \times 8$ grid followed by DCT. The DCT operation was intended to make the descriptor more compact for fast database queries.

Different from previous usages in medical image annotation, the original implementation of CLD in MPEG-7 XM (Manjunath et al., 2001) is not employed. Firstly, since radiography images are of grayscale only, there is no need for colour space conversion. Also in our implementation of CLD we skipped the DCT and quantization process, using the average grayscale intensity value in each grid cell directly. This simplified version of CLD, denoted as ‘sCLD’, is a 64-dimension vector of floating point values.

For example, the sCLD of the image in Fig.3(a) is displayed as a grayscale block image shown in Fig.5.

Favourable results have been reported in (Tian et al., 2008) (Jeanne et al., 2009) where all the images were pre-scaled to $512 \times 512$ or $128 \times 128$ in size.

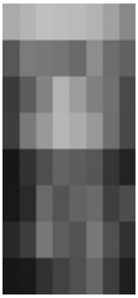


**Fig. 5** Example of sCLD, extracted from the image shown in Fig.3(a).

Since the aspect ratio of the original images is not kept with rescaling, we suspect that artificial information (rather than features extracted from the images per se) was introduced. In this study, we do not scale the images at all.

To allow for easy comparison with previous results, we employed nearest-neighbour classifiers and support vector machines (SVM), the same as in (Güld and Deserno, 2008) and (Deselaers et al., 2008).

**Feature ranking.** The effectiveness of various feature schemes is first assessed using mutual information based evaluators (Peng et al., 2005). Because of the large scale of the datasets, both vertically (i.e., with 10,000 samples) and horizontally (e.g., often with more than 100 dimensions), we have obtained only partial results on selected feature schemes. Here we report the results on the EHD, sCLD and CED2 features using the following evaluator: Information Gain (IG), ReliefF and mRMR. The first 30 top-ranked features are listed in Table 1. The numbers following the feature code indicate the ordered element of the relevant feature vector. For instance, 'EHD8' means the the 8-th element of the EHD feature.

From these results it can be seen that different attribute evaluators have good agreement with each other, despite some difference in their reported ranking order. Among the first 30 attributes, all feature schemes have strong presence. While CLD dominates in the outcome of the first two evaluations, EHD and CED also take good portions for mRMR probably due to the effect that the mRMR evaluator takes redundancy within feature selection into account.

**Classification results.** Next, classifiers are trained on datasets extracted by different feature schemes. The average prediction accuracies in 10-fold cross validation are recorded. The performance of different feature schemes is summarized in Table 2. Three classifiers are tested across various feature schemes. The complexity value of all SVM models is set as 100.

**Table 1** Top 30 attributes ranked by the three evaluators. Here 'CLD' stands for sCLD, and 'CED' stands for CED2. Numbers following the code indicate the relevant component in that feature vector.

| Rank | IG | Attr. | ReliefF | Attr. | mRMR | Attr. |
|---|---|---|---|---|---|---|
| 1 | 0.927 | CLD5 | 0.161657 | CLD5 | 0.484 | EHD8 |
| 2 | 0.819 | CLD4 | 0.148411 | CLD59 | 0.367 | CED12 |
| 3 | 0.780 | CLD59 | 0.148082 | CLD4 | 0.266 | CED6 |
| 4 | 0.763 | CLD17 | 0.143255 | CLD6 | 0.269 | EHD6 |
| 5 | 0.757 | CLD6 | 0.138884 | CLD60 | 0.177 | CED2 |
| 6 | 0.750 | CLD53 | 0.137485 | CLD17 | 0.179 | EHD7 |
| 7 | 0.737 | CLD9 | 0.136569 | CLD58 | 0.179 | CED9 |
| 8 | 0.736 | CED6 | 0.134569 | CLD61 | 0.187 | CED14 |
| 9 | 0.725 | CLD60 | 0.132183 | CLD9 | 0.148 | CED10 |
| 10 | 0.720 | EHD1 | 0.131201 | CLD62 | 0.145 | CED3 |
| 11 | 0.712 | CLD62 | 0.124875 | CLD63 | 0.153 | CED15 |
| 12 | 0.707 | CLD61 | 0.116896 | CLD25 | 0.134 | EHD3 |
| 13 | 0.706 | CED9 | 0.113067 | CLD53 | 0.133 | CLD57 |
| 14 | 0.704 | CLD25 | 0.112321 | CLD7 | 0.147 | CED1 |
| 15 | 0.696 | EHD2 | 0.11155 | CLD52 | 0.126 | CED7 |
| 16 | 0.694 | CLD24 | 0.108843 | CLD16 | 0.125 | CLD4 |
| 17 | 0.687 | CLD58 | 0.10849 | CLD24 | 0.119 | CED13 |
| 18 | 0.679 | CED3 | 0.105808 | CLD50 | 0.113 | CED16 |
| 19 | 0.662 | CLD52 | 0.104584 | CLD15 | 0.112 | CED8 |
| 20 | 0.655 | CLD32 | 0.104034 | CLD51 | 0.11 | EHD4 |
| 21 | 0.650 | EHD7 | 0.103739 | EHD8 | 0.108 | CLD52 |
| 22 | 0.638 | CLD15 | 0.102721 | CLD3 | 0.108 | CLD23 |
| 23 | 0.620 | CED10 | 0.101651 | CLD64 | 0.099 | CED5 |
| 24 | 0.618 | CLD63 | 0.099635 | CLD54 | 0.098 | CLD5 |
| 25 | 0.608 | CED2 | 0.099609 | CLD33 | 0.094 | CLD56 |
| 26 | 0.608 | CLD40 | 0.095902 | CLD55 | 0.094 | EHD5 |
| 27 | 0.604 | CLD45 | 0.095897 | CLD32 | 0.093 | CED4 |
| 28 | 0.596 | CLD16 | 0.094111 | CLD45 | 0.091 | CLD12 |
| 29 | 0.592 | CLD10 | 0.093179 | CLD10 | 0.091 | CLD49 |
| 30 | 0.590 | CLD33 | 0.091933 | CLD49 | 0.092 | CED17 |

Our experiment results show that the proposed Canny descriptor CED1 outperforms the MPEG-7 EHD significantly. Due to the spatial information included, CED2 performs even better than CED1. This, however, is compromised by a higher dimensionality and consequently, a more costly computation process.

Since texture descriptors include more information than edge descriptors, it is not surprising to see that the LBP outperforms all the edge descriptors. Actually its performance is better than that of Gabor filtering as reported in (Deselaers et al., 2008). The texture descriptors dimensionality is however much greater than the edge descriptors.

**Table 2** Performance comparison of the feature schemes. The classification accuracy and the corresponding F-measure are reported for each feature scheme - classifier combination.

| Feature scheme | Dimensions | Performance: Accu.(%)/F-meas. | | |
|---|---|---|---|---|
| | | 1-NN | 10-NN | SVM |
| EHDg | 8 | 29.7/0.30 | 38.2/0.35 | 41.3/0.36 |
| CED1 | 21 | 43.0/0.42 | 47.8/0.42 | 56.1/0.49 |
| CED2 | 43 | 57.4/0.57 | 60.0/0.57 | - |
| Gabor (Deselaers et al., 2008) | - | 55.1 | - | - |
| LBP | 59 | 67.5/0.67 | 67.2/0.64 | 77.2/0.76 |
| CLD (Deselaers et al., 2008) | 64 | 52.3 | - | - |
| sCLD | 64 | 72.6/0.72 | 73.4/0.70 | 77.5/0.77 |
| sCLD+CED1 | 85 | 72.1/0.75 | 75.0/0.73 | 81.0/0.80 |
| sCLD+LBP | 123 | 76.9/0.76 | 75.6/0.74 | 83.4/0.73 |
| sCLD+LBP / mRMR | 80 | 76.4/0.76 | 75.1/0.73 | 82.3/0.82 |
| sCLD+LBP / PCA | 34 | 77.4/0.77 | 76.8/0.75 | 82.5/0.82 |

To contrast our results on feature analysis with the classification performance, it is interesting to note that despite the relatively low classification accuracy achieved by individual feature scheme, EHD and other feature schemes all present positive contribution to category prediction. This also suggests that potentially better performance can be achieved through combining some of these feature schemes.

On the other hand, the modified CLD, despite being a simple descriptor reporting spatially-located average intensity, gives the best performance. This is a significant improvement over previously reported results (Deselaers et al., 2008), noting that they used the original CLD implementation on the same dataset. The CLD performance is slightly lower than what was reported in (Jeanne et al., 2009) using SVM.

To find out whether a joint feature scheme can improve the performance, we also tested two feature combinations: ‘sCLD+CED1’, ‘sCLD+LBP’. The latter scheme, consisting of the best two individual feature sets, achieved a much enhanced accuracy of 83.4% using SVM. Even though these are rather *different* features, we attempted feature reduction on this combination, using mRMR and PCA. The results are quite interesting. The reduced feature sets manage to achieve the performance level of the original combined set of 123 dimensions. Despite being an unsupervised algorithm, PCA has delivered a successful dimension reduction (down to 34 dimensions while keeping 95% of variance), indicating significant redundancy in the joint feature set.

## Wildlife Scene Categorization and Novelty Detection

The second case study we present here is a wildlife image analysis task. The goal is to categorize the semantic context of scenes and detect the potential novelty, e.g., hunting scenes or artificial scenes (Yong et al., 2010). Possible application of such a technique can be multimedia content management as well as wildlife monitoring.

**Experiment settings.** We took 172 images from the ImageNet dataset (Deng et al., 2009) and from Google Image Search. These wildlife images usually feature one animal in scene and contain relatively simple semantic context, each belonging to one of the following 4 scene types (number of instances for each type in brackets): dolphin (39), elephant (43), penguin (45) and zebra (45). Image sizes vary from 200 × 154 to 1024 × 768. Some sample images are shown in Fig. 6. For block labelling we have therefore 14 classes, including both the background and animals in foreground. After

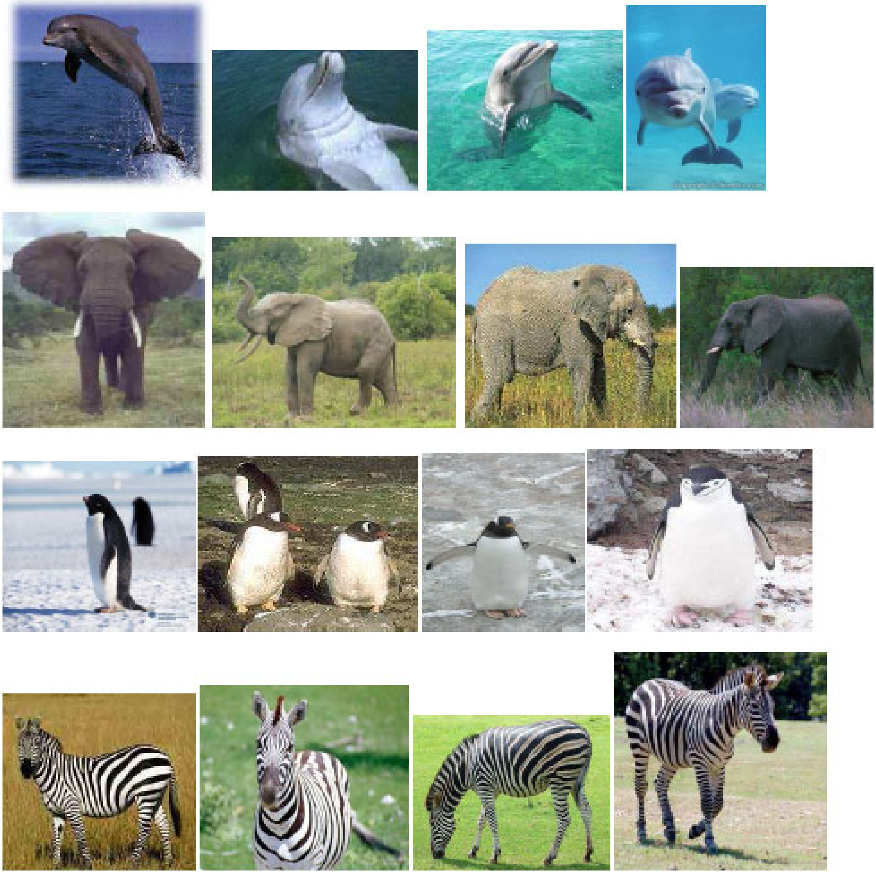


**Fig. 6** Samples of training images from four scene classes: ‘dolphin’, ‘elephant’, ‘penguin’ and ‘zebra’.

segmentation, there are over 50,000 image blocks extracted. We used an image annotation tool to manually annotate image segments and the blocks within each segment inherit the segment's label. We randomly chose 5,000 images for feature analysis experiments.

First, scene images are segmented into homogeneous regions using an image segmentation algorithm. A scene usually contains multiple objects of different visual characteristics. The segmentation facilitates the detection or classification of objects. Those segments will then be classified manually as the ground truth for training the classifiers. Each segment image will be tiled into blocks of size 32 × 32. Image blocks falling out of the segment edges will be ignored. For semantic analysis of new images, we have found it is more effective to train classifiers on segment blocks instead of on segment images directly.

Different from previous studies (Mojsilovic and Gomes, 2002; Greenspan and Pinhas, 2007), we don't adopt the bag-of-words approach to cluster the local image features and build histograms on the local feature labels. Rather, with the help of limited number of objects and scene types, the locally extracted block features are directly classified and assigned a semantic label. The scene category can then be modelled based on the co-occurrence of the semantic labels. Furthermore, scene novelty can be detected by examining the one-class distribution of each scene category. Therefore, this approach not only gives us the capability of handling scene modelling and novelty detection at the same time, but also it limits the feature evaluation process within the classification scenario (Yong et al., 2010). Here, we concentrate on the initial stage of block labelling and investigate possible feature selection.

### Feature Schemes

**LUV colour histogram.** Visual features are then extracted from image segment blocks. First we employ the LUV colour histogram to encode the colour information of image blocks. Colour histograms are found to be robust to resolution and rotation changes (Ma and Zhang, 2003). Our previous work (Deng and Zhang, 2005) found that other colour descriptors such as the Dominant Color Descriptor (Manjunath et al., 2001) are not so effective for object recognition. The LUV colour space is adopted because it models humans perception of colour similarity very well, and it is also machine independent. Each colour channel is quantized with the same interval, thus we have 20 bins for the L channel, 70 bins for U-channel, and 52 bins for V-channel respectively. The standard deviation of the LUV histogram values is also calculated. The LUV histogram feature therefore has taken 143 dimensions.

**Haralick texture features.** Texture features extracted from the image blocks are also included as local features. We compared Edge Histogram Descriptor and Gabor filtering features with the Haralick features (Haralick et al., 1973), and the latter performed the best in our test and are therefore adopted in further experiments.

The Haralick texture features consist of a few statistical properties based on the gray scale co-occurrence matrix. The image block, denoted as $B$, is first converted to gray scale. The co-occurrence matrix is a two-dimensional histogram composed from the pairwise statistics of gray scale co-occurrence among adjacent pixels. Given two grayscale levels $i$ and $j$, the co-occurrence of these two levels over an offset $(\Delta x, \Delta y)$ is defined as

$$C_{\Delta x, \Delta y}(i,j) = \sum_{x \in B} \sum_{y \in B} \begin{cases} 1, & \text{if } p(x,y) = i \;\&\; p(x+\Delta x, y+\Delta y) = j \\ 0, & \text{otherwise} \end{cases} . \tag{9}$$

Four orientations of the offset are considered, each giving a co-occurrence matrix for the image block. A total of 13 statistical measures, including angular second moment, contrast and correlation etc., can be calculated for each co-occurrence matrix (Haralick et al., 1973). The mean and deviation values of each of these 13 measures over the four orientations, form a feature vector of 26 dimensions. We denote the mean value and the deviation of the $X$-th moment value as HARm$X$ and HARd$X$ respectively.

**Feature combination.** Finally, the colour and texture features are concatenated together, giving a feature vector of 169 dimensions to represent an image block. Through manual labelling of image segments, semantic ground truth is assigned to the training images. The image blocks inherit semantic labels from their corresponding segments. Their feature codes along with the relevant class labels are used to train object classifiers.

**Feature analysis.** The first top 30 features ranked by their IG values are listed in Figure 7. As we see the feature list is dominated by the Haralick features, but there exist also important LUV features. Another notable phenomenon is that after 100 attributes the IG values stop being positive.

For our 14-class classification problem, we remove all LUV and Haralick features that have zero IG values, resulting in a 100-dimension feature set. Using a 1-NN classifier the classification accuracy is achieved at 88.6% in 10-fold cross validation, the same as using the full feature set of 169 attributes in total.

Can this approach of histogram features selection scale to general object and scene classification? Probably not. It is unlikely to expect many LUV histogram elements to be redundant for image patches of all types of background and foreground. However, a potential workable approach, as revealed

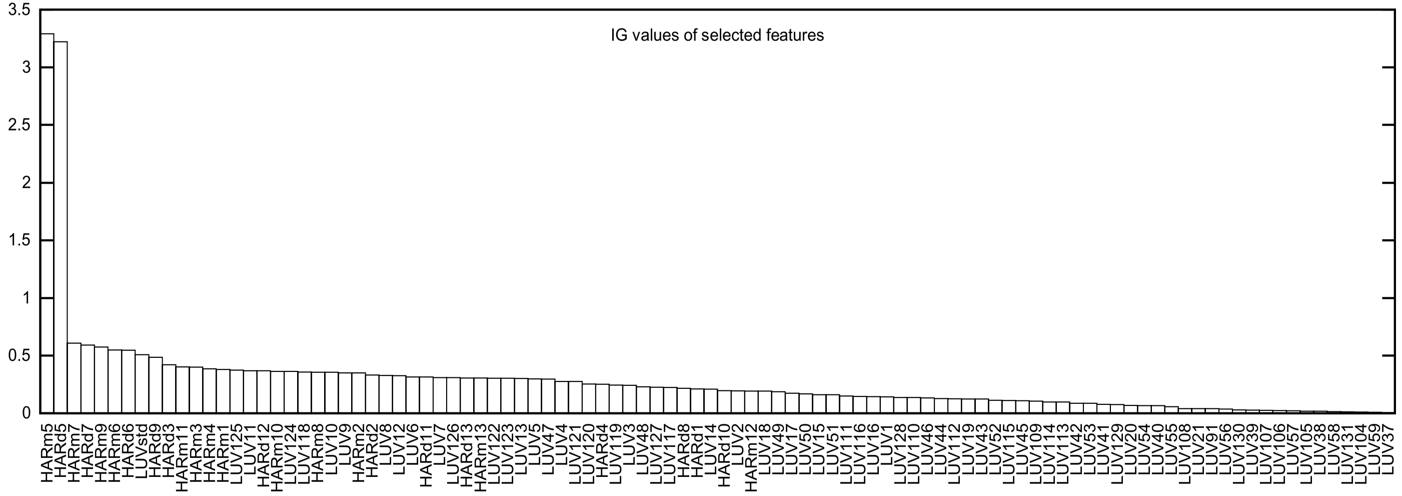


**Fig. 7** The IG values of the first 60 features. While there is a significant drop after the first few, the IG values of many features remain above zero.

by our experiments, is that we can use some kind of hierarchical classification systems (Cesa-Bianchi et al., 2006; Fan et al., 2008) that work in a divide-and-conquer manner, where each classifier can be trained using dimension-reduced colour histogram features and texture features.

## 5 Discussion and Conclusion

With the advances of digital imaging and Internet technologies, the challenge on effective retrieval and mining of the ever-increasing image data is unprecedented. It is highly desirable to develop automatic image annotation systems that can enable more efficient management and more effective utilization of these digital resources. There have been many feature schemes proposed for image scene and object categorization in general, and also for specific domains such as medical image annotation. While most previous authors have verified the effectiveness of these feature schemes in their empirical studies, we feel the work on feature analysis and selection is still lacking and therefore the potential of these features and their combinations has not been fully explored.

We argue that by selecting the most useful features that are most relevant to the image's categories and less redundant to each other, we can build more compact classifiers that remain most effective, as shown in the two case studies presented in this chapter.

Our feature analysis experiments, despite of limited scope, reveal the effectiveness as well as the limitation of different low-level features. There seems to be a necessity of combining feature schemes. Feature analysis in this regards may give us better ideas in finding feature combinations that are likely to be successful in solving complicated categorization problems.

On the other hand, since features are usually extracted from different modalities (e.g., colour and shapes), there is limited room of dimension reduction. The dimensionality of the combined datasets, even after feature selection, can become formidable if we simply concatenate all useful features into one long vector. Classifier construction would be challenged because of the potentially very high dimensionality. This puts classifier combination into a more desirable position (Ko et al., 2007). Ensembles of classifiers, with each classifier trained on an individual feature scheme and then combined to produce the classification decision, may be more promising in delivering better performance. So far we haven't seen much development on this direction for scene and object categorization, and our attempts in using existing combination schemes such as AdaBoost and Random Forest implementations in Weka (Hall et al., 2009) did not produce promising outcome, but we believe it remains a relevant research direction for the future.

## Acknowledgment

The author thanks Prof. Terence Doyle, Dunedin School of Medicine, for his valuable inputs on medical imaging, and Vivian Yong, for the provision of the wildlife image dataset. This work is supported by Information Science Research Grant and UORG-2010, University of Otago.

## References

Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence 8, 679–714 (1986)

Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Hierarchical classification: combining Bayes with SVM. In: Proceedings of the 23rd international conference on Machine learning, ICML 2006, pp. 177–184. ACM, New York (2006)

Deng, D., Zhang, J.: Combining multiple precision-boosted classifiers for indoor-outdoor scene classification. In: Proc. of ICITA 2005, vol. I, pp. 720–725. IEEE Computer Society Press, Los Alamitos (2005)

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE Computer Society, Los Alamitos (2009)

Deng, J.D., Brinkworth, R., O'Carroll, D.: Assessing the naturalness of scenes: an approach using statistics of local features. In: Proc. of IVCNZ. IEEE Explore, pp. 1–6 (2008a)

Deng, J.D., Simmermacher, C., Cranefield, S.: A study on feature analysis for musical instrument classification. IEEE Trans. System, Man and Cybernetics - Part B 38(2), 429–438 (2008b)

Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. Information Retrieval 11, 77–107 (2008)

Fan, J., Gao, Y., Luo, H., Satoh, S.: New approach for hierarchical classifier training and multi-level image annotation. In: Satoh, S., Nack, F., Etoh, M. (eds.) MMM 2008. LNCS, vol. 4903, pp. 45–57. Springer, Heidelberg (2008)

Fei-Fei, L., Iyer, A., Koch, C., Perona, P.: What do we see in a glance of a scene. Journal of Vision 7(1-10), 10:1–29 (2007)

Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE CVPR 2005, vol. 2, pp. 524–531. IEEE Computer Society, Los Alamitos (2005)

Greenspan, H., Pinhas, A.: Medical image categorization and retrieval for pacs using the gmm-kl framework. IEEE Trans. Inform. Tech. in BioMedicine 11, 190–202 (2007)

Güld, M.O., Deserno, T.M.: Baseline results for the imageclef 2007 medical automatic annotation task using global image features. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 637–640. Springer, Heidelberg (2008)

Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)

Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 3, 610–621 (1973)

Jeanne, V., Unay, D., Jacquet, V.: Automatic detection of body parts in x-ray images. In: Proc. Workshop on Mathematical Methods in Biomedical Image Analysis, CVPR 2009, pp. 25–30 (2009)

Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Proceedings of International Conference on Machine Learning, pp. 249–256 (1992)

Ko, A.H., Sabourin, R., de Souza Britto Jr., A., Oliveira, L.: Pairwise fusion matrix for combining classifiers. Pattern Recognition 40(8), 2198–2210 (2007); Part Special Issue on Visual Information Processing

Lehmann, T.M., Guld, M.O., Thies, C., et al.: Content-based image retrieval in medical applications. Methods of Information in Medicine 43(4), 354–361 (2004)

Li, J., Allinson, N.M.: A comprehensive review of current local features for computer vision. Neurocomputing 71(10-12), 1771–1787 (2008)

Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Computer Vision 60(2), 91–110 (2004)

Luo, J., Boutell, M.: Natural scene classification using overcomplete ICA. Pattern Recognition 38, 1507–1519 (2005)

Ma, Y.-F., Zhang, H.-J.: Contrast-based image attention analysis by using fuzzy growing. In: Proceedings of the 11th ACM International Conference on Multimedia, Multimedia 2003, pp. 374–381. ACM, New York (2003)

Manjunath, B., Ohm, J., Vasudevan, V., Yamada, A.: Colour and texture descriptors. IEEE Trans. on Circuits and Systems for Video Technology 11(6), 703–715 (2001)

Mojsilovic, A., Gomes, J.: Semantic based categorization, browsing and retrieval in medical image databases. In: IEEE Inter. Conf. on Image Processing (ICIP 2002), vol. III, pp. 145–148 (2002)

Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)

Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. Jour. Computer Vision 42(3), 145–175 (2001)

Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. Progress in Brain Research 155, 23–36 (2006)

Payne, A., Singh, S.: Indoor vs. outdoor scene classification in digital photographs. Pattern Recognition 38(10), 1533–1545 (2005)

Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Analysis and Machine Intelligence 27, 1226–1238 (2005)

Serrano, N., Savakis, A., Luo, J.: Improved scene classification using efficient low-level features and semantic cues. Pattern Recognition 37(9), 1773–1784 (2004)

Srivastava, A., Lee, A.B., Simoncelli, E.P., Zhu, S.-C.: On advances in statistical modeling of natural images. Journal of Mathematical Imaging and Vision 18, 17–33 (2003)

Szummer, M., Picard, R.W.: Indoor-outdoor image classification. In: Proc. of IEEE International Workshop on Content-based Access of Image and Video Databases, CAIVD, pp. 42–51 (1998)

Tian, G., Fu, H., Feng, D.D.: Automatic medical image categorization and annotation using lbp and mpeg-7 edge histogram. In: Proc. Inter. Conf. On Inform. Tech. and Appl. in Biomedicine, pp. 51–53 (2008)

Torralba, A., Oliva, A.: Statistics of natural image categories. Network: Computation in Neural Systems 14, 391–412 (2003)

Vailaya, A., Jain, A., Zhang, H.J.: On image classification: City vs. landscape. In: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries, CBAIVL 1998, p. 3. IEEE Computer Society, Washington (1998)

Xu, F., Zhang, Y.: Integrated patch model: A generative model for image categorization based on feature selection. Pattern Recognition Letters 28, 1581–1591 (2007)

Yong, S.-P., Deng, J.D., Puvis, P.K.: Modelling semantic context for novelty detection in wildlife scenes. In: Proc. IEEE Inter. Conf. Multimedia and Expo (ICME 2010), pp. 1254–1259 (2010)

Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. Int. J. Comput. Vision 73(2), 213–238 (2007)

# Chapter 11
# Introduction to Curve and Edge Parametrization by Moments

Irina Popovici and Wm. Douglas Withers

**Abstract.** Curve parametrization is the task of determining the parameters of a general curve equation describing a structure in an image (or a surface in a higher-dimensional dataset). A common example is the widespread use of the Hough transform to determine parameters of straight lines in an image. Moment-based methods offer an attractive alternative to Hough-type methods for this task, especially as the number of parameters or the dimension of the space increases. Moment-based methods require no large accumulator array, are computationally efficient, and robust with respect to pixelization and high-frequency noise. This paper presents an overview of the state of the art for moment-based curve parametrization techniques. We discuss both abstract mathematical results guaranteeing the existence of a unique curve corresponding to a given set of moment values and allowing determination of parameter values for specialized quadrature-domain boundary curves, along with broad practical reconstructive results for a wide class of curves and hypersurfaces with arbitrarily many parameters and in arbitrarily many dimensions. Examples show the methods applied to analytically-defined image functions, generated images, and real-world images.

**Keywords:** Edge location, subpixel edge location, edge parametrization, circle detection, circle location, conic parametrization, curve parametrization, moments.

## 1 Background

Many science and engineering applications rely on analysis of images or higher-dimensional data, and on identifying certain structures within the data

Irina Popovici · Wm. Douglas Withers
Department of Mathematics
United States Naval Academy
e-mail: popovici@usna.edu,wdw@usna.edu

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 245–274.
springerlink.com

in particular. A common image-analysis task is edge description, whether as a set of pixels or parametrically by a curve equation, for further use in quality enhancement, shape or character detection, image restoration, calibration, or image coding. Higher-dimensional applications include volumetric 3-D imaging, where filaments and tubes are important, and statistical data analysis where the data cloud concentrates near special low-dimensional subsets. Real-world data applications demand computational techniques that are noise-resistant, fast, preferably parallelizable, and of reasonable memory requirement. Moment-based methods have long been recognized as befitting these problems, with an extensive literature on the theoretical and computational aspects of their use. The purpose of this chapter is two-fold: to present techniques (new and old) that perform well in the aforementioned contexts, and to present answers to some more theoretical lines of inquiry, with the goal of bridging the gap between engineers' and mathematicians' bodies of work.

Historically, edge detection in images came to prominence in the 1970's, focused on finding object boundaries as pixel subsets of the original picture ([8]). Although such methods seemed to perform better on near-vertical or near-horizontal edges, they made no special assumptions about the class of shapes applied to. At about the same time parametrization methods were developed for the special classes of straight, and then circular, edges, using the Hough transform ([2]) and its generalizations. Hough-type transforms use accumulator arrays, two-dimensional for straight lines, whose size increase exponentially with the number of parameters considered. This issue does not arise with the moment-based methods described in Section 3, which entail modest memory requirements essentially independent of the number of parameters or the dimensionality of the space.

An additional, more fundamental, distinction between moment-based and Hough-type methods is philosophical in nature. Consider the problem of identifying a circle or circles in the image shown in Fig. 1a. One possible answer is to locate the "trees": the individual small circles, as shown in Fig. 1b. But another possible answer is to locate the single-circle "forest", as shown in Fig. 1c. Which answer is preferable depends on the situation. But the single-solution "forest" approach offers the possibility of solution by direct computation, as opposed to accumulating and searching, with associated economy of computational resources. Hough-transform-type methods are better suited for parametrizing individual "trees"; the moment approach yields a single directly-calculated solution representing the "forest."

The moment-based approach to edge parametrization was pioneered in the 1980s. In this approach, a region assumed to contain an image structure is integrated against several functions of position $(x, y)$, yielding moment values from which various edge parameters are calculated. Early work (Machuca and Gilbert [6], Reeves *et al.* [7], and Lyvers *et al.* [9]) used moments of power functions $x^\alpha y^\beta$ on a circular mask to derive parameters for a straight edge, namely the edge angle and the distance $l$ from the center of the mask to the

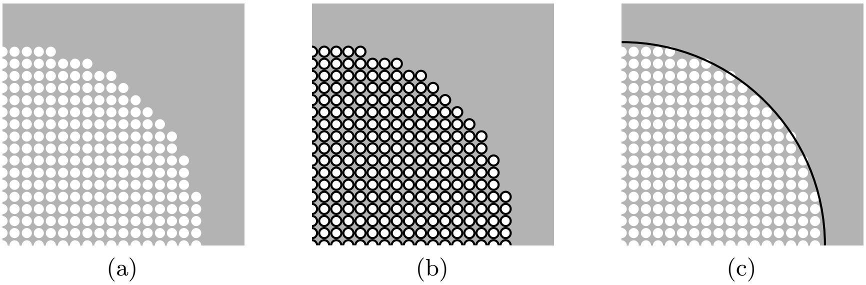


**Fig. 1** (a) Raw image. (b) A multiple-circle solution, such as might be produced by a generalized Hough transform. (c) A single-circle solution, such as might be produced by a moment-based method.

edge. The orientation angle is computed based on first-degree moments alone; the authors employ this simplicity to show that noise alone induces no bias in the angle computation, although other factors, such as pixelization and quantization, may do so. The location $l$ is computed via rotated moments of degree not exceeding two; empirical analysis of the impact of pixelization shows a small bias in this computation. Correcting for that bias brings the accuracy of the location calculation to better than 0.45 %.

Ghosal and Mehrota ([11], [15]) modify the method in [7] by working with Zernike moments, complex-valued orthogonal moment functions, also with circular support. They also examine roof edges which are marked by a discontinuity in the slope of the brightness function rather than the brightness itself (see also [22]). Also used for image analysis were Legendre polynomials ([14]), and discrete Chebyshev polynomials ([19]).

The newer method presented in Section 3 generates "custom-built" moment functions supported on any desired shape—rectangles are particularly useful in imaging applcations. Tailoring the mask to the region to be represented enhances both accuracy and efficiency. For example, for the straight-edge parametrization, a square mask ([21]) entails 4 moment calculation operations per pixel, whereas the circular mask entails $2\pi$ moment calculations per pixel in the method of Zernicke moments, or $3\pi$ moment calculations per pixel in the method of power moments (which uses six moment values rather than four).

Additionally, this flexibility makes the method adaptable to structure imposed on the image data, either *a priori* (such as JPEG-format images, which are coded in terms of cosine functions on $8 \times 8$ blocks), or subsequently (such as a wedgelet representation on a recursively refined square ([17], [20]), triangular, or other grid).

The problem of locating curved edges was motivated by the prevalence of certain shapes (circles, ellipses, etc) in nature and in man-made objects. Most studies in the literature focus on the parametrization of conic curves, although other shapes—such as airplane silhouettes—have been considered.

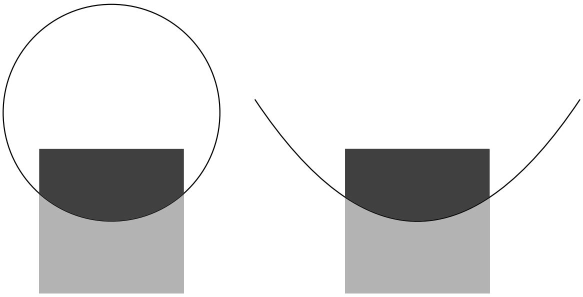


**Fig. 2** A circle and a parabola may be difficult to differentiate based on local information.

Edge description in terms of geometric or algebraic parameters rather than as a set of pixels is important in applications such as biometrics, calibration, image registration, and multi-scale geometric image coding methods such as wedgelets and curvelets.

With the exception of the theoretical results related to the reconstruction of quadrature domains (see the next section), most methods described in the literature up to [24] employ the special characteristics of a single class of curves—circles (Xu *et al.* [10]), ellipses (Heikkilä [16], Yoo and Sethi [12]), or parabolas (Jafri and Deravi [13])—and are accordingly limited to a single curve class. For many applications (particularly those involving oblique views, re-scaling and localization, or occluded curves), this *a priori* restriction is unwarranted (Fig. 2). The algebra-based methods presented in Section 3, an extension of [24], are not limited to specific geometric curve classes.

Related to this issue, another feature of the methods presented in Section 3 is the ability to trade off simplicity of representation versus goodness of fit (for example, choose a straight line segment over a low-curvature contour, or a circle over a small-eccentricity ellipse), using techniques illustrated in [24].

## 2 Mathematical Theory of Solvability

This section reviews the abstract mathematical work pertaining to the question of whether determining the equation of a curve in the plane (or hypersurface in $\mathbf{R}^K$) from moment values is solvable at all, and how many moment values are required to guarantee uniqueness of the solution. This section presents nonconstructive existence results, in contrast to the practical constructive-oriented approach of the next section.

The mathematical problem of recovering information about an object from indirect measurements can be traced back to Markov and Chebyshev, who were interested in finding approximations for integrals of various classes of functions $f$ with respect to a distribution $d\sigma$ based on the evaluation of a number of moment values for the distribution.

This one-dimensional moment problem, nowadays known as the $L$-problem, has long been solved. Uniqueness and existence results can be found in Ahiezer and Kreĭn's comprehensive book [3].

Interest in the multivariable analogue of this problem is more recent, heightened by its relevance to areas such as image processing, geophysics, and tomography. Frequently, the object or shape to be identified and located in these applications has near-constant brightness, and the problem of describing it essentially reduces to finding a "shape" function $G$ whose support coincides with the location of the object. If $\Gamma$ denotes the set of points belonging to the object, then the object boundary $\partial\Gamma$ consists of points $\vec{x}$ satisfying $G(\vec{x}) = 0$.

The mathematical foundation for the reconstruction problem, formulated as finding the brightness function $I(\vec{x})$ or the shape function $G$ using monomials as moment functions, can be found in [5]. We use standard multi-index notation: if $\alpha = (\alpha_1, \alpha_2, \dots \alpha_K)$ is a multi-index with $\alpha_i$ non-negative integers, then the degree of $\alpha$, denoted $|\alpha|$ is defined as $|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_K$.

Herewith are some theoretical results ensuring that the recovery of the shape function is a well posed problem:

**Theorem 1.** *Let $S$ be a closed cube in $\mathbf{R}^K$. Given a fixed degree $d$ and fixed constants $\{u_\alpha : \ |\alpha| \le d\}$, then Markov's problem:*

$$\int_S I(\vec{x})\, x_1^{\alpha_1} x_2^{\alpha_2} \dots x_K^{\alpha_K}\, dV = u_\alpha \quad \textit{for } |\alpha| \le d$$

*admits a measurable function $I : S \to [-1, 1]$ as a solution if and only if for all polynomials $P$ of degree not exceeding $d$:*

$$P(\vec{x}) = \sum_{|\alpha| \le d} c_\alpha\, x_1^{\alpha_1} x_2^{\alpha_2} \dots x_K^{\alpha_K}$$

*the inequality*

$$\sum_{|\alpha| \le d} c_\alpha u_\alpha \le \int_S |P(\vec{x})|\, dV(\vec{x})$$

*holds. Moreover, Markov's problem has unique solution if and only if there exists a polynomial $G$ of degree not exceeding $d$ such that*

$$u_\alpha = \int_S I_G(\vec{x})\, x_1^{\alpha_1} x_2^{\alpha_2} \dots x_K^{\alpha_K}\, dV,$$

*where*

$$I_G(\vec{x}) = \begin{cases} 1 & \textit{if } G(\vec{x}) \ge 0 \\ -1 & \textit{otherwise.} \end{cases}$$

Note that this result guarantees the existence of an appropriate shape function but provides no reconstructive algorithm for such. For the two-dimensional problem, reconstruction algorithms have been developed for the case where $\Gamma$ is a quadrature domain satisfying restricted incidence relations with $S$. As complex analytic functions play an essential role in these algorithms, our description uses complex representations for functions and moments. Any function of $(x, y)$ is expressible as a function of variables $z = x+iy$ and $\overline{z} = x - iy$. For example monomials can be represented as:

$$x^m y^n = \frac{(z+\overline{z})^m (z-\overline{z})^n}{2^m (2i)^n}.$$

We use the complex representation $I(z, \overline{z})$ of the real-valued brightness function $I$, along with its complex-valued moments

$$u_{mn}(I) = \iint_S I(z, \overline{z})\, z^n \overline{z}^m \, dA(z).$$

A brief overview of the method ([18]) follows.

Let $I$ be a real-valued function whose support is contained in a fixed ball $S$ in the complex plane $\mathbf{C}$. Assume $I$ to be integrable with respect to area on $S$, and that $0 \leq I \leq 1$ everywhere. Fix a positive integer $d$ and denote by $\vec{u}_d(I)$ the $(d+1)^2$-dimensional complex vector whose entries are the complex monomial moments of $I$:

$$\vec{u}_d(I) = (u_{mn}(I) : m, n = 0, \ldots, d).$$

Denote by $\Sigma_d$ the set of all possible moments of such functions:

$$\Sigma_d = \{\vec{u}_d(I) : 0 \leq I \leq 1 \text{ on } S\}.$$

It can then be shown that the admissible set $\Sigma_d$ is closed, bounded, and convex and that the following hold:

**Theorem 2.** *An arbitrary vector $\vec{v} \in \Sigma_d$ has a unique representing function $I$ such that $\vec{v} = \vec{u}_d(I)$ if and only if $\vec{v}$ is an extremal point of $\Sigma_d$.*

**Theorem 3.** *A point $\vec{v} = (v_{mn}) \in \Sigma_d$ is extremal if and only if there exists a real-valued polynomial $G(z, \overline{z})$, of degree not exceeding $d$ in each of $z$ and $\overline{z}$, satisfying $\{z : G(z, \overline{z}) \geq 0\} \subset S$ and*

$$v_{mn} = \iint_{\{z : G(z,\overline{z}) \geq 0\}} z^n \overline{z}^m dA(z)$$

*for all $0 \leq n, m \leq d$.*

In other words, a real-valued function $I$ bounded by 0 and 1 is uniquely determined by its complex moments of order up to $d$ if and only if a real-valued polynomial $G$ exists such that $\{z : G(z, \overline{z}) \geq 0\}$ is fully contained in the set $S$ on which the moments are computed, and $I$ takes the form:

$$I(z) = \begin{cases} 1 \text{ if } G(z, \overline{z}) \geq 0 \\ 0 \text{ otherwise.} \end{cases}$$

Although $I$ is uniquely determined in this case no algorithm is known for reconstructing $G$ from $\vec{u}(I)$, except in the limited case when the set where $G \geq 0$ is a quadrature domain fully contained in the ball $S$.

A *quadrature domain* $\Omega$ is a bounded, open, connected set in $\mathbf{C}$ with the property that there exist points $\gamma_1, \ldots, \gamma_m$ in $\Omega$ (called *nodes*), integer multiplicity values $\mu_1, \ldots, \mu_m$, and complex numbers $\alpha_{k,j}$ such that

$$\iint_{\Omega} f(z)\, dA(z) = \sum_{k=1}^{m} \sum_{j=0}^{\mu_k - 1} \alpha_{k,j} f^{(j)}(\gamma_k)$$

for all complex analytic functions $f$ integrable on $\Omega$. The *order* $d$ of a quadrature domain is defined to be the sum of the multiplicities of its nodes: $d = \mu_1 + \mu_2 + \ldots + \mu_m$.

A disk $\Omega$ of center $c$ and radius $r$ is the simplest quadrature domain and the only order-one quadrature domain. It satisfies the quadrature equation

$$\iint_{\Omega} f(z)\, dA(z) = \pi R^2 f(c),$$

also known as the Mean-Value Theorem for harmonic functions. Other simple quadrature domains include *double-node* domains, which are the images of the unit disk under conformal maps such as $z \rightarrow z^2 + bz$, where $b \geq 2$, and polygons. For a polygon $\Omega$ with vertices $\gamma_1 \ldots \gamma_n$ counted cyclicly counterclockwise, the quadrature identity ([1], [4]) states:

$$\iint_{\Omega} f''(z)\, dA(z) = \sum_{j=0}^{n} \alpha_k f(\gamma_k)$$

where

$$\alpha_k = \sin(\phi_{k-1} - \phi_k) e^{-i(\phi_{k-1} + \phi_k)}$$

and $\phi_k$ is the angle made by the side $\gamma_k \gamma_{k+1}$ with the positive real axis.

Quadrature domains are a very small subclass of those domains whose boundaries are defined as zero-sets of real-valued polynomials; for example the class of ellipses, frequently encountered in applications, is *not* a quadrature domain.

The reconstruction algorithms for a quadrature domain $\Omega$ of order $d$, as presented in [18], requires the following transform: given moment values $v_{mn}$ define the *exponential transform* to be the formal power series:

$$\exp\left[-\frac{1}{\pi} \sum_{m,n=0}^{d} \frac{v_{mn}}{z^{n+1}\overline{w}^{m+1}}\right] = 1 - \sum_{m,n=0}^{\infty} \frac{V_{mn}}{z^{n+1}\overline{w}^{m+1}} \tag{1}$$

For quadrature domains the matrix $V = (V_{mn})_{m,n=0}^{d}$ is singular, thus it admits a complex eigenvector $\vec{c} = (c_0, \dots c_d)$, satisfying $V\vec{c} = 0$. Assuming $d$ to be the minimal order of $\Omega$, normalize $\vec{c}$ so that $c_d = 1$. Define the polynomial $F(z) = z^d + c_{d-1}z^{d-1} + \dots c_0$ and expand the product

$$F(z)\overline{F(\overline{w})}\left(1 - \sum_{m,n=0}^{d} \frac{V_{mn}}{z^{n+1}\overline{w}^{m+1}}\right) = G(z, \overline{w}) + O(1/z, 1/\overline{w}). \qquad (2)$$

where $G$ is a polynomial in $z$ and $\overline{w}$. The domain $\Omega$ is then given by

$$\Omega = \{z : G(z, \overline{z}) < 0\}.$$

To illustrate this method, consider the simple case where $\Omega$ is a disk of center $c$ and radius $r$ entirely contained in the disk $S$. The brightness function $I$, whose moments up to degree one are needed, becomes:

$$I(z) = \begin{cases} 1 \text{ if } |z - c| \leq r \\ 0 \text{ otherwise.} \end{cases}$$

Start with:

$$\begin{aligned}
v_{00}{=}u_{00}(I) &= \iint_S I(z)\,dA(z) = \iint_\Omega 1\,dA(z) = \pi r^2, \\
v_{01}{=}u_{01}(I) &= \iint_S zI(z)\,dA(z) = \pi r^2 c, \\
v_{10}{=}u_{10}(I) &= \iint_S \overline{z}I(z)\,dA(z) = \pi r^2 \overline{c}, \\
v_{11}{=}u_{11}(I) &= \iint_S z\overline{z}I(z)\,dA(z) = \frac{\pi r^4}{2} + \pi r^2 c\overline{c}.
\end{aligned}$$

The exponential expansion (1) yields:

$$V_{00} = \frac{v_{00}}{\pi},\ V_{01} = \frac{v_{01}}{\pi},\ V_{10} = \frac{v_{01}}{\pi},\ V_{11} = \frac{v_{11}}{\pi} - \frac{v_{00}^2}{2\pi^2}.$$

Substituting in the specific moment values $v_{mn}$ for the disk $\Omega$, and constructing the $2 \times 2$ matrix, we find:

$$V = \begin{pmatrix} r^2 & r^2 c \\ r^2\overline{c} & r^2 c\overline{c} \end{pmatrix}.$$

This matrix is singular, with has a one-dimensional eigenspace with eigenvector

$$\vec{c} = \begin{pmatrix} -c \\ 1 \end{pmatrix}.$$

The expansion in (2) becomes:

$$(z-c)(\overline{w}-\overline{c})\left(1-\frac{r^2}{z\overline{w}}-\frac{r^2c}{z^2\overline{w}}-\frac{r^2\overline{c}}{z\overline{w}^2}-\dots\right)=$$

$$(z-c)(\overline{w}-\overline{c})-r^2+O\left(\frac{1}{z},\frac{1}{\overline{w}}\right)=G(z,\overline{w})+O\left(\frac{1}{z},\frac{1}{\overline{w}}\right).$$

We therefore recover the equation of the boundary circle of $\Omega$ as:

$$G(z,\overline{z})=(z-c)(\overline{z}-\overline{c})-r^2=0.$$

This equation has degree one in each of $z,\overline{z}$, consistent with the disk being a quadrature domain with $d=1$.

Note that the complete containment of $\Omega$ within $S$ is critical to the success of this construction. If $\Omega$ is only partially contained within $S$, then the matrix $V$ fails to be singular.

The following section describes our methods, which provide for the reconstruction of the functions $I$ and $G$ in cases where the domain partially overlaps $S$, and $G$ need not be a polynomial function, while the number of moments used may exceed $d$ in cases where $G$ is a polynomial.

## 3 The Constructive Moment-Analysis Method

*General framework:* We consider a two-valued function $I(\vec{x})$, $\vec{x}=(x_1,\dots,x_K)$:

$$I(\vec{x})=\begin{cases} q \text{ if } \vec{x}\in\Gamma \\ r \text{ otherwise.}\end{cases} \tag{3}$$

Here $\Gamma$ is a subset of $K$-dimensional space whose boundary $\partial\Gamma$ is a finite collection of piecewise-smooth hypersurfaces. For example, if $K=2$ then $I$ could represent an image consisting of two solid-color regions; if $K=3$ then $I$ might represent a volumetric data set consisting of two homogeneous regions. Let $S$ be a compact subset of the $K$-dimensional dataset bounded by a finite collection of simple closed piecewise-smooth $(K-1)$-dimensional hypersurfaces.

Fig. 3 shows a practical example of the case $K=2$, an image. Although the image as a whole manifests many different values of $I(\vec{x})$ (gray-levels), $I$ is approximately two-valued on a certain subrectangle $S$.

We further suppose all points $\vec{x}\in S\cap\partial\Gamma$ satisfy $G(\vec{x})=0$, where $G$ is expressible as an unknown linear combination of known functions $z_n(\vec{x})$:

$$G(\vec{x})=\sum_{n=1}^{N}p_n z_n(\vec{x}), \tag{4}$$

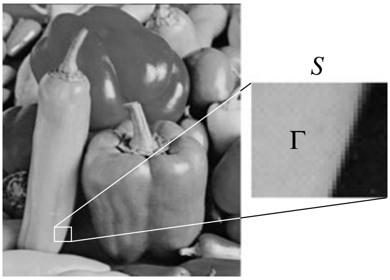


**Fig. 3** In a neighborhood $S$ of a sharp edge, a multi-value image is dominated by just two values.

with at least one nonzero $p_n$. Since the curve described by (4) is invariant under multiplication of the coefficients $p_n$ by a nonzero constant, the class of curves described by (4) has $(N-1)$ parameters.

For example, one such class is that defining the general equation of a two-dimensional conic ($K = 2$, $N = 6$):

$$G(\vec{x}) = p_1 + p_2x_1 + p_3x_2 + p_4x_1^2 + p_5x_1x_2 + p_6x_2^2 = 0. \tag{5}$$

Another is the general equation of a sphere ($K = 3, N = 5$):

$$G(\vec{x}) = p_1 + p_2x_1 + p_3x_2 + p_4x_3 + p_5(x_1^2 + x_2^2 + x_3^2) = 0.$$

And another is a general hyperplane in $K$ dimensions ($N = K + 1$):

$$G(\vec{x}) = p_1x_1 + p_2x_2 + \cdots + p_Kx_K + p_{K+1} = 0. \tag{6}$$

For an integrable function $U(\vec{x})$ defined on $S$, we define the moment value with respect to $I$:

$$\langle U, I\rangle = \int_S I(\vec{x})\, U(\vec{x})\, dV,$$

where $dV$ denotes the $K$-dimensional volume element.

## 4 Moment-Based Line Parametrization

As an example of the type of result provable in this situation, we present the following method for recovering the equation of a line, relatively modest in scope and yet with a wide variety of potential uses.

**Theorem 4.** *Let $K = 2$ and let $\Gamma$ be a subset of the plane bounded by a straight line, with equation*

$$p_1x + p_2y + p_3 = 0,$$

*(writing $x$ for $x_1$ and $y$ for $x_2$). Let $V$ be an open set containing $S$. Choose a function $F(x, y)$ continuously differentiable on $V$ such that $F(x, y) = 0$ for $(x, y) \notin S$. Define*

$$U_1 = F_x,\ U_2 = F_y,\ U_3 = -2F - xF_x - yF_y,$$

*the subscripts denoting partial differentiation. For $n = 1, 2, 3$, let*

$$u_n = \langle U_n, I\rangle.$$

*Then all $(x, y) \in \partial\Gamma$ satisfy:*

$$u_1 x + u_2 y + u_3 = 0. \tag{7}$$

We omit the proof of this theorem as it is a special case of both Theorem 8 and Theorem 9 presented below.

As an example of the application of this theorem, consider an image function $I(x, y)$ defined by:

$$I(x, y) = \begin{cases} 9 \text{ if } (x, y) \in \Gamma, \\ 4 \text{ otherwise,} \end{cases}$$

where $\Gamma$ is the part of the plane $5x + 3y \leq 4$. We take $S$ to be the square $0 \leq x, y \leq 1$. Our goal is to recover the equation of the boundary line $\partial\Gamma$ by analyzing the values of $I$ within $S$.

We choose $F(x, y) = \max(0, xy[1-x][1-y])$, so that $F(x, y) = 0$ outside $S$. (Note that $F$ does not strictly satisfy the twice-differentiable hypothesis, but this condition can be relaxed somewhat.) In practice the shape of $S$ heavily influences the choice of $F$. The three moment functions are therefore:

$$\begin{aligned} U_1(x, y) &= F_x(x, y) = (2x - 1)(y^2 - y), \\ U_2(x, y) &= F_y(x, y) = (2y - 1)(x^2 - x), \\ U_3(x, y) &= -2F(x, y) - xF_x(x, y) - yF_y(x, y) \\ &= -xy(6xy - 5y - 5x + 4). \end{aligned}$$

We can then calculate $\langle U_1, I\rangle$:

$$\int_0^1 \left( \int_0^{(4-3y)/5} 9(2x - 1)(y^2 - y)\, dx + \int_{(4-3y)/5}^1 4(2x - 1)(y^2 - y)\, dx \right) dy$$
$$= 5 \cdot \frac{29}{750}.$$

Similarly:

$$\langle U_2, I\rangle = 3 \cdot \frac{29}{750}, \langle U_3, I\rangle = -4 \cdot \frac{29}{750},$$

so that (7) is equivalent to the equation $5x + 3y = 4$ of $\partial\Gamma$.

One practical application of this technique is for wedgelet or similar image coding methods which employ successive refinement of local straight-edge models. Theorem 4 defines a predominant edge by straightforward computation, in contrast to searching a dictionary of possible edge positions. Fig. 4 shows the effect of decomposing an image into subblocks $S$ of various sizes and using moments to determine an edge location for each. Each block is fitted with a straight edge, even if the original block content was more complex in structure.

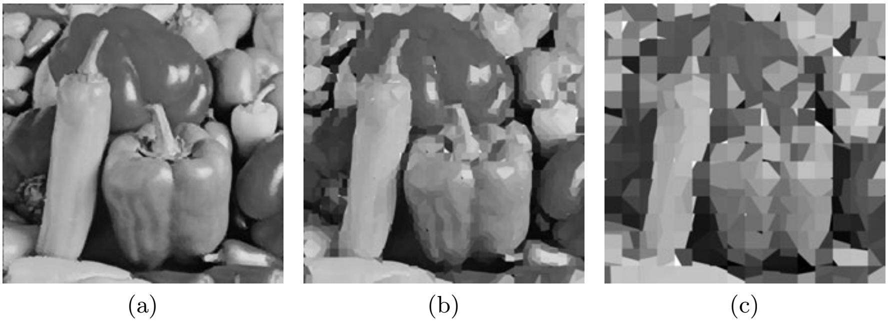


**Fig. 4** (a) Approximation to *Peppers* image based on subdivision into $8 \times 8$ blocks and representing each block by two values separated by a straight-line edge. (b) Similar approximation with $16 \times 16$ blocks. (c) Similar approximation with $32 \times 32$ blocks.

Alternatively, we could take $F(x, y) = \sin(\pi x)\sin(\pi y)$ within $S$ and $F \equiv 0$ outside $S$. The three moment functions are then:

$$\begin{aligned}
U_1(x, y) &= F_x(x, y) = \pi\cos(\pi x)\sin(\pi y),\\
U_2(x, y) &= F_y(x, y) = \pi\sin(\pi x)\cos(\pi y),\\
U_3(x, y) &= -2F(x, y) - xF_x(x, y) - yF_y(x, y)\\
&= -\pi y\sin(\pi x)\cos(\pi y) - \pi x\cos(\pi x)\sin(\pi y) - 2\sin(\pi x)\sin(\pi y).
\end{aligned}$$

We can then calculate:

$$\langle U_1, I\rangle = 5\lambda, \quad \langle U_2, I\rangle = 3\lambda, \quad \langle U_3, I\rangle = -4\lambda,$$

where

$$\lambda = \frac{25\sqrt{10 - 2\sqrt{5}}}{32\pi} \doteq 0.58468,$$

so that once again (7) recovers the equation $5x + 3y = 4$ of $\partial\Gamma$. Note that Theorem 4 provides options. For example, this second alternative is more useful for edge location in a DCT-coded image (JPEG) in transform space—as the moment calculation is equally simple in transform space. This capability is useful, for example, in controlled filtering to suppress compression artifacts while preserving sharp edges ([23]).

The JPEG image format partitions the image into blocks of $8 \times 8$ pixels; for our purposes we consider such a block to correspond to the $[0,1] \times [0,1]$ square here discussed, the image values $I(x,y)$ on this square represented by an $8 \times 8$ grid of pixel values $i_{\eta\xi}$, for $\xi, \eta = 0, \ldots 7$. JPEG specifies these pixel values not directly but rather in terms of their DCT coefficients $C_{n,m}$, for $m, n = 0, \ldots 7$ :

$$i_{\eta\xi} = \frac{1}{4} \sum_{m=0}^{7} \sum_{n=0}^{7} k_m \, k_n \, C_{nm} \cos \frac{(2m+1)\pi m}{16} \cos \frac{(2n+1)\pi n}{16} \qquad (8)$$

where $k_0 = 1/\sqrt{2}$ and $k_m = 1$ for $m = 1 \ldots 7$.

Calculating $u_1$, $u_2$, and $u_3$ first requires an approximation of $u_1 = \langle U_1, I \rangle$ by a Riemann sum based on the discrete set of pixel values $i_{\eta\xi}$:

$$u_1 \approx \sum_{\xi=0}^{7} \sum_{\eta=0}^{7} \int_{\eta/8}^{(\eta+1)/8} \int_{\xi/8}^{(\xi+1)/8} i_{\eta\xi} \, \pi \cos(\pi x) \sin(\pi y) \, dx \, dy,$$

each individual integral over the region subtended by a single pixel. Integrating, we obtain:

$$u_1 \approx \sum_{\xi=0}^{7} \sum_{\eta=0}^{7} \frac{i_{\eta\xi}}{\pi} \left[ \sin \frac{\pi(\xi+1)}{8} - \sin \frac{\pi\xi}{8} \right] \left[ \cos \frac{\pi(\eta+1)}{8} - \cos \frac{\pi\eta}{8} \right].$$

Substituting (8) for $i_{\xi\eta}$ yields:

$$u_1 \approx \frac{1}{4\pi} \sum_{m=0}^{7} \sum_{n=0}^{7} \left( k_m \sum_{\xi=0}^{7} \cos \frac{(2m+1)\pi m}{16} \left[ \sin \frac{\pi(\xi+1)}{8} - \sin \frac{\pi\xi}{8} \right] \right) \times$$
$$\left( k_n \sum_{\eta=0}^{7} \cos \frac{(2n+1)\pi n}{16} \left[ \cos \frac{\pi(\eta+1)}{8} - \cos \frac{\pi\eta}{8} \right] \right) C_{nm}.$$

The inside sums above are pre-computable for each given $m, n$, and zero-valued for all even values of $m$ and for $n \neq 1$. We therefore find:

$$u_1 = 0.17564 \, C_{1,1} - 0.0806 \, C_{1,3} - 0.01447 \, C_{1,5} - 0.00444 \, C_{1,7}.$$

A similar argument reduces the computations for $u_2$ to

$$u_2 = 0.17564 \, C_{1,1} - 0.0806 \, C_{3,1} - 0.01447 \, C_{5,1} - 0.00444 \, C_{7,1}$$

and the integral needed for $u_3$ becomes

$$u_3 = -\frac{u_1 + u_2}{2} + \sum_{m=0}^{7} \sum_{n=0}^{7} \frac{1}{64} M_{nm} C_{nm} \tag{9}$$

where the matrix $M_{nm}$ is precalculable and has 15 nonzero entries, 9 distinct (Table 1).

**Table 1** Coefficient matrix $M_{nm}$ for calculating moment value $u_3$ directly from DCT coefficient values. (Omitted columns and rows contain no nonzero entries.)

| $n\backslash m$ | 0 | 2 | 4 | 6 |
|---|---|---|---|---|
| 0 | 0.000000 | 3.968077 | 0.570846 | 0.169225 |
| 2 | 3.968077 | −3.641138 | −0.588845 | −0.177868 |
| 4 | 0.570846 | −0.588845 | −0.094067 | −0.028361 |
| 6 | 0.169225 | −0.177868 | −0.028361 | −0.008549 |

For (7) to hold trivially by $u_1 = u_2 = u_3 = 0$ is of no use in recovering the line equation. As noted in the discussion of Theorem 9 below, this possibility can be eliminated by choosing $F$ to be positive on the interior of $\varGamma$—provided of course the line in question intersects the interior of $\varGamma$.

## 5 Parametrizing Higher-Order Curves and Surfaces

The next theorem, proved in [24], treats general two-dimensional boundary curves:

**Theorem 5.** *Let $K = 2$ and write $x$ for $x_1$, $y$ for $x_2$. Let $V$ be an open set containing $S$. Assume the functions $z_1, \ldots, z_N$ in (4) to be twice continuously differentiable on $V$. Choose functions $s(x, y)$ twice continuously differentiable on $V$ and $F(x, y)$ continuously differentiable on $V$ such that $F(x, y) = 0$ for $(x, y) \notin S$. For $n = 1, \ldots, N$, define $U_n(x, y) = \varUpsilon(F, s, z_n)$, where*

$$\varUpsilon(F, s, z) = \left([sz_y - zs_y]\, F\right)_x - \left([sz_x - zs_x]\, F\right)_y ,$$

*the $x$ and $y$ subscripts indicating partial differentiation. Define moment values $u_n = \langle U_n, I \rangle$ for $n = 1, \ldots, N$. Then*

$$\sum_{n=1}^{N} u_n p_n = 0. \tag{10}$$

*Example:* Let the class of curves under consideration be the general conic (5), with equation

$$p_1 + p_2 x + p_3 y + p_4 x^2 + p_5 y^2 + p_6 xy = 0,$$

so that

$$z_1 = 1,\ z_2 = x,\ z_3 = y,\ z_4 = x^2,\ z_5 = y^2,\ z_6 = xy.$$

Take $S$ to be the square $-1 \le x, y \le 1$. Choose

$$F(x,y) = \max\left(0, \left[1 - x^2\right]\left[1 - y^2\right]\right).$$

Choose $s(x,y) \equiv 1$. Then we have:

$$\begin{aligned}
U_1(x,y) &= \Upsilon(F,s,1) = 0,\\
U_2(x,y) &= \Upsilon(F,s,x) = 2y(1 - x^2),\\
U_3(x,y) &= \Upsilon(F,s,y) = -2x(1 - y^2),\\
U_4(x,y) &= \Upsilon(F,s,x^2) = 4xy(1 - x^2),\\
U_5(x,y) &= \Upsilon(F,s,y^2) = -4xy(1 - y^2),\\
U_6(x,y) &= \Upsilon(F,s,xy) = 2y^2 - 2x^2
\end{aligned}$$

for $\vec{x} \in S$; all $U_n$ are zero-valued for $\vec{x} \notin S$.

If we now further suppose that our image function $I(x,y)$ is zero-valued outside a circle of radius $3/2$ centered at $(-1/2, -1)$ and has value 1 inside the circle, we can calculate the corresponding moment values over $S$:

$$u_1 = 0,\ u_2 = -1.13136,\ u_3 = 0.561783,$$

$$u_4 = -0.088179,\ u_5 = 0.095968,\ u_6 = -0.083716.$$

It is easily verified that (10) holds for the circle equation

$$-1 + x + 2y + x^2 + y^2 = 0.$$

Note that the quadrature-domain construction of Section 2 fails for this circle, as it is only partially contained in the domain $S$ of integration.

As a second example, consider the general equation of a circle:

$$p_1 + p_2 x + p_3 y + p_4\left(x^2 + y^2\right) = 0.$$

Let $S$ be the disk $x^2 + y^2 \le 1$. Choose

$$F(x,y) = \max\left(0, 1 - x^2 - y^2\right).$$

Choose $s(x,y) \equiv x$. Then we have:

$$\begin{aligned}
U_1(x,y) &= \Upsilon(F,s,1) = -2y,\\
U_2(x,y) &= \Upsilon(F,s,x) = 0,\\
U_3(x,y) &= \Upsilon(F,s,y) = 2 - 4x^2 - 4y^2,\\
U_4(x,y) &= \Upsilon\left(F,s,x^2 + y^2\right) = 4y - 6x^2 - 6y^2.
\end{aligned}$$

Further supposing $I(x,y)$ to be zero-valued outside a circle of radius 1 centered at $(-2/5,-4/5)$ and having value 1 inside the circle, we can calculate the corresponding moment values over $S$:

$$\begin{aligned} u_1 &= 1.13144, \\ u_2 &= 0, \\ u_3 &= 0.29712, \\ u_4 &= -0.24911. \end{aligned}$$

It is easily verified that (10) holds for the circle equation

$$-1+4x+8y+5\left(x^2+y^2\right)=0.$$

Once again, the quadrature-domain construction of Section 2 fails for this circle, as it is only partially contained in the domain $S$ of integration.

*Solving for the parameter vector $\vec{p}$:* Of course no single equation of the form (10) suffices to determine $p_1,\ldots,p_N$ (unless $N=2$). However, by varying the choice of $s$ and/or $F$, one can generate as many such equations as desired. Together these form a linear system:

$$A\vec{p}=0 \tag{11}$$

to be solved for the vector $\vec{p}$ of parameter values $p_1,\ldots,p_N$, $A$ being an $M \times N$ coefficient matrix. Since the curve defined by $G(\vec{x})=0$ is invariant under multiplication of $\vec{p}$ by a nonzero scalar, the system (11) is degenerate in the sense that it has at least a one-dimensional space of solutions. The specific number of equations and choices of $F$ and $s$ are influenced by the particular class of curves under consideration. At a minimum $(N-1)$ equations are needed.

As a practical matter, in solving (11) for $\vec{p}$, we must deal with two opposing concerns: First, the matrix $A$ may be *hyperdegenerate* (having a null space of dimension greater than one)—or nearly so, leading to numerical instability in the solution. Second, due to round-off error, pixelization effects, or because the boundary curve imperfectly satisfies $G(\vec{x})=0$, the matrix $A$ may not be truly degenerate but only nearly so.

A standard approach to solving a linear system which provides an avenue to addressing both these concerns is to recast the problem as minimizing $||\,A\vec{p}\,||$ subject to the constraint $||\,\vec{p}\,||=1$. The solution is obtained by taking $\vec{p}$ to be an eigenvector of $A^TA$ ($A^T$ denoting the transpose of $A$) corresponding to the eigenvalue of $A^TA$ closest to zero (equalling zero if $A$ is indeed degenerate). Experiment shows that in general $A^TA$ does have an eigenvalue very close to zero (in comparison to the magnitude of entries of $A$), and that the corresponding eigenvector does yield quite a good fit to the given boundary curve. Nonexistence of a near-zero eigenvalue indicates that the boundary curve is poorly described by the equation $G(\vec{x})=0$.

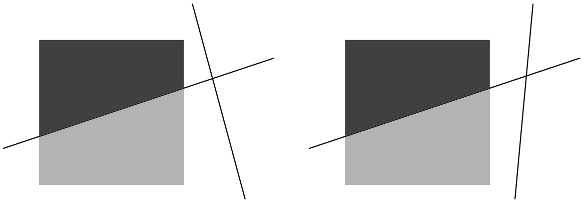
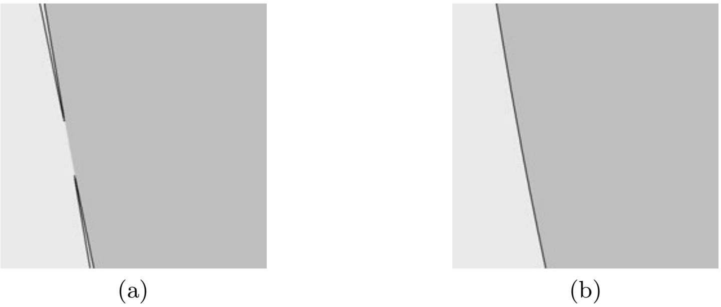


**Fig. 5** It is impossible to distinguish the quadratic equations representing sets of crossed lines (a) and (b) based on the image content in the set $S$ (the shaded region).

Hyperdegeneracy or near-hyperdegeneracy is indicated by the existence of two or more near-zero eigenvalues. For example, for the conic described by five independent parameters, hyperdegeneracy always occurs for a system of fewer than five equations. Robustness of the method can be enhanced by overdetermining the system, using more equations (and thus more rows of $A$) than the bare minimum necessary. For the conic case ([24]), we elected to use 18 equations rather than five. The Appendix provides a comprehensive list of the resulting moment functions.

Hyperdegeneracy is geometrically inevitable in situations where the domain $S$ of integration simply does not contain sufficient information to uniquely determine the image parameters. An obvious case is when the boundary curve $\partial\Gamma$ fails to intersect $S$; a less trivial example is shown in Fig. 5. Practical consequences of geometric hyperdegeneracy are shown in Fig. 6a. The eigenvalue problem used to find $\vec{p}$ can be perturbed, however, to drive the solution toward the geometrically simplest among multiple possibilities, as shown in Fig. 6b. Details of this technique are given in [24].

We follow with a few examples showing the robustness of the moment-based approach in the face of certain types of image degradation. Figs. 7-9 show edges located using Theorem 5, in images characterized by blurring, coarse pixelization, and probabilistically defined edges. In all these cases, the method was applied with no filtering, sharpening, edge detection, or any



(a) (b)

**Fig. 6** (a) Result of fitting a nearly-straight boundary curve with a quadratic equation. (b) Result of applying perturbed technique to the same curve.

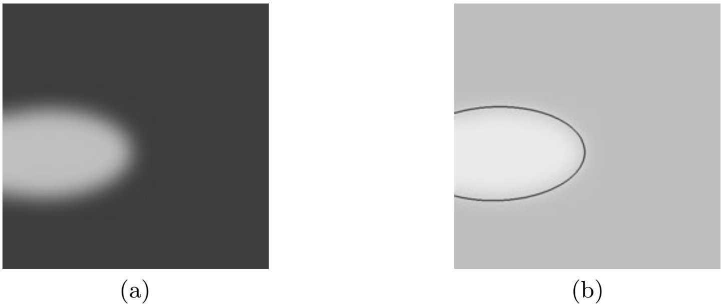

(a) (b)

**Fig. 7** (a) Image containing a blurred edge. (b) Boundary curve parametrized by moments.

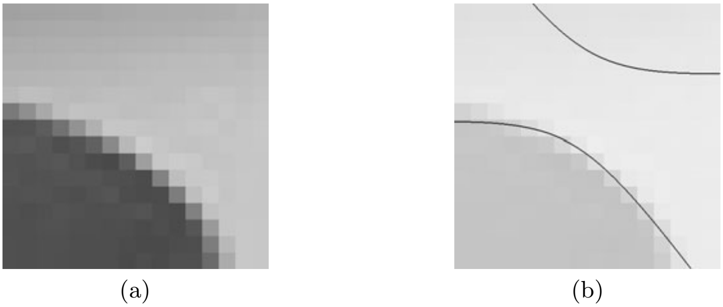

(a) (b)

**Fig. 8** (a) Image containing a highly pixellated edge. (b) Boundary curve parametrized by moments.

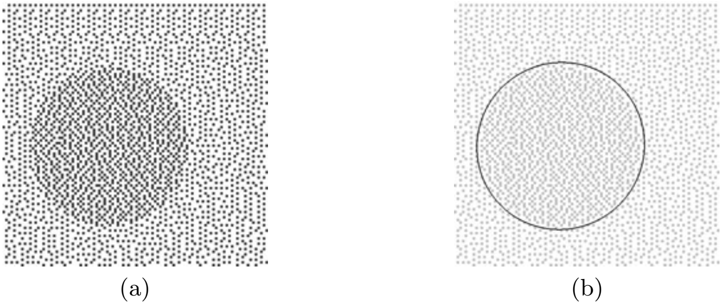

(a) (b)

**Fig. 9** (a) Image containing an edge marked only by change in probability distribution. (b) Boundary curve parametrized by moments.

preprocessing whatsoever. Figure 8 illustrates another characteristic of Theorem 5: there is no requirement that the entire graph of the equation to be recovered be manifested—only that the boundary curve be a subset of the graph of the equation.

Figure 10 shows the possible application of our method to the problem of locating the pupil in a red-eye-removal application. In this case simple preprocessing (paint a pixel white if its red component value exceeds its green component value by 77 or more, gray otherwise) was used to isolate the red-flare region. Subsequent application of our method serves to locate the pupil with subpixel accuracy.

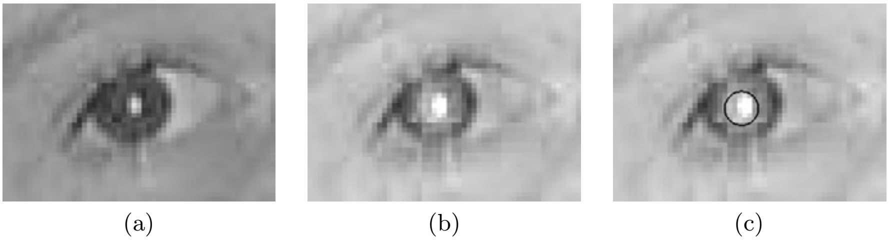


**Fig. 10** Locating pupil in a potential red-eye removal application: (a) grayscale representation of the (originally color) image; (b) red component of the image, showing flare in the pupil; (c) pupil located to sub-pixel accuracy.

The preceding theorem is generalizable with little modification to the general case of $K$ dimensions:

**Theorem 6.** *Let $V$ be an open set containing $S$. Assume $z_1, \ldots, z_N$ in (4) to be twice continuously differentiable on $V$. Choose functions $s(\vec{x})$ twice continuously differentiable on $V$ and $F(\vec{x})$ continuously differentiable on $V$ such that $F(\vec{x}) = 0$ for $\vec{x} \notin S$. Choose two coordinates $x_i, x_j$, $i \neq j$, and write $x$ for $x_i$, $y$ for $x_j$. For $n = 1, \ldots, N$, define $U_n(\vec{x}) = \varUpsilon(F, s, z_n)$, where*

$$\varUpsilon(F, s, z) = \left([sz_y - zs_y]\, F\right)_x - \left([sz_x - zs_x]\, F\right)_y ,$$

*the $x$ and $y$ subscripts indicating partial differentiation. Define moment values $u_n = \langle U_n, I \rangle$ for $n = 1, \ldots, N$. Then*

$$\sum_{n=1}^{N} u_n p_n = 0. \tag{12}$$

*Proof:* The proof of Theorem 5 as presented in [24] applies also in this case, except that Green's Theorem is unavailable in more than two dimensions. However, we can employ a variation of the same argument as follows. Once again assuming $q = 1$, $r = 0$, and $s(x, y) \neq 0$, we find by the Divergence Theorem:

$$u_n = \int_S I(\vec{x})\, U_n(\vec{x})\, dV = \int_{S \cap \varGamma} U_n(\vec{x})\, dV$$

$$
\begin{aligned}
&= \int_{S\cap\Gamma} \left\{ ([sz_{n,y} - z_n s_y]\, F)_x - ([sz_{n,x} - z_n s_x]\, F)_y \right\} dV. \\
&= \int_{\partial(S\cap\Gamma)} \left\{ ([sz_{n,y} - z_n s_y]\, F)\, \vec{\imath} - ([sz_{n,x} - z_n s_x]\, F)\, \vec{\jmath} \right\} \cdot \vec{n}\, dA \\
&= \int_{\partial(S\cap\Gamma)} F(x,y) \left( [sz_{n,y} - z_n s_y]\, \vec{\imath} - [sz_{n,x} - z_n s_x]\, \vec{\jmath} \right) \cdot \vec{n}\, dA,
\end{aligned}
$$

where $\vec{\imath}$ and $\vec{\jmath}$ denote the unit basis vectors in the $x$- and $y$-directions, respectively, and $\vec{n}$ and $dA$ are the unit normal vector and area element for $\partial\Gamma$, respectively. From (4), we then further have:

$$
\begin{aligned}
\sum_{n=1}^{N} p_n u_n &= \int_{\partial(S\cap\Gamma)} F(x,y) \left( [sG_y - Gs_y]\, \vec{\imath} - [sG_x - Gs_x]\, \vec{\jmath} \right) \cdot \vec{n}\, dA \\
&= \int_{\partial(S\cap\Gamma)} F(x,y) s^2 \left( [G/s]_y \vec{\imath} - [G/s]_x \vec{\jmath} \right) \cdot \vec{n}\, dA.
\end{aligned}
$$

The integrand is zero along all parts of $\partial(S \cap \Gamma)$, as $F \equiv 0$ on $\partial S$, whereas on $S \cap \partial\Gamma$ the vector field

$$(G/s)_y \vec{\imath} - (G/s)_x \vec{\jmath}$$

is orthogonal to $\nabla(G/s)$, which in turn is parallel to $\vec{n}$. The remainder of the proof plays out exactly as that of Theorem 5. Q.E.D.

*Example:* Consider the general class of quadric surfaces in three dimensions, with equation:

$$p_1 + p_2 x + p_3 y + p_4 z + p_5 x^2 + p_6 xy + p_7 y^2 + p_8 yz + p_9 z^2 + p_{10} xz = 0.$$

Take $S$ to be the tetrahedral region $x \geq 0$, $y \geq 0$, $z \geq 0$, $x + y + z \leq 1$. Choose

$$F(x,y,z) = \max\left(0, xyz[1 - x - y - z]\right). \tag{13}$$

Choose $s(\vec{x}) \equiv 1$. Select also "$x$" and "$y$" of the theorem to be the same as the first two coordinates $x$ and $y$. Then we have:

$$
\begin{aligned}
U_1(\vec{x}) &= \Upsilon(F, s, 1) = 0, \\
U_2(\vec{x}) &= \Upsilon(F, s, x) = (x + 2y + z - 1)xz, \\
U_3(\vec{x}) &= \Upsilon(F, s, y) = (1 - 2x - y - z)yz, \\
U_4(\vec{x}) &= \Upsilon(F, s, z) = 0, \\
U_5(\vec{x}) &= \Upsilon(F, s, x^2) = 2(x + 2y + z - 1)x^2 z, \\
U_6(\vec{x}) &= \Upsilon(F, s, xy) = (y - x)xyz, \\
U_7(\vec{x}) &= \Upsilon(F, s, y^2) = 2(1 - 2x - y - z)y^2 z, \\
U_8(\vec{x}) &= \Upsilon(F, s, yz) = (1 - 2x - y - z)yz^2,
\end{aligned}
$$

$$\begin{aligned} U_9(\vec{x}) &= \Upsilon(F, s, z^2) = 0, \\ U_{10}(\vec{x}) &= \Upsilon(F, s, xz) = (x + 2y + z - 1)xz^2. \end{aligned}$$

If we now further suppose that our volumetric dataset $I(x, y, z)$ is zero-valued above the parabolic cylinder with equation $-4 + 8z + 16x^2 + 24xy + 9y^2 = 0$ and has value 10000 below the cylinder (as shown in Fig. 11), we can calculate the corresponding moment values over $S$:

$$\begin{array}{rrr} u_1 = 0, & u_2 = -2.79707, & u_3 = 3.72942 \\ u_4 = 0, & u_5 = 1.06555, & u_6 = 0.04899, \\ u_7 = 1.76367, & u_8 = 0.81002, & u_9 = 0, \\ u_{10} = -0.60751. & & \end{array}$$

It is easily verified that (12) holds for the equation of the parabolic cylinder.

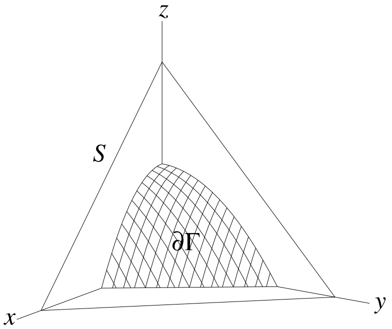


**Fig. 11** Parabolic cylinder intersecting a tetrahedral region of three-dimensional space.

## 6 Implicit versus Explicit Solution

Theorems 5 and 6 differ in character from Theorem 4 in that the latter gives an explicit formula for recovering the parameters of an equation whereas the former give only implicit information in the form of a linear equation to be satisfied by the parameters. Solving an appropriate problem requires only a single application of Theorem 4 versus multiple applications of Theorem 5 or 6 (with various choices of $F$, $s$, and possibly the pair of coordinates involved). Obviously the explicit solution is preferable; however, this may not be available in all situations, as the following theorem demonstrates.

**Theorem 7.** *Let $U_1$, $U_2$, $U_3$, $U_4$ be integrable functions with bounded support whose associated moment values recover the equation of a circle; in other words, if a two-valued image function $I(x, y)$ has circular boundary curve $\partial\Gamma$ with equation*

$$p_1 + p_2 x + p_3 y + p_4 \left(x^2 + y^2\right) = 0, \tag{14}$$

*then*

$$u_n = \langle U_n, I \rangle = c p_n, \tag{15}$$

*where $c$ is a constant possibly depending on the particular boundary circle but independent of $n$. Then (15) holds trivially by $u_n = c = 0$ for all circles.*

*Proof:* The proof hinges on the fact that a straight line is describable by (14) as a limiting case of a circle. We first establish a relationship between the Fourier transform of a moment function $U$:

$$\mathcal{U}(\omega_1, \omega_2) = \mathcal{F}(U) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i(\omega_1 x + \omega_2 y)} U(x, y)\, dx\, dy$$

and the moment values of $U$ for straight-line images. For given $\theta$, $\tau$ define $I^{\theta\tau}$ to be an image function with straight-line boundary parametrized by $\theta$ and $\tau$:

$$I^{\theta\tau}(x, y) = \begin{cases} 1 \text{ if } x\cos\theta + y\sin\theta \le \tau \\ 0 \text{ otherwise.} \end{cases}$$

Let

$$\omega = \sqrt{\omega_1^2 + \omega_2^2}$$

and fix $\theta$ so that $\omega_1 = \omega\cos\theta$, $\omega_2 = \omega\sin\theta$. Define rotated coordinates $(s, t)$:

$$s = -x\sin\theta + y\cos\theta,\ t = x\cos\theta + y\sin\theta.$$

Then

$$\mathcal{U}(\omega_1, \omega_2) = \int_{-\infty}^{\infty} e^{-i\omega t} \int_{-\infty}^{\infty} U_{\mathrm{R}}(s, t)\, ds\, dt,$$

where $U_{\mathrm{R}}$ denotes $U$ as expressed in rotated coordinates.

Consider on the other hand the moment value $u^{\theta\tau}$ of $U$ with respect to $I^{\theta\tau}$:

$$u^{\theta\tau} = \langle U, I^{\theta\tau} \rangle = \int_{-\infty}^{\tau} \int_{-\infty}^{\infty} U_{\mathrm{R}}(s, t)\, ds\, dt.$$

Then

$$\left.\frac{\partial u^{\theta\tau}}{\partial \tau}\right|_{\tau = t} = \int_{-\infty}^{\infty} U_{\mathrm{R}}(s, t)\, ds.$$

We can now rewrite the Fourier transform of $U$:

$$\mathcal{U}(\omega_1, \omega_2) = \int_{-\infty}^{\infty} e^{-i\omega t} \left.\frac{\partial u^{\theta\tau}}{\partial \tau}\right|_{\tau = t} dt, \tag{16}$$

To complete the proof, apply (16) to $U = U_4$. Since $p_4 = 0$ for any straight line and $u_4 = c p_4$, it follows that $u_4^{\theta\tau} \equiv 0$ for any $\theta$ and $\tau$. We therefore find

$$0 \equiv \frac{\partial u_4^{\theta\tau}}{\partial\tau} \equiv \mathcal{U}_4(\omega_1, \omega_2) \equiv U_4(x_1, x_2).$$

It follows finally that $u_4 = 0$ for any curve described by (14). For any circle, $p_4 \neq 0$ and we therefore have $c = 0$. Q.E.D.

With a bit more effort one can prove that $U_1 \equiv U_2 \equiv U_3 \equiv 0$ as well.

An explicit parametrization of the circle in terms of moment values is, however, achievable by performing the calculation in two stages ([25]). The first stage effectively reduces the number of free parameters by one, so that the second stage yields an explicit solution in accordance with Theorem 8 below.

The existence of an explicit moment-function solution for at least one curve class (straight lines) together with nonexistence of an explicit moment-function solution for another class (circles) raises the question of what classes have explicit solutions. Our remaining theorems present explicit solutions for particular curve and surface classes.

The first such class is a *two-parameter* class; i.e. one for which (4) has just three terms.

**Theorem 8.** *Suppose the equation for boundary curve $\partial\Gamma$ consists of three terms:*

$$p_1 z_1(\vec{x}) + p_2 z_2(\vec{x}) + p_3 z_3(\vec{x}) = 0,$$

*at least one of $p_1$, $p_2$, and $p_3$ being nonzero. Let $V$ be an open set containing $S$. Assume the functions $z_1$, $z_2$, $z_3$ to be twice continuously differentiable on $V$. Choose a function $F(\vec{x})$ continuously differentiable on $V$ such that $F(\vec{x}) = 0$ for $\vec{x} \notin S$. Let $1 \leq i, j \leq K$ be given, $i \neq j$, and write $x$ for $x_i$, $y$ for $x_j$. Define:*

$$\begin{aligned} U_1(\vec{x}) &= \Upsilon(F, z_2, z_3), \\ U_2(\vec{x}) &= \Upsilon(F, z_3, z_1), \\ U_3(\vec{x}) &= \Upsilon(F, z_1, z_2), \end{aligned}$$

*where*

$$\Upsilon(F, s, z) = \left(\left[s z_y - z s_y\right] F\right)_x - \left(\left[s z_x - z s_x\right] F\right)_y,$$

*the $x$ and $y$ subscripts indicating partial differentiation. Let*

$$u_1 = \langle U_1, I\rangle, \ u_2 = \langle U_2, I\rangle, \ u_3 = \langle U_3, I\rangle.$$

*Then all points in $\partial\Gamma$ satisfy:*

$$u_1 z_1(\vec{x}) + u_2 z_2(\vec{x}) + u_3 z_3(\vec{x}) = 0.$$

*Proof:* Note that $\Upsilon(F, s, z) = -\Upsilon(F, z, s)$ and $\Upsilon(F, z, z) = 0$. Applying Theorem 6, we find:

$$u_3 p_2 - u_2 p_3 = 0,$$

$$-u_3p_1 + u_1p_3 = 0,$$
$$u_2p_1 - u_1p_2 = 0.$$

Clearly this system is satisfied provided $p_n = cu_n$ for an arbitrary constant $c$. It remains to prove no other nontrivial solutions exist. This can be seen by considering the coefficient matrix of the preceding linear system:

$$\begin{pmatrix} 0 & u_3 & -u_2 \\ -u_3 & 0 & u_1 \\ u_2 & -u_1 & 0 \end{pmatrix}.$$

As an antisymmetric real matrix, this must have even rank: either 0 or 2. If the rank is 2, then the given solution is unique up to a scalar multiple. If the rank is 0, then we have $u_1 = u_2 = u_3 = 0$ and the theorem holds trivially. Q.E.D.

Note that Theorem 4 follows as a corollary of Theorem 8 if $K = 2$ and $z_1(x, y) = x$, $z_2(x, y) = y$, $z_3(x, y) = 1$.

Our next theorem gives a construction of moment functions suitable for directly recovering the equation of a hyperplane in $K$ dimensions:

**Theorem 9.** *Suppose $\partial\Gamma$ is a hyperplane, so that all points $\vec{x} \in S \cap \partial\Gamma$ satisfy (6). Let $V$ be an open set containing $S$. Choose $F(\vec{x})$ continuously differentiable on $V$ such that $F(\vec{x}) = 0$ for $\vec{x} \notin S$. For $n = 1, \ldots, K$, define*

$$U_n = \frac{\partial F}{\partial x_n}.$$

*Further define*

$$U_{K+1} = -\sum_{n=1}^{K} \frac{\partial}{\partial x_n} (x_n F).$$

*For $n = 1, \ldots, K+1$, let*

$$u_n = \langle U_n, I \rangle.$$

*Then all points in $S \cap \partial\Gamma$ satisfy:*

$$\sum_{n=1}^{K} u_n x_n + u_{K+1} = 0. \tag{17}$$

*Proof:* First consider the case $q = 1$, $r = 0$ in (3), so that for a general function $U$, we have:

$$\langle U, I \rangle = \int_S I(\vec{x}) U(\vec{x})\, dV = \int_{S\cap\Gamma} U(\vec{x})\, dV.$$

Let $\vec{U}$ denote the $K$-dimensional vector with components $\langle U_1, \ldots, U_K\rangle$; then $\vec{U} = \nabla F$. Moreover, note $U_{K+1} = -\nabla \cdot (\vec{x}F)$. We can therefore recast our desired result (17) in coordinate-free form:

$$\langle \nabla F, I\rangle \cdot \vec{x} - \langle \nabla \cdot (\vec{x}F), I\rangle = 0.$$

We may now without loss of generality assume that our hyperplane $\partial\Gamma$ is oriented with unit normal vector $\vec{n}$ parallel to the $x_1$-axis, so that it satisfies equation $x_1 + c = 0$ for some constant $c$.

Choose $\Omega$ to be a closed piecewise-smooth hypersurface contained in $V$ such that $S \cap \Omega = S \cap \partial\Gamma$ (as shown in Fig. 12a).

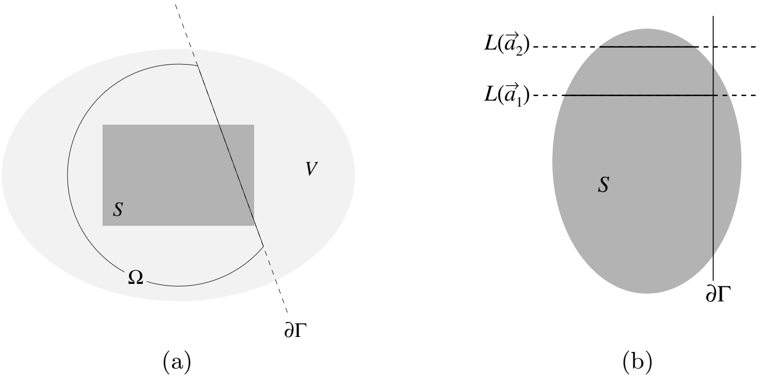


**Fig. 12** (a) Relationships among $S$, $V$, $\partial\Gamma$, and $\Omega$. Compact set $S$ is contained in open set $V$. $\partial\Gamma$ and $\Omega$ coincide where both interesect $S$. $\Omega$ is a closed hypersurface contained in $V$. (b) A line orthogonal to the hyperplane $\partial\Gamma$ may or may not interset $\partial\Gamma$ at a point within $S$.

We find then by the Divergence Theorem:

$$\langle U_{K+1}, I\rangle = -\int_\Omega F(\vec{x})\, \vec{x} \cdot \vec{n}\, dA = -\int_\Omega F(\vec{x})\, x_1\, dA,$$

where $dA$ is the area element for $\partial\Gamma$. Since $\Omega \cap S = \partial\Gamma \cap S$ and $F(\vec{x}) = 0$ for $\vec{x} \notin S$,

$$\int_\Omega F(\vec{x})\, x_1\, dA = \int_{S\cap\Omega} F(\vec{x})\, x_1\, dA = \int_{S\cap\partial\Gamma} F(\vec{x})\, x_1\, dA = \int_{\partial\Gamma} F(\vec{x})\, x_1\, dA.$$

Further, since $x_1 = -c$ on $\partial\Gamma$, we find:

$$\langle U_{K+1}, I\rangle = -\int_{\partial\Gamma} F(\vec{x})\, x_1\, dA = c\int_{\partial\Gamma} F(\vec{x})\, dA. \tag{18}$$

Consider a point $\vec{a} = (-c, a_2, a_3, \ldots, a_K)$ in $\partial\Gamma$. Denote the line through $\vec{a}$ and parallel to $\vec{n}$ by $L(\vec{a})$. This line has equation $x_n = a_n$, $n = 2, 3, , \ldots, K$. Since $F(\vec{x}) = 0$ for $x \in \partial S$, we find

$$\int_{S \cap L(\vec{a})} \frac{\partial F}{\partial x_1}\, dx_1 = \begin{cases} F(\vec{a}) \text{ if } \vec{a} \in \partial\Gamma \\ 0 \qquad \text{otherwise.} \end{cases}$$

We can write $dV = dx_1 dA$, where $dx_1$ is the length element in the direction of $\vec{n}$ and $dA$ is the area element orthogonal to $\vec{n}$. It follows that

$$\langle U_1, I \rangle = \int_{\vec{a} \in S \cap \partial\Gamma} \int_{S \cap L(\vec{a})} \frac{\partial F}{\partial x_1}\, dx_1 dA = \int_{\partial\Gamma} F(\vec{x})\, dA. \tag{19}$$

$$\langle U_{K+1}, I \rangle = -\int_{\partial\Gamma} F(\vec{x})\, x_1\, dA = c \int_{\partial\Gamma} F(\vec{x})\, dA, \tag{20}$$

as illustrated in Fig. 12b.

Similarly, let $L_\perp$ be a line orthogonal to $\vec{n}$. For $j \neq 1$, we find

$$\int_{L_\perp \cap S} \frac{\partial F}{\partial x_j}\, dx_j = 0.$$

It follows that

$$\langle U_j, I \rangle = \int_{S \cap \Gamma} \frac{\partial F}{\partial x_j}\, dV = 0. \tag{21}$$

We can now combine (20), (19), and (21), to yield (17):

$$\begin{aligned} \sum_{n=1}^{K} u_n x_n + u_{K+1} &= \langle U_1, I \rangle x_1 + \sum_{n=2}^{K} \langle U_n, I \rangle x_n + \langle U_{K+1}, I \rangle \\ &= \left( \int_{\partial\Gamma} F(\vec{x})\, dA \right) (-c) + 0 + c \int_{\partial\Gamma} F(\vec{x})\, dA \\ &= 0. \end{aligned}$$

For the general case of arbitrary $q$ and $r$, consider first the case of a constant dataset function $I'(\vec{x})$. Such a function can be considered as the same type as the foregoing $I(\vec{x})$, except that $S$ lies entirely on one side of the hyperplane, so that $\partial\Gamma$ does not intersect $S$. We then have $\langle U_n, I' \rangle = 0$ for all $n = 1, \ldots, K+1$.

A dataset function $I''(\vec{x})$ with arbitrary $q$ and $r$ can then be represented in the form:

$$I''(\vec{x}) = (q - r) I(\vec{x}) + r I'(\vec{x}),$$

where $I$ satisfies the previous special condition $q = 1, r = 0$ and $I'$ is constant. We have then

$$\langle U_n, I'' \rangle = (q - r) \langle U_n, I \rangle \text{ for } n = 1, \ldots, K+1.$$

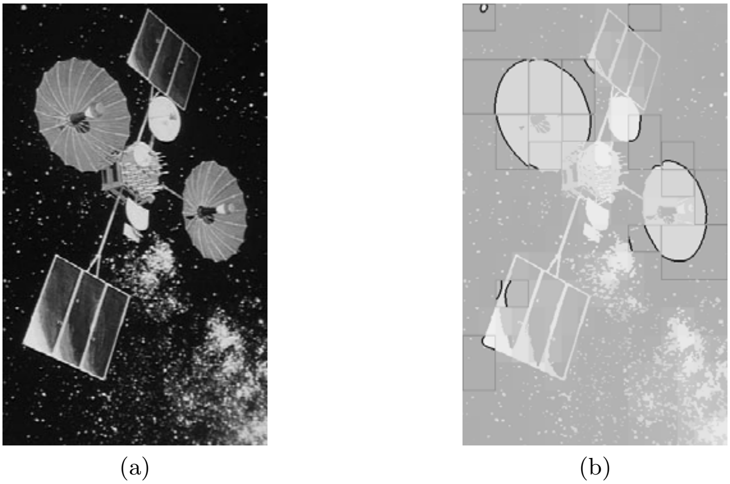


**Fig. 13** (a) Original image characterized by multiple boundary curves, both conic and approximately piecewise-conic. (b) Various boundary curve located with moments controlled by recursive subdivision.

This reduces (17) to the special case $q = 1$, $r = 0$. Q.E.D.

Note that it is possible for Theorem 9 to hold vacuously, in the case that all the specified moment values are zero. However, since, as noted in Formula (19),

$$\langle U_1, I \rangle = \int_{\partial \Gamma} F(\vec{x})\, dA,$$

this possibility is forestalled provided $F$ is chosen to be nonnegative and the hyperplane intersects the support of $F$ nontrivially.

Note also that Theorem 4 is the special case $K = 2$ of Theorem 9.

We conclude this section by returning to the "trees-versus-forest" issue raised in the introductory section. More sophisicated strategies can improve the performance of moment-based techniques on images characterized by low-frequency distortion, obstruction, and clutter. One approach is to attempt to fit a curve to a large region, and subdivide the region if no good fit is found, continuing recursively until either a good fit is found or the region is judged too small to be interesting. Fig. 13 shows an example of this approach. Space limitations preclude more details here, but more discussion can be found in [24].

## Appendix

This Appendix lists the moment functions used in our examples of parametrization of conic curves. We use an $18 \times 6$ matrix of moment values. The 6 columns are defined by the 6 functions $z_n$ in (5) corresponding to the general conic equation. The 18 rows are defined by combinations of the 3 functions

$$F_A(x,y) = (1-x^2)(1-y^2),\ F_B(x,y) = xF_A(x,y),\ F_C(x,y) = yF_A(x,y)$$

with the 6 functions $s_1, \ldots, s_6$, where $s_n = z_n$, the moment function corresponding to a given matrix entry given by the operator $\Upsilon(F_i, s_j, z_n)$.

It is convenient to consider the $18 \times 6$ matrix as a combination of three $6 \times 6$ matrices $A$, $B$, and $C$, each generated by a particular $F_i$:

$$\begin{aligned} A_{j,n} &= \Upsilon(F_A, z_j, z_n), \\ B_{j,n} &= \Upsilon(F_B, z_j, z_n), \\ C_{j,n} &= \Upsilon(F_C, z_j, z_n). \end{aligned}$$

Each of $A$, $B$, $C$ is an antisymmetric matrix, since in general

$$\Upsilon(F, s, z) = -\Upsilon(F, z, s);$$

It therefore suffices to specify the 15 elements lying above the diagonal of each:

$$\begin{aligned}
A_{1,2} &= 2y(1-x^2), \\
A_{1,3} &= -2x(1-y^2), \\
A_{1,4} &= 4xy(1-x^2), \\
A_{1,5} &= -4xy(1-y^2), \\
A_{1,6} &= 2y^2 - 2x^2, \\
A_{2,3} &= 2 - 4x^2 - 4y^2 + 6x^2y^2, \\
A_{2,4} &= 2x^2y(1-x^2), \\
A_{2,5} &= 2y(2 - 4x^2 - 3y^2 + 5x^2y^2), \\
A_{2,6} &= 2x(1-2x^2)(1-y^2), \\
A_{3,4} &= -2x(2 - 3x^2 - 4y^2 + 5x^2y^2), \\
A_{3,5} &= -2xy^2(1-y^2), \\
A_{3,6} &= -2y(1-x^2)(1-2y^2), \\
A_{4,5} &= 4xy(2 - 3x^2 - 3y^2 + 4x^2y^2), \\
A_{4,6} &= 2x^2(2 - 3x^2 - 3y^2 + 4x^2y^2), \\
A_{5,6} &= -2y^2(2 - 3x^2 - 3y^2 + 4x^2y^2), \\
B_{1,2} &= 2xy(1-x^2), \\
B_{1,3} &= (1-3x^2)(1-y^2), \\
B_{1,4} &= 4x^2y(1-x^2),
\end{aligned}$$

$$\begin{aligned}
B_{2,6} &= x^2(3-5x^2)(1-y^2), \\
B_{3,4} &= -x^2(5 - 7x^2 - 9y^2 + 11x^2y^2), \\
B_{3,5} &= y^2(1-3x^2)(1-y^2), \\
B_{3,6} &= -2xy(1-x^2)(1-2y^2), \\
B_{4,5} &= 2x^2y(5 - 7x^2 - 7y^2 + 9x^2y^2), \\
B_{4,6} &= x^3(5 - 7x^2 - 7y^2 + 9x^2y^2), \\
B_{5,6} &= -xy^2(5 - 7x^2 - 7y^2 + 9x^2y^2), \\
C_{1,2} &= -(1-x^2)(1-3y^2), \\
C_{1,3} &= -2xy(1-y^2), \\
C_{1,4} &= -2x(1-x^2)(1-3y^2), \\
C_{1,5} &= -4xy^2(1-y^2), \\
C_{1,6} &= -y(1 + x^2 - 3y^2 + x^2y^2), \\
C_{2,3} &= y(3 - 5x^2 - 5y^2 + 7x^2y^2), \\
C_{2,4} &= -x^2(1-x^2)(1-3y^2), \\
C_{2,5} &= y^2(5 - 9x^2 - 7y^2 + 11x^2y^2), \\
C_{2,6} &= 2xy(1-2x^2)(1-y^2), \\
C_{3,4} &= -2xy(3 - 4x^2 - 5y^2 + 6x^2y^2), \\
C_{3,5} &= -2xy^3(1-y^2),
\end{aligned}$$

$$\begin{aligned} B_{1,5} &= 2y(1-3x^2)(1-y^2), & C_{3,6} &= -y^2(1-x^2)(3-5y^2),\\ B_{1,6} &= x(1-3x^2+y^2+x^2y^2), & C_{4,5} &= 2xy^2(5-7x^2-7y^2+9x^2y^2),\\ B_{2,3} &= x(3-5x^2-5y^2+7x^2y^2), & C_{4,6} &= x^2y(5-7x^2-7y^2+9x^2y^2),\\ B_{2,4} &= 2x^3y(1-x^2), & C_{5,6} &= -y^3(5-7x^2-7y^2+9x^2y^2),\\ B_{2,5} &= 2xy(3-5x^2-4y^2+6x^2y^2). \end{aligned}$$

The coefficients $p_n$ in (5) are then obtained by solving the system of 18 equations:

$$\sum_{n=1}^{6} A_{mn}p_n = 0,\ \sum_{n=1}^{6} B_{mn}p_n = 0,\ \sum_{n=1}^{6} C_{mn}p_n = 0,$$

$m = 1, \ldots, 6$.

## References

[1] Schoenberg, I.: Approximation: Theory and Practice. Stanford University, Stanford (1955)

[2] Hough, P.V.C.: Method and Means for Recognizing Complex Patterns. U.S. Patent 3,069,654 (1962)

[3] Ahiezer, N.I., Kreĭn, M.G.: Some Questions in the Theory of Moments. American Mathematical Society, Providence (1962)

[4] Davis, J.: Triangle formulas in the complex plane. Math. of Comp. 18, 569–577 (1964)

[5] Kreĭn, M.G., Nudelman, A.A.: The Markov Moment Problem and Extremal Problems. American Mathematical Society, Providence (1977)

[6] Machuca, R., Gilbert, A.L.: Finding Edges in Noisy Scenes. IEEE Trans. Pattern Anal. and Machine Intell. 3(1), 103–110 (1981)

[7] Reeves, A.P., Akey, M.L., Mitchell, O.R.: A Moment Based Two-Dimensional Edge Operator. In: IEEE Comput. Soc. Symp. Computer Vision and Pattern Recognition, pp. 312–317 (1983)

[8] Canny, J.F.: A Computational Approach to Edge Detection. IEEE Trans. Pattern Anal. and Machine Intel. 8(6), 679–698 (1986)

[9] Lyvers, E.P., Mitchell, O.R., Akey, M.L., Reeves, A.P.: Subpixel Measurements Using a Moment-Based Edge Operator. IEEE Trans. Pattern Anal. Machine Intell. 11(12), 1293–1309 (1989)

[10] Xu, L., Oja, E., Kultanen, P.: A New Curve Detection Method: Randomized Hough Transform (RHT). Pattern Recognition Letters 11(5), 331–338 (1990)

[11] Ghosal, S., Mehrotra, R.: Orthogonal Moment Operators for Subpixel Edge Detection. Pattern Recognition 26(2), 295–306 (1993)

[12] Yoo, J.H., Sethi, I.K.: Ellipse Detection Method from the Polar and Pole Definition of Conics. Pattern Recognition 26(2), 307–315 (1993)

[13] Mat Jafri, M.Z., Deravi, F.: Efficient Algorithm for Detection of Parabolic Curves. In: Proc. SPIE. Vision Geometry III, vol. 2356, pp. 53–61 (January 1995)

[14] Liao, S.X., Pawlak, M.: On image analysis by moments. IEEE Trans. Pattern Anal. Machine Intell. 18(3), 254–266 (1996)

[15] Ghosal, S., Mehrotra, R.: A Moment-Based Unified approach to Image Feature Detection. IEEE Trans. Image Processing 6(6), 781–793 (1997)

[16] Heikkilä, J.: Moment and Curvature Preserving Technique for Accurate Ellipse Boundary Detection. In: Proc. 14th Int'l Conf. Pattern Recognition (ICPR 1998), vol. 1, pp. 734–737 (1998)

[17] Donoho, D.L.: Wedgelets: nearly minimax estimations of edges. The Annals of Statistics 27(3), 859–897 (1999)

[18] Gustafsson, B., He, C., Milanfar, P., Putinar, M.: Reconstructing Planar Domains from their Moments. Inverse Problems 16(4), 1053–1070 (2000)

[19] Mukundan, R., Ong, S.H., Lee, P.A.: Image analysis by Tchebichef moments. IEEE Trans. Image Processing 10(9), 1357–1364 (2001)

[20] Romberg, J.K., Wakin, M.B., Baraniuk, R.G.: Multiscale Geometric Image Processing. In: Proc. SPIE Visual Commun. Image Process., vol. 5150, pp. 1265–1272 (2003)

[21] Popovici, I., Withers, W.D.: Custom-built moments for edge location. IEEE Trans. Pattern Anal. and Machine Intell. 28(4), 637–642 (2006)

[22] Popovici, I., Withers, W.D.: Locating thin lines and roof edges by custom-built moments. In: Proc. IEEE Int'l. Conf. Image Processing, pp. 753–756 (2006)

[23] Popovici, I., Withers, W.D.: Locating edges and removing ringing artifacts in JPEG images by frequency-domain analysis. IEEE Trans. Image Processing 16(5), 1470–1474 (2007)

[24] Popovici, I., Withers, W.D.: Curve parametrization by moments. IEEE Trans. Pattern Anal. and Machine Intell. 31(1), 15–26 (2009)

[25] Cho, V., Withers, W.D.: Circle Location by Moments. In: Elmoataz, A., et al. (eds.) ICISP 2010. LNCS, vol. 6134, pp. 94–102. Springer, Heidelberg (2010)

# Chapter 12
# Intelligent Approaches to Colour Palette Design

Gerald Schaefer

**Abstract.** Colour palettes are used for representing image data using a limited number of colours. As the image quality directly depends on the chosen colours in the palette, deriving algorithms for colour palette design is a crucial task. In this chapter we show how computational intelligence approaches can be employed for this task. In particular, we discuss the use of generic optimisation techniques such as simulated annealing, and of soft computing based clustering algorithms founded on fuzzy and rough set ideas in the context of colour quantisation. We show that these methods are capable of deriving good colour palettes and that they outperform standard colour quantisation techniques in terms of image quality.

**Keywords:** Colour imaging, colour quantisation, colour palette, optimisation, clustering, simulated annealing, fuzzy c-means, rough c-means.

## 1 Introduction

Colour quantisation is a common image processing technique that allows the representation of true colour images using only a small number of colours. True colour images typically use 24 bits per pixel resulting overall in $2^{24}$, i.e. more than 16 million different colours. Colour quantisation uses a colour palette that contains only a small number of distinct colours (usually between 8 and 256) and pixel data are then stored as indices to this palette. Clearly, the choice of the colours that make up the palette is of crucial importance for the quality of the quantised image.

A common way of expressing this quality is to calculate the difference between the original (unquantised) image $O$ and its colour quantised counterpart $Q$ for which the mean-squared error (MSE)

Gerald Schaefer
Department of Computer Science
Loughborough University
Loughborough, U.K.
e-mail: gerald.schaefer@ieee.org

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 275–289.
springerlink.com

$$\mathrm{MSE}(O,Q) = \frac{1}{3nm}\sum_{i=1}^{n}\sum_{j=1}^{m}((R_O(i,j) - R_Q(i,j))^2 + (G_O(i,j) - G_Q(i,j))^2 + (B_O(i,j) - B_Q(i,j))^2), \tag{1}$$

where $R(i,j)$, $G(i,j)$, and $B(i,j)$ are the red, green, and blue pixel values at location $(i,j)$ and $n$ and $m$ are the dimensions of images, is the most widely used measure.

However, the selection of the optimal colour palette is known to be an np-hard problem (Heckbert, 1982). In the image processing literature many different algorithms have been introduced that aim to find a palette that allows for good image quality of the quantised image. A relatively simple approach is the Popularity algorithm (Heckbert, 1982), which - typically following a uniform quantisation to 5 bits per channel - selects the $n$ colours that are represented most often to form the colour palette. In Median cut quantisation (Heckbert, 1982), an iterative procedure repeatedly splits (by a plane through the median point) colour cells into sub-cells. In Octree quantisation (Gervautz and Purgathofer, 1990), the colour space is represented as an octree where sub-branches are successively merged to form the palette, while Neuquant (Dekker, 1994) employs a one-dimensional self-organising Kohonen neural network to generate the colour map.

In this chapter, we present some computational intelligence approaches to colour quantisation. In particular, in Section 2 we show how general purpose optimisation algorithms such as simulated annealing can be used to arrive at a good colour palette. Moreover, we demonstrate how a hybrid optimisation scheme can be developed for colour quantisation. Colour quantisation can also be regarded as a clustering problem. Consequently, in Section 3 we present several fuzzy-based clustering algorithms for this task while in Section 4 we introduce a rough set based clustering approach for colour palette design. In Section 5 we present experimental results that confirm that the introduced methods are effective approaches for colour quantisation outperforming several standard algorithms. Finally, Section 6 concludes the chapter.

## 2 Optimisation for Colour Palette Design

### Simulated Annealing

The main advantage of black-box optimisation algorithms is that they do not require any domain specific knowledge yet are able to provide a near optimal solution. Simulated annealing (SA) was first introduced as a general optimisation method by Kirkpatrick et al. (1983). It simulates the annealing of metal, in which the metal is heated-up to a temperature near its melting point and then slowly cooled down. This allows the particles to move towards a minimum energy state, with a more uniform crystalline structure. The process therefore permits some control over the microstructure.

Simulated annealing is a variation of the hill-climbing algorithm. Both start from a randomly selected point within the search space of all the possible solutions. Each point in the search space has a measurable error value, $E$, associated with it, which indicates the quality of the solution. From the current point in search space, new trial solutions are selected for testing from the neighborhood of the current solution. This is usually done by moving a small step in a random direction. In this application, small and equally distributed random numbers from the interval $[-s_{max}, s_{max}]$ are added to each component of the current solution vector, where $s_{max}$ is called the maximum step width. The values for $s_{max}$ need to be chosen from the interval between 0 and the upper limit of the search space dimension. The decrease in error values is denoted as $\Delta E$. If $\Delta E$ is negative, i.e. the error of a trial solution is less than the error of the current one, the trial solution is accepted as the current solution.

Unlike hill-climbing SA does not automatically reject a new candidate solution if $\Delta E$ is positive. Instead it becomes the current solution with probability $p(T)$ which is usually determined using

$$p(T) = e^{-\Delta E/T} \tag{2}$$

where $T$ is referred to as "temperature", an abstract control parameter for the cooling schedule. For a given temperature and positive values of $\Delta E$ the probability function shown in Equation (2) has a defined upper limit of 1, and tends towards 0 for large positive values of $\Delta E$.

The algorithm starts with a high temperature i.e. with a high transition probability. The temperature is then reduced towards 0, usually in steps, according to a cooling schedule such as

$$T_{n+1} = \alpha T_n \tag{3}$$

where $T_n$ is the temperature at step $n$ and $\alpha$ is the cooling coefficient (usually between 0.8 and 0.99).

During each step the temperature must be held constant for an appropriate number of iterations in order to allow the algorithm to settle into a "thermal equilibrium", i.e. a balanced state. If the number of iterations is too small the algorithm is likely to converge to a local minimum.

## Stepwidth Adaptive Simulated Annealing

For both continuous parameter optimisation and discrete parameters with large search ranges, it is practically impossible to choose direct neighbours of the current solution as new candidate solutions, simply because of the vast number of points in the search space. Therefore it is necessary to choose new candidates at some distance in a random direction of the current solution in order to navigate in an acceptable time through the search space. This distance could either be a fixed step width $s$ or it could have an upper limit $s_{max}$. In the first case, the neighbourhood would be defined as the surface of a hypersphere around the current solution, whereas in the second case the neighbourhood would be the volume of the hypersphere. In the latter, new candidate solutions might be generated by adding small, equally distributed random numbers from the interval $[-s_{max}, s_{max}]$ to each component of the current solution vector.

The maximum step width $s_{max}$ is crucial to the success of SA. If $s_{max}$ is chosen too small and the start point for a search run is too far away from the global optimum, the algorithm might not be able to get near that optimum before the algorithm "freezes", i.e. the temperature becomes so small that $p(T)$ is virtually zero and the algorithm starts to perform only hill climbing and will consequently get stuck in the nearest local optimum rather than finding the global one. If, on the other hand, the step width has been chosen to be too large and the peak of the optimum is very narrow, the algorithm might well get near the global optimum before the algorithm freezes, but never reaches the top because most of the steps are too large so that new candidate solutions "fall off" the peak. Hence, there is always a trade-off between accuracy and robustness in selecting an appropriate maximum step width. If $s_{max}$ is too small, SA has the potential to reach the peak of the "frozen-in" optimum, but it cannot be guaranteed that this optimum is the global one. On the other hand, if $s_{max}$ is too large, SA has the potential to get near the global optimum, but it might never reach the top of it.

Step width adapting simulated annealing (SWASA) (Nolle, 2004) overcomes the problems associated with constant values for $s_{max}$ by using a scaling function to adapt the maximum step width to the current iteration by

$$s_{max}(n) = \frac{2s_0}{1 + e^{\beta n / n_{\max}}} \tag{4}$$

where $s_{max}(n)$ is the maximum step width at iteration $n$, $s_0$ is the initial maximum step width, $n_{max}$ the maximum number of iterations and $\beta$ is an adaptation constant.

We employ a population based version of the SWASA algorithm with a population size of 10. The start temperature $T_0$ was chosen to be 20 and the cooling coefficient $\alpha$ set to 0.9. The parameters $s_0$ and $\beta$ were set to 100 and 5.3 respectively. The temperature was kept constant over 20 iterations and the maximum number of iterations was set to 10000.

For colour quantisation the objective is, as mentioned, to minimise the total error introduced through the application of a colour palette. The colour palette $C$ for an image $I$, a codebook of $k$ colour vectors, is chosen so as to minimise the error function

$$\text{error}(C, I) = \frac{1}{\sum_{j=1}^{k} l_j} \sum_{i=1}^{k} \sum_{j=1}^{l_i} ||C_i - I_j|| + p(C, I) \tag{5}$$

with

$$p(C, I) = \sum_{i=1}^{k} \delta a_i, \quad a_i = \begin{cases} 1 & \text{if } l_i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $l_i$ is the number of pixels $I_j$ represented by colour $C_i$ of the palette, $||.||$ is the Euclidean distance in RGB space, and $\delta$ is a constant ($\delta = 10$ in our experiments). The objective function error$(C, I)$ used is hence a combination of the mean Euclidean distance (and hence the error measure of Equation 1) and a penalty function. The penalty function $p(C, I)$ was integrated in order to avoid unused palette colours by adding a constant penalty value to the error for each entry in the codebook that is not used in the resulting picture.

As can be seen from Equation (5) the objective function is highly non-linear, i.e. it has a high degree of epistasis (Davidor, 1990). Past experience (Nolle et al., 2002) has shown, that for this kind of optimisation problems simulated annealing outperforms other generic optimisation algorithms like genetic algorithms (Holland, 1975) and therefore justifies the use of simulated annealing as an effective method for colour palette design (Nolle and Schaefer, 2007). Nevertheless, of course other optimisation techniques could also be employed to derive a colour palette. For example, in (Scheunders, 1997) genetic algorithms were used for colour quantisation, while in (Omran et al., 2005) particle swarm optimisation (Kennedy and Eberhart, 1995) was utilised.

## Hybrid Optimisation

Figure 1 shows a typical run of the SA method applied to colour quantisation. The solid line represents the average quantisation error over time (iterations) while the dashed line represents the best solution of each iteration.

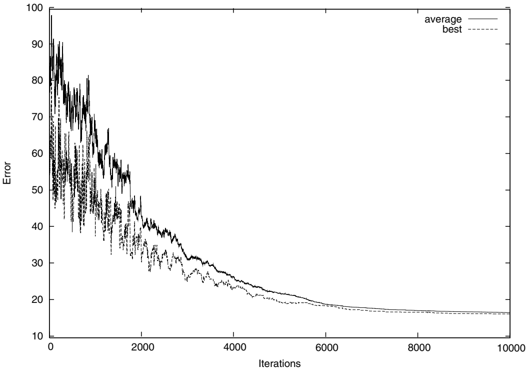


**Fig. 1** Typical run of SA optimisation for colour quantisation.

As can be seen from Figure 1 there is always a variation in error values within the population which indicates that although Simulated annealing is able to find good solutions, i.e. solutions from within the region around the global optimum, it rarely exploits that region completely. Therefore, in a second step, we combine the SA approach with a standard c-means clustering algorithm (Linde et al., 1980) to provide a stacked hybrid optimisation method. C-means clustering is guaranteed to converge towards the local clustering minimum by iteratively carrying out the following two steps:

- Each input vector should be mapped to its closest codeword by a nearest neighbour search.

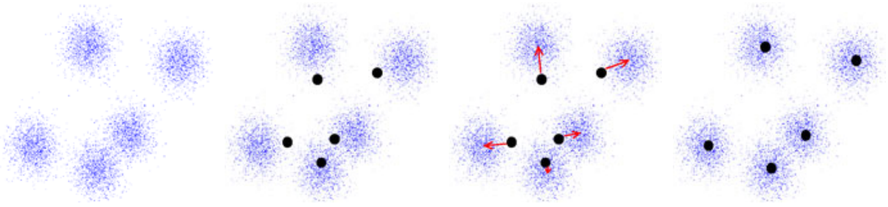
- The input vectors assigned in each class (i.e. for each codeword) are best represented by the centroid of the vectors in that class.

In this hybridised algorithm the SA component is hence responsible for identifying the region in the search space that will contain the global optimum while the c-means component will then descend into the minimum present in that region (Nolle and Schaefer, 2007).

## 3 Fuzzy Clustering for Colour Palette Design

### Fuzzy C-Means

Colour quantisation can also be seen as a clustering problem where the task is to identify those clusters that best represent the colours in an image. Fuzzy c-means (FCM) is based on the idea of finding cluster centres by iteratively adjusting their positions and evaluation of an objective function as in conventional c-means, yet it allows more flexibility by introducing the possibility of partial memberships to clusters. The general FCM algorithm is illustrated in Figure 2.



(a) Data to cluster (b) Random centres (c) Converging (d) Final settlement

**Fig. 2** Illustration of classical FCM that attempts to find appropriate cluster centres.

For colour quantisation, the error function follows this form:

$$E = \sum_{j=1}^{C} \sum_{i=1}^{N} \mu_{ij}^{k} ||x_i - c_j||^2, \tag{7}$$

where $\mu_{ij}^{k}$ is the fuzzy membership of pixel $x_i$ and the colour cluster identified by its centre $c_j$, and $k$ is a constant that defines the fuzziness of the resulting partitions.

$E$ can reach the global minimum when pixels nearby the centroid of corresponding clusters are assigned higher membership values, while lower membership values are assigned to pixels far from the centroid (Chuang et al., 2006). Here, the membership is proportional to the probability that a pixel belongs to a specific cluster where the probability is only dependent on the distance between the image pixel and each independent cluster centre. The membership functions and the cluster centres are updated by

$$\mu_{ij} = \frac{1}{\sum_{m=1}^{C} \left( \frac{||x_j - c_i||}{||x_j - c_m||)^{2/(k-1)}} \right)}, \tag{8}$$

and

$$c_i = \frac{\sum_{j=1}^{N} \mu_{ij}^k x_j}{\sum_{j=1}^{N} \mu_{ij}^k}. \tag{9}$$

The steps involved in fuzzy c-means are (Bezdek, 1980):

1. Initialise the cluster centres $c_i$ and let $t = 0$.
2. Initialise the fuzzy partition memberships functions $\mu_{ij}$ according to Equation (8).
3. Let $t = t + 1$ and compute new cluster centres $c_i$ using Equation (9).
4. Repeat Steps 2 to 3 until convergence.

An initial setting for each cluster centre is required and FCM also converges to a local minimisation solution. The efficiency of FCM has been comprehensively investigated in (Hu and Hathaway, 2002). To effectively address the inefficiency of the original FCM algorithm several variants of the fuzzy c-means algorithm will be introduced in the following.

## Random Sampling FCM

To combat the computational complexity of FCM, Cheng *et al.* (Cheng et al., 1998) proposed a multistage random sampling strategy. This method has a lower number of feature vectors and also needs fewer iterations to converge. The basic idea is to randomly sample and obtain a small subset of the dataset in order to approximate the cluster centres of the full dataset. This approximation is then used to reduce the number of iterations. The random sampling FCM algorithm consists of two phases. First, a multistage iterative process of a modified FCM is performed. Phase 2 is then a standard FCM with the cluster centres approximated by the final cluster centres from Phase 1.

**Phase 1:**
Let $X_{\Delta\%}$ be a subset whose number of subsamples is $\Delta\%$ of the $N$ samples contained in the full dataset $X$ and denote the number of stages as $n$. $\epsilon_1$ and $\epsilon_2$ are parameters used as stopping criteria. After the following steps the dataset (denoted as $X_{(n_s * \Delta\%)}$) will include $N * \Delta\%$ samples:

Step 1: Select $X_{(\Delta\%)}$ from the set of the original feature vectors matrix ($z$ = 1).
Step 2: Initialise the fuzzy memberships functions $\mu$ using Equation (8) with $X_{(z*\Delta\%)}$.
Step 3: Compute the stopping condition $\epsilon$ = $\epsilon_1$-z*(($\epsilon_1$-$\epsilon_2$)/$n_s$) and let $t = 0$
Step 4: Set $t = t + 1$
Step 5: Compute the cluster centres $c_{(z*\Delta\%)}$ using Equation (9).
Step 6: Compute $\mu_{(z*\Delta\%)}$ using Equation (8).
Step 7: If $||\mu^j_{(z*\Delta\%)} - \mu^{j-1}_{(z*\Delta\%)}|| \geq \epsilon$, then go to Step 4.

Step 8: If $z \leq n_s$ then select another $X_{(\Delta\%)}$ and merge it with the current $X_{(z*\Delta\%)}$ and set $z = z + 1$, otherwise move to Phase 2 of the algorithm.

**Phase 2:**

Step 1: Initialise $\mu_{ij}$ using the results from Phase 1, i.e. $c_{(n_s*\Delta\%)}$ with Equation (9) for the full data set.

Step 2: Go to Steps 3 of the conventional FCM algorithm and iterate the algorithm stopping criterion $\epsilon_2$ is met.

Evidence has shown that this improved FCM with random sampling is able to reduce the computation requested in the classical FCM method.

## Fast Generalised FCM / EnFCM

(Ahmed et al., 2002) introduced an alternative to the classical FCM by adding a term that enables the labelling of a pixel to be associated with its neighbourhood. As a regulator, the neighbourhood term can change the solution towards piecewise homogeneous labelling. As a further extension of this work, in (Szilagyi et al., 2003) the EnFCM algorithm was presented. In order to reduce the computational complexity, a linearity-weighted sum image $g$ is formed from the original image, and the local neighbour average image evaluated as

$$g_m = \frac{1}{1+\alpha}\left(x_m + \frac{\alpha}{N_R}\sum_{j\in N_r} x_j\right) \tag{10}$$

where $g_m$ denotes the gray value of the $m$-th pixel of the image $g$, $x_j$ represents the neighbours of $x_m$, $N_R$ is the cardinality of a cluster, $N_r$ represents the set of neighbours falling into a window around $x_m$.

The objective function used is defined as

$$J = \sum_{i=1}^{C}\sum_{i=1}^{q_c} \gamma_l \mu_{ij}^m (g_l - c_i)^2 \tag{11}$$

where $q_c$ denotes the number of colours in the image, and $\gamma_l$ is the number of the pixels of colour $l$ with $l = 1, 2, \ldots, q_c$. Thus, $\sum_{l=1}^{q_c} \gamma_l = N$ under the constraint that $\sum_{i=1}^{C} \mu_{ij} = 1$ for any $l$.

We can obtain the following expressions for membership functions and cluster centres (Cai et al., 2007):.

$$\mu_{il} = \frac{(g_l - c_i)^{-2/m-1}}{\sum_{j=1}^{C}(g_l - c_j)^{-2/m-1}} \tag{12}$$

and

$$s_i = \frac{\sum_{l=1}^{q_c} \gamma_l \mu_{il}^m g_l}{\sum_{l=1}^{q_c} \gamma_l \mu_{il}^m} \tag{13}$$

EnFCM considers a number of pixels with similar colours as a weight. Thus, this process may accelerate the convergence of searching for global similarity. On the other hand, to avoid image blur during the segmentation, which may lead to inaccurate clustering, in (Cai et al., 2007) a measure $S_{ij}$, which incorporates the local spatial relationship $S^s_{ij}$ and the local gray-level relationship $S^g_{ij}$ is used, which is defined as

$$S_{ij} = \begin{cases} S^s_{ij} \times S^g_{ij}, & j \neq i \\ 0, & j = i \end{cases} \tag{14}$$

with

$$S^s_{ij} = \exp\left( \frac{-\max(|p_{cj} - p_{ci}|, |q_{cj} - q_{ci}|)}{\lambda_s} \right) \tag{15}$$

and

$$S^g_{ij} = \exp\left( \frac{-||x_i - x_j||^2}{\lambda_g \times \sigma_g^2} \right) \tag{16}$$

where $(p_{ci}, q_{ci})$ describe the co-ordinates of the $i$-th pixel, $\sigma_g$ is a global scale factor of the spread of $S^s_{ij}$, and $\lambda_s$ and $\lambda_g$ represent scaling factors. $S_{ij}$ replaces $\alpha$ in Equation (10).

Hence, the newly generated image $g$ is updated as

$$g_i = \frac{\sum_{j \in N_i} S_{ij} x_j}{S_{ij}} \tag{17}$$

and is restricted to [0, 255] due to the denominator.

Given a pre-defined number of clusters $C$ and a threshold value $\epsilon > 0$, the fast generalised FCM algorithm proceeds in the following steps:

Step 1: Initialise the clusters $c_j$.
Step 2: Compute the local similarity measures $S_{ij}$ using Equation (14) for all neighbours and windows over the image.
Step 3: Compute linearly-weighted summed image $g$ using Equation (17).
Step 4: Update the membership partitions using Equation (12).
Step 5: Update the cluster centres $c_i$ using Equation (13).
Step 6: If $\sum_{i=1}^{C} ||c_{i(old)} - c_{i(new)}||^2 > \epsilon$ go to Step 4.

## Anisotropic Mean Shift Based FCM

Anisotropic mean shift based FCM is an efficient approach to fuzzy c-means clustering which utilises an anisotropic mean shift algorithm coupled with fuzzy clustering (Schaefer and Zhou, 2009). Mean shift based techniques have been shown to be capable of estimating the local density gradients of similar pixels. These gradient estimates are iteratively performed so that all pixels can find similar pixels in the same image (Comaniciu and Meer, 2002). A standard mean shift approach method uses radially symmetric kernels. Unfortunately, the temporal coherence will be reduced in the presence of irregular structures and noise in the image. This reduced coherence may not be properly detected by radially symmetric kernels and thus, an improved mean shift approach, namely anisotropic kernel mean shift (Wang et al., 2004), provides better performance.

In mean shift algorithms the image clusters are iteratively moved along the gradient of the density function before they become stationary. Those points gathering in an outlined area are treated as the members of the same segment. A kernel density estimate is defined by

$$\tilde{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K(x - x_i), \tag{18}$$

with

$$K(x) = |H|^{-0.5} K(H^{-0.5}x), \tag{19}$$

where $N$ is the number of samples, and $x_i$ stands for a sample from an unknown density function $f$. $K(\cdot)$ is the $d$-variate kernel function with compact support satisfying the regularity constraints, and $H$ is a symmetric positive definite $d \times d$ bandwidth matrix. Usually, we have $K(x) = k_e(\phi)$, where $k_e(\phi)$ is a convex decreasing function, e.g. $c_t e^{-\phi/2}$ for a Gaussian kernel or $c_t \max(1-\phi, 0)$ for an Epanechnikov kernel where $c_t$ is a normalising constant.

If a single global spherical bandwidth is applied, $H = h^2\mathbf{I}$ ($\mathbf{I}$ is identity matrix), then we have

$$\tilde{f}(x) = \frac{1}{Nh^d}\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \tag{20}$$

Since the kernel can be divided into two different radially symmetric kernels, we have the kernel density estimate as

$$\begin{aligned}\tilde{f}(x) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{h^\beta (H_i^\alpha)^q} k^\alpha \cdot \\ \cdot(d(c_i^\alpha, x_i^\alpha, H_i^\alpha)) k^\beta \left(||(c_i^\beta - x_i^\beta)/(h^\beta(H_i^\alpha))||^2\right)\end{aligned} \tag{21}$$

where and $\alpha$ and $\beta$ denote the spatial and temporal components respectively and $d(c_i^\alpha, x_i^\alpha, H_i^\alpha)$ is the Mahalanobis metric, i.e.

$$d(c_i^\alpha, x_i^\alpha, H_i^\alpha) = (x_i^\alpha - c_i^\alpha)^T H_i^{\alpha-1}(x_i^\alpha - c_i^\alpha). \tag{22}$$

Anisotropic mean shift is intended to modulate the kernels during the mean shift procedure. The objective is to keep reducing the Mahalanobis distance so as to group similar samples as much as possible. First, the anisotropic bandwidth matrix $H_i^\alpha$ is estimated with the following constraints:

$$\begin{cases} k_e^\alpha(d(x, x_i, H_i^\alpha)) < 1 \\ k_e^\beta\left(||(x - x_i)/h^\beta(H_i^\alpha)||^2\right) < 1 \end{cases} \tag{23}$$

The bandwidth matrix can be decomposed to

$$H_i^\alpha = \lambda V A V^T \tag{24}$$

where $\lambda$ is a scalar, $V$ is a matrix of normalised eigenvectors, and $A$ is a diagonal matrix of eigenvalues whose diagonal elements $a_i$ satisfy $\prod_{i=1}^{p} a_i = 1$. The bandwidth matrix is updated by adding more and more points to the: if these points are similar in intensity or colour, then the Mahalanobis distance will be consistently reduced. Otherwise, if the Mahalanobis distance is increased, these points will not be considered in the computation.

Anisotropic mean shift based FCM (AMSFCM) proceeds in the following steps:

Step 1: Initialise the cluster centres $c_i$. Let $j = 0$.
Step 2: Initialise the fuzzy partitions $\mu_{ij}$ using Equation (8).
Step 3: Set $j = j + 1$ and compute $c_i$ using Equation (9) for all clusters.
Step 4: Update $\mu_{ij}$ using Equation (8).
Step 5: For each pixel $x_i$ determine anisotropic kernel and related colour radius using Equations (22) and (24). Note that mean shift is applied to the outcome image of FCM.
Step 6: Calculate the mean shift vector and then iterate until the mean shift, $M^+(x_i) - M^-(x_i)$, is less than a pixel considering the previous position and a normalised position change:
$M^+(x_i) = \nu M^-(x_i) + (1-\nu) \frac{\sum_{j=1}^{N}(x_j - M^-(x_i))||(M^-(x_i^\beta) - x_j^\beta)/(h^\beta H_j^\alpha)||^2}{\sum_{j=1}^{N}||(M^-(x_i^\beta) - x_j^\beta)/(h^\beta H_j^\alpha)||^2}$
with $\nu = 0.5$.
Step 7: Merge pixels with similar colour.
Step 8: Repeat Steps 3 to 6 until convergence.

## 4 Rough Clustering for Colour Palette Design

(Lingras and West, 2004) introduced a rough set inspired clustering algorithm based on the well known c-means algorithm. In their rough c-means approach, each cluster $c_k$ is described not only by its centre $m_k$, but also contains additional information, in particular its lower approximation $\underline{c_k}$, its upper approximation $\overline{c_k}$, and its boundary area $c_k^b = \overline{c_k} - \underline{c_k}$. Lingras *et al.*'s algorithm proceeds in the following steps:

Step 1: *Initialisation:* Each data sample is randomly assigned to one lower approximation. As the lower approximation of a cluster is a subset of its upper approximation, this also automatically assigns the sample to the upper approximation of the same cluster.
Step 2: *Cluster centre calculation:* The cluster centres are updated as

$$m_k = \begin{cases} \omega_l \sum_{x_i \in \underline{c_k}} \frac{x_i}{|\underline{c_k}|} + \omega_b \sum_{x_i \in c_k^b} \frac{x_i}{|c_k^b|} & \text{if } c_k^b \neq \{\} \\ \omega_l \sum_{x_i \in \underline{c_k}} \frac{x_i}{|\underline{c_k}|} & \text{otherwise} \end{cases} \tag{25}$$

The cluster centres are hence determined as a weighted average of the samples belonging to the lower approximation and the boundary area, where the weights $\omega_l$ and $\omega_b$ define the relative importance of the two sets.

Step 3: *Sample assignment:* For each data sample the closest cluster centre is determined and the sample assigned to its upper approximation. Then, all clusters that are at most $\epsilon$ further away than the closest cluster are determined. If such clusters exist, the sample will also be assigned to their upper approximations. If no such cluster exist, the sample is assigned also to the lower approximation of the closest cluster.

Step 4: *Termination:* If the algorithm has converged (i.e., if the cluster centres do not change any more, or after a pre-set number of iterations), terminate, otherwise go to Step 2.

Strictly speaking, this algorithm does not implement all properties set out for rough sets (Pawlak, 1982), and hence belongs to the reduced interpretation of rough sets as lower and upper approximations of data (Yao et al., 1994).

(Peters, 2006) pointed out some potential pitfalls of the algorithm in terms of objective function and numerical stability, and suggested some improvements to overcome these. Equation (25) is revised to

$$m_k = \omega_l \sum_{x_i \in \underline{c_k}} \frac{x_i}{|\underline{c_k}|} + \omega_u \sum_{x_i \in \overline{c_k}} \frac{x_i}{|\overline{c_k}|} \tag{26}$$

with $\omega_l + \omega_u = 1$, i.e. as a convex combination of lower and upper approximation means. In order to overcome the possibility of situations with empty lower approximations, Peters suggests two possible ways of addressing this, either by modifying the calculation of cluster centres so that for empty lower approximations the cluster centre is calculated as the average of samples in the upper approximation, or by ensuring that each lower approximation has at least one member. In our approach we choose the latter by assigning the data sample closest to the cluster centre to its lower approximation.

In addition, we perform a different initialisation procedure than Lingras *et al.* and Peters. Rather than randomly assigning samples to clusters, we generate random cluster centres first and then proceed with Steps 3, 2 and 4 (i.e., steps 2 and 3 reversed) of the algorithm in order to arrive at a good colour palette (Schaefer et al., 2009).

## 5 Experimental Results

In order to evaluate the various colour quantisation algorithms, we have taken a set of six standard images commonly used in the colour quantisation literature (*Lenna, Peppers, Mandrill, Sailboat, Airplane, and Pool*) and applied all seven discussed algorithms, that is Simulated annealing, Hybrid simulated annealing, Fuzzy c-means, RSFCM, EnFCM, AMSFCM, and Rough c-means, to generate quantised images with a palette of 16 colours.

To put the results we obtain into context we have also implemented four popular colour quantisation algorithms (which are often integrated in typical image processing software) to generate corresponding quantised images with palette size 16. The algorithms we have tested were: Popularity algorithm (Heckbert, 1982), Median cut quantisation (Heckbert, 1982), Octree quantisation (Gervautz and Purgathofer, 1990), and Neuquant (Dekker, 1994). For all algorithms, pixels in the quantised images were assigned to their nearest neighbours in the colour palette to provide the best possible image quality.

The results are listed in Table 1, expressed in terms of average (over 10 runs of the algorithms) peak-signal-to-noise-ratio (PSNR) defined as

$$\mathrm{PSNR}(O,Q) = 10\log_{10}\frac{255^2}{\mathrm{MSE}(I_1,I_2)} \tag{27}$$

with $\mathrm{MSE}(O,Q)$ calculated as in Equation (1).

**Table 1** Quantisation results, given in terms of PSNR [dB].

| | Lenna | Peppers | Mandrill | Sailboat | Pool | Airplane | average |
|---|---|---|---|---|---|---|---|
| Popularity algorithm | 22.24 | 18.56 | 18.00 | 8.73 | 19.87 | 15.91 | 17.22 |
| Median cut | 23.79 | 24.10 | 21.52 | 22.01 | 24.57 | 24.32 | 23.39 |
| Octree | 27.45 | 25.80 | 24.21 | 26.04 | 29.39 | 28.77 | 26.94 |
| Neuquant | 27.82 | 26.04 | 24.59 | 26.81 | 27.08 | 28.24 | 26.73 |
| SWASA | 27.79 | 26.16 | 24.46 | 26.69 | 29.84 | 29.43 | 27.40 |
| Hybrid SWASA | 29.70 | 27.17 | 25.37 | 27.95 | 31.57 | 32.94 | 28.97 |
| FCM | 28.81 | 26.77 | 25.03 | 27.25 | 31.03 | 30.23 | 28.17 |
| RSFCM | 28.70 | 26.70 | 24.98 | 27.32 | 30.81 | 30.73 | 28.20 |
| EnFCM | 28.61 | 26.74 | 24.87 | 27.22 | 31.11 | 29.92 | 28.08 |
| AMSFCM | 28.63 | 26.71 | 24.66 | 27.24 | 30.87 | 29.96 | 28.01 |
| Rough c-means | 28.63 | 26.67 | 25.02 | 27.62 | 29.40 | 30.50 | 27.98 |

As can be seen from Table 1, all the algorithms presented in this chapter provide very good results and clearly outperform standard colour quantisation algorithms. The results of the different fuzzy clustering approaches are fairly similar which suggests that the computationally more efficient versions (RSFCM, EnFCM, AMSFCM) can be employed without sacrificing image quality. Also, the rough set approach gives similar performance and the presented rough colour quantisation approach hence adds to the applications of rough sets in the field of imaging and vision. The best results are achieved by the hybrid SWASA approach which demonstrates that generic optimisation algorithms can provide a powerful tool for colour quantisation and that the further adjustment through application of a subsequent clustering step does indeed improve image quality significantly.

## 6 Conclusions

In this chapter we have presented several successful colour quantisation approaches based on computational intelligence principles. We have shown that generic optimisation approaches can be used for deriving a good colour palette and that hybridisation with a clustering algorithm can further improve the performance. We have furthermore demonstrated that several clustering approaches based on fuzzy and rough set concepts can also be used effectively for colour quantisation. All techniques presented in this chapter were compared against standard colour quantisation methods and were shown to clearly outperform these.

## Acknowledgements

The author would like to acknowledge Dr Huiyu Zhou of Queen's University Belfast and Dr Lars Nolle of Nottingham Trent University for their involvement in the algorithms presented in this chapter.

## References

Ahmed, M., Yamany, S., Mohamed, N., Farag, A., Moriaty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Trans. Medical Imaging 21, 193–199 (2002)

Bezdek, J.: A convergence theorem for the fuzzy ISODATA clustering algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence 2, 1–8 (1980)

Cai, W., Chen, S., Zhang, D.: Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Pattern Recognition 40(3), 825–838 (2007)

Cheng, T., Goldgof, D., Hall, L.: Fast fuzzy clustering. Fuzzy Sets and Systems 93, 49–56 (1998)

Chuang, K., Tzeng, S., Chen, H., Wu, J., Chen, T.: Fuzzy c-means clustering with spatial information for image segmentation. Computerized Medical Imaging and Graphics 30, 9–15 (2006)

Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 603–619 (2002)

Davidor, Y.: Epistasis variance: Suitability of a representation to genetic algorithms. Complex Systems 4, 369–383 (1990)

Dekker, A.H.: Kohonen neural networks for optimal colour quantization. Network: Computation in Neural Systems 5, 351–367 (1994)

Gervautz, M., Purgathofer, W.: A simple method for color quantization: Octree quantization. In: Glassner, A.S. (ed.) Graphics Gems, pp. 287–293 (1990)

Heckbert, P.S.: Color image quantization for frame buffer display. ACM Computer Graphics (ACM SIGGRAPH 1982 Proceedings) 16(3), 297–307 (1982)

Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)

Hu, R., Hathaway, L.: On efficiency of optimization in fuzzy c-means. Neural, Parallel and Scientific Computation 10, 141–156 (2002)

Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE Int. Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)

Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. Communications 28, 84–95 (1980)

Lingras, P., West, C.: Interval set clustering of web users with rough k-means. Journal Intell. Inform. Syst. 23, 5–16 (2004)

Nolle, L.: On the effect of step width selection schemes on the performance of stochastic local search strategies. In: 18th European Simulation Multi-Conference, pp. 149–153 (2004)

Nolle, L., Armstrong, D.A., Hopgood, A.A., Ware, J.A.: Simulated annealing and genetic algorithms applied to finishing mill optimisation for hot rolling of wide steel strip. International Journal of Knowledge-Based Intelligent Engineering Systems 6(2), 104–111 (2002)

Nolle, L., Schaefer, G.: Color map design through optimization. Engineering Optimization 39(3), 327–343 (2007)

Omran, M., Engelbrecht, A., Salman, A.: A color image quantization algorithm based on particle swarm optimization. Informatica 29, 263–271 (2005)

Pawlak, Z.: Rough sets. Int. Journal Inform. Comput. Sci. 11, 145–172 (1982)

Peters, G.: Some refinements of rough k-means clustering. Pattern Recognition 39, 1481–1491 (2006)

Schaefer, G., Zhou, H.: Fuzzy clustering for colour reduction in images. Telecommunication Systems 40(1-2), 17–25 (2009)

Schaefer, G., Zhou, H., Hu, Q., Hassanien, A.E.: Rough image colour quantisation. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS (LNAI), vol. 5908, pp. 217–222. Springer, Heidelberg (2009)

Scheunders, P.: A genetic c-means clustering algorithm applied to color image quantization. Pattern Recognition 30(6), 859–866 (1997)

Szilagyi, L., Benyo, Z., Szilagyii, S.M., Adam, H.S.: MR brain image segmentation using an enhanced fuzzy c-means algorithm. In: 25th IEEE Int. Conference on Engineering in Medicine and Biology, vol. 1, pp. 724–726 (2003)

Wang, J., Thiesson, B., Xu, Y., Cohen, M.: Image and video segmentation by anisotropic kernel mean shift. In: 8th European Conference on Computer Vision, pp. 238–249 (2004)

Yao, Y.Y., Li, X., Lin, T.Y., Liu, Q.: Representation and classification of rough set models. In: 3rd Int. Workshop on Rough Sets and Soft Computing, pp. 630–637 (1994)

# Chapter 13
# Mean Shift and Its Application in Image Segmentation

Huiyu Zhou, Xun Wang, and Gerald Schaefer

**Abstract.** Mean shift techniques have been demonstrated to be capable of estimating the local density gradients of similar image pixels. These gradient estimates are iteratively performed so that for all pixels similar pixels in corresponding images can be identified. In this chapter, we show how the application of a mean shift process can lead to improved image segmentation performance. We present several mean shift-based segmentation algorithms and demonstrate their superior performance against the classical approaches. Conclusions are drawn with respect to the effectiveness, efficiency and robustness of image segmentation using these approaches.

**Keywords:** Image segmentation, mean shift, fuzzy c-means, Dirichlet process mixture, gradient vector flow snake.

## 1 Introduction

Image segmentation is the process of categorising the intensity/colour and/or texture of an image into different regions where each region attains homogeneity, and is a key stage in many image analysis and pattern recognition applications including

Huiyu Zhou
Queen's University Belfast
Belfast, United Kingdom
e-mail: h.zhou@ecit.qub.ac.uk

Xun Wang
Yucheng Technological Limited
Beijing, China
e-mail: wangxun@yuchengtech.com

Gerald Schaefer
Loughborough University
Loughborough, United Kingdom
e-mail: gerald.schaefer@ieee.org

H. Kwaśnicka & L.C. Jain (Eds.): Innovations in Intell. Image Analysis, SCI 339, pp. 291–312.
springerlink.com

object tracking and recognition, image retrieval, and volumetric reconstruction. Image segmentation is a long standing problem in computer vision. A major problem in image segmentation is the appropriate determination of thresholds to be used to separate various regions or intensity/colour levels.

Established image segmentation algorithms can be generally classified into three major categories: feature-based clustering, spatial segmentation, and graph-based approaches (Tao et al., 2007). Feature-based clustering approaches use the characteristics of the image through extraction and selection schemes (Jacobs et al., 2000). Using a determined distance measure, which intentionally ignores the spatial information in an image, features are shaped as vectors that are then grouped into various clusters (Duda et al., 2000). Spatial segmentation methods are referred to as region-based when derived from region entities. For example, watershed algorithms (e.g. (Vincent and Soille, 1991)) can be considered as an extensive methodology, which may or may not apply merging algorithms for formation of quasi-homogeneous regions (Makrogiannis et al., 2005). Graph-based approaches are usually regarded as image perceptual grouping and organisation methods, based on the fusion of the feature and spatial information (e.g. (Morris et al., 1986)). In these approaches, the visual group is based on several key components such as similarity, proximity, and continuation (Caselles et al., 1996).

Despite the successes achieved in image segmentation, the developed techniques can be further improved in terms of segmentation accuracy and automation. For this purpose, people have considered supervised and unsupervised models according to human intervention (Pham et al., 2000). The former comes up with manual class labels for image pixels or regions. These labels are capable of providing a predefined interpretation and hence achieve higher accuracy. Unfortunately, supervised models sacrifice work efficiency as they require a large amount of labeling practice. Unsupervised models usually lack a proper map from labelled areas to meaningful object classes as no semantical annotation information is available.

Supervised models consist of $k$-nearest-neighbour ($k$NN), Bayes or other classifiers. $k$NN is an approach of classifying image pixels according to the majority vote of the $k$ closest data items, while Bayes classifiers assign each image pixel to the class with the highest probability. One of the weaknesses of supervised models is the lack of spatial modelling (Wells et al., 1995). Another drawback is the need of manual labelling. Unsupervised models include $k$-means or ISODATA algorithms (Venkateswarlu and Raju, 1992), fuzzy c-means (Zhou et al., 2009b), and expectation-maximisation (EM) algorithms (Belongie et al., 1998). $k$-means approaches classify each pixel to the class with the closest mean; fuzzy c-means is a generalised version of $k$-means by allowing soft segmentation using fuzzy set principles. EM algorithms assign pixels to the class the pixel shares the closest mean/covariance with.

Mean shift is a method to cluster an image by associating each pixel with a peak of the image's probability density. This peak is computed by first defining a window in the neighbourhood of the pixel and then calculating the mean of the pixel that lie within the window. The window is then shifted to the mean, and similar steps are repeated until convergence. The outcome of mean shift is only controlled by the kernel size (bandwidth) and therefore requires less manual intervention compared to other algorithms. However, determining an appropriate bandwidth is not an easy task. Too large or small bandwidths may lead to over- or under-segmentation. To effectively handle this problem, one can integrate different algorithms with mean shift in order to find an optimal solution.

In this chapter, we present three segmentation algorithms, which incorporate a mean shift process in some state-of-the-art segmentation techniques. These algorithms are motivated by the fact that these combinations will improve the performance of the individual approaches in general circumstances.

## 2 Mean Shift

First, we briefly review the principle of mean shift. Given an image point sequence $\mathrm{s}_i$ $(i = 1, 2, ..., n)$ in the $m$-dimensional space $R^m$, the multivariate kernel density estimate with kernel $K(\mathrm{s})$ and window radius $r$ is given as

$$F(\mathrm{s}) = \frac{1}{nr^m} \sum_{i=1}^{n} K\Big(\frac{\mathrm{s} - \mathrm{s}_i}{r}\Big). \tag{1}$$

The multivariate *Epanechnikov* kernel can be estimated by

$$K_E(\mathrm{s}) = \begin{cases} \frac{(m+2)(1-\|\mathrm{s}\|^2)}{2c_m}, \| \mathrm{s} \| < 1 \\ 0, otherwise \end{cases} \tag{2}$$

where $c_m$ is the volume of the unit $m$-dimensional sphere.

Assuming a kernel $\Psi(\mathrm{s}) = c_0\psi(\| \mathrm{s} \|^2)$, where $c_0$ is a normalisation constant, the mean shift vector is expressed as

$$MS(\mathrm{s}) \equiv \frac{\sum_{i=1}^{n} \mathrm{s}_i \psi(\| (\mathrm{s} - \mathrm{s}_i)/r \|^2)}{\sum_{i=1}^{n} \psi(\| (\mathrm{s} - \mathrm{s}_i)/r \|^2)} - \mathrm{s}, \tag{3}$$

where $\psi(\cdot)$ is an intermediate function (Comaniciu et al., 2000). The mean shift procedure is a recursive evolution by computing the mean shift vector $MS(\mathrm{s})$ and adjusting the centroid of kernel $\Psi$ by $MS(\mathrm{s})$. In theory, the Euclidean distance between the centroids $d$ is proportional to the mean of the mean shift:

$$d \propto MS(\mathrm{s})^m. \tag{4}$$

## 3 Fuzzy C-Means with Mean Shift

Fuzzy c-means (FCM) is based on the idea of finding cluster centres by iteratively adjusting their positions and the evaluation of an objective function is similar to the original hard c-means, yet it allows more flexibility by introducing the possibility of partial memberships to clusters (Bezdek, 1980). The objective function usually follows the form

$$E = \sum_{j=1}^{C} \sum_{i=1}^{N} \mu_{ij}^{k} ||x_i - c_j||^2, \tag{5}$$

where $\mu_{ij}^{k}$ is the fuzzy membership of sample (or pixel) $x_i$ and the cluster identified by its centre $c_j$, and $k$ is a constant that defines the fuzziness of the resulting partitions.

$E$ can reach the global minimum when pixels nearby the centroid of corresponding clusters are assigned higher membership values, while lower membership values are assigned to pixels far from the centroid (Chuang et al., 2006). Here, the membership is proportional to the probability that a pixel belongs to a specific cluster where the probability is only dependent on the distance between the image pixel and each independent cluster centre. The membership functions and the cluster centres are updated by

$$\mu_{ij} = \frac{1}{\sum_{m=1}^{C} \left( \frac{||x_j - c_i||}{||x_j - c_m||)^{2/(k-1)}} \right)}, \tag{6}$$

and

$$c_i = \frac{\sum_{j=1}^{N} \mu_{ij}^{k} x_j}{\sum_{j=1}^{N} \mu_{ij}^{k}}. \tag{7}$$

## 3.1 Anisotropic Mean Shift

To further enhance the performance of classical FCM algorithms, it is possible to derive an anisotropic mean shift algorithm coupled with fuzzy segmentation (Zhou et al., 2009b). In mean shift algorithms the image clusters are continuously moved along the gradient of the density function before they become stationary. Those points gathering in an outlined area are treated as the members of the same segment. To determine the membership of an image point, a density estimate at the point needs to be conducted. In other words, similarity computation must be achieved between this point and the centre of the segment. Furthermore, the coherence between this point and its surrounding image points needs to be discovered (e.g. colour or intensity consistency), as this coherence can be used to remove any inconsistency such as image artefacts or noise. In this section, we mainly discuss the estimation of the density function of an image point (this kernel density estimation is also known as the Parzen window technique).

The motivation for introducing density estimation based segmentation is that the image space can be represented by empirical probability density functions (PDFs) of certain parameters (e.g. colour or intensity). Dense or sparse regions of similar image points correspond to local maxima or minima of the PDF (or the modes of the unknown density) (Comaniciu and Meer, 2002). After the modes have been located in the image, the membership of an image point to a particular segment will be determined.

A kernel density estimate on an image point is defined by

$$\tilde{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K(x - x_i), \tag{8}$$

with

$$K(x) = |H|^{-1/2} K(H^{-1/2}x), \tag{9}$$

where $N$ is the number of samples, and $x_i$ stands for a sample from an unknown density function $f$. $K(\cdot)$ is the $d$-variate kernel function with compact support satisfying the regularity constraints, and $H$ is a symmetric positive definite $d \times d$ bandwidth matrix. Usually, we have $K(x) = k_e(x)$, where $k_e(x)$ is a convex decreasing function, e.g. for a Gaussian kernel

$$k_e(x) = c_t e^{-x/2}, \tag{10}$$

or for an Epanechnikov kernel,

$$k_e(x) = c_t \max(1 - x, 0), \tag{11}$$

where $c_t$ is a normalising constant.

If a single global spherical bandwidth is applied, $H = h^2\mathbf{I}$ (where $\mathbf{I}$ is the identity matrix), then we have the classical form as

$$\tilde{f}(x) = \frac{1}{Nh^d}\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right). \tag{12}$$

Since the kernel can be divided into two different radially symmetric kernels, we have the kernel density estimate as

$$\begin{aligned}\tilde{f}(x;\mathbf{c}) = \sum_{i=1}^{N} \frac{1}{N(h^\alpha)^p(h^\beta)^q} k^\alpha\Big(||(\mathbf{c}^\alpha - x_i^\alpha)/h^\alpha||^2\Big) \\ k^\beta\Big(||(\mathbf{c}^\beta - x_i^\beta)/h^\beta||^2\Big),\end{aligned} \tag{13}$$

where $\mathbf{c}$ represents a vector of cluster centres, $p$ and $q$ are two ratios, and $\alpha$ and $\beta$ denote the spatial and temporal components respectively (Wang et al., 2004). Classical mean shift utilises symmetric kernels that may experience a lack of temporal coherence in regions where intensity gradients exist with a slope relative to the evolving segment. In contrast, anisotropic kernel mean shift links with every data point by an anisotropic kernel. This kernel associated with a pixel can update its shape, scale and orientation. The density estimator is represented by

$$\tilde{f}(x;\mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h^{\beta}(H_i^{\alpha})^{q}}k^{\alpha}(d(\mathbf{c}^{\alpha},x_i^{\alpha},H_i^{\alpha})) \\ k^{\beta}\left(||(\mathbf{c}^{\beta}-x_i^{\beta})/(h^{\beta}H_i^{\alpha})||^{2}\right), \tag{14}$$

where $d(c_i^{\alpha}, x_i^{\alpha}, H_i^{\alpha})$ is the Mahalanobis distance

$$d(\mathbf{c}^{\alpha},x_i^{\alpha},H_i^{\alpha}) = (x_i^{\alpha}-\mathbf{c}^{\alpha})^{T}H_i^{\alpha-1}(x_i^{\alpha}-\mathbf{c}^{\alpha}). \tag{15}$$

Anisotropic mean shift is intended to modulate the kernels during the mean shift procedure. The objective is to keep reducing the Mahalanobis distance so as to group similar samples as much as possible. First, the anisotropic bandwidth matrix $H_i^{\alpha}$ is estimated using a standard radially symmetric diagonal $H_i^{\alpha}$ and $h^{\beta}$. The neighbourhood of pixels around $\mathbf{c}$ has the following constraints:

$$\begin{cases} k_e^{\alpha}(d(\mathbf{c},x_i,H_i^{\alpha})) < 1 \\ k_e^{\beta}\left(||(\mathbf{c}-x_i)/(h^{\beta}H_i^{\alpha})||^{2}\right) < 1 \end{cases} \tag{16}$$

A new full matrix $\bar{H}_i^{\alpha}$ will use the variance of $(\mathbf{c}-x_i)$ as its components. To show how the modulation of $\bar{H}_i^{\alpha}$ happens we first decompose the required bandwidth matrix to

$$\bar{H}_i^{\alpha} = \lambda V A V^{T}, \tag{17}$$

where $\lambda$ is a scalar, $V$ is a matrix of normalised eigenvectors, and $A$ is a diagonal matrix of eigenvalues whose diagonal elements $a_i$ satisfy (Wang et al., 2004)

$$\prod_{i=1}^{p} a_i = 1. \tag{18}$$

The bandwidth matrix is updated by adding more and more points to the computational list: the more image points with similar colour or intensity gather in the same segments, the smaller the total Mahalanobis distance between the image points and the centres of individual segments.

## 3.2 Anisotropic Mean Shift Based FCM

In the combined algorithm, fuzzy c-means and anisotropic mean shift segmentation are integrated into one framework. A significant difference between this approach and other similar methods is that our algorithm continuously inherits and updates the states, based on the interaction of FCM and mean shift. Stemming from the algorithm reported in (Wang et al., 2004), this anisotropic mean shift based FCM (AMSFCM) (Zhou et al., 2009b) proceeds in the following steps:

Step 1: Initialise the cluster centres $c_j$. Let the iteration count $t = 0$.
Step 2: Initialise the fuzzy partitions $\mu_{ij}$ using Eq. (6).
Step 3: Increment $t = t + 1$ and compute $c_j$ using Eq. (7) for all clusters.

Step 4: Update $\mu_{ij}$ using Eq. (6). This is an FCM process.

Step 5: For each pixel $x_i$ one needs to estimate the density with anisotropic kernels and related colour radius using Eqs. (14)-(18). For simplicity, $\bar{H}_i^{\alpha}$ can apply variances at the diagonal items with other zero components. Note that mean shift is employed after the FCM stage.

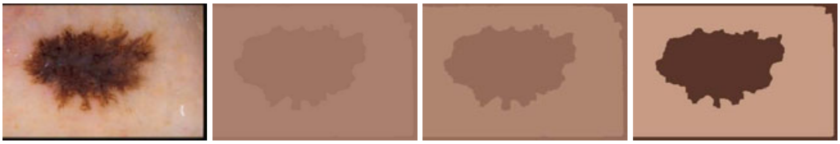
Step 6: Calculate the mean shift vector and then iterate until the mean shift, $M^{+}(x_i) - M^{-}(x_i)$, is less than 0.01 considering the previous position and a normalised position change:
$M^{+}(x_i) = \nu M^{-}(x_i) + (1-\nu)\frac{\sum_{j=1}^{N}(x_j - M^{-}(x_i))||(M^{-}(x_i^{\beta}) - x_j^{\beta})/(h^{\beta}H_j^{\alpha})||^2}{\sum_{j=1}^{N}||(M^{-}(x_i^{\beta}) - x_j^{\beta})/(h^{\beta}H_j^{\alpha})||^2}$
with $\nu = 0.5$.

Step 7: Merge pixels with Mahalanobis distances below the pre-defined threshold.

Step 8: Repeat Steps 3 to 7 until $|\mu_{ij}^{t} - \mu_{ij}^{t-1}| < \epsilon_0$ ($\epsilon_0$ is a pre-set threshold).

Note, that the number of clusters has been assigned before the optimisation. An alternative would be to apply mean shift clustering to find out the number of clusters, and afterwards deploy the above algorithm.

Figure 1 illustrates how the segmentation evolves using the proposed AMSFCM algorithm. In this example, the segmentation optimally converges after 6 iterations.



(a) original image (b) 3 iterations (c) 4 iterations (d) 6 iterations

**Fig. 1** Examples of AMSFCM iterative segmentation.

## 4 Dirichlet Process Mixture Models with Mean Shift

Assume that image segmentation aims to group a set $\mathbf{x}_1$, ..., $\mathbf{x}_i$ of inputs (local image features) into individual classes. Let's denote the number of classes by $N_C$, then the class assignment of input $\mathbf{x}_i$ is represented by an indicator $S_i$ ($i$ = 1,...,$N_C$). A class in image clustering will be of a form with finite mixture models

$$p(\mathbf{x}) = \sum_{j=1}^{N_C} c_j p_j(\mathbf{x}), \tag{19}$$

where $c_j$ = Pr$\{S = j\}$ for an input drawn randomly from the entire model, and $\sum_j c_j = 1$, and $p_j$ indicates a single probability distribution. This model may have a two-stage generative process for the input $\mathbf{x}$: $\mathbf{x} \sim p_S$, and $S \sim (c_1, ..., c_{N_C})$. If the distribution $p_j(\mathbf{x})$ can be parameterised as $p_j(\mathbf{x}) = p_j(\mathbf{x}|\theta_j)$, then we can parameterise the distribution shown in Eq. (19) with $c_1, ..., c_{N_C}$ and $\theta_1, ..., \theta_{N_C}$ ($\theta$ is a parameter value of the clusters).

MDP models draw a random prior $G$ from a Dirichlet Process (DP). Combining it with a parametric likelihood $F(\mathbf{x}|\theta)$, this results in the following form:

$$\begin{aligned} \mathbf{x}_i &\sim F(\mathbf{x}|\theta_i), \\ \theta_i &\sim G, \\ G &\sim DP(\alpha G_0), \end{aligned} \tag{20}$$

where $\alpha$ is a scalar and $G_0$ is the base measure of the process. In a Bayesian framework, $F(\cdot|\cdot)$ is considered to be the posterior estimation of the data. Prior and posterior in Eq. (20) refer to the same model class and the posterior parameters are dynamically updated during the process. A conditional distribution for property 3 of Eq. (20) can be computed as (Orbanz and Buhmann, 2008)

$$p(\theta_{n+1}|\theta_1, ..., \theta_n) = \frac{1}{n+\alpha} \sum_{i=1}^{n} \varDelta_{\theta_i}(\theta_{n+1}) + \frac{\alpha}{n+\alpha} G_0(\theta_{n+1}), \tag{21}$$

where $\varDelta_{\theta_i}$ is the Dirac measure centered at $\theta_i$.

## *4.1 Dirichlet Process Mixtures with Markov Random Fields*

Markov random fields (MRFs) are stochastic models that characterise local spatial interactions in data. MRFs are used with MDP in order to enforce spatial constraints during the segmentation process. An MRF consists of random variables defining an undirected and weighted graph with sites, edges, and weights.

Let an MRF distribution $\varXi$ be decomposed into a site-wise term $\mathcal{S}$ and the remaining interaction term $\mathcal{R}$. The MRF distribution can be written as follows to generate MRF of $\mathbf{x}_i$

$$\varXi \propto \mathcal{S}(\theta_1, ..., \theta_n)\mathcal{R}(\theta_1, ..., \theta_n), \tag{22}$$

with the site $\mathcal{S}(\theta_1, ..., \theta_n) := \frac{1}{Z_S} \exp(-\sum_i H_i(\theta_i))$ and the remaining term $\mathcal{R}(\theta_1, ..., \theta_n) := \frac{1}{Z_{\mathcal{R}}} \exp(-\sum_{C \in C_2} H_C(\theta_C))$, where $Z_{\mathcal{R}}$ indicates the partition function, singleton cliques $C = \{i\}$ and $C_2 := \{C \in \mathcal{C} || C| \geq 2\}$ ($\mathcal{C}$ is the set of all cliques), and $H(\sim)$ is a cost function that defines a distribution by $\varXi(\theta_1, ..., \theta_n) := \frac{1}{Z_H} \exp(-H(\theta_1, ..., \theta_n))$ with a normalisation term $Z_H$.

Omitting intermediate steps, we arrive at the following form for the posterior estimation (Orbanz and Buhmann, 2008):

$$\varXi(\theta_i|\theta_{-i}) \propto \sum_{k=1}^{N_C} \mathcal{R}(\theta_i|\theta_{-1}) n_k^{-i} \delta_{\theta_k^\star}(\theta_i) + \frac{\alpha}{Z_H} G_0(\theta_i), \tag{23}$$

where $\theta_{-i} := \{\theta_1, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_n\}$, $n_k^{-i}$ is the number of values accumulated in group $k$, $\delta$ is the Kronecker symbol, and $\theta_k^\star$ denotes the parameters in class $k$.

## 4.2 MDP/MRF with Mean Shift

A Gibbs sampler can be used for implementing MDP models (see Eq. (20)). There are two steps in this sampling algorithm: an assignment step and a parameter update step. The assignment of inputs $\mathbf{x}_i$ to cluster $k$ relies only on the current state of the model. If $\mathbf{x}_i$ falls in the known classes, then $\mathbf{x}_i$ is assigned to cluster $k$. Otherwise, a new cluster will be created. Image segmentation is a clustering problem where two class labels can only be identical or different. We have

$$H(\theta_i|\theta_{-i}) = \frac{1}{n}\sum_{j=1}^{n} K_H(\theta_i - \theta_j), \tag{24}$$

where $K_H(\theta) = |H|^{-1/2}K(H^{-1/2}\theta)$ and $K$ is the $d$-variate kernel function. Assuming $H = h^2\mathbf{I}$ ($h^2$ refers to variance estimates and $\mathbf{I}$ is identity matrix), we have

$$H(\theta_i|\theta_{-i}) = \frac{1}{nh^d}\sum_{j=1}^{n} K_H\Big(\frac{\theta_i - \theta_j}{h}\Big), \tag{25}$$

which indicates that image segmentation can be reached subject to the measurements by the mean of the square error between the density and the estimate. Using the established mean shift algorithm (Comaniciu and Meer, 2002), we iteratively calculate the difference between the mean of the cluster centres and image pixels until this difference is less than a pre-defined threshold:

$$m(\theta_i) = \frac{\sum_{j=1}^{n}\theta_j g\Big(\parallel \frac{\theta_i-\theta_j}{h} \parallel^2\Big)}{\sum_{j=1}^{n} g\Big(\parallel \frac{\theta_i-\theta_j}{h} \parallel^2\Big)} - \theta_i. \tag{26}$$

Given an image, we extract features by formulating histogram bins from the image. Each histogram is described by a vector $\mathbf{r}_i$ = ($r_{i1}$,...,$r_{iN}$), where $N$ is the number of the bins.

The initial class probability $q_{i0}$ can be represented as (Orbanz and Buhmann, 2008)

$$q_{i0} \propto \int_{\Omega_\theta} F(\mathbf{r}_i|\theta_i)G_0(\theta_i)d\theta_i = \frac{D_G(\mathbf{r}_i + \beta\pi)}{D_F(\mathbf{r}_i)D_G(\beta\pi)}, \tag{27}$$

where $D_F$ is the multinomial partition function and $D_G$ is a normalisation term.

Let $k$ = 1,...,$N_C$, then we have the class probability

$$\begin{aligned} q_{ik} &\propto n_k^{-i}\exp(-H(\theta_k^\star|\theta_{-i}))F(\mathbf{r}_i|\theta_k^\star) \\ &= \frac{n_k^{-i}}{D_F(\mathbf{r}_i)}\exp\Big(\frac{1}{nh^d}\sum_{j=1}^{n} K_H\Big(\frac{\theta_i-\theta_j}{h}\Big) + \sum_j h_{ij}\log(\theta_{kj}^\star)\Big). \end{aligned} \tag{28}$$

Therefore, a mean shift based MDP/MRF segmentation algorithm (Zhou et al., 2009a) and can be implemented as follows:

Step 1: Initiate the algorithm with a single cluster $\theta_1^\star$.
Step 2: Generate random samples from data indices.
Step 3: Compute cluster probabilities $q_{i0}$ and $q_{ik}$ using Eqs. (27) and (28).
Step 4: Assign observations $\mathbf{x}_i$ to cluster $k$ based on the value of $k$.
Step 5: Update cluster parameters $\theta_k^\star$ by sampling $\theta_k^\star \sim G_0(\theta_k^\star) \prod_i F(\mathbf{x}_i|\theta_k^\star)$.

## 5 Gradient Vector Flow Snakes with Mean Shift

Snake (active contour) algorithms are used to detect object boundaries or edges given an initial guess of the evolving contours. The classical snake model considers a combination of internal and external energy, where the boundary will stop evolving based on a compromise of the two energies. In general, the internal energy term maintains smoothness and compactness of the curve shape, while the external energy term tunes the curve in order to be consistent with the intrinsic image gradients. Often, the negative of the image gradient magnitude is used as the external energy. Hence, larger gradient magnitudes will drive the evolution of the contour towards the real object boundary (Witkin et al., 1987).

The external energy force in the snake model is restricted to a small area next to the real boundary. If it is far from the real boundary, the snake will be less likely to converge to the correct position. (Xu and Prince, 1998) proposed a GVF map to represent the external energy force in the snake model. This GVF term is sensitive to object boundaries or edges appearing in the image and hence effectively pulls the snake towards the real edges.

Let a snake be a curve $\mathbf{x}(s) = [x(s), y(s)], s \in [0, 1]$, which evolves in an image domain to reach a minimisation of the energy function

$$E(\mathbf{x}) = \int_0^1 \Big[\frac{1}{2}\Big(\alpha\Big|\frac{\partial x}{\partial s}\Big|^2 + \beta\Big|\frac{\partial^2 x}{\partial s^2}\Big|^2\Big) + E_{ext}(\mathbf{x})\Big] ds, \tag{29}$$

where $\alpha$ and $\beta$ are weights that dominate the tension and rigidity of the snake respectively. The first order derivative $\frac{\partial \mathbf{x}}{\partial s}$ encourages stretching while the second order derivative $\frac{\partial^2 \mathbf{x}}{\partial s^2}$ leads to bending. The first two terms on the right-hand side of Eq. (29) describe the internal energy of the snake, while the third term is the external energy. In the presence of high gradients at image boundaries (e.g. step edges) the external energy is represented by $- \bigtriangledown (G_\sigma(x, y) * I(x, y))^2$. In the case of line drawings, $\pm G_\sigma(x, y) * I(x, y)$ is used instead, where $G_\sigma$ is a two-dimensional Gaussian function with standard deviation $\sigma$.

When the GVF contour (Xu and Prince, 1998) has converged, i.e. when internal and external forces are balanced, one can have the Euler equation, expressed as

$$\alpha C''(s) - \beta C''''(s) + \gamma V = 0, \tag{30}$$

where $\alpha$ and $\beta$ are the weighting parameters that are used to control the strength of the snake's tension and rigidity respectively, $\gamma$ is a proportional coefficient and $V$ is the external force. Practically, these three parameters are set to be constants within the equation. $C(s)$ is the contour that delineates the desired boundaries, and $s$ ∈[0,1].

Before exploring any improvement based on the original GVF platform, we rewrite Eq. (30) as

$$g_1(d)C''(s) - g_2(d^{-1})C''''(s) + g_3(d)V = 0, \tag{31}$$

where $g_1(d)$, $g_2(d^{-1})$ and $g_3(d)$ are the *weighting functionals* of the internal and external energy terms, respectively; $d$ is the Euclidean distance between the presumed centroid of the real boundary and the estimated one of the snake. In fact, if the snake is ideally located on the real boundary, then they both most likely share a common centroid in addition to the merging of the contours. As a result, we consider the Euclidean distance between the centroids as an index of *proximity*. Note, that there exists a significant difference between the functional $g_1(d)$ and $g_2(d^{-1})$ in terms of the variables. This is due to the opposite behaviours of $d$ in the elatisity and rigidity terms, where the former is dominant in the energy function when $d$ is large and the latter plays a key role when $d$ is small (Zhou et al., 2005; Liu et al., 2008).

Alternatively, we can use a simplified version of Eq. (31) as

$$\tilde{g}_1(d)C''(s) - \tilde{g}_2(d^{-1})C''''(s) + \gamma V = 0, \tag{32}$$

which is dependent on the assumption that as the snake evolves, the GVF field keeps stationary (this assumption may lead to lower computation in the optimisation). Evidence shows that this assumption strictly holds in static images but might fail in dynamically variable images, e.g. motion artefacts, occluded images, etc.

Suppose that $\tilde{g}_1(d)$ and $\tilde{g}_2(d^{-1})$ have continuous derivatives. Then, one has a Taylor series, which can be defined as

$$\begin{cases} \tilde{g}_1(d) = \tilde{g}_1(d_1) + \tilde{g}_1'(d_1)(d - d_1) + \frac{\tilde{g}_1''(d_1)(d-d_1)^2}{2} + \cdots, \\ \tilde{g}_2(\frac{1}{d}) = \tilde{g}_2(d_2) + \tilde{g}_2'(d_2)(\frac{1}{d} - d_2) + \frac{\tilde{g}_2''(d_2)(\frac{1}{d}-d_2)^2}{2} + \cdots, \end{cases} \tag{33}$$

where $d_1$ and $d_2$ are two constants. The snake normally approaches the real boundary consistently and dynamically, indicating that the evolution of the snake can be linearised. Thus, the higher order terms ($\geq$ 2) in the Taylor series can be ignored.

Assuming that the snake starts from an initial contour, then the terms not related to $d$ (in Eq. (33)) will be replaced by 0. It should be noticed that during the iteration these terms may, or may not, be 0. However, setting them to 0 will avoid the side-effects of these constant terms during the evolution (e.g., slow convergence), and hence improve the efficiency of convergence towards the ideal contour. Consequently, Eq. (32) has a different form, defined by

$$\tilde{\alpha} d C''(s) - \frac{\tilde{\beta}}{d} C''''(s) + \gamma V = 0, \tag{34}$$

where $\tilde{\alpha} = \tilde{g}_1(d_1)$, $\tilde{\beta} = \tilde{g}_2(d_2)$, which both are constants in practice.

## *5.1 Segmentation Constrained by Mean Shift*

As mentioned before, when matched, the snake and the real contour will share a common centroid. Alternatively, having a common centroid is a necessary condition for registration of the two contours. We are therefore interested in driving the centroids of the two contours towards the same settlement whilst performing an appropriate segmentation. To do so, we again exploit the mean shift algorithm. In particular, CAMSHIFT (Bradski, 1998) is used due to its advantage of accounting for dynamically changing distributions during the contour evolution.

The application of mean shift for improved GVF segmentation works as follows. The mean shift algorithm is employed to find the contour candidate that is similar to the real boundary, where the similarity is measured by the Euclidean distance between the centroid of the area outlined by the deformable snake and that of the region surrounded by the real boundary, while satisfying the object energy function as well.

The following steps are conducted: The centroid $(x_c, y_c)$ of a contour is calculated by

$$\begin{cases} x_c = \frac{M_{10}}{M_{00}}, \\ y_c = \frac{M_{01}}{M_{00}}, \end{cases} \tag{35}$$

where we have the initial (0-th) moment $M_{00}$, the moment $M_{10}$ for $x$-coordinates, and the moment $M_{01}$ for $y$-coordinates of image points within the contour. The initial centroid of the real boundary is set to be the centre of the image. Then, the centroid of the region surrounded by the deforming snake is calculated. Once the centroids have been obtained, the Euclidean distance $d$ between these two centroids then becomes defined, which is used in Eq. (34) for energy minimisation in the revised GVF domain. This is followed by executing the standard CAMSHIFT algorithm (Bradski, 1998), (Wang et al., 2004), where Eq. (3) is deployed. Afterwards, we again compute the Euclidean distance between the centroids. We then subsequently apply the enhanced GVF and CAMSHIFT schemes. This process is iterated until the Euclidean distance of the centroids falls below a threshold ($< 0.1$ pixels in our experiments).

## *5.2 GVF-Means Shift Segmentation*

The combined gradient vector flow/mean shift algorithm (Zhou et al., 2010) proceeds in the following steps:

Step 1: Initialisation of a contour and the corresponding parameters.
Step 2: Employment of the standard GVF scheme for evolving the contour.
Step 3: Computation of the individual means of the two image regions.
Step 4: Computation of the Euclidean distance between the two centroids.
Step 5: Applying the revised GVF strategy.
Step 6: Conducting the standard CAMSHIFT process.

Step 7: If the Euclidean distance between the centroids $< 0.1$ pixel stop; otherwise repeat Steps 2-7.

Fig. 2 illustrates the contour evolution using this extended GVF algorithm, where the initial and final contours are demonstrated in Fig. 2(d).

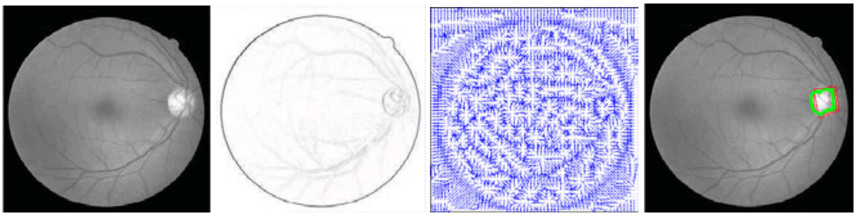


**Fig. 2** Illustration of contour evolution using the extended GVF performance: (a) Original gray image, (b) blurred image using Gaussian filtering, (c) GVF field map and (d) contour evolution (red colour indicates the deformation).

## 6 Evaluation and Discussion

In this section, we evaluate the presented mean shift based segmentation algorithms. For fairness, we compare each method against similar approaches.

### *6.1 Anisotropic Mean Shift Based FCM*

The presented AMSFCM segmentation algorithm was evaluated on a set of 100 dermoscopy images (30 invasive malignant melanoma and 70 benign) obtained from the EDRA Interactive Atlas of Dermoscopy (Argenziano et al., 2002) and the dermatology practises of Dr. Ashfaq Marghoob (New York, NY), Dr. Harold Rabinovitz (Plantation, FL) and Dr. Scott Meznies (Sydney, Australia). The benign lesions included nevocellular nevi and dysplastic nevi. Three sets of manual borders were determined by expert dermatologists, and serve as a ground truth for the experiments.

The algorithms that we compared are conventional FCM (Bezdek, 1980), EnFCM (Szilagyi et al., 2003), RSFCM (Cheng et al., 1998) and AMSFCM. In a final stage, morphological processing is employed for smoothing the segmentation outcomes, especially image borders, and removing small isolated areas.

An example of the segmentations obtained by the various algorithms is given in Figure 3 which shows one of the ground truth segmentations together with the results by all four methods. It can be observed that the segmentations produced by classical FCM and RSFCM are less smooth than those by EnFCM and AMSFCM. This is due to (1) RSFCM uses FCM in the second phase so they both have approximate convergence characteristics, and (2) EnFCM and AMSFCM take into account weighted image pixels so their outcomes are smoothed in the FCM stage. Clearly, smoother borders are more realistic and also conform better to the manual segmentations derived by the dermatologists. The second observation is also reflected in Figure 4, where original images are segmented using different FCM algorithms and the lesion borders are then extracted. It is also noticed that different algorithms generate similar results for Figure 4, while the proposed AMSFCM algorithm has clearly the best border result for the third example.

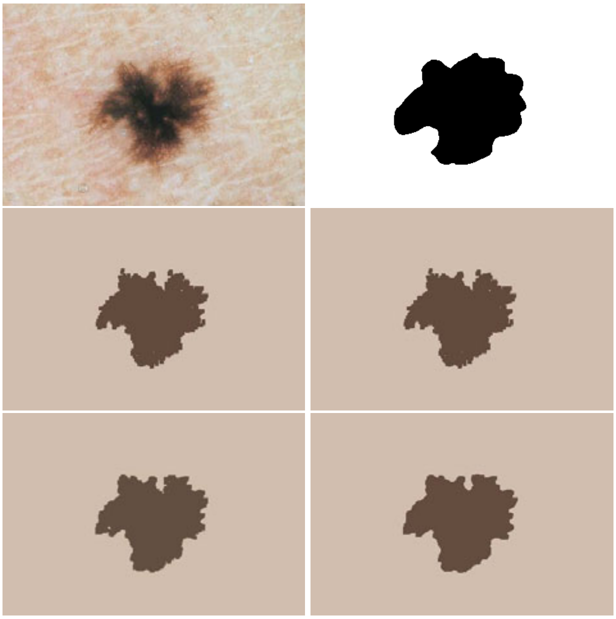


**Fig. 3** Segmentation comparison of original image (upper left), ground truth (upper right), FCM (middle left), RSFCM (middle right), EnFCM (bottom left) and AMSFCM (bottom right) for image 15.

For each image segmentation we record the number of True Positives $TP$ (the number of pixels that were classified both by the algorithm and the expert as lesion pixels), True Negatives $TN$ (the number of pixels that were classified both by the algorithm and the experts as non-lesion pixels), False Positives $FP$ (the number of instances where a non-lesion pixel was falsely classified as part of a lesion by an algorithm) and False Negatives $FN$ (the number of instances where an lesion pixels was falsely classified as non-lesion by an algorithm). From this we can then calculate the sensitivity $SE$ (or true positive rate) as

$$SE = \frac{TP}{TP + FN} \tag{36}$$

and the specificity $SP$ (or true negative rate) as

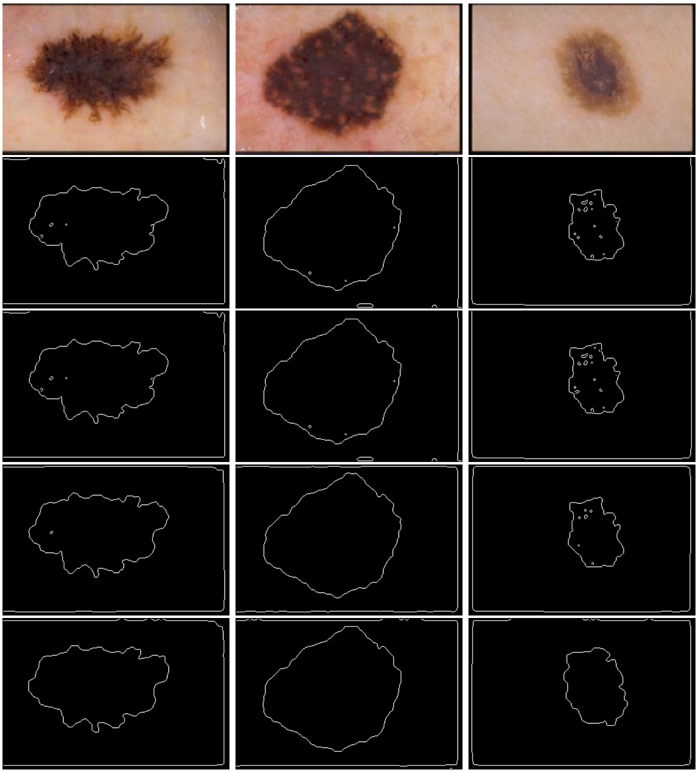


**Fig. 4** Border detection of exemplar segmented images (row 1: original images; row 2 - FCM results; row 3 - RSFCM results; row 4 - EnFCM results and row 5 - AMSFCM results).

$$SP = \frac{TN}{TN + FP} \tag{37}$$

In Table 1 we list the sensitivity and specificity obtained by all algorithms over the entire database and compared to all three ground truth segmentations (average $SE$ and $SP$ based on all three manual segmentations are reported). It can be seen that the proposed AMSFCM performs significantly better with an average sensitivity of about $78\%$ while the other algorithms achieve only a sensitivity of about $74\%$. In addition, our algorithm provides more consistent results as indicated by the lower variance of $SE$. As specificity is fairly similar for all algorithms, we can conclude that AMSFCM provides the best segmentation on the given dataset.

Computational efficiency is a crucial issue when considering FCM based segmentation. We record the number of iteration required in each FCM approach for evaluation, which in turn enables us to make a comparison regarding the relative efficiency of the different approaches. We normalised them so that the classical FCM algorithm is assigned 1.00 while the other ones represent the relative fractions they take compared to this. The results are also presented in Table 1 from which it can be

**Table 1** Segmentation performance on the dermoscopy dataset. For each algorithm the average sensitivity and specificity and the relative efficiency are given. The values in brackets indicate the standard deviations of the measures.

| Algorithm | Sensitivity | Specificity | computational cost |
|---|---|---|---|
| FCM | 0.739 (0.120) | 0.99 (0.056) | 1.00 (0.00) |
| RSFCM | 0.738 (0.118) | 0.99 (0.052) | 0.67 (0.11) |
| EnFCM | 0.740 (0.118) | 0.99 (0.061) | 0.80 (0.09) |
| AMSFCM | 0.776 (0.113) | 0.99 (0.065) | 0.63 (0.09) |

seen that the proposed AMSFCM takes computation efforts of 37%, 4% and 17% less than compared to FCM, RSFCM and EnFCM respectively.

Overall, it is evident that AMSFCM provides a very useful tool for the analysis of dermoscopic images. Not only does it provide the best segmentation results among the algorithms investigated, it also is the most efficient method.

## 6.2 Dirichlet Process Mixture Models with Mean Shift

We evaluated the combined Dirichlet process mixture/mean shift algorithm on a subset (50 images in total) of the Berkeley Segmentation dataset (Martin et al., 2001). The algorithms that we compared are the conventional MDP/MRF algorithm from (Orbanz and Buhmann, 2008) and the presented MDP/MRF with mean shift algorithm. In the final stage of both algorithms, morphological processing is

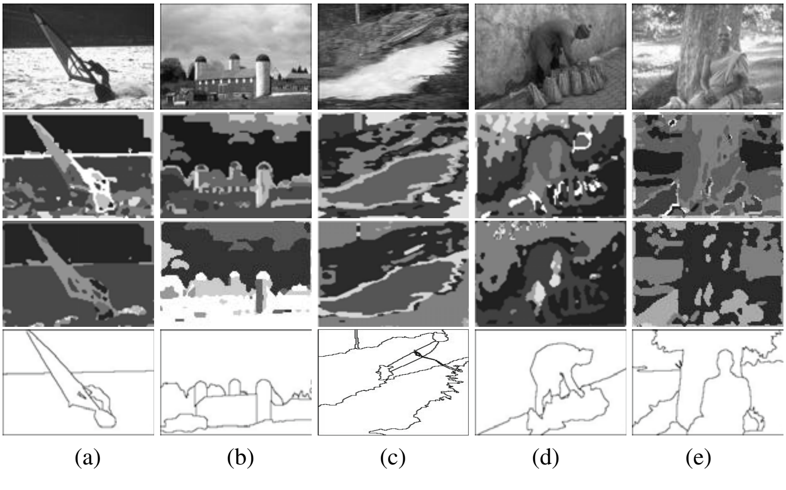


**Fig. 5** Performance comparison of image segmentation using the classical MDP/MRF and the proposed MDP/MRF with mean shift algorithms respectively: images (a) 5, (b) 7, (c) 10, (d) 18 and (e) 22 (1st row - original images, 2nd row - MDP/MRF, 3rd row - MDP/MRF with mean shift, 4th - ground truth segmentation).

employed for smoothing the segmentation outcomes and removing small isolated areas.

Examples of the segmentations obtained by both algorithms are given in Fig. 5 which shows the original images together with the results of the two methods and the ground truth segmentation. It can be observed that while the algorithms produce similar results, the segmentations produced by the classical MDP/MRF algorithm subjectively are less smooth than those by the proposed mean shift based MDP/MRF approach. This is due to the fact that the latter takes into account mean values in the sampling. Clearly, smoother borders are more realistic and also conform better to the manual segmentations.

**Table 2** Statistics of image segmentation performance.

| Algorithm | Recall | Fall-out | Accuracy |
|---|---|---|---|
| Classical | 0.4105 | 0.2116 | 0.7547 |
| Proposed | 0.4117 | 0.1588 | 0.8020 |

To obtain quantitative results, we calculate recall and fall-out, defined as

$$\text{recall} = \frac{TP}{TP+FN}, \tag{38}$$

and

$$\text{fall-out} = \frac{FP}{FP+TN}. \tag{39}$$

as measures for segmentation quality as is suggested in (Bowyer et al., 2001). We also calculate the overal accuracy, defined as

$$\text{accuracy} = \frac{TP+TN}{TP+FN+TN+FP}. \tag{40}$$

In Table 2 we list recall, fall-out and accuracy values for both algorithms averaged over all test images. It can be seen that both algorithms have similar recall values but that the proposed MDP/MRF with mean shift algorithm outperforms the classical MDP/MRF algorithm in terms of fall-out and accuracy by about 5%.

### *6.3 Gradient Vector Flow with Mean Shift*

The gradient vector flow/mean shift segmentation algorithm was evaluated on a set of 40 retinal images obtained from the DRIVE database (Staal et al., 2004). These images have been randomly selected from a screening database of 400 diabetic subjects aged 25-90. 33 of the images do not show any sign of diabetic retinopathy while in 7 signs of mild diabetic retinopathy are apparent. Each image is a true colour image of 768 by 584 pixels. The field of view of each image is circular with a diameter of approximately 540 pixels. No ground truth for the delineation of the optic disc is available, therefore subjective evaluation will be used in the following.

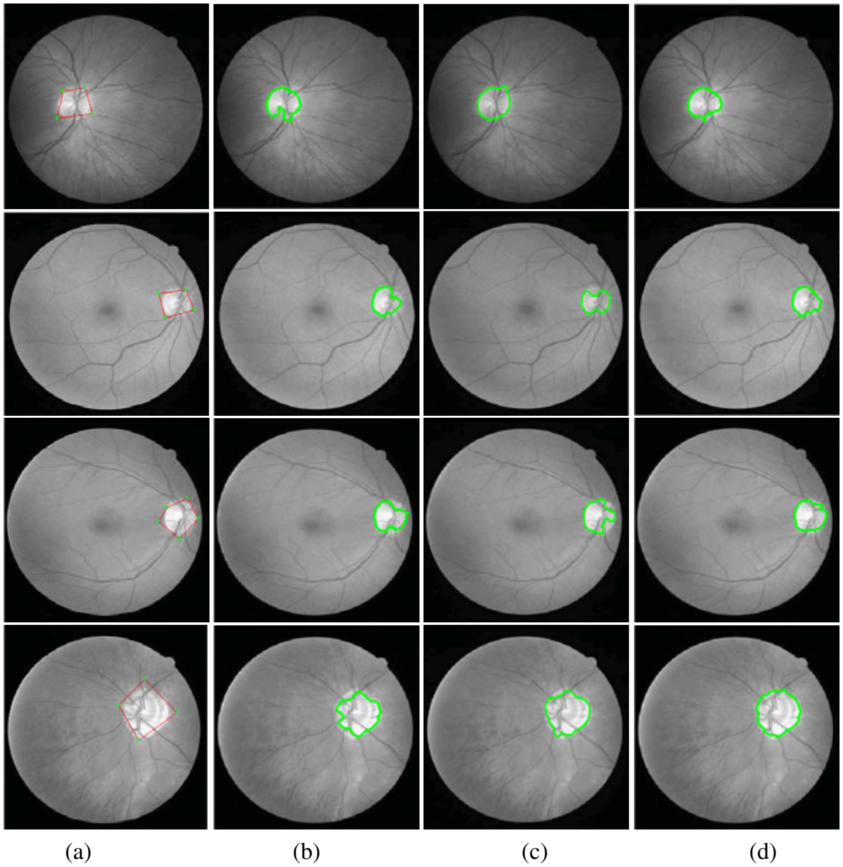


**Fig. 6** Performance comparison of different algorithms when the optic disk clearly appears, where (a) the original image superimposed by the initial contour, (b) GVF, (c) level set, and (d) proposed (image index from top to bottom: 15, 16, 27 and 31).

The algorithms we compare are the classical GVF algorithm (Xu and Prince, 1998), level set segmentation (Li et al., 2005), and the proposed improved GVF algorithm. For the two GVF based methods, the parameters have been set to: $\alpha$ (tension of the snake) = 0.05, $\beta$ (rigidity of the snake) = 0.0, $\gamma$ (step size in one iteration) = 1.0, and $\kappa$ (external force weight) = 0.6.

The entire evaluation consists of three major parts. Firstly, the three algorithms are evaluated using retinal images where the OD is clearly visible. This is the simplest case in our evaluation. Image examples of the experimental results are illustrated in Fig. 6. As can be seen, for image 15, level set segmentation and the proposed algorithm are superior to the classical GVF scheme. The results of images 16, 27 and 31 clearly show that the proposed algorithm has the best performance in terms of segmentation accuracy. This can be attributes to the computation of mean

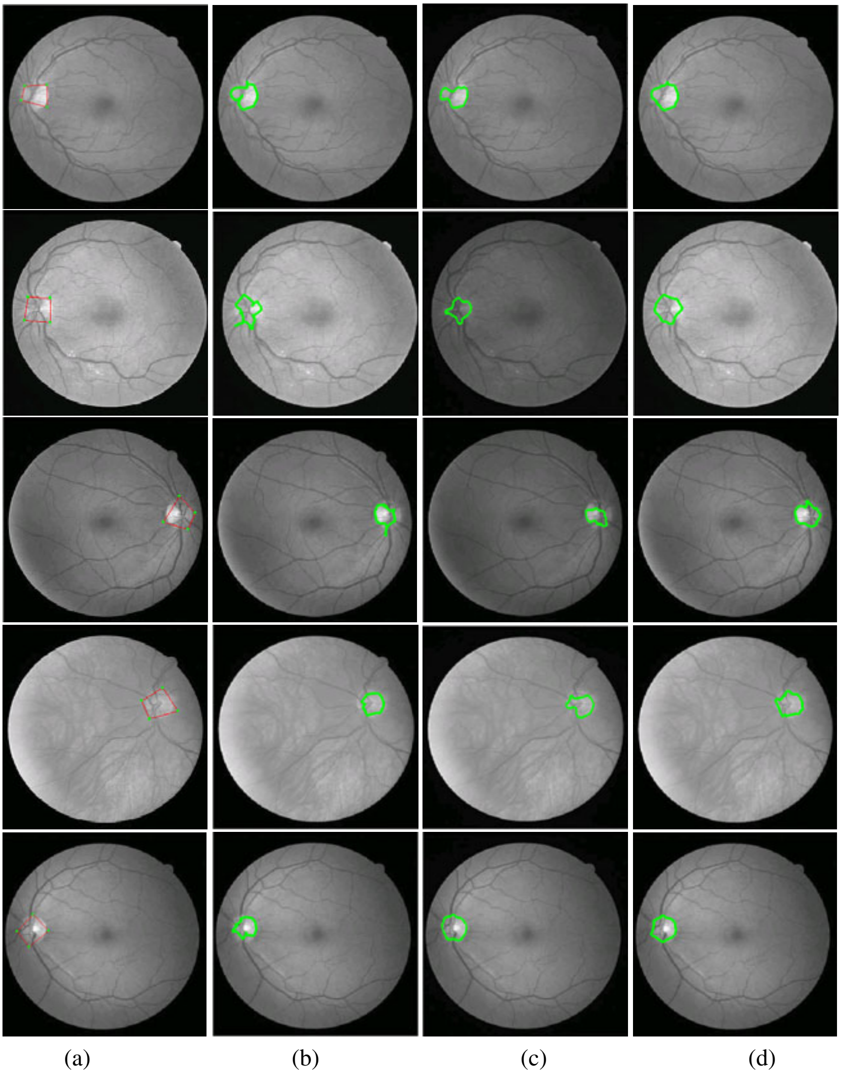


**Fig. 7** Performance comparison of different algorithms when the optic disk is vague, where (a) the original image superimposed by the initial contour, (b) GVF, (c) level set, and (d) proposed (image index from top to bottom: 1, 3, 19, 23 and 35).

fields in the domain of the proposed approach, which dynamically balances internal and external energy forces during the contour evolution.

In the second test group, the optic disc is less clearly defined and correct segmentation is hence more challenging. Results for this group of images are presented in Fig. 7. For image 23 it can be observed that its segmentation results by the classical GVF and our proposed algorithms are somehow similar, although the former leads

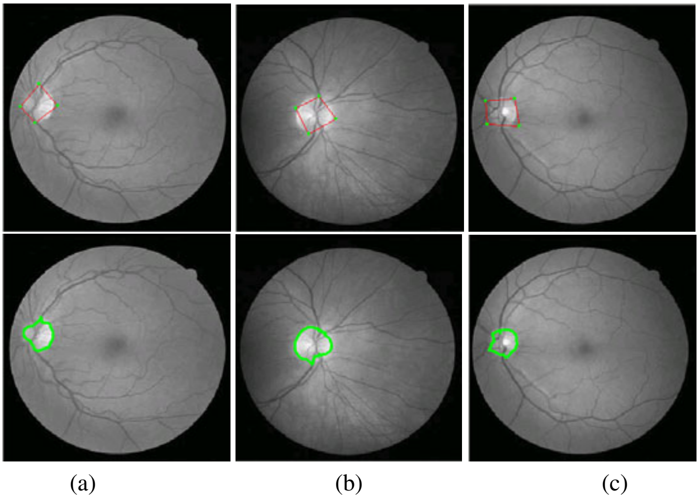


**Fig. 8** Performance evaluation after the positions of the initial contours change, where upper row indicates the original images superimposed by the different initial contours, and lower row shows the segmentation results by the proposed algorithm (image index from left to right: 1, 15, and 35).

to slightly lower accuracy on the OD edge. On the other hand, level set segmentation and the proposed approaches achieve comparable results in image 35. For the remainder of the images in this group, our proposed algorithm clearly outperforms the other two methods.

One of the challenges in image segmentation is whether or not the performance of a segmentation system can be consistently kept in different initialisation circumstances. To validate this, we randomly specify the starting contours for the involved images. This is followed by the regular routine of the proposed algorithm. Fig. 8 confirms that despite varied initial contours, the segmentation borders are virtually indistinguishable from those presented in Figs. 6 and 7. This confirms that the proposed algorithm is indeed strongly initialisation-invariant.

## 7 Summary

Image segmentation is frequently used in image analysis and pattern recognition. In this chapter, we have presented three mean shift based image segmentation algorithm. These methods incorporate a mean field term within individual standard fuzzy c-means, Dirichlet process mixture models and gradient vector flows segmentation frameworks. Based on a large set of dermoscopic images, we have shown that the AMSFCM algorithm is not only more efficient than other fuzzy c-means approaches but that it is also capable of providing superior segmentation. Similarly, it has been shown that the mean shift-based MDP/MRF image segmentation outperforms the classical MDP/MRF algorithm. Experimental results on a large dataset of retinal images have also demonstrated that the mean shift-based GVF method optimally detects the border of the optic disc.

## References

Argenziano, G., Soyer, H.P., De Giorgi, V.: Dermoscopy: A Tutorial. EDRE Medical Publishing & New Media (2002)

Belongie, S., Carson, C., Greenspan, H., Malik, J.: Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In: Proc. International Conference on Computer Vision, pp. 675–682 (1998)

Bezdek, J.: A convergence theorem for the fuzzy isodata clustering algorithms. IEEE Trans. Pattern Anal. Machine Intell. 2, 1–8 (1980)

Bowyer, K., Kranenburg, C., Dougherty, S.: Edge detector evaluation using empirical ROC curves. Computer Vision and Image Understanding 84(1), 77–103 (2001)

Bradski, G.R.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, 2nd Quarter (1998)

Caselles, B., Coll, B., Miorel, J.-M.: A kanisza programme. Progr. Nonlinear Differential Equations Appl. 25, 35–55 (1996)

Cheng, T.W., Goldgof, D.B., Hall, L.O.: Fast fuzzy clustering. Fuzzy Sets Sys. 93, 49–56 (1998)

Chuang, K., Tzeng, H., Chen, S., Wu, J., Chen, T.: Fuzzy c-means clustering with spatial information for image segmentation. Comput. Med. Imag. and Graph. 30, 9–15 (2006)

Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)

Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 142–149 (2000)

Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication, Hoboken (2000)

Jacobs, D.W., Weinshall, D., Gdalyahu, Y.: Classification with nonmetric distances: Image retrieval and class representation. IEEE Trans. Pattern Anal. Mach. Intell. 22(6), 583–600 (2000)

Li, C., Liu, J., Fox, M.D.: Segmentation of edge preserving gradient vector flow: An approach toward automatically initializing and splitting of snakes. In: IEEE Conf. on Comput. Vis. Patter. Rec., pp. 162–167 (2005)

Liu, T., Zhou, H., Lin, F., Pang, Y., Wu, J.: Improving image segmentation by gradient vector flow and mean shift. Pattern Recognition Letters 29(1), 90–95 (2008)

Makrogiannis, S., Economou, G., Fotopoulos, S.: A region dissimilarity relation that combines feature-space and spatial information for color image segmentation. IEEE Trans. Syst., Man and Cyber., Part B 35(1), 44–53 (2005)

Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (2001)

Morris, O.J., Lee, J., Constantinides, A.G.: Graph theory for image analysis: An approach based on the shortest spanning tree. Proc. Inst. Electr. Eng. 133, 146–152 (1986)

Orbanz, P., Buhmann, J.M.: Nonparametric Bayesian image segmentation. Int. Journal of Computer Vision 77(1/3), 25–45 (2008)

Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. Annual Review of Biomedical Engineering 2, 315–337 (2000)

Staal, J.J., Niemeijer, A.M., Viergever, M.A., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Trans. on Medical Imaging 23, 501–509 (2004)

Szilagyi, L., Benyo, Z., Szilagyi, S.M., Adam, H.S.: Mr brain image segmentation using an enhanced fuzzy c-means algorithm. In: 25th Annual Information Conf. of IEEE EMBS, pp. 17–21 (2003)

Tao, W., Jin, H., Zhang, Y.: Color image segmentation based on mean shift and normalized cuts. IEEE Trans. Syst., Man and Cyber., Part B 37(5), 1382–1389 (2007)

Venkateswarlu, N.B., Raju, P.S.V.S.K.: Fast isodata clustering algorithms. Pattern Recogn. 25(3), 335–342 (1992)

Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. 13(6), 583–598 (1991)

Wang, J., Thiesson, B., Xu, Y., Cohen, M.: Image and video segmentation by anisotropic kernel mean shift. In: Proc. European Conference on Computer Vision, pp. 238–249 (2004)

Wells III, W.M., Grimson, W.E.L., Kikinis, R., Jolesz, F.A.: Adaptive segmentation of mri data. In: Proc. of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine, pp. 59–69 (1995)

Witkin, A.P., Terzopoulos, D., Kass, M.: Signal matching through scale space. International Journal of Computer Vision 1(2), 133–144 (1987)

Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Trans. on Image Processing 7(3), 359–369 (1998)

Zhou, H., Liu, T., Hu, H., Pang, Y., Lin, F., Wu, J.: A hybrid framework for image segmentation. In: Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 749–752 (2005)

Zhou, H., Schaefer, G., Celebi, M.E., Fei, M.: Bayesian image segmentation with mean shift. In: Proc. of IEEE Conference on Image Processing, pp. 2405–2408 (2009a)

Zhou, H., Schaefer, G., Liu, T., Lin, F.: Segmentation of optic disc in retinal images using an improved gradient vector flow algorithm. Multimedia Tools and Applications 49(3), 447–462 (2010)

Zhou, H., Schaefer, G., Sadka, A.H., Celebi, M.E.: Anisotropic mean shift based fuzzy c-means segmentation of dermoscopy images. IEEE J. Select. Topics in Sig. Proc. 3(1), 26–34 (2009b)

# Author Index