1998

3D Object Recognition Based On Constrained 2D Views

Toth, Levente

3D Object Recognition Based On Constrained 2D Views

By

Levente Toth

A thesis submitted to the University of Plymouth

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Electronic, Communication and

Electrical Engineering

12 October 1998

3D Object Recognition Based On Constrained 2D Views

By

Levente Toth

Abstract

The aim of the present work was to build a novel 3D object recognition system capable of classifying man–made and natural objects based on single 2D views. The approach to this problem has been one motivated by recent theories on biological vision and multiresolution analysis. The project's objectives were the implementation of a system that is able to deal with simple 3D scenes and constitutes an engineering solution to the problem of 3D object recognition, allowing the proposed recognition system to operate in a practically acceptable time frame.

The developed system takes further the work on automatic classification of marine phytoplanktons, carried out at the Centre for Intelligent Systems, University of Plymouth. The thesis discusses the main theoretical issues that prompted the fundamental system design options. The principles and the implementation of the coarse data channels used in the system are described. A new multiresolution representation of 2D views is presented, which provides the classifier module of the system with coarse–coded descriptions of the scale–space distribution of potentially interesting features. A multiresolution analysis–based mechanism is proposed, which directs the system's attention towards potentially salient features. Unsupervised similarity–based feature grouping is introduced, which is used in coarse data channels to yield feature signatures that are not spatially coherent and provide the classifier module with salient descriptions of object views. A simple texture descriptor is described, which is based on properties of a special wavelet transform.

The system has been tested on computer–generated and natural image data sets, in conditions where the inter–object similarity was monitored and quantitatively assessed by human subjects, or the analysed objects were very similar and their discrimination constituted a difficult task even for human experts. The validity of the above described approaches has been proven. The studies conducted with various statistical and artificial neural network–based classifiers have shown that the system is able to perform well in all of the above mentioned situations. These investigations also made possible to take further and generalise a number of important conclusions drawn during previous work carried out in the field of 2D shape (plankton) recognition, regarding the behaviour of multiple coarse data channels–based pattern recognition systems and various classifier architectures.

The system possesses the ability of dealing with difficult field–collected images of objects and the techniques employed by its component modules make possible its extension to the domain of complex multiple–object 3D scene recognition. The system is expected to find immediate applicability in the field of marine biota classification.

Acknowledgements

The author gratefully acknowledges the assistance of Dr. Phil Culverhouse and Dr. Rob Ellis for their supervision, invaluable help, guidance and nevertheless, for the great brain–storming sessions.

I gratefully acknowledge the support of the University of Plymouth, which through a research studentship made all this work possible.

The author acknowledges the help of Paul Rankine from the Marine Laboratory, Agriculture and Fisheries Department, The Scottish Office, Aberdeen, for providing an important test data set.

I whole-heartedly thank my family for their support and encouragement.

I gratefully acknowledge the immense help, guidance and friendship of the late Roman Dan, my former tutor and mentor, who introduced me to the world of image processing.

Many thanks to my colleagues Julian, Nick and Steve for the help in getting started and for the considerable amount of good quality humour that compensated for the weather.

Also, an informal thanks to the sound painters Klaus Schulze and Vangelis for succeeding with their creations to make adrenalin levels at the end of the day come back to normal.

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed by an University of Plymouth Research Scholarship and an University of Plymouth Research Studentship.

The work has been regularly presented at research seminars and a journal paper has been prepared.

Publications:

One major journal paper has been accepted for publication during the course of this study:

Toth, L., Culverhouse, P. F., 3D object recognition from static 2D views using multiple coarse data channels, Image & Vision Computing Journal., (In press).

Signed: Levente Toth

Date: 5/12/98

Glossary of terms

ANN – Artificial Neural Network

ANOVA – Analysis of Variance

ART – Adaptive Resonance Theory

ARTMAP – Adaptive Resonance Theory Map

CMC – Committee with Most Confident member

CSO – Committee with Sum of Outputs

DA – Discriminant Analysis

DWT – Discrete Wavelet Transform

DWPT – Discrete Wavelet Packet Transform

FFT – Fast Fourier Transform

FWT – Fast Wavelet Transform

MRA – Multiresolution Analysis

MSE – Mean Square Error

RF – Receptive Field

SOM – Self–Organising Map

SNNS – Stuttgart Neural Network Simulator

WT – Wavelet Transform

WPT – Wavelet Packet Transform

Table of contents

List of figures

List of tables

Chapter 1. Introduction

1.1. Prologue

"... seeing, regarded as a supply for the primary wants of life, and in its direct effects, is the superior sense... The faculty of seeing, thanks to the fact that all bodies are coloured, brings tidings of multitudes of distinctive qualities of all sorts; whence it is through this sense especially that we perceive the common sensibles, viz. figure, magnitude, motion, number..."

(Aristotle – On Sense and the Sensible)

From the practical importance of being able to navigate in and to shape the surrounding world, to the ability of emotionally contemplating a work of art, vision is the 'supreme' sense that defined and defines the ways in which humans as species exist and manifest themselves. Beyond the tangible results of our practical activities, we developed signs for visual encoding of languages – but also signs for the mathematics that reflect our current understanding of the Universe. Conversely, the importance of vision as primary source of information led to countless linguistic idioms. Expressions like 'to see the point' or 'seeing is believing' show not only the dominant role of vision in our perceptions of the world, but also how infallible this sense was thought to be.

Not surprisingly, there has been a continuous quest for a clear *picture* on how vision works, from the reflections of Democritos and Aristotle to the current neurophysiological and psychological studies. This quest is far from over. The time span of this gnostic search and its failure to produce a final model of vision seems to be in curious contradiction with the apparent effortlessness that characterises vision as task. We contemplate the world, perform complex operations in complex three–dimensional environments, without being consciously preoccupied with seeing. The attempts in artificial intelligence research to imitate vision, i.e. the research into its computational modelling unveiled the latent complexity of the involved processes. The various approaches towards building a computer vision system not only brought to surface new aspects of the problem, but also helped in testing our theories on biological vision.

This thesis describes another attempt to construct a computer vision system, one that is able to

recognise three–dimensional objects from single two–dimensional views. The objectives of the work described herein were the realisation of a system that presented with simple 3D scenes can perform categorisation of artificial and natural objects. Furthermore, the system was meant to represent an engineering solution to the problem, where the resulting working model can find immediate application in automatic classification of marine biota. The long–term aim behind this is to make the system capable of operating in real–world situations (i.e. complex scenes with multiple occluding objects). Beyond the immediate practical applicability of the result of this piece of research, the other main concern has been the increase of our knowledge on the *modus operandi* of vision systems.

1.2. Background

The ideas for this piece of research were prompted by previous work on automatic classification of marine phytoplankton, carried out in the Centre for Intelligent Systems, University of Plymouth, in collaboration with the Plymouth Marine Laboratory (as reported in Culverhouse *et al.*, 1996; Ellis *et al.*, 1994, 1997), as part of the European Union–funded MAST programme (contract MAS2–CT92–0015).

The above project resulted in the realisation of the DiCANN system that dealt with images of 23 species of field–collected planktons and classified them accordingly. The system employed multiple data channels that presented information produced by various feature extraction modules to a neural network–based classifier. Due to the quality of the images, the system had to operate in difficult conditions: in the images that served as input to the system, detritus, illumination gradients and noise were present. Furthermore, the morphological variations in the plankton specimens made the classification task more difficult. Therefore a core concept in the design of the system has been the use of so–called coarse data channels. Instead of relying on exact shape descriptors and feature encoding, the system utilised coarse coding of features. The coarseness of the representation was expected to compensate for variations of the image attributes enlisted above. The validity of such an approach has been proved by the success of the DiCANN system in achieving a classification accuracy close to that of expert human taxonomists.

A total of six coarse data channels have been used in the DiCANN system, all of them encoding features extracted from the whole surface of the analysed objects. This approach paid off since the system dealt with objects that had an essentially two–dimensional shape. In the world of 3D

objects though, a recognition system is confronted with the effects of changes in viewpoint and illumination, the latter altering shading of visible surfaces. The nature of the feature data channels employed in DiCANN can give one clues on the problems that one encounters when moving towards the problem of recognising 3D shapes. For instance, surface texture will change with perspective; object boundaries' shape will change also, hence boundary descriptors will be affected. A move from global methods that work on the whole input shape towards local feature–based techniques seems necessary. Since noise, variable viewpoints and different contrast conditions are to be considered, the encoding of the features must be sufficiently coarse, at least in part compensating for the variable conditions that can alter the details in the 2D views used as input.

The work described in this thesis is a result of an attempt to extend the concept of coarse data channels to the domain of 3D object recognition. As in the case of DiCANN, the proposed object recognition system was meant to be able to classify very similar shapes, whose categorisation in many cases is difficult even for human experts. The above considerations led finally to particular choices for the feature extraction modules employed in the proposed 3D object recognition system, that was meant to operate also on non–rigid natural shapes and noisy images. The preprocessing modules performing multiscale analysis were based on a wavelet transform that, although has not been used extensively in object recognition applications as far as the literature on this subject shows, offered very attractive properties from the point of view of pattern recognition. Also, it led to a computationally inexpensive implementation, which was one of the aims of the project, having in mind the issues behind the practical applicability of the proposed system. As it will be pointed out in the thesis, the study of the transform leads to elegant image analysis techniques (e.g. directional texture descriptors, edge detectors), although some of these were not used in the present implementation of the recognition system.

This project's contribution to knowledge is summarised below. A novel multiresolution skeleton representation of 2D views has been developed and used in the system as a coarse data channel. It proved to be able to register salient shape properties of rigid and non–rigid objects, in conditions of significant variations in viewpoint and poor quality images. This scale–space representation also has the potential to register multiple objects in the image. An extended version of the recognition system (as it will be described in the chapter focusing on future work) would therefore be able to operate with several objects of arbitrary size in the field of view. This scale–space

representation therefore constitutes an improvement to the way in which the DiCANN system operated with single objects in the input.

Several feature extraction channels used in the system rely on classic techniques (like spatial frequency extraction), but as a novel approach, these are parts of a wavelet–driven attention focusing sub–system. This uses wavelet transform in order to direct the extraction of features in areas of interest of the input image, these areas containing details that are potentially salient for recognition. Unsupervised similarity–based grouping of features was proposed, that provide coarse signatures of the features collected from these areas of interest. Therefore extracted features have no spatial coherence when fed into the categoriser, the spatial information being supplied separately by the scale–space channel in a coarse code form. A very simple and coarse texture descriptor is developed, which exploits the advantages of the A Trous wavelet transform over other classic wavelet transforms. Several useful properties of the A Trous transform were uncovered during these studies, and it has been proven that this transform can become a very versatile and sophisticated tool in shape recognition applications.

The performance of the system has been evaluated using statistical and neural network–based classifiers. The use of collective and committee machines as categorisers (Ellis *et al.*, 1997) helped in obtaining a picture on how the various feature channels contribute to the system's performance in recognition tasks. As a piece of research, the comparative studies carried out on the performance of collective and committee machines made possible the generalisation to the field of 3D shape recognition of the findings of the DiCANN experiments.

1.3. Structure of the thesis

The following chapters describe the theoretical grounds of the research, together with the details of the proposed object recognition system and its performance in various object classification experiments.

1.3.1. Review of theories

The second chapter of this thesis consists of a review of theories on vision, with emphasis on the current experimental evidence and theories that provided the support for the rationales behind the design of the proposed system. Considerations on the computational modelling of biological

vision are presented, discussing the levels of abstraction in the study of vision. The problem of internal representation of the visible world is introduced, which is an important element in any visual system that attempts recognition. The theories on object– and viewer–centred representation are discussed, pointing out the body of experimental evidence in support of the latter. The idea of feature–based representation is introduced. The final section of the chapter discusses the issue of multiscale representation of images and the plausibility of such a technique in the light of studies on biological vision. This section prepares the ground for the in–depth discussion of wavelet–based multiresolution analysis, which can be found in chapter three.

1.3.2. Multiresolution analysis

Chapter three presents the mathematical bases of the wavelet transform and multiresolution analysis that constitute a core element of the proposed system. This chapter underlines the particularities of various mathematical and algorithmic solutions and explains the reasons behind the choices made in the implementation phase. The intuitive aspects of multiscale analysis of signals and the properties of wavelets are described, together with a brief overview of other multiresolution analysis methods and their drawbacks. The continuous wavelet transform is defined and the subsequent sections present the mathematical path that leads from this to the so–called A Trous algorithm in an attempt to unify the various descriptions found in the literature. A particular implementation of the latter is discussed in the final section of the chapter, that points out the practical advantages of the particular choices made in the algorithm's mathematics. Also, a number of important properties of this transform are pointed out, these properties being used in the proposed system or provide elegant image analysis tools for future applications. At all stages of the discussion, connections between the transforms' mathematics and the world of signal processing are made via filter bank analogy.

1.3.3. Overview of the system's structure

Having discussed the theoretical issues that have to be taken into account when designing a vision system and the mathematics of multiresolution analysis, chapter four describes the architecture of the proposed system. This chapter presents the rationale behind the architectural choices and the design options, based on the theories discussed in the earlier chapters. The structure of the system is outlined and component modules are described. Their function is presented, together

with a discussion of the ideas and practical necessities that led to the particular choices in their design. An overview of related work is included in the descriptions, highlighting the issues that prompted the ideas behind certain parts of the system. This chapter prepares the ground for the in–depth description and discussion of each of the component modules.

1.3.4. Preprocessing and feature extraction

This chapter describes in detail the *modus operandi* of the image preprocessing modules employed in the system, together with the structure and functions of the feature extraction and coarse coding modules. The generated data structures are exemplified and pseudocode sequences help the understanding of the algorithms used in the processing. The unsupervised feature grouping processes are also described. This chapter completes the task of describing the coarse data channels, the issues of data classification being the subject of a subsequent chapter.

1.3.5. Categorising the data

The categoriser module is described in this chapter. The fundamental issues of classification and classifiers are briefly described, subsequent sections presenting the theory behind and the functions of statistical and neural network–based categorisers. Discriminant analysis is described, highlighting the most important theoretical aspects and the properties of this method. Multilayer feedforward networks are presented, together with details of training and testing methods. Collective and committee machines based on neural networks are described, pointing out the reason for their utilisation in the system's testing.

1.3.6. Data sets and evaluation methods

This chapter presents the data and methods used to test the system's performance. The used sets of synthetic and natural images are described and this chapter also includes an overview of the evaluation methods used during the assessment of performance. The considerations taken into account during the design of the synthetic test objects and their images are detailed, presenting the characteristics of the resulting data sets. Similarly, the problems posed by natural images are described, highlighting the reasons in choosing such a data set. The theoretical issues of the used evaluation methods and statistical descriptors are briefly introduced, pointing out their role in revealing aspects of the system's behaviour.

1.3.7. Preliminary tests on coarse data channels

The results of the various preliminary tests performed on the system are described in detail in chapter eight. These classification trials allowed certain parameters of the scale–space channels to be chosen, as a compromise between good feature representation, computational load and practical feasibility. Also, these tests made possible the study of the behaviour of the individual coarse data channels in conditions of monitored viewpoint changes. Parallels between human perception of the test shapes and the resulting classification accuracies could be drawn.

1.3.8. Classification of synthetic shapes

Object recognition trials carried out on synthetic images are presented and the performance achieved by the system is evaluated in these conditions. Statistical and neural network–based categorisers are employed. Statistical methods are used to investigate the effects of changes in feature channel configurations, classifiers' internal architecture and data set sizes. The extensive set of trials allows the construction of a detailed picture on the behaviour of the system in various conditions and it helps in defining the possible modifications that can be brought to the system's structure in practical applications.

1.3.9. Classification of natural shapes

The system is tested on a set of fish larvae and detritus images, in order to study its behaviour when presented with noisy, poor quality images, very similar non–rigid shapes and morphological variations of the objects. The issues raised by such a data set are described, followed by a detailed report of the results of tests that used statistical & ANN–based collective machines and committees. The performance and its variation with training set size, hidden layer size, channel configuration is studied, revealing also the discriminatory power of each coarse data channel.

1.3.10. Conclusions and future work

The final chapter of the thesis presents the set of conclusions that can be drawn from the system's behaviour in the performed tests and also describes the ways in which the work can be taken further. After summarising the system's performance and behaviour in various conditions, a number of sections describe future directions of research, in some cases offering outlines of algorithms that could make certain extensions of the system's functionality possible.

1.3.11. Appendices

The appendices contain example code sequences used in the preprocessing and categoriser modules of the developed system. The code listings in C++, PopLog 11, Matlab languages illustrate the implementation of preprocessing algorithms, wavelet transform, feature extraction modules and classifiers.

1.4. Summary

This introductory chapter presented the issues related to the difficulty of the computational modelling of vision and outlined the background work that served as basis for this piece of research. The novelty of this project has been briefly described. In the final sections of the chapter, the structure of the thesis was described briefly, by presenting the contents of each chapter.

Chapter 2. Theoretical considerations

"... a mental image is something completely different from a visual image, and however much one exerts oneself, one can never manage to capture the fullness of that perfection which hovers in the mind and which one thinks of, quite falsely, as something that is 'seen'. "

(Maurits Cornelis Escher)

2.1. Introduction

The present chapter is an overview of the theories and experimental evidence in vision research that led to particular choices in the design of the proposed computer vision system. The first section constitutes a preamble, that presents the difficulty of modelling biological systems, more precisely, biological vision. The problem of internal representation of objects is introduced in a subsequent section, followed by a discussion of the theories on object– and viewer–centred representation in biological vision. A further section presents the concept of multiple feature–based representation and discusses the plausibility of its use in a vision system. The final section of this chapter puts the emphasis on theories on multiscale representations in biological vision and on the rationales behind the use of multiresolution analysis in the structure of the proposed object recognition system.

2.2. Modelling biological vision

The neurophysiological, psychophysical and computational studies of biological vision have unveiled the remarkable complexity of the involved processes. In the field of computer vision attempts are made to model some of these processes that take place in the brain. Vision research seeks to uncover the nature of, and the mechanisms behind, the information processing tasks that allow biological systems to acquire a rich understanding of the world, based on the retinal image. The desire to imitate nature comes perhaps from the idea that the versatile biological vision sys-

tem must be the most optimal practical solution to the problem of vision, since we are so good at it.

At the beginnings of the computer revolution, everything seemed to be possible in computational modelling. Computers were soon being heroically programmed to write poems, compose music, create paintings, understand language– all this as an attempt to emulate functions of the brain. In vision (and not only), when the limitations of hardware and algorithms together with the lack of valid theories became evident, a solution to the modelling of brain functions seemed to be (at least theoretically) the use of hugely complex computer architectures that would employ potentially millions of processing units and therefore would emulate the functions of a huge mass of neurons. But a more sane approach was to separate the hardware from the tasks that were meant to be imitated by the machine. This approach, emerging from the realisation of the complexity of the brain's hardware and its essentially different nature from that of computers, found a clear expression in the work of Marr (1982).

Marr defined three distinct levels of abstraction in the computational study of biological systems, and more concretely, of vision. The first would be the mechanism level – the study of the brain at this level would imply mapping of groups of neurons, the study of their behaviour, interconnections and interactions. The study of computers at this level would mean analysis of the circuits and of their interactions. In both cases, the description of a process with these means becomes extremely complicated – one would say, even practically useless. Neurophysiological studies provide essential information on how certain modules of the brain work, by measuring the activations of a neuron or classes of neurons in conditions of monitored stimuli. These studies have played an important role in the development of many theories on vision, as it will become apparent in the following chapters. But to use these findings directly, in order to build a computer hardware that imitates the interactions and behaviour of the neurons, is a non–practical path. It could work for very small functional units (modelling, let's say, the retina or the receptive fields of the visual cortex), but it does not provide at the moment a computationally feasible model of vision.

The second level of abstraction is the algorithmic level, the name being self–explanatory. In this case, one arrives at a set of algorithms that describe step by step the path followed by a process one wishes to describe. The exact implementational solution depends of course on the hardware. Nevertheless, such a description gives a comprehensive image on the studied processes and their ways of interacting. Such algorithms for sub–modules of the biological system can be worked

out by treating the studied module as a black box. By providing certain inputs to the subsystem, and monitoring its output, it is possible to describe its behaviour in well–defined circumstances. This can be considered to be the purpose of psychophysical experiments. Being confronted with a system as complex as the brain, one submits human or animal subjects to experiments where systematically designed stimuli are presented to them and the subjects are required to perform various operations based on these. Response times, error rates or other measurable aspects of their performance during the experiments can lead to conclusions on the involved brain processes.

The third and most abstract level according to Marr, is the computational level. By determining the nature of the information processing tasks, computational theories can be elaborated. These provide models that describe what data is processed and how, the latter usually leading to mathematical descriptions. These can be later implemented on a suitable hardware platform.

The implementation of the computational strategies can have as direct benefit the validation of the theories, if the system successfully emulates the functions of a biological (sub)system. Conversely, it can uncover the complexity of certain processes and help in the definition of the constraints that the emulated system operates with. As Marr pointed out, investigation of vision at one single level of abstraction can not lead to the understanding of what vision is or how it works.

2.3. Representing the visible world

The computational modelling of vision from its early days aimed at the creation of a computer system that would imitate the brain's behaviour and deliver similar abilities when performing the perhaps most practical task of vision: object recognition. Because of the problem's tremendous difficulty, there was a trend in computer vision research that hypothesised: in order to recognise an object, the visual system needs every possible kind of information on that object and its environment. This holistic approach reflected in the work of Freuder, Tenenbaum & Barrow (reviewed by Marr, 1982) was contradicted by research that proved that even in difficult circumstances, shapes of objects could be determined by vision alone. An eloquent example in this sense is Warrington's work (Warrington & Taylor, 1973). Her experiments showed, that subjects by vision alone could build an internal description of the shape of a viewed object, even when the subjects could not work out/understand the object's use and purpose. This led to the idea that shapes of objects are stored in the brain separately from the information on their use. Hence vision

came to be regarded upon as a chain of processes that, based on the retinal image, yield increasingly complex representations of the visible world.

A computational study of the information processing strategies employed by biological systems led later to a distinction between low and high–level vision (Marr, 1982; Hildreth & Ullman, 1989). In the case of low–level vision, thought to act independently of the visual task (Yuille & Ullman, 1990), the visual information is the retinal image provided by receptors in the eye. This light–intensity information is the input to a series of parallel processing modules that lead to early visual representations. These capture the geometry of the viewed shapes, surface orientations, information on movement, depth, colour, texture etc., such that the light intensity information provided by the retina is organised into an early representation of relevant events in the image. In Marr's model of vision, such low–level visual information processing tasks are performed in two stages (as illustrated below in Fig. 2.1.).

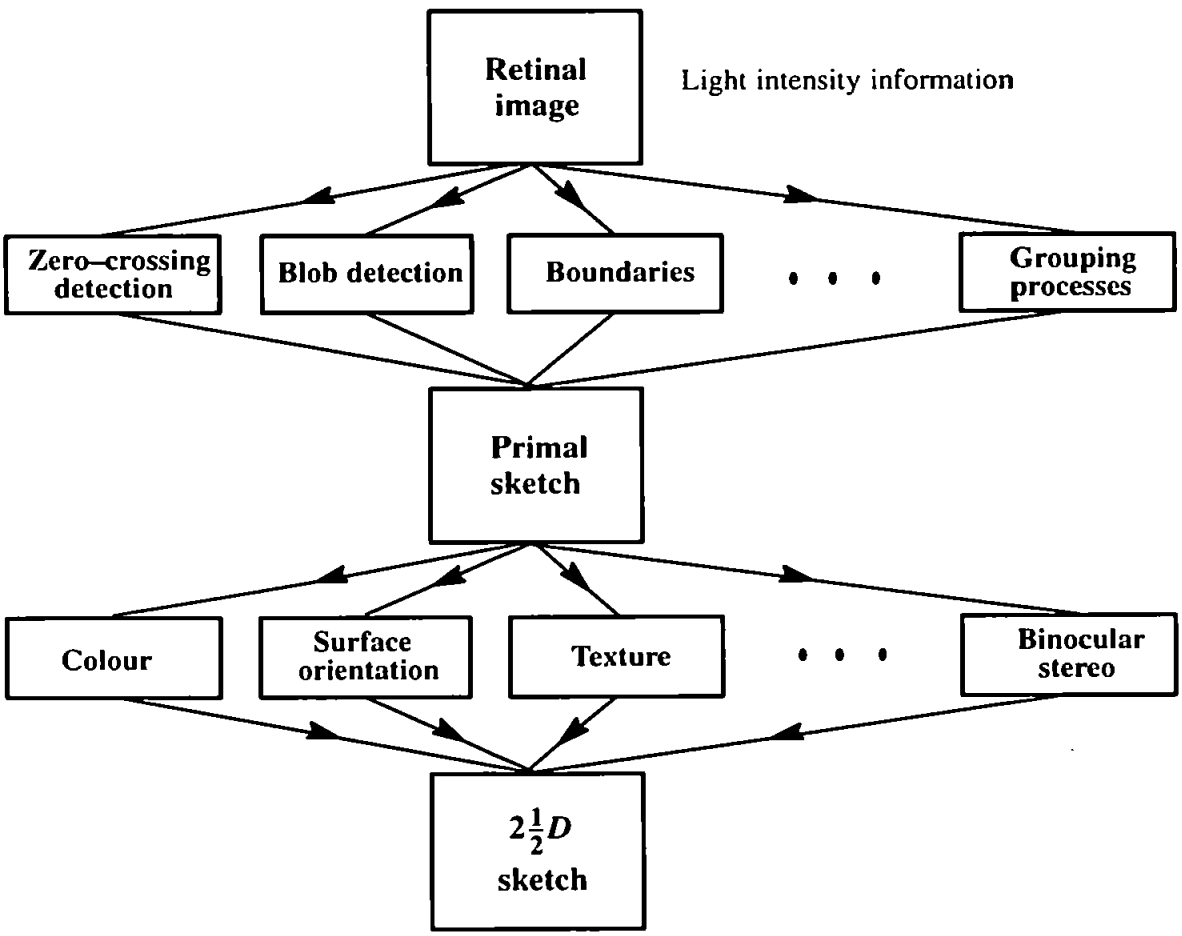


Fig. 2.1. Marr's model of early vision

The early representation consists of a primal sketch that contains information on intensity

changes, local geometry (like the output of the blob and line detectors proposed by Marr). Detection of edges, or, in a more relaxed framework, of zero crossings of the signal's second derivative are an important element in this model. The importance of these comes from the realisation of the fact that sharp changes in light intensity in the image can be attributed to object boundaries and pronounced changes in surface properties (e.g. orientation). These seem to contain essential information on the viewed scene, since humans are very good at understanding 3D scenes from line drawings, as studies involving sketch–like representations showed (Kanade's origami world, described in Sabbah, 1985; Ballard & Brown, 1982). Having built the primal sketch, the analysis of texture can add information on the surfaces and their orientation. This leads to the $2\frac{1}{2}$D sketch, that represents visible surfaces, their discontinuities and adds binocular stereo information (e.g. depth). Yuille & Ullman, 1990 points out though, that stereo and motion information could be extracted from edge data only, without relying on the primal sketch.

Based on these early representations, the role of high–level vision then is to recognise the objects and in the context of locomotor skills, to obtain the necessary information for manipulating them. Once we recognised an object, together with the object's name (label) other information comes to us naturally: the purpose of the object, its utility etc. These associations between the object label and what we know about it from our contacts with it in practical life are triggered by recognition. Therefore the latter could be easily considered the primary task of higher–level vision. To reiterate Warrington's conclusions, everything we know about an object's use and purpose, its possible arrangements in space does not constitute necessary input data to the recognition process. Therefore in order to recognise a three–dimensional object, the visual system must arrive at a representation of the input that can be matched against a library of learnt object representations. The theories on the way in which a biological system performs this particular task defined the main directions that computer vision research has taken in the past decades (as reviewed in Marr, 1982; Edelman & Bulthoff, 1992).

Is the brain extracting and storing extensive amounts of geometrical and physical information on the objects? Sutherland (1979) suggested that this is a crucial requirement, stored 3D models being necessary for us to be able to manipulate objects and to navigate in the surrounding environment. Also, when moving around in the object world, we need volumetric information on the objects so that we can avoid them. The reconstructivist approach represented by Marr, too, suggested that our mental image of the world allows geometric reconstruction of a 3D scene,

therefore the representation contains all information necessary for performing any visual task. It would seem that due to the complexity of the visible world and to the huge variety of visual tasks that humans have to perform, we need a mental image that is a perfect representation of the surrounding 3D environment. Or perhaps the brain's internal representations essentially differ from the geometric and physical rules that characterise the material world (like constraints introduced by the presence of gravity), operating only with snapshots of reality, with sets of features? Escher's visions that defy topology and gravity can hardly be the result of a mental imagery system that is hard–wired to reflect and respect the physics and geometry of the material world. Minsky (1975) gave an eloquent common–sensical example of how much we can disregard the geometry of our environment, without compromising our ability to orientate in it and to perform complex locomotor tasks. He describes his discovery of the true geometry of Boston's central park years after he moved there. All the previous misconceptions on the shape of that environment that rendered his reference system geometrically absurd did not affect his ability of navigating in that environment. This in itself would be a counter–argument to holistic or reconstructivist approaches.

The above rhetoric questions mark the perhaps most prominent ideological battle in vision research between the advocates of theories on object– and viewer–centred representation, which is the subject of the next section.

2.4. Towards viewer–centred representations

The effects of changing lighting conditions and viewpoint on the appearance of objects led to the idea that the brain must somehow discount these variations for successful recognition. Illumination changes alter contrasts, shading can hide important features (e.g. surface texture). The visible surfaces of the objects can be of very different reflectances, these essentially altering the light intensity informations received by the retina. Viewpoint changes can lead to the the occlusion of elements of the viewed object, that are potentially crucial for its recognition. Furthermore, the presence of several objects in the scene can lead to the occlusion of some of the objects, thus altering significantly the visible shapes. It seemed impossible, that the brain can operate with such alterable informations on the object world. The constancy of objects, i.e. the fact that they appear as stable entities when the viewpoint, illumination and other scene parameters change, led to the theories on object–centred representation.

The central concept of the theories on object–centred representation is that the objects are internally represented in the brain in a viewpoint–invariant form. Therefore recognition is a process of computing a similar invariant encoding of the viewed object and then comparing it to previously stored representations.

Such theories hypothesised the use of a 3D model and of a coordinate system that is centred on the object. It was assumed that the set of stored models has a modular organisation, as presented in Marr, 1982. In Marr's vision model, the visual system moves from the retinocentric representations (the $2\frac{1}{2}$ D sketch) to an object–centred 3D model, employing coordinate transformations. The matter of choosing a canonical object–centred coordinate system is a focal point in Marr's work, the emphasis being on the objects' principal axis as a base for constructing such a coordinate system. If the visual system used such a representation, it would explain in Marr's view the difficulty of Warrington's subjects (Warrington & Taylor, 1973) in recognising objects from views where the principal axis got severely foreshortened. As an example, a vase could be represented by its axis and very few additional data on the general organisation of its shape around that axis. A complex shape, like that of a tree, would have a much more elaborated representation, from the general elongated shape to the branches and the leaves. Such a model needs to be structured according to the components of the object. The combination of several 3D models into an organisational hierarchy leads to the possibility of building shape descriptions with certain levels of details, depending on the complexity of the shape. Recognition is attempted based on a catalogue of such 3D models, organised hierarchically according to the level of details they represent (in Marr's terms, the precision of the information they carry). During recognition the brain advances from the coarse data (like the principal axis of a model) to more and more detailed models, if necessary.

Another, more recent theory on object–centred representation is the recognition–by–component theory (RBC) described in Biederman, 1990. Its central idea is that objects' views are represented as sets of geometric primitives called geons, together with the relations between them. Such representations are assumed to be computed from spatially stable features, like the non–accidental properties described by Lowe, 1987 (e.g. the number of coterminating edges in a given point, which characterises a vertex independently from viewpoint). The resulting representations are economic, since Biederman showed that a very limited number of geons are needed for describing any 3D shape. He argued, that with only 3 geons of 24 possible types, together with the rela-

tionships between them, a number of 1.4 billion objects can be represented. Hence the RBC model provides an elegant way of describing with very few elements a huge variety of visual entities. Also, such representations are easy to generate, since the visual system only has to process edge information in order to arrive at a set of geons, without computing fine geometric characteristics (e.g. curvatures). The time required for the computation of the geon representation is further reduced by the parallelism of the involved processes, in the RBC model geons being activated and their relationships established in a parallel manner. Since the geons are synthesised from viewpoint–invariant properties, the geons themselves will present viewpoint invariance. Furthermore, these being simple geometrical constructions, they could be easily restored even in the presence of noise, therefore the representations of objects will exhibit a robustness towards visual noise.

If the above theories are correct, there should be no variations of human recognition performance when presented with different views of an object, simply because the internal representations are viewpoint–independent – unless the subject is presented with a view that lacks essential information about the object, hence making recognition difficult or impossible. Indeed, Biederman (1987) pointed out that human subjects' response times did not vary with viewpoint when their task was to identify what general class an objects belongs to (the so–called basic–level classification). This was the very fact that prompted the main ideas of the RBC model. But the subjects' response times and error rates in experiments involving subordinate–level classification (i.e. they were required to identify not just the class of an object, but the particular instance they belong to) was dependent on the viewpoint (Palmer *et al.*, 1981; Tarr & Pinker, 1989). These findings led to the idea that the brain operates with viewpoint–dependent representations, at least at the subordinate level of recognition.

Rosenfeld (1987) suggested, that judging by the amount of time necessary for the recognition of objects, it is unlikely that the brain performs complex operations required for the computation of a viewpoint–invariant 3D model – the brain would have to compute 3D volumetric representations, and arrive at a hypothesis regarding the nature of the viewed object. One is inevitably puzzled by the contradiction that arises from the acceptance of a recognition mechanism that operates with complex transformations and complete geometric representations in 3D space. If the brain can perform all these in such a manner that makes vision so effortless and our mental imagery system 'knows' so well the laws and mathematics of 3D space, then why aren't we so good

at imagining and transforming things in 3D? These operations require a conscious effort from our part. The truth is, operating with mental images of a large number of 3D elements doesn't come easy to us. But the diametrically opposite assumption to object–centred representation, namely that the visual system stores every possible view of every possible object and directly matches the input view to these learnt representations during recognition, is highly unlikely. Let's assume for a moment that we have such an extensive library of views stored for each object. A huge amount of memory would be required for the storage of so many views, and the representation of an object by all its views would be highly redundant – simple shapes would need only a few stored views for their recognition. It is to be noted, that the latter considerations on memory requirements provided the main counter–arguments for the advocates of object–centred representation. If, during recognition, our brain was only performing direct matching of the input stimulus to the previously learnt views of objects, its ability to generalise to novel views would be compromised. But as it is pointed out in Hildreth & Ullman, 1989, we are able to recognise objects from views that are dissimilar to the available (learnt) views of those objects. This suggests that a visual brain operating with viewer–centred representations would use a more sophisticated way of storing object descriptions.

In the case of the class of theories postulating such viewer–centred representations, direct matching between the stored models and the input shape is not possible, since the stored models are themselves viewpoint–dependent. Certain theories assume that in order to discount the effects of viewpoint changes, the brain computes transformations of the viewed shape in order to arrive at a description that can be matched with the internal representations. These ideas pop up in surprisingly mature form in Minsky, 1975, which is remarkable considering the dominance of the object–centred vision trend at that time. In his theory, different representational frames correspond to different views and a set of 'pointers' between these frames correspond to motions that change the viewpoint. Based on experimental evidence, Shepard & Metzler (1971) and Tarr & Pinker (1989) suggested the occurrence of mental rotations of the input shape prior to its matching against stored representations. It has been found that the subjects' response time in object recognition experiments depended on the orientation of the stimulus. The work of Bartram and Jolicoeur (quoted in Biederman, 1990) also showed that rotation of objects in the image plane or in depth slows down recognition. One of the theoretical approaches to the problem was the recognition–by–alignment theory of Ullman (1989). According to this, the visual system estimates the

pose of the viewed object, performs 3D perspective transformations based on this information and aligns the result to the stored representations. The latter operation provides the necessary conditions for the recognition task. It is notable though, that the above theories still assume the existence of 3D viewpoint–specific representations of objects in the brain and/or mental mechanisms performing 3D transformations.

A different approach to viewpoint–dependent representation and recognition consists of theories that exclude 3D models and operations, relying entirely on 2D views stored in the brain. Minsky (1975) argued that incremental mental rotation would work with a few 2D views of objects. This later found an elaborated expression in the theory of linear combination of views, presented by Ullman & Basri, 1991. This hypothesised that during recognition, the visual system attempts to express the input as a linear combination of stored (learnt) views; the success of this operation leads to recognition. From the point of view of the information processing tasks, this linear combination of views intuitively seems to be a much simpler task than the use of pose estimates and complex 3D transformations, as suggested previously.

Poggio & Edelman (1990) proposed a conceptually related model in which interpolation between multiple stored views of an object is performed in order to be able to compare the internal representations with the input shape. They postulate that for each object there is a smooth function that maps any view of that object into a 'standard' view and, more importantly, this function can be synthesised from a number of views of the object. The mentioned work proposes a neural network architecture that carries out these operations. Recent psychophysical evidence is in support of such an approach. Variations of human recognition performance across viewpoints show that objects are recognised easier from certain, so–called canonical views (Palmer *et al.*, 1981; Cutzu & Edelman, 1994); Minsky also hypothesised the existence of a few 'standard' views in the brain for each object. Cutzu and Edelman have demonstrated though, that there are no universally valid canonical views of objects in human vision. These views seem to emerge from the tendency of the subjects to select the features of the viewed object which are of potential diagnostic. A view of an object is likely to become canonical due to the presence of salient features that are visible over a range of viewpoints. The experiments conducted by Edelman also led to results (reported in Edelman & Bulthoff, 1992) that can be interpreted by using the multiple–view interpolation model. Judging by the patterns of the subjects' response times, more views of novel objects are stored in the brain with practice, the recognition of novel objects or of objects from novel views

becoming increasingly easier. The experiments also showed a limited generalisation ability to novel views – fact that points towards the idea of the existence of a finite number of views stored in the brain. Edelman concluded that 3D objects could be represented by collections of specific views, each view being augmented by limited depth information and that during recognition the input is expressed as an interpolation of these. Results of recent psychophysical studies, conducted by Hayward (1998) also support such multiple–view models of representation. His experiments, concentrated on the recognition of outline shapes, confirm the prediction of the multiple–view model according to which recognition performance deteriorates as a function of stimulus dissimilarity.

Edelman & Weinshall (1991) reported success in creating a self–organising neuronal architecture that arrived at multiple–view representations of 3D objects, its behaviour being similar to that of human subjects submitted to psychophysical experiments. Some results of neurophysiological research also seem to support the theories on such a viewpoint–dependent representation. Perrett and others (quoted by Seibert & Waxman, 1992) reported the existence of classes of cells in the brain of macaque monkeys that are activated by certain views of familiar heads and faces. Different views activate different groups of cells and some classes of cells are activated by view transitions. Logothetis *et al.* (1994) found that certain neurons in the monkey infratemporal cortex respond selectively to a particular trained view of an object. Also, as it is described in Niemann *et al.*, 1996, the existence of similar cell groups tuned to particular perspective views of bodies has been proved. This body of evidence is incompatible with the classical theories on object–centred representation.

Based on the above, the proposed machine vision system has been designed to learn sets of views of 3D objects, attempting recognition based on these learnt representations. The main question raised by this is how should these representations be constructed by the system – an issue discussed in the following section.

2.5. Representing features

Research in neurophysiology revealed the fact that a single neuron in the visual system can perform a complex task, being able to signal the presence of a particular structure in the input, such a structure (in a broad sense) being referred to as feature. As Marr, 1982 points out, ganglion cells

in frog retina were found to be able to act as 'bug detectors' by detecting a certain pattern (in this case, small, moving black spots) and discounting irrelevant variations in the input. Such discoveries (Lettvin *et al.*, 1959) played an important role in the emergence of feature–based recognition theories, where the role of vision was to detect features in the image and to recognise the objects based on these.

Minsky (1975) has suggested that in the case of representing objects by sets of views the necessary amount of views could be reduced by using shape primitives and other characteristics. Furthermore, he argued that the visual system, when attempting recognition, would work with features of different saliency. Some features might have a strong 'confirming effect' – the existence of such a feature would immediately lead to an accurate decision regarding the nature of the object. In his theory, "the normal procedure [...] is to gather in sample features until either some satisfaction level is reached and the hypothesis is accepted or until a clear violation or the weight of several minor violation sends the system off in search of something better." He also suggests, that the extraction of features is a task–dependent process.

The validity of such an approach has been denied for decades, since the advocates of object–centred representation argued that the visible world is too complex to be represented based on features (Marr, 1982). Marr admitted that such a scheme could work, but only if "the visual environment can be rigidly constrained – the lighting, the vantage point, the domain of visible elements, and so forth." Although later Biederman's RBC model introduced features like edges and vertices necessary for the activation of geons, a scheme for representing objects by relying purely on image features was considered to be unlikely (Biederman, 1990). Still, he admitted that such a feature–based representation could work if the brain learns multiple versions of these descriptions, with the price of slowing down the recognition process.

Recent developments in research into feature–based vision suggest though, that for recognition, sufficient information is contained in the 2D locations of features detected on the objects' view (Edelman, 1995a). The outcome of the psychophysical experiments set up by Edelman are in support of his theory on 'features of recognition', that unifies the model of basic– and subordinate–level recognition. In these tests, the degree of similarity between object classes was varied in a controlled manner. It was found that as this similarity increased, the viewpoint–dependence of recognition performance became more pronounced. Also, viewpoint–invariant performance occurred even in conditions where two stimuli contained the same geons – in the context of Bieder-

man's RBC model, geon–level differences would have been necessary for this. Furthermore, geon–level differences on their own did not lead to viewpoint–invariant performance. As Hayward, 1998 points out, little behavioural evidence could be obtained on the issue of geon extraction from visual stimuli in psychophysical experiments.

The above findings prove that the viewpoint–dependency of recognition performance varies with the degree of similarity between stimuli. Edelman (1995a) proposed the idea that recognition starts with the extraction of a large variety of image–based features, the system proceeding to synthesise various recognition procedures according to need. Localised features (corner positions, for instance) can help recognition in some circumstances, but make the recognition viewpoint–dependent, since these features are the most likely to be altered by changes in objects' pose. Non–local features like texture can contribute to the viewpoint–invariance of recognition when the local features are not accessible.

This approach strikes a chord with research that has been done not so specifically into vision, but into classification. Sokal (1974) pointed out the existence of individual differences in taxonomic judgement: a group of human classifiers could arrive at correct categorisation of objects (imaginary lifeforms) based on quite different sets of features considered to be salient. In the experiment described by Sokal, the taxonomists, after learning the various instances of the imaginary lifeforms, attempted the classification of newly presented views of these objects. They were able to correctly label them, but the groups of features they used when labelling these objects were different. This leads to the important realisation of the fact that there is no universally valid set of features for classification/recognition of objects. Naturally, a recognition system's task would be easy if there existed features whose presence would be unique to a particular object. The presence of such, so–called critical features in the shape of a viewed object would immediately lead to the correct identification of the object. Detection of such critical features would provide a well–defined practical recipe for the recognition of objects. But what is a critical feature? We are able to recognise an avant-garde piece of furniture as chair, even when it lacks elements that common–sensically would be candidates for the position of critical attribute (like the presence of 4 legs). One has to arrive at the conclusion that the brain is likely to work with much more relaxed rules than the ones provided by a system of critical feature lists.

As Hildreth & Ullman (1989) also pointed out, different visual tasks may require the extraction of different features. A theory on how the visual system performs the extraction of these and their

relations was formulated by Ullman (1984), by introducing the concept of visual routines. These, being "sequences of basic operations that are wired into the system", are assumed to be used by the brain to extract a variety of features, shape properties and spatial relations, based on the early visual representations. Although at that time these ideas were only expressing theoretically possible scenarios of processes occurring in biological vision, they gained significance in the light of the outcome of experiments like the ones conducted by Edelman. It is argued, that representation is a process of schematisation that depends both on the task and on context (Cutzu & Edelman, 1994).

These experiments showed a significant correlation between the subjects' performance and the sum of feature–by–feature 2D distances between the stimuli and the 'best' views – the latter being characterised by shortest response time and lowest recognition error rate of the subjects. Such correlation would suggest once again the use of subject–specific feature patterns in recognition, the brain calculating 2D similarity between stored feature patterns and the one extracted from the stimulus (Edelman, 1995b). The idea of task–dependent emergence of features in categorisation is also supported by studies reviewed by Singh & Landau (1998). The psychophysical studies quoted therein showed again the fact that we 'create' new features in a flexible manner, depending on the visual task.

In Edelman, 1995b, the idea of representation by similarity is introduced. The generic representation scheme proposed therein is based on similarity relative to a small set of available prototypes. As Edelman points out, a general definition of similarity is not available to us – the extent to which two objects are dissimilar or similar can depend on the presence or absence of an infinite number of attributes. Furthermore, there is no way of deciding what these attributes should be, if one proposes to arrive at a universal similarity measure. But for a biological system operating with activations of neurons, the similarity of a number of stimuli actually means the "distance between their representations in the space spanned by the activities of ganglion cells". A similar concept ("perceptual ruler") is proposed by Newell (1998), based on the outcome of her experiments in which the degree of inter–stimulus similarities was controlled. Newell found that in some cases, the visual system seemed to rely on contrasting features between similar objects that would generalise across views. When subjects were presented with dissimilar objects, the system would emphasise the invariant contrasting features. Therefore we might be operating with such distances between objects in representational space, as Edelman, 1995b hypothesised.

These ideas lead to a feature extraction scheme that works with relaxed rules on what features are extracted and how. Imitating the function of units in the visual cortex that respond in a certain way to stimuli located in a particular point of the visual field, the Chorus model proposed by Edelman operates with receptive fields placed on the input image. Contrary to what intuition would tell us, an array of large receptive fields can yield high resolution descriptions of the visual stimuli. The work of Eurich & Schwegler (1997) provides a good quantitative description of the relationship between receptive field size and the obtainable resolution. Hence in Edelman's model, the activations of receptive fields provide the primitive features that lead to the emergence of class prototypes in the system. Such prototypes of object categories consist of stable patterns of primitive features that are recurrent in the input stimuli – hence the primitive features supplied by receptive fields are grouped into higher–level representations by essentially unsupervised processes occurring in the system. Not the representation of views is important any more in Edelman's model, but the representation of similarities of stimuli. This was elaborated later into a model of visual recognition and categorisation (Edelman & Duvdevani–Bar, 1997). Some physiological and psychophysical evidence seems to be in support of such a theory (Cutzu & Edelman, 1996; Sugihara *et al.*, 1998).

This scheme does not rely on exact shape or feature descriptions, in the sense that the information extracted from the input does not contain data on well–defined features like edges, curvatures, textures etc. With these relaxed internal rules and the use of arrays of receptive fields, such a representation scheme can be easily considered a diametrically opposite approach to the 3D object representations that operate with 3D models and it yields a computationally feasible way of dealing with such abstract concepts as similarity.

2.6. Multiscale representations

In parallel with the investigations into the physiology of the visual cortex, a series of vision researchers arrived at the conclusion that biological systems are highly likely to employ multiscale representations of visual information. The emergence of theories on multiscale processing that are assumed to be taking place in the brain can be attributed to the realisation of the fact that details crucial to recognition occur at a wide range of scales. In some cases, very coarse information is sufficient for the correct identification of an object (e.g. general shape), in other circumstances, very fine details (e.g. surface texture) are necessary. For instance, one might want to find out vis-

ually the nature of the material that a chess board is made of, which operation requires fine–scale investigation of surface texture. But in order to get an information on the surface's orientation, a coarser scale processing at the level of the pattern painted on the board is sufficient.

The psychophysical experiments conducted by Campbell & Robson (quoted in Marr,1982) led to the conclusion that the visual system employs a number of spatial frequency–selective channels. Subjects being exposed to grids of various densities and orientations, their sensitivity decayed in a manner specific to the patterns' spatial frequency and orientation. Campbell suggested that fine details might be explored using a high–pass filter, and the overall outline may be derived from a low–pass filtered image. This early idea shows remarkable analogy with the later developed wavelet decompositions of images, as it will become apparent in the next chapter. Studies of the mammalian visual cortex also showed the existence of multiscale processing in the visual system (Daugman, 1980, 1988; Hubel, 1982). It was found that simple cortical cells act as directional bandpass filters and that the receptive fields of these cells consist of areas of given size and orientation – an overview of these findings has been published in Unser & Aldroubi, 1996. Furthermore, research into active vision unveiled the important role of multiresolution analysis in gathering fine visual details. The role of rapid eye movements during the visual investigation of a scene has been established as the means of directing the fovea towards areas of interest in the image (Niemann *et al.*, 1996), thus facilitating the extraction of fine details. Computational modelling of active vision eloquently showed, that low–resolution information is likely to be the driving force directing this process (Burt, 1988; Rao *et al.*, 1996). With this, the brain does not have to build a fine scale description of the whole viewed scene, but can work with finer or coarser details as needed to carry out a particular visual task.

Such findings prompted several theoretical approaches to the problem of multiscale processing in biological vision. Marr (1982) included modules performing multiresolution image analysis into his model of early vision (zero–crossing detection with operators of various sizes), and his model of stereopsis, too, was based on image matching on multiple scales. He underlined, that "the primitives of our representation must work at a number of different scales". Canny (1987) proposed a computational model for multiresolution edge detection, where coarse–to–fine search extracts the relevant information and helps in discounting noise. Relevant structures in the image (groupings of edge elements, for instance) can be extracted by following the evolution of features from coarse to fine representations. The significance of such processing techniques has been un-

derlined in works on low and high level vision (Yuille & Ullman, 1990; Hildreth & Ullman, 1992). These argued that coarse–to–fine search effectively optimises the amount of information that has to be processed by the visual system. Starting with coarse representations, the system can move towards finer details if necessary. The ability of systems operating with such representations to understand noisy images has been demonstrated by practical work (e.g. Sajda *et al.*, 1995).

Considerations of noise reduction, good localisation in space, optimisation of the amount of information required for representation led to several mathematical models of the processes that are supposed to deliver multiscale descriptions of stimuli in biological systems. Hence spline pyramids (Burt & Adelson, 1983), differences of Gaussians (Marr, 1982), Gabor functions (Daugman, 1988) and wavelets (Gaudart *et al.*, 1993; Mallat, 1996) have been proposed as mathematical bases for these processes. All of these models were developed from attempts to match mathematical functions with the response profiles of cells measured in neurophysiological studies. The approach based on Gabor functions (sine or cosine waves modulated with a Gaussian) was perhaps the most hailed, since it provided an analytically similar model to what has been experimentally measured and explained in a consistent way the aspects of spatial frequency selectivity of cortical cells. Also, due to the possibility of creating oriented filters with Gabor functions, this technique provided also a tool for the description of textures. Later, the emergence of wavelets as mathematically elegant means for the computation of multiresolution analysis and the similarities bared by some of them with activation functions measured in biological systems (Gaudart *et al.*, 1993; Unser & Aldroubi, 1996) led to a considerable volume of vision research based on wavelet transforms. The theoretical aspects of wavelet transforms and their applicability in vision is the subject of the next chapter.

2.7. Summary

The main theories on visual information representation have been reviewed, together with the results of related psychophysical and physiological studies. The theories on viewer–centred, feature–based representation have been discussed, pointing out the supporting body of experimental evidence. In the light of these, conclusions regarding the central concepts of the proposed recognition system have been drawn. The final section of the chapter discussed the plausibility of

multiscale image analysis methods in vision, these becoming a core element in the developed system.

The above considerations lead to a model of a 3D recognition system that incorporates in its structure a number of parallel channels, each of which performs feature extraction and grouping processes. The way in which spatial or similarity–based grouping of features occurs is meant to be an essentially unsupervised process, since it has been shown that there are no universal rules that can govern such processes in vision. The resulting representations of features, depending on how well each of them registers salient properties of the input stimulus, contribute in variable extent to the success of the classification process. Multiscale descriptions help in representing potentially salient features on a range of resolutions and these also direct the feature extraction process as an analogy to mechanisms that seem to occur in active vision. As it will be detailed in chapter 4., these ideas provide the ground for the proposed system's structure and ways of operation. The next chapter discusses another important issue that constitutes a central concept in the system, namely multiresolution analysis.

Chapter 3. Wavelet transforms and multiresolution analysis

3.1. Introduction

The present chapter introduces the concepts of wavelet transform and multiscale processing. It describes the mathematical path that leads from the continuous wavelet transform to the A Trous algorithm, which is at the heart of the proposed recognition system's preprocessing and feature extraction modules. In an attempt to unify the often confusing variety of descriptions and conventions found in the literature, this chapter presents in a consistent way the main steps that yield the fundamental equations of continuous and discrete wavelet transform, Mallat's multiresolution algorithm and A Trous transform. The essential advantages/disadvantages of each method in the context of pattern recognition are highlighted, arriving at the reasons for the use of the A Trous algorithm in the proposed application. The final section of this chapter presents a possible implementation of the A Trous transform, that bares attractive properties in the field of pattern recognition applications.

3.2. Fundamentals

In an intuitive formulation, a mathematical transform of a signal (function) leads to a description of the function as a combination of so–called basis functions. A well–established method is the Fourier transform, which leads to a description of a signal as a linear combination of sine and cosine waves of various frequencies. This has the effect of transforming the description of the signal from the time domain into frequency domain.

3.2.1. Localisation in time and frequency. Scale–space

The problem with such transforms is that the basis functions used in the decomposition have infinite duration, i.e. are not localised in time. It is therefore impossible to capture with such a transform local characteristics of the input signal. Attempts like the use of short–term Fourier transform (where a transform is calculated in shifting time windows placed on the signal) did not lead

to mathematically rigourous solutions. Basis functions that are localised in time are needed. In order to be able to capture the signal's frequency contents, these basis functions must also be localised in frequency. But good localisation in time brings poor localisation in frequency– as a basis function becomes more and more compact in time, its spectrum in frequency domain expands. Conversely, as the function dilates in time, its spectrum will contract in frequency domain.

The discovery of wavelets – small waves localised in time and frequency – led to a whole new signal analysis technique, the wavelet transform. This, by decomposing a signal into combinations of dilated and translated versions of the so–called "mother wavelet", can capture details of the signal both in time and frequency. As the wavelet is shrunk in time, it can help describing very fine details of the signal (sharp transitions, for instance) when used as basis function. As it is dilated (expanded in time), its spectrum shrinks in frequency domain, hence when used as basis function, it can describe fine details of the signal's behaviour in frequency. The translation of the wavelet allows the transform to capture details at various time positions in the signal. The transform acts as a mathematical zoom, allowing the investigation of the signal on a particular level of details (Graps, 1995). The difference between the short–term Fourier and the wavelet transform is illustrated in Fig. 3.1.

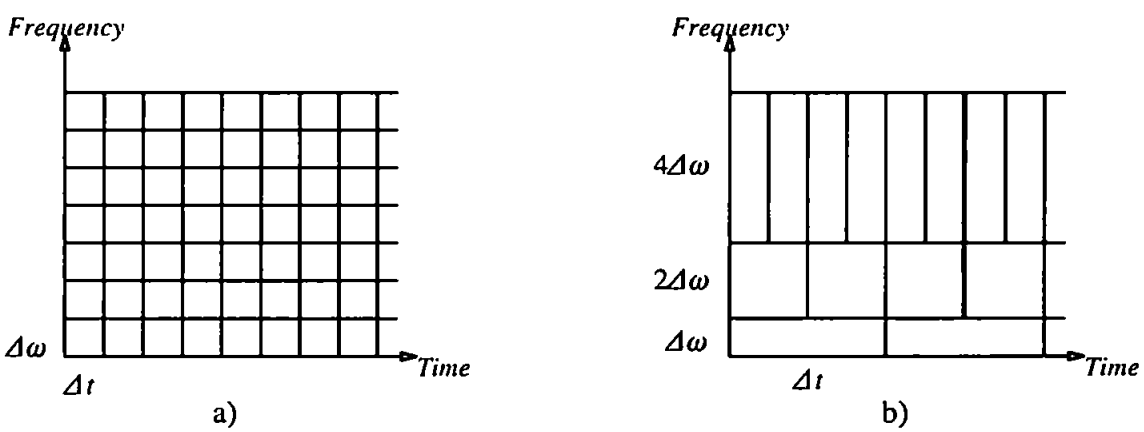


Fig. 3.1. The split of time and frequency space in the case of the a) Short–term Fourier transform. b) Wavelet transform.

The short–term Fourier transform, calculated in time windows of Δt duration, describes the signal in frequency domain in constant $\Delta\omega$ bands. This is a disadvantage when one tries to study the signal's frequency domain details from low to high frequencies. The wavelet transform in its

most widely used form divides the frequency domain description into octave bands: as the resolution doubles (the wavelet shrinks), $\Delta\omega$ doubles.

In wavelet theory, the amount with which the mother wavelet is dilated/shrunk during the analysis is defined by the dilation parameter. As its value increases, the wavelet is dilated more and more, therefore the transform describes coarser and coarser details of the signal. Hence the dilation parameter is directly linked to the resolution (scale) on which the transform operates. When representing the output of the wavelet transform, the scale becomes an extra parameter besides time (or space in image processing).

With the introduction of scale, one arrives at the concept of scale–space, illustrated in Fig. 3.2. in the case of the 2D wavelet transform. As the scale parameter increases, the representation becomes coarser, the resolution decreases.

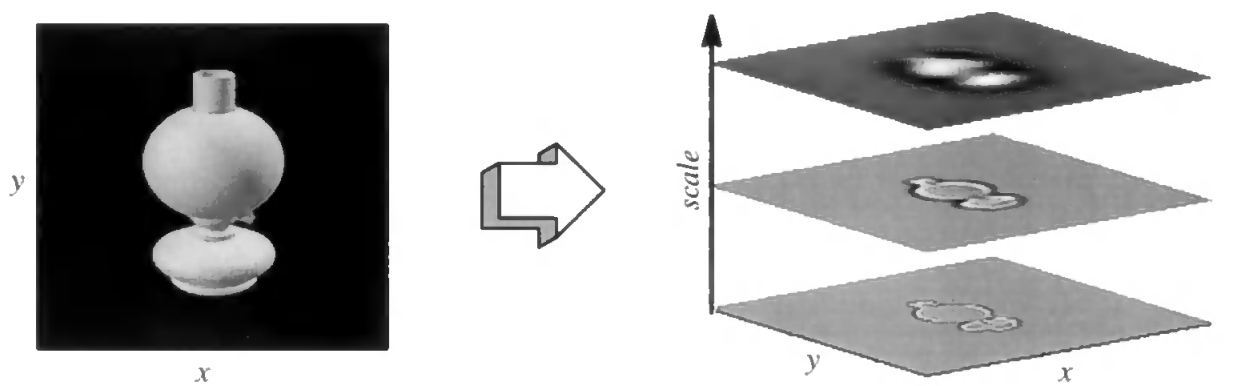


Fig. 3.2. Scale–space decomposition of an image. Scale becomes an extra coordinate.

In image processing applications of the wavelet transform, the scale appears as a third coordinate, as shown in the above figure. On each plane corresponding to a value of the scale coordinate, the signal (image) is represented with a given level of details. Various coarse–to–fine scale feature search and topological mapping operations can be imagined in such a 3–dimensional space. As it will be described in the next chapter, the proposed system operates with such a scale–space representation.

3.2.2. Other multiscale analysis methods

In order to produce multiresolution descriptions of the input data, it is not necessary to use wave-

let transform. In a trivial case, simple decimation of the signal's samples leads to an increasingly poor representation of the data. Other simple mathematical methods involve successive filtering of the input with low pass filters, yielding smoothed versions of the signal, hence loosing more and more details as the algorithm proceeds.

In practical applications, smoothing operations like median filtering (Starck *et al.*, 1995a) have been used. Canny (1987) proposed Gaussian averaging prior to extraction of edges in order to reduce noise. The scheme proposed by Burt & Adelson (1983) utilises the Laplacian of Gaussian as operator, in which case the Gaussian smoothes out details that are smaller than a chosen size and the second–order derivative has the effect of enhancing changes in the smoothed signal. Spline–derived filters for image approximations (Unser *et al*, 1993) have been also used. The main problem with these methods is the redundancy of the resulting representation. Details of a particular size remain detectable on successive scale planes in the multiresolution representation. As Mallat, 1989 also points out, there is correlation between the data on multiple levels of the multiscale description, hence scales are not properly separated. The use of Gabor filters (Porat & Zeevi, 1988) do not lead to basis functions and the representation is not complete: with Gabor functions, the spatial frequency plane is not completely covered and this means that one might miss out details of the signal's frequency behaviour. A tessellation of the frequency plane that would decrease the loss of details requires a huge number of operators of various sizes and orientations. The problem becomes computationally difficult.

These problems disappear in wavelet transform–based multiresolution analysis. Since it provides basis functions for the description of a signal, it leads to complete representations in both frequency and time (space) domain. It has also the advantage, that singularities characterising events at a given level of detail in the signal can be detected from the wavelet transform. These issues will be discussed in the following sections.

3.3. The continuous wavelet transform

The continuous wavelet transform of a square–integrable function $f(x)$ is by definition:

$$W[f(x)](a,b) = \frac{1}{\sqrt{a}} \int \overline{\psi}\left(\frac{x-b}{a}\right) f(x)dx \qquad (3.1.)$$

where $\overline{\psi}$ is the complex conjugate of the mother wavelet ψ, a is the dilation and b is the translation parameter; a and b can vary continuously. The term $\sqrt{a}$ preserves the energy (i.e., satisfies the equation $\| \psi_{a,b} \|^2 = \| \psi \|^2$, where $\psi_{a,b}$ is the wavelet function on scale a, translated with b and $\| . \|$ is the standard L_2 norm). With this scaling factor, the wavelet function occupies the same area on all scales. Also, the wavelet must have a finite energy ($\| \psi \|^2 < \infty$), which is satisfied if the function is localised in space. ψ must be square integrable and must satisfy the condition:

$$C = \int \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \tag{3.2.}$$

where $\hat{\psi}(\omega)$ is the Fourier transform of the wavelet function. This condition means that the wavelet must be localised in frequency domain, too. If these conditions are satisfied, the wavelet transform is *invertible* on its range; the inverse transform is :

$$f(x) = \frac{1}{C} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W[f(x)](a,b) \psi\left(\frac{x-b}{a}\right) \frac{da db}{a^2} \tag{3.3.}$$

A detailed discussion of the admissibility conditions and the restrictions posed on the wavelet function can be found in Shensa, 1992; Jawerth & Sweldens, 1994.

An illustration of the dilated/translated versions of a wavelet is shown in Fig. 3.3.

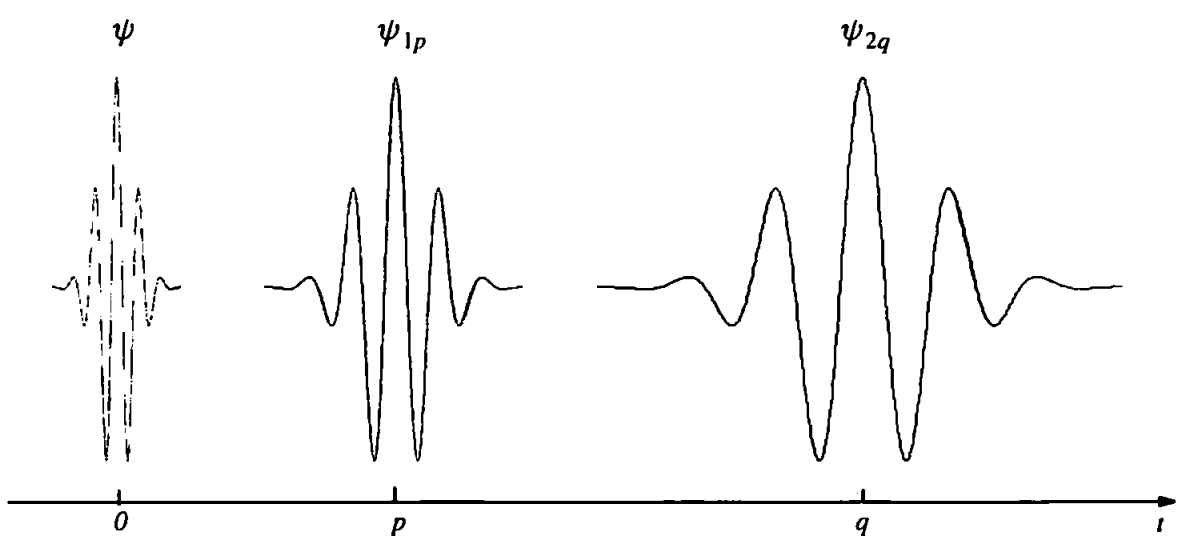


Fig. 3.3. A mother wavelet ψ and its dilated/translated versions.

The main properties of the transform are as follows:

1. it is a linear transformation
2. it is invariant under translation:

$$W[f(x-k)](a,b) = W[f(x)](a,b-k) \tag{3.4.}$$

3. it is invariant under dilations:

$$W[f(nx)](a,b) = \frac{1}{\sqrt{n}} W[f(x)](na,nb) \tag{3.5.}$$

It is important to note the fact that in the wavelet literature, the term 'invariance' is used in the above described less–strict sense. The first property is evident from the definition of the transform. The other two properties result easily from the definition and changes of variable in the integral ($u = x-k$ and $u=nx$, respectively). The invariance under translation is an important feature of the transform, since it means that a shift in the input signal leads only to the same shift of the transform. This property is crucial to pattern recognition applications – if the transform does not exhibit invariance under translation, then any shift of a pattern in the input will lead to an entirely different output, making detection based on the transformed signal impossible. Therefore in practice, it is important to search for a discrete version of the wavelet transform that inherits this property, as it will be pointed out in the following sections. The transform's invariance under dilations means that it can be used to 'zoom in' on a signal's finer details, without altering the properties of the transform.

One practical way of arriving at the discrete version of the wavelet transform is multiresolution analysis. Its theoretical aspects and the resulting discrete wavelet transform (DWT) are discussed in the following section.

3.4. The discrete wavelet transform

In practice, the input is not available as a continuous signal, but as a series of samples resulting from the digitisation of the analysed signal. Therefore a discrete wavelet transform is needed, that can be implemented on a computer platform.

3.4.1. Definition

One can arrive at the definition of the discrete wavelet transform by using discrete values for a and b in equation (3.1.). A practical approach is the use of the so–called dyadic grid, where the dilation parameter is $a = 2^j$, and the translation b is a multiple of a , i.e. $b = 2^j k$, $j, k \in \mathbf{Z}$. In this case, equation (3.1.) becomes:

$$W_{jk}[f(x)] = \frac{1}{\sqrt{2^j}} \int \overline{\psi}\left(\frac{x}{2^j} - k\right) f(x)dx \qquad (3.6.)$$

The parameter j is usually referred to as scale. In practice, f is the sampled input signal; it is a function constant on unit intervals, its frequency band is $[0,\pi]$, $\omega = 2\pi$ being the Nyquist rate. In the implementation of this "classic" discrete wavelet transform, the aim is to obtain an algorithm, that doesn't require the analytic expression of the mother wavelet ψ during the computation of the transform. A way of solving this problem is Mallat's multiresolution analysis (Mallat, 1989).

3.4.2. Mallat's algorithm

3.4.2.1. The mathematics

The core concept of multiresolution analysis (MRA) is the splitting of the space of square integrable functions (i.e. $L^2(\boldsymbol{R})$) into closed subspaces $V_j \subset L^2(\boldsymbol{R}), j \in \mathbf{Z}$. Each subspace V_j consists of all possible approximations of functions $f \in L^2(\boldsymbol{R})$ at the resolution j. The multiresolution analysis of a function f consists of its approximation with orthogonal projections of f onto the series of subspaces V_j. As a convention for the following descriptions, scale index *0* will denote the finest resolution, and as the index increases, the decomposition moves towards coarser representations of the input data.

Modifying Mallat's definitions according to the adopted conventions, the sequence of decreasing subspaces $V_j \subset L^2(\boldsymbol{R}), j \in \mathbf{Z}$ are defined by the following properties:

i. $V_{j+1} \subset V_j$;

ii. $s(2x) \in V_j \Leftrightarrow s(x) \in V_{j+1}$;

iii. $s(x) \in V_0 \Leftrightarrow s(x+1) \in V_0$;

iv. $\lim_{j \to \infty} V_j = \{0\}$

v. $\lim_{j \to -\infty} V_j$ is dense in $L^2(\boldsymbol{R})$;

The first property expresses the fact that a subspace associated with a higher resolution j contains all possible approximations at the lower resolution $j+1$ of functions f. The operation of approximation must be similar at all resolutions – condition expressed by (ii). It means that when passing from a resolution to another, the approximation of f can be written with compressions/dilations of functions s contained in the subspaces associated with each considered resolution. Naturally, as (iii) states it, any translation of a function $s \in V_j$ must also reside in V_j. As resolution decreases (j increases), the subspace associated with it finally becomes empty (iv); conversely, as the resolution increases, the approximation of f based on the associated subspace converges to f, as (v) describes it.

Furthermore, Mallat (1989) proved that there is a unique function ϕ (called scaling function), whose translations and dilations constitute orthonormal bases of V_j. This fact is of fundamental importance in arriving at the discrete wavelet transform, without explicitly knowing the mother wavelet, as it will be shown below. It means that the approximations of a function f that are orthogonal projections onto V_j can be written as scalar products between f and the translated/dilated versions of the scaling function. If the approximation of f on a scale $j+1$ and position k is denoted $c_{j+1,k}$, then:

$$c_{j+1,k} = < f \ , \ \phi_{j+1,k} > \tag{3.7.}$$

where the basis functions spanning V_{j+1} are:

$$\phi_{j+1,k}(x) = \frac{1}{\sqrt{2^{j+1}}}\phi\left(\frac{x}{2^{j+1}} - k\right) \tag{3.8.}$$

According to the adopted conventions, $\phi \equiv \phi_0$ is the scaling function associated with the finest scale subspace V_0 – the latter contains all functions in $L^2(\boldsymbol{R})$ that are constant on unit interval. If ϕ_0 is supported on [0, N], then ϕ_1 is supported on [0, 2N], and so forth. The frequency spectrum of ϕ_0 occupies the $[0, \pi]$ band. Dilations of ϕ bring contractions in frequency domain, thus the spectrum of ϕ_j will be localised in the band $\left[0, \frac{\pi}{2^j}\right]$.

Since in the light of property (i), $\phi_{j+1,k}$ can be written as a linear combination of the basis functions of V_j, the following equation results:

$$\begin{aligned} \phi_{j+1,k}(x) &= \sum_l < \phi_{j+1,k}(u) \ , \ \phi_{j,l}(u) > \phi_{j,l}(x) = \\ &= \sum_l < \frac{1}{\sqrt{2^{j+1}}}\phi\left(\frac{u}{2^{j+1}} - k\right), \frac{1}{\sqrt{2^j}}\phi\left(\frac{u}{2^j} - l\right) > \frac{1}{\sqrt{2^j}}\phi\left(\frac{x}{2^j} - l\right) \end{aligned} \tag{3.9.}$$

With a change of variable, i.e. making $v = u - 2^{j+1}k$ (hence $dv = du$) in the scalar product integral, the above expression becomes:

$$\phi_{j+1,k}(x) = \sum_l < \frac{1}{\sqrt{2^{j+1}}}\phi\left(\frac{v}{2^{j+1}}\right), \frac{1}{\sqrt{2^j}}\phi\left(\frac{v}{2^j} + 2k - l\right) > \frac{1}{\sqrt{2^j}}\phi\left(\frac{x}{2^j} - l\right) \tag{3.10.}$$

By making $x = \frac{v}{2^j}$ in the scalar product and keeping in mind that in the corresponding integral $dx = \frac{dv}{2^j}$, the scaling function on scale $j+1$ can be written as:

$$\begin{aligned} \phi_{j+1,k}(x) &= \sum_l < \frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right), \phi(x + 2k - l) > \frac{1}{\sqrt{2^j}}\phi\left(\frac{x}{2^j} - l\right) = \\ &= \sum_l < \phi_{1,0}(x), \phi_{0,l-2k}(x) > \phi_{j,l}(x) = \sum_l h_{l-2k}\phi_{j,l}(x) \end{aligned} \tag{3.11.}$$

where h_{l-2k} denotes the scalar product that is independent of j. The scalar product of f and a sum of terms being equal to the sum of scalar products of f and the terms, equations (3.7.) and (3.11.),

by making $m = l - 2k$, yield the equation that expresses the approximation of the function f on scale $j+1$ as a combination of the approximations on scale j:

$$c_{j+1,k} = \sum_m h_m c_{j,m+2k} \tag{3.12.}$$

This allows the iterative computation of the approximations of f on successively coarser resolutions. Another fundamental equation in wavelet theory results neatly from (3.11.) and (3.8.), for $j=0$ and $k=0$:

$$\frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right) = \sum_m h_m \phi(x - m) \tag{3.13.}$$

This is the *dilation equation*, also called the two–scale difference equation, since it expresses the relationship between two scaling functions on two consecutive resolutions (scales). It is a direct consequence of the condition (i) and it is an essential equation, since it introduces filters. Having the scaling function known explicitly, from the dilation equation the filter coefficients h_m can be obtained – these represent the impulse response of the so–called smoothing filter usually denoted H, that has low pass character. Its pass band is by definition $\left[0, \frac{\pi}{2}\right]$ – it is a so–called half–band filter. A detailed discussion of multiresolution analysis using filter–bank analogy can be found in Strang & Nguyen, 1996; Vetterli & Herley, 1992.

Then (3.12.) yields the approximations c_{jk} that are also called smoothed coefficients, since they result from successive smoothing of the input data with the H filter. But this successive smoothing leads to loss of information (detail) at each stage of the iteration. In Mallat's MRA, this detail is calculated at each resolution by projecting f onto the orthogonal complements of V_j, which will be denoted here W_j. By definition:

$$V_j = V_{j+1} \oplus W_{j+1} \quad ; \quad W_{j+1} \perp V_{j+1} \tag{3.14.}$$

Basically, a projection onto W_{j+1} represents the detail lost when passing from scale j to $j+1$ in equation (3.12.). A subspace W_j is spanned by translated and dilated versions of the wavelet function ψ (Mallat, 1989; Strang & Nguyen, 1996), denoted ψ_{jk}. When $W_j \perp V_j$, then ψ is an orthogo-

nal wavelet (Daubechies, 1988). Since the spectrum of ϕ_j occupies the band $\left[0, \frac{\pi}{2^j}\right]$, from (3.14.) results, that ψ_j is localised in frequency in the band $\left[\frac{\pi}{2^j}, \frac{\pi}{2^{j-1}}\right]$. Practically, the subspaces V_j and W_j split in two equal parts the band of the signal on scale j. The higher frequencies responsible for the details on a particular scale will be captured by W_j.

The advantage of Mallat's algorithm is that one can calculate the detail (wavelet) coefficients without using the analytical expression of the wavelet function. The detail d_{jk} can be written as:

$$d_{jk} = < f , \psi_{jk} > \tag{3.15.}$$

From (3.14.) results immediately, that $W_{j+1} \subset V_j$, therefore $\psi_{j+1,k}$ can be written as a linear combination of ϕ_{jk}, in very similar manner to (3.9.). Following the steps outlined in (3.9.) and (3.11.), the wavelet coefficients of the decomposition can be written as:

$$d_{j+1,k} = \sum_m g_m c_{j,m+2k} \tag{3.16.}$$

where g_m is the impulse response of the G detail filter (of high pass character), with pass band $[\frac{\pi}{2}, \pi]$. The frequency characteristics of the H and G filters that are met in practice are illustrated in Fig. 3.4.

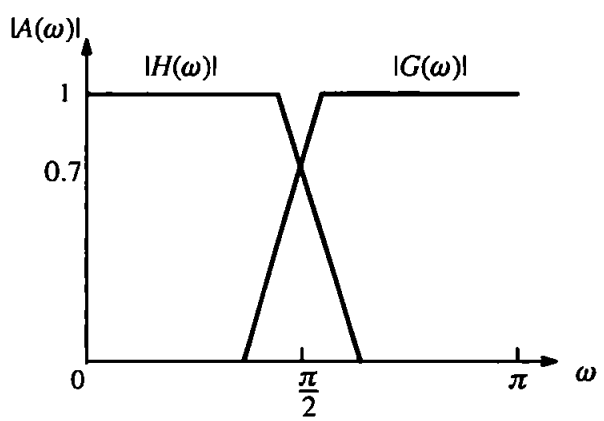


Fig. 3.4. Frequency characteristics of the smoothing H and detail G filters, in a realistic case.

The equations (3.12.) and (3.16.) describe the discrete wavelet transform. These show eloquently, that downsampling with a factor of 2 occurs at each stage of the decomposition, only half of the coefficients from the previous scale entering the calculations (due to the $2k$ term in the coefficients' index). The way in which the coefficients are decimated becomes apparent from the graphic illustration of the filtering process, shown in Fig. 3.5. for two stages of the algorithm. For the purpose of this illustration, a filter length of 3 coefficients was assumed. While on scale level 1, the coefficient indexes advance with 1, the level 0 coefficients are taken into calculation by advancing the filter by steps of 2. Hence the decimation of the data.

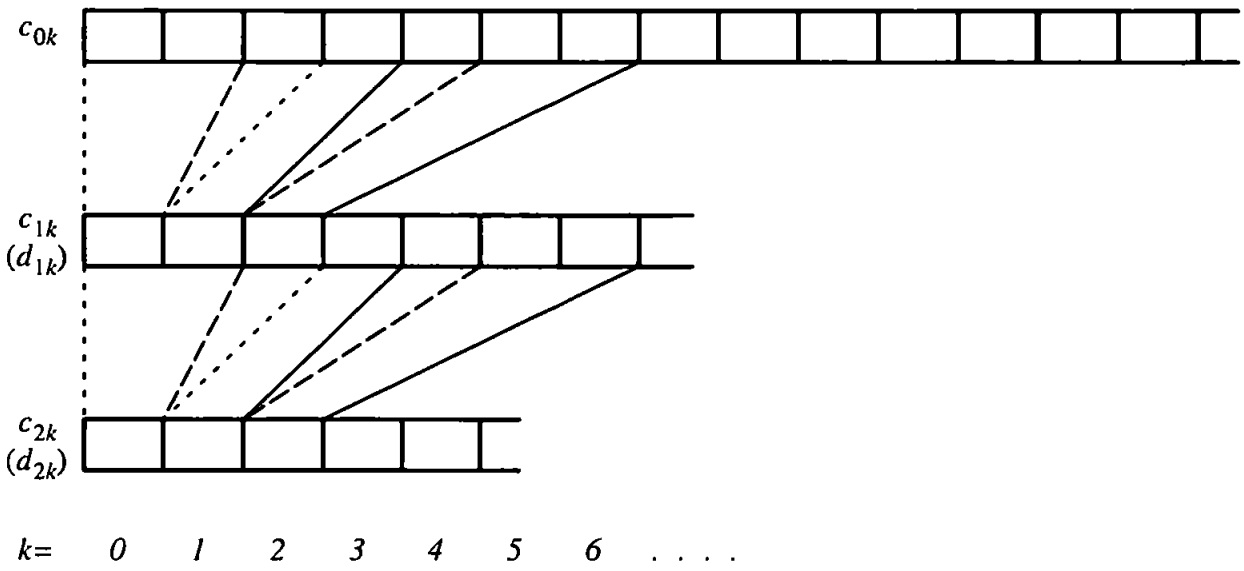


Fig. 3.5. First two stages of the Mallat algorithm

A schematic representation of Mallat's MRA that yields the DWT is shown in Fig. 3.6.

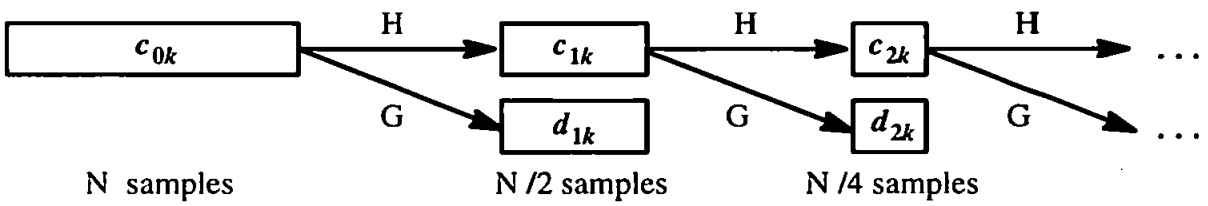


Fig. 3.6. The cascade decomposition in Mallat's algorithm. On each stage, the filters are applied on decimated data.

At each stage of the algorithm, the filters H and G are applied to downsampled data. But downsampling with a factor of 2 (contraction in time) brings dilation in frequency, the band of c_{jk} being

doubled before the filters are applied. The filters having constant pass bands, at each stage of the decomposition, the absolute bandwidth of the signal let through halves. It is trivial to deduct that c_{jk} captures the events in the signal in the band $\left[0, \frac{\pi}{2^j}\right]$ and d_{jk} in the band $\left[\frac{\pi}{2^j}, \frac{\pi}{2^{j-1}}\right]$. With this, the description of the connection between the subspaces V_j and W_j, the basis functions ϕ_{jk} and ψ_{jk} and the filters H and G is complete.

It is only necessary to keep the detail coefficients d_{jk} and the final smoothed data c_{Jk} on a chosen coarsest scale J, since c_{0k} can be reconstructed perfectly based on these with the use of conjugated (reconstruction) filters (Mallat, 1989; Daubechies, 1989).

In practice, the h_m filter coefficients are calculated from the dilation equation, and G is calculated as a quadrature mirror filter (Shensa, 1992; Strang & Nguyen, 1996). Naturally, these filters are expected to be causal (i.e. in the language of signal processing, they do not anticipate data samples, an event in the input signal causing an effect in the output with a delay of a certain number of samples).

The conditions they must satisfy besides causality result as follows:

1. H is low pass filter $\Rightarrow$ DC component of the signal ($\omega = 0$) is unaltered $\Rightarrow$ H(0) = 1 $\Rightarrow$ $\sum_m h_m = 1$ (low pass condition)
2. G is high pass filter $\Rightarrow$ DC component does not go through $\Rightarrow$ G(0) = 0 $\Rightarrow$ $\sum_m g_m = 0$ (high pass condition)

3.4.2.2. Properties

i. As it has been described, the set c_{0k} is the starting point in the calculation of c_{jk} and d_{jk}; c_{0k} must lie in subspace V_0 – but in practice, ϕ is not used explicitly to project f onto V_0, like in (3.7.). Instead, an initial approximation is used; the easiest choice is to set c_{0k} equal to the samples of f. But to have an exact wavelet transform, i.e. d_{jk} to be the samples of the continuous wavelet transform and g_m to be exactly the samples of the wavelet function, several conditions must be satisfied (Shensa, 1992). This problem of initial approximation disappears in the case of the A Trous algorithm.

ii. A translation–variant wavelet transform resulted, due to the downsampling that occurs at each decomposition stage. Therefore this is a decimated DWT. A shift of the input samples with an amount that is not a multiple of 2^j leads to a complete alteration of the smoothed and detail coefficients at scale j, as it is evident from the equations (3.12.) and (3.16.).

iii. An extension of the algorithm can be done, by applying the filters not only to the smoothed data, but also to the detail data at each iteration. This leads to the discrete wavelet packet transform (DWPT), which is an overcomplete representation of the signal. It yields 4 sets of coefficients on each scale plane. With particular selection of bases and coefficient packets, this transformation can be rendered translation–invariant (Cohen *et al.*, 1997). Shensa, 1992 describes other sophisticated ways of obtaining a non–decimated DWT.

iv. A major disadvantage of the transform is the fact that when extended to 2D, it produces multiple sets of detail coefficients on each scale. This problem is characteristic to both decimated and non–decimated DWTs. Practically, it means that detection of features from these coefficients is made difficult – any interpretation of the coefficients must be done by taking into consideration several sets of these on more than one scale. These issues are treated in the next two sections.

3.4.2.3. Extension to 2D

Any form of discrete wavelet transform can be easily extended to 2D by separating the variables. Practically it means that one can apply the smoothing and detail filters successively along the horizontal and vertical dimensions of the 2D signal (image). A mathematically more rigourous alternative is to arrive at 2D filter kernels, by calculating the tensor products $h \otimes h$ and $g \otimes g$. Both solutions are detailed in Mallat, 1989.

In practice, the successive filtering is preferred, since it is easier to implement based on the 1D transform and it reduces the computational load. Mallat, 1989 proves the validity of the main theoretical aspects of multiresolution analysis in this case of separated variables and 2D input data. Starting with the input image, one stage of the transform is outlined in Fig. 3.7.

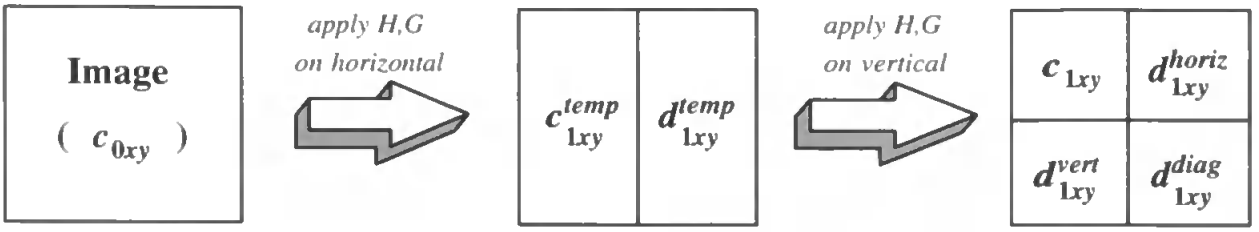


Fig. 3.7. The 2D version of Mallat's algorithm.

The decomposition starts by applying the smoothing and detail filters along the horizontal direction. This leads to the intermediate sets of smoothed and detail coefficients c^{temp}_{1xy} and d^{temp}_{1xy}. These sets, due to downsampling, have only half of the size of the set c_{0xy}. After applying the filters along the vertical direction, besides the smoothed coefficients c_{1xy}, 3 sets of detail coefficients result, as shown in Fig. 3.7. These form the so–called coefficient plane; the detail coefficients pick up details along the horizontal, vertical and diagonal directions. The analysis continues by applying the filters in the same way to c_{1xy}. An example of an image decomposition is shown in Fig. 3.8.

a) b)

Fig. 3.8. A 2–level wavelet decomposition using the 2D version of Mallat's algorithm.
a) The original image b) The 2 levels of detail coefficients and the smoothed data.

Since at each stage, the number of data samples to process is reduced by a factor of 4, the algorithm considerably speeds up as it advances towards coarser scales. Hence this transform algorithm is often called fast wavelet transform (FWT) in the literature, as an analogy to the fast

Fourier transform. When the decomposition is stopped at a level corresponding to the coarsest scale, the resulting set of coefficient planes will have the same size as the input image.

3.4.2.4. Problems with the decimated DWT

As it has been mentioned before, the lack of translation invariance is a serious disadvantage when pattern analysis is attempted based on wavelet coefficients. Mallat himself (in Mallat, 1996) states that wavelet bases have not yet found any application for visual pattern recognition, because of this drawback.

Another disadvantage of the MRA algorithm is that in the 2D case, using the smoothing and detail filters on x and y directions separately, 3 sets of wavelet coefficients are obtained on scale $j+1$ for one set of coefficients on scale j. This, together with the fact that downsampling occurs, renders impossible the interpretation of a wavelet coefficient at a certain scale independently from other scales. The separation of scales, not accomplished by this transform, is a crucial requirement in the case of multiresolution image analysis for pattern recognition. These problems are described in Bijaoui *et al.*, 1994; Allen *et al.*, 1993. Even transforms that deliver translation–invariance (like the previously mentioned DWPT) do not provide the possibility of a one–to–one mapping between wavelet coefficients and image pixels. The DWPT, for instance, yields 4^n sets of coefficients at step n of the decomposition.

As it will be shown in the following section of this chapter, these problems disappear in the case of the non–decimated A Trous algorithm. The advantages of this compared to some non–decimated DWT algorithms are also pointed out.

3.4.3. The A Trous algorithm

In contrast with the MRA algorithm described in the previous section, the A Trous algorithm (Holschneider *et al.*, 1989; Dutilleux, 1989) leads to a non–orthogonal and non–decimated wavelet transform. The restrictions on the scaling function ϕ, on the h and g filters are much more relaxed, orthogonality is no longer required.

3.4.3.1. Definitions

The fundamental property of the A Trous algorithm is that the involved smoothing (low pass)

filter h is an interpolating filter. In the discrete case, as the corresponding scaling function dilates, zeros are inserted among the samples of the function. The subtle mathematical implications of this and of the non–orthogonality of the transform are treated in Shensa, 1992. The name of the algorithm ("with holes") comes exactly from the fact that the filter h interpolates between samples: leaves the even points on each scale fixed, and obtains the odd points by interpolating – therefore no downsampling occurs. From the algorithm's point of view (hence not counting the filter properties, that differ from the orthogonal MRA), A Trous is equivalent to an MRA, where on every scale the subsets of odd and even samples are submitted to the discrete wavelet transform described in the previous section, and the two sets of outputs are interlaced, thus they become the odd and even samples of the output. This can be seen from the analytical description of A Trous, detailed below.

The essential difference from Mallat's MRA in choosing the discrete values for a and b is that the translation parameter b is no longer a multiple of a (i.e. $b = k$, $k \in \mathbf{Z}$ in this case) – this means that the algorithm will produce output for every input sample position on a certain scale. Therefore, the integral form of the transform can be written as:

$$W_{jk}[f(x)] = \frac{1}{\sqrt{2^j}} \int \overline{\psi}\left(\frac{x-k}{2^j}\right) f(x)dx \tag{3.17.}$$

The associated scaling function is in this case:

$$\phi_{jk}(x) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{x-k}{2^j}\right) \tag{3.18.}$$

By using equations (3.7.) and (3.18.), substituting the new scaling function into (3.9.) and making new variable changes ($v = u - k$ and $x = \frac{v}{2^j}$ in the scalar product integral, $m = \frac{l-k}{2^j}$ in the sum), in a similar way to MRA, one arrives at:

$$c_{j+1,k} = < f, \phi_{j+1,k} > = \sum_m h_m c_{j,k+2^j m} \tag{3.19.}$$

$$d_{j+1,k} = < f, \psi_{j+1,k} > = \sum_m g_m c_{j,k+2^j m} \qquad (3.20.)$$

It is evident, that no downsampling occurs. The additive term $2^j m$ in the index of the smoothed (detail) coefficients makes the filters skip $2^j - 1$ coefficients when applied to data on scale j. Therefore the filter H (G) on a scale j acts like one that has $2^j - 1$ zeros inserted between each pair of its h (g) coefficients. This becomes apparent from the illustration of the way in which the coefficients enter the calculations (Fig. 3.9.).

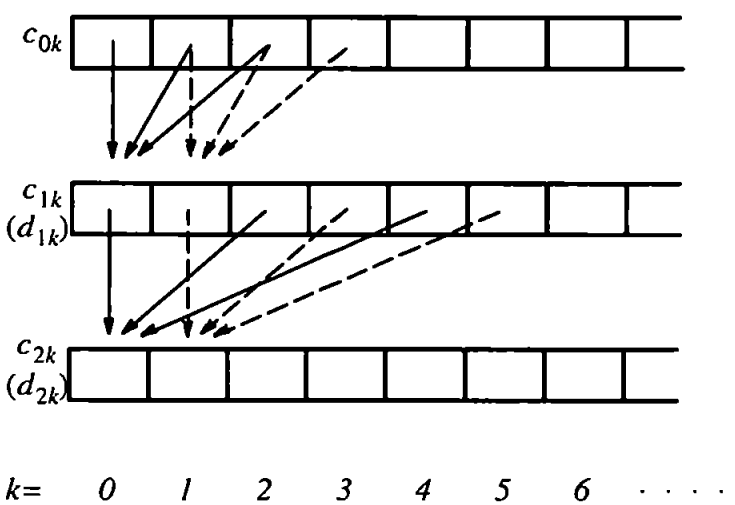


Fig. 3.9. The A Trous algorithm's first two stages.

A schematic representation of the algorithm is shown in Fig. 3.10. In practice, due to the increasing width of the filters that are applied to the data, periodicisation of the data is used (mirroring on boundaries or modulo arithmetic that produces a wrap–around of the sample indexes).

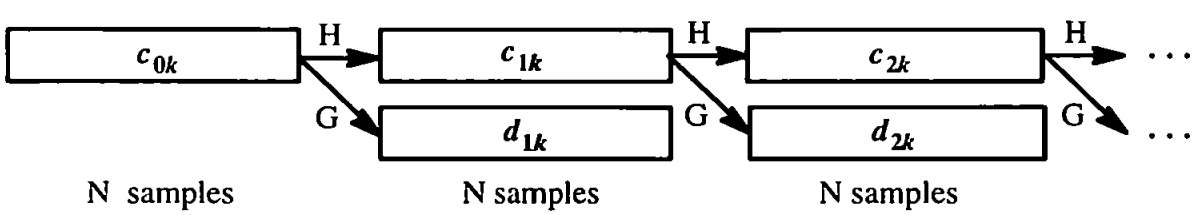


Fig. 3.10. The cascade decomposition in the A Trous algorithm.

The halving of the bandwidth of c_{jk} and d_{jk} from one stage to the next is not achieved by applying constant bandwidth filters to dilated spectrums, as it happens in the case of MRA. Instead, the insertion of zeros between the pairs of filter coefficients at each stage produces essentially an up-sampling of the filter coefficients, therefore the filters' bandwidth halves.

Reconstruction can be obtained with the use of conjugated filters, by using the d_{jk} detail coefficients and the final smoothed data c_{Jk} (J denoting the considered coarsest scale where the analysis was stopped).

3.4.3.2. Properties

i. In practice, c_{0k} is set equal to the samples of f; if h is A Trous (i.e. interpolating) filter, then with this initial setup, the A Trous algorithm is an exact implementation of the discrete wavelet transform. Furthermore, the detail filter coefficients (g_m) are samples of the wavelet function ψ. These two properties, that prove to be essential when it comes to interpreting the physical meaning of the A Trous transform (like using filter bank analogy inherited from the "classic" MRA), are proved in Shensa, 1992.

ii. The transform is translation invariant, as equations (3.19.) and (3.20.) show it. Decimation of data not occurring during the iterations, for N input samples N smoothed and detail coefficients result on each scale (as it is apparent from Fig. 3.10.).

iii. The construction of G from H is possible here, too. As Bijaoui *et al.*, 1994 and Shensa, 1992 point out, the conditions that these filters and their conjugates have to satisfy are much more relaxed than in the case of Mallat's algorithm. This allows the construction of computationally extremely elegant filters, that can even lead to the computation of the detail coefficients without filtering, hence minimising computational load. Such a construction is described in the following chapter detailing the structure of the designed recognition system.

3.4.3.3. The A Trous algorithm in 2D

The extension to 2D can be done in the same way as in the case of Mallat's algorithm, by separating the variables. In the case of the 2D A Trous transform, on every scale an array of coefficients results, that has the same size as the image. Due to this property, a one–to–one mapping between

pixels and wavelet coefficients is possible. Therefore the coefficients can be interpreted on a certain scale without having to take into consideration coefficients on other wavelet planes – detection of certain features in the image from wavelet coefficients becomes possible. In this sense, A Trous is one of the rare wavelet transforms, that respect the scale separation property (Starck *et al.*, 1995a). A schematic representation of one stage of decomposition is showed in Fig. 3.11.

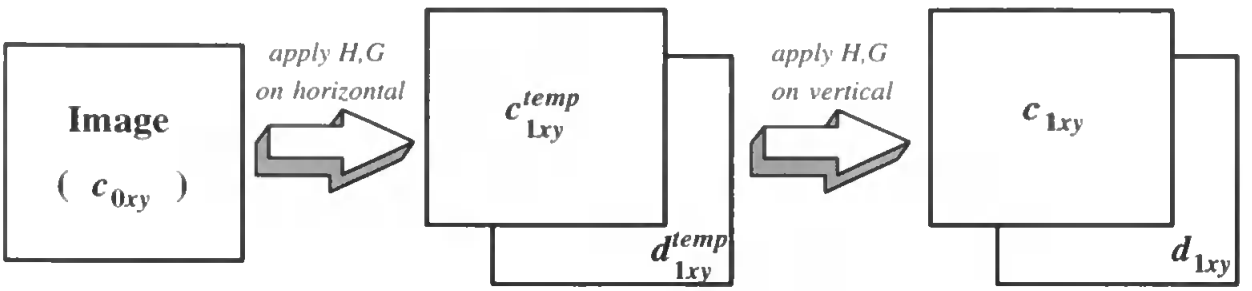


Fig. 3.11. The 2D version of the A Trous algorithm.

An example of image decomposition using a spline scaling function and the derived wavelet is shown in Fig. 3.12.

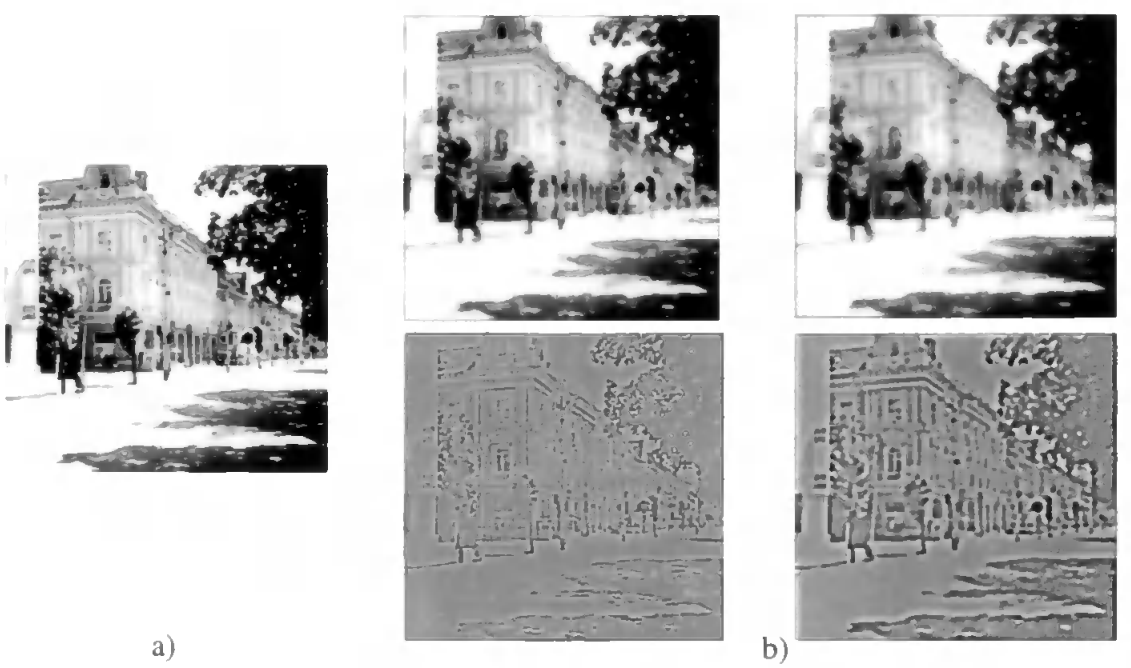


Fig. 3.12. First 2 levels of the A Trous decomposition of an image.
a) The original image. b) The resulting coefficient planes
(top row: smoothed coefficients; bottom row: detail coefficients)

It is visible how coarser and coarser details are picked up by the detail coefficients (the bottom row), while the smoothed coefficients (top row) provide only increasingly blurred versions of the original input.

An important property of the wavelet transform is that the detail (wavelet) coefficients, resulted from applying a high pass filter on the data, register various events in the image that can be of interest in pattern recognition application. The high pass filter acts as a derivative operator, hence singularities in the detail coefficients (like local maxima) can show the location of edges, corners, high–density texture regions in the image. In the case of such a non–decimated transform like A Trous, the detection of such features in the image based on a set of wavelet coefficients on a scale j becomes possible, since for each image pixel there is one corresponding wavelet coefficient. An event detected in a wavelet coefficient plane associated with scale j can be mapped back onto a location in the input image. One can follow directly the evolution of the image from one scale plane to the next (Starck *et al.*, 1995b).

A detailed discussion on the ways in which image singularities are reflected in wavelet coefficient planes can be found in Mallat & Hwang, 1992. This paper demonstrates, that local maxima in the wavelet transform's coefficient planes characterise all singularities of the input signal. Detection of sharp transitions in the image (characteristic to edges, corners), fast oscillations (dense textures) can be performed by local maxima search on the detail coefficient planes d_{jxy}. In the section 3.5., details are given on these in the context of the particular implementation used in the proposed recognition system.

3.4.3.4. Drawbacks

It is apparent, that with the absence of decimation (downsampling) of the data, the resulting transform is overcomplete and leads to a significant increase in data storage requirements. The overcompleteness of the transform is a consequence of its non–orthogonality, as pointed out in Shensa, 1992.

As a consequence, since the amount of processed data does not decrease as the algorithm moves towards coarser scales, the computational load stays constant and it is higher, than in the decimated DWT's case. This problem can be solved by constructing computationally advantageous filters (Bijaoui *et al.* 1994), as it will be shown in the next section of the chapter.

The A Trous transform, contrary to the 2D version of Mallat's MRA, does not yield separate detail coefficient planes for horizontal and vertical directions in the image. This is a benefit, as it has been pointed out above, but it can be a drawback in applications that utilise directional sensitivity (like phase or texture analysis).

Still, in applications where directional descriptors are needed, such a transform that yields orientation–sensitive coefficient sets would be useful. A straightforward procedure can turn the above described transform into a directional one. In order to support this with a few examples, section 3.5.3. will describe the modifications to the transform and its potential. Prior to that, the following section presents an implementation of the discrete wavelet transform using the A Trous algorithm. This particular solution has been used in the proposed recognition system.

3.5. An implementation

An elegant, computationally very economic implementation of the discrete wavelet transform can be produced based on the cubic spline function as scaling function in the A Trous transform (as described in Bijaoui *et al.*, 1994; Starck *et al.*, 1995b). This, as it will be pointed out, not only has the advantage of reducing computational load, but also has attractive properties that make it a favoured candidate in pattern recognition applications. This solution has been somewhat overlooked in the literature, and its reported applications were in the field of astronomical object detection.

3.5.1. Mathematical background

The 3rd –order spline, which is the chosen scaling function in this algorithm, can be obtained by successive convolutions, starting with the box function B defined on the interval [0,1]. The cubic spline is the result of the operation $\phi = B * B * B * B$, where $*$ denotes the convolution operation. The box function and the resulting spline is shown in Fig. 3.13.

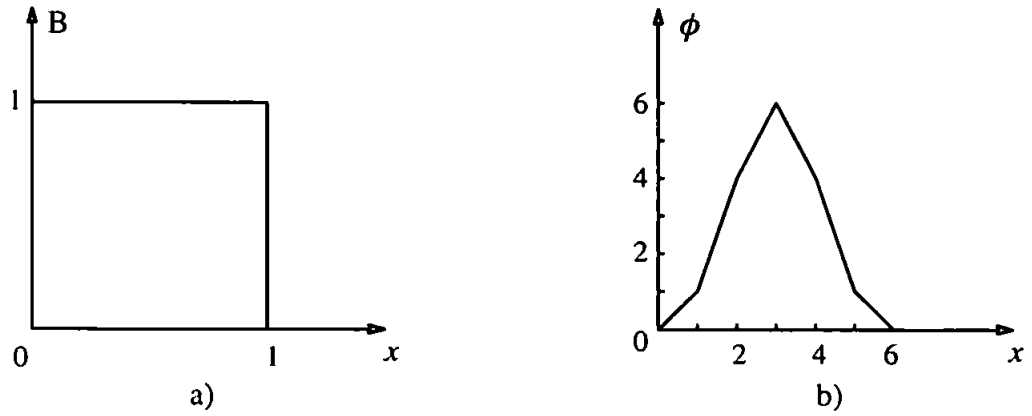


Fig. 3.13. a) The box function. b) The resulting 3rd–order spline.

The smoothing filter's impulse and frequency response can be calculated based on the dilation equation and the scaling function, as it is detailed in Strang & Nguyen, 1996. Its transfer function in frequency domain can be written easily as:

$$H(\omega) = \left(\frac{1 + e^{-j\omega}}{2}\right)^4 \tag{3.21.}$$

Therefore the smoothing filter has a zero of order 4 at $\omega = \pi$, these zeros give flat response near $\omega = \pi$, $\omega = 0$. Also, it has a pole of order 4 in $\omega = 0$, this producing a sharp 24 db/octave slope of the filter's frequency response.

If the degree of the spline function approaches infinity, the spline tends towards a Gaussian. Also, the derived g filter's impulse response converges to a cosine–Gabor function (Unser & Aldroubi, 1996). Even for an order of 3, this provides a good localisation in frequency. Hence knowing the analytical expression of ϕ, using equation (3.13.) results that the h_m coefficients will be defined by the set:

$$h = \left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\} \tag{3.22.}$$

This obviously satisfies the low pass condition $\sum_m h_m = 1$. So the spline scaling function leads to a short filter and simple filter coefficients, but since the scaling function doesn't produce an orthogonal basis in a space V_j, the resulting filters will not be orthogonal (Strang & Nguyen, 1996).

For constructing the g detail filter, the high pass condition $\sum_m g_m = 0$ must be satisfied. The filter could be constructed as a quadrature mirror filter and then the computation of the A Trous transform would involve two convolutions at each stage, as described previously. But a more inexpensive way is provided by the restrictions on the filters. Adding to the picture the invertibility condition, that introduces the conjugate filters $\tilde{h}$, $\tilde{g}$ used during reconstruction, a simple g filter can be derived (Bijaoui *et al.*, 1994). This invertibility condition in frequency domain can be written as:

$$H(\omega)\tilde{H}(\omega) + G(\omega)\tilde{G}(\omega) = 1 \tag{3.23.}$$

where $H(\omega), G(\omega)$ are the frequency responses of the smoothing and detail filter (calculated as the Fourier transform of their impulse responses) and $\tilde{H}(\omega), \tilde{G}(\omega)$ are the reconstruction (conjugated) filters' frequency responses. The exact way that leads to this condition, can be found in Shensa, 1992 and Bijaoui *et al.*, 1994. Intuitively, it means that applying the conjugated filters to the smoothed and detail data $c_{j+1,k}$ and $d_{j+1,k}$ (provided by the H and G filters applied during the direct transform to c_{jk}) and adding up the outputs, one should recover exactly c_{jk}. The relations between the filters in frequency domain, chosen in Bijaoui *et al.*, 1994 are:

$$\begin{cases} G(\omega) = 1 - H(\omega) \\ \tilde{H}(\omega) = \tilde{G}(\omega) = 1 \end{cases} \tag{3.24.}$$

If the first relation is submitted to inverse discrete Fourier transform, in time domain we obtain the impulse response of the g filter, i.e. the g_m , $m \in Z$ coefficients:

$$g_m = \delta_m - h_m \tag{3.25.}$$

where δ_m is the discrete Dirac impulse, with $\delta_0 = 1$ and 0 in rest. Based on this and on equation (3.20.), results:

$$d_{j,k} = -\sum_{m \neq 0} h_m c_{j-1,k+2^j m} + (1 - h_0)c_{j-1,k} = c_{j-1,k} - c_{j,k} \tag{3.26.}$$

This equation is the mathematical proof of the fact just briefly stated in Bijaoui's paper, that in order to compute the detail data, the algorithm has to simply subtract the data located on two consecutive smoothed coefficient planes, without any convolution. This leads to a trivial inverse transform, due to the successive subtraction between the previous and the current smoothed plane data throughout the decomposition: the sum of all detail planes and of the final smoothed plane gives back the original data. Returning to the g filter coefficients, from (3.25.) we obtain:

$$g = \left\{ -\frac{1}{16}, -\frac{1}{4}, \frac{5}{8}, -\frac{1}{4}, -\frac{1}{16} \right\} \tag{3.27.}$$

This obviously satisfies the high pass condition. The resulting filters are very short, and due to the construction of the detail filter, filtering with G is not necessary during the transform.

The impulse responses and frequency characteristics of the H and G filters are shown below in Fig. 3.14.

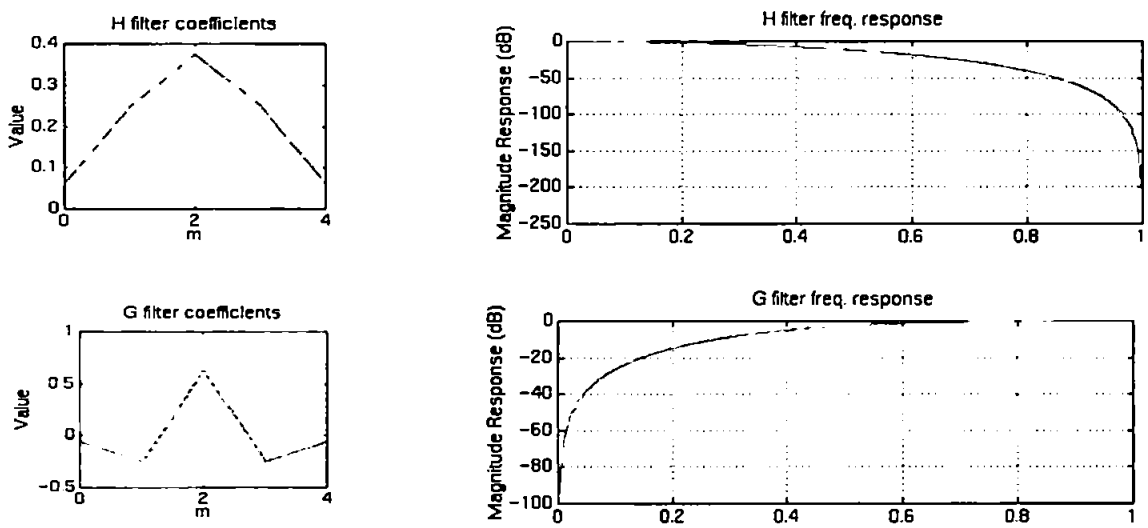


Fig. 3.14. Impulse responses and frequency characteristics of H and G filters. The frequencies are normalised (Nyquist rate/2 = 1)

The H filter kernel of size KxK in 2D will result from the tensor product $h \otimes h$, while the G filter kernel, based on equation (3.25.), can be computed as $g_{m,n} = \delta_{m,n} - h_{m,n}$, where $h_{m,n}$ is the element of the H kernel, and $\delta_{m,n}$ is a 2D Dirac impulse. In the case of $K = 5$, $\delta_{m,n} = 1$ for

$m = n = 3$ and $\delta_{m,n} = 0$ for rest. The resulting filter kernels (the impulse responses) are listed below:

$$[h] = \begin{bmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{bmatrix} \quad ; \quad [g] = \begin{bmatrix} -\frac{1}{256} & -\frac{1}{64} & -\frac{3}{128} & -\frac{1}{64} & -\frac{1}{256} \\ -\frac{1}{64} & -\frac{1}{16} & -\frac{3}{32} & -\frac{1}{16} & -\frac{1}{64} \\ -\frac{3}{128} & -\frac{3}{32} & \frac{55}{64} & -\frac{3}{32} & -\frac{3}{128} \\ -\frac{1}{64} & -\frac{1}{16} & -\frac{3}{32} & -\frac{1}{16} & -\frac{1}{64} \\ -\frac{1}{256} & -\frac{1}{64} & -\frac{3}{128} & -\frac{1}{64} & -\frac{1}{256} \end{bmatrix} \qquad (3.28.)$$

These and the corresponding frequency responses of the 2D filters are shown below in Fig. 3.15.

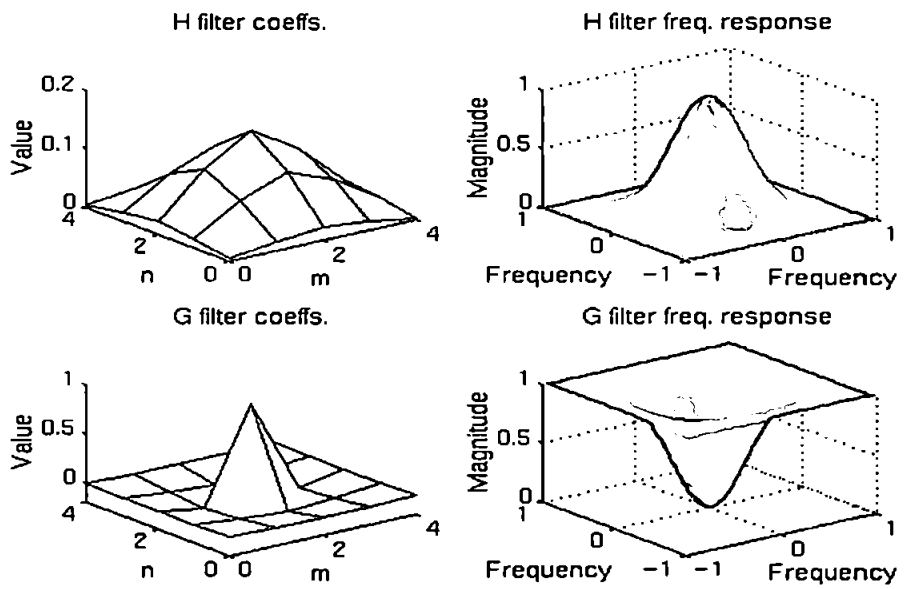


Fig. 3.15. Impulse responses and frequency characteristics of the 2D H and G filters. The frequencies are normalised (Nyquist rate/2 = 1).

3.5.2. Properties

i. As it is mentioned in Bijaoui *et al.*, 1996, the fact that the detail filter's impulse response has only one positive peak, is essential in object detection. For object or singularity detection, the

only relevant local maximum that marks a singularity on a certain scale is the central positive peak produced by the detail filter.

ii. The similarities between the impulse response of such a detail filter and the response profile of simple cortical cells are shown eloquently in Unser & Aldroubi, 1996. This makes it a computationally elegant emulation of multiscale processing tasks thought to be taking place in early vision. It shows how a mathematically complex operation like the wavelet transform can be carried out with extremely simple means.

iii. Reiterating the computational aspects of the algorithm, the wavelet decomposition at each stage needs only one filtering operation, the detail coefficients resulting as differences of the smoothed coefficients.

iv. As it was described in section 3.4.3.3. (see page 45), the transform has a number of attractive properties for pattern recognition applications. These properties will be supported with examples based on this particular version of the algorithm. The algorithm has been used in the past for the detection of point–like astronomical objects and of their groupings (Murtagh *et al.*, 1995). An algorithm with such relaxed conditions on its filters still exhibits a behaviour of its wavelet coefficients like the one described in Mallat & Hwang, 1992 for non–decimated wavelet transforms, as it is pointed out below:

- The detail filter acting as high pass filter, it enhances the changes in light intensity in the image (as it became visible also in Fig. 3.12.). Local maxima in wavelet coefficient planes corresponding to finer scale representations mark the location of edge elements and points of high curvature (corners). This property is illustrated in Fig. 3.16. below, where the fine scale wavelet plane is represented.

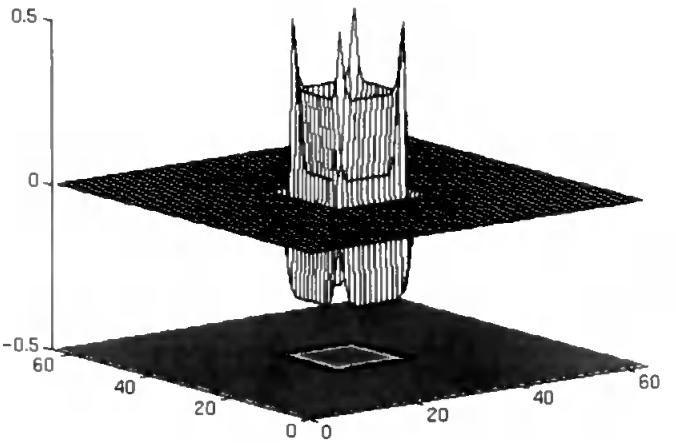


Fig. 3.16. Response of detail filter to an image of a white rectangle on black background

- The maximum of the transform on the coefficient plane associated with the coarsest scale marks the location of the object in the image. At this lowest resolution, the singularity in the image is the object itself; smaller entities in the image can produce significant maxima only on finer scale coefficient planes, hence the global maximum detected on the coarsest scale is useful in object localisation. This property is illustrated by Fig. 3.17. below; images of field–collected marine phytoplanktons have been used for this purpose, since these had noise and smaller objects (detritus or other biota) in the field of view.

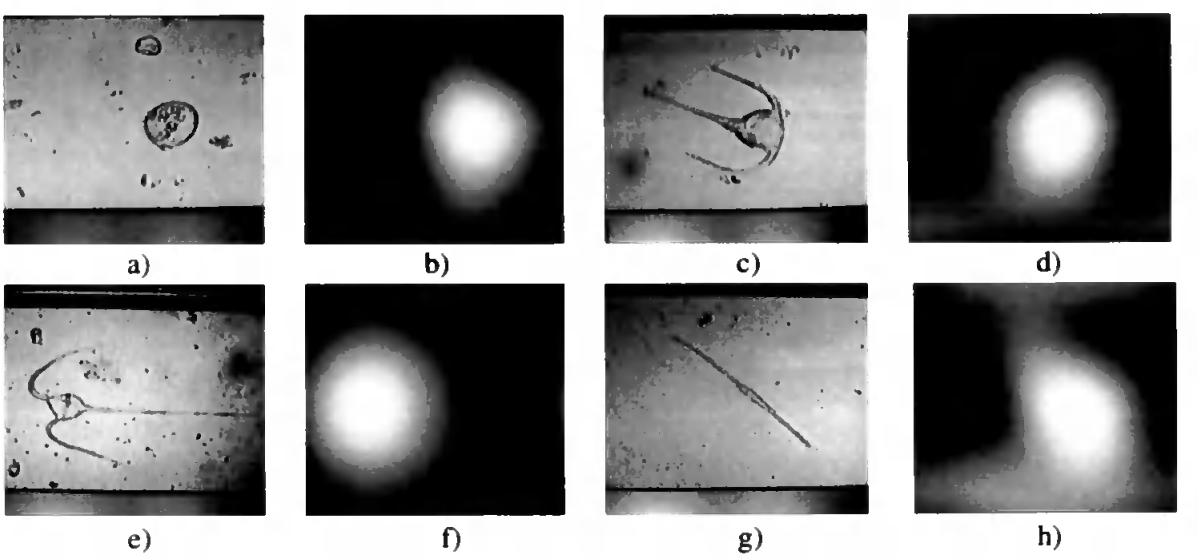
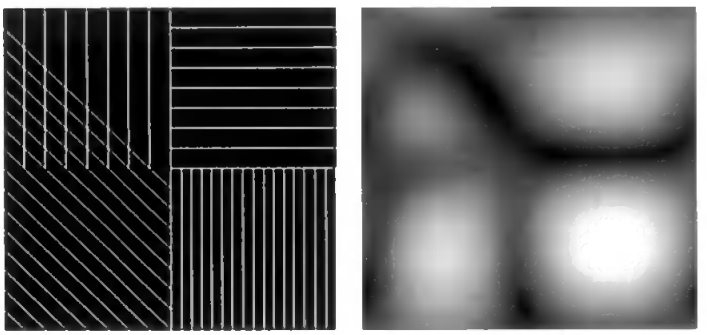


Fig. 3.17. Localisation of maxima on coarsest scale wavelet planes; the original images (a,c,e,g) and the detail planes associated with the coarsest resolution are shown (b,d,f,h)

- A wavelet coefficient maximum can mark areas with high frequency details (for instance, regions with dense textures). In Fig. 3.18. below, a mosaic of 5 different synthetic textures are presented – it is apparent, that the coarse scale wavelet coefficients exhibit the highest local maximum in the area that contains the densest texture.



a) b)

Fig. 3.18. A mosaic of five different synthetic textures (a) and a corresponding coarse–scale detail coefficient plane (b).

These properties of the transform were used in the system for building scale–space skeleton representations of objects' views and for directing the feature extraction process in the coarse data channels, as it will become apparent in the next chapter.

3.5.3. Potentials

The A Trous transform, as it was mentioned before in section 3.4.3.4., can be easily turned into a directionally sensitive transform. At this point, having described an elegant implementation of the A Trous algorithm and pointed out a few of its essential properties, the discussion of the directional transform can be supported by examples.

The solution to directional sensitivity is trivial, although it has been overlooked in the literature: in a given stage of the modified (i.e. directional) A Trous transform, when applying the smoothing and detail filters to the data along horizontal and vertical directions, the d_{1xy}^{temp} coefficients are also kept. These provide the horizontal details. As an additional step, the filters are applied vertically on the input data at that particular stage of the decomposition and this operation yields the vertical detail coefficients. These additions lead to a mathematically consistent A Trous trans-

form, with three detail coefficient planes (horizontal, vertical and the 'default' diagonal). The price to pay is evidently a further increase in data storage requirements.

Although the method is not used in the present implementation of the recognition system, the horizontal and vertical parts of the directional transform have the potential to describe the position and the orientation of singularities in the image (for example, edges). If the detail filter has the allure of a derivative operator, then evidently the horizontal and vertical parts of the transform become the approximations of partial derivatives of the signal along these directions. Therefore in an analogue way to the magnitude and phase measures calculated from a gradient, the following data can be computed on a given scale plane j:

$$P_{jxy} = \sqrt{d_{jxy}^{H^2} + d_{jxy}^{V^2}} \quad ; \quad \theta_{jxy} = \arctan\frac{d_{jxy}^{V}}{d_{jxy}^{H}} \tag{3.29.}$$

where d_{jxy}^{H} and d_{jxy}^{V} are the detail coefficients on scale plane j obtained by filtering along the horizontal and vertical direction, respectively. Having one–to–one correspondence between a pixel of a certain position on the image plane and a coefficient in the same position on a scale plane, these magnitude and phase estimates (the orientation of the normal in the considered point) can be used as in the case of classic edge detectors.

The A Trous transform implementation described in section 3.5.1. is used here to give a few examples of how a directional version of it can be employed for the above tasks. First of all, the response of the detail filter to a discrete unity step function is illustrated below in Fig. 3.19.

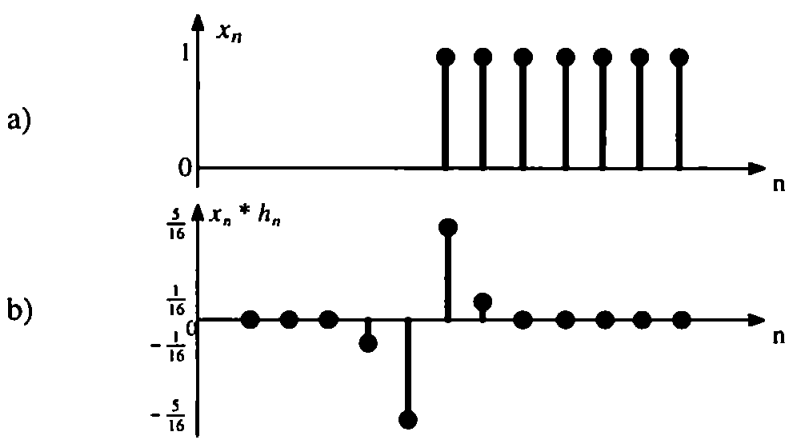


Fig. 3.19. a) Discrete unity step signal. b) Response of detail filter.

It is apparent, that in the case of a sharp transition in light intensity in the image, the filter will respond with positive and/or negative local extremes (a thin edge profile will produce a response close to the impulse response). The multiscale nature of the operator provides an elegant descriptor that by the magnitude measure P_{jxy} can describe locations of singularities in the image. The orientation of an edge can be estimated from the phase information θ_{jxy} on a fine scale plane. It would be suggested, based on the properties of the A Trous transform outlined in the previous section, that the phase is to be calculated in the positions where local extremes occur on the wavelet coefficient planes. On finer scales, the extremes would be produced by edges and corners/ points of high curvature (see figure 3.16.).

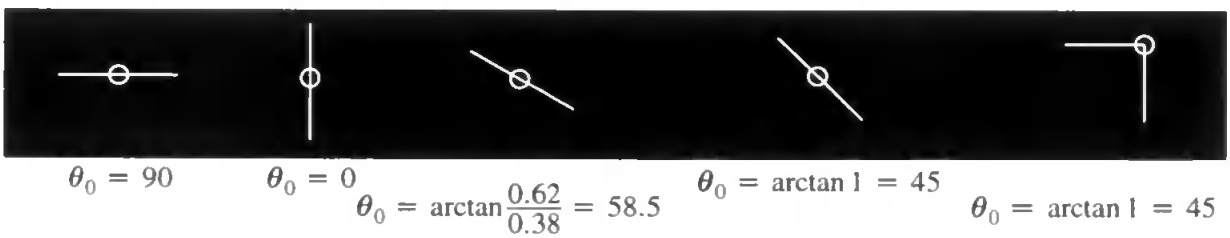


Fig. 3.20. Singularities (horizontal, vertical, 30°, 45°, corner) *and orientation of the normal, calculated from detail coefficients on scale plane 0 of the directional A Trous transform.*

To exemplify the validity of the approach, the phase information in the case of a few idealistic edges and corners are calculated on the finest scale plane with the detail filter described 3.5.1. and listed in Fig. 3.20. The location where the phase is calculated is marked with a circle.

Based on the above properties and possible extensions, the A Trous transform has the potential to become a versatile image analysis tool in situations where scale–space descriptions, informations on image singularities and textures are needed simultaneously. As it will be shown in detail in chapter 5., the directional A Trous transform offers attractive alternatives to texture analysis methods based on classic wavelet transforms.

3.6. Summary

The ideas behind multiscale analysis and representation of data have been introduced in this chapter. The wavelet transform as a mathematical transformation based on functions localised

both in space and frequency has been presented, together with the concept of scale–space that naturally emerges from such a transform. Starting from the mathematical definition and properties of the continuous wavelet transform, the mathematical steps that lead to the A Trous algorithm were detailed. The purely mathematical aspects of the transforms were placed into the practical context of signal processing by pointing out the connections between the involved functions and filter banks. Along the route, the advantages and disadvantages of the discrete wavelet transform algorithms were highlighted, thus clarifying the reasons for choosing the A Trous transform in the proposed application. An implementation of the A Trous algorithm has been described, that was chosen for the present application due to its properties. A number of very important properties of this transform have been uncovered and described. The following chapter presents the structure of the proposed system and the rationales behind each of its components.

Chapter 4. System structure

4.1. Introduction

Based on the theoretical aspects discussed in the previous chapters, this chapter offers an overview of the proposed system's structure. The following section presents the rationales behind the main design decisions taken when elaborating the overall structure of the system. The architecture of the system is discussed in the context of 3D object recognition systems described in the literature. In a subsequent section, the issues of coarse data coding and multiple coarse data channels are discussed, which constitute an important element in the system. Following the general discussion of the structure, brief descriptions of the functional modules and their interactions are given, highlighting their role in the system.

4.2. Rationales

This section reiterates the theoretical aspects that led to the fundamental concepts of the proposed system and presents the main considerations taken into account during the outlining of the system's structure.

4.2.1. Main guidelines in design

The theoretical issues discussed in the second chapter constituted the grounds for the design of the proposed object recognition system. As a recapitulation, in the light of the theories and the evidence reviewed in chapter two, the following aspects have been the starting points during the first stages of design:

- *The use of viewer–centred representations.* The system attempts recognition of objects from single 2D views and it operates with features located on the image plane. No viewpoint–invariant, object–centred models are constructed from the stimuli.

- *Representation by features.* The system operates on data provided by feature detectors, features being grouped by self–organising processes that yield signature vectors as input to the categoriser module of the system.
- *Multiple feature channels.* Various features are to be extracted from the input image, and the ensemble of the feature descriptors produced by the processing channels constitute the input to the categoriser. Depending on how salient a certain feature is for the correct identification of an object, a particular channel can contribute in a variable degree to the success of the recognition.
- *The utilisation of multiscale representations.* Multiresolution analysis provides information on the details of various sizes in the input image, this information being used to build a multiscale representation and to direct the extraction of features.

Besides these broad theoretical guidelines, a number of practical issues had to be taken into account. The proposed system was meant to be an engineering solution: one that can be implemented and used on reasonable computer platforms, allowing its use in real laboratory conditions and in acceptable operational time frames. These considerations played an important role in making particular choices in the system's architecture.

4.2.2. Architectural choices

At one extreme of vision research one finds computational systems that attempt the emulation of building blocks of the biological visual system close to the implementation level (in Marr's sense). Such solutions deal with complex and dynamic neuronal architectures, oscillatory phenomena among artificial neurons and succeed to perform complex information processing tasks with these means. As an example, work in this field has shown how figure–ground separation can be achieved without image processing in the conventional sense, with the use of a matrix of neural oscillators that attempt to model structures in the visual cortex (Hirakura *et al.*, 1996). A similar approach is described by Henkel (1995), where scale–space representation of the input data is analysed by synchronised neural oscillators. Chandrasekaran *et al.*, 1995 describes a system composed of gated neurons that compete in a selective manner in a Kohonen map–like structure (Kohonen, 1987), achieving object recognition from texture data. Some researchers suc-

ceeded to solve essentially mathematical problems with purely neural network–based systems, which is remarkable. For instance, Seibert & Waxman, 1992 describes a bi–layer of artificial neurons that is able to localise the centroid of a 2D shape based on features extracted from it – a problem, that in a classical way is solved with the help of analytical geometry. These approaches involve the modelling of complex dynamic, oscillatory systems and the simulation of neural structures of considerable size.

Another extreme would be the reliance on purely mathematical means in feature extraction and encoding. As an example of mathematically intensive techniques, Khotanzad & Liou, 1996 reports a method that achieves recognition of unoccluded 3D objects based on rotation, translation and scale invariant feature encoding. The encoding method itself, based on a special transform (Zernike moments) is far from the processes hypothesised to take place in the visual cortex. Still, it shows eloquently the possibility of simulating with exclusively mathematical means complex information processing tasks, that lead finally to invariant representations of objects. Similar approaches, that attempt classification of 3D objects based on a set of essentially mathematic descriptors, are illustrated by Botha *et al.*, 1996. Radar targets are described by geometrical moments, shape features, quantified energy strips – the task of recognising the target from this set of descriptors being performed by an artificial neural network module. Combination of multi-scale wavelet–based approximation methods and invariant curvature & boundary descriptors constitute the basis for the system described by Yoon *et al.*, 1998. Deschenes *et al.*, 1998 describes a system in which contours are extracted with the help of wavelet–based edge detectors and these are encoded in an invariant form with Fourier descriptors.

In the proposed system, a hybrid approach was preferred. Sophisticated mathematical methods can provide (when implemented in a computationally inexpensive way) elegant techniques for dealing with feature extraction tasks. Reasonably simple encoding techniques can be used for compensating the effects of minor changes in viewpoint. Relatively simple neural network architectures can add flexibility to the system, performing unsupervised grouping of features and providing the ability of learning multiple views of objects and categorising the data. The essential characteristic of such a combined approach is the trade–off between the employed mathematical tools and the neural computing aspects, in favour of reasonable computational load and hardware resources.

From this aspect, the proposed architecture is related to a large family of recognition systems described in the literature. Seibert & Waxman, 1992 and Waxman *et al.*, 1995 describe a system that is characterised by the joint use of geometric means (like log–polar mapping of features) and of special neural network architectures that achieve flexible learning. In the system described in Bradski & Grossberg, 1995 sophisticated filters modelled from the behaviour of certain cortex cells are used as preprocessing modules, in combination with flexible neural network classifiers. Edelman's Chorus scheme (Edelman, 1995b) utilises receptive fields and the data provided by these are fed into a hierarchic neural network. Systems, like those proposed by Mel (1997), use feature extractor modules performing complex mathematical operations and neural network–based classifiers. In the case of systems, like the one described by Lin *et al.*, 1998, sophisticated multiscale descriptors of features are used in conjunction with Hopfield neural networks.

Being confronted with the task of recognising objects from their 2D views without relying on sophisticated methods that deliver viewpoint–invariant descriptions of shapes, an alternative technique was chosen for feature extraction and encoding, namely the use of coarse data channels.

4.2.3. Coarse data channels

In analysing images for object recognition purposes, researchers usually have tended to focus on object and image properties that are also salient to human enquiry. Thus texture descriptions (Van Hulle & Tollenaere, 1993), edge positions (Canny, 1987) and statistical descriptions of pixel densities (Helterbrand *et al.*, 1994) have all been used to segment images into their component parts. Categorisation follows, providing object recognition. Many of these methods rely on the extraction of very precise measurements of, for example, symmetries or shape description. Such approaches are illustrated by the methods proposed in Brady, 1987 for extraction of shape symmetries; Ballard, 1987 for shape descriptions with generalised Hough transform and Khotanzad & Liou, 1996 for invariant contour descriptions. None have proved reliable analysis tools for understanding natural images or images with noise and clutter obscuring the objects of interest. As an alternative, a method has been developed by Ellis *et al.* (1994) that draws on the concept of Ullman's multiple visual routines (1984). The principle of operation is the low–resolution registration of multiple parameters that describe the object scene in an image. If many of these 'coarse channels' are analysed in concert a solution to the particular analysis may be found – one which

may not be apparent when using high resolution data. The coarse channel principle has been applied successfully to the automatic categorisation of 23 species of field collected marine plankton, in a system developed by Culverhouse *et al.* (1996), known as DiCANN.

The structure of the DiCANN system and the experimental protocol is illustrated below in Fig. 4.1.

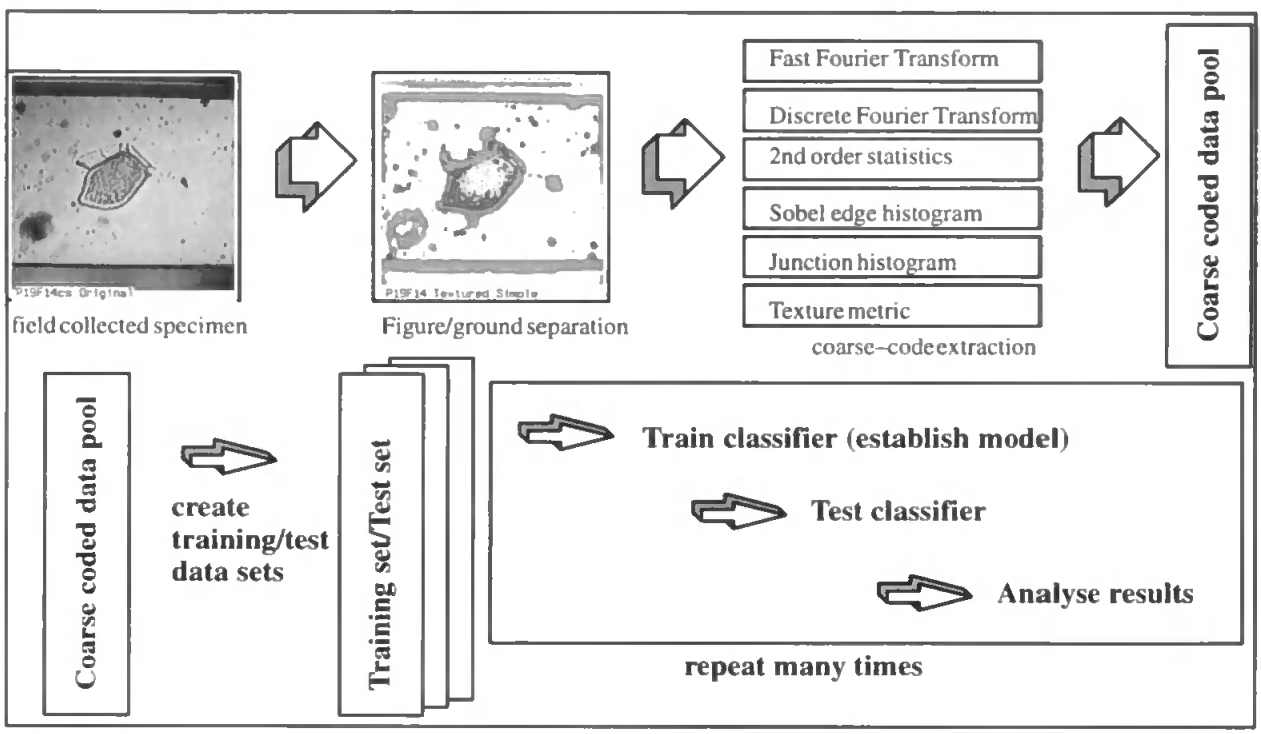


Fig. 4.1. Outline of the DiCANN automatic phytoplankton classifier system and of the experimental protocol

The system employed 6 coarse data channels, each of which being responsible for the extraction and coarse coding of features, operating on the entire 2D shape of the marine planktons presented as input. As it has been described in the introductory chapter, due to the viewpoint–variance of the shapes of 3D objects, the image processing strategies that work on the whole input shape were abandoned in the present project. When applying the coarse data channel approach to 3D object recognition, feature–based methods were preferred. With the coarse coding of features and their unsupervised grouping, one expects to compensate at least in part for the variations in input data caused by small changes in viewpoint. As in the case of the DiCANN system, an additional benefit of such an encoding is the decrease of the system's sensitivity to noise. The proposed object

recognition system was meant to deal with natural images of marine biota, too, hence the input images were expected to be affected by noise.

This approach of non–exact feature description and low–resolution encoding of features also constitutes the central concept of other recently developed systems that do not necessarily employ multiple data channels. Bradski & Grossberg, 1995 describes a system that uses in preprocessing an array of Gaussian receptive fields in order to decrease the dimensionality of the data. A similar approach is used for coarse coding log–polar transforms of feature locations in the system proposed by Waxman *et al.*, 1995. The system developed by Mel (1997) employs a large array of viewpoint–invariant filters placed on the input image, the outputs being coarse coded as histograms. In a conceptually related way, Schiele & Crowley (1997) have used multidimensional receptive field histograms characterising 2D views of objects in classification and in determination of favourable viewpoints for recognition. The Chorus scheme, too described in Edelman, 1995b utilises a receptive field array that provides low–dimensional description of the input data.

The main difference between these approaches and the proposed system is that the attention of the system is directed by a module employing multiresolution analysis towards areas of the image that contain potentially relevant features for categorisation. The extraction of these features is directed by low–resolution information, following work on visual inspection through eye tracking (Niemann *et al.*, 1996 and Rao *et al.*, 1996). The studies and computational models described in these papers have shown that the brain is very likely to use low–resolution description of the visual stimuli when deciding where to direct the visual attention to. Therefore as a novel approach, in the proposed recognition system the extraction of high–resolution information from the image is conditioned by the location of areas of interest marked by events in the coarse scale description of the input.

Such a mechanism certainly reduces the computational load in a biological system, since we don't need very high resolution description of the entire visual stimulus and the retina need not contain such a high number of receptors necessary for the fine–detail description of the whole visual field. The area of the retina with the highest density of receptors can be oriented towards locations where extraction of fine details is important. In a computer vision system, too, a similar mechanism would reduce the computational load if fine scale details were extracted only in areas that seem to be of interest.

With the use of multiresolution description of the input image, one can arrive at an elegant implementation of such a mechanism. By assembling the feature descriptions obtained from the areas of interest of the image into coarse coded signatures, the resulting data is hoped to register salient characteristics of the shapes presented as input. Hence the computation of descriptors of the entire input shapes (contours, spatial frequency distribution etc.) can be avoided – such descriptors would be strongly affected by viewpoint changes. The next section gives an outline of the system's structure and discusses briefly the role of each module.

4.3. Overview of the system's structure

The recognition system has three components: (i) a multiresolution analyser that uses wavelet filter banks, (ii) a multiple coarse channel feature analyser and (iii) an object categoriser. The spatial organisation of features is analysed through multiple coarse data channels. In the present work, the superposition of four channels has been explored, which are described in the next chapter. At this point of the discussion, in parallel with the brief description of the system components, the rationale behind each of these is presented.

The outline of the system's structure is presented in Fig. 4.2. below.

The preprocessing modules are responsible for the computation of wavelet transform, contrast enhancement and operations with wavelet coefficients. The latter involves detection of local maxima on the detail coefficient planes corresponding to various scale parameters and region growing on these coefficient planes.

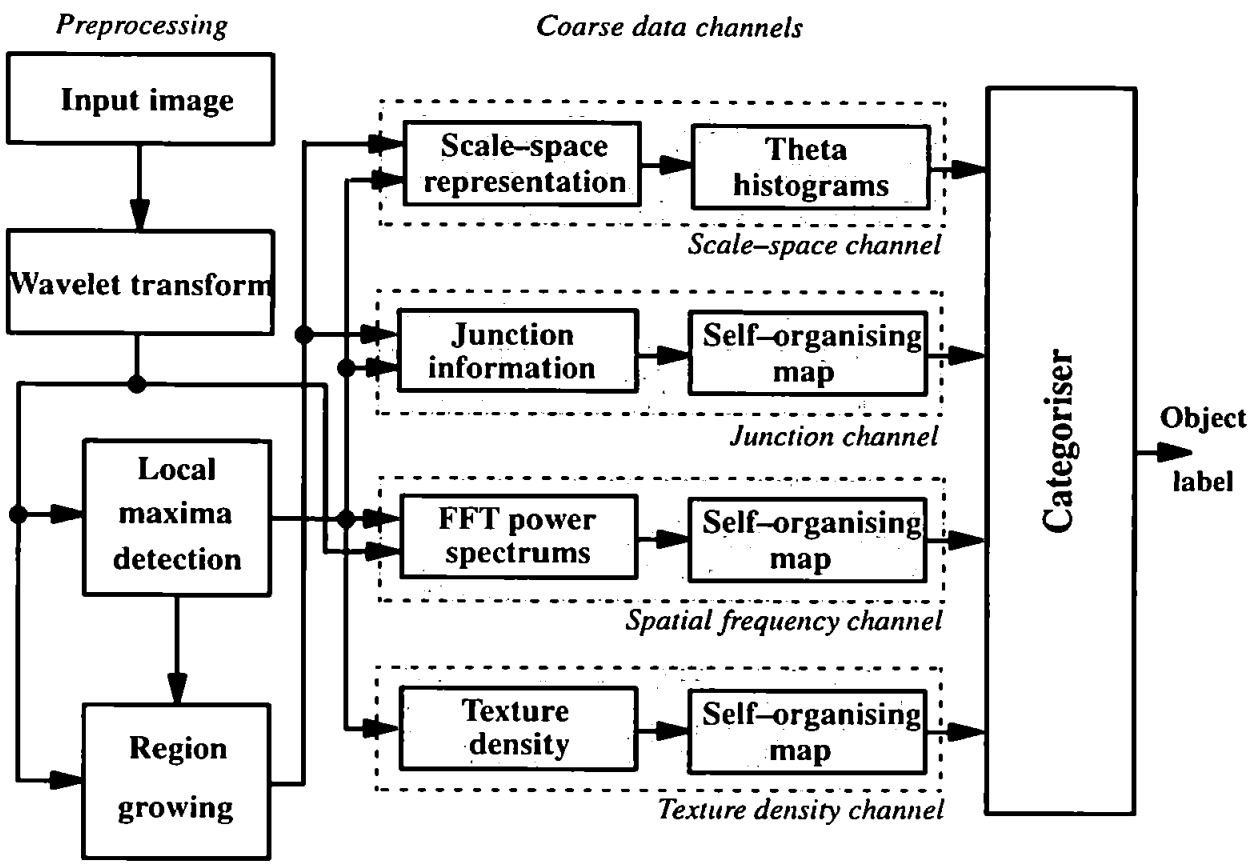


Fig. 4.2. The system's structure

The first data channel is based on spatial angle and norm descriptions in scale–space that lead to so–called theta histograms or rho–theta receptive field activation patterns, as it will be described in detail in the next section. This channel uses a novel multiscale representation of shapes. Hierarchic lists of link vectors represented in scale–space are calculated from multiscale tree structures that record the relationships between regions of positive wavelet coefficients and wavelet local maxima. The information on the orientations in scale–space of links between the leaves situated on successive scale planes of maxima trees are coarse–coded into the so–called theta histograms. The proposed multiresolution representation makes the description of multiple visible objects possible, having the potential to serve as a basis for future versions of the system that deal with several shapes present in the field of view.

The scale–space representation based on multiscale trees is related in its essence to other multiresolution tree structures found in the literature. Systems based on multiscale tracking of corners and edges (like the ones developed by Rattarangsi & Chin, 1992; Lee *et al.*, 1995; Hsieh *et al.*, 1997), contours or segments (Ren *et al.*, 1990; Liu & Yang, 1994) all attempt to explicitly label events (e.g. singularities) in scale–space. The present approach takes a different route, by em-

ploying much more relaxed rules when generating the scale–space representation. The concept draws on the theoretical arguments described in chapter two, which are in favour of internal representations that have a considerable degree of freedom in deciding which features to look for in a particular input image. Instead of looking for specific features like edges, corners in scale–space and building multiscale maps of edges, corners etc., the proposed technique represents in a multiscale form only potentially relevant object features. Since wavelet maxima –as described in the previous chapter– can mark such features, a representation of these maxima yields a description that is very generic. It was expected that such a representation would give a coarse description of the objects' shape and their local features' distribution in scale–space, such a description capturing salient properties of object shapes. The specific feature information would be then added by the other data channels.

Such generic wavelet maxima trees have been used recently for detecting and grouping of point source and extended astronomical objects, as the work of Rue & Bijaoui (1997) illustrates. The tree structures described therein only localise groups of objects and give some information on their inter–relationships. The method proposed here is designed for the characterisation of shapes of 2D views of 3D objects, taking into account relationships between wavelet local maxima and regions of significant coefficients in order to encode in multiple tree–like structures the spatial organisation of points of interest in the image. This channel provides a 'where' information to the system, while the channels described below add to the categoriser input the 'what' information on features in a coarse coded form. This separation of descriptors is inspired by neurobiological evidence (Ungerleider & Mishkin, 1982) that seems to suggest that 'where' and 'what' channels co–exist as separate processing streams in the cortex. The 'where' channel provides information on the location of objects in the image, while the 'what' channel describes the objects.

In subsequent descriptions, until future extensions of the system are discussed, the scale–space channel will often be referred to as a 'where' channel, emphasising the fact that this channel describes the location of features on the surface of a single object's 2D view. As it will be described in the final chapter of this work, with extensions brought to the system, the scale–space channel can become a 'where' channel in the above mentioned conventional sense, localising objects in the field of view and also describing the layout of their visible features in scale–space. Such separ-

ation between analysis channels provided the ground for the design and implementation of computer vision systems (Carpenter *et al.*, 1998).

One of the 'what' channels in the system introduces descriptions of edge coterminations into the system. As it has been pointed out by Biederman (1990), the types of junctions created by edge coterminations are non–accidental properties that can provide information on object shape in a viewpoint–invariant way. A junction of edges that mark surface boundaries, for instance, can give important information on how many surfaces meet there and in what way, such a feature being visible from a wide range of viewpoints. Showing the feasibility of such recognition approaches based on junction types, these features have been extensively used in the interpretation of line drawings and views of origami objects. An example would be the work of Kanade (quoted in Ballard & Brown, 1982).

The junction information is extracted from a map of regions of significant wavelet coefficients, these regions being obtained from monitoring the zero crossings of the coefficients. Therefore at this stage of the system, for obtaining junction information no decision making process is used (e.g. the utilisation of a hard threshold for obtaining an edge map). In the proposed system, instead of analysing the whole shape of a given object's view, the extraction of junction information is directed by low–resolution information obtained from the wavelet decomposition. Hence features are extracted only from areas centred on wavelet maxima detected on a sufficiently coarse scale coefficient plane. The resulting junction histograms, when propagated through a self–organising map, produce a node activation pattern which is then used as feature data by the classifier. Again, the above scheme emerged from the theoretical aspects regarding the unsupervised way in which features are thought to be grouped and used by the visual system, depending on their salience and the particular visual task. The Kohonen self–organising map (Kohonen, 1987) was chosen with regard to the fact that such a structure allows the projection of the multidimensional feature vectors onto a 2D map and the study of the resulting feature maps. In future applications, the sequential presentation of features could be traced on this map of neurons, adding a temporal component to the grouping process. The Kohonen map also provides a biologically plausible model to the unsupervised feature grouping process. The histogramming of junction types and the self–organising grouping process in training stage also achieves a coarse coding of the features. Information on the location of these features not being kept, this data channel provides a signature of the ensemble of edge co–terminations found in the input shape, which is expected

to be salient for recognition in conditions of variable viewpoint.

Local spatial contrast relationships are extracted in the third data channel, in the form of fast Fourier transform (FFT) spectra, coarse coded and propagated through a self–organising map. This spatial frequency channel extracts 2D–FFT power spectra in neighbourhoods of wavelet maxima. The spectra are coarse coded, yielding one–dimensional vectors that provide information on the spatial frequency content of the analysed areas. Global Fourier descriptors have been used with success in the past for identification of object shapes. Real–time recognition of objects based on two–dimensional Fourier transform has been reported by Reichel & Loffler, 1994, where optical means were used for obtaining the transform of the images. Radial slices of 2D Fourier transform were also used successfully in obtaining rotation, scale and translation invariant descriptions of shapes, as described by Chandran *et al.*, 1997. The coarse–coded frequency plane description of entire object shapes has proven to be a feasible method in automatic plankton classification (the DiCANN system described in Ellis *et al.*, 1994; Culverhouse *et al.*, 1996). As a contrast to the way in which FFT spectra have been used in the DiCANN system (i.e. computed from the entire input shape), here the locally extracted spectra provide information on localised events in the input (e.g. presence or absence of high frequencies due to texture, sharp transitions in light intensity etc.) at a finer scale that is not represented in the scale–space channel. It was hoped that the coarse coded and grouped local spatial frequency descriptors contribute to the performance of the system in recognising views of 3D objects.

The fourth data channel extracts and coarse codes texture information (fine scale details not registered by the previously described channels). As the models of low–level vision (reviewed in chapter two) showed, texture information is important for surface boundary and orientation descriptions. No mathematical model is available at the moment for the characterisation of textures, though. In practice, neural network–based techniques were used in combination with a variety of mathematical methods. Augusteijn & Clemens, 1996 reports the successful use of co–occurrence measures in texture description. More sophisticated descriptors were based on Markov models – these and self–organising maps were used in texture segmentation (Yin & Allinson, 1994).

As a departure from the above exemplified statistical descriptors, spatial frequency analysis has proven to be able to provide feasible techniques for the characterisation of textures. Oriented

Gabor filters were widely used for texture analysis (Van Hulle & Tollenaere, 1993; Mel, 1997), due to their ability of registering texture characteristics of various sizes and orientations. Recently, techniques based on wavelet and wavelet packet transforms were proposed for the description of textures (Laine & Fan, 1993, 1996; Unser, 1995). In the case of 3D object recognition, surface textures change with the orientation of the objects' surfaces. Therefore the proposed system employs a non–directional texture descriptor that delivers coarse texture density information. This is in contrast with the texture description strategies used in the DiCANN system, where directional Gabor filters were utilised.

It is possible to obtain a directional–sensitive texture descriptor based only on the A Trous transform, as it will be shown in the next chapter. But due to above mentioned variability of surface texture, a coarser descriptor has been chosen. Texture information is extracted from the multi-scale decomposition of the input image in areas of interest marked by wavelet maxima. In a similar way to the previous two coarse data channels, these localised descriptions are propagated through a self–organising map that yields signature vectors for the categoriser module.

The obtained multi–channel information on the input is classified by the categoriser module of the system. Due to multi–channel nature of the data submitted to the categoriser, a number of issues regarding the structure and the *modus operandi* of this module are raised. As a piece of research, the investigation of several categoriser architectures was proposed. This study allowed conclusions to be drawn on the way in which multiple coarse data channels contribute to the system's performance. On the practical side, the tests conducted with several different categorisers helped in establishing a robust classifier that maximised the system's performance.

In the tests carried out on various sets of 3D objects, statistical and artificial neural network–based categorisers were used. The statistical method allowed a robust evaluation of the system's performance and of the individual channels' ability of providing salient data for recognition. Artificial neural networks were also employed in experiments. Unlike discriminant analysis, neural networks are non–linear classifiers, hence are expected to build more powerful general models based on the input data. The neural network–based categorisers were used in two different architectural setups. A collective machine, that consisted of one neural network fed with the ensemble of data provided by all data channels allowed the tracking of performance as the number of channels was increased. Such tests made possible the study of the influence of new data channels on

the system's performance. Also, it made possible the generalisation to the field of 3D shape recognition of the studies carried out with the DiCANN system and 2D plankton shapes.

In a different setup, a committee machine was used, that incorporated a neural network for each data channel and the final decision on the objects' categories were taken based on the verdict of these individual networks (for a review and comparative study, see Ellis *et al.*, 1997). This essentially different setup from that of collective machines allowed the study of the strength of each channel in helping the system to reach a correct decision regarding the presented object's class. Two different structures of committees were tested. In the first case, the committee machine was a competitive system, where the most confident channel's vote provided the committee's verdict. In the case of the second tested structure, the sum of committee members' corresponding outputs provided the committee's output and decision was taken based on the highest output activation. This architecture enabled a certain degree of interaction between channels' decisions. The studies allowed comparisons between committee and collective machines, leading to a set of decisions regarding the practical version of the system's structure.

4.4. Conclusions

The above sections described the rationales behind architectural and design choices, reiterating also some of the theoretical issues discussed in chapter two and discussing the issues of coarse data coding, multiple data channels. An overview of the system's structure has been included, which briefly describes the component modules and places them in the context of other object recognition system's way of operation. The novelty of the approach has been pointed out, by describing the concepts behind the new scale–space channel, its properties, the employed attention focusing mechanism and the proposed feature extraction methods in the context of this proposed multiresolution information–directed system. These descriptions reveal the differences between the proposed system and the approaches reported in the literature. This chapter prepared the ground for a detailed description of the system's functional blocks, which follows in the next chapters.

Chapter 5. Preprocessing and feature extraction

5.1. Introduction

This section presents the *modus operandi* of the image preprocessing modules of the proposed system and the coarse data channels. The function and the algorithms behind each image preprocessing component is described, pointing out their role in preparing data for the coarse data channels. The feature extraction and coarse coding methods are detailed and discussed in subsequent sections, also giving examples which make the understanding of the algorithms easier. The final section of this chapter describes the implementation of these components.

5.2. Preprocessing modules

The modules responsible for preprocessing the input images compute the wavelet transform of the input, detect local maxima on wavelet coefficient planes and perform region growing on these planes. These modules are described in detail in the following sub–sections.

5.2.1. Multiscale analysis

The multiscale data necessary for scale–space representation and directing of feature extraction is supplied by the multiresolution analysis module that computes the A Trous wavelet transform. The implementation described in section 3.5. was chosen, due to its advantages pointed out therein. This preprocessing module, prior to the computation of the wavelet transform, also performs omnidirectional Sobel filtering (without thresholding the output) for contrast enhancement and elimination of very low frequency illumination gradients. It has been found, that wavelet maxima on coefficients planes corresponding to coarse scales (hence describing large details in the image) shift if illumination gradients are present in the image. Such slow changes are registered on very coarse planes (being perceived as low frequency details), therefore localisation of objects by wavelet maxima on coarse scales would be affected. The Sobel filtering compensates for this, eliminating slow changes in the image and has also the benefit of enhancing transitions in light intensity in the image.

More sophisticated contrast enhancement/normalisation preprocessing methods have been used in the past, a good example being the use of shunting off–centre/on–surround and on–centre/off–surround filters (Bradski & Grossberg, 1995). A drawback of the simple, Sobel filter–based technique is the enhancement of high–frequency noise together with the image details. But since in subsequent processing operations meant to build a scale–space skeleton representation only the coarse scale descriptions are used, this should not be detrimental.

The processing modules were designed to work with images of 256x256 pixels, due to characteristics of the used image capture equipment. In these conditions, the highest frequency present in an image is 128 cycles/image. This corresponds to a detail with a period of 2 pixels. The frequencies registered by the smoothed and detail coefficients on each stage of the wavelet decomposition are listed below in Table 5.1.

Table 5.1. Frequency bands of filters (in cycles/image)

Scale planes	Input coefficients	h filter	g filter	Output coefficients
plane 0	c_{0xy}	0..64	64..128	c_{1xy} , d_{1xy}
plane 1	c_{1xy}	0..32	32..64	c_{2xy} , d_{2xy}
plane 2	c_{2xy}	0..16	16..32	c_{3xy} , d_{3xy}
plane 3	c_{3xy}	0..8	8..16	c_{4xy} , d_{4xy}
plane 4	c_{4xy}	0..4	4..8	c_{5xy} , d_{5xy}
plane 5	c_{5xy}	0..2	2..4	c_{6xy} , d_{6xy}
plane 6	c_{6xy}	0..1	1..2	c_{7xy} , d_{7xy}

The analysis could be carried further as far as the algorithm is concerned, but due to the frequency bands that finally register the slowest variations in the image, coefficients obtained from further steps have no physical meaning. For future analysis, the detail coefficient planes are kept and the smoothed coefficients are discarded (since reconstruction from the transform is not proposed).

5.2.2. Local maxima detection

The resulting wavelet decomposition is the input to a local maxima detector, which produces maps of these maxima for each coefficient plane. A local maximum is defined as a location in the coefficient plane where the value of the detail coefficient is greater than those of the neigh-

bouring coefficients. A border–line case is when there are coefficients of equal value in the considered neighbourhood defined by a moving analysis window. If the centre of the window is marked as local maximum, then the algorithm must make sure that the window is not situated on a flat plane. Otherwise, the whole area –as the window moves– will be marked as local maximum. Such situations often occur when analysing noise–free synthetic images that contain uniform background.

The algorithm is described below, in the form of a pseudocode sequence, that also introduces many notations used in subsequent sections of this chapter. W_j denotes the wavelet coefficient plane on scale j and M_j is the map of local maxima on that scale. In future descriptions, an element m of M_j is considered to have as attributes the (x,y) coordinates of the position of the maximum and the value of the coefficient at that position. Using m as an index in a 2D structure means the use of its (x,y) position as index in the respective 2D structure. The size of the neighbourhood centred on the analysed coefficients d was set to 8x8 values. This, while not slowing down considerably the algorithm, allowed a reliable rejection of flat planes. The local maxima detection follows the steps listed below (in the case of a scale j):

```
false → M_j                                  ; map of local maxima for scale j
for each d ∈ W_j do                          ; for each coefficient on scale j
  true → flag
  true → flat_plane
 for each d' ∈ neighbourhood(d) do           ; investigate neighbouring coefficient values
  if d < d' or (d == d' and d not identical to d') then
     false → flag                            ; something larger is in d's neighbourhood
  endif
  if d >  d' or d < d' then
     false → flat_plane                      ; d is not on a flat plane
  endif
  if not flag then
     break_loop                              ; speed up algorithm if d is not local maximum
  endif
```

```
  endfor
  if flag and not flat_plain and d > 0 then
    value(d) → M_j(d)                    ; mark position of d as local maximum by
  endif                                  ; storing the value of the coefficient
endfor
```

The above process is repeated for all wavelet coefficient planes. These local maxima maps will be used to build the scale–space skeleton representation and for directing feature extraction. For the skeleton representations, maps of regions of positive wavelet coefficients are generated based on these maxima maps.

5.2.3. Region growing

Due to the characteristics of the detail filter (described in section 3.5 of the third chapter), significant events in the image produce positive wavelet coefficients. Therefore regions of positive detail coefficients mark areas on the image that contain such events: on finer scales, the registered events can be edgelets/edges, dense textures, high curvature points etc., while on coarse scales such a significant even can be the presence of an object. Such regions allow the grouping of wavelet local maxima. Maxima that stay in the same region are likely to be produced by closely grouped features. Maxima that end up in different regions or regions that split up when moving from coarser to finer scales will signify possibly well–separated groups of features in the image.

The wavelet coefficients are used by a sub–component of the preprocessing module that performs region growing on each detail plane. An illustration of the way in which these regions emerge on successive detail planes is shown in Fig. 5.1. below. A preprocessed image of a plankton was used as an example, since it has noise, detritus and smaller biota in the field of view and it shows that as the resolution increases, the regions mark outlines and details of the object, but also pick up smaller objects and detritus.

Fig. 5.1. A preprocessed plankton image (a) and regions found on detail planes 6,5,4,3 (b,c,d,e).

The used algorithm is a classic region growing algorithm, with modifications brought to it in order to take into consideration special cases, as it will become apparent from the following descriptions. The region growing algorithm uses on each detail plane as seedpoints the local maxima detected on that coefficient plane. Because of the properties of the A Trous transform that uses a 3rd order B–spline scaling function, the region growing algorithm ignores the negative values: any singularity in the image that produced a local maximum on a wavelet plane will produce a detail filter response that fades out around this maximum (i.e. the response gradually becomes negative). Therefore the boundaries of the regions are marked by zero–crossings of the wavelet coefficients. These areas of positive coefficients around the wavelet local maxima are marked by the algorithm as regions with individual labels.

There is one special case, though: when two relatively close local maxima end up in the same region, because the distance between them is not sufficiently large to produce a negative valley in the coefficients located between the maxima. In this case, a region around one of the maxima will include the other, too. The region that should have started from the already covered maximum is an empty set. The region labels do not have a meaning on their own, they are used for mapping purposes only. These aspects are described in the next section.

In the form of a pseudocode sequence, the algorithm is described briefly below. The set of regions on a scale j is denoted Γ_j, a particular region grown from a seedpoint is denoted R. The latter is a set of (x,y) coordinate pairs of coefficients that became part of the region. The algorithm operates on copies of the coefficient planes W_j, an already processed coefficient (i.e. one that ended up in a region before) being marked with a large negative value. This ensures that it is ignored by the algorithm in subsequent passes through that position in the coefficient plane. The size of

the neighbourhood used in the algorithm is 5x5 coefficients. The steps of computation are listed below for a particular scale j:

```
init_stack;
W_j → W                                   ; map of processed points
{} → Γ_j                                  ; init the set of regions for scale j.
for each m ∈ M_j do                       ; for each local maximum on M_j
  {} → R                                  ; start with an empty region.
  if W(m) <> -MaxFloat then               ; if seedpoint was not yet covered
                                          ; by a region, then process it.
    m → seedpoint                         ; the position of m is a seedpoint
    Push seedpoint                        ; push seedpoint on stack
    while stack not empty do
      Pop pnt
      if W(pnt) > 0 then                  ; if in the position marked by pnt
        R ⋃ pnt → R                       ; the wavelet coefficient is positive,
      endif                               ; add pnt to the region R.
      for each p ∈ neighbourhood(pnt) do
        if W(p) > 0 then                  ; push any point p in the neighbourhood
          Push p                          ; of pnt onto the stack, if a positive
          -MaxFloat → W(p)                ; coefficient is located there and mark
        endif                             ; the position as processed
      endfor
    endwhile
  endif                                   ; if seedpoint was already covered by
                                          ; another region, then continue here.
  Γ_j ⋃ {R} → Γ_j                         ; add R to the set of regions
endfor
```

Evidently, the local maxima will be located in these regions of positive coefficients. The analysis of regions in concert with the maxima offers the possibility of obtaining additional information

on object topology. The way in which the regions and the local wavelet maxima are analysed to yield a scale–space representation is described in the next section.

5.3. The scale–space channel

This coarse data channel provides the classifier with information on the scale–space distribution of potentially relevant object features. It incorporates a module that based on wavelet maxima, regions of positive coefficients, and the relationships between these builds a tree–like structure in scale–space. A subsequent processing module generates descriptions in polar scale–space of this tree, that is used in coarse coding to produce the so–called theta histograms. A more sophisticated coarse coding method has been also developed, that uses receptive fields placed on polar scale–space maps.

5.3.1. Wavelet maxima tree in scale–space

The wavelet coefficient planes corresponding to the coarsest four scales were used in building the representation. This choice was made based on considerations related to the coarseness of the description. Small details (of period less than 16 pixels) are not kept, events in the image occurring at such a fine scale being described by the other data channels.

As it has been described in the previous chapter, this multiscale representation is only meant to describe the scale–space organisation of areas of potential interest in the objects' views. Since the tree is built from the coarsest wavelet coefficient plane (d_{7xy}) towards finer scale planes, in the new context of scale–space tree structures the natural choice is to denote the coarsest scale plane (i.e. the lowest level of a tree) with index $j=1$. In the present implementation, a fixed number of scale planes are used for building the tree, due to considerations related to the known sizes of objects in the test data sets. As the final chapter of the thesis will point out, a variable number of scale planes would be used by an extended version of the system, when analysing multiple objects of arbitrary sizes.

At this point, the naming conventions used in the following sections can be introduced. All notations used in the previous sections of this chapter are kept. The list of wavelet local maxima will be called *maxima list*. This is a hierarchical structure (a list of lists), that for each of the considered 4 coefficient planes contains a list of local maxima detected on that wavelet plane. An *object*,

from the tree building algorithm's point of view, is a tree of maxima represented in Cartesian scale–space that has a root on a coarser scale layer and due to the maxima's position on the finer scale layers, it might register details of the same object in the image. These objects are organised in an *object–list* that will have as elements *object–maxlists* (i.e. maxima trees) for every potential object detected. These object–maxlists are similar to the maxima list (they have the same structure), but contain only the maxima that seem to belong to the same object. The resulting data structures are shown in Fig. 5.2. The synonymous terms object–maxlist and maxima tree will be both used in subsequent descriptions, emphasising the possible correspondence between the data structure and objects in the image or, respectively, topological aspects of the structure.

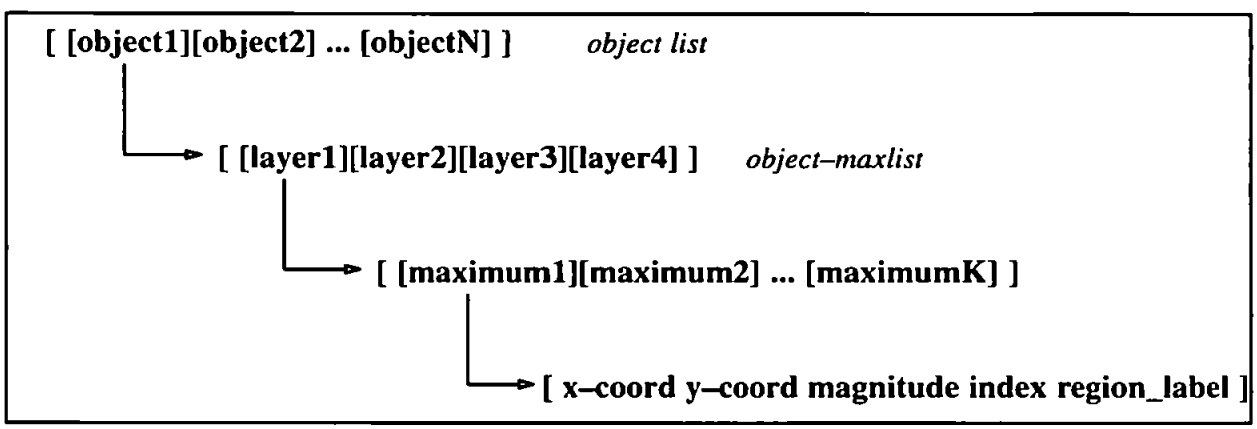


Fig. 5.2. Structure of object lists

The first step in generating a maxima tree is the choice of a root node. A dominant tree would have its root on the coarsest layer. Local maxima that appear to be isolated in the region map of finer scale planes become roots of sub–trees. More precisely, if a maximum on a given plane j is inside a region R of positive wavelet coefficients that does not include orthogonal projections onto level j of any of the maxima found on one level below, it will become a root node for a tree starting on layer j. This rule has been chosen for intuitive reasons: a maximum located in such an isolated region far from other clusters of local maxima is probably due to well–separated sub–structures in the image of the object. If more than one object is present in the image, then such a maximum is likely to be due to another object. In these cases, separate maxima trees are built that register the configuration of these sub–structures. Such trees will evidently have root nodes situated on layers above the one corresponding to the coarsest scale (i.e. the root node will be

picked from a layer with $j>1$). The set of root nodes can be obtained according to equation (5.1.):

$$\boldsymbol{B} = \{m \mid m \in M_1\} \bigcup \{m \mid m \in R \wedge (P(m', W_j) \notin R, \forall m' \in M_{j-1}), \forall R \subset \Gamma_j, \forall m \in M_j \; ; \; j = \overline{2, N_L} \} \quad (5.1.)$$

where $P(m, W_j)$ is the orthogonal projection of a maximum m onto the wavelet coefficient plane W_j, N_L is the number of analysed scale planes (N_L=4 in this case). The root nodes in $\boldsymbol{B}$ will be the base for the algorithm that builds a tree from each root node, by analysing the evolution of regions and wavelet maxima when passing from coarse scale planes to finer scale ones.

The tree $\boldsymbol{T}_t$ for each root is generated by the following algorithm:

```
t = 1;
for each b ∈ B do                          ; for every root node, build a tree.
  for j = 1 to N_L do                      ; seek coefficient layer that contains the node chosen
    if b ∉ W_j then                        ; previously to be root and empty layers below
      {} → T_tj                            ; the root node.
    else
      {b} → T_tj , j → n                   ; place node on the appropriate layer of the tree
    endif                                  ; and remember the index of the layer.
  endfor
  for j = n+1 to N_L do                    ; build tree on layers above root node
    T_tj ⋃ {m | m ∈ R ∧ (∃m' ∈ T_t,j-1 : P(m', W_j) ∈ R),
                ∀R ⊂ Γ_j , ∀m ∈ M_j } → T_tj
  endfor
  t = t+1;
endfor
```

This algorithm, by starting from a root node, will map all maxima above the level that contains the root, with regard to the interrelationships between the maxima and the regions that contain them. A maximum, in order to become a leaf on level j of the tree must be inside a region that

includes orthogonal projections of at least one maximum found on level $j-1$. This constraint will help in tracing sets of grouped maxima across scale–space, without adding to the tree nodes that are likely to be produced by other well–separated objects or isolated details in the 2D image. Such details of the image are registered by trees starting from root nodes that emerged exactly because of these details' isolated position when constructing the set $\boldsymbol{B}$.

In the case of such maxima trees that have their root on a layer different from the coarsest one, the root node has no corresponding local maximum on the coarsest layer. It might signal a separate, smaller object than those, that produced the dominant tree (which has root node on the coarsest layer), or might mark a part of an object that is far from the object's centroid and has singularities on a finer scale. It is impossible to tell at this stage which situation occurred, just by looking at the tree; so the actual interpretation of these higher–layer root maxima would be the task of a future recognition system, that uses other image analysis data as well and deals with multiple objects.

Another special case that can appear during the construction of the tree is that of region–splitting. On images containing a relatively large object with many details, a region detected on a coarse wavelet plane (e.g. a layer j) can split up into several, relatively close regions on a finer level (layer $j+1$). This might give information on a set of well grouped singularities that can be detected individually only on a finer resolution. On the other hand, this might indicate two separate components of the object, each of them producing separate local maxima and even separate regions on layer $j+1$. If a wavelet local maximum on layer j does not project onto layer $j+1$ in a point that is inside a region, the tree's branch terminates with this maximum.

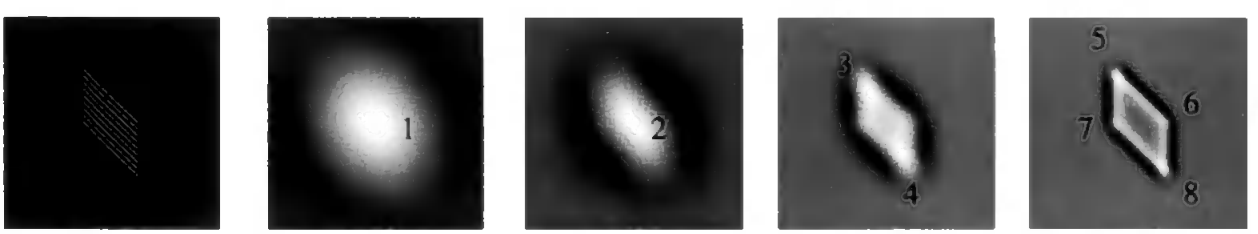


Fig. 5.3. Synthetic image (a) and local maxima found on detail planes 6,5,4,3 (b,c,d,e).

For a simple synthetic pattern, an example is given for the construction of object–maxlists. The pattern, the local maxima and their indexes given by the algorithm are shown in Fig. 5.3. This

simple example will be used in following discussions to exemplify the ways in which tree structures are built.

The resulting object–maxlist is listed below (the node indexes are shown in italics, to make identification of maxima easier):

```
[                                          ; object list
 [                                         ; start of the object–maxlist
  [[127 127 8.0 1 1]]                      ; sublist for layer 1: [x y magn index region]
  [[127 127 13.5 2 2]]                     ; sublist for layer 2  (1 maximum)
  [[108 91 10.5 3 3][146 163 10.5 4 3]]    ; layer 3  (2 maxima)
  [[101 74 10.8 5 4][152 126 7.3 6 4]
   [102 128 7.3 7 4][153 180 10.8 8 4]]    ; layer 4  (4 maxima)
 ]
]
```

With regard to the 2D translation invariance of this coarse coding module's output, a crucial role is played by the translation invariance of the A Trous wavelet transform. Due to this property, the position on each scale plane of the local maxima of the transform will reflect the 2D translations of an object' view in the analysed frame. Hence the maxima trees calculated relative to the root nodes chosen according to equation (5.1.) will exhibit translation invariance.

The maxima trees (object–maxlists) are the basis for the construction of the connectivity trees, as it is described below.

5.3.2. Connectivity tree in polar scale–space

Maxima trees stored as object–maxlists contain just a hierarchical list of maxima for every potential object. Based on this, a connectivity tree for each of these objects is generated, that encodes the links between maxima found in the layers of the object–maxlist.

The reason for generating such a representation of links is to render the data structures approxi-

mately invariant under image–plane rotations of the image. The links are encoded in polar scale–space, and with an appropriate choice of reference orientation for the link vectors, their orientations can be expressed in a 2D rotation–invariant form. Therefore a coarse coding method that has as input this orientation information can deliver to the categoriser module of the system a description that does not depend on the image plane orientation of the object to be recognised.

The simplest method for generating such trees of scale–space links would be to encode the links between all nodes on consecutive layers of object–maxlists starting from the coarsest layer and moving towards finer layers. A link between a node on level j and $j+1$ is seen as a vector in scale–space. It has a norm ϱ and an orientation θ. The calculation of the orientation angles is straightforward, as it is shown in Fig. 5.4. together with the used notations. The θ information is calculated as if the link's source node (e.g. node A) was projected orthogonally on the coefficient plane that contains the destination node (e.g. B). Only the first three scale planes with maxima and regions in the case of two object–maxlists are shown.

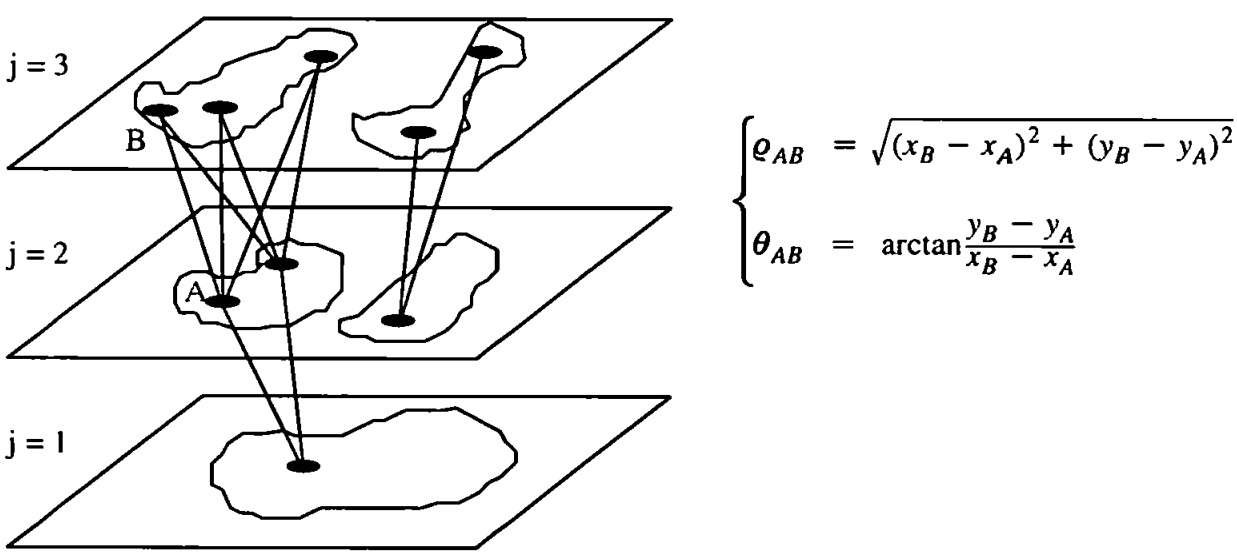


Fig. 5.4. Example of connectivity trees generated from two object–maxlists and the calculation of norm and orientation data for a particular link in the tree.

The obtained values will be variant under 2D (image–plane) rotations of the object characterised by the tree. Therefore a reference angle must be chosen, that only depends on the data stored in the tree. A straightforward choice is to use as reference the orientation of a dominant link between

two consecutive layers of the object–maxlist. For a given source node on layer j, the dominant link would the one that connects the source node to the largest maximum found on layer $j+1$ of the object–maxlist. All θ data is stored in the link tree as relative angle to the reference orientation.

As a convention, if the source and the destination nodes of a link have the same (x,y) coordinates on the coefficient planes, both norm and orientation information is stored as 0.

One connectivity tree being generated from each object–maxlist, the data structure results as illustrated in Fig. 5.5. It has three layers of links for each potential object.

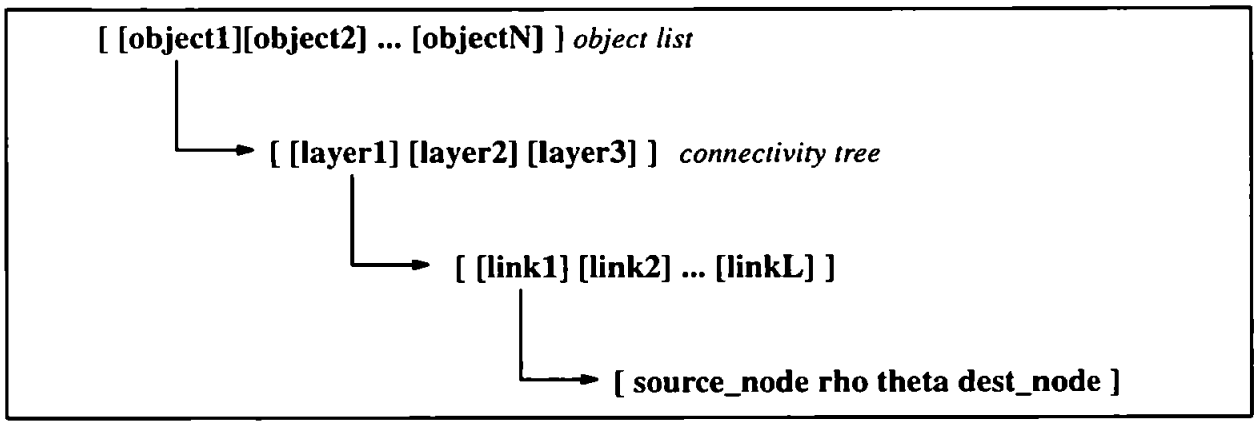


Fig. 5.5. Structure of link tree.

Every link has encoded the index of the source node in the maxima tree, distance to destination node, orientation and index of destination node in maxima tree. As an example, the connectivity tree for the simple pattern shown in Fig. 5.3. (built from the object–maxlist listed in the previous section) is the following:

```
[                                                    ; object list
 [                                                   ; start of connectivity tree
  [[1  ϱ=0.0  θ=0.0  2]]                             ; links between layers 1 and 2 of object–maxlist
  [[2  ϱ=40.7  θ=0.0  4][2  ϱ=40.7  θ=180.0  3]]     ; between layers 2 and 3
  [[3  ϱ=99.7  θ=0.0  8][3  ϱ=18.3  θ=184.4  5]
   [3  ϱ=56.2  θ=335.3  6][3  ϱ=37.4  θ=36.0  7]
```

```
   [4  ϱ=18.3  θ=0.0  8][4  ϱ=99.7  θ=175.5  5]
   [4  ϱ=37.4  θ=211.5  6][4  ϱ=56.2  θ=150.8  7]]          ; between layers 3 and 4
 ]
]
```

An important characteristic of the connectivity tree is its redundancy: between two layers in scale–space, links are constructed from every node on layer j to every node on layer $j+1$. Hence there are multiple links having the same destination node on layer $j+1$. Since any node – in the case of split region events or details that show up only on finer scales– can become a root for a tree that has its own reference theta, the direction encoding is not based on a unique reference throughout the tree. In the above link tree example, dominant links between nodes 3 and 8, 4 and 8 both became orientation references (with θ=0), but their orientation is obviously different. This can be confusing when the link orientations are interpreted or used in classification : the reference theta angles are not the same on every layer of the connectivity tree and are not the same even in different trees with roots on the same layer $j>1$ and leaves on layer $j+1$. The dominant links pointing towards the same nodes on layer $j+1$ have different direction for different source nodes on layer j.

This tree contains useful information on the behaviour of wavelet transform local maxima in scale–space, that can be used for extracting information on grouping, region split etc. events in the image. Still, in the case of a coarse–coding method that proposes to register in the form of a theta histogram the orientations of these links, such a tree presents several disadvantages. The redundant links make the coarse coding to register the same information several times. Furthermore, the reference angles –as exemplified above– can change several times when tracing the links from coarse to fine scale layers. The latter problem, i.e. the presence of non–unique reference orientations in the tree can make a theta histogram or similar coarse coded description confusing, with the risk of loosing potentially salient information on the links and their orientations.

It is possible to prune the tree by eliminating redundant links. Only those links are selected, that have a source node that became part of orientation reference links on lower layers, hence the multiple links to the same destination nodes disappear. The different reference angles' problem re-

mains. The connectivity tree for the example presented in the previous section becomes the following pruned tree:

```
[
 [
  [[1  ϱ=0.0  θ=0.0  2]]                          ; links between layers 1 and 2 of object–maxlist
  [[2  ϱ=40.7  θ=0.0  4][2  ϱ=40.7  θ=180.0  3]]  ; between layers 2 and 3 of object–maxlist
  [[4  ϱ=18.3  θ=0.0  8][4  ϱ=99.7  θ=175.5  5]
   [4  ϱ=37.4  θ=211.5  6][4  ϱ=56.2  θ=150.8  7]]  ; between layers 3 and 4 of object–maxlist
 ]
]
```

It is clear, that between layer 3 and 4 only links with source node No. 4 are kept, since this is the destination node of the dominant link between layers 2 and 3 (i.e. which became orientation reference with relative $\theta = 0$). Still, the reference orientation is not the same on every level of the tree. Therefore in tests a different route has been taken in order to solve the problem of different reference angles used for link orientation encoding. The link tree is illustrated in Fig. 5.6.; the 3 layers of maxima shown as example in the figure lead to 2 layers of links.

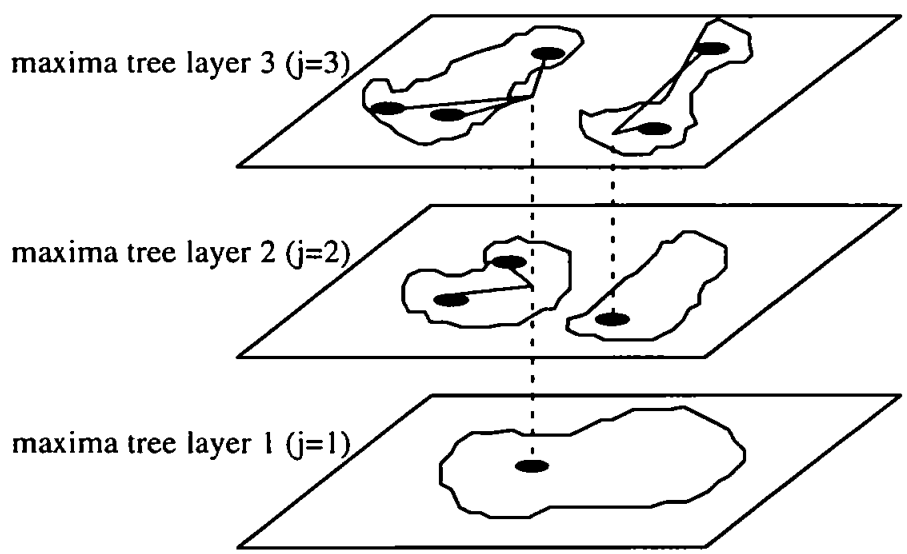


Fig. 5.6. Simplified tree structure with local maxima and regions. Orthogonal projections of roots become link starting points.

By starting on layer 1 (the coarsest wavelet plane) of an object–maxlist, a connectivity tree is built using the largest maxima on layer 1 as root (if there are more than one maxima on this layer). This root and its orthogonal projections onto higher layers are used as origins for the link vectors; the reference angle is defined by the first link that is found to connect the root's projection onto layer 2 with the node that has the largest magnitude on layer 2 or above. If no such node is found on layer 2 (e.g. due to a split region event), the search continues on upper layers. This link vector defines the reference direction and all theta information will be calculated relative to this. In the same way, trees are built from unprocessed root nodes located on higher layers (as the tree with root on layer 2 in Fig. 5.6. illustrates).

In the general case, a set of link projection vectors are generated for each T_t maxima tree, according to the following rules:

$$L_{t,j-1} = \begin{cases} \{\ \}\ , & \forall q < j\ :\ b \notin T_{tq} \\ \{\vec{l} = (\vec{a,m})\ |\ a = P(b, W_j),\ \forall m \in T_{tj} \subset W_j\ \},\ \exists q < j\ :\ b \in T_{tq} \end{cases} \tag{5.2.}$$

where b is the root of T_t, $j = \overline{2, N_L}$. The following link tree results, based on the input data shown in Fig. 5.3. (the orientations are shown relative to the reference angle):

```
[
 [
  [[1  ϱ=0.0  θ=0.0  2]]                          ; links between layers 1 and 2
  [[1  ϱ=40.7  θ=0.0  4] [1  ϱ=40.7  θ=180.0  3]]  ; between layers 2 and 3
  [[1  ϱ=59.0  θ=181.6  5] [1  ϱ=25.0  θ=295.5  6]
   [1  ϱ=25.0  θ=115.5  7] [1  ϱ=59.0  θ=1.6  8]]  ; between layers 3 and 4
 ]
]
```

The sets L_t constitute the input to the coarse coding sub–component that calculates the theta histograms.

5.3.3. Theta histograms

The previously described skeletonised representation in scale–space is invariant under image–plane rotation of the analysed object, provided that the transform which supplied the local maxima exhibits the same invariance.

It became obvious that 2D rotations can affect a wavelet transform that is calculated with successive convolutions along the X and Y axes of the input image. For example, only the vertical filter will respond to a horizontal discontinuity in the light intensity, but as soon as the object is rotated, the singularity by becoming diagonal, will produce responses from both (horizontal and vertical) detail filters. This can lead in the resulting wavelet coefficients to a local maximum that was not present in the transform of the original image. Thus, the tree building algorithm will produce a different connectivity tree in scale–space, due to extra maxima. The question is how to encode for a classifier the resulting connectivity tree, if there are slight variations in the number of nodes, number and orientation of links? At this stage of research, only the encoding of orientation (the theta information) of links was proposed as a solution.

A histogramming method was adopted for coarse coding the orientation of links in scale–space. The 360 degrees circle is divided into N angle intervals. Thus, by scanning the connectivity tree, on each layer of the tree an N–bin angle histogram is generated. The resulting theta histograms yield approximately invariant descriptions under 2D rotations of objects' views. This is satisfactory and adheres to the principle of coarse data channels.

In all tests, the trees that begin on higher layers than the coarsest one were concatenated, ignoring split region events. All these modifications were done due to the *a priori* knowledge on the test data: only one object was represented in each test image, thus extra topological information contained in the split region events and higher–layer subtrees was not relevant. In the case of future applications, where several objects can be present in the field of view, this information will be important for describing the configuration of the objects in the field of view. The final chapter of the present work discusses in detail this situation.

In order to obtain the feature vectors, N_L-1 histograms of relative orientation angles of links found in all hierarchic L_t link trees are computed. The reference orientation θ_t^{ref} is taken from L_t. For a certain L_t, θ_t^{ref} is the orientation angle of the vector l that satisfies $\| \vec{l} \| \neq 0$ (norm in

2D Euclidian space) and links the orthogonal projection of the root to the largest maximum found on the first level above the root of the corresponding T_t tree. At this stage, the relative θ values are collected from all link lists:

$$\Theta_j = \{ \arg(\vec{l}) - \theta_t^{ref} \mid \forall \vec{l} \in L_{ij} \ , \forall t \ \}, \quad j = \overline{1, N_L - 1} \tag{5.3.}$$

On each Θ_j set a histogram is calculated. For N histogram bins on each layer of the link tree, the obtained feature vectors are $(N_L-1)N$ dimensional. These feature vectors constitute the output of the scale–space channel.

5.3.4. Extension to rho–theta receptive fields

The previously described theta histogramming method captures the angular distribution of potentially interesting details in scale space. As an extension to the method, the registration of not only the link orientation, but also of the link vector norms has been proposed. The rationale behind such an extension was to add more scale–space geometry information to the descriptions, without compromising their coarse nature. A representation scheme that registers distances of areas of interest from the approximate centroid of an analysed shape would evidently carry more information on the shape's spatial distribution, than the theta histograms.

The added norm (ϱ) information has to be coarse coded, together with the link orientation data. First of all, the norms had to be rendered invariant under changes in scale. Assuming that reasonable amount of scaling of the input view does not modify the structure of the object–maxlists and link trees, therefore only the norms of link vectors will change, a normalisation with regard to the largest norm in a tree can solve the problem. The encoding of these normalised ϱ values in each link tree can be based on the same crisp splitting of the range of possible values into bins and histogramming. A more fuzzy method though, is the use of overlapping receptive fields placed on the rho–theta planes. If one imagines a layer of the link tree represented in a polar coordinate system, each link becoming a point of a given norm and orientation, a set of receptive fields can encode the configuration of such points in rho–theta space. Receptive fields are biologically plausible modules, but they also have the advantage of yielding a representation of the input which is not quantified by crisp value boundaries. The activation pattern produced by a grid of receptive fields can in the same time coarse code both ϱ and θ informations of links in the tree.

The ϱ information on a layer j of a link tree L_t is extracted as:

$$\varrho_{tj} = \{ \| \vec{l} \| \mid \forall \vec{l} \in L_{tj} \}, \quad 1 \le j \le N_L - 1 \tag{5.4.}$$

where $\| . \|$ is the Euclidian norm. In order to keep an approximate scale invariance, the distances are normalised by dividing them by the largest ϱ found in a set ϱ_t. This leads to a set of normalised distances that will be subsequently denoted $\tilde{\varrho}$. The relative theta values in the corresponding set Θ_t are normalised (divided by 360), hence all ϱ and θ values have range 0..1 at the end. These normalised values are denoted $\tilde{\theta}$.

On each of the three layers of links of the normalised rho–theta tree, a rectangular grid of Gaussian receptive fields is placed. The activation function of a receptive field on a layer j of the rho–theta tree is a 2D Gaussian in the $(\tilde{\varrho}, \tilde{\theta})$ plane, with standard deviation σ. Then the activation of a particular receptive field q on layer j can be written as:

$$RF_{qj} = \sum_p e^{-\frac{(\tilde{\varrho}_q - \tilde{\varrho}_p)^2 + (\tilde{\theta}_q - \tilde{\theta}_p)^2}{\sigma^2}}, \quad \forall t,p : \vec{l}_p = (\tilde{\varrho}_p \cos\tilde{\theta}_p, \tilde{\varrho}_p \sin\tilde{\theta}_p) \in L_{tj} \tag{5.5.}$$

where $(\tilde{\varrho}_q, \tilde{\theta}_q)$ is the centre of the Gaussian and $(\tilde{\varrho}_p, \tilde{\theta}_p)$ are the norm and relative orientation of links l_p located on layer j of a link tree.

For an NxN grid of receptive fields, the centres of Gaussians are placed at equidistant grid points in the $(\tilde{\varrho}, \tilde{\theta})$ plane; the activations of the receptive fields on each of the 3 link tree layers are collected in a feature vector of size $3N^2$ which is used as input to the classifier module of the system.

An example of receptive field grid placed on layer 3 of a link tree, together with a resulted activation pattern is shown in Fig. 5.7.

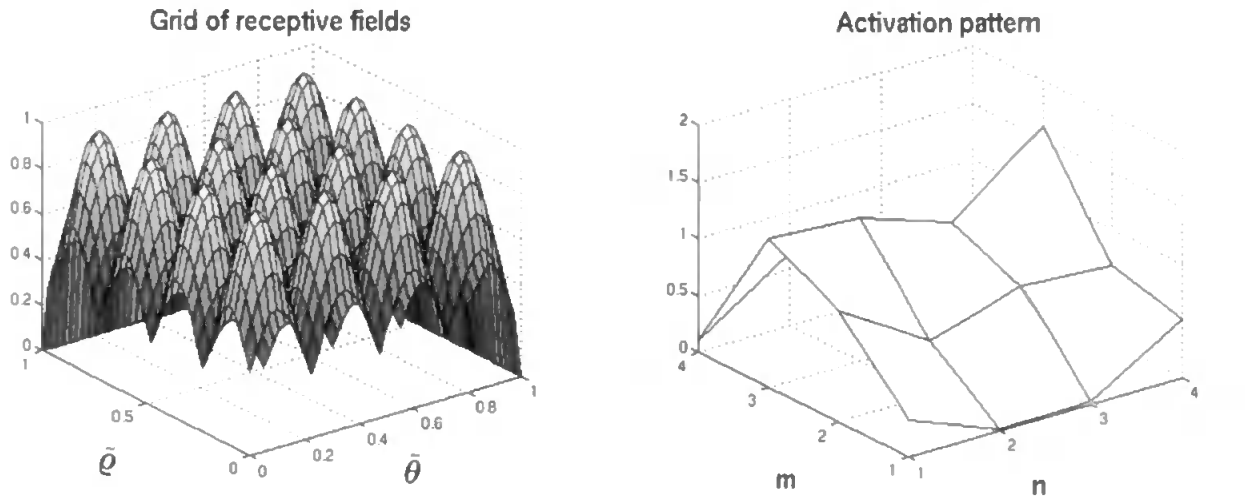


Fig. 5.7. The receptive field grid (a) and an activation pattern (b). Grid is placed on layer 3 of a link tree; grid size is 4x4 , $\sigma = 0.1$

Receptive fields placed on polar representations have been used in computer vision systems in the past. An eloquent example is Seibert & Waxman's work (1992), where log–polar mapping of features (corners) are the input to a grid of receptive fields. The above described method is evidently an extension to multiscale descriptions of polar mapping. Also, as it has been described extensively in previous sections, it does not rely on maps of particular features (like corners). Since translation and rotation invariance is built–in to the scale–space representation (the link trees), there is no need for double log–polar mapping and data alignment with regard to object centroid, as it happens in the system described in the above quoted paper. Bradski & Grossberg's VIEWNET system (1995) also employs log–polar mapping and receptive fields for dimensionality reduction of the data, but that system represents the silhouettes of objects instead of a skeletonised shape representation. It was believed, that the methods described in the above sections detailing the functions of the scale–space channel can give a salient multiscale description of shapes.

5.4. The junction channel

The second coarse data channel provides information on edge junctions detected in the image. As opposite to classic approaches that perform an analysis of the whole shape presented as input, the extraction of junction information is restrained to the areas of interest in the image, identified

by the multiscale data. For each neighbourhood of wavelet maxima (located on a sufficiently coarse coefficient plane), a junction histogram is produced.

5.4.1. Extraction of junction information

In all tests, local maxima detected on the fourth coarsest coefficient plane have been used to designate the locations where junction information is extracted from. In order to illustrate the principle, an example of an edge and region map, the points of interest marked by wavelet maxima resulted from MRA of the original image and the contents of the processing windows centred on these are shown in Fig. 5.8.

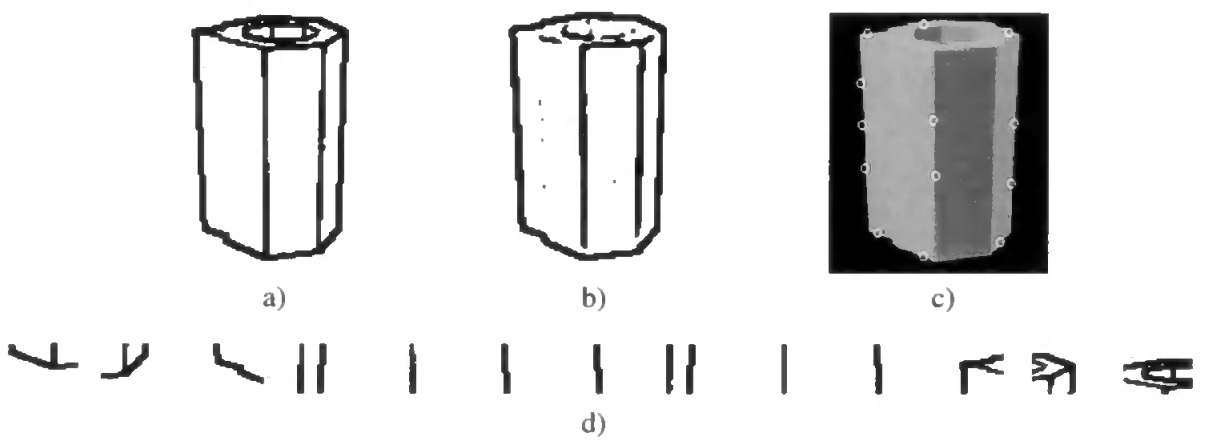


Fig. 5.8. Extracted junction data: a) Edge data. b) Regions on second finest scale plane. c) Set of points of interest. d) Contents of processing windows centred on wavelet maxima.

The actual junction extraction and categorisation is based on a simplified version of Ramsay & Barrett's algorithm (1982). The algorithm described therein has been stripped from all the sophisticated functions that use detection of closed boundaries ('cycles') and it has been adapted to higher resolution input data. The principles of operation of the simplified algorithm are listed below:

- Identify contiguous regions of active pixels in the input and try to fit lines on these. A set of generalised segments result, which are piecewise approximately linear.

- Detect the terminations of the constructed generalised segments and calculate angles between segments that meet in a close neighbourhood.
- Classify the junctions according to the number of segments that meet and the angles between them. Impose a tolerance of 10 degrees on angle values when deciding the type of junction, since it can not be expected to have precise angle data.

As an alternative to the use of edge maps as input, the regions of positive coefficients obtained from the second finest scale wavelet plane (d_{2xy}) were used. It has been found that these resemble to a satisfactory degree the edge maps (as the example in Fig. 5.8.b shows), with the advantage of discounting very small details that can be produced by noise.

The recognised junction types are shown in Fig. 5.9.

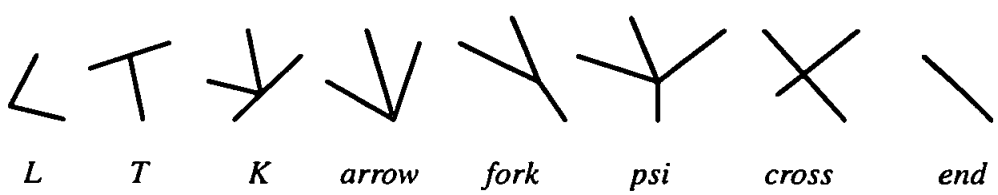


Fig. 5.9. The classic junction types recognised by the algorithm.

This set of categories was extended with regard to the cases in which several segments meet in the analysed area. Therefore the *jnN* categories were added, where N=4..9 and it describes the number of segments that meet in a point. For a number of 4 segments, *jn4* category is used only if no specific (*psi*, *cross*, *K*) junction type can be identified from the angles under which the segments connect. The *'other'* category was introduced for those junctions that can not be placed in any of the previously listed categories.

This junction detection algorithm being designed to work with good quality edge maps, if not ideal line drawings, evidently produces non–exact junction descriptions when presented with noisy images, not straight lines that do not meet exactly in one point and/or have several pixels

width. Still, it has proven to be a sufficiently coarse descriptor, leading to very good classification results, as future chapters will describe.

5.4.2. Unsupervised feature grouping

A Kohonen self–organising map (Kohonen, 1982; 1987) performs the unsupervised grouping of junction histograms. Briefly, the way in which this map operates is described below.

The map consists of a bi–dimensional array of nodes, where each node is basically an n–dimensional vector. Each of these vectors' components are initialised with a random number. As data is presented to the map, in the form of n–dimensional vectors $\vec{v}$, the distances between the input vector and the vectors $\vec{w}$ associated with the map's nodes are calculated. Any usual distance measure in n–dimensional space can be used (e.g. Euclidian distance, scalar product). The node that has the vector $\vec{w}$ closest to $\vec{v}$ is updated in such a way, that $\vec{w}$ is brought even closer to the input pattern. Therefore each component k of the vector $\vec{w}$ is modified following the rule expressed below:

$$w_k = w_k + \alpha(v_k - w_k) \quad , \quad k = \overline{1,n} \tag{5.6.}$$

where α is a constant that defines the amount of modification brought to $\vec{w}_k$. This is the update of the 'winner' node, competition among the map's nodes being introduced by this updating process. Also, nodes in the neighbourhood are updated. In the simplest case, the amount of modification that is brought to the neighbouring nodes' vectors decreases with the 2D distance between the updated and the winner node in the map. This means that the value of α decreases with this distance. In the case of more complex neighbourhood update functions, regions of inhibited nodes can be introduced, where node vectors are not modified.

After presenting all input patterns to the map, the whole process is repeated several times, each time decreasing the extent of the neighbourhood around the winner nodes and the value of α. As a result of this so–called training stage, the self–organising map learns the proximity relationships of the presented n–dimensional data, hence performing a clustering of the data, as Kohonen describes. Intuitively, due to the update procedure describe above, the vectors associated with the nodes of the self–organising map will describe the centroids of clusters present in the n–dimen-

sional input data. If one presents again the input patterns to the map, and finds for each input vector $\vec{v}$ the winner node with the closest $\vec{w}$, these nodes' topological distribution in the map will show a grouping that reflects the clustering in the input data.

Such self–organising process has the effect of grouping the junction histograms obtained from each processing window centred on wavelet maxima, according to their proximity in multidimensional space. Intuitively, similar histograms describing similar patterns of edge co–terminations will lead to groups of nodes in the Kohonen map with vectors $\vec{w}$ characterising these patterns. Also, the use of such a map has advantages in several aspects regarding data dimensionality. Since the number of wavelet maxima that defines the number of processing windows changes from image to image, the number of resulting junction histograms is also variable. Concatenation of these histograms prior to their presentation to the classifier module is not feasible for several reasons. The dimensionality of the resulting vector will be variable. Also, since there can not be any predefined order in which the junction histograms are extracted from images (the order depending on the configuration of wavelet local maxima), no ordering of histograms can be performed in the concatenated feature vector. But the presentation of the junction histograms to the self–organising map and the construction of node activation signatures leads to a fixed–length vector, with a dimensionality that can be much less than that of the concatenated of histograms and this can be presented to the classifier.

5.4.3. Obtaining the feature vectors

The junction histograms are extracted from processing windows centred around wavelet maxima found on the 4th coarsest detail plane of the A Trous decomposition. This layer was chosen with regard to the range of object sizes found in the data sets used in testing. In the general case, as it is described in the final chapter of this thesis, the system would operate with an adaptive process that would choose a wavelet plane for attention focusing based on the scale–space trees' populated layers.

For each processing window, the junction detection algorithm yields a junction histogram, in which each considered junction type is represented by a bin. Such a histogram describes how many occurrences of each junction category were detected in the current processing window. These junction histograms (one for each processing window) are propagated through a 5x6 node

Kohonen self–organising map, i.e. the histograms are presented as input vectors to the map. For a particular input image, the number of these vectors will be equal to the number of junction processing windows marked by wavelet maxima on the considered resolution plane. The size of the map has been chosen as a trade–off between the amount of dimensionality reduction that results (relative to the dimensionality of the concatenation of histograms) and the ability of the map of producing satisfactory grouping. As far as the latter aspect is concerned, a small map would not be able to yield nodes that approximate cluster centroids in situations where the number of such clusters present in the input data is large. This number not being known *a priori*, a reasonably large map has been constructed, in order to accommodate for a wide range of possible features. It presents the risk of yielding many nodes that are never activated, but this is not so detrimental compared to the opposite situation described above.

The set of junction histograms obtained from each image of a model data set is used for training the Kohonen self–organising map. The algorithm of the training stage is described below, using the notations introduced in previous sections:

for each image $I \in$ training_data_set

  get T_t ; read all object–maxlists (maxima
  ; trees) generated from this image.
  for each $m \in T_{ij}, \forall t$; for all maxima on layer j of each tree
    { **get_junctions**(neighbourhood(P(m, I))) } $\bigcup H \rightarrow H$; obtain a junction histogram from a
    ; window centred on maximum
  endfor ; and store histograms in set H.
  train_SOM(H) ; train self–organising map with
  ; the histograms.

endfor

After training, a histogram of node activations is generated by propagating through the trained map the junction data obtained from a set of test images. In this testing regime, when a novel image I is presented to the system for categorisation, the junction channel performs the following actions:

```
get T_t                                          ; read all trees generated for test image
0 → ActivationSignature[30];                     ; init node activation signature vector
for each m ∈ T_ij, ∀t                            ; for all maxima on layer j of each tree
  get_junctions(neighbourhood(P(m, I))) → H      ; obtain a junction histogram.
                                                 ; obtain an array of node activations,
  test_SOM(H) → NodeActivations                  ; where all elements are 0, except the
                                                 ; one corresponding to the winner node
  ActivationSignature + NodeActivations → ActivationSignature
endfor
```

The output vector *ActivationSignature* is used as feature vector, that describes how many times was each node a winner during the presentation of the input patterns (the junction histograms). This is presented to the classifier module in chorus with the vectors supplied by the other coarse data channels.

5.5. The spatial frequency channel

Rotation invariant encoding of FFT spectra is used in the implementation of the spatial frequency coarse data channel, in order to provide information on the spatial frequencies in areas of interest on the image. These FFT spectra are collected from locations marked by wavelet transform maxima, instead of the classical approach of computing a spectrum that characterises the whole object.

The 2D power spectra calculated in the processing windows are collapsed into a feature vector, following work reported in Simpson, 1992. This operation (briefly outlined in the latter work) renders the measure of frequency contents of the image rotation–invariant, as it is described below.

5.5.1. Obtaining rotation–invariant spatial frequency measures

The continuous form of the Fourier transform of a 2D signal is defined as:

$$F(\omega_x,\omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j(\omega_x x+\omega_y y)}\,dxdy \tag{5.7.}$$

This expression produces the complex spectrum at horizontal frequency ω_x and vertical frequency ω_y. In practice, one is interested in information in the complex spectrum along a direction θ. This describes the behaviour of the signal along this direction that is expressed as:

$$\theta = \arctan\frac{\omega_y}{\omega_x} \tag{5.8.}$$

The Fourier transform's definition can be re–written in polar frequency space, by expressing ω_x and ω_y as a function of frequency ω and orientation $\theta \in [0, 2\pi)$. The expression that yields the complex spectrum's value at a given frequency and orientation is therefore:

$$F(\omega,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j\omega(x\cos\theta+y\sin\theta)}dxdy \tag{5.9.}$$

The power spectrum is calculated as $P(\omega,\theta) = |F(\omega,\theta)|^2$. One can use the latter expression for calculating the transform and analysing the power spectrum, when it is important to gather information on the signal's behaviour along various directions in the image plane. Such polar description offers the means for calculating a rotation–invariant frequency–domain description of an image.

In order to discount the orientation information, one must calculate the sum of the transform along a circle of radius ω:

$$C(\omega) = \int_{o}^{2\pi} P(\omega,\theta)d\theta \tag{5.10.}$$

If the image is rotated in 2D, the value of $C(\omega)$ will not change. This measure will describe the frequency contents of the image, in a rotation–invariant way.

In the case of an image of $N_r x N_c$ pixels, the discrete version of the Fourier transform can be

written as:

$$DFT(\omega_p,\omega_q) = \frac{1}{N_r N_c}\sum_{y=0}^{N_r-1}\sum_{x=0}^{N_c-1} f(x,y)e^{-j(\omega_p y+\omega_q x)} \qquad (5.11.)$$

where the horizontal and vertical frequencies take the discrete values:

$$\omega_p = p\frac{2\pi}{N_r}\ ,\ p = 0,...,N_r-1\ ;\ \omega_q = q\frac{2\pi}{N_c}\ ,\ q = 0,...,N_c-1 \qquad (5.12.)$$

In order to obtain a rotation–invariant measure of the frequency contents of the image, the above transform has to be written in polar form. But in the case of this discrete transform, the polar form can not be obtained in such a straightforward manner as in the case of the continuous Fourier transform. Since the complex and the power spectrum is obtained as a rectangular grid of coefficients, the values that ω and θ can take are constrained by the grid points to discrete amounts. This becomes apparent from the illustration of the frequency plane shown in Fig. 5.10. This depicts the frequency plane after the diagonally opposite quadrants have been swapped, i.e. placing the lowest frequencies in the centre of the grid – a widely used representation in signal processing. The number of rows and columns have been taken equal.

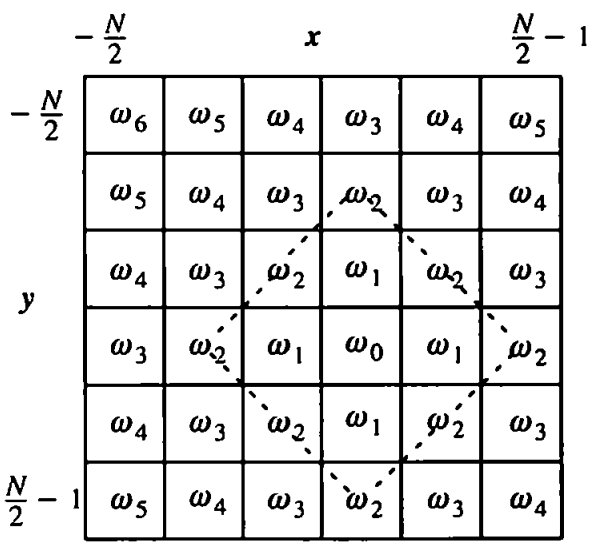


Fig. 5.10. The discrete frequency plane with the DC component (ω_0) centred by swapping diagonally opposite quadrants of the transform.

From the grid, the polar form of the discrete Fourier transform results:

$$DFT(\omega_k,\theta_l) = \frac{1}{N^2}\sum_{x=-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{y=-\frac{N}{2}}^{\frac{N}{2}-1} f(x,y)e^{-j\frac{2\pi}{N}(Int\{k\cos\theta_l\}+Int\{k\sin\theta_l\})} \qquad (5.13.)$$

where the function *Int{a}* signifies the integer part of a (i.e. a truncated to integer). From the frequency grid, the possible values that the frequency and the orientation can take result:

$$\begin{cases} \omega_k = k\frac{2\pi}{N} \;,\; k = 0,\ldots,N \\ \theta_l = l\frac{\pi}{2k} \;,\; l = 0,\ldots,4k-1 \end{cases} \qquad (5.14.)$$

It is evident, that the orientation values depend on the frequency ω_k. The higher k is (i.e. the further away one is from the centre of the frequency grid), the more orientations can be taken into consideration. Of course, like in the grid shown above, elements corresponding to frequencies with $k > \frac{N}{2} - 1$ are clipped by the bounds of the grid (e.g. ω_6 appears only once). In order to obtain the rotation–invariant frequency measure, the following sum of power spectrum components must be calculated (along the paths in the frequency grid marked with dotted lines in Fig. 5.10.):

$$C(\omega_k) = \sum_{l=0}^{4k-1} P(\omega_k,\theta_l) \;,\; k = 0,\ldots,\frac{N}{2}-1 \qquad (5.15.)$$

As an example, a synthetic image, its 2D Fourier transform computed on the whole image and the resulting spatial frequency descriptor is shown in Fig. 5.11.

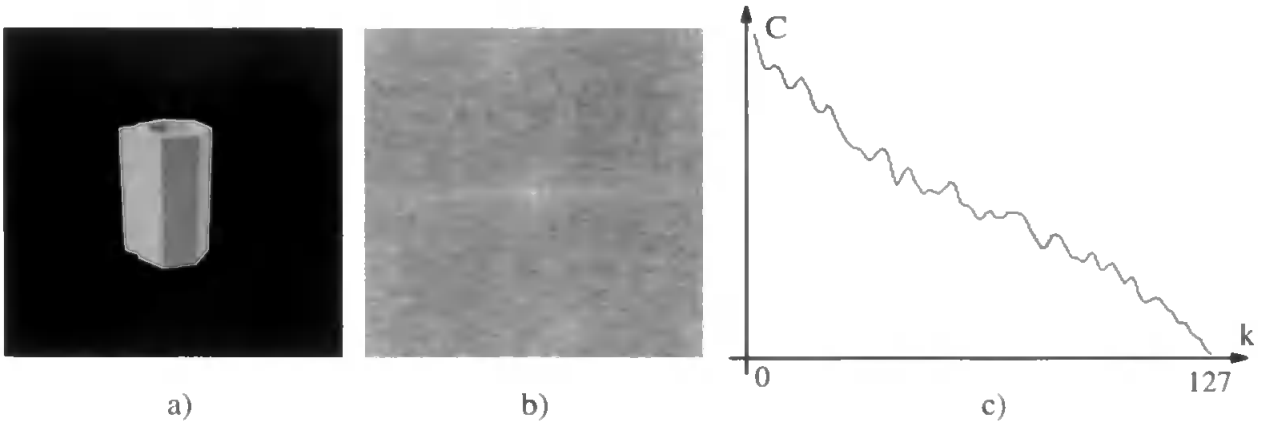


Fig. 5.11. a) A synthetic 256x256 pixels gray scale image of a 3D object. b) The Fourier transform of the image. c) The resulting $C(\omega_k)$ vector of 127 elements.

The $C(\omega_k)$ vector is used as spatial frequency descriptor, as it is described below.

5.5.2. Coarse coding of spatial frequency descriptors

The frequency descriptor defined in equation (5.15.) is calculated in processing windows centred on wavelet maxima. As in the case of junction analysis, the 4th layer of maxima trees have been used for marking the locations of these windows, taking into account the particularities of the test data sets.

For the computation of the Fourier transform, the 2D–FFT algorithm has been used. The size of the windows has been chosen to be 16x16. As input to the Fourier algorithm, the data stored on the second finest wavelet coefficient plane is used instead of the input image. This data (d_{2xy}) being the output of the detail filter in the second stage of the A Trous decomposition, does not contain very fine details (of period less than 4 pixels). Therefore the Fourier transform of this data will not register high frequencies that are due to image noise – an advantage in the analysis of natural images. DC information is not kept, and the orientation–invariant $C(\omega_k)$ descriptors are used as a 1D array in training a self–organising map. The methodology is the same as the one described in the sections on the junction channel.

In testing regime, the activation signatures obtained from the self–organising map are used as feature vectors for the categoriser module. The output of this channel is expected to give information on the spatial frequency contents of areas marked by wavelet maxima, that would help in categorisation.

5.6. The texture channel

As it has been mentioned in the overview of the system's structure (p. 70), wavelet–based descriptors have been used with success in texture analysis and segmentation applications. Having in the proposed system a module that computes wavelet transform, it would be an advantage to use the transform for calculating texture descriptors, instead of employing an additional image analysis module that would considerably increase the computational load.

5.6.1. Texture descriptors based on directionally sensitive A Trous transform

The possibilities of obtaining a texture descriptor based purely on the A Trous transform coefficients have been investigated. In the literature, Laine & Fan (1993) reported almost perfect classification of 25 natural textures chosen from the Brodatz album, using a method based on discrete wavelet packet transform (DWPT) and energy signatures calculated from all coefficient sets of the resulting quad–tree. The key role in registering the orientation of textures is played by the directional sensitivity of the transform, which is due to the horizontal, vertical and horizontal+vertical filtering of the data.

When opting for a wavelet–based texture description method, the main question posed was: can a similar, simple wavelet coefficient energy measure provide salient texture description when applied to A Trous transform data? As it was pointed out in chapter 3, this transform presents several relaxations to the usual constraints imposed on filters in wavelet theory. Still, it is a sampled version of the continuous wavelet transform, therefore any theoretical assumptions regarding this transform should be valid, with the exception of those regarding orthogonality. Also, directional sensitivity is not built–in to this transform. It became clear, that in order to render the A Trous transform directionally sensitive, one needs also the detail coefficients resulted from the horizontal and vertical filtering of the input data. Therefore the modifications proposed in section 3.5.3. (p. 55) were brought to the transform.

Energy measures (sum of squared wavelet coefficients) were computed on all 7 scale planes of the A Trous decomposition, on each scale plane 3 coefficient sets being obtained (horizontal, vertical, diagonal details). Hence 21–dimensional texture descriptors resulted.

In preliminary tests, 10 Brodatz textures were used. The 256x256 pixel gray–scale images were

histogram equalised, so that discrimination between the textures would not be possible just based on first–order statistics. The textures are shown in Fig. 5.12. For each image, energy measures of wavelet coefficients have been calculated in 121 overlapping windows placed on each wavelet coefficient plane. The window size is an important factor in the performance of the method, since it has to capture salient properties of the textures. If it is chosen to be very small, in the case of sparse textures it might not capture all salient details of the texture. The window size in practical situations has to be adapted to the texture density, but in these simple tests it was fixed to 32x32 coefficients. The 121 feature vectors obtained from each texture image were submitted to a simple categorisation using discriminant analysis (described in detail in chapter 6).

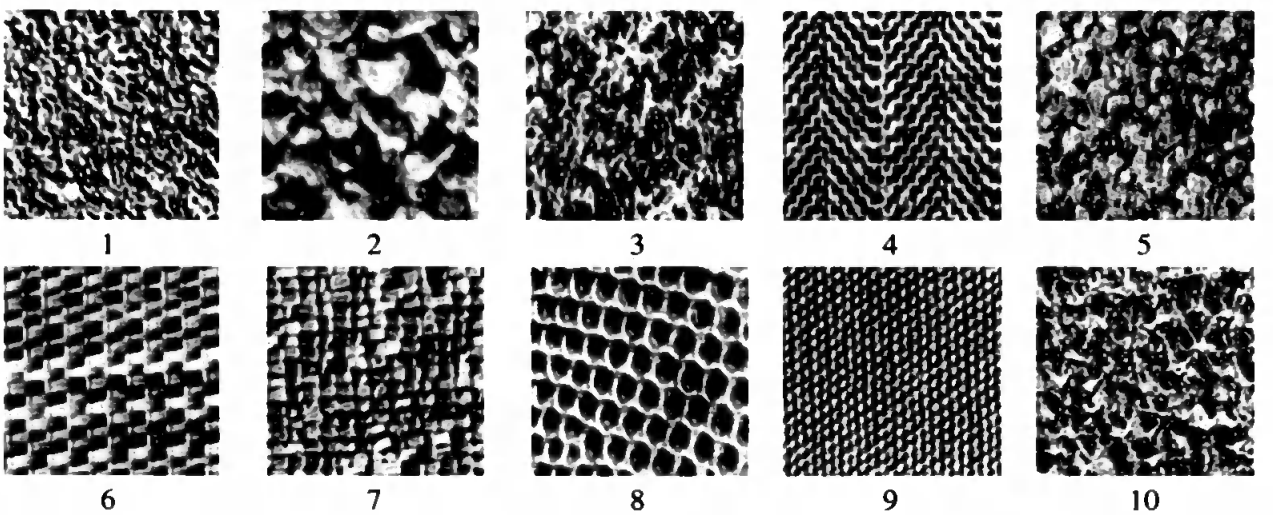


Fig. 5.12. The 10 textures, presented as 256x256 pixel greyscale images

The percentage of match between predicted and actual categories was on average 95.9 %, ranging from 90.1% to 100%. This simple test has shown the ability of the modified A Trous transform to describe textures with high accuracy.

Another test was performed, using a mosaic image of four very similar textures. The reason for this trial was to find out whether the A Trous transform of an image of non–uniform texture will maintain its ability of characterising textures. Having computed the transform of the whole image, one would evaluate the energy signatures in different areas of interest and hope that the spread of the filters applied to the data would not affect the ability of the obtained feature vectors to characterise the textures. It is evident, that in the case of the coarser scale planes, the smoothing and detail filter's kernel width (due to their upsampling on each stage of the A Trous algorithm)

can become larger than the size of the areas of different texture. Therefore coefficients on these scales would pick up characteristics of not one, but several texture regions.

In the case of the methods based on decimated transforms (like the DWPT used by Laine & Fan, 1993), the transform must be computed in every processing window in every analysed area of the image. This is due to the downsampling and to the multiple coefficient sets produced on each filtering direction by decimated transforms, as it is pointed out in chapter 3. In the case of the A Trous algorithm, the computation of the transform would not be necessary in each image region that is analysed: once the transform is computed on the whole image, one can directly analyse the coefficients in various areas, since each wavelet coefficient in a given position of the coefficient plane corresponds to a pixel located at the same coordinates of the image plane.

The used image is shown in Fig. 5.13. The images of the 4 textures in the mosaic have been histogram equalised, to eliminate possible differences in their first–order statistics. A number of 132 overlapping windows per texture region were used, these providing the energy signatures. In the DA test, the resulting average percentage of match between the predicted and actual group memberships was 90.34% for the 4 textures. This has shown that the above described effect of dilating filters on coarse scales does not affect to great extent the possibility of characterising textures with this method. This would open the way to texture segmentation applications based on such techniques.

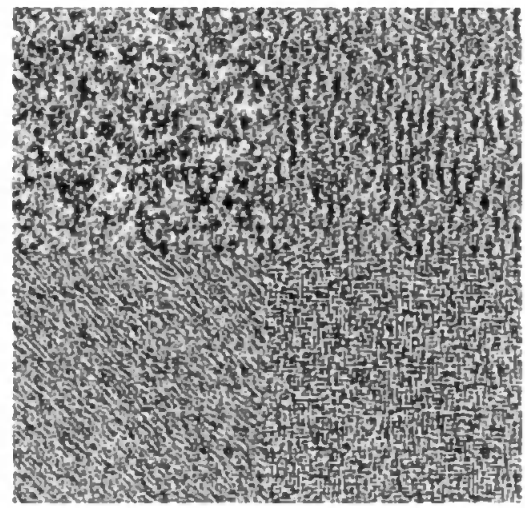


Fig. 5.13. A 4–texture mosaic

Still, a few essential drawbacks of such methods must be pointed out. The following issues were usually overlooked in papers on wavelet–based texture analysis (Laine & Fan, 1993, 1996; Unser, 1995; Lu *et al.*, 1997). In the case of 3D applications, methods based on directionally sensitive wavelet transforms have to be carefully scrutinised for their applicability. Since texture orientations change with surface orientation, such transforms must be used with extreme care when extracting texture descriptors. This is pointed out also in Van de Wouwer, 1998. A simple rotation in 2D can completely change the responses of horizontal and vertical filters to the visible details in the image, hence affecting to a major degree the feature vectors. Therefore rotation–invariant texture measures would be a possible solution to this particular problem (Van de Wouwer *et al.*, 1997), but in the context of the proposed system this would very likely mean abandoning the A Trous transform and its advantages.

Another problem is the presence of illumination gradients in the images that the proposed system has to analyse. Such slow changes in illumination (not present in idealistic planar texture images used usually as test images) are inevitably picked up by the methods described in the previously enumerated papers. Such changes are coarse scale events (low–frequency details), therefore description on coarse scales of texture by wavelet coefficients could be compromised. The above tested A Trous algorithm–based method has the same drawback. As an example, the images presented in Fig. 5.14. were submitted to the above described texture description method. The two images were classified with 100% accuracy by discriminant analysis based on the wavelet energy signatures collected from 121 coefficient windows.

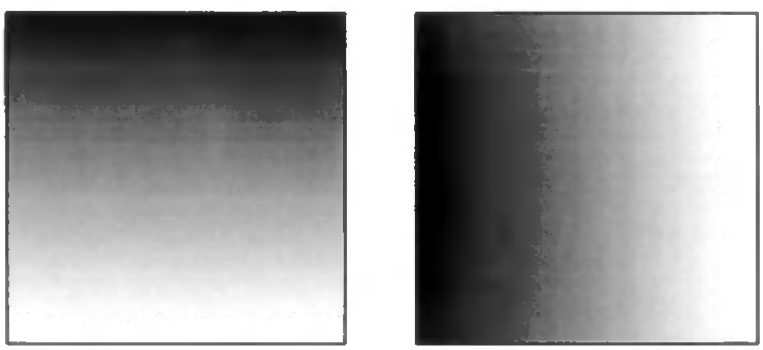


Fig. 5.14. Vertical and horizontal gray–scale gradients (256 grey levels)

It is clear, that pre–processing is necessary if one uses the above described methods for obtaining

texture descriptors. In practice, due to the lack of *a priori* knowledge on the categories or even the number of categories of textures present in an input image, unsupervised methods should be preferred when proposing segmentation of texture regions (e.g. Lu *et al.*, 1997).

Although the A Trous transform's combination with simple energy measures proved to be an elegant texture description technique, due to the above outlined problems with directionally sensitive wavelet transforms, a more robust texture descriptor was sought.

5.6.2. Texture density descriptors from wavelet coefficients

Although textons can not be considered a definitive model for texture description and their visual discrimination, the work of Julesz & Bergen (1987) helped in arriving at a structure of the texture channel in the design of the proposed system.

As Julesz and Bergen describes, psychophysical experiments have shown that global texture measures (like Fourier spectra, global statistics) are not used in preattentive texture classification (i.e. a subject identifies the different texture regions in an image in a very short time that doesn't allow close scrutiny of the image details). The authors conclude that the density of the texture elements (textons) are used by the visual system in these situations. Discrimination seems to be based on "local granularity differences that correspond to differences in local density (number) of elongated blobs of certain sizes and orientations". No definitive neurophysiological or psychophysical proof is available in the support of this theory, but such texture descriptors seemed to be a simple way of collecting information on the surface textures present in an image. Density measures would be sufficiently insensitive to changes in surface orientations.

The implementation of blob and edgelet detectors (like the ones that were suggested in Marr's model of early vision and are supported by physiological evidence quoted by Julesz & Bergen) would evidently mean additional image processing and feature extraction operations that would be unrelated to the already established framework of the proposed system. These detectors would identify textons, i.e. edge terminations, crossings of line elements, edge elements, blobs. On the other hand, wavelet local maxima detected on coefficient planes corresponding to very fine scales mark the location of small–size singularities in the image. Light intensity discontinuities grouped into granules, small groups of pixels of different intensity at these fine scales would be close to what a texton is understood to be. Such singularities causing wavelet local maxima, the density

of these maxima would give one a descriptor for robust characterisation of textures.

5.6.3. Extracting and coarse coding texture information

These density descriptors are extracted in windows centred around wavelet maxima detected on the 4th coarsest scale plane. This channel therefore is similar to the junction and spatial frequency channel in its operating mode. The size of the processing window has been set to 32x32 coefficients.

Local maxima are counted inside these windows on the finest 4 scale planes. The resulting 4–dimensional vector would give an information on the density of image singularities of sizes less than 16 pixels. The multiscale density measures are expected to give a robust description of the textures, although the one calculated on the finest scale plane is susceptible to noise. Since texture orientation is not registered by this method, the obtained descriptor was expected to lead to considerably poorer classification of textures compared to the previously described method. Indeed, the density descriptors applied to the 10 Brodatz textures shown in Fig. 5.12. led to an average classification accuracy of 44 % in discriminant analysis trials, the worst performance being 11% (just above chance level) and the best being 85.3 %.

The texture descriptor vectors are propagated through a self–organising map, in a similar manner to that described in the case of the junction and spatial frequency channel. The obtained node activation signatures are used in chorus with the data provided by the other data channels, being presented to the categoriser module that will produce an object label based on the analysed 2D view.

5.7. Implementation details

The preprocessing operations being time–consuming and involving the processing of large amounts of data, were implemented in C++. Having in view the future expansions of the system, the code has been written in a purely object–oriented way. By designing a generic class hierarchy, starting from the most abstract level (an image) down to the particular data structures and processing algorithms, the resulting code facilitates modifications and additions of new modules by horizontal and/or vertical extensions of the class hierarchy. The implementation of this hierarchy can be found in Appendix A. The code was written for the shareware Gnu C++ compiler running

on a Sun SPARC workstation. The pre–processing, maxima detection and region growing operations take around 40 seconds to run on a SPARC 5 platform.

Based on the local maxima lists and region maps generated by the above described code, the maxima and link trees are generated by a code written in Sussex PopLog 11 (Barrett *et al.*, 1986) running under Unix. This language was chosen due to its facilities in manipulating complex, hierarchical data structures. With the developments brought to the Matlab mathematical software package, the tree generating algorithms were later re–coded in Matlab v5.1 for portability. The junction extraction algorithm runs under PopLog, while the computationally more intensive 2D FFT algorithm was implemented in Gnu C++. The texture descriptors are extracted by an algorithm implemented in C++, since it uses the A Trous transform calculated in the same software module. The coarse coding that employs self–organising maps has been implemented under Matlab v5.1 for Unix, since it provided elegant Kohonen map simulation and visualisation tools. The code sequences implementing the coarse data channels are listed in Appendix B.

The computer time required to perform all coarse data coding and data file preparations for the classifier module in an essentially sequential processing setup was on average 3 minutes on a Sun SPARC 5 workstation.

5.8. Conclusions

This chapter described the system components that are responsible for the preprocessing of the images presented to the system as input. The four coarse data channels have been presented in detail, the feature extraction and coarse coding operations being described. A novel scale–space descriptor was presented, which is expected to register salient characteristics of object shapes in conditions of changing viewpoint, non–rigid shapes, poor quality input images. The proposed way in which the attention of the system is directed towards areas of potential interest in the image was described, together with the coarse data channels that are controlled by this mechanism. The possibilities of describing textures by wavelet coefficients were discussed and a rotation–invariant robust texture density descriptor was proposed. Having taken a number of design options meant to make the feature descriptors robust, it is expected that this set of coarse data channels can lead to good performance in the classification of man–made and natural objects. The concluding section of this chapter presented the solutions that were chosen in implementation stage. Having described the coarse data channels, the module that categorises the coarse coded feature data

is to be described in the next chapter.

Chapter 6. Categorising the data

6.1. Introduction

The coarse coded feature information provided by the coarse data channels is presented as input to the system's categoriser module. Statistical and artificial neural network–based categorisers were used in the testing of the system, the following sections describing these categorisers. The main theoretical considerations are pointed out, with emphasis on the essence of each technique and on the aspects that had to be taken into account when using these categorisers. The mathematical background of discriminant analysis is described, together with its characteristics and way in which it was used in tests. The fundamentals of multilayer feedforward neural networks are presented, followed by the description of the training and testing methods. A brief description of the implementation of the various categorisers is given in the concluding section of the chapter.

6.2. Classification

The purpose of the system is the classification of feature data obtained from 2D views of objects into multiple categories. These categories correspond to the 3D objects that the system is able to recognise.

The categoriser module therefore has to learn a set of data (called training data) that characterises these objects and then has to be able to generalise to novel data based on the learnt set. It is virtually impossible to present to a classifier all possible aspects of the data belonging to each of the categories to be learnt. Still, one has to construct the training set in such a way that it becomes a good representation of all data categories. If categories are poorly described, the system will find it difficult to generalise to novel instances of the categories. In order to ensure that all categories are well represented by the training set, one can select training data samples that cover the whole range of variations in the data. In the case of object recognition, one could insert into the training set views of objects collected from a limited number of viewpoints that span the whole viewing sphere.

It is more likely though, that one has no information on how well the available samples describe

the entire variety of the data categories. In this case, when building the training set, one selects randomly the training items, hoping that the selected data will be sufficiently representative of the categories.

When the goal is object recognition based on 2D views, the training set would contain views of objects collected from a range of viewpoints, so that the system can learn the various aspects of these and later can generalise based on this information. During training, these images are presented to the system as input and the known object category that a given image belongs to is used in the training algorithm. Due to the latter element, this process is called supervised learning, since we impose on the categoriser what decision it has to take in order to correctly classify the input.

Once the training of the system is complete, it hopefully learnt the training data and can classify this with high accuracy. But in practice, one is interested in the ability of the categoriser to generalise to novel data based on the learnt descriptions. Being confronted with a previously not 'seen' view of an object, it has to classify it into one of the learnt object categories, hopefully with good accuracy. This generalising ability of a system is assessed by using a so–called test data set, that contains –as a fundamental rule– data not presented to the system in training stage. When testing the system, the test data set is used in conjunction with the categories that each test data pattern belongs to. In this way, by presenting the whole test set to the trained system and comparing the known categories to the decisions taken by the categoriser, one can get a quantitative measure of the system's generalising ability.

The next sections describe two types of categorisers used in the proposed object recognition system, namely discriminant analysis and feedforward neural networks.

6.3. Discriminant analysis

The existence of usually two multivariate data sets –the training and test set– constitutes the starting point of discriminant analysis. The items in the training set belong to groups (categories) known *a priori*, each multivariate training set item being associated with a group label. The purpose of DA is to construct a mathematical model that allows one to correctly separate the training data into these groups. The next step is to take the items in the test set (whose group membership is not known) and categorise them with the help of the already constructed model.

6.3.1. The method

The mathematical model in the case of G groups is based on the linear discriminant function proposed by Fisher (1936). A discriminant function is a linear combination of the variables in the multivariate data set and its main property is that it separates the groups as much as possible. A detailed description can be found in Rao, 1952. In the subsequent paragraphs, the main points of the analysis are described.

It is considered in the subsequent descriptions, that a data item is a p–dimensional column vector, $x^{(g)}$ is the set of data items that belong in group g. Therefore any $x^{(g)}$ is a p rows by n_g columns matrix, where n_g is the number of data items in group g. The $\bar{x}^{(g)}$ p–dimensional column vector is the vector mean of $x^{(g)}$. As initial conditions, the data must have a multivariate normal distribution and a common covariance matrix.

When constructing the model, first the within–groups covariance matrix (of size pxp) is calculated:

$$S = [s_{ij}] \quad ; \quad s_{ij} = \frac{1}{n-G} \sum_{g=1}^{G} \sum_{k=1}^{n_g} (x_{ik}^{(g)} - \bar{x}_i^{(g)})(x_{jk}^{(g)} - \bar{x}_j^{(g)}) \tag{6.1.}$$

where i,j=1,...,p and n is the total number of data items (the sum of all n_g).

The covariance matrix of the G group means is called the between–groups covariance matrix (of size pxp):

$$B = \frac{1}{G-1} \sum_{g=1}^{G} n_g (\bar{x}^{(g)} - \bar{x})(\bar{x}^{(g)} - \bar{x})^T \tag{6.2.}$$

where $\bar{x}$ is is the mean calculated as:

$$\bar{x} = \frac{1}{n} \sum_{g=1}^{G} n_g \bar{x}^{(g)} \tag{6.3.}$$

When calculating the discriminant functions, one tries to maximise the distance between group

means as compared with the variances. Like in the case of Fisher's original linear discriminant function for the separation of two groups, the aim here is to seek an a (a p–dimensional column vector) that maximises the ratio:

$$v = \frac{a^T B a}{a^T S a} \tag{6.4.}$$

As Gnanadesikan *et al.*, 1988 points out, the a vectors are the eigenvectors corresponding to the eigenvalues of the matrix $S^{-1}B$ (a number of m non–zero eigenvectors, m being the smallest among G–1 and p). A discriminant function u_i can then be written as $u_i = a_i^T x$, being a linear combination of variables that separates the groups as much as possible. When categorising the data items into the G groups, the data item x is classified into group i if:

$$\sum_{j=1}^{m}\left[a_j^T(x - \bar{x}^{(i)})\right]^2 = \min_{g=1,\ldots,G}\sum_{j=1}^{m}\left[a_j^T(x - \bar{x}^{(g)})\right]^2 \tag{6.5.}$$

In an intuitive approach, this rule means that the data x is classified into the group i that has the cluster centroid closest to x in discriminant space defined by the discriminant functions.

The success of the model in separating the multivariate data into the known groups is usually checked by the so–called resubstitution procedure. Each training set item is categorised based on the model constructed from the training set. The number of misclassified training items is the apparent error– the larger this is, the poorer the constructed model's ability is to correctly separate the data into the known groups. In the literature on discriminant analysis it is pointed out that if the number of available data items per group is not much larger than the number of variables in each data item, the apparent error can be very misleading (Gnanadesikan *et al.*, 1988) . No far–reaching conclusion on the validity of the constructed mathematical model can be drawn based on the apparent error in this situation. An alternative in this case of small data set is the use of the leave–one–out method, where the model is calculated from all data items but one, then the item left out is classified based on the model and the correctness or the error of the categorisation is recorded. The above procedure is repeated for all data items, and the sum of misclassified items will become the error measure at the end. This is a far better descriptor of the model's ability of correctly grouping the data.

When a large data set is available, the categoriser's ability to generalise to unknown data can be tested by splitting the data set into training (model) and test sub–sets. The mathematical model (the discriminant functions) are built based on the data items in the model set. Then generalisation can be assessed on the test data, the classification accuracy being a measure of how well the test data items are assigned to their correct categories based on the mathematical model built previously.

The DA method has the drawback of being sensitive to outliers present in the data. The presence of such data items that do not belong to any of the groups evidently affects the resulting mathematical model used for classification. Also, significant differences in group size (i.e. the number of data items in each group) affect the model. Furthermore, one has to make sure that the number of data items per group is significantly larger than the dimensionality of the data (i.e. the number of variables), as it has been described above.

The advantage of the method is the rapid computation of the discriminant functions – unlike neural network–based categorisation techniques, lengthy training procedures are avoided. It is a robust method, that needs very few assumptions about the data and can give an estimate on how well the data is separated into groups. It is a widely used procedure to plot the projections of the multivariate data set onto the plane defined by 2 discriminant functions. Usually the ones that account for the largest variance in the data are used. Such a diagram can visualise the separation of the data into groups.

6.3.2. Using discriminant analysis in tests

In the proposed system, DA has been used when evaluating the ability of feature data provided by certain data channels to characterise 3D objects. When changing structural parameters in the system, this statistical classifier allowed fast evaluation of the effects of these changes and therefore the arrival at a set of parameters used in more sophisticated tests. In these preliminary trials, resubstitution and leave–one–out classification were used, depending on the number of available data items.

Also, as a robust classifier method, it has been used in categorising the chorus of feature data provided by several coarse data channels and checking the resulting category labels against the known object categories that the input 2D views come from. The data sets have been split into

training and test sets prior to these tests, therefore the ability of the classifier to generalise to unknown data could be tested. The mathematical model is built based on the training (i.e. model) data, then the classification accuracy is tested on the test set. The procedure involves classifying each item in the test set based on the discriminant functions obtained from the model data. Parallels between the performance of this method and of neural network–based categorisation techniques could be drawn, as it will be described in future chapters.

6.4. Artificial neural networks

An artificial neural network (ANN) is a connectionist system: it consists of units called artificial neurons that are connected to each other by links and work in parallel. They function in a similar way to the biological neurons, by transferring neuron activations from one group of neurons to another via the links that in essence resemble the synapses. One of the simplest and most widely used neural network architectures is the multilayer feedforward network, described in the next section.

6.4.1. Multilayer feedforward neural networks

The building block of any neural network is the artificial neuron. This is a simple computational unit, that takes several inputs and yields their weighted sum as output, as schematically represented in Fig. 6.1.

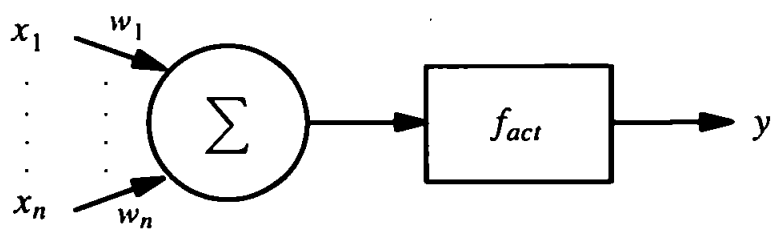


Fig. 6.1. An artificial neuron with n–dimensional input x_i, corresponding weights w_i and activation function f_{act}.

The output of such a neuron can be expressed analytically as:

$$y = f_{act}(\sum_{i=1}^{n} x_i w_i) \tag{6.6.}$$

where the activation function f_{act} can take several forms (Masters, 1993). Perhaps the most widely used activation function is the logistic function, that is defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{6.7.}$$

This has as effect the compression of the dynamic range of the data, making sure that the weighted sum of the inputs does not produce extremely high values that would affect operations in the network.

In a feedforward network, multiple layers of neurons are connected to each other, the information flowing from the input to the output of the network in one direction (hence the name of the network). Such a network, where each neuron in a layer is connected via links to every neuron in the next layer, is shown below in Fig. 6.2.

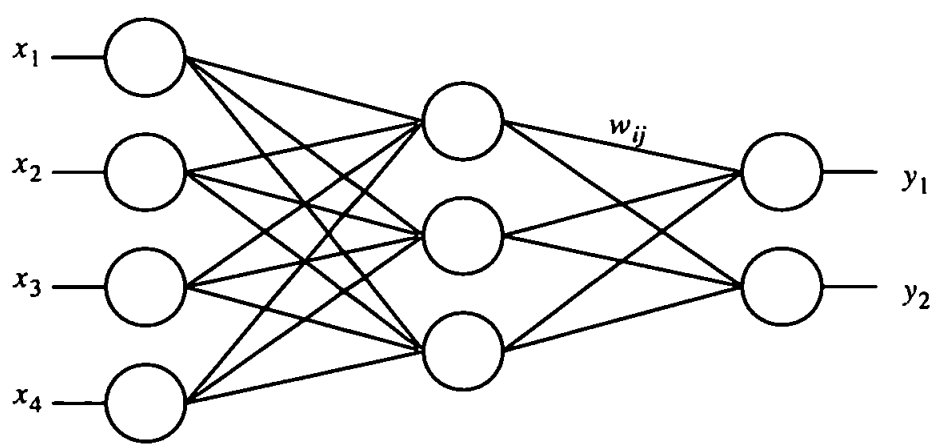


Fig. 6.2. An example of multilayer feedforward neural network.

The first layer is called input layer. In the usual case of a multidimensional input, there is one input neuron for each variable. Such neurons have the task of transmitting the input activations to the next, so–called hidden layer. Depending on the application, more than one hidden layers can be present in the network's structure. The last layer is called output layer, the output of the neural network consisting of the activations of these neurons.

For instance, in a classification task one would expect the neuron corresponding to a category to be activated, while the rest of the output neurons to be inactive. In such a task, the network is expected to learn the categories from the data presented to it in training stage and to generalise to unknown data, as it will be described below.

6.4.2. Training with error backpropagation

The main task in training is to adjust the weights of the links in a network in such a way that one ends up with a well–defined, desired system behaviour. A first step in a practical application is to train the network to produce desired output activation patterns for the input patterns selected as training data. A frequently used measure of how well a network learnt a particular training data pattern p is the mean square error (MSE) defined as:

$$E_p = \frac{1}{n_o}\sum_{i=1}^{n_o}(t_{pi} - y_{pi})^2 \qquad (6.8.)$$

where n_o is the number of output neurons, y_{pi} is the activation of the output neuron i obtained for pattern p and t_{pi} is the so–called target activation of the same neuron (the activation that is expected to appear for the pattern p presented as input to the network). The mean E of all E_p calculated for all training patterns presented to the network is used as a measure of the error of the network, this showing the extent to which the network learnt all training data items (Masters, 1993). The aim in training is to minimise this error, i.e. to seek the global minimum in the error E.

Training occurs in multiple steps, following usually the Hebbian learning rule (Hebb, 1949), according to which the weight of a link that connects two active neurons is increased (i.e. the link is strengthened). In the case of the feedforward neural network used in the categoriser module of the system, the simple error backpropagation algorithm was used for training. A detailed description and discussion of training methods can be found in Rumelhart & McClelland, 1986. In this section, the main points of the procedure are highlighted.

Before the training algorithm is started, the weights of the network are initialised with small random numbers (usually in the range [–1, 1]). During one stage of the algorithm, an input pattern x is presented to the network. The activations are calculated from the input layer towards the output layer (i.e. the input is forward propagated through the net). For the given input pattern, one

expects a certain pattern of activations in the output layer – this target pattern is compared to the output layer activations calculated based on the presented input.

We'll denote the error (difference) between the calculated and desired activation of an output neuron j with δ_j. Based on this error and the activation of the neurons i linked to the output neuron j, the changes in the weights w_{ij} are determined. In a similar way, the error is backward propagated to the previous layers, according to the following rule that defines the modification that is brought to a given weight that links a neuron i of a certain layer to a neuron j situated in the next layer:

$$\Delta w_{ij} = \alpha \delta_j y_i \tag{6.9.}$$

where α is the so–called learning rate constant, y_i is the output of the neuron i and δ_j is the error. The latter is calculated depending on which layer the neuron j is in. If this is in the output layer, the error is calculated according to:

$$\delta_j = f'_{act}(u_j)(t_j - y_j) \tag{6.10.}$$

where the argument of the derivative of the activation function is the input to the neuron j and t_j signifies the target activation required from this neuron. If the neuron j is in a hidden layer, then the error is calculated as:

$$\delta_j = f'_{act}(u_j) \sum_k \delta_k w_{jk} \tag{6.11.}$$

where k is the index of a neuron that is in a layer that follows the one in which neuron j is situated. A more sophisticated version of the above algorithm is the backpropagation with momentum term. In the case of this training algorithm, when updating a weight, the previous change brought to this weight is taken into account (a kind of inertia is introduced into the training). Therefore the update formula (6.9.) takes a modified form:

$$\Delta w_{ij} = \alpha \delta_j y_i + \mu \Delta_{ij}^{(p)} \tag{6.12.}$$

where μ is the momentum term and $\Delta_{ij}^{(p)}$ is the change brought to the weight in the previous update. The benefit of having a momentum in training is that it acts as a low–pass filter in the search for a global minimum in the network error (Masters, 1993). By adding the previous weight change to the calculation of the current weight update, it does not allow fast fluctuations in weight changes. In the case of the simple update rule expressed in equation (6.9.), the training algorithm might miss the global minimum of the error E by bringing large modifications to the weights, hence needing further lengthy search for that global minimum. Such an 'overshoot' can actually increase the error of the network.

The training algorithm can have two distinct forms. In the so–called on–line learning, the modifications to the weights are brought after each presentation of an input pattern and error backpropagation. In batch learning, the necessary weight changes are summed up and the modification of the weights happens after all input patterns are presented to the network.

A cycle in which all input patterns are presented to the net, errors are backpropagated and weights adjusted is called an epoch. In training, one uses a large number of epochs until reaching a satisfactory low error. The purpose of the training is therefore to minimise the error between the desired outputs and the ones that are calculated by propagating the input.

In classification, the most attractive property of a neural network trained on a data set is its ability to generalise to new data which hasn't been presented to it during learning. In practice, one is interested especially in the network's accuracy in categorising novel data based on what it learnt in training stage.

6.4.3. Training and testing the network

In training, due to the way in which the error backpropagation learning works, a few important factors have to be taken into account when deciding the size and structure of the network, the value of the learning parameters and the way in which data is presented to the network during training.

During training, the patterns in the training data are usually presented to the network in a random order in each epoch. This is motivated again by the network's learning. If the data patterns were presented in the same order to the network in each training epoch, due to the weight update pro-

cedure that always tries to fit the weights according to the currently presented training data, the first few categories learnt in early stages of the training epoch would be 'forgotten'. This would lead to an increase in the network error. Another practical issue is that of the dynamic range of the input data. In practice, the multivariate input is usually normalised so that the dynamic range of each variable becomes similar. As Masters, 1993 points out, since the range of value of the weights is limited during training, extreme values in the data can affect negatively the training procedure.

Another issue related to the way in which the training and test data is prepared is the group size. If the number of patterns belonging to different groups is very different, a larger group will produce a bias in learning. It is evident, that the network will adjust its weights to learn the larger group, while much smaller groups in the training set will not be able to induce sufficient changes in these weights. As a result, the performance of the network on test data will be poor on the groups not sufficiently represented in the training data.

Regarding the size of the network, one has to decide how many neurons to use in the hidden layers and how many hidden layers to include in the network structure. Both parameters affect the way in which the network behaves during training. A well–known negative phenomenon is the so–called overfitting. If the number of hidden layer neurons is large, the network will be able to learn unimportant characteristics of the training data. This is due to the fact that such a large network will possess a huge information processing capability, being able to adjust its many weights in such a way that the resulting weight configuration will be able to register all patterns in the data, no matter how insignificant they are (Masters, 1993). Once this happens, the ability of the network to generalise to unknown data will be seriously compromised. As an end result, the performance of the network on test data (not presented to it before) will be poor. In the opposite situation, if the number of neurons in the hidden layer(s) is small and the training set is very large, the few weights of the network might not be able to capture important characteristics of the data.

Since there is no universal rule based on which one can calculate the number of hidden nodes from the training set size, input dimensionality etc., the choice of a particular number of hidden nodes depends on the application and can be arrived at by training/testing runs meant to optimise the network structure. During such runs, one can choose a hidden layer size that leads to satisfactory performance. Usually, the situation in which the number of hidden nodes is equal to the number

of output nodes is avoided, so that the network builds a more effective general model of the data (Masters, 1993).

The other aspect regarding the network's size is the number of hidden layers in its structure. As it is pointed out in Masters, 1993, more hidden layers make the learning process unstable by adding local minima to the surface of the error function E. This can compromise the search for a global minimum. Usually one hidden layer is sufficient for successfully training the network.

In order to find a compromise as far as the network size is concerned, in practice one starts with few hidden layer neurons and by repeated training/testing of the network, gradually increasing the number of these neurons and monitoring network performance.

With this method involving preliminary network simulations, it is possible to arrive at a practically acceptable solution. In the proposed system, the neural networks employed as categorisers were designed with this method. During these preliminary trials, the values of the learning rate and the momentum term were chosen by monitoring network performance. A too large learning rate causes sudden, large changes in the weights, this affecting negatively the search for a global error minimum. Also, the momentum term defines the smoothness of the update procedure. The aim was to obtain a network that presents a satisfactory generalisation ability to novel data and does not increase simulation time enormously.

During training, a sufficiently large number of epochs are used so that in the last epochs no significant improvement of the network's performance (measured on the training set by the error E) can be observed. This does not give a definitive assurance that the network will perform in the best possible way on the novel (test) data, but it shows that as far as the training set is concerned, the data has been learnt. To use the test set in deciding when to stop the training (by monitoring the network error on the test set during training) would violate the rule according to which the training and test sets must be separated.

In a practical situation, when testing a classifier system based on feedforward networks trained with error backpropagation, several training/testing runs are performed. Since usually the network starts with randomly initialised weights, one trials involving training and testing does not give a full picture on the system's performance. Due to the initial weights, the network might not learn in a satisfactory way, i.e. the search for a global minimum can fail. Therefore a sufficiently

large number of trials, each involving random initialisation of the weights can offer a better picture on the general behaviour and performance of the system.

Due to the structure of the system, more precisely the existence of multiple data channels, the above described feedforward networks were used as building blocks in more complex classifiers, namely collective and committee machines.

6.4.4. Collective machines

A collective machine (Ellis *et al.*, 1997) is a simple architecture where a categoriser (a neural network) is presented with the ensemble of feature data supplied by a number of data channels. Therefore a decision regarding the category represented by the data is brought based on the joint description of features.

Such a categoriser architecture strikes a chord with the theoretical issues discussed in chapter 2, regarding the way in which features seem to be selected and used in biological vision systems during recognition. A collective machine being presented with a chorus of various feature data extracted from the input image, a decision regarding the object's category is brought based on this ensemble, individual channels providing feature information that contribute in variable degree to the success of the recognition. One can imagine a situation where two object have similar shapes, but very different surface texture – in this case, the most salient information about the nature of the objects will be supplied by the texture channel. In other circumstances, for the correct recognition of an object, other channels would play an important role. No *a priori* ranking of the channels is taken into considerations and no rules on which channel's data is used in what conditions are imposed on the system. All decisions are made based on operations in an N–dimensional feature space resulted from the concatenation of individual channels' data.

In the proposed system, the data provided by the four coarse channels has been concatenated into a feature vector and this presented to the categoriser. In previous work on automatic plankton classification (Culverhouse *et al.*, 1996; Ellis *et al.*, 1997) it has been shown that the grouping of data channels improves classification performance, even a poorly performing individual feature channel has a positive effect on the recognition accuracy when joined with other channels.

In a similar way, the effects on system performance of the superposition of the four coarse data

channels have been investigated in the present work. The recognition accuracy of the system was assessed in conditions where data channels were grouped together and it was expected to improve when adding more channels to the categoriser input.

In tests, a different architecture called committee machine has been also used to assess the role of each data channel in leading the categoriser to correct classification.

6.4.5. Committee machines

In contrast with the collective machine, in a committee machine there is one categoriser for each data channel and a final decision regarding the class of the input is taken based on the decisions of these categorisers. The structure is illustrated below in Fig. 6.3., showing the situation where a neural network is associated with each of the coarse data channel outputs.

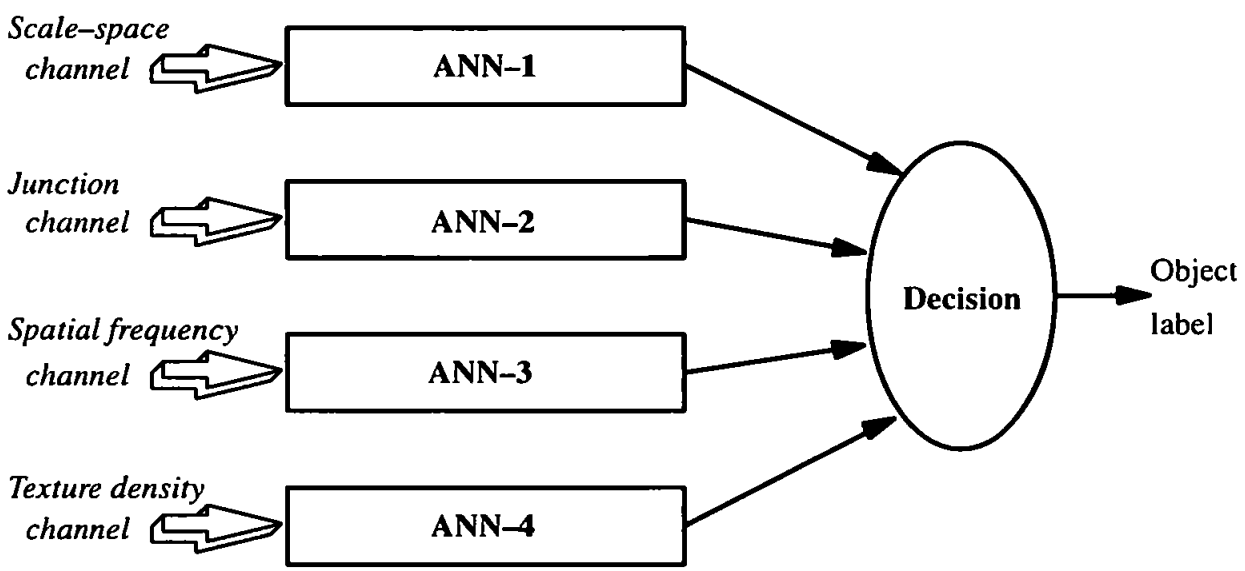


Fig. 6.3. The structure of a committee machine realised with neural networks.

The rules according to which the final decision is taken can vary. Perhaps the most simple method is to regard the individual verdicts of the neural networks associated with each data channel as votes and classify the input according to the majority's decision (for a review and comparative studies, see Battiti & Colla, 1994). The risk taken when using such a decision rule is that a majority of poor categorisers (i.e. those associated with data channels that in a given situation do not

provide sufficiently salient descriptions of the input) can seriously reduce the performance of the system.

More sophisticated decision rules can be used instead, e.g. the decision of the most confident categoriser is taken as final decision. In this context, confidence is measured by the difference between the highest and the second highest neuron activation found in the output layers of each individual classifier. Also, as an alternative, one can sum the corresponding output neuron activations of all individual channel categorisers and choose as committee decision the category corresponding to the highest summed output activation. In the literature, the use of more complex committee machines has been reported, where the output of the committee is decided by another categoriser trained with the outputs of the channel categorisers or an adaptive system (Wolpert, 1992; Jacobs *et al.*, 1991). The benefit of these methods, as Ellis *et al.*, 1997 points out, is that in comparison with the voting procedure, poor classifiers have less influence on the final decision than the good categorisers (associated with data channels that provide very good description of the input).

During the object recognition system's testing, committee machines were used for evaluating the role of each data channel in correct classification. The effect on the system's performance of the combination of channels could be studied in this way. Here, an interesting issue seemed to be the behaviour of the categorisers when combining channels with poor discriminatory power with 'good' channels. Also, by comparing the performance of the system employing collective machines with that of the one using committee machines as categoriser, the studies done by Ellis *et al.* (1997) could be extended to the field of 3D shape recognition.

6.5. Implementation details

As statistical categoriser, the discriminant analysis implemented in the SPSS v7.5 statistical package was used. The neural network–based collective machine was implemented with the shareware Stuttgart Neural Network Simulator (SNNS v4.1) under Unix (Zell *et al.*, 1991). This allowed a considerably faster simulation of network training and testing, when compared to the ANN toolbox in the Matlab package. This helped in establishing the learning parameters and network sizes by fast preliminary tests performed on various network structures and with different training parameters.

Committee machines were also implemented with the SNNS package, the final decision taking algorithm being written in Matlab under Unix. The latter had the role of taking as input the result files saved by SNNS and based on the output activations of the neural networks associated with the individual coarse data channels, calculated the decision of the committee machine. Listings of the Matlab code can be found in Appendix C.

6.6. Summary

The issues of classification, statistical and neural network–based categorisers have been described in this chapter. Following the brief presentation of the background of classification, the mathematics and the use of discriminant analysis as statistical categoriser has been described. The fundamental concepts behind neural networks, their training and testing were presented, together with the possible categoriser architectures that can be built based on these and multiple data channels. Following the description of each categoriser, the practical issues of their use and the reasons behind their utilisation in the proposed system was highlighted. This chapter, that presented the last component module of the object recognition system, prepared the ground for the description of the ways in which the system has been tested and its performance evaluated, these issues being the subject of the following chapters.

Chapter 7. Data sets and evaluation methods

7.1. Introduction

The behaviour of the proposed object recognition system has been tested in a variety of conditions. The image data sets used in these tests are described in the following section, the characteristics of each data set and the problems posed by them in recognition tasks being discussed. The computer–generated objects and their rendering is described, followed by the presentation of the natural images. The subsequent sections present the evaluation methods used in the assessment of the system's performance and behaviour in various conditions. A number of issues regarding the suitability of certain performance measures are discussed. In subsequent sections, the basics of the employed statistical measures and methods are described, together with the reasons that motivated their use. Also, the conditions in which they can be used are pointed out.

7.2. The image data sets

In experiments, initially two synthetic (computer–generated) image data sets were used as test data. Since at the time of the tests, suitable data sets were not available to us, these images were generated by taking into consideration the objectives of the tests. These data sets generated with Autodesk 3D Studio are described in the subsequent sections, together with the Aberdeen data set that contained field–collected images of marine lifeforms.

7.2.1. Constraints on the test data

The test objects and their views used in the classification experiments had a number of constraints imposed on them. These were the following:

- The image objects' surface area changes with up to 10%, as the viewpoint changes. This allows for non–exact recognition. Since by design, the current version of the scale–space channel is sensitive to large variations in scale due to the fixed number and location of scale planes taken into consideration during the analysis, the test views had this constraint

imposed on them. This made possible the testing of the 'where' channel's fundamental properties and behaviour in controlled conditions.

- The objects had to be very similar and where possible, their similarity had to be assessed quantitatively by human subjects. The computer–generated test objects had similar geometry, human subjects providing numeric scores that described the inter–category similarity. Such quantitative measures allow one to draw parallels between the human perception of inter–object similarity and the system's performance. The natural images represented very similar objects that were difficult to be categorised by human experts. These difficulties were introduced in order to test the system's coarse representation in situations close to those that the system was designed for (e.g. classification of marine biota).
- Each data set contains a single category that is very different from the others. This tested the coarse data channels' ability of yielding descriptions that helped the categoriser in recognising with very high accuracy this category in all conditions. Such a feature of the data sets was introduced having in mind the practical situations where certain categories present in the input must be very accurately rejected with minimum effort. Such a situation would be the rejection of detritus images in the analysis of field–collected marine data.
- A single object must be present in the scene. The current implementation of the system only makes possible the analysis of one shape, future extensions to multiple object analysis being outlined later in this work.

Having pointed out these constraints, the subsequent sections describe each test data set in detail.

7.2.2. The 8–object data set

This data set contained computer–generated 2D views of 8 synthetic objects in the form of 256x256 pixel images, that were rendered gray–scale before submitting them to the system for recognition. The purpose of this set of images was the fast evaluation of the ability of the scale–space channel to characterise 3D shapes.

When designing the 3D scene containing light sources, camera and the synthetic objects, a number of issues related to the properties of the resulting 2D views have been taken into account. First of all, in the 3D scene containing omnidirectional and ambient light sources the same light source and camera setup was used for all image rendering. In order to minimise the effect of scaling on the wavelet transform maxima, hence on the resulting link trees, the objects were constrained to the same height and radius (where geometry allowed). All objects had the same matte surface texture. These constraints were to minimise the dissimilarities between objects (apart from their shapes). The scene is described by the data listed in Table 7.1.

Table 7.1. The 3D scene for the 8–object synthetic data set

Camera:			
Focal length	78 mm		
Position	X:347	Y:–354	Z:373
Target	X:0	Y:0	Z:0
Bank angle	0°		
Ambient light:			
Colour	Red=0.33	Green=0.33	Blue=0.33
Direct light:			
Position	X:2490	Y:1150	Z:1660
Colour	Red=0.71	Green=0.71	Blue=0.71

The focal length of the camera was chosen such that it does not produce strong perspective distortions. As it can be seen in the above table, a dim ambient light source was used to avoid complete shading of surfaces, the contrasts between object surfaces being assured by a stronger directional light source situated in the right of the field of view. The objects were designed to have simple geometric shapes of different similarity, as it is shown in Fig. 7.1. below.

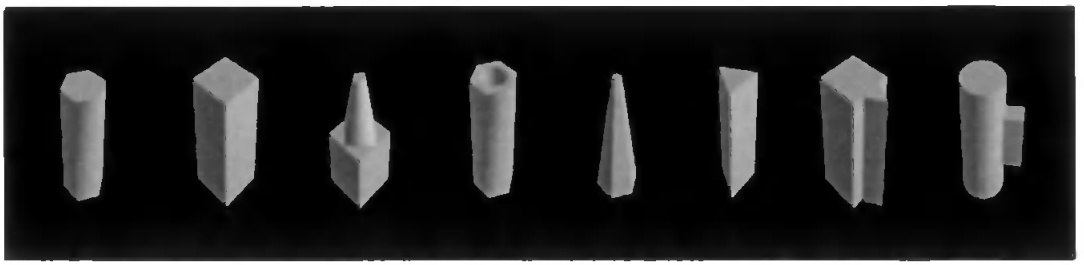


Fig. 7.1. The 8 computer–generated objects.

Objects No. 1 and 4 were designed to be very similar, the only difference between them being the presence of a cavity in the latter. Also, objects 2 and 7 have the same overall shape, the latter having an asymmetric element added to it. Objects No. 3 and 5 were designed to be more dissimilar from the rest. When generating the image data set, the objects were rotated about the longitudinal (Z) axis with +/– 10 degrees relative to their reference position showed in Fig. 7.1., a step of 2 degrees being used in rotation. The rotation about the Z axis produced self–occlusion of the asymmetric elements present in the structure of objects No. 7 and 8. Some facets became occluded, as it is shown in Fig. 7.2. The objects in their reference position were also rotated about the X and Y axis with +/– 10 degrees. Rotation in depth, hence foreshortening of the objects occurred (an 18% foreshortening of vertical vertices resulted). 15 views per object have been generated following this protocol.

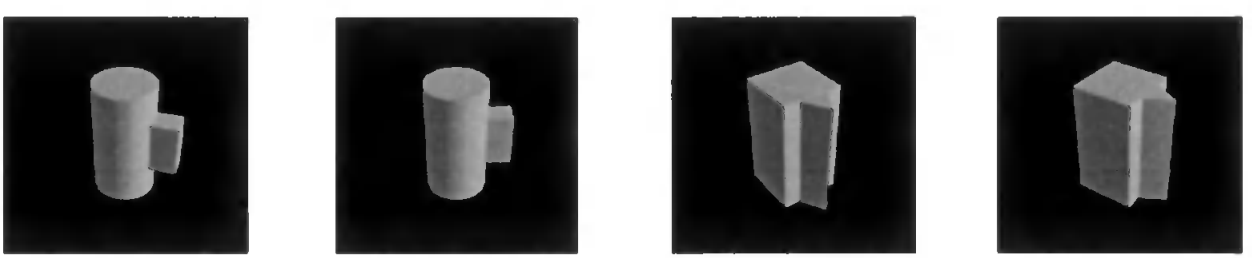


Fig. 7.2. The extreme positions of objects No. 7 and 8 after rotation about the Z axis.

This small image data set was used in tests involving discriminant analysis. As it will be described in the next chapter, properties of the scale–space channels were investigated with this data. More sophisticated trials involving large variations of viewpoint were performed on the second synthetic data set, described in the next section.

7.2.3. The 5–object data set

When designing this set of objects and rendering the views, similar considerations regarding object size and surface texture to the ones described in the previous section were taken into account. The 3D scene used in generating these images contained an animated camera, an ambiental and two directional light sources, as the scene parameters listed in Table 7.2. show.

Again, the choice made for the camera's focal length avoided distortions of the visual field. The light sources were positioned in such a way, that they produced balanced illumination of the object surfaces and revealed surface boundaries. The presence of shading was desired in order to test the robustness of the feature coding methods, but with the use of a dim ambient light source the situation of complete shading of some of the object surfaces was avoided.

Table 7.2. The 3D scene for the 5–object data set

Camera:			
Focal length	62 mm		
Start position	elevation=90°	azimuth=0°	radius=610
End position	elevation=0°	azimuth=350°	radius=610
Target	X:0	Y:0	Z:0
Bank angle	0°		
Ambient light:			
Colour	Red=0.23	Green=0.23	Blue=0.23
Direct light No. 1:			
Position	X:2450	Y:1324	Z:1780
Colour	Red=0.59	Green=0.59	Blue=0.59
Direct light No. 2:			
Position	X:–2305	Y:–1204	Z:1780
Colour	Red=0.71	Green=0.71	Blue=0.71

A set of views was generated by animating the simulated camera that spanned the upper viewing hemisphere. The camera moved along 36 paths located at 10 degrees azimuth from each another, on each such path the step in elevation angle (spanning 0...90 degrees interval) was 10 degrees. Therefore 360 images per object were generated. This set of images has been split in various proportions into two subsets for classification experiments involving training and test data. In all such experiments described in subsequent chapters, a convention was adopted for the splitting

of the data set into training and test sub–sets. The images in the data set were ordered according to the camera positions used in rendering.

The camera being animated along arcs with constant azimuth angle, the first view in the sequence corresponded to the camera start position of azimuth=0° and elevation=90° (as Table 7.2. shows). Along the arc of constant azimuth, the camera moves towards the point with elevation=0°, then moves to the next arc of 10° azimuth and so on. When using a $1/k$ part of the whole data set as test data, modulo arithmetic is used to pick the test views. From the sequence of rendered views, the ones with multiple–of–k index are added to the test data set, the remainder of the views being kept in the training (i.e. model) set. With this, a satisfactory and uniform representation of the data groups is achieved in the training set, which was a desirable characteristic in the classification experiments carried out in controlled conditions. Such a data set splitting method makes possible the study of the system's behaviour in situations where the angular gap between training and test views can be controlled (as it will be described in the subsequent chapters). In all experiments that involved the 5–object data set, this split method was used.

All objects had elements that would present self–occlusion when viewed from certain camera positions. The 5 objects are shown in Fig. 7.3.

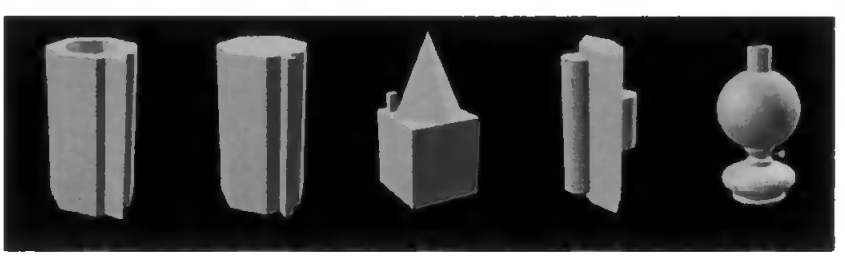


Fig. 7.3. The 5 computer–generated objects.

Two objects were designed to be very similar (objects No.1 and 2), so that discrimination would be very difficult from certain viewpoints. From viewpoints with small elevation angles, the distinctive feature is only the geometry of the asymmetric element attached to each of the objects. When these are occluded, the two objects have the same apparent shape when they are viewed from viewpoints with small elevation. While having the same height and radius, the main difference between these two objects is the presence of a cavity in the case of object No.1. Object 4 was designed to have similar elongated shape to the first two objects, while objects 3 and 5 were

meant to have clearly distinctive shapes. The latter, object No. 5 was designed to be the most different from all the other objects, with rounded shape that is subject to strong self–occlusion in views with high elevation angles. Object 3 possesses a small asymmetric element that alters the otherwise regular, symmetric shape of the object when it is visible.

The similarity of the 5 objects has been assessed by a panel of 10 human subjects who were presented with views of all 5 objects, these views showing all distinctive features of the objects. They were asked to mark the similarities between objects on a 5–point scale (following Elmes *et al.*, 1989). The mean ratings are listed in Table 7.3.

Table 7.3. Similarity of objects in the 5–object data set, as assessed by 10 human subjects (1=very different, 5=very similar)

Object	1	2	3	4	5
1	.	4.4	1.5	2.7	1.2
2	–	.	1.9	2.7	1.2
3	–	–	.	1.3	1.3
4	–	–	–	.	1.4
5	–	–	–	–	.

It is apparent, that objects 1 and 2 were judged to be the most similar, object No. 5 being perceived as the most different from the rest. Object 4 was judged to be reasonably similar to object 1 and 2, and object 3 was perceived as relatively dissimilar to other objects. These results validated the design considerations taken into account during the synthesis of the objects and their views.

This table gives a measure of the objects' similarity as perceived by human observers and it provided the means for comparing this perception with the confusion between objects reported by the system, as it will be described in the following chapters.

7.2.4. Aberdeen data set

The Aberdeen data set contained photomicrographs of fish larvae, made available to us by Paul Rankine from the Marine Laboratory, Agriculture and Fisheries Department, The Scottish Office, Aberdeen.

The images were of herring, sprat and sandeel larvae in 4 developmental stages. They have been chosen because of the particularly difficult task of discriminating between them. The specimens' similarity, their non–rigid elongated shapes, morphological variations occurring at different stages of their development and factors related to image quality present a series of problems for any classifier, whether it is an expert human taxonomist or an automatic system. The photomicrographs show the larvae in dorsal and lateral views, specimens also appearing twisted. Example specimens from each species of larvae are presented in Fig. 7.4.

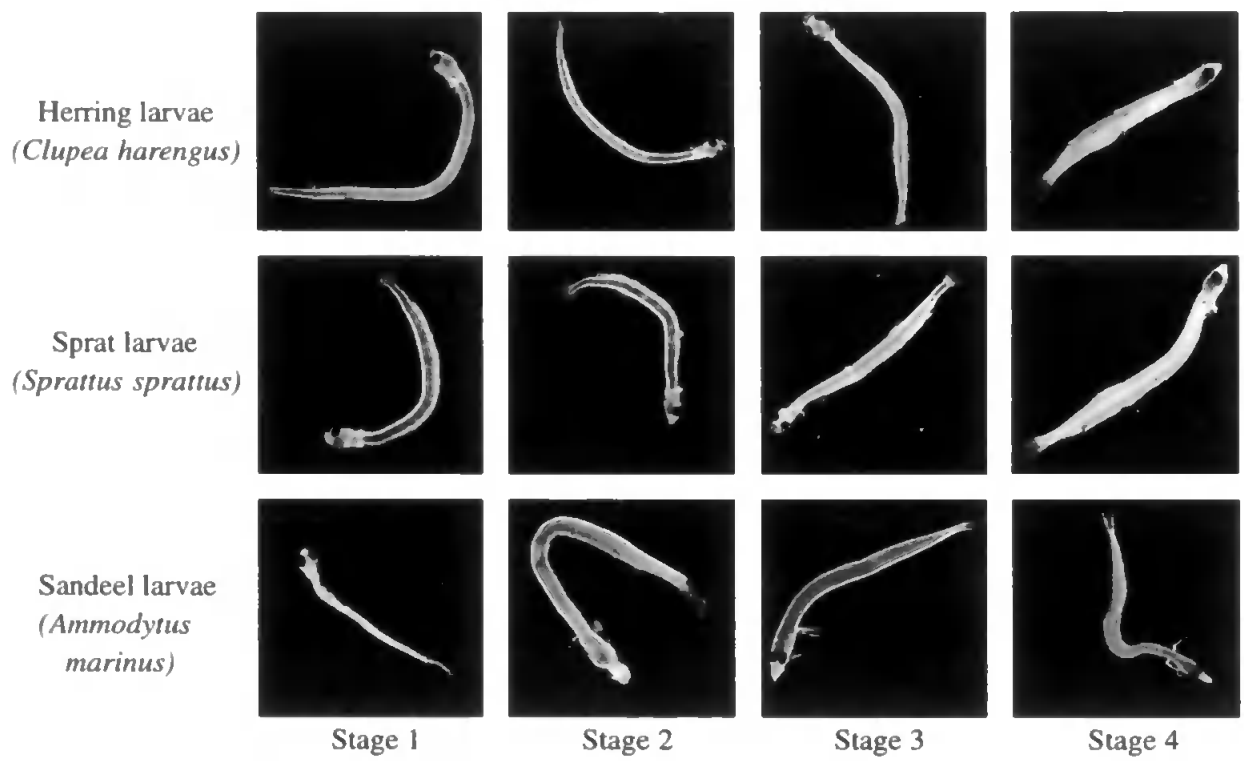


Fig. 7.4. Specimen examples from the Aberdeen data set.

Regarding the quality of the images, the following problems are encountered in this data set:

- the images are corrupted by detritus and dust present in the water sample, which in some cases appear to be attached to the larvae.
- scratches on tray surface or condensation effects are present, sometimes superimposed on the larvae.
- some larvae appear twisted or squashed, especially the ones in early stages of development.

- the presence of untypically shaped larvae (e.g. older sandeels in dorsal view have fins radically splayed outwards).

The images have uniform background and all were histogram equalised prior to analysis. Since the larvae in various stages of development have significantly different sizes, the magnification of the microscope was adjusted so that each specimen appears with similar size in the photomicrographs. Because large variations in scale affect the link trees, the approximate normalisation of the apparent size of larvae removes a factor that can affect in an unpredictable way the recognition process.

The images of detritus (various particles, planktons etc.) have been labelled as belonging to a 4th object category and were removed from the original photomicrographs. The detritus category is meant to represent a group of objects that in a real–world application (i.e. automatic classification of images in a marine laboratory) would have to be ignored in the analysis. 50 images per specie were available and after separating the detritus 1562 images in the 4th category resulted. All photomicrographs are 256x256 pixel 8–bit gray level images.

7.3. Evaluation methods

When assessing the performance of the object recognition system, a number of methods have been employed for measuring the system's classification accuracy and behaviour under conditions of changing internal configuration, feature data sets and amount of training data.

7.3.1. General considerations

A widely used way of reporting the performance of a classifier is the confusion table, which shows the number or proportion of cases belonging to a given known category, classified by the system into one of the categories. Since various ways of constructing the confusion table exist, here we adopt a convention for all subsequent descriptions of test results (when not stated otherwise).

Each row corresponds to a known category, the category memberships that the system arrived at are listed along these rows. An entry in the confusion table, corresponding to known category i (in row i) and predicted category j (column j) is listed as:

$$c_{ij} = \frac{n_j}{n_i} \tag{7.1.}$$

where n_j is the number of cases classified by the system into category j, but actually belonging to known category i and n_i is the number of cases in category i. These entries in the table will be expressed as percentages.

With these conventions, the terms on the main diagonal of the table represent the correct classification. Usually, the mean of these terms is reported in literature for characterising the performance of a classifier. This is a rather primitive measure, for two main reasons.

Firstly, the mean of diagonal terms (the mean classification accuracy) does not take into account the chance level. If the system labelled randomly the cases in the test data set, one expects to have non–zero values along the main diagonal of the confusion table that are due to chance. In order to discuss the percentages of correctly classified cases, one has to report the chance level for each category. In the case of equal group sizes, the chance of correctly classifying a case into a category (group) would be $\frac{1}{G}$, where G is the number of categories. When group sizes are not equal, the chance for category g will be equal to $\frac{n_g}{n}$, where n_g is the number of cases in group g and n is the size of the data set. In these conditions, the mean of diagonal confusion table entries can not be compared to a single chance level. In order to see how well the system was doing, the individual performances on each category must be investigated, too. A more sophisticated and meaningful measure for overall system performance across a number of categories is the kappa statistic, described in the following section.

Secondly, the mean accuracy so often used in literature might be above chance level even when the system categorises data belonging to a particular group with an accuracy that sinks below chance. Therefore it does not signal pronounced differences from group to group in categorisation accuracy. Again, the investigation of performance on each group is needed, or at least the lowest and highest recognition accuracy must be reported.

When assessing the performance of neural network–based categorisers, sometimes the mean square error between the output and the target activation for a given data item is used as measure. But this can be meaningless, since the error does not specify which category nodes are activated in what way, therefore it does not give any information on the system's behaviour when tested on different categories. A detailed discussion of these issues can be found in Simpson, 1992.

During all tests involving neural network–based categorisers, multiple training/testing runs were performed. The mean classification accuracy across all categories and for each category has been computed, together with the mean kappa statistic.

Apart from the assessment of the system's performance, the changes in the performance were studied when alterations of the experimental conditions were introduced. The effect of changing coarse data channel configuration was investigated. It had to be seen whether adding new channels with variable discriminatory power affects the performance of the system. This was regarded as an extension to the field of 3D object recognition of the work on multiple coarse data channels reported in Culverhouse *et al.*, 1996 and Ellis *et al.*, 1997. It was expected that the addition of new channels improves the performance, even when an added channel on its own had poor discriminatory power.

The effect of changing network size (number of hidden layer neurons) was also investigated by registering the variations in the categorisers' generalisation ability. The size of the training set has been varied in trials, this providing the grounds for another study on the way in which the system generalises based on variable amount of learnt data.

The presence of these effects was tested with analysis of variance, which is a well–established statistical method. This and the above mentioned kappa statistic are described in the following sections.

7.3.2. The kappa statistic

As a measure of agreement between two raters, the kappa statistic was proposed by Cohen (1960). This constitutes a more sophisticated alternative to the simple average of confusion table diagonal terms, since it takes into account the agreement expected by chance. In simple terms, it can be described as the proportion of agreement between raters when chance agreement is removed from consideration.

7.3.2.1. Background and definitions

In calculating kappa, two quantities have to be taken into account: the observed agreement (p_o) and the agreement expected by chance (p_e). The former is given by the diagonal terms in the confusion table, while the latter is calculated from the off–diagonal terms.

The fundamental assumptions of kappa are as follows (as listed by Cohen, 1960):

- the cases present in the data are independent
- the categories the data is attributed to are independent, they mutually exclude each other and they are exhaustive
- the raters operate independently. There are no *a priori* defined criteria for the correctness or competence of the raters.

To briefly describe the way in which one arrives at the kappa measure, an example of a confusion table is shown below for the case in which there are $n=3$ categories and two raters.

Table 7.4. An example confusion table for three categories

	Rater 2			
Rater 1	**Category 1**	**Category 2**	**Category 3**	**Row total**
Category 1	p_{11}	p_{12}	p_{13}	$p_{1.}$
Category 2	p_{21}	p_{22}	p_{23}	$p_{2.}$
Category 3	p_{31}	p_{32}	p_{33}	$p_{3.}$
Column total	$p_{.1}$	$p_{.2}$	$p_{.3}$	1

In the case of such confusion tables constructed for the calculation of kappa, the proportions p_{ij} are calculated as number of cases per data set size. Following the definitions found in Cohen, 1960 and extended to $n > 2$ categories by Fleiss, 1981, the observed agreement can be calculated as the sum of diagonal terms in the confusion table:

$$p_o = \sum_{i=1}^{n} p_{ii} \tag{7.2.}$$

The agreement that is expected to be produced by chance is a sum of products of row and column totals of proportions:

$$p_e = \sum_{i=1}^{n} p_{i} p_{.i} \tag{7.3.}$$

The overall kappa statistic for multiple categories then results from the definition:

$$k = \frac{p_o - p_e}{1 - p_e} \tag{7.4.}$$

The difference between p_o and p_e represents the proportion of cases that can be attributed to beyond–chance agreement. The denominator represents the extent to which disagreement between raters is expected.

Kappa can take values between –1 and 1. If the agreement observed between raters is purely due to chance (i.e. p_o equals p_e), kappa is zero. If there is above chance agreement between raters, then kappa takes positive values. In practice, one is interested in high positive values of kappa, since this signifies very good agreement beyond chance between the considered raters. Landis & Koch, 1977 proposed a scale for kappa values. Usually, $k > 0.75$ is taken as very good agreement beyond chance, $k = 0.4 \ldots 0.75$ is considered fairly good agreement and kappas below 0.4 signify poor agreement beyond chance.

A significance level is also associated with kappa, in the form of a z ratio calculated by dividing kappa with its estimated standard error (Fleiss, 1981). Large z values indicate high statistical significance of kappa.

7.3.2.2. Using kappa in system performance evaluation

In the tests described in detail in the following chapters, kappa has been added to the evaluation process. Besides the mean classification accuracies calculated from the confusion tables, kappa and its significance level was calculated.

At first glance, an important benefit of using kappa in the description of categoriser performance (accuracy) is that it provides one a measure of performance without the need for specifying in each case of experimental setup the chance level (depending on the number of cases in each data category). In the case of non–equal group sizes, the chance level must be reported explicitly when listing mean classification accuracies in order to make assessment of the performance possible based on the diagonal terms of the confusion tables. Kappa provides a less ambiguous measure of accuracy, comparisons between test results can be made directly.

Also, the mean accuracies obtained from confusion table diagonal terms are a very coarse indica-

tor of performance in cases when data belonging to certain categories is significantly more correctly classified than those in other categories. Since kappa takes into account the non–diagonal terms of the confusion tables, it provides a more sophisticated measure of performance and based on the widely accepted scale of kappa values described previously, its meaning can be assessed independently from the characteristics of the data set.

One potential source of problems in the use of kappa in classification experiments is the presence of known categories and decisions taken by a single rater (the recognition system in this case). In the case of the Aberdeen data set, the category labels of each specimen were provided by expert taxonomists, hence kappa calculated from classification results of this data set can be considered a measure of agreement between human and computer judges. But in the case of the synthetic data sets, where object views are rendered to belong to known categories, one rater is replaced by the *a priori* defined category memberships of data items. Does this fundamentally affect the meaning of kappa?

Fleiss, 1981 is also preoccupied with this issue, arriving at the opinion according to which the use of kappa in the above described conditions *may* not be appropriate. Variants of kappa, like the measure proposed in Wackerly *et al.*, 1978 are addressing the problems of small data set. No definitive verdict on kappa's applicability in the above case was found in the literature. After all, the known categories can be viewed without significant mental effort as verdicts of a rater that happens to have 100% classification accuracy.

In these conditions, the kappa statistic was kept in the system's performance evaluation process. Keeping in mind the above described circumstances, its values can still be interpreted as indicators of beyond–chance classification accuracy, being more refined than the simple means calculated from confusion table diagonal terms.

7.3.3. Analysis of variance

The statistical method known as analysis of variance (or ANOVA for short) is a useful tool in investigating effects of certain factors on the data. In our case, the data comes from neural network trials in the form of classification accuracies, as it will become apparent in the following chapters. The factors that affect the data are modifications brought to the structure (e.g. size) of the neural networks used as categorisers, the configuration of coarse data channels that provide

the input to the categoriser etc. The effect of these factors can be reliably assessed with the help of ANOVA.

7.3.3.1. Background

Before describing the use of ANOVA in the system's testing, the most relevant theoretical aspects are highlighted in this section, a detailed discussion of the method and its several variants being done in Hays, 1988.

In the literature, the term 'factor level' means a particular manipulation of the subjects. In our case, the subjects are neural network–based classifiers initialised with random weights and trained/tested on feature data. A manipulation can be the change of the number of hidden nodes, in which case each value that this number can take is a level of the factor. As a ground rule, the factor must be an independent variable in the analysis. The experimental layout is that of randomised groups: individuals (i.e. randomly initialised neural networks) are picked randomly from the population (i.e. from the theoretical set of all possible neural network categorisers) and assigned to groups. Then each group is manipulated with a different level of the factor and the presence or the absence of an effect is investigated. An effect is considered to be present, if the data coming from the groups manipulated with different factor levels is significantly different.

When one determines the above with analysis of variance, it is crucial to respect the fundamental assumptions of ANOVA:

- the data obtained from all subject groups is assumed to have normal distribution
- the observation data coming from each group has the same variance
- the errors associated with any pair of observations (measurements) are independent (in other terms, the observations are independent)

In ANOVA, the null hypothesis is that the factor had no effect, i.e. the data measured in each group comes from the same population. In order to test this hypothesis, two variance estimates are calculated.

The within–groups variance shows how much the individuals in each group differ from the mean

of that group. This is estimated by and hence calculated as the within–groups mean square of cases:

$$MS_w = \frac{1}{N-G}\sum_{g=1}^{G}\sum_{i=1}^{n_g}(x_{gi} - \bar{x}_g)^2 \qquad (7.5.)$$

where G is the number of groups, N is the total number of cases in the data set, n_g is the size of group g. The group mean is $\bar{x}_g$ and x_{gi} is a case in group g.

The between–groups variance shows how much the means of the groups differ from each other. This is estimated by the between–groups mean square, defined below:

$$MS_b = \frac{1}{G-1}\sum_{g=1}^{G} n_g(\bar{x}_g - \bar{x})^2 \qquad (7.6.)$$

where $\bar{x}$ is the mean of all cases in the data set.

The more different the groups are, the larger the value of the between–groups variance will be. The null hypothesis then says that the two variances are equal. In order to reject the null hypothesis (i.e. to say that the factor had a significant effect on the individuals in the groups), the differences between the groups must be more pronounced than the differences between individuals within groups. As a measure, the F–ratio is used:

$$F = \frac{between-groups\quad variance}{within-groups\quad variance} = \frac{MS_b}{MS_w} \qquad (7.7.)$$

The value of F must be above 1 for an effect to be considered present, F providing a one–tailed test for the null hypothesis (Howell, 1982; Hays, 1988). The degrees of freedom in the experiment (depending on the number of groups and the number of subjects in each group) define how large the value of the F–ratio has to be for the presence of an effect to be noted.

When deciding whether to reject or accept the null hypothesis, the significance level α is taken into consideration. This is an *a priori* chosen probability of incorrectly rejecting the null hypothesis. In statistics, a widely used significance level is 5% (α=0.05). This is used in conjunction with the so–called p–value, that is supplied by the analysis. If this value, expressing the probability of observing the result is below the significance level, the null hypothesis can be rejected. In

the following chapters a 5% significance level is chosen when reporting the results of the analysis and the p–values will be quoted in association with the obtained F ratios.

The above described method is usually called one–way ANOVA, since the effect of only one factor is investigated.

7.3.3.2. Using ANOVA in tests

As it has been mentioned above, when studying the effect of certain factors on the system, the subjects in each experimental group are neural network–based categorisers. Each categoriser was initialised with random weights and was trained using the same learning parameters and method. The factors employed in studies were coarse data channel configuration, the size of hidden layer and training set.

In ANOVA terms, the situation that occurred during the tests corresponded with the so–called fixed model (Howell, 1982). In this experiment model, the levels of a given factor are fixed, these don't change randomly from one replication of the experiment to another. For example, when testing the effect of the number of hidden nodes on the system performance, the same values were used in several tests performed with different coarse channel data, so comparative studies could be made.

A number of training/testing runs being performed for each factor level and the accuracy of the categoriser in producing correct object labels being evaluated in each run as mean of confusion table diagonal terms, the initial assumptions of the analysis had to be checked.

For the normality assumption, one has to make sure that the sample size is large enough. The central limit theorem (Spiegel, 1992) states that as the sample size increases, the sampling distribution of the mean tends towards a normal distribution. So the number of subjects in each group (i.e. the number of randomly initialised networks trained/tested in each group) had to be statistically acceptable. Usually, although there is no agreement between statisticians on an exact minimum sample size, numbers around 20–30 are usually considered satisfactory (Hays, 1988).

In trials, 20 networks were trained/tested in each group. As Hays points out, when the group sizes are equal, departures from normal distribution of the data can be tolerated. The same point is supported by Howell, 1982 as well. Also, both authors agree that in the case of equal group sizes, the assumption of equal group variances can be violated. But it is pointed out, that one has to as-

sure that the observations are independent, both within and across groups, in order to satisfy the assumption of independent error components. This condition is evidently satisfied in the case of randomly initialised, independently trained/tested neural networks.

7.4. Summary

This chapter described the computer–generated and natural data sets that were used in the classification experiments, giving an insight to the considerations taken into account during the design of the 3D objects and rendering conditions. Also, it presented the problems that each data set is expected to pose to a recognition system, issues like shape similarity, foreshortening, self–occlusion, variable shapes and noise being discussed. The second main section of the chapter focused on the evaluation methods employed during these tests. A few general considerations regarding the nature of these tests and the used performance measures suitability were introduced. Since two rather sophisticated statistical tools, namely kappa statistic and analysis of variance were used during tests, these techniques were briefly described. Their theoretical foundations were outlined and emphasis was placed on the way in which these in particular testing situations can give useful informations on the system's behaviour. Having the data sets and statistical techniques described, the following chapters present the tests performed on the system and the obtained results.

Chapter 8. Preliminary tests on coarse data channels

8.1. Introduction

Having generated the sets of synthetic images, these provided the grounds for a number of preliminary tests that were meant to reveal properties of the coarse data channels under conditions of monitored viewpoint changes. These tests were designed to constitute a study into the ways in which variations in the data channels' functional parameters affect their ability of characterising shapes. A practical benefit of such tests was the arrival at a set of parameters that were accepted as close–to–optimal practical solution and used in all subsequent, more detailed analyses of the system. Also, these tests provided the means for verifying the assumptions on coarse data channel behaviour and discriminatory power of individual channels. The following sections describe these tests and report the results, including a discussion of these in the light of the theoretical issues and design choices detailed in previous chapters.

8.2. Testing the scale–space channel

The properties of the channel which employs theta histogramming and in its refined form, rho–theta receptive fields stayed in the focus of attention during the first tests performed on the system. As opposite to the discriminatory power of junction, spatial frequency and texture descriptors, mostly intuitive assumptions were available regarding the ability of the scale–space link trees and derived descriptors to characterise views of 3D objects.

8.2.1. Coarseness and discriminatory power

The theta histograms' discriminatory power has been studied on the two synthetic data sets, in conditions of variable number of histogram bins per link tree layer. It was expected that with an increase of the number of bins, the representation of shapes becomes so precise, that it affects the system's performance when large variations of viewpoint occur in the input.

Discriminant analysis was used as categoriser and the experimental setup was tuned to the characteristics of each data set. In the case of the 8–object set, due to the small number of views available for each object (a number of 15), leave–one–out classification had to be used. As it was

pointed out in the previous chapter, this provides a more accurate estimate of performance than the resubstitution method, in the case of small data sets. The number of theta histogram bins was varied from 8 to 40 bins per link tree layer. Three layers of link tree being used, the upper limit for the number of bins was defined by the number of cases in the data set (120), since the number of variables in DA must be less or equal than the number of cases for the matrix algebra to make sense.

The mean leave–one–out classification accuracy for the 8 objects is represented in Fig. 8.1., together with the lowest and highest accuracies. The chance–level classification accuracy in the case of this data set is 12.5%.

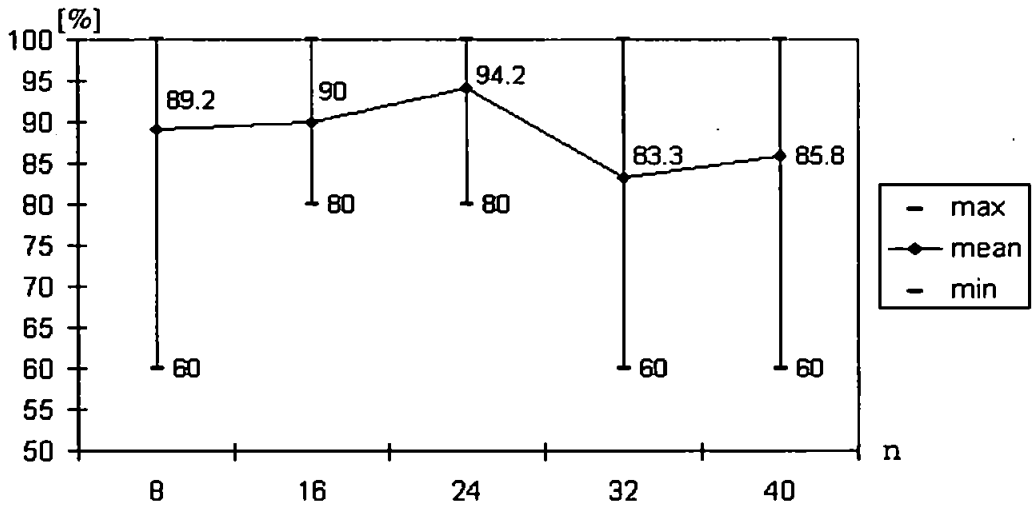


Fig. 8.1. Discriminant analysis (leave–one–out) classification accuracies for the 8–object data set, theta histograms with various number of bins per link tree layer (n).

The lowest classification accuracies reported in the above figure were obtained for objects No. 7 and 8, while the best discrimination was achieved for objects No. 1 and 5. Figure 8.2. below illustrates the clusters that the feature vectors form in discriminant space (i.e. projected onto the plane defined by the first two discriminant functions).

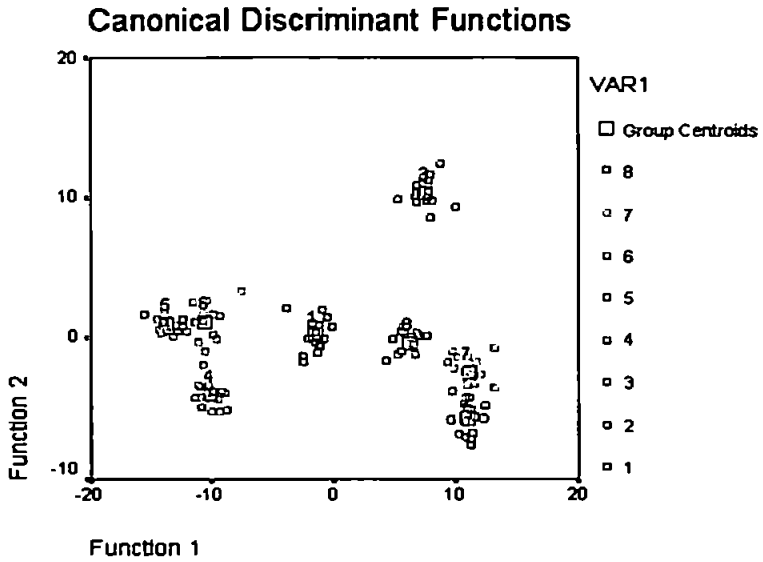


Fig. 8.2. Scatter plot of the feature vectors for the 8 objects

All 8 objects' views lead to theta histograms that present highly satisfactory separation in discriminant space, this being an indication of the ability of this channel to characterise shapes and to compensate for small changes in viewpoint.

Before drawing conclusions based on the performance plots, the classification accuracy was assessed on the 5–object data set, too. This made possible a better testing of the generalisation ability of the categoriser, since it allowed the use of large model and test data sets and wide variations in viewpoint. The set of computer–generated views was split in two equal parts for this test, each sub–set having 180 views per object. With the convention described in section 7.2.3., every second image in the view sequence ended up in the test set. The accuracies reported by DA on the test data set are plotted in Fig. 8.3. below. In the case of this data set, the chance–level classification accuracy is 20%.

The most accurately identified object was No. 3 and 5, and the lowest classification accuracies were obtained for objects No. 1 and 2. The highest confusion between categories was observed for these two objects (26.1%). From Table 7.3. (p. 132) it becomes apparent, that object No. 5 was judged by humans, too as being the most dissimilar one and objects 1 and 2 were judged as being the most similar.

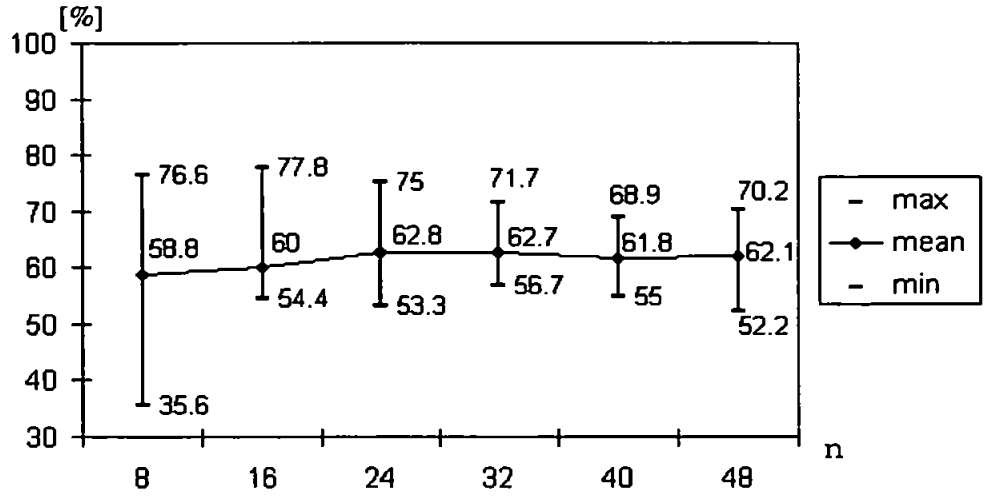


Fig. 8.3. DA classification accuracies for the 5–object test data set, theta histograms with various number of bins per link tree layer (n).

In both diagrams (shown in figures 8.1. and 8.3.), it is apparent that a significant increase in the number of theta histogram bins does not bring considerable improvement to the classification accuracy. The tests on the 8–object data set shows that the best mean accuracies are obtained for 16–24 histogram bins. The 5–object set trials also show, that significant departure from the performance obtained for 16, 24 bins does not appear for larger histograms. A relevant aspect of the performance plots is the pattern of variation of the lowest classification accuracies reported in each trial. For small (<16) number of histogram bins per link tree layer, the lowest observed accuracies are inferior to those obtained for mid–range histogram sizes. Also, for large number of bins (≥ 32), the lowest classification accuracies decay. The most strongly confused objects that correspond to these accuracies seem to be poorly discriminated when too coarse or too fine representation is used. Also, the highest classification accuracies in the tests involving the 5–object data set were reported for the same mid–range histogram sizes, as in the 8 object set's case.

Taking into consideration the above aspects, together with the computational load during neural network simulations and the complexity of the coarse feature representation, a mid–range value for the number of theta histogram bins was chosen. Further tests involving theta histograms employed 16 bins per link tree layer, therefore the resolution of the histogram representation resulted as 22.5 degrees. The dimensionality of the theta histogram feature vector at the output of the data channel was 48.

When adding link vector norm (ϱ) information to the scale–space representation by using rho–theta receptive fields, the resulting extended data channel's behaviour had to be tested in similar

circumstances. As it is described in the following section, the relationships between the coarseness of the rho–theta representation and the channel's discriminatory power have been investigated.

8.2.2. Parameters of rho–theta receptive field grids

The 5–object data set has been used to test the performance of the extended scale–space representation. It allowed the use of large training and test data sets, hence it made possible classification trials in conditions of high dimensionality of the computed feature vectors (unlike the small 8–object data set).

As in previous tests that used theta histograms, the 5–object data set was split into model and test sets of equal size using the splitting method defined in section 7.2.3. The tests have been performed for various receptive field grid sizes and receptive field widths. The former directly affects the coarseness of the representation, while the latter defines the degree of overlap between receptive fields placed on rho–theta planes.

In order to be able to directly compare results of tests that involve 16–bin theta histograms on each link tree layer with those employing receptive field grids, the size of the grid was set initially to 4x4 receptive fields. In further tests, this size was increased to 8x8. The standard deviation of the Gaussian receptive fields was varied to span a range of values that took into account the density of receptive fields on the polar planes.

The mean DA classification accuracies obtained for the test set are represented below in Fig. 8.4. for both grid sizes, together with the lowest and highest accuracies achieved in each test. Chance level accuracy is 20%. Again, the lowest classification accuracies were obtained for objects No. 1 and 2, the most well discriminated object being No. 5.

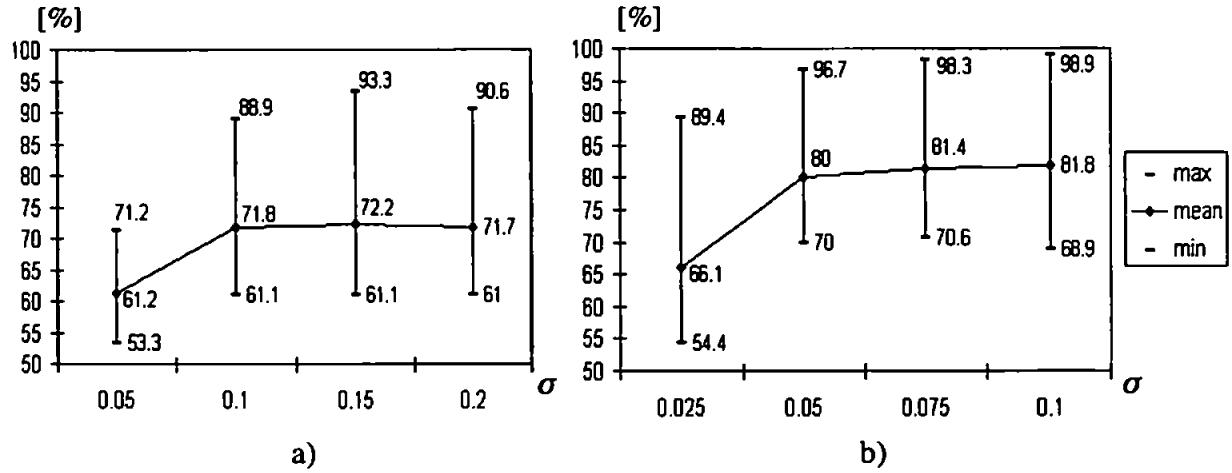


Fig. 8.4. DA classification accuracies for rho–theta receptive field activation patterns (test set). a) 4x4 grid on each link tree layer b) 8x8 grid on each link tree layer

It is apparent, that as the overlap between receptive fields increases (as σ acquires larger values), the performance registers an increase as well. Standard deviations above 0.1 (for the 4x4 grid) or 0.05 (in the case of the 8x8 grid) do not produce significant improvements in classification accuracy. Also, an increase in the number of receptive fields has a positive effect on the performance, but evidently the dimensionality of the feature vectors sees a strong increase. For an 8x8 receptive field grid placed on each link tree layer, 192–dimensional feature vectors result, instead of 48 in the case of a 4x4 grid. Such an increase in the dimensionality of the feature vectors leads directly to very significant increase in simulation time during tests involving neural network–based categorisers, with moderate gain in classification accuracy.

For subsequent experiments, as a compromise, the size of the grid was chosen to be 4x4 and the standard deviation of the Gaussians was set to $\sigma = 0.15$. This choice for the size of the receptive field grid made possible direct comparisons with the performances achieved by the system when using a 48–dimensional theta histogram vector, since the dimensionality of the input is kept the same. In the case of the rho–theta representation, these feature vectors encode in the same number of variables not only the θ, but also the ϱ information.

8.3. Coarse data channels' discriminatory power

Having tested the scale–space channel's ability of characterising the input shapes in its both forms (theta histograms and receptive field grid activation patterns), the subsequent preliminary performance trials were meant to test the other coarse data channels, too and make comparisons be-

tween channels' discriminatory power in monitored test conditions (e.g. viewpoint changes).

8.3.1. Regions of the viewing hemisphere

Using all images of the 5–object data set, DA tests were performed in order to assess the clustering of data vectors provided by each coarse channel when the analysed sets of views capture essentially different aspects of the 5 objects.

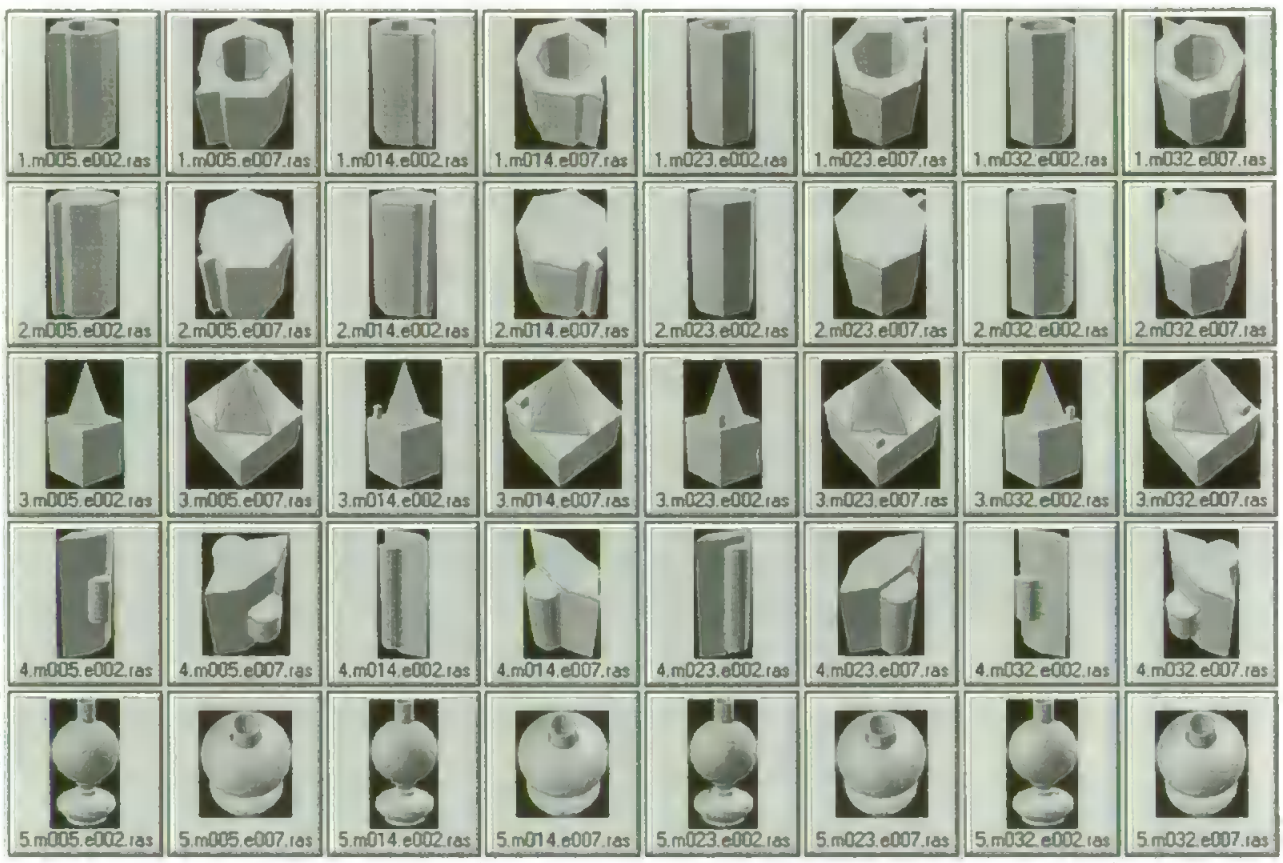


Fig. 8.5. The 5 objects viewed from middle of 8 viewpoint regions.

In order to be able to perform tests on views collected from well–defined ranges of viewpoint, the set of images has been split into 8 sub–sets according to the 8 regions of the upper viewing hemisphere. Each region was chosen to have a 90 degree azimuth and 45 degree elevation angle span. In subsequent descriptions, the 8 equal regions of the viewing hemisphere will be referred to as M.N, where M denotes the 90 degrees azimuth interval (I...IV) and N denotes the lower/upper 45 elevation angle interval (I,II).

The 5 objects viewed from all 8 regions of the viewing hemisphere are shown above in Fig. 8.5. In order to illustrate the aspects related to the visibility of certain elements of the objects, the changes in their shape due to foreshortening, the alteration of the apparent size due to the changes in distance from the visible object elements to the camera in close–to–top views, this figure shows the objects from viewpoints that are situated in the middle of each of the 8 regions. As a detail that could prove to be useful in the interpretation of the images, the image labels contain object number, the azimuth and the elevation angle codes. A label has the syntax *O*.**m***azimuth*.**e***elevation*.**ras**, where *O* is the object number, the camera's azimuth angle is (*azimuth*–1)*10 degrees and the elevation angle is *elevation**10 degrees.

Some of these objects (No. 1 and 2) have distinctive features visible only from certain viewpoints (the cavity in object No. 1 is visible only from viewpoints with higher elevation). Also, the asymmetric elements in these two objects' structure are fully visible only from a limited range of azimuth angles (first and especially the second quadrant, i.e. azimuth angles between 0 and 180 degrees). Object No. 3 shows its asymmetric detail only when viewed from a quadrant other than the first. The shape of object No. 4 changes in lateral/frontal views (quadrants I and III). Besides the self–occlusion of certain elements of the objects, an important phenomenon is also the change in the objects' apparent size. Since all have shapes that extend along the vertical axis, when the camera moves to higher elevation angles, parts of the objects become closer to the camera than the viewing target situated in the object centroid (coordinates 0,0,0 as listed in Table 7.2., p. 130). This has as an effect the increase in the apparent diameter of the objects, as it can be seen from the higher elevation angle views in Fig. 8.5. Since the scale–space representations used by the scale–space channel are affected by large variations in scale, it is interesting to see how performance changes in regions x.II.

Each set of views corresponding to a particular region of the viewing hemisphere contains 45 views per object. On these data sets, DA with resubstitution and leave–one–out classification was performed. Since the number of variables (48 for the scale–space channel, 30 for the other channels that use a self–organising map) is close to the number of data items per category, the leave–one–out method provides a more correct estimate of the accuracy than the resubstitution method.

8.3.2. Tests on individual channels

The views collected from the 8 regions of the viewing hemisphere were submitted to feature ex-

traction and coarse coding. The feature vectors provided by each data channel were used as input in DA trials. In all tests, chance level is 20% (five groups of equal size in the data).

The scale space channel employing theta histogramming with 16 bins per link tree layer produced the classification accuracies listed in Table 8.1. below. Since the number of variables (48) is slightly larger than the number of data items per category in each M.N region data, the resubstitution results must be interpreted with caution. Still, it gives an idea on the ability of the categoriser to build an accurate model based on this feature data. For comparisons between channels and regions, the leave–one–out classification results will be taken as basis. Also, when reporting the lowest and highest classification accuracies in subsequent tests, we will be referring to leave–one–out classification.

Table 8.1. Mean classification accuracies for 8 regions of viewing hemisphere (%) – theta histograms; 45 items per object per region

Region	I.I	I.II	II.I	II.II	III.I	III.II	IV.I	IV.II
Resubstitution	88.0	84.4	88.9	85.3	88.9	88.4	88.9	90.2
Leave–one–out	72.4	72.9	75.1	72.4	75.6	75.1	74.2	75.1

The overall lowest classification accuracy was obtained for object No. 1 (46.7%), the highest being obtained for object 5 (100%). In a similar manner, the scale–space channel employing receptive field grids was tested, the results being listed in Table 8.2. Based on the preliminary tests described in section 8.2.2., a 4x4 grid of receptive fields was used on each link tree layer and the standard deviation of the Gaussians was set to 0.15.

Table 8.2. Mean classification accuracies for 8 regions of viewing hemisphere (%) – rho–theta receptive field grid; 45 items per object per region

Region	I.I	I.II	II.I	II.II	III.I	III.II	IV.I	IV.II
Resubstitution	95.1	89.9	94.7	89.3	96.9	90.2	94.2	92.4
Leave–one–out	80.9	76.0	85.8	76.0	86.2	80.0	84.9	81.3

Object No. 1 was the most poorly discriminated (51.1%) and the highest classification accuracy resulted for object No. 5 (100%). There is a 4–10% gain in leave–one–out classification accuracies compared to the theta histogram method.

The junction channel has been tested in similar conditions, the self–organising map being trained on each region data sub–set. An initial learning rate of 1.0 was used, which decayed during the training. The number of epochs was set to 1000, since no noticeable changes in the vectors associated with each node could be observed with further training. These learning parameters were used in all subsequent trainings on 8–region data of self–organising maps. The node activation patterns resulted from propagating the data items through the map were used as input in the 8 DA trials. The set of results is listed below in Table 8.3.

Table 8.3. Mean classification accuracies for 8 regions of viewing hemisphere (%) – junction histogram channel; 45 items per object per region

Region	I.I	I.II	II.I	II.II	III.I	III.II	IV.I	IV.II
Resubstitution	89.3	84.0	90.7	86.2	89.3	80.4	91.1	90.2
Leave–one–out	80.4	69.3	84.4	75.1	81.3	70.2	80.9	79.6

The obtained lowest and highest classification accuracies were 48.9% and 100%, for objects No. 2 and 5, respectively.

The spatial frequency channel has been tested in a similar way, the self–organising map activation patterns being submitted to the DA categoriser. The results are reported in Table 8.4.; object 2 was the most poorly discriminated (with the lowest accuracy of 37.8%), and again object 5 was identified with 100% accuracy.

Table 8.4. Mean classification accuracies for 8 regions of viewing hemisphere (%) – spatial frequency channel; 45 items per object per region

Region	I.I	I.II	II.I	II.II	III.I	III.II	IV.I	IV.II
Resubstitution	82.2	79.1	80.4	77.3	76.4	76.9	76.0	73.8
Leave–one–out	70.7	65.8	67.6	66.2	64.0	64.4	57.3	59.1

The texture channel's Kohonen map node activation patterns, submitted to DA classification led to the following mean classification accuracies in the 8 regions:

Table 8.5. Mean classification accuracies for 8 regions of viewing hemisphere (%) – texture channel; 45 items per object per region								
Region	**I.I**	**I.II**	**II.I**	**II.II**	**III.I**	**III.II**	**IV.I**	**IV.II**
Resubstitution	77.3	78.2	72.9	82.2	65.3	76.0	68.0	80.9
Leave–one–out	63.6	63.6	59.6	64.4	48.4	62.2	47.6	66.7

The lowest and highest accuracies were obtained for object No. 1 and 5 (28.9% and 100%, respectively).

For comparison, the mean leave–one–out classification accuracies obtained in the 8 regions for each of the coarse data channels are summarised in Fig. 8.6.

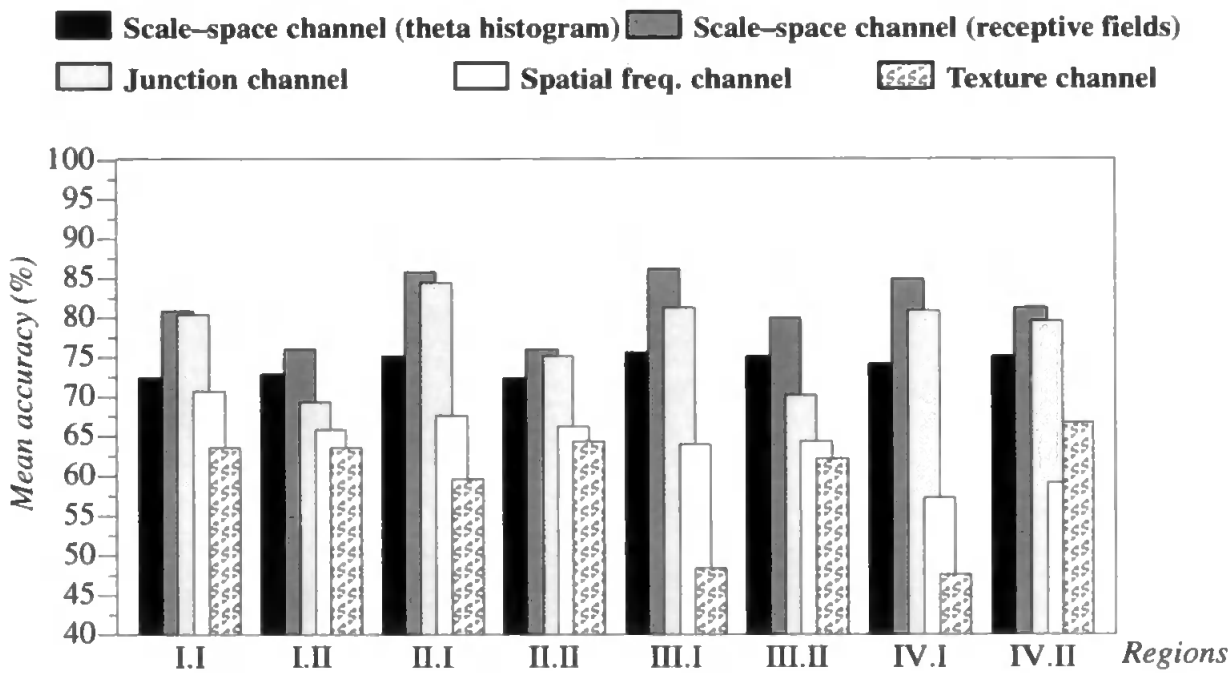


Fig. 8.6. Mean DA leave–one–out classification accuracies for all individual data channels in 8 regions of the viewing hemisphere.

Due to the aspects described in section 8.3.1. (related to the visibility of the discriminatory features of objects No. 1 and 2), the confusion between these two objects has been studied in various regions of the viewing hemisphere. The evaluation of the confusion between these objects led to a number of comparisons between coarse data channels, as it will be discussed in the following section. An investigation of all inter–object confusions in the case of joint channels and the use of the whole data set will be reported in the next chapter, once the extensive neural network–based classification trials are discussed.

Using the symmetric non–diagonal terms of the DA leave–one–out classifications' confusion tables, the confusion between objects 1 and 2 was calculated as the sum of these terms. For all of the used data channels, the resulted confusion values are listed in Table 8.6.

Table 8.6. Confusion (%) between objects No. 1 and 2 for all data channels and regions

Region	I.I	I.II	II.I	II.II	III.I	III.II	IV.I	IV.II
Theta	20.0	8.8	37.8	15.5	48.9	17.8	48.9	37.8
RF	26.7	8.8	22.2	8.9	40.0	8.9	37.7	8.9
Junction	53.3	4.4	46.7	22.2	44.4	4.4	53.3	11.1
Spatial frequency	40.0	20.0	93.3	51.1	64.5	33.3	88.9	48.9
Texture	33.3	68.9	48.8	57.8	64.4	68.9	77.7	51.1

As in the case of mean classification accuracies, the confusion scores show significant dependency on the regions, as it would be expected based on the design of these objects.

The conclusions that can be drawn based on these performance plots and classification accuracies obtained during the above presented tests are described in the next section.

8.3.3. Discussion

The DA trials performed on the individual data channels' feature data has shown that each channel can characterise the synthetic shapes presented to them in a way that leads to satisfactory classification accuracies.

Regarding the 'where' channel which is the scale–space channel, from the results reported in the previous section and Fig. 8.6. it becomes apparent, that joint encoding of scale–space link orientations and norms has a positive effect on classification, compared to the accuracies obtained from the theta histogram data. The receptive field (RF)–based approach evidently yields a stricter shape description than the theta histogramming, by capturing additional information on the shape geometry. This makes it more sensitive to phenomena that significantly affect the objects' visible shapes. It can be seen from Table 8.1. and 8.2., that the DA based on RF activation patterns is more affected by strong changes in shape occurring close to top views. As Fig. 8.6. makes it visible, the mean classification accuracies in regions x.I and x.II show a consistent pattern of decay in all quadrants, those registered in regions x.II being on average 8% lower than those in regions

x.I. In the case of the theta histograms, this decay does not occur consistently in all four quadrants and it is only about 2% in magnitude.

A possible factor that plays a role in this decrease in overall performance is the strong shrinking of object vertices in the upper regions x.II. It seems that with the overall shape of an object becoming unavailable to the feature extraction module, the objects are less accurately described from close–to–top viewpoints. This decay of performance has been observed in humans, too when presented with object views that had their principal axis foreshortened (Marr, 1982). But the effect of the change of apparent object size (diameter) also has to be taken into account in regions x.II. The theta histograms do not register link norm information, hence they are affected only by large scale changes that produce significant alterations of the number of maxima nodes in the scale–space trees. The receptive field–based coarse coding, though is expected to be affected by less radical changes in link trees (e.g. in conditions when the distances in scale–space between maxima nodes are altered, without the number of maxima being changed by scale variations).

The differences between mean accuracies obtained in regions x.I and x.II do not present consistent patterns for the 'what' channels like the texture and spatial frequency channel. These, by not registering informations on overall shape geometry, provide feature descriptions that seem to be less affected by foreshortening. This does not happen, though in the case of the junction channel. Although it was designed to be another 'what' channel and not to register feature position information, it is more affected by self–occlusion of object surfaces. An explanation lies in the information encoded by this channel. As it has been mentioned above, due to the characteristics of the data set, the fundamental differences between objects are those related to their geometry. With the strong foreshortening and even disappearance of several longitudinal surfaces of the objects when viewed from close–to–top camera positions, the descriptions of the junctions that emerge from the way in which object surfaces meet are strongly affected. This would explain the significant (up to 11%) decay in classification accuracy between regions x.I and x.II, registered in the junction channel tests.

Regarding the confusion between objects No. 1 and 2 (scores listed in Table 8.6.), a number of conclusions can be drawn about the behaviour of the coarse data channels. All but the texture channel leads to significant decrease of the confusion between the mentioned objects in regions x.II, compared to the confusion scores observed in regions x.I. As a parallel with human perception of these shapes, it seems that with the most pronounced difference between the two objects

(i.e. the presence of a cavity in one of them) becoming fully visible in regions x.II, most of the data channels register this difference. The texture channel results show an opposite trend, the confusion between the two objects increasing as the elevation angle of the camera moves into the upper 45 degrees. A possible explanation would be the fact that there is no significant connection between the geometric properties of the objects' elements and the object characteristics registered by the texture channel. The inconsistencies in the variation of mean accuracies in regions x.I and x.II for this channel (described above) and in the trend mentioned here lead one to the conclusion that no parallel can be drawn between the salience of geometric elements and the classification performance based on the texture channel. Although the other 'what' channels record features that are mostly due to object and surface geometry, the texture channel does not capture such characteristics of object shapes.

It can be seen from Table 8.6. that when the geometry of the asymmetric element in the structure of the two objects is not sufficiently visible (regions III.I, IV.II), confusion between these objects increases compared to the scores observed in other quadrants' x.I regions. The junction channel is an exception, similar confusion scores being observed in all x.I regions. It seems that in these regions it does not register sufficient details of the distinctive elements of the two objects, hence the confusion between the objects varies only about 9% in regions x.I. The spatial frequency channel, too seems to lead to very large confusion between the objects No. 1 and 2 in region II.I, where the asymmetric elements are fully visible. It seems that the channel fails to record the small details of the asymmetric object elements (these being the most salient features for distinguishing the two objects from these viewpoints), the overall shape properties being more prominently registered.

A visual illustration of the clustering of these shape–specific feature vectors can be obtained by producing a scatterplot of these vectors in the space defined by discriminant functions. As an example of the trends observed in the behaviour of both theta histogram and rho–theta receptive field–based feature vectors, the latter are included here. In Fig. 8.7. shown below, the rho–theta receptive field activation patterns' projections onto the plane defined by the first two discriminant functions are represented as scatterplot. Since the first two discriminant functions account for most of the variance of the data, these plots can be taken as indications of the feature vectors' clustering. Regions II.x were chosen for this representation, since in quadrant II of the viewing

hemisphere all asymmetric details of objects 1, 2 and 3 are visible. The group centroid labels were enhanced for visibility.

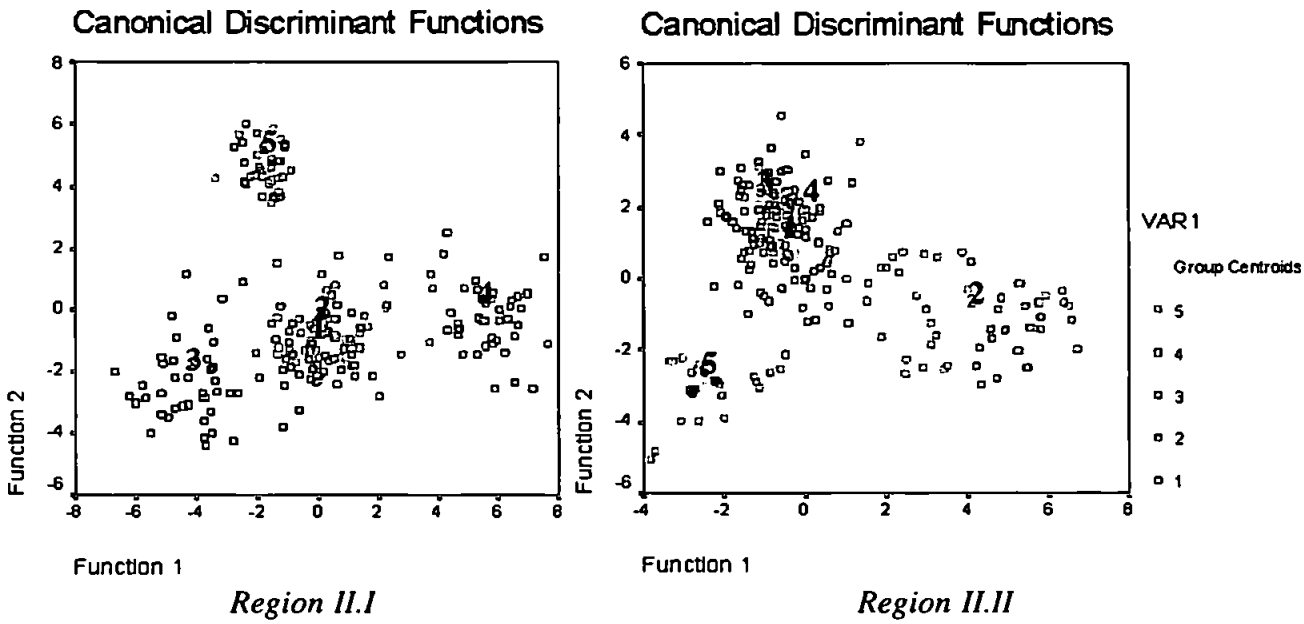


Fig. 8.7. DA scatter plots for rho–theta receptive field activation patterns in quadrant II of the viewing hemisphere.

It is apparent, that feature vectors describing the shape of objects 1 and 2 cluster together in region II.I, while groups corresponding to objects 3,4 and 5 are reasonably well separated. In region II.II, where the main difference between objects 1 and 2 become prominent, group 2 separates well from group 1, also the pronounced difference in shape between object 5 and the rest is still well represented. It is noticeable, that with the changes in apparent object size and foreshortening occurring in this upper region, the feature vectors spread out in discriminant space. The same spread was observed for all channels in regions x.II.

As an example of typical clustering of feature vectors that are generated by the 'what' channels, Fig. 8.8. below shows the scatterplot in discriminant space of the spatial frequency channel outputs. The group centroid labels were enhanced for visibility.

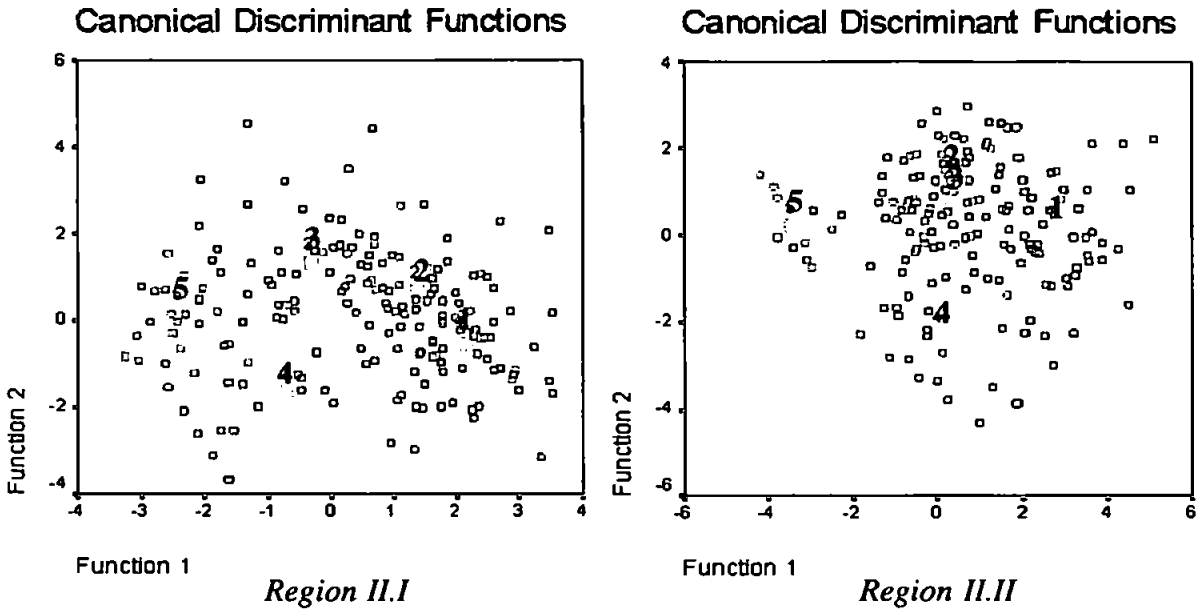


Fig. 8.8. DA scatterplots of spatial frequency feature vectors for quadrant II.

The clusters observed in the case of these data channels' feature data are considerably spread out in comparison with the clusters that the shape descriptor scale–space channel vectors produce. The better separation of groups 1 and 2 in regions x.II is also exemplified by the above figure. As the scatterplots show, the feature vectors describing object 5 are in all cases well–separated from the other clusters.

Having generated the views in the 5–object data set from synthetic objects of the same surface texture, it is hardly surprising that the texture channel has the lowest discriminatory power in classification trials. The same goes for the spatial–frequency channel, since the surfaces of the objects in this data set are rather 'eventless' due to the use of uniform synthetic surface textures. The only features that can produce distinctive spatial frequency signatures are illumination contrasts between surfaces. The main differences between the objects are those defined by their geometry, i.e. the number of surfaces and the way in which they combine. The latter characteristics are described by the junction channel, which in a way that was expected, yielded significantly higher classification accuracies than the texture and FFT channels. The most salient characteristics of the test objects therefore seem to be captured by the scale–space and junction channels. For other test data, where for instance objects differ significantly by their surface textures, the discriminatory power of the channels would be expected to change accordingly.

8.4. Summary

The above described preliminary tests carried out on computer–generated data sets revealed several aspects of the coarse data channels' behaviour in various conditions. The classification trials that employed a statistical classifier, theta histograms and later rho–theta receptive fields made possible the choice of values for key functional parameters of the scale–space channel. These values used later in all subsequent tests were chosen based on test results as trade–offs between performance, practical feasibility, computational load.

Tests that followed were meant to reveal possible patterns in classifier performance when it is presented with data provided by individual data channels, in conditions of monitored viewpoint changes. These trials led to a series of conclusions regarding the way in which each channel characterises the input shapes, these conclusions being in support of the theoretical assumptions behind each data channel's design. It was found that the 'where' channels play the main role in describing the geometry of shapes, as it was intended, the 'what' channels adding general information on object and surface properties.

Extensive tests carried out on the system, involving multiple coarse data channels, training and test data sets of variable size, assessments of effects of a number of factors on the performance and others are described in the following chapter.

Chapter 9. Classification of synthetic shapes

9.1. Introduction

Based on the preliminary tests described in the previous chapter, a number of structural parameters were chosen for subsequent experiments. Having tested the properties and the behaviour of the coarse data channels in conditions of relatively small variations in viewpoint, the following experiments were meant to face the system with a more realistic situation: large variations of viewpoint, hundreds of training and unknown views of a limited number of objects. The present chapter describes the tests and their results when the computer–generated 5–object data set was used as input. Collective and committee machines were tested, their performance under various conditions revealed a number of properties of the system. Beyond the practicalities of assessing the classification accuracies of the various classifiers presented with coarse feature data, as a piece of research, these tests allowed the study of several aspects that were investigated in the past in work carried out on 2D shape classification. Issues like the contribution of various data channels to the system's performance, superiority of collaborating channels over individuals, collectives over committees were studied in depth. These investigations made possible the generalisation of findings from the 2D shape recognition work to the sphere of 3D shape classification.

9.2. Experimental protocol

As it has been mentioned before, the experiments described in the following sections were based on different training and test set sizes, variable coarse data channel configurations and neural network sizes. This section describes the objectives of the tests, these objectives defining the way in which the data sets were prepared and trials were set up.

9.2.1. Objectives

In contrast with the preliminary tests described in the previous chapter, the recognition experiments involving different coarse data channel configurations had the aim of testing the system in more realistic conditions. In a practical situation, one expects the system to learn a set of views

and generalise based on these to unknown views that represent the objects from viewpoints that span the whole possible range of camera positions. Therefore, in the case of the 5–object synthetic data set, the training and test images represented the objects from viewpoints that spanned the whole viewing hemisphere. These experiments tested the generalisation ability of the system operating with various groups of data channels, in conditions of wide variations of viewpoint.

The size of the training set was varied in experiments, the observed performance of the system in each case showing its ability to generalise based on variable amount of learnt data. Also, by modifying the size of the neural networks (i.e. the number of neurons in the hidden layer), the classifier's ability to fit a model on the training data was affected. In these experiments, the effect of the change in network size on the performance was observed. The practical significance of these trials lies in the fact that no universal rule exists for the choice of network size in relation to the amount and dimensionality of training data. With several trials, the best choice can be made for a particular data set, and tendencies in the system's behaviour can be studied.

Collective machines based on statistical methods (DA) and neural networks were used in tests that involved various coarse data channel configurations. For comparative studies, neural networks were also trained and tested on individual channels' data. The output layer activation patterns were saved in these tests, these stored patterns being used later as input to the committee machines' decision making process. This made direct comparisons between collectives and committees possible.

The committee machine architectures detailed in section 6.4.5. (p. 123) were used in the cases where four coarse data channels were gradually introduced into the system's structure. This made possible the assessment of the contribution of each data channel to a correct decision of the classifier. Due to the properties of the 5 objects and the 3D scene (identical surface texture, rigid geometrical shapes, similar lighting conditions), it was expected that the 'where' scale–space channel and the 'what' junction channel have a more important role in helping the system to categorise the views than the spatial frequency and texture channels have.

The performance obtained by statistical and ANN–based collectives, in comparison with committees was studied. This made possible a number of conclusions to be drawn, regarding cooperation between channels and allowed generalisation from the studies done in the field of 2D shape recognition (Culverhouse *et al.*, 1996; Ellis *et al.*, 1997).

9.2.2. Data preparation

When selecting the data for training and test sets used in classification experiments, the goal was to provide the classifier with good representation of all categories. Also, the categories had to be equally well represented, so that learning would not be biased towards the category that is present with more specimens in the training set. The training views of the 5 objects were chosen in such a way that the corresponding viewpoints spanned the viewing hemisphere.

The available synthetic data set was split into training and test sets according to the method described in section 7.2.3. (p. 130). As it was mentioned therein, this splitting method made possible the choice of training and test views with uniform coverage of the viewing hemisphere and control over the viewpoints. This constituted a step forward in generalising the test conditions, in comparison to the viewpoint region–based preliminary tests. Five different split ratios were used in tests. Three of these will be referred to in the form m:1, where m=1,2,3. According to the adopted convention, the views with multiple–of–(m+1) indexes in the view sequence were placed in the test data set, the rest of the views being inserted into the training set. Therefore, with the above choices for m, the whole 5–object image set was split into training and test sets in 1/2:1/2, 2/3:1/3, 3/4:1/4 proportions.

In addition to this, the split ratios 1:2, 1:3 were used in order to generate training sets that are smaller than the test sets. Hence the system has to generalise to significantly more views than the ones available in the training data. The data sets for such an 1:m split ratio was obtained by swapping the training and test sets obtained from an m:1 split. The gap between two consecutive training views is (m+1)*10 degrees elevation if both viewpoints have the same azimuth. If the views are located on different arcs, there is a 10 degree azimuth angle difference and a difference in elevation between 0 and (m–1)*10 degrees.

The size of the training and test sets for each split ratio is listed in Table 9.1. below.

Table 9.1. Size of training and test sets for all split ratios of the 5–object data set

Split ratio	Nr. of training views/object	Nr. of test views/object
1:3	90	270
1:2	120	240
1:1	180	180
2:1	240	120
3:1	270	90

When testing the effect of changes in the configuration of the coarse data channels, the scale–space channel with its essential role of describing spatial distribution of features was considered to be the kernel of the feature representation. Therefore, besides the tests on individual channels, the performance of the system was evaluated in conditions where the other ('what') data channels were associated with this 'where' channel. Therefore in all subsequent descriptions, when referring to tests performed on 2,3 and 4 channels, the junction, spatial frequency and texture channels (in this order) are associated with the scale–space channel.

9.2.3. Training and testing neural network–based classifiers

The number of hidden nodes in neural network trials was varied in a range that kept the simulation time at an acceptable level. Since in the case of associated channels, for less than 4 hidden nodes no satisfactory convergence was observed during training, the starting value for the number of hidden nodes was chosen to be 4.

The above choices for minimum number of hidden nodes made possible satisfactory training of the network in all cases of channel configurations (i.e. all input vector sizes). The number of the hidden nodes was increased to 8 in steps of two. For more than 8 hidden nodes, the simulation times increased beyond practical limits. As an example, a single set of 20 training/testing runs using the SNNS simulator and 4 channels, 8 hidden nodes in a collective machine took 4 days to run on a SUN Enterprise 3000 platform. Considering that each experiment involved multiple sets of runs for different channel configurations, training set sizes etc., due to this simulation time further increase in the network size was not planned. Conclusions were drawn based on the tendencies observed during tests that involved these limited numbers of hidden nodes. The main purpose of these tests that involved changes in network size was to show how a network of reasonable size (configured based on a few very fast preliminary trials) can achieve good performance.

Since three of the coarse data channels incorporate self–organising maps that need to be trained on feature data prior to presenting their node activation patterns to the classifier module, these maps were trained and tested with the established data sets. The number of epochs and learning parameters were chosen to be equal to the ones listed in the previous chapter (p. 153). Therefore the learning rate was set to an initial value of 1.0, which decayed during training (using the adaptive procedure built into the MATLAB language). The size of the maps was kept at 5x6 nodes.

It was found that 1000 epochs were sufficient to arrive at a map which did not present noticeable weight changes during the last epochs of the training. Few training/test runs were performed (a number of 3), since the maps tended to arrive at nearly identical weight configurations at the end of the runs. Indeed, when the node activations of each map trained in separate runs were presented to a DA classifier, the mean accuracies obtained on test data differed by less than 3%. In the case of each data channel that incorporated a SOM, the trained map that led to the best results was kept in the experiment and used in trials involving other data channels.

For each particular set of structural/experimental parameters (i.e. data set size, number of hidden nodes, data channel configuration), a number of 20 neural network training/test trials were performed. This allowed statistical evaluation of the data by supplying a sufficiently high sample size to methods that rely on assumptions about the normality of the data (as it was discussed in section 7.3.3.2., p. 142). Also, this number of runs was statistically sufficient for drawing conclusions on the system's mean performance.

At the beginning of each trial, the neural networks were initialised with random weights. During each trial, the training data has been presented to the network in random order. The learning parameters were decided during preliminary network simulations. Such simulations made possible the choice of learning parameters (i.e. learning rate, momentum and number of epochs) that led to good network learning in all cases of channel configuration and training set sizes. Once arrived at such a set of values, it was used in all subsequent trials. The values of these parameters will be listed in the subsequent sections.

The evaluation of the results has been carried out with the help of MATLAB scripts that organised the large amount of resulted confusion tables and network output activation patterns, performed the calculation of kappa statistics, means and one–way ANOVAs.

9.3. Tests with collective machines

The system's performance has been tested on the 5–object computer–generated data set, using collective machines. As classifiers, discriminant analysis and feedforward neural networks were used. These tests allowed the assessment of the system's recognition accuracy in the absence of image noise and in conditions of controlled shape geometry, surface properties and lighting.

9.3.1. Discriminant analysis trials

For all channel configurations, training and test set sizes (i.e. data set split ratios), the data has been fed into a discriminant analysis–based classifier. When testing the system with an ensemble of coarse data channels, the 'what' channels' data was concatenated with the feature vectors supplied by the scale–space channel. Initially, theta histograms were used as scale–space coarse coding method.

The mean classification accuracies on the 5 object categories, for all considered channel configurations and training/test set sizes are represented in Fig. 9.1. Chance level is 20%.

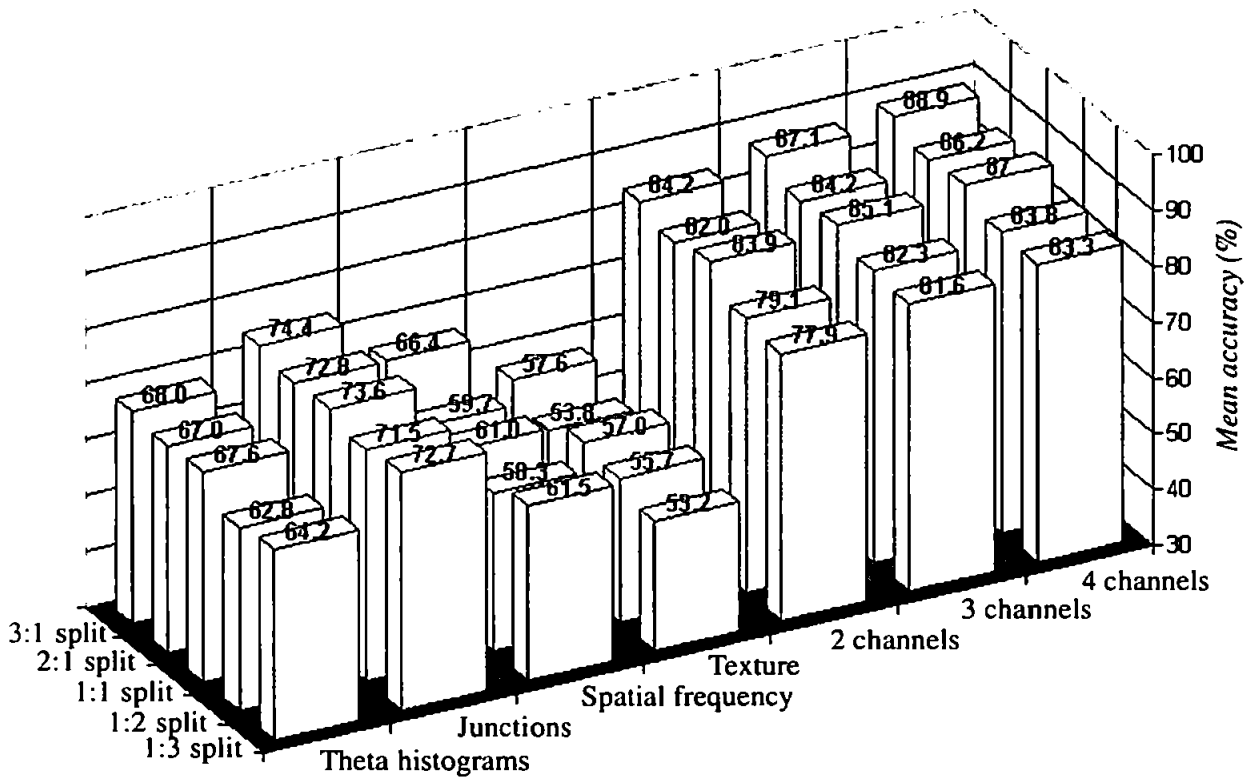


Fig. 9.1. Mean DA 5–object test set classification accuracies for different training/test set sizes (split ratios) and coarse data channel configurations.

It is clear, that with the association of several channels, the performance of the categoriser increases significantly in comparison with the accuracies reported for individual channels. On average, there is a difference of 10% in the observed mean classification accuracy between the tests based on the best individual channel (junctions) and the grouped channels. As the training set's size increases, an increase of 2–6% in the mean accuracies can be observed in the case of a particu-

lar data channel configuration.

By replacing the theta histograms with a stronger shape descriptor, i.e. rho–theta receptive field grid activation patterns, the mean DA test set classification accuracies improve. Fig. 9.2. shows the obtained results. Based on the preliminary tests described in the previous chapter, a 4x4 RF grid was used on each of the considered 3 scale planes and the standard deviation of the Gaussians was set to σ=0.15. These parameters were kept the same in all subsequent tests (involving collectives and committees).

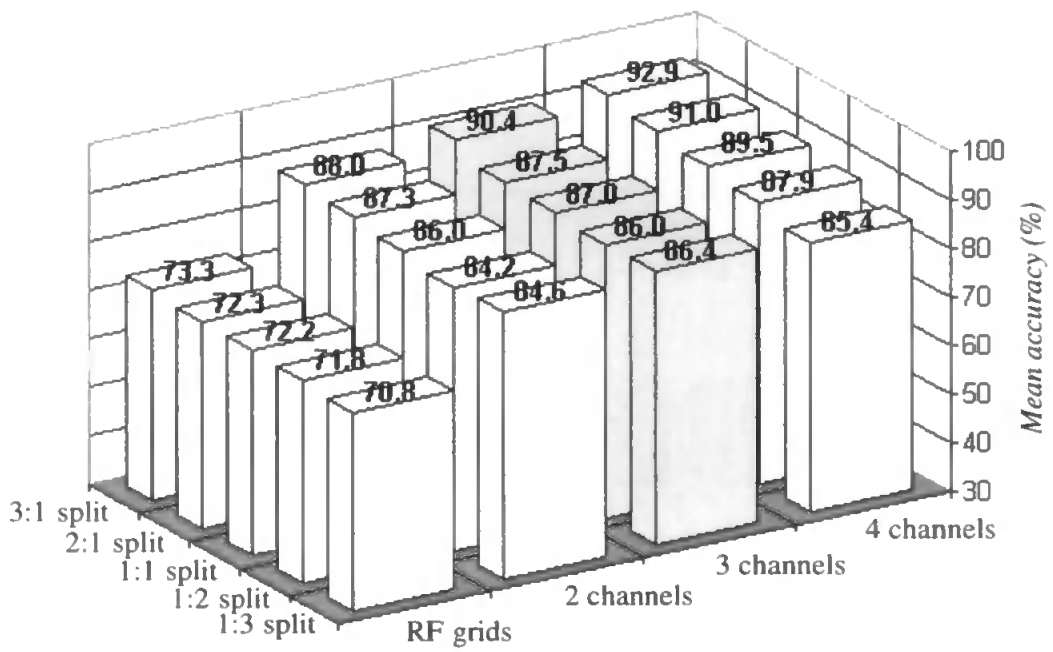


Fig. 9.2. Mean DA test set mean accuracies obtained for the RF–based scale–space channel and 'what' channels associated with it.

For a clearer picture on the categoriser's performance, the overall kappa statistics (across all categories) were calculated based on the above described tests. These results are listed in Table 9.2.; the overall kappas shown in brackets were obtained in tests that used the RF–based scale–space descriptors.

It is apparent, that good to excellent agreement beyond chance was obtained in all of these tests, with high significance values associated with the kappas (e.g. the lowest significance was 19.93 for split 3:1, texture channel). The system's performance is slightly poorer for split ratios 1:2, 1:3 than in the cases in which the training set is larger than the test set.

Table 9.2. Overall kappa values for DA test set results (5–object data set)							
Split ratio	**Theta/ RF**	**Junctions**	**Spatial freq.**	**Texture**	**2 channels**	**3 channels**	**4 channels**
1:3	0.55 (0.64)	0.66	0.52	0.41	0.72 (0.81)	0.77 (0.83)	0.79 (0.82)
1:2	0.54 (0.65)	0.64	0.48	0.45	0.74 (0.80)	0.78 (0.83)	0.80 (0.85)
1:1	0.59 (0.65)	0.67	0.51	0.58	0.79 (0.82)	0.81 (0.84)	0.84 (0.87)
2:1	0.59 (0.65)	0.66	0.49	0.42	0.77 (0.84)	0.80 (0.84)	0.83 (0.89)
3:1	0.60 (0.67)	0.68	0.58	0.47	0.80 (0.85)	0.84 (0.88)	0.86 (0.91)

Since these results don't give a clear picture on how the classification accuracy for each of the objects changes with the channel configurations, for each such configuration and object category the classification accuracies have been calculated and represented in Fig. 9.3. Chance level is 20% and the used data set split ratio was 3:1.

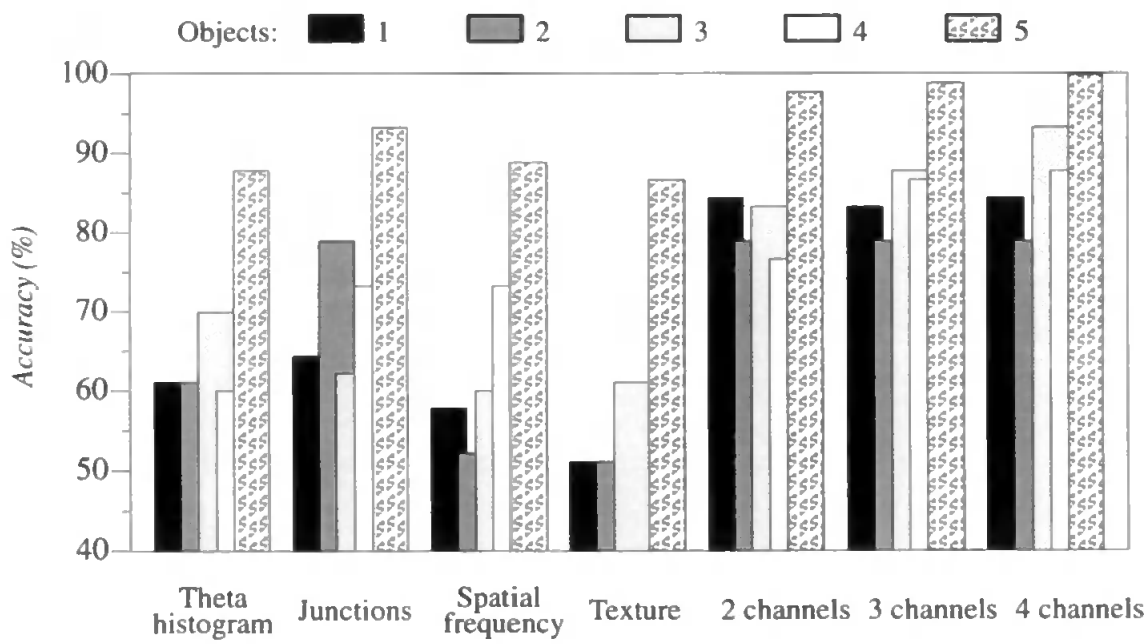


Fig. 9.3. DA classification accuracies for each of the 5 objects, in the case of all channel configurations and 3:1 data set split.

In a similar way, the accuracy of the classification was assessed for each of the 5 categories, dur-

ing tests that used RF grid activation patterns instead of theta histograms. Fig. 9.4. represents these results.

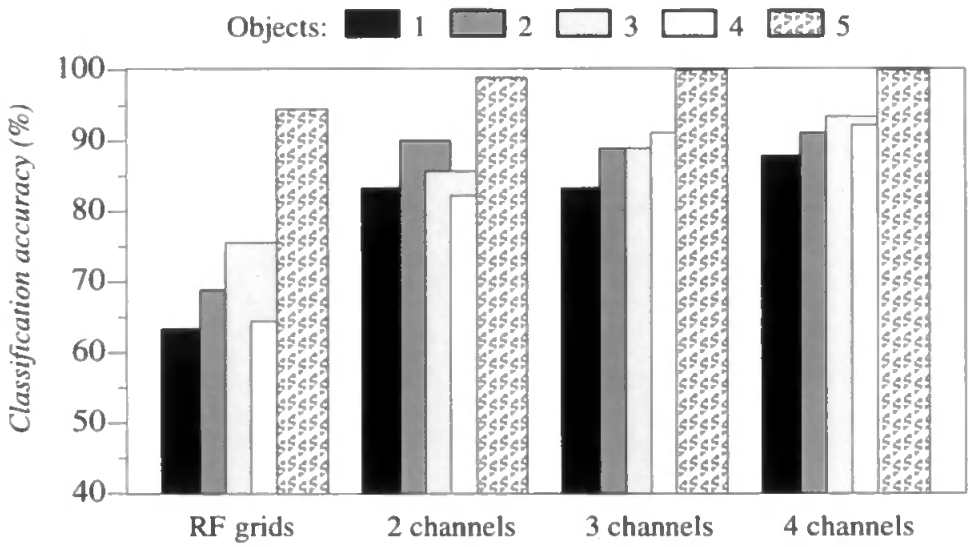


Fig. 9.4. DA test set classification accuracies for each of the 5 objects (split 3:1). Theta histograms are replaced with RF grid activations.

From the above reported results it becomes apparent, that the performance of the system improves when the theta histograms are replaced with rho–theta RF activations. The mean classification accuracies show a 2–6% increase when the theta histograms are replaced by RF activations. When grouped channels are used, all objects are correctly identified with >80% accuracy.

It is evident, that object No. 5 was recognised with the highest accuracy in all cases of channel configuration, while objects 1, 2 and 4 were more confused. For a clearer quantitative measure, the amount of confusion between objects was calculated as the sum of symmetric non–diagonal terms of the confusion table.

Table 9.3. presents the amount of confusion between objects, observed during the trials that used all 4 channels and a data set split ratio of 3:1.

Table 9.3. Mean inter–object confusion in DA trials using all 4 channels' data (%). Split ratio is 3:1.											
Theta histogram + 3 channels						Receptive fields + 3 channels					
Object	1	2	3	4	5	Object	1	2	3	4	5
1	–	23.3	6.6	3.3	0.0	1	–	14.5	4.4	3.3	0.0
2	.	–	7.8	3.3	0.0	2	.	–	3.3	2.2	0.0
3	.	.	–	11.1	0.0	3	.	.	–	7.7	0.0
4	.	.	.	–	0.0	4	.	.	.	–	0.0
5	.	.	.	.	–	5	.	.	.	.	–

As it was expected based on the preliminary trials described in the previous chapter, objects No. 1 and 2 are the most confused with each other. Object No. 5 is clearly distinguished from all the other objects. This is in concordance with the human perception of these objects' similarity to the other objects in the data set (Table 7.3., p. 132). A departure from the way in which human subjects assessed the objects' similarity can be observed in the case of the confusion between objects 3 and 4, which is higher than expected. Object No. 4 is perceived by human subjects as being more similar to objects 1 and 2 than object 3 is. The system, though seems to confuse the latter with the first 2 objects more than it confuses object 4. With the increase of classification accuracy in trials that used RF grids instead of theta histograms, it is not surprising that the inter–object confusion is lower in the RF–based tests.

The training and test sets used in these experiments that employed a statistical classifier were used in sets of neural network–based trials.

9.3.2. Neural network trials

As it has been described in section 9.2.3., sets of 20 network runs were performed for each case of channel configuration, network size and data set split ratio. The effects of changes in these parameters were studied.

Preliminary training/testing trials helped in arriving at a set of learning parameters that led the networks to good performance in a well limited number of training epochs, in all situations of input data dimensionality (i.e. channel configurations). Backpropagation of error with momentum was used as training method. A learning rate of 0.01 and a momentum term of 0.96 was used in all training runs, these parameters leading to unnoticeable changes in the network's mean

square error at the end of 3000 epochs. The sets of 20 training/test runs were set up as automated procedures with the help of the SNNS simulator's batch language. As it was mentioned previously, the networks were initialised at the beginning of each run with random weights between –1.0 and 1.0.

The mean test set classification accuracies obtained during 20 trials, for each of the data set split ratios and data channel configurations are shown below in Fig. 9.5. A middle–of–the–range number of hidden nodes were chosen here, namely 6.

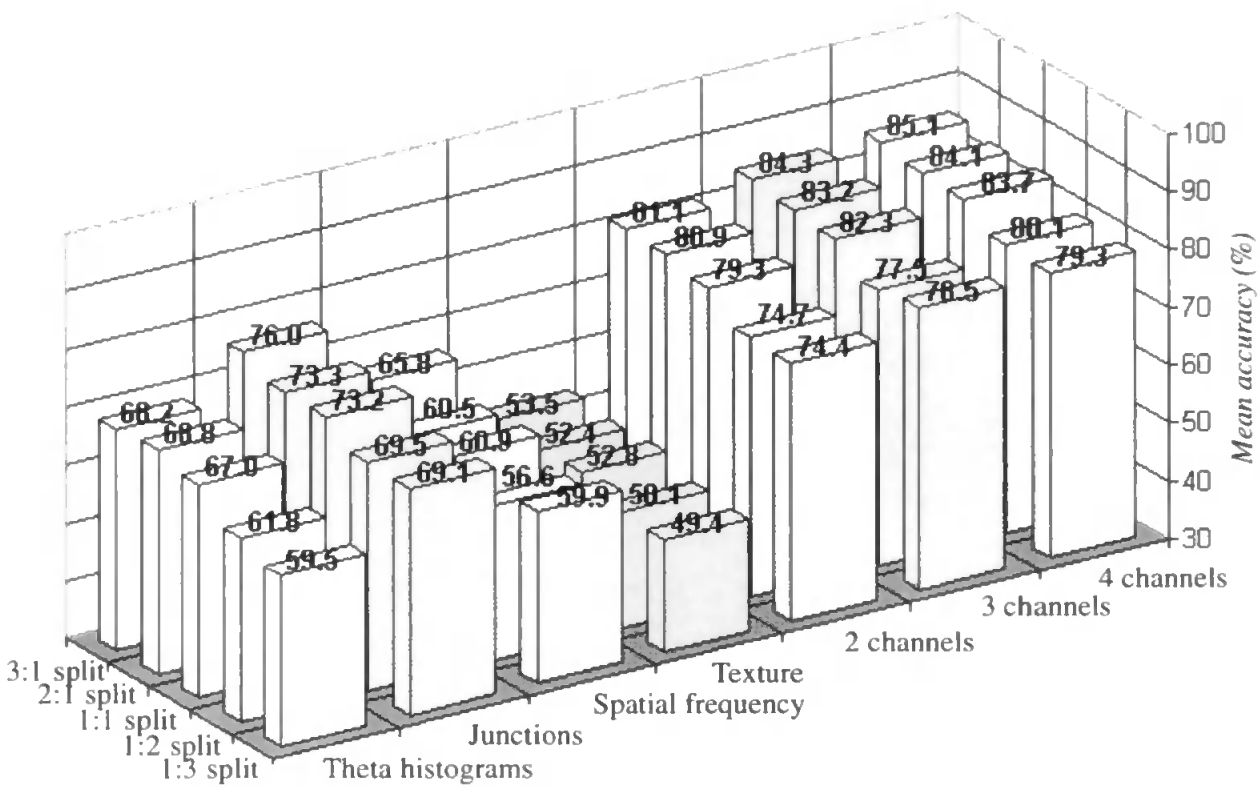


Fig. 9.5. Mean test set classification accuracies for all sets of 20 network runs (5–object data set, 6 hidden nodes).

The same performance measures for the tests in which the theta histograms were substituted with RF activations are shown in Fig. 9.6. below. The mean classification accuracies clearly outrank the ones obtained by using theta histograms.

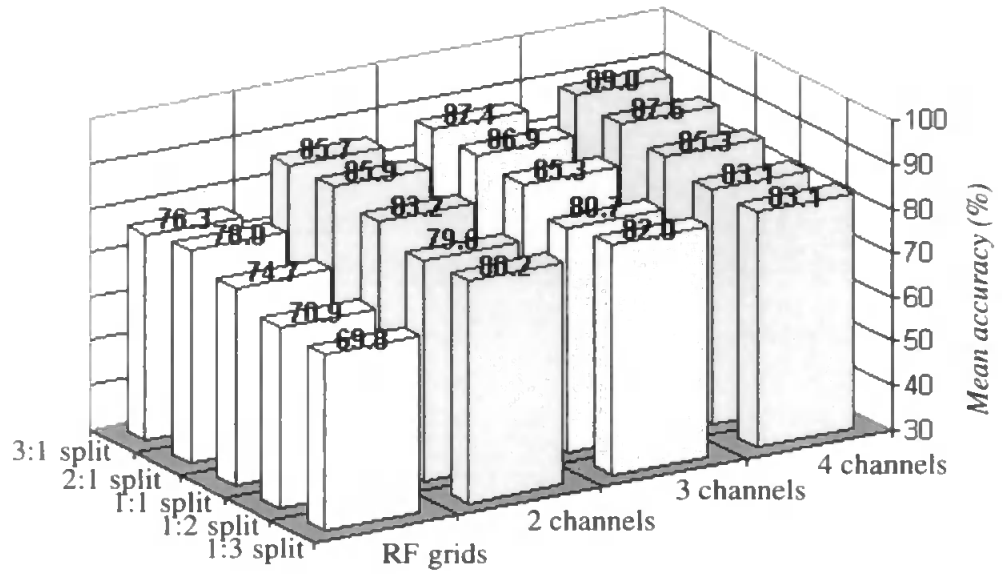


Fig. 9.6. Mean test set classification accuracies obtained by replacing the theta histograms with RF activations (6 hidden nodes).

As it has been observed during the DA trials, the mean classification accuracy increases with the size of the training set. The mean accuracies observed in the 20 trials (performed for each training set size, with 6 hidden nodes in the networks) were used in one–way ANOVA tests. Therefore 5 groups of 20 data samples resulted. For the practical situation, where at least one 'what' channel is associated with the scale–space channel (theta histograms or RF activations), the ANOVA results are reported in Table 9.4. below. The table also contains the degrees of freedom (ν_1, ν_2) in the experiment and the F–table entries for neighbouring values of ν_1 and ν_2.

Table 9.4. Effect of changes in training set size (data set split ratio) on ANN–based collective machine with 6 hidden nodes			
F(4,60)=2.53 ; F(4,120)=2.45 (α=0.05)			
Nr. of channels	**F(4,95)**	***p***	**Effect**
theta + 1	85.5	< 2E–308	extremely strong
theta + 2	87.7	< 2E–308	extremely strong
theta + 3	11.9	6.6E–8	very strong
RF + 1	108.9	< 2E–308	extremely strong
RF + 2	14.0	5.1E–9	very strong
RF + 3	21.6	1.11E–12	very strong

This demonstrates the fact that training set size has a significant effect on the system's perform-

ance – an effect that is suggested by the trend in the mean classification accuracies represented in Fig. 9.5. and 9.6.

It is also clear from these figures, that by associating 'what' channels with the scale–space channel, the system's performance radically improves (as the DA trials also showed). The mean classification accuracy in two–channel trials increased with at least 5% in comparison with the tests based on the best individual channel (junction). This difference further increases as more channels are grouped together.

One–way ANOVA experiments confirmed the presence of the channel effect. The mean classification accuracies observed in 20 runs for the theta histogram channel and group of 2,3 and 4 channels were placed into 4 groups of data. For a mid–range split ratio (namely 1:1), the results of a one–way ANOVA test were F(3,76)=539.0, with p < 2E–308. Considering the fact that F(3,60)=2.76 and F(3,120)=2.68, the results show an extremely significant influence of the association of channels on the system's performance. When the ANOVA test was repeated with 3 groups (i.e. the classification accuracies from only the 2,3 and 4 channel tests), the result was still very significant: F(2,57)=50.02, p =2.9E–13, with F(2,40)=3.23, F(2,60)=3.15.

Similar significant effects were observed in the tests which used RF grid activations as scale–space descriptors. The ANOVAs repeated for these results resulted in an F(3,76)=226.1, p < 2E–308 for the 4–group test (RF channel, then 2,3,4 channels), while the test performed on only the 2, 3 and 4 grouped channel results led to an F(2,57)=28.8, p=2.2E–9.

To illustrate the spread of performance in the multiple trials, for the same number of hidden nodes and for the 3:1 split of the data set (that led to the best results), the lowest, mean and highest classification accuracies obtained during the 20 runs are represented in Fig. 9.7. for all channel configurations.

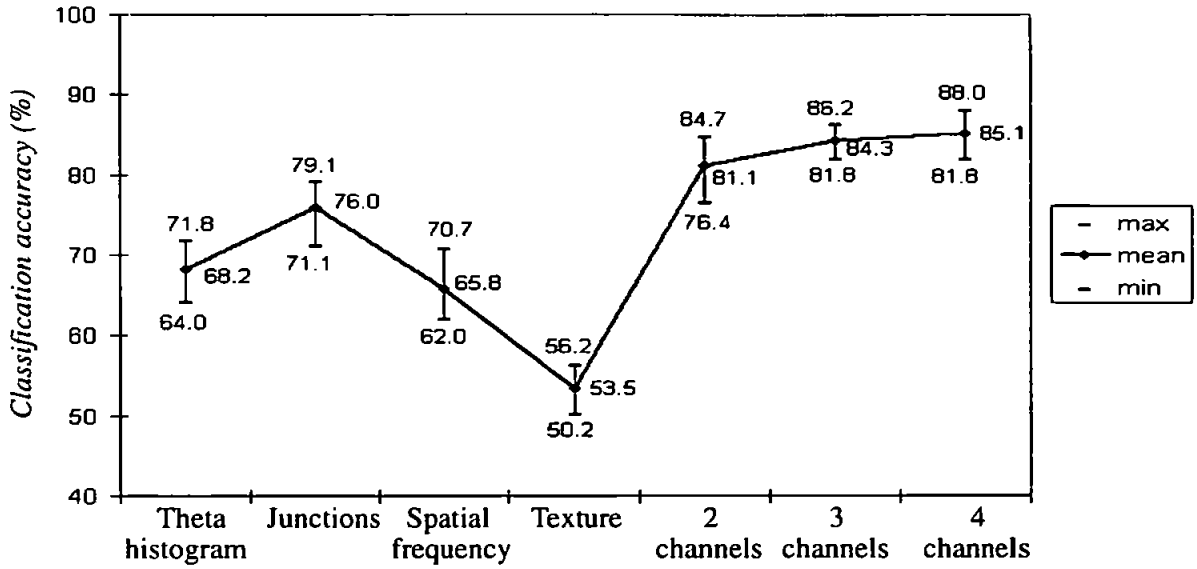


Fig. 9.7. Lowest, mean and highest test set classification accuracies observed during 20 runs (6 hidden nodes, 3:1 split) for the 5–object data set.

In a similar manner, the extremes and the mean accuracies are plotted in Fig. 9.8. below for the RF–based scale–space channel and added 'what' channels.

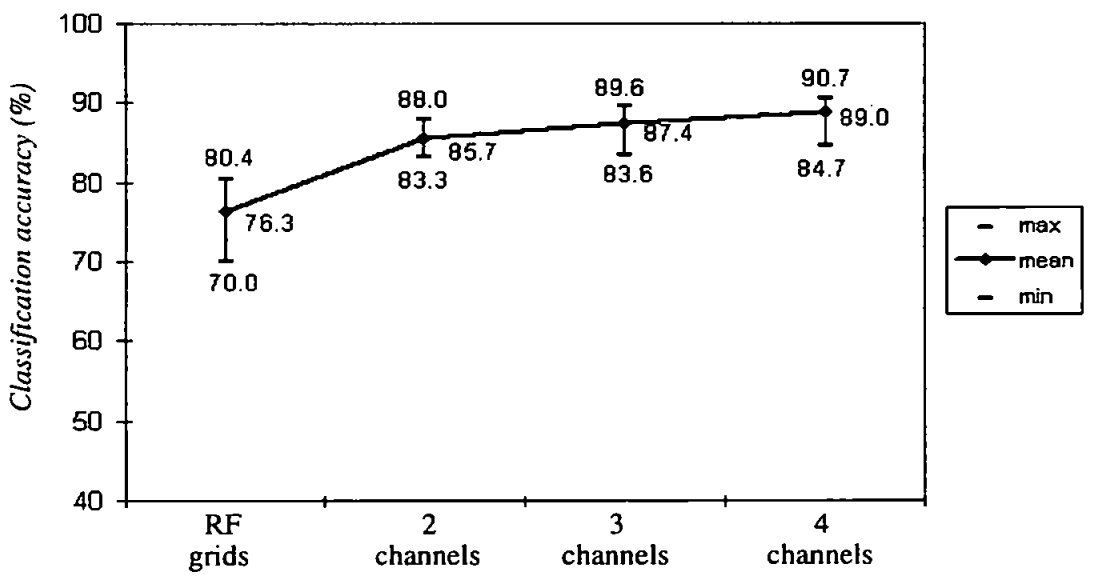


Fig. 9.8. Lowest, mean and highest test set classification accuracies obtained by replacing the theta histograms with RF activations (6 hidden nodes, 3:1 split)

These diagrams, too clearly show the positive effect of channel association. The channel that leads to the lowest performance is the texture channel, which is not surprising in the light of the surface properties of the test objects. All of these observed patterns in the performance statistics are consistent with the DA test results.

The overall kappas across the 5 categories were averaged over 20 runs for each data set split and coarse data channel configuration, when using 6 hidden nodes. These are reported in Table 9.5. below. The mean overall kappas show eloquently the improvement in performance, when RF activations are used instead of theta histograms.

Table 9.5. Mean overall kappas from 20 test runs, for each of the channel configurations and data set splits (6 hidden nodes)

Split ratio	Theta/ RF	Junction	Spatial freq.	Texture	2 channels	3 channels	4 channels
1:3	0.49 (0.62)	0.61	0.50	0.37	0.68 (0.75)	0.73 (0.78)	0.74 (0.79)
1:2	0.52 (0.64)	0.62	0.46	0.38	0.68 (0.74)	0.72 (0.76)	0.75 (0.79)
1:1	0.59 (0.68)	0.66	0.51	0.41	0.74 (0.79)	0.78 (0.82)	0.80 (0.82)
2:1	0.61 (0.73)	0.67	0.51	0.41	0.76 (0.82)	0.79 (0.84)	0.80 (0.85)
3:1	0.60 (0.70)	0.69	0.57	0.42	0.76 (0.82)	0.80 (0.84)	0.81 (0.86)

The lowest kappa (0.33) was observed during the network runs performed on texture data only, while using split 1:3. All kappas had high significance values being associated with them, the lowest significance score being 16.35. Based on the above mean kappas, it can be stated that good to excellent agreement beyond chance has been observed between the categories and the neural network–based categoriser.

The classification accuracies for each category, averaged over 20 runs are represented in Fig. 9.9., for the situation in which 6 hidden nodes were used in the networks and the data set split ratio was 3:1.

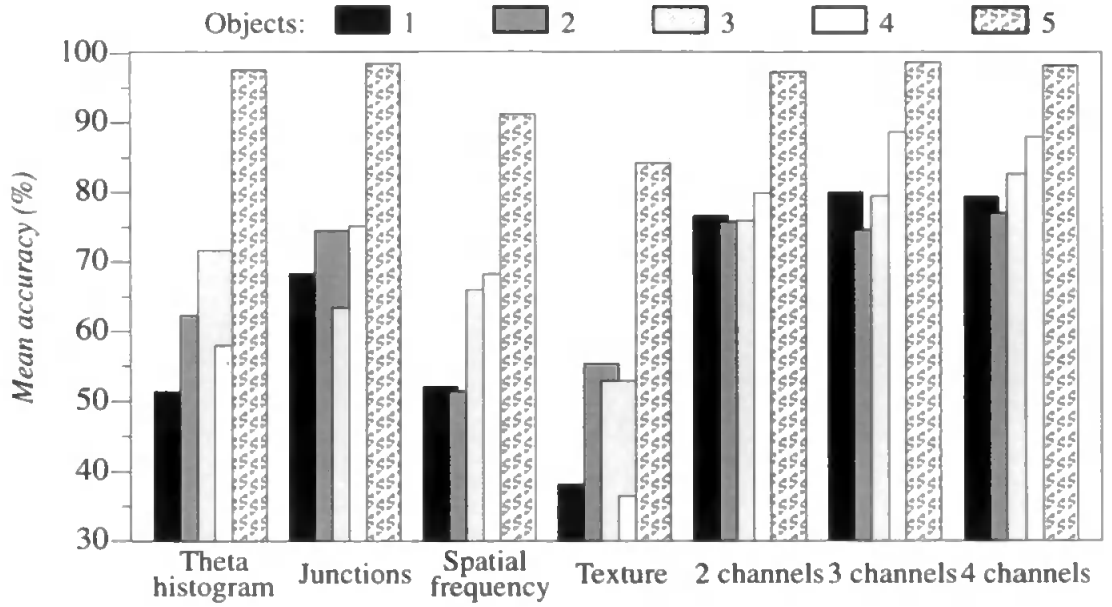


Fig. 9.9. Test set classification accuracies observed for each category of the 5–object data set, averaged over 20 runs (3:1 split, 6 hidden nodes).

In a similar fashion, the object–specific accuracies were evaluated when the theta histograms were replaced by rho–theta receptive field activation patterns. The results for split 3:1 are shown below in Fig. 9.10.

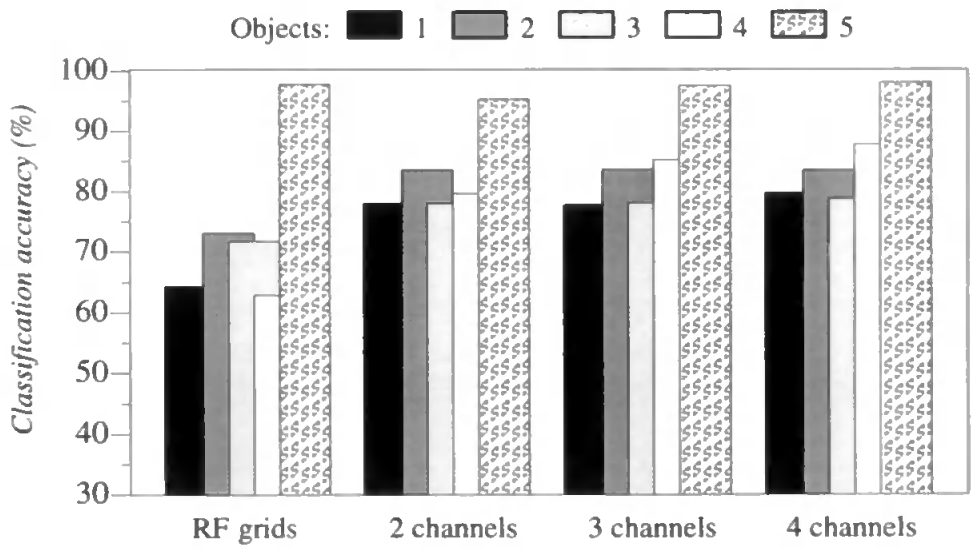


Fig. 9.10. Test set classification accuracies for each of the 5 objects (split 3:1). Theta histograms are replaced with RF grid activations.

The mean confusions between the object categories were calculated by averaging the sums of

symmetric non–diagonal terms of the confusion tables over all 20 runs. Table 9.6. lists these values for the practical situation when all 4 data channels are used (the data set split ratio was 3:1). As in the previous evaluations, the results are those obtained from network runs that used a hidden layer of 6 nodes.

Table 9.6. Mean inter–object confusion in ANN trials using all 4 channels' data (%). Split ratio is 3:1.

Theta histogram + 3 channels						Receptive fields + 3 channels					
Object	1	2	3	4	5	Object	1	2	3	4	5
1	–	25.7	15.1	5.3	0.1	1	–	18.9	13.5	2.1	0.8
2	.	–	10.8	2.9	0.6	2	.	–	6.4	3.6	0.3
3	.	.	–	9.8	2.0	3	.	.	–	7.7	0.9
4	.	.	.	–	2.2	4	.	.	.	–	1.5
5	.	.	.	.	–	5	.	.	.	.	–

The above figures and the amounts of confusion between objects show a similar pattern to the one observed during the DA tests.

In addition to the studies on channel configurations and training/test set sizes, the effect of changes in the number of hidden nodes has been assessed. The mean recognition accuracies observed during the sets of 20 runs, performed for 4, 6 and 8 hidden nodes only showed an increase of 2–3% for each increase of the number of hidden nodes by 2. One–way ANOVAs were calculated based on the 20 classification accuracies observed for each of the three network sizes.

The size of the networks had only a relatively weak effect on the performance in the case of 3 and 4 channel trials. The lowest F was obtained for the 3–channel test results (1:1 split, theta histogram + 2 channels): $F(2,57)=6.85$, $p=0.002$, with $F(2,40)=3.23$, $F(2,60)=3.15$. The 2–channel tests led to ANOVA results that show the absence of network size effect. The highest F and lowest p was obtained for 1:1 split, theta histogram + 1 channel: $F(2,57)=2.4$, $p=0.09$, which is still clearly insignificant.

Having completed the above described tests and evaluations, the categoriser module was replaced with one that employed committee machines. These made possible comparisons between classifier performances obtained from an ensemble of feature data and decisions based on individual channel behaviour.

9.4. Tests with committee machines

Two different committee machine architectures were used in these tests. The committee that produced as final verdict the decision of the most confident channel allowed the study of the discriminatory power of each channel in circumstances where a channel verdict is competing with other channels' categorisers. To briefly reiterate the principles stated in section 6.4.5. (p. 123), confidence is defined as the difference between the highest and the second highest output neuron activation found in a classifier associated with a coarse data channel.

The disadvantage of this decision taking method is the fact that the feature data produced by each channel does not contribute in chorus to the classification process. A particular channel's categoriser can be very confident and wrong in the same time, this affecting the performance of the system. Therefore such a committee was expected to deliver significantly poorer performance than a collective machine.

An improvement to the decision taking process can be brought by considering the sum of activations of corresponding output neurons of each channel categoriser. With this, the activations of all channel categorisers contribute to the final output activation pattern, but the classification still doesn't happen based on high–dimensional multi–channel feature data.

As it has been mentioned in section 9.2.1., the output activations of the neural networks trained/tested on individual channels' data have been saved in all of the runs and tests reported above. These activations being used in the committee machine trials, direct comparisons with the collective machine tests can be made.

As a convention, the committee machine using most confident channel for decision taking will be denoted CMC, while the one using sum of output activations will be denoted as CSO.

The mean classification accuracies obtained in 20 runs with a CMC machine are reported below in Fig. 9.11. This figure shows the results of the trials that used theta histograms and RF activations as scale–space data. The tests were carried out in situations where 2,3 and 4 channel categorisers were members in the committee. The texture channel had no contribution to performance in this committee machine, therefore the very same results were obtained in the 3 and 4–channel trials.

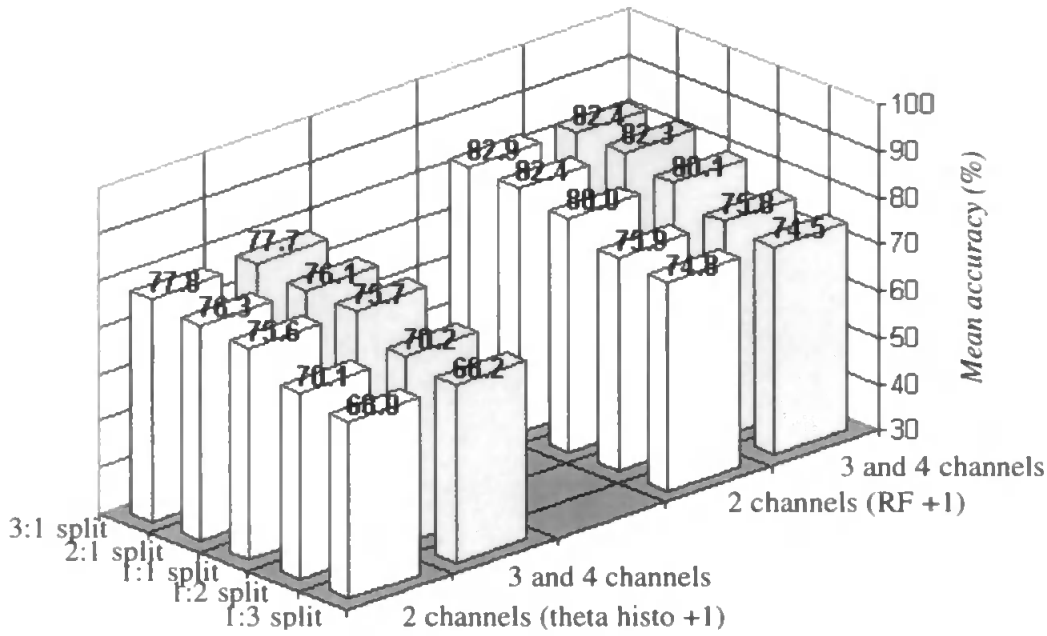


Fig. 9.11. Mean test set classification accuracies obtained for most–confident–channel, committee machine in 20 runs (6 hidden nodes).

The contribution of a particular committee member (i.e. categoriser operating on an individual channel's data) to the performance of the committee can be assessed by counting the number of correctly classified cases for which the investigated member was the most confident. When only two channels were introduced into the CMC machine, it has been found that on average, the scale–space channel led the committee to correct decisions in 51.1% of all correct decisions and the junction channel 48.9% of all correct decisions. When three or all four committee members were present, these proportions were an average of 47.8% for the scale–space channel, 45% for the junction channel and 7.2% for the spatial frequency channel. This shows that the scale–space channel categoriser had the highest contribution to correct committee decisions, with the junction channel as a close second. The spatial frequency channel in a few cases contributed to the overall performance of the committee machine, but the texture channel had no effect at all on the committee's decisions.

In a similar manner, the mean accuracies were obtained for the CSO machine. These results, for both theta histogram and RF–based scale space channels are listed in Fig. 9.12. below. In this case, the texture channel had a contribution to the performance – although it is a channel with low discriminatory power, due to the unweighted sum of output neuron activations its decisions did count in the final decision taking process.

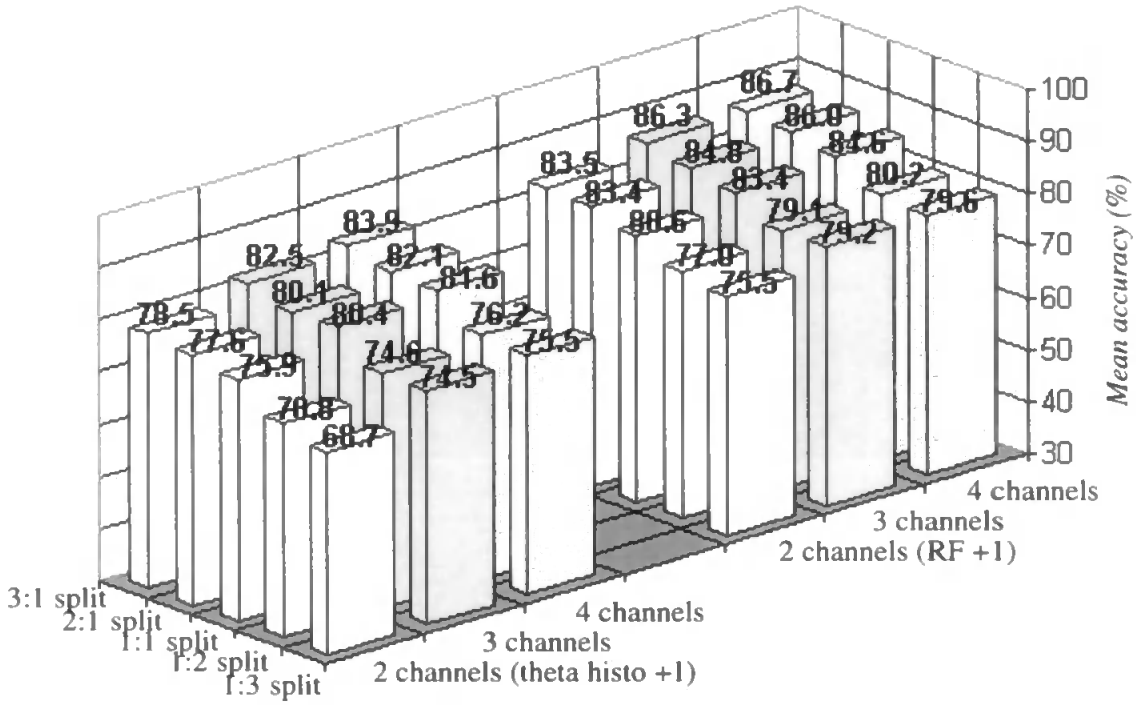


Fig. 9.12. Mean test set classification accuracies obtained in 20 runs from sum–of–outputs committee machine (6 hidden nodes).

The mean classification accuracies show that the size of the training set significantly affects the performance. One–way ANOVA tests carried out on the results of each trial for each of the 5 split ratios have shown the presence of a very strong data set size effect in all experimental situations, as Table 9.7. below shows. These tests were based on committee members that had 6 hidden nodes in the trials.

Table 9.7. Effect of changes in training set size (data set split ratio) on committee machines

F(4,60)=2.53 ; F(4,120)=2.45 (α=0.05)

CMC machine			
Nr. of channels	**F(4,95)**	***p***	**Effect**
theta (RF) + 1	140.9 (124.3)	< 2E–308	extremely strong
theta (RF) + 2,3	135.3 (123.6)	< 2E–308	– " –
CSO machine			
theta (RF) + 1	140.9 (119.6)	< 2E–308	extremely strong
theta (RF) + 2	125.7 (131.6)	< 2E–308	– " –
theta (RF) + 3	210.5 (173.5)	< 2E–308	– " –

The effect of channels being added to the system (i.e. outputs of categorisers associated with channels) has been evaluated, too. Since in the case of the CMC machine, the insertion of FFT and texture channels had extremely minor or no influence on the performance, it is not surprising that the ANOVA tests showed insignificant F values for this committee machine. When submitting to one–way ANOVA the overall classification accuracies obtained for split 1:1, 6 hidden nodes and grouped according to the number of used channels, the results were F(2,57)=0.004, $p = 0.95$ in the presence of theta histogram–based scale space channel, and F(2,57)=1.96, $p = 0.17$ when RF activations were used in conjunction with other channels.

The introduction of more channel categorisers into the committee that considers the sum of output neurons in the decision taking process had a major influence on the performance. The ANOVAs carried out in this case on data obtained from tests that used 1:1 data set split ratio led to F(2,57)=75.10, p = 1.11E–16 and F(2,57)=37.68, p = 3.74E–11 for theta histogram and RF–based scale space channel + 'what' channels, respectively.

The lowest, highest and mean accuracies obtained by the CMC machine in the 20 runs that the committee members were submitted to are shown in Fig. 9.13.

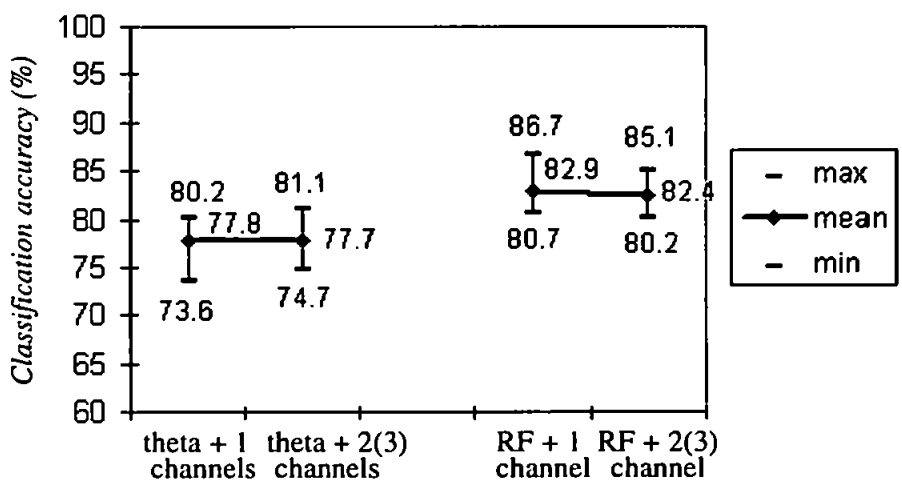


Fig. 9.13. Lowest, mean and highest test set classification accuracies obtained in 20 runs (6 hidden nodes, 3:1 split) by CMC machine.

The same measures of performance are plotted in Fig. 9.14. for the CSO machine. It can be observed, that the adding of more channels to the CMC machine did not improve the performance of the committee, since the committee members do not collaborate in reaching a decision. But in the case of the CSO machine, where the outputs of the committee members are summed and

the decision depends on this joint output, the adding of new channels improves the accuracy of the committee.

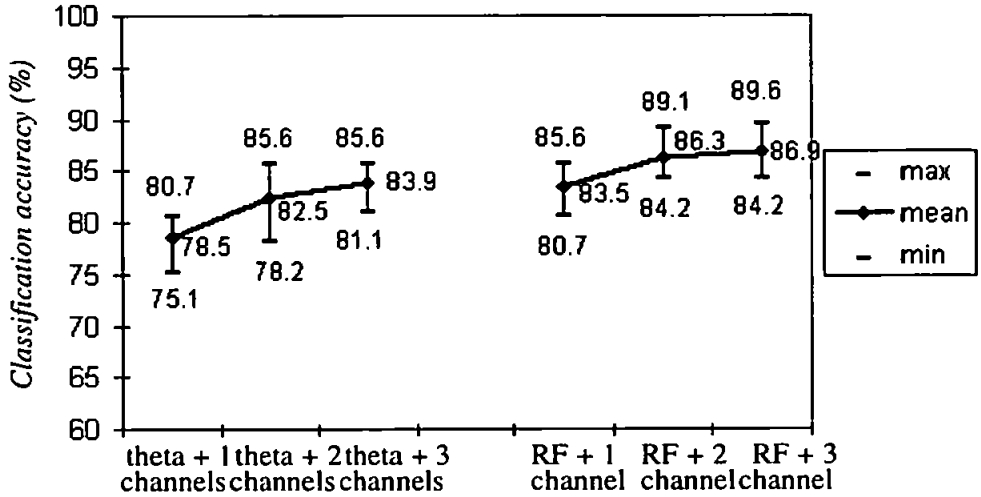


Fig. 9.14. Lowest, mean and highest test set classification accuracies obtained in 20 runs (6 hidden nodes, 3:1 split) by CSO machine.

The overall kappas calculated from these test results are summarised for both committee machines and scale space channels in Table 9.8. below. All kappas were obtained with high significance levels, the lowest significance being 28.41 and the smallest kappa value was 0.56. Both these minima were obtained for 1:3 split, 2 channel setup, CMC machine.

Table 9.8. Mean overall kappas calculated from 20 runs in the case of two committee machines. Kappas in brackets are the RF+1,2,3 channels results.

	CMC machine		CSO machine		
Split	**2 channels**	**3 & 4 channels**	**2 channels**	**3 channels**	**4 channels**
1:3	0.60 (0.68)	0.60 (0.68)	0.61 (0.69)	0.68 (0.74)	0.69 (0.75)
1:2	0.62 (0.70)	0.63 (0.69)	0.63 (0.71)	0.68 (0.74)	0.70 (0.75)
1:1	0.69 (0.75)	0.69 (0.75)	0.70 (0.76)	0.75 (0.79)	0.77 (0.81)
2:1	0.70 (0.78)	0.70 (0.78)	0.72 (0.79)	0.75 (0.81)	0.78 (0.82)
3:1	0.72 (0.78)	0.72 (0.79)	0.73 (0.79)	0.78 (0.83)	0.80 (0.84)

The kappas, in addition to the mean accuracies prove that the committee machines' performance is more than satisfactory.

In order to investigate the accuracy of the committee in identifying each object in the data set, the test set classification accuracies observed for each object were averaged over the 20 runs and plotted for the CMC machine in Fig. 9.15. below. As in the case of collective machines, objects 1 and 2 are the least accurately recognised, while object No. 5 clearly stands out even in this situation where a single, most confident channel delivers the object label.

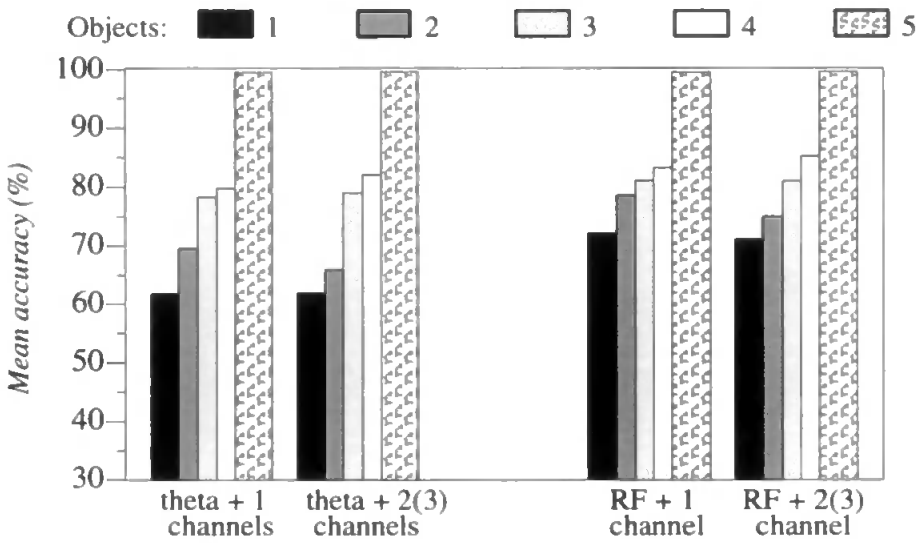


Fig. 9.15. Test set classification accuracies obtained for each object, averaged over 20 runs (3:1 split, 6 hidden nodes) of CMC machine.

In a similar manner, the category–specific classification accuracies averaged over 20 runs are represented in Fig. 9.16. below for the CSO machine. The pattern of recognition accuracies is similar to the ones observed previously.

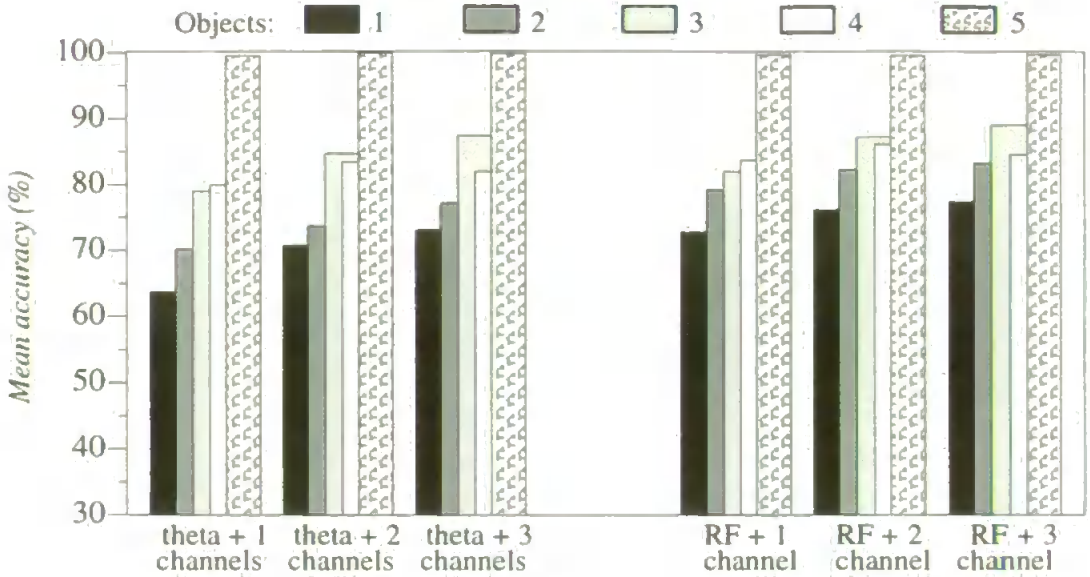


Fig. 9.16. Test set classification accuracies obtained for each object, averaged over 20 runs (3:1 split, 6 hidden nodes) of CSO machine.

The mean inter–object confusion has been evaluated as in the collective machine tests. These percentages are reported for both committee machine types and scale space channel structures in Table 9.9. The results are based on tests with 3:1 data set split and 4 channels' categoriser outputs, which led to the best results. The categorisers (i.e. committee members) used 6 hidden nodes.

Table 9.9. Mean inter–object confusion in committee machine trials using all 4 channels' data (%). RF–based test results in brackets. Split ratio is 3:1.

CMC machine						CSO machine					
Object	**1**	**2**	**3**	**4**	**5**	**Object**	**1**	**2**	**3**	**4**	**5**
1	–	29.8 (21.4)	18.1 (17.6)	11.0 (8.8)	3.3 (2.6)	**1**	–	28.9 (24.2)	13.8 (11.1)	7.6 (5.9)	0.7 (0.1)
2	.	–	11.4 (7.3)	10.1 (6.8)	3.1 (4.1)	**2**	.	–	8.4 (5.1)	5.7 (3.8)	0.1 (0.2)
3	.	.	–	16.3 (13.2)	3.1 (1.8)	**3**	.	.	–	12.6 (11.8)	0.5 (0.3)
4	.	.	.	–	5.2 (4.6)	**4**	.	.	.	–	2.2 (3.3)
5	.	.	.	.	–	**5**	.	.	.	.	–

For testing the effect of hidden nodes in each of the committee members on the performance of the committee, one–way ANOVAs were calculated from results of tests that used 1:1 split ratio,

all 4 channels and both scale–space channel variants. The overall (mean) classification accuracies in each set of 20 runs for all considered hidden node numbers show an increase of only 3–4% as the hidden layer's size increases. The ANOVAs based on the individual classification accuracies (20 for each hidden node number) has shown that this difference can be attributed to the size of the hidden layer of channel categorisers. The lowest, still significant F value was $F(2,57)=5.3$, $p = 0.02$ for the theta histogram–based scale space channel + 3 channels and CMC machine, while for RF–based trials the lowest F was $F(2,57)=8.85$, $p = 0.005$ (CSO machine).

This set of tests concludes the investigation on the collective and committee machines operating on 5–object synthetic data set. The following section discusses the results obtained in the tests reported above.

9.5. Conclusions and discussion

The tests carried out on the collective and committee machine architectures, with feature data that characterises the views in the computer–generated 5–object data set showed that good classification can be achieved based on the developed coarse data channels.

The statistical classifier–based collective machine outperformed the ANN–based collective machine, the mean accuracies obtained in DA tests being on average 4% higher than the ANN results. The DA overall kappas, too are slightly superior to the ANN tests' coefficients of agreement (there is a difference of about 0.05). No far reaching conclusions can be drawn from these minor differences, but the aspect of data pruning is to be noted. The DA–based categoriser automatically prunes histogram bins that are unpopulated in all of the presented cases, hence only the relevant data is kept in the analysis. With the increase of the input dimensionality by grouping channels together, the number of empty histogram bins can increase – it is likely that the performance of ANN–based classifiers trained on such data can be improved by employing weight pruning algorithms.

Further to the preliminary tests reported in the previous chapter, the discriminatory power of the data channels in circumstances of large training/test sets with ample variations of viewpoint has been investigated during the above described trials. For comparative study, individual channels' coarse coded feature data has been submitted to DA and ANN–based categorisation during the tests on collective machines. In these trials, the minimum distance between the feature descriptor

and a cluster centre (in the case of DA) or the winner output neuron decided the category label. Individual channels defined the decision of a categoriser also in the case of the CMC machine, in this case the most confident channel delivered the verdict. In the former trials it could be seen, that the channels with the best discriminatory power were the junction and the scale–space channel (in this order). Due to the geometric properties of the test objects, the junction information alone was sufficient for reaching an above 70% classification accuracy. Since the outputs of the categorisers trained and tested on individual channel data were saved during these tests, committee machines could operate based on these outputs and direct comparisons could be made. The CMC machine trials revealed, that the categoriser that was the most confident and correct in the same time was the one presented with scale–space channel data. The junction channel and spatial frequency channel followed, while the texture channel had no contribution to the correct decisions made by this committee. Since the test objects' surface properties, shading and illumination are very similar, the low discriminatory power of channels that operate with contrast measures and texture densities could be expected.

By grouping data channels together, the performance of the system improves significantly, as proven by one–way ANOVA tests that investigated the channel effect. In the case of collective machines, with the increase in the number of channels and therefore the dimensionality of the feature data, the classification accuracy of the system monotonously increases. Even the insertion of a channel with very low discriminatory power has a positive effect on performance, as it has been observed in the past during the 2D shape recognition work carried out on the DiCANN system (Ellis *et al.*, 1997). In this case, due to the identical surface properties of the objects, the texture density channel was proven to be the weakest from the point of view of discriminatory power. By adding the texture data to the other 3 channels, on average a 2% increase in mean classification accuracy has been observed in tests on collective machines, in comparison with the 3–channel results. The channel effect failed to appear in only one case, where the classifier did not reach its decisions based on joint channel information. This case, represented by the CMC machine showed that verdicts reached based on individual channel outputs is not affected positively by the insertion of more channels into the system. The presence of the texture channel in the committee of the CMC machine did not have any effect on the performance, since other channels proved to be more confident in their (correct or erroneous) category label decisions. As it became apparent in Fig. 9.13., the presence of more 'what' channels actually caused a slight (0.5–1%) decrease

in performance in comparison with 2–channel test results. As channels with low discriminatory power became members of the committee and decided confidently in an erroneous way, this negatively affected the overall performance of the committee machine. After all, a committee machine like CMC can reach a decision based only on a confident 'what' channel, without the support of the 'where' scale–space channel.

Collectives are clearly better classifiers than committees, as it was expected based on preliminary tests and work in 2D shape recognition. ANN–based collective machines were on average 8% better than CMC machines and on average 3% better than CSO machines. These differences increase with a further 4% when comparing DA–based collectives with the two committee machine architectures. As it has been described above, committees like the CMC have no collaboration between members and if a very confident member is wrong, the categoriser's performance suffers. As the above figures show, CSO machines were more correct than CMC, with 5% on average. Despite the fact that the former machines' decision is not based on the joint feature data supplied by all member channels (as it happens in a collective), the fact that these took into account the sum of outputs of the members introduced a weak collaboration between these. An erroneous confident output of a member could be countered by the equally weighted sum of the other members' output.

The scale–space channel variant that uses RF grid activation patterns instead of theta histograms was more sensitive to changes in the 2D shape in the preliminary trials described in the previous chapter. The tests on these rigid synthetic shapes reported in previous sections did not confirm the worries expressed in chapter 8. It seems that with sufficient amount of training data, a categoriser presented with the RF–based scale–space channel's data successfully learns the object shapes, their variations and actually achieves better performance than classifiers operating with theta histogram data. In DA and ANN–based collective machine trials, the use of the RF–based scale–space channel on average led to a 5% increase in classification accuracies, compared to the ones obtained in theta histogram–based tests. In trials that involved CMC and CSO machines, this difference was around 4% and 6%, respectively. It is still a question whether the RF–based shape descriptor has the same advantage in the case of non–rigid shapes.

Changes in training set size had an effect on all tested categorisers. Committee machines seemed to be more sensitive to the amount of training data, judging by the difference between the mean classification accuracies measured in the case of data set split ratios 3:1 and 1:3. This difference

is about 8% for the CMC and CSO machines, while the ANN and DA–based collectives displayed a difference of about 5% and 7%, respectively. The differences in overall kappas, too (observed for these two extreme split ratios) showed that committees are more affected by changes in the amount of training data. These differences were on average 0.11 for committee machines, while ANN and DA–based collectives showed a difference of 0.08 and 0.06, respectively. One–way ANOVAs carried out for all the categoriser tests has shown that these variations are due to changes in the split ratio. The fact that collectives are less affected by the variation in the size of the training set proves again the advantage of categorisation based on an ensemble of feature data. The above described lower differences in performance and the better mean classification accuracies achieved by collectives show that collectives generalise better from a limited number of learnt feature data items, than committees do. The latter categorisers have to operate with the decisions of individual channel classifiers, while the classification based on joint feature data proves to be superior. This is in accordance to the conclusions drawn in work on 2D shape classification (Culverhouse *et al.*, 1996; Ellis *et al.*, 1997).

Due to limitations on the available simulation time and storage space, large variations of the size of the used neural networks was not imposed. Starting with a number of hidden nodes that made good network learning possible, a few nodes were added in steps. The effect of hidden layer size on ANN–based collective machines was proven to be present, one–way ANOVAs showing that 1–2% increases in mean classification accuracy were due to this factor. Significant effect has not been observed in the case of collectives operating with only the two strongest channels (scale–space and junction). Since the choice of the number of hidden nodes in a network is linked, among other things, with the dimensionality of the input, one could assume based on the above results that the 4 hidden nodes are not sufficient for providing good generalisation based on high–dimensional learnt data and more hidden nodes are necessary in order to improve the performance. Conversely, in the 2–channel case, the 4 hidden nodes would be sufficient for good learning and generalisation of such low–dimensional data, therefore more hidden nodes, without leading to overfitting, do not have any significant effect on performance. But the observed absence of hidden layer size effect in the latter case can not be attributed to the low input dimensionality, since in the case of all committee machines (operating with even lower dimensional inputs, i.e. individual channels) this effect was present and was significant. It is likely that the data in the 2–channel collective machine's case was sufficiently good descriptor of the analysed shapes to

lead to good generalisation, independently from minor changes in the number of hidden nodes.

In all cases of categoriser structure (collectives or committees), it was interesting to draw a parallel between the human perception of the objects' similarity and a quantitative measure of how the system confuses these test objects. The inter–object similarity scores measured and reported in chapter 7 and the mean inter–object confusion tables were used for these comparisons. In all trials, the most dissimilar object (No. 5) was easily identified by categorisers, even by the CMC machine operating on individual channel decisions. The first two objects were the most confused between each other, and confused with objects No. 3 and 4 more than they were mistaken for object 5. This, too is similar to the way in which human subjects assessed shape similarity. Departures from the human perception of the degrees of inter–object similarity was observed in the case of objects No. 3 and 4, which were more confused between each other by the system than it was desired based on the human subjects' mean similarity scores. This is very likely due to the way in which the shapes of the objects are characterised by the feature vectors, in circumstances of large variations in viewpoint. The two objects' signature in feature space for a number of relevant views may have ended up to be similar, hence the confusion.

9.6. Summary

This chapter described the object recognition experiments that involved the computer–generated 5–object data set. A number of different classifiers have been used to categorise the feature data provided by various configurations of coarse data channels. Statistical and ANN–based collective machines, committee machines have been employed, their performance registered and compared. The system's ability to categorise a limited number of synthetic shapes has been studied in conditions of variable training set size, network size, channel configuration. A number of conclusions regarding the behaviour of the system, the discriminatory power of certain coarse data channels could be drawn based on the results. In all cases, the system proved to be able to generalise from training sets of various sizes to unknown data with very satisfactory accuracy. Some of the observed tendencies and changes in performance made possible the generalisation of a number of aspects studied in the past in the field of 2D shape recognition base on multiple coarse data channels.

The following chapter describes a similar set of tests that used as input a set of natural images. Due to the very different nature of the images contained in that set, it provides a further tool for

the study of the coarse data channels' discriminatory power and the system's performance when presented with very similar non–rigid shapes and poor quality images.

Chapter 10. Classification of natural shapes

10.1. Introduction

The experimental protocol established for the computer–generated set of object views was repeated for the Aberdeen data set that contained images of fish larvae and detritus separated from the original photomicrographs. The following sections reiterate the characteristics of this data set, describe the objectives of the experiments and the way in which the data was prepared. Tests that involve collective and committee machines are described, while the concluding section of this chapter discusses the aspects related to the system's behaviour that are different from the ones observed during the experiments conducted with the 5–object data set. Conclusions regarding the performance of collectives vs. committees, statistical vs. ANN–based categorisers are drawn and the discriminatory power of the coarse data channels is discussed.

10.2. Setting up the experiments

This section describes the objectives of the tests and, based on these, the ways in which the data and experimental parameters were prepared. The main issues raised by the properties of the Aberdeen data set are highlighted.

10.2.1. Objectives

As it has been described in section 7.2.4. (p. 132), this data set presents quite different problems to a categoriser. Images are corrupted by particles present in the water and noise partly due to the imaging equipment. The larvae are extremely similar and their discrimination is a difficult task even for expert taxonomists. In these conditions and with no control over the viewpoints (like in the case of the computer–generated 5–object data set), the recognition system faces a different challenge from the tests described in the previous chapter.

The larvae have non–rigid shapes, their appearance changing in different developmental stages. Therefore a strong dependency of test set classification accuracy on training set size was expected. It is thought that a small training set would not be able to capture the characteristics of each larvae in all morphological stages, hence the system's ability to generalise to novel speci-

mens would be significantly compromised. The detritus (present as a 4th category in the data set) easily distinguishes from larvae even in cases of very small training sets, since the particles are very different from the larvae.

Because of the non–rigid shapes of the larvae, it is interesting to investigate how the discriminatory power of stricter shape descriptors (like the rho–theta receptive field activation patterns) is affected. Are these more affected by the variations in object shape than theta histograms? Also, surface textures and local contrasts differ more in the case of larvae images than they did in the 5–object data set, therefore the role of spatial frequency and texture channels in categorisation would be expected to be different from previous observations.

Tests carried out on collective and committee machines would build a picture on the role of each coarse data channel in the classification, as they did in the trials that involved the 5–object data set. Following the protocol established during the latter trials, the effects of channel configuration, hidden layer size, training set size on classification accuracy are going to be studied, together with the performance of statistical and neural network–based categorisers.

10.2.2. Preparations

In the case of the Aberdeen set, since fish larvae were represented in different developmental stages, the training set contained randomly picked images of specimens in all stages of development, hoping to achieve a desired good representation of the categories. The data set was split into training and test sets by keeping a number of n images per category in the training set and placing the remainder of the images in the test set. This was due to the small number of views that were available for each specie of larvae. The values of n were chosen to be 10, 20, 30 and 40, this allowing the study of performance variations with training set size in conditions of a limited number of training/test set pairs. Since the data set is very small and the larvae in different stages of development present significant morphological variations, the training views were chosen in such a way that they capture aspects of the larvae in all stages of development, in order to have a satisfactory representation of each category. In all subsequent descriptions, the way in which the data set has been split into training and test sets will be described by the number of specimens in the training set. For n training images per category, the training set contains $4n$ images, the test set contains $50–n$ images for each of the types of larvae and $1562–n$ images of detritus.

A special case is represented by category No. 4 (detritus), since the number of images belonging to this group is significantly larger than the number of available fish larvae images. By choosing on a number of n training images of detritus and the same number of training images of each type of larvae, the introduction of a bias (i.e. favouring the group represented by significantly larger amount of items) during training was avoided. A potential problem is the fact that, with this method, an extremely small proportion of the data in the detritus category is represented in the training set (e.g. 2.6% of all detritus images for n=40). But since the detritus specimens are very different in size and aspect from the larvae images (the majority of them are blobs of a few pixels), it was expected that such a small number of training samples would still be sufficient for good recognition of this distinct category.

The tests conducted with these data sets allowed comparisons to be made with classification that was carried out only on fish larvae body size data (Paul Rankine, personal communication). These results are listed below in Table 10.1. below. It is apparent, how traditional image analysis methods (measuring the geometry of the shapes) fail to register salient characteristics of the specimens.

Table 10.1. Categorisation accuracy (%) of Aberdeen data set, as reported by DA (resubstitution) carried out on body size data

Species	Herring	Sprat	Sandeel
Herring	0.0	70.0	30.0
Sprat	20.0	60.0	20.0
Sandeel	90.0	10.0	0.0

Regarding the number of hidden nodes of the neural networks used during collective and committee machine trials, a hidden layer containing more than 2 neurons was able to produce acceptable network convergence in a limited number of epochs during preliminary trials. In order to avoid the situation where the number of hidden nodes equals the number of output nodes and to introduce a 'bottle-neck' in the structure of the network, forcing it to efficiently encode the data and to generalise from the learnt model, a number of 3 hidden neurons were chosen as a start (in contrast with the tests described in the previous chapter). Additional trials involved hidden layers of 6 and 9 neurons.

The learning parameters for the error backpropagation algorithm in all subsequent neural net-

work trials were set based on the preliminary trials meant to optimise the training process by arriving at a set of parameters that proved to lead to satisfactory and consistent behaviour in a fixed number of epochs. Therefore the learning rate was set to 0.01 and the momentum term to 0.8. The number of epochs during each training run was set to 3000.

Prior to trials that involved rho–theta receptive fields, the size of the receptive fields grids and the standard deviation of the Gaussians had to be decided. It was preferred to set the size of the grids placed on each link tree layer to 4x4 receptive fields, as in previous tests. This choice makes possible comparisons between theta histogram and RF–based tests by not introducing changes in the dimensionality of the involved feature vectors. Preliminary tests were carried out for standard deviations of 0.05, 0.1, 0.15 and 0.2, as in the trials reported in section 8.2.2. (p. 148). Leave–one–out classification performed with DA showed that the highest mean accuracy over 4 categories (64.5%) was obtained for σ=0.15 and σ=0.2. Therefore the standard deviation of the Gaussian receptive fields was set to σ=0.15 in all subsequent trials, this being consistent with the choice made in previous tests reported in chapters 8 and 9.

As in previous experiments that involved synthetic shapes, 20 training/test runs were carried out in each experiment that involved neural network–based categorisers. The output node activations of neural networks trained/tested on individual coarse channels' data were saved during the collective machine trials and used later in committee machine tests.

10.3. Tests with collective machines

In a similar way to the collective machine trials reported in the previous chapter, DA and ANN–based categorisers operating on grouped data channels' feature descriptions were tested on the Aberdeen data set.

10.3.1. Discriminant analyses

Individual and grouped channels' feature data has been submitted to DA–based categorisers. The mean classification accuracy over the 4 categories was calculated for each case of training set size and channel configuration.

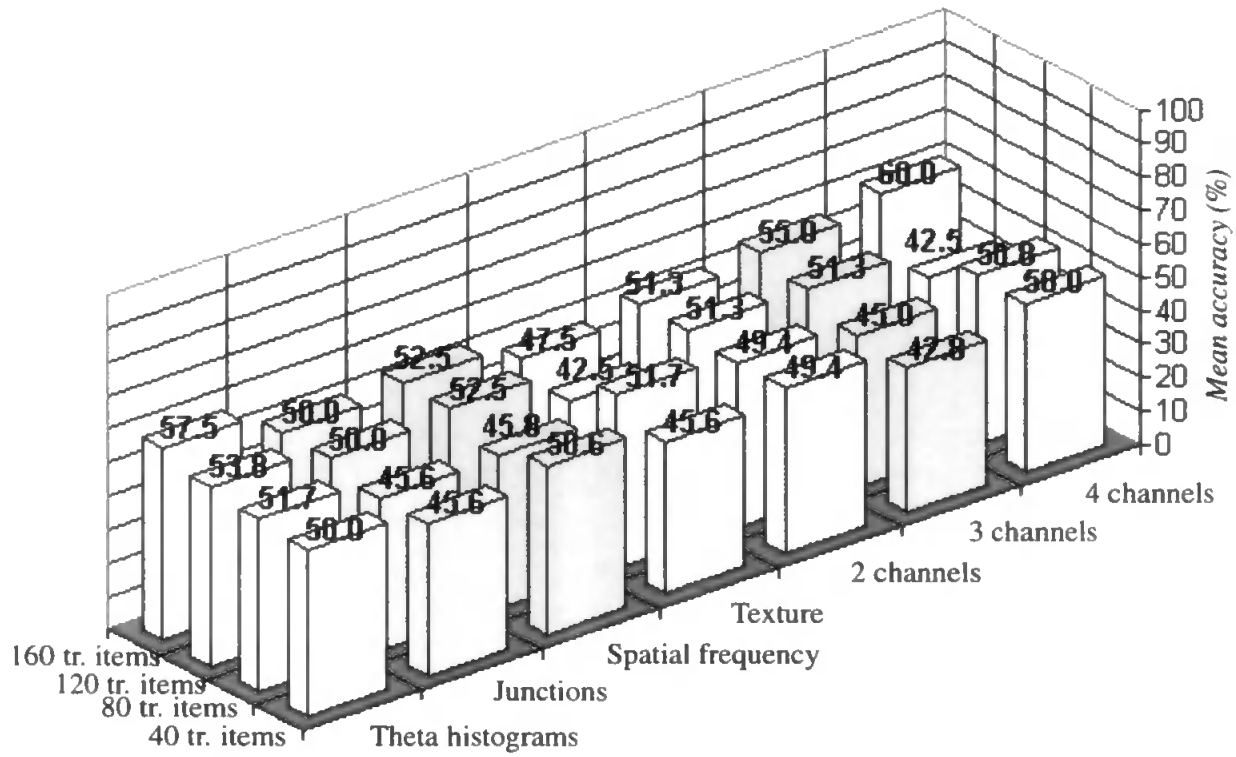


Fig. 10.1. Mean DA Aberdeen test set classification accuracies for all channel configurations and training set sizes. The scale–space channel employed theta histograms.

These mean accuracies are represented in Fig. 10.1. above for the case of theta histogram–based scale–space channel and associated 'what' channels.

In a similar manner, the mean performance for the RF–based scale–space channel and associated 'what' channels is plotted in Fig. 10.2.

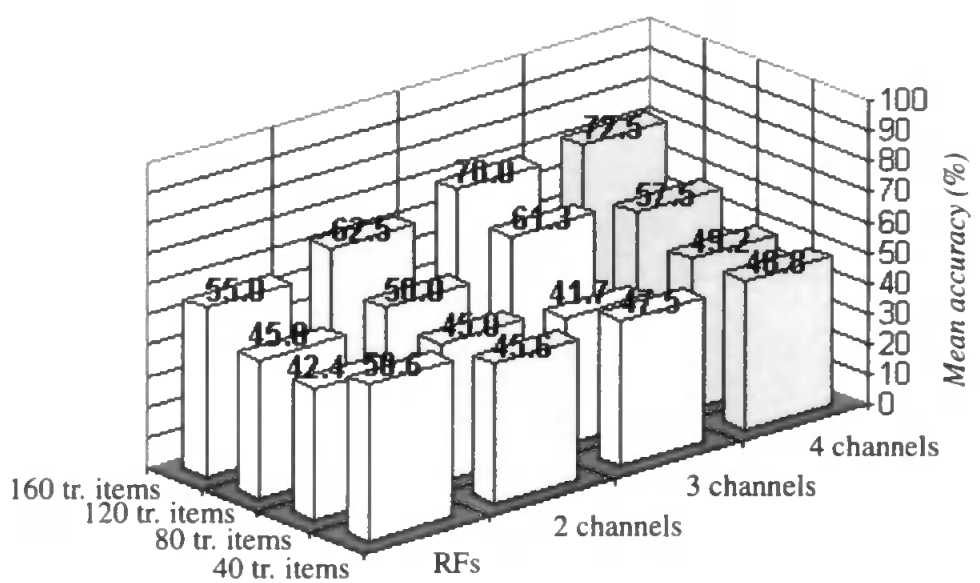


Fig. 10.2. Mean DA Aberdeen test set classification accuracies for all training set sizes, in the case of RF–based scale space channel and associated 'what' channels.

In both cases, the grouping of channels improves performance, this increase appearing to be more significant in the case of trials that involve the RF–based scale–space channel. It also becomes apparent, that the size of the training set has a significant effect on performance – due to the particularities of this data set, such an effect is expected. ANN trials, as it will be described in the next section, proved the presence of this effect.

The overall kappas were calculated from the confusion tables obtained in these tests. As it will become apparent later, the detritus category is identified with significantly higher accuracy than the species of larvae. Therefore the kappas give a better description of the system's performance across categories, than the simple average of classification accuracies does. The overall coefficients of agreement in the case of these experiments actually measure the agreement between the system and the expert taxonomists who labelled the images of the data set.

The kappas for both scale–space channel variants are listed in Table 10.2., for all channel configurations.

Table 10.2. Overall kappa values for DA test set results (5–object data set)

Training items	Theta/ RF	Junctions	Spatial freq.	Texture	2 channels	3 channels	4 channels
40	0.60 (0.61)	0.61	0.64	0.60	0.60 (0.58)	0.47 (0.62)	0.62 (0.63)
80	0.64 (0.53)	0.61	0.61	0.64	0.60 (0.58)	0.57 (0.58)	0.62 (0.63)
120	0.67 (0.60)	0.65	0.67	0.60	0.65 (0.62)	0.64 (0.73)	0.56 (0.70)
160	0.71 (0.70)	0.66	0.68	0.64	0.66 (0.74)	0.67 (0.79)	0.69 (0.81)

All kappas values were obtained with high significance, the lowest z observed was 30.78 (training set of 120 images, 4 channels, RF–based scale–space channel). These kappas confirm the observations made previously on synthetic data and the effect of training set size and channel grouping on performance.

The classification accuracies observed for each category are plotted in Fig. 10.3., in the case of the data set split that led to the best results (training set of 40 specimens per category).

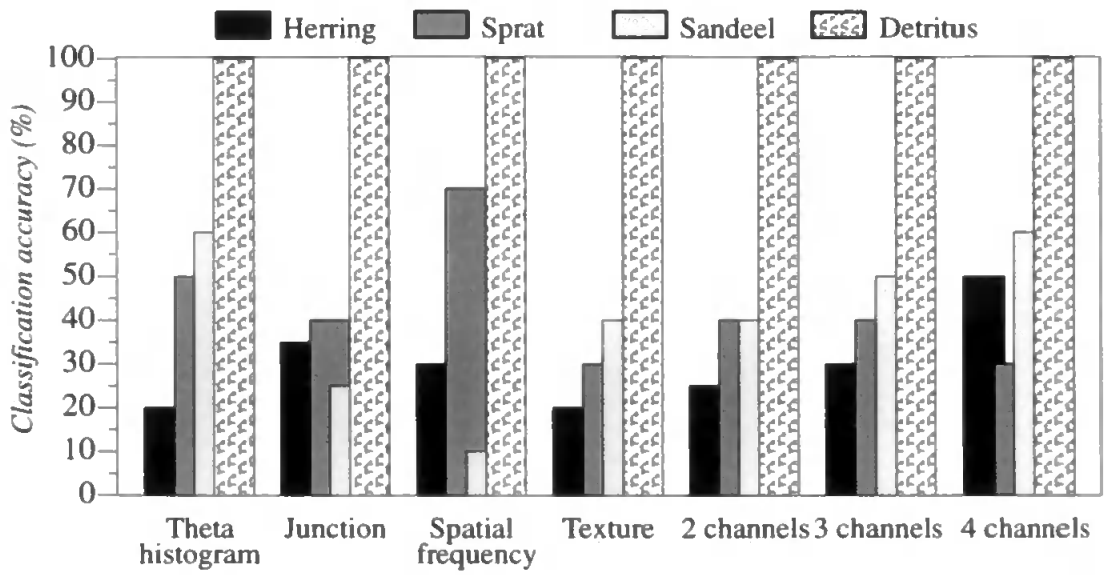


Fig. 10.3. The DA test set classification accuracies observed for each of the categories. The training set contained 40 images per category.

It is apparent, that detritus is recognised with at least 30% better accuracy than the larvae species, even when the training set constitutes less than 3% of all detritus images. As it was expected, due to the very different nature of these objects, they are correctly identified based on very few learnt instances.

The same measures of performance were evaluated for the RF–based scale–space channel and 'what' channels associated with it. These category–specific accuracies are represented in Fig. 10.4. below.

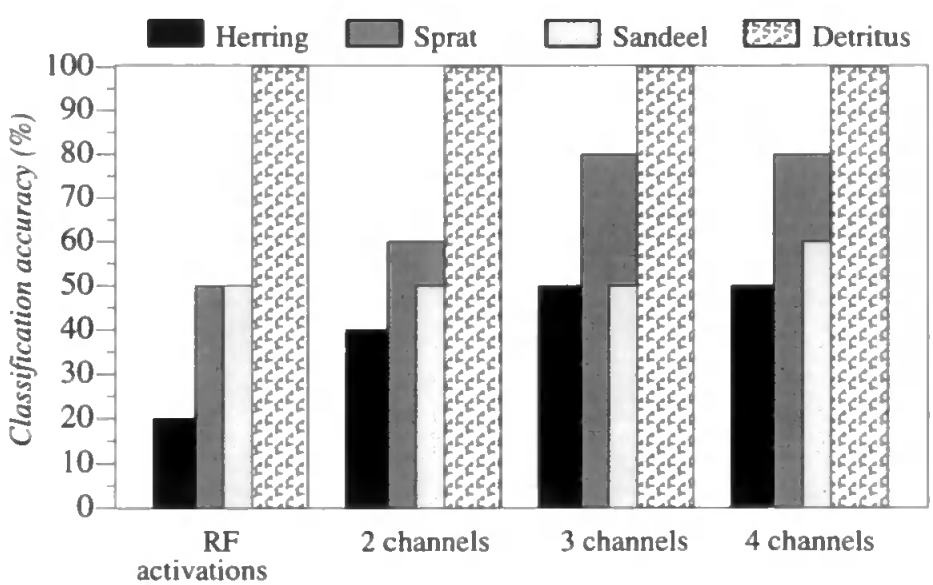


Fig. 10.4. The DA test set classification accuracies obtained for each category. The scale–space channel used RF activations, training set contained 40 images per category.

In the latter case, the fish larvae images are significantly better identified than in tests that used the theta histogram–based scale–space channel. The most marked improvement can be observed in the sprat category (around 20% for 4 channels). This is an important aspect, since it seems to prove that the rho–theta receptive fields activation patterns, although being a stricter shape descriptor than theta histograms, are still coarse enough to provide the system with salient object descriptions in conditions of non–rigid shapes.

The inter–category confusion has been assessed following the method introduced in the previous chapter. For the situation in which all 4 channels are used and the training set contains 40 images per category (this led to the best results), the amount of inter–category confusion is listed in Table 10.3. for both theta histogram and RF–base scale space channels.

Table 10.3. Mean inter–category confusion in DA trials using all 4 channels' data (%). Training set size was 160 images.

	Theta histogram + 3 channels				Receptive fields + 3 channels			
Category	**Herring**	**Sprat**	**Sandeel**	**Detritus**	**Herring**	**Sprat**	**Sandeel**	**Detritus**
Herring	–	60.0	30.0	30.0	–	40.0	70.0	0.0
Sprat	.	–	0.0	20.0	.	–	0.0	0.0
Sandeel	.	.	–	20.0	.	.	–	0.0
Detritus	.	.	.	–	.	.	.	–

It is evident, that the herring is the most confused with other categories, while sprat and sandeel are significantly less confused between each other. The detritus is the least mistaken for a larvae species. Still, the DA classifier has incorrectly identified many of the larvae as detritus, when the theta histogram–based scale–space channel was used.

10.3.2. Neural network trials

Following the already established protocol, the data sets used in DA trials were used in tests that involved ANN–based collective machines.

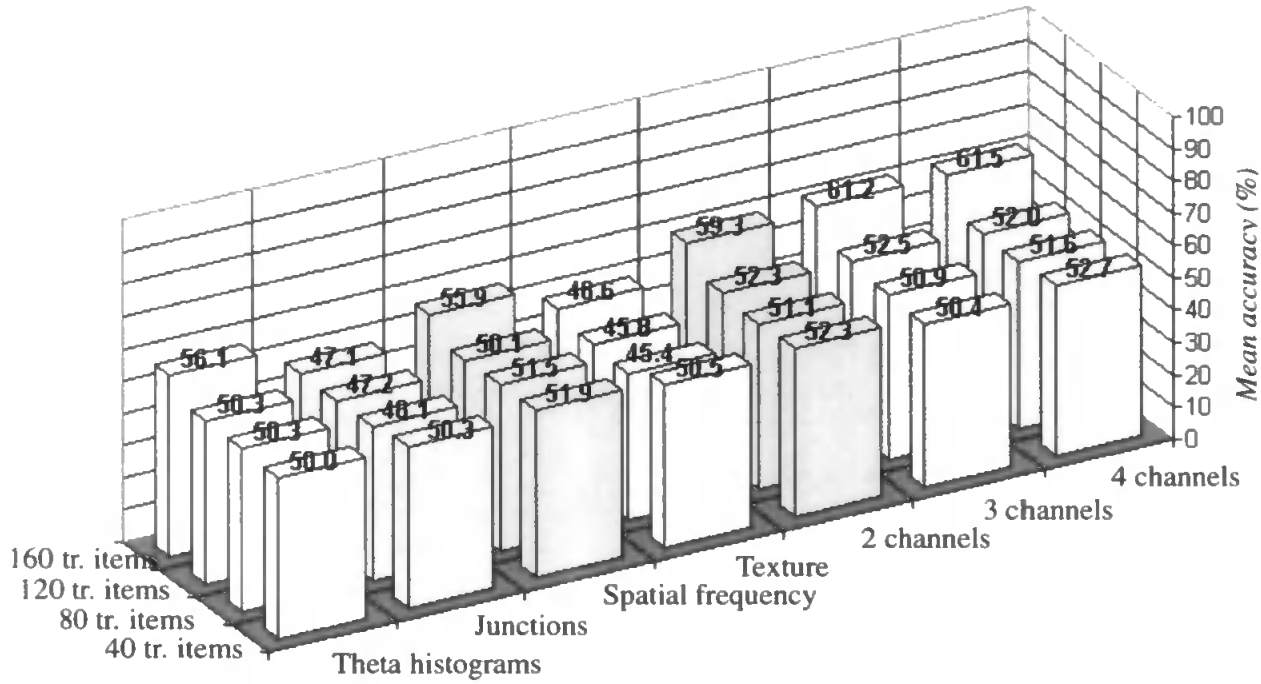


Fig. 10.5. Mean ANN test set classification accuracies for all sets of 20 trials (Aberdeen data set, 6 hidden nodes).

The mean classification accuracy has been evaluated from the 20 training/testing runs performed for each channel configuration and training set size. These means are reported in Fig. 10.5. for the case where the hidden layer of the networks consisted of 6 nodes.

The same measures of performance were evaluated from the tests that employed the rho–theta RF–based scale–space channel. The mean classification accuracies for this case are plotted in Fig. 10.5. below. Both sets of results for theta and RF–based scale–space channel show an increase in performance with the increase of the training set size and the number of grouped channels, the latter tendency being more clear in the case of the RF–based tests.

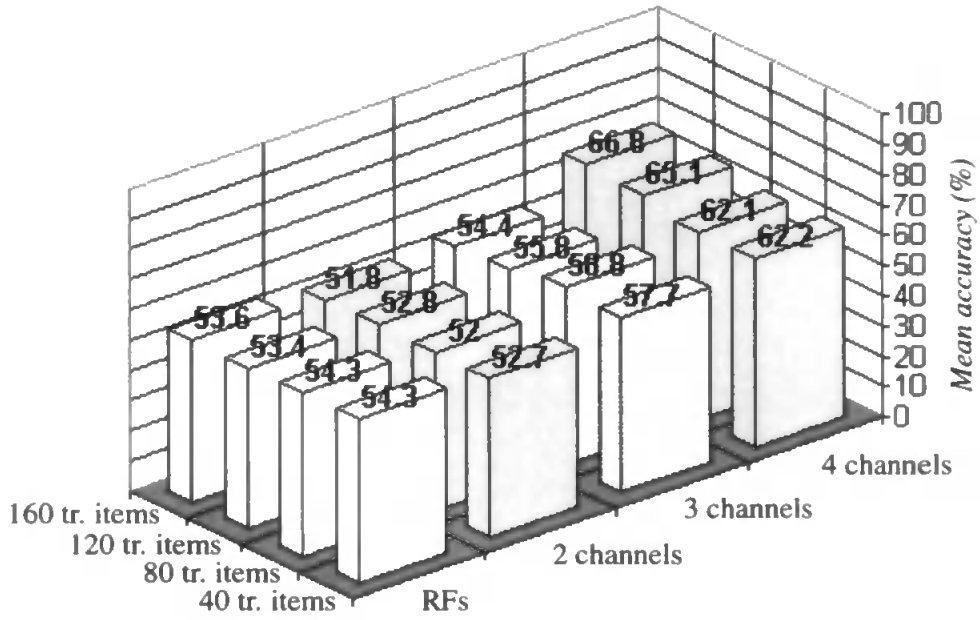


Fig. 10.6. Mean ANN test set classification accuracies for all sets of 20 network runs, using RF–based scale–space channel (Aberdeen data set, 6 hidden nodes).

Whether this increase is due to the presumed factors (i.e. training set size and channel configuration) had to be investigated. As in the experiments carried out on the computer–generated data set, one–way ANOVAs were used to study the effect of these factors. The classification accuracies obtained in the sets of 20 runs performed for each configuration of grouped channels were grouped according to the size of the training set and submitted to ANOVA. The results from trials that used networks with 6 hidden nodes were utilised. The F and p values obtained for data coming from tests that employed theta or RF–based scale–space channel are listed in Table 10.4.

Table 10.4. Effect of changes in training set size on ANN–based collective machine with 6 hidden nodes

F(3,60) = 2.76 ; F(3,120) = 2.68 (α=0.05)

Nr. of channels	F(3,76)	p	Effect
theta + 1	21.29	4.13E–10	very strong
theta + 2	29.49	9.43E–13	– " –
theta + 3	34.70	3.12E–14	– " –
RF + 1	18.03	4.92E–09	– " –
RF + 2	38.75	2.66E–15	– " –
RF + 3	59.52	< 2E–308	extremely strong

These results prove that the tendencies observed in the performance measures are due to changes in the amount of learnt data.

The effect of the number of grouped channels has been studied in a similar manner, the results of the 20 runs being grouped according to the levels of the channel factor. The results from tests that used a mid–range training set size (120) were used. For 6 hidden nodes and a training set of 160 items, the one–way ANOVA results were as it follows.

For 4 factor levels (theta histograms, 2, 3, 4 channels) the outcome of the ANOVA was an $F(3,76)=6.61$, $p = 4.95E{-}04$ (significant). For the more realistic case where the collective is composed of at least 2 channels, the results were $F(2,57)=0.008$, $p = 0.92$. With $F(2,40)=3.23$ and $F(2,60)=3.15$, these results show the presence of an effect only when comparing the accuracies obtained with standalone theta histogram channel and associated channels.

In the case of the trials that used the RF–based scale–space channel, the ANOVA results were $F(3.76)=4.27$, $p = 0.007$ when 4 ANOVA groups were investigated (RF channel, 2, 3, 4 channels). When only the 2 or more grouped channels' case was analysed, the outcome was $F(2,57)=6.81$, $p=0.004$. In both cases, the results show the presence of the channel effect.

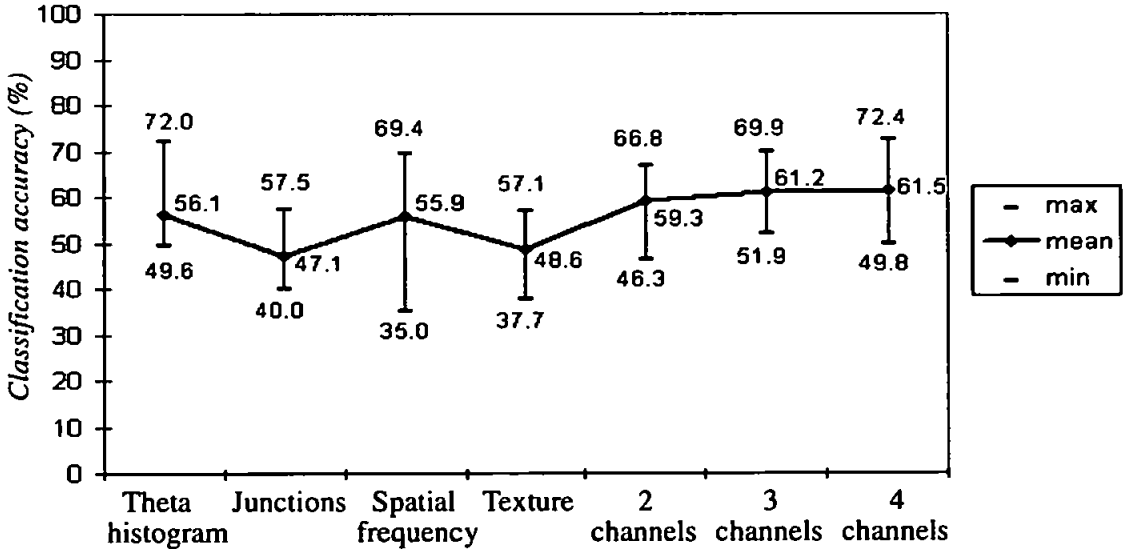


Fig. 10.7. Lowest, mean and highest test set classification accuracies obtained in ANN trials (6 hidden nodes, 160 items in training set).

The lowest, mean and highest classification accuracies obtained during 20 runs were plotted for the case when 6 hidden nodes were used in the networks and the training set contained 160 speci-

mens. Fig. 10.7. above shows these accuracies for the test that used theta histogram–based scale–space channel.

In a similar manner, Fig. 10.8. shows the accuracies obtained in the runs that involved the 'where' channel based on rho–theta receptive fields.

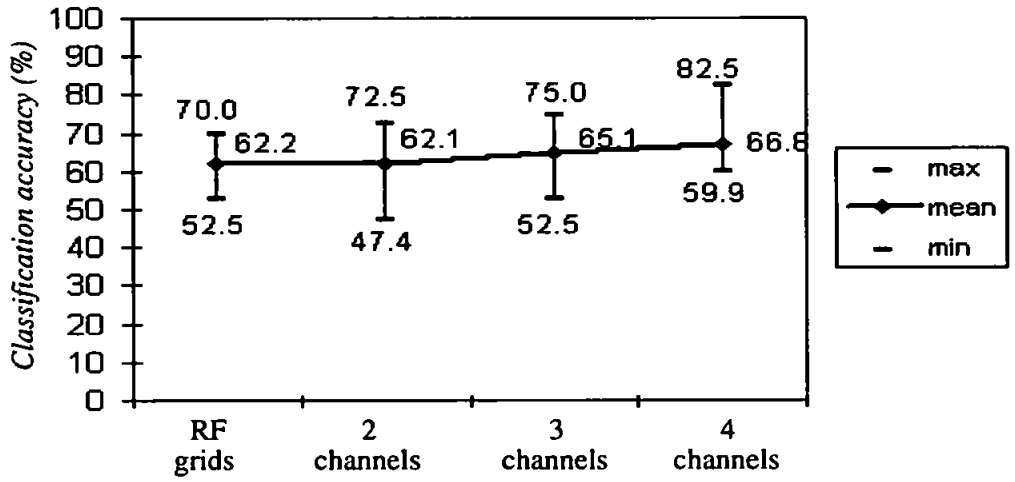


Fig. 10.8. Lowest, mean and highest test set classification accuracies obtained in ANN trials that employed the RF–based scale–space channel (6 hidden nodes, 160 items in training set).

The overall kappa scores are listed in Table 10.5. for tests that involved the theta and RF–based scale–space channel. The kappas given in brackets are the ones obtained in the latter case.

Table 10.5. Mean overall kappas from 20 test runs, for each of the channel configurations and data set splits (6 hidden nodes)

Training items	Theta/ RF	Junction	Spatial freq.	Texture	2 channels	3 channels	4 channels
40	0.38 (0.68)	0.62	0.54	0.49	0.48 (0.68)	0.50 (0.67)	0.55 (0.66)
80	0.49 (0.67)	0.63	0.51	0.51	0.57 (0.66)	0.57 (0.67)	0.58 (0.66)
120	0.59 (0.70)	0.63	0.58	0.59	0.63 (0.71)	0.62 (0.68)	0.63 (0.68)
160	0.59 (0.71)	0.65	0.54	0.64	0.63 (0.72)	0.61 (0.74)	0.62 (0.74)

The lowest significance of kappa (z=13.14) was observed during the trials that used theta histo-

grams and a training set of 40 items. The lowest kappa was 0.16 (theta histogram channel only, 40 training items), the highest was 0.77 (4 channels, 120 training items).

Evidently, in all the measures of mean performance, the detritus category's high recognition rate (as it becomes apparent from the following results) introduces a stable bias. This relates to the similar aspect of the 5–object tests, where the 5th category was meant to be very different, therefore highly recognisable. Hence the full picture on the system's behaviour can only be obtained by studying the way in which each category is classified in various circumstances (i.e. channel configurations).

The classification accuracies for each category has been averaged over 20 runs that used networks with 6 hidden nodes and a training set of 160 specimens. These are reported in the diagram shown in Fig. 10.9. in the case of the theta histogram–based tests.

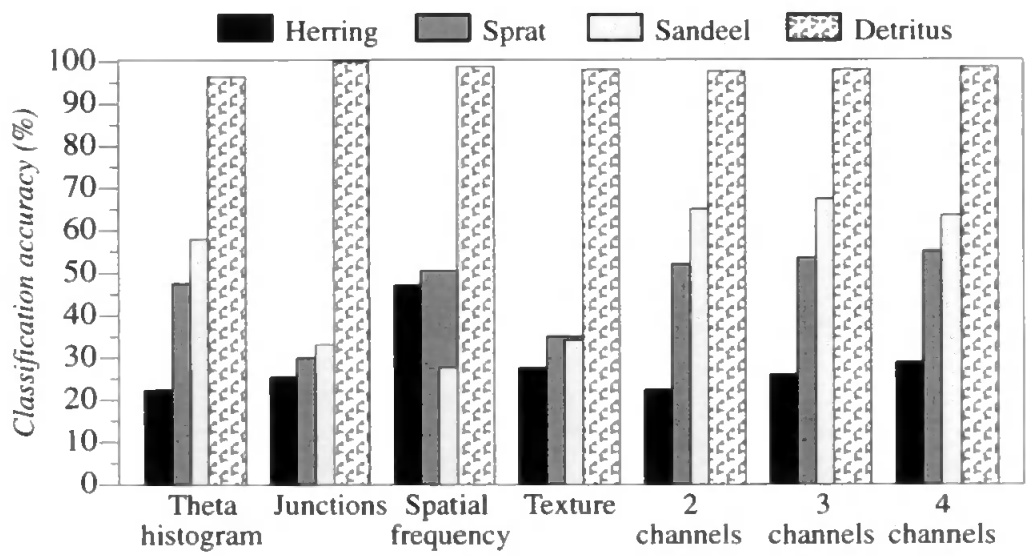


Fig. 10.9. Test set classification accuracies observed for each category of the Aberdeen data set, averaged over 20 runs (6 hidden nodes, training set of 160 specimens).

Besides the means reported in Fig. 10.9., the best classification accuracies were obtained in one of the 20 runs carried out with 4 channels and a training set of 160 items: for this best run, 50% of herring, 60% of sprat, 80% of sandeel larvae and 99.9% of detritus were identified correctly.

In a similar manner, the category–specific performance has been evaluated for the RF–based trials, too, in the same conditions of hidden layer and training set size. These results are shown in Fig. 10.10.

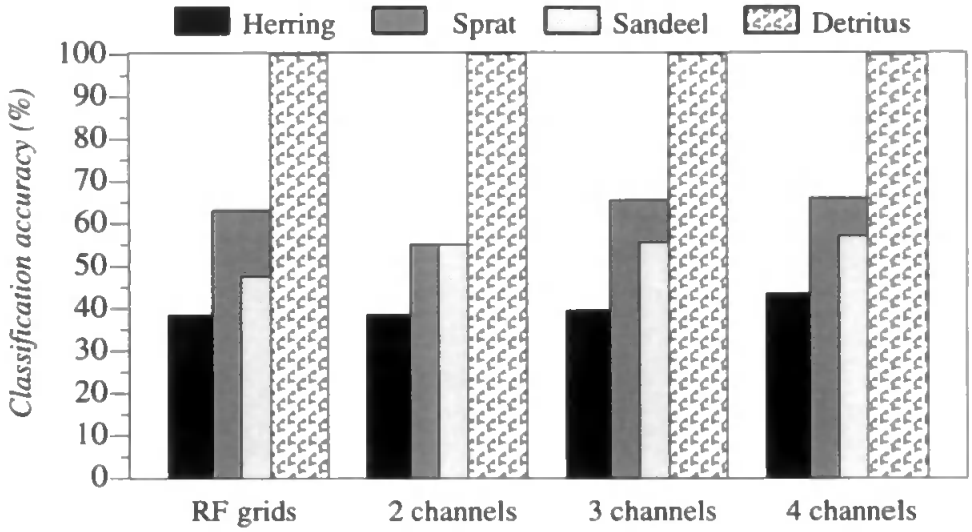


Fig. 10.10. Test set classification accuracies observed for each category of the Aberdeen data set, averaged over 20 runs that used RF–based scale–space channel (6 hidden nodes, training set of 160 specimens).

The best performance was obtained in a run performed with 4 channels and training set of 160 items: 70% of herring, 90% of sprat, 70% of sandeel larvae and 100% of detritus were classified correctly.

It is apparent, that the detritus is identified with >95% accuracy in all cases, independent of the channel configuration. Also, herring larvae seem to be the most difficult category to recognise. The amount of inter–category confusion averaged the 20 runs with 6 hidden nodes and training set of 160 specimens shows this eloquently, as it can be seen in Table 10.6.

Table 10.6. Mean inter–category confusion in ANN trials using all 4 channels' data (%). Training set contained 40 items per category.

	Theta histogram + 3 channels				Receptive fields + 3 channels			
Category	**Herring**	**Sprat**	**Sandeel**	**Detritus**	**Herring**	**Sprat**	**Sandeel**	**Detritus**
Herring	–	64.0	63.0	0.3	–	49.0	54.5	2.0
Sprat	.	–	24.5	1.0	.	–	28.0	0.1
Sandeel	.	.	–	1.1	.	.	–	0.1
Detritus	.	.	.	–	.	.	.	–

Herring is the most confused with other species of larvae, while sprat and sandeel are significantly

less confused with each other. The 4th category clearly stands out with occasional images being mistaken for detritus.

The effect of hidden layer size on the performance has been studied with one–way ANOVAs. The results from the tests carried out with 2, 3 and 4 channels, involving theta histogram or RF–based scale–space channels were submitted to analysis of variance. In none of the cases did ANOVA show the presence of hidden node effect. The highest, still insignificant result was $F(2,57)=2.60$ with $p = 0.08$ (theta histogram–based 'where' channel + junction channel). Hence for the used hidden layers of 3, 6, 9 nodes, the generalisation ability of the categorisers does not show significant change.

10.4. Tests with committee machines

The experiments involving committee machines, reported in the previous chapter, were repeated for the Aberdeen data set. As in the case of the computer–generated 5–object image data set, the two types of committee machines allowed the study of the discriminatory power of individual channels. Although for comparative reasons, the collective machine trials did involve training/ testing of networks using individual channels' data, these channel categorisers did not co–operate in any way. The output node activations of these categorisers were stored and used in the committee decision making procedure.

Following the notations introduced in the previous chapter, the committees that take a decision based on the most confident member will be referred to as CMC and the ones that consider the sum of the corresponding output neurons of each member will be called CSO.

The mean classification accuracies obtained in 20 runs with the CMC machine (6 hidden nodes employed in each committee member) are shown below in Fig. 10.11., for all cases of training set size and number of channels. There has been no change in performance when the texture channel was added to the committee.

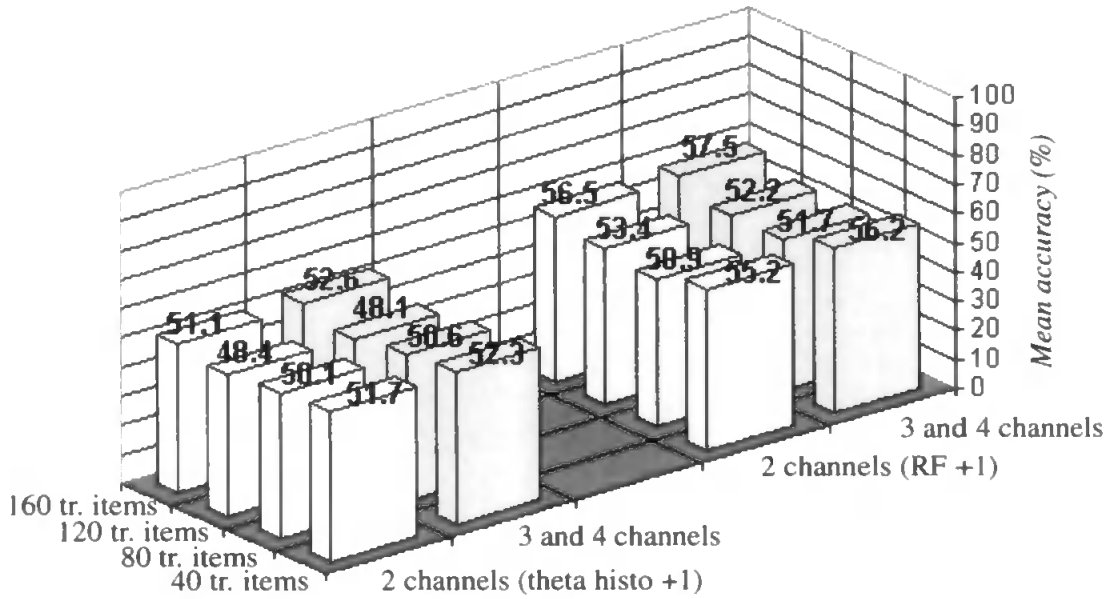


Fig. 10.11. Mean test set classification accuracies obtained in 20 runs carried out with CMC machine.

Regarding the number of correct committee decisions imposed by the most confident channels in tests, this was evaluated for each committee member, as a mean of number of correct decisions in the four cases of training set size. The texture channel had no contribution to the correct committee decisions, as it was mentioned before. Unlike the CMC machine tests carried out on the 5–object data set, in this case there is a prominent difference between the contribution of channels, depending on which scale–space channel architecture was used. When the theta histograms were used and only two members were in the committee (theta + junctions), the scale–space channel imposed on average 46.1% of the correct decisions of the committee. When the RF–based scale–space channel was used, this voted correctly in 66.7% of the correct decisions taken by the committee.

When 3 or 4 channels were used, the theta histogram channel was responsible on average for 31% of the correct decisions taken by the committee, the junction channel for 38.7% and the spatial frequency channel for 30.3%. The RF–based scale–space channel was better, leading to 52.5% of the correct decisions of the committee, the junction channel in this case provided 27.5% of the correct decisions and the spatial frequency 20%.

These results show that the RF–based channel was the most confident in the majority of the correctly classified cases, while the theta histogram–based scale–space channel was outranked by the junction channel.

Fig. 10.12. shows the mean classification accuracies obtained in tests that involved the CSO machine, each committee member operated with a hidden layer of 6 nodes.

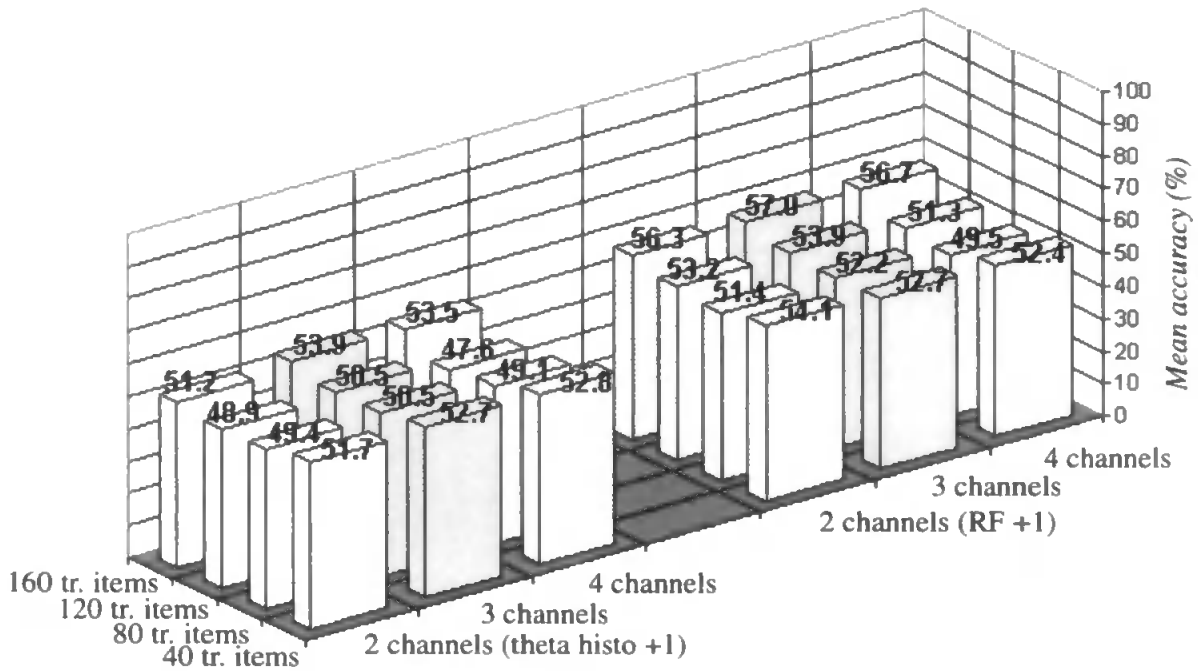


Fig. 10.12. Mean test set classification accuracies obtained in 20 runs conducted with CSO machine.

One–way ANOVA carried out on the results obtained in the above tests showed a variable effect of training set size on performance. Table 10.7. lists the obtained F and *p* values for both committee machine and both scale–space channel architectures.

Table 10.7. Effect of changes in training set size on committee machines			
F(3,60)= 2.76 ; F(3,120)= 2.68 (α=0.05)			
CMC machine			
Nr. of channels	**F(3,76)**	***p***	**Effect**
theta + 1 **RF + 1**	2.14 8.01	0.1 1.04E–04	not present significant
theta + 2,3 **RF + 2,3**	4.73 11.55	0.004 2.55E–06	significant very strong
CSO machine			
theta + 1 **RF + 1**	2.12 4.48	0.1 0.003	not present strong
theta + 2 **RF + 2**	2.53 5.47	0.06 0.002	not present strong
theta + 3 **RF + 3**	7.41 10.78	2.01E–04 5.63E–06	very strong very strong

Significant effect could be observed for all cases in which the RF–based scale–space channel was used. The theta histograms and associated 'what' channels showed the presence of an effect only in the case of 4 channels collaborating in a CSO machine.

As more committee members are added to CMC, no significant effect on performance can be observed, as one–way ANOVAs show. In the case of this committee machine, for theta histogram or RF–based scale–space channel and associated 'what' channels, the results were $F(2,57)=0.16$, $p = 0.85$ (theta histograms + 'what' channels) and $F(2,57)=1.04$, $p = 0.35$ (RFs + 'what' channels). As in the tests conducted on the 5–object data set, the texture channel played no role in improving the performance of the system, other channels being always more confident voters.

In the case of the CSO machine, significant channel effect could only be observed when the RF–base scale–space channel was employed as 'where' channel. The ANOVA results were $F(2,57)=12.4$, $p = 0.25$ (theta histograms + other channels) and $F(2,57)=4.97$, $p = 0.01$ (RFs + other channels).

As in previous experiments, the lowest and highest test set classification accuracies were compared in the case of both committee machines, when they were trained with 40 images per category. These extreme and the mean accuracies averaged over 4 categories are plotted in Fig. 10.13. for the CMC machine.

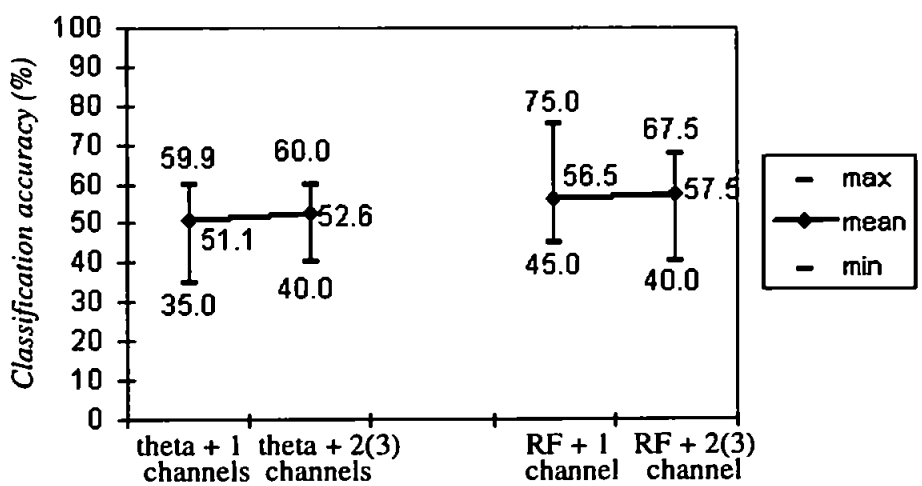


Fig. 10.13. Lowest, mean and highest test set classification accuracies observed in 20 runs performed with the CMC machine. Training set contained 160 specimens.

The same performance measures were assessed in tests that used the CSO machine; these are represented in Fig. 10.14. for the case when the training set contained 40 images per category.

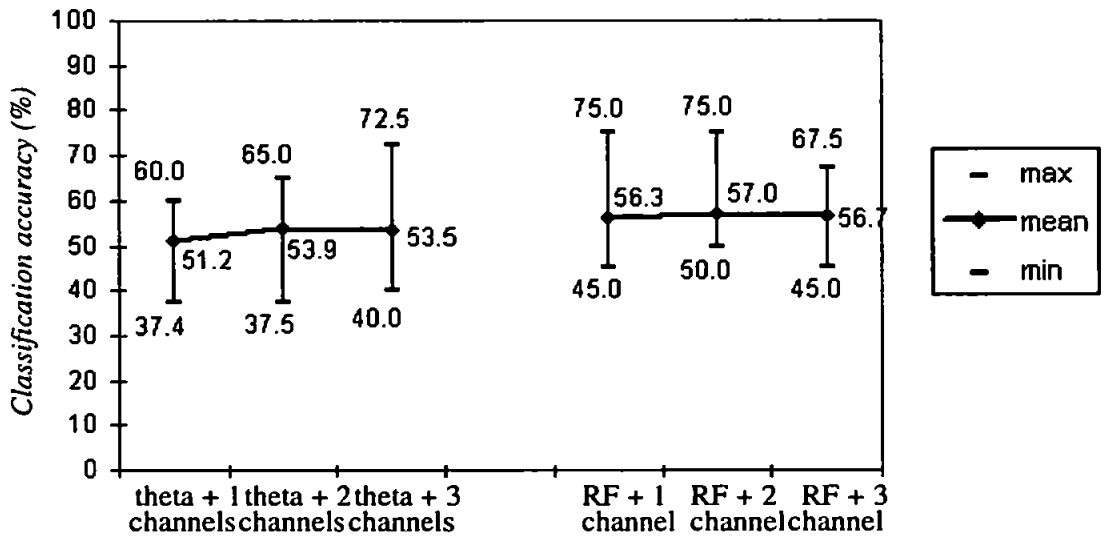


Fig. 10.14.Lowest, mean and highest test set classification accuracies observed in 20 runs performed with the CSO machine. Training set contained 160 specimens.

The overall cross–category kappas are listed in Table 10.8. below. No significant change in these performance measures can be observed as more members are added to the committees, which is consistent with the ANOVA results.

Table 10.8. Mean overall kappas calculated from 20 runs in the case of two committee machines. Kappas in brackets are the RF+1,2,3 channels results.

	CMC machine		CSO machine		
Training set size	2 channels	3 & 4 channels	2 channels	3 channels	4 channels
40	0.64 (0.69)	0.66 (0.69)	0.64 (0.69)	0.67 (0.70)	0.67 (0.70)
80	0.63 (0.68)	0.63 (0.66)	0.63 (0.67)	0.65 (0.68)	0.64 (0.66)
120	0.65 (0.67)	0.65 (0.67)	0.64 (0.67)	0.66 (0.67)	0.65 (0.65)
160	0.66 (0.70)	0.66 (0.70)	0.66 (0.69)	0.67 (0.68)	0.67 (0.68)

The category–specific test set classification accuracies (averaged over 20 runs), obtained from

tests that used the CMC machine are shown in Fig. 10.15. An improvement can be observed in the tests that replaced theta histograms with rho–theta receptive fields.

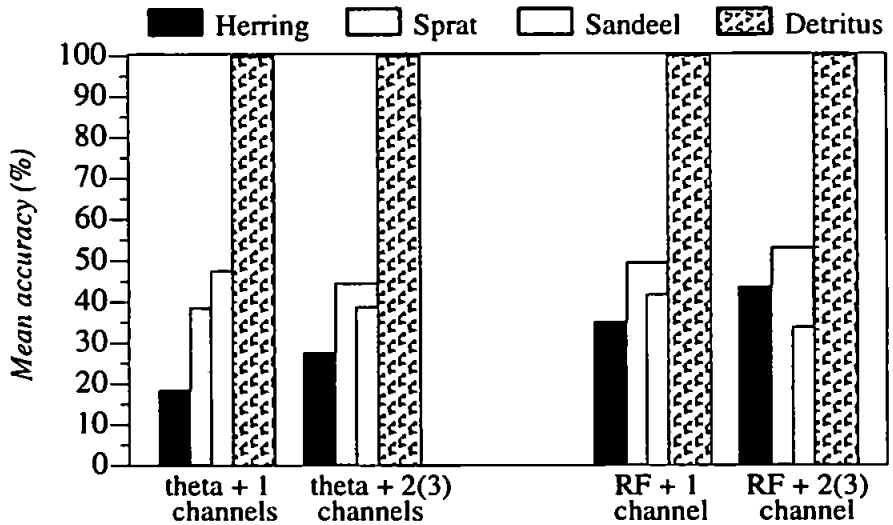


Fig. 10.15. Mean category–specific test set classification accuracies obtained in 20 trials carried out with the CMC machine. Training set contained 160 specimens.

The best classification accuracies during tests that used theta histograms were observed for 3 channels and a training set of 160 specimens: 30% of herring, 50% of sprat, 60% of sandeel larvae, 100% of detritus were correctly identified when theta histogram–based scale–space channel was used. With RF–based 'where' channel, the best result was obtained in a run carried out with 2 channels and a system trained with 160 specimens: 70%, 80%, 50% and 100% respectively.

The same category–by–category representation of test set classification accuracies averaged over 20 test runs are shown below in Fig. 10.16. for the CSO machine.

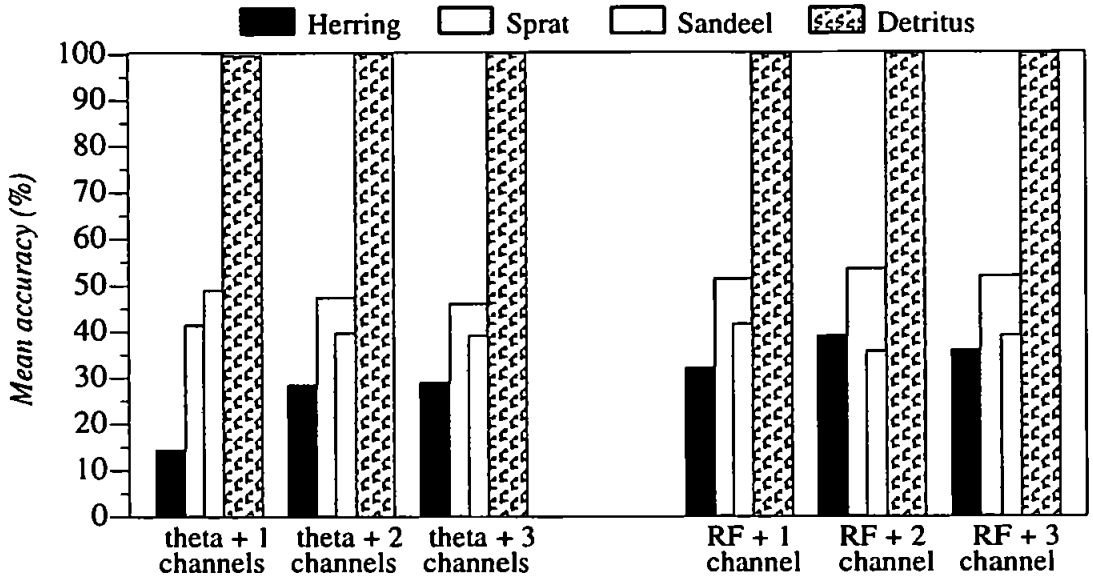


Fig. 10.16.Mean category–specific test set classification accuracies obtained in 20 trials carried out with the CSO machine. Training set contained 160 specimens.

The best results in tests that used the CSO machine and theta histograms were observed in one of the runs when a training set of 160 items was employed. For theta histograms + 3 channels the best classification accuracies for herring, sprat, sandeel larvae and detritus were 60%, 70%, 60% and 100%, respectively. When the RF–based scale–space channel was used in the CSO machine, the best results were obtained in one of the 20 runs performed with RF + 2 channels and a system trained with 160 specimens: 60%, 80%, 60% and 100%.

In both cases of committee machine architectures, the detritus –as in the collective machine trials– is easily recognised. Herring is still the least correctly identified category. The mean confusion between categories, calculated from tests that used committee members trained with 40 images per category and committees of 4 members is listed in Table 10.9.

Table 10.9. Mean inter–object confusion in committee machine trials using all 4 channels' data (%). RF–based test results in brackets.

	CMC machine				CSO machine			
Category	**Herring**	**Sprat**	**Sandeel**	**Detritus**	**Herring**	**Sprat**	**Sandeel**	**Detritus**
Herring	.	67.5 (60.0)	62.5 (56.0)	0.1 (0.1)	.	67.5 (64.0)	57.0 (54.0)	0.5 (0.0)
Sprat	–	.	59.5 (53.0)	0.1 (0.0)	–	.	61.0 (54.5)	0.0 (0.0)
Sandeel	–	–	.	0.0 (0.0)	–	–	.	0.0 (0.0)
Detritus	–	–	–	.	–	–	–	.

The mean classification accuracies and the amount of confusion shows how poorly the committees discriminate between larvae species. The only consistently well recognised category is detritus.

No significant effect of hidden layer size could be detected in the tests that involved theta histogram–based scales–space channel. None of the committee machines showed in one–way ANOVAs the presence of such effect, the largest, still insignificant results being $F(2,57)=3.13$, $p=0.08$ for 3(4) channels in CMC machine and $F(2,57)=3.22$, $p=0.08$ for 3 channels in a CSO machine.

When the RF–based scale–space channel was used as one member in the committees, significant hidden layer effect was detected. The lowest still significant results were $F(2,57)=4.51$, $p=0.04$ for CMC machine with 4 channels and $F(2,57)=5.28$, $p=0.02$ for CSO machine with 3 channels.

10.5. Conclusions and discussion

The Aberdeen data set presented the system with a set of problems that were not present in previous tests carried out on synthetic image data. The image quality was poor, the larvae shapes were non–rigid and changed with the developmental stages of the life–forms, the objects to be classified had very similar shapes and these were not geometrically regular. Also, in conjunction with the morphological variations of the larvae, a small amount of data was available, which caused difficulty in selecting training set specimens that would represent well all of the categories.

The system, whether it used collective or committee classifiers, categorised the fish larvae with satisfactory accuracy. The spread of classification accuracies and overall kappa scores observed during the ANN runs showed that the generalisation to novel data was proved to be difficult and it depended on the initial weights of the networks and on the training set. Still, in the best training/test runs, the system was able to classify larvae with above 50% accuracy when using theta histograms in collectives, and above 70% when RF–based scale–space channel was used. Committees achieved an above 60% performance with RFs in CSO machine and above 70% in the case of RFs used in CMC machines. More than 95% of detritus was classified correctly in all cases.

In all tests, herring larvae was the most confused with other categories, especially in tests that involved the theta histogram–based scale space channel. Sprat and sandeel larvae were significantly (on average 20%) more correctly classified than herring larvae. Detritus presented no

problem to any of the tested categorisers, showing that a category of objects that are very different in size and shape are easily recognised by the system, even when an extremely small amount of training samples are available. It would be interesting to compare the inter–species confusion observed in tests with the assessment of human experts' performance, but such data was not available at the time of this report.

The DA–based categoriser that is presented with theta histograms and associated channels commits several noteworthy mistakes by classifying many larvae specimens as detritus. The mean classification accuracy of the system that uses theta histograms as 'where' channel and DA collective machine is 1 to 6% lower than the corresponding ANN classifier's mean performance, depending on the channel configuration. Kappa scores in these tests show that in fact, the agreement between the classifier and the human rater (taxonomist) is better in the DA categoriser's case. In this situation, the kappas should be trusted, since the mean classification accuracy of the ANN classifier is the average of 20 mean accuracies calculated over all categories, while the mean overall kappa is the average of 20 coefficients of agreement that in each run gives a better indication of performance than the mean accuracy. The ANN–based collectives clearly show that generalisation is difficult in the case of this data set and that the performance depends on the initial weights of the networks. Although in all runs, the networks learnt the training data with above 95% accuracy, the test set results are very rhapsodic, in the sense that the variation of the classification accuracy and related performance measures do not have a consistent pattern. This fact is reflected by the swing of classification accuracies (Fig. 10.7.) and of the kappa values observed in the sets of 20 runs. Overall kappas register a variation between 0.16 (which shows very poor agreement) and 0.77 (excellent agreement).

When the RF–based scale space channel is used in collectives, the DA categoriser outperforms ANN by 2 to 15% when the training set is large (160 items). This is confirmed by overall kappa scores. When the system is trained with small training sets ($\leq$ 120 items), the test set classification accuracies oscillate and ANN–based categorisers lead to 4 to 15% better mean classification accuracies than DA collectives. Overall kappas show the same tendency, the ANN results being 0.07 – 0.2 higher than the DA kappas obtained for small training sets. Also, the range of variation of the kappa scores, observed in sets of 20 runs is only 0.54 ... 0.85 in the case of the tests that involved the RF–based scale–space channel. The results are more consistent for these tests where RF–based 'where' channel is utilised and they show that ANNs build a better general model from

small amounts of training data than DA does.

This leads to the discussion of theta histograms vs. rho–theta receptive fields as scale space shape descriptors. RFs proved to be better descriptors in the tests conducted on computer–generated data. On such noisy and difficult data like the Aberdeen set, RFs consistently lead to better classification performance in all tests (collectives and committees). This superiority is apparent from mean classification accuracies, overall kappas, category–specific accuracies. Furthermore, as it will be discussed below, theta histograms lead to inconsistencies in effect studies that investigate channel configuration and training set size as factors. It is notable, that in CMC machine trials, the theta histogram–based channel is taken over by junction and/or spatial frequency channels. This could be explained by the nature of the data, but the consistent supremacy of the RF–based scale–space channel in these CMC tests show that the configuration of wavelet local maxima is still the descriptor with better discriminatory power. Although RF grid activation patterns have coarser theta resolution and are stricter shape descriptors than theta histograms, it seems that even in the case of such flexible and very similar shapes, they are more reliable. They not only add scale–space distance information to the coarse representation, but contrary to theta histograms, the RF activation patterns are fuzzy descriptors. Having witnessed the superiority of the rho–theta receptive field–based scale–space channel over theta histograms in tests that used two so different image sets, it can be concluded that the RF–based 'where' channel' leads to better, more consistent system performance.

Training set size was expected to have a very significant effect on the system's ability to generalise to novel views, due to the nature of the data set. The ANOVAs carried out on collective machine results showed a consistently strong effect of this factor on performance. In the case of committees, this effect proved to be present in all tests performed with the RF–based scale–space channel, while in trials that involved theta histograms the training set's size affected significantly only when 3 or more channels were used. For 2 channels, the theta histogram–based CMC and CSO test results are so rhapsodic, that no significant change in performance can be attributed by ANOVA to the effect of training set size.

Channel configuration was also expected to be a factor in the experiments, based on previous results and the philosophy of multiple coarse data channels. Significant channel effect was consistently detected by ANOVA only in collective machine tests that involved the RF–based 'where' channel. The ANN–based collectives did not show significant improvement when theta histo-

grams were associated with 'what' channels. In the CSO machines' results, channel effect could only be observed when RFs were used as 'where' channel. CMC machines did not show any significant channel effect in any of the tests, indeed, the mean classification accuracies and overall kappas showed negligible or no change as more channels were added. This behaviour is very likely due to the difficulties presented by the data set to the system. Stand–alone channels, especially when theta histograms are present in the committees, lead to satisfactory, but rhapsodic performance and no consistent increase can be observed as more channels are added to the committee. As in the case of the 5–object data set, a channel with weaker discriminatory power can vote more confidently and erroneously, affecting the committee's overall performance. In the case of such a difficult data set, this aspect can have even more significant effect.

Neural network–based collectives proved to be clearly superior to committees. Mean classification accuracies in collectives employing theta histograms are about 8% higher than CMC/CSO machines and about 5% higher when RFs are used. Overall kappas confirm this. The difference is less convincing in tests on systems trained with small amount of data, results becoming more rhapsodic. For 40 items in the training set, CMC machine actually is better with about 4% than collectives when using theta histograms. Not surprisingly, the inter–category confusion is higher in the case of committee machines. When comparing CMC and CSO machines, the mean classification accuracies, inter–category confusions and overall kappas don't show noticeable difference between them. The fact that the addition of new channels had positive effect only on the CSO machine and that the lowest–highest classification accuracies observed in the 20 network runs (Fig. 10.14.) are better than the ones obtained with CMC machine, lead to the conclusion that in these experiments, too, the CSO machine yields better performance than the competitive CMC. It proves that any collaboration between channels, whether it is in the form of a basic summation of committee members' corresponding outputs or a concatenation of the channel data in a collective machine, leads to better performance and system behaviour.

CMC trials showed that 'where' channels can be overtaken in decisions by 'what' channels. When the scale–space channel is a reliable descriptor, like in the tests conducted with RFs, the 'where' channel contributes to the majority of the correct decisions of the committee. From CMC machine tests and mean performances measure in individual channel categorisers trained/tested for comparisons during collective machine trials, it can be observed that the 'where' channels are at least second best, junction and spatial frequency channels having similar discriminatory

powers, while the texture channel is the weakest. It only contributes to the system's performance in collectives and CSO machines, where its output is combined with other channels' data, but plays no role in the competitive CMC committees.

At this point, having summarised several aspects of the system's behaviour, a brief return to the matter of the two different scale space channel architectures is due. It is interesting to remark, that while the 'what' information presented to categorisers operating with more than one channels was exactly the same in theta histogram and RF–based tests, the behaviour of the system changed in a noticeable way with the substitution of theta histograms with RF activations. With theta histograms being used as 'where' channel, the categorisers (collectives and committees) produced much more rhapsodic results than in the cases where RF activations were used. This becomes apparent from the pattern of variation of mean performance with training set sizes, channel configurations, ANOVA effect studies, CMC machine channel rankings summarised above. It seems that as it was intended, the nature of the data provided by the scale–space channel, i.e. the coarse coded scale–space distribution of potentially relevant shape features plays the major role in defining the way in which the system builds a model of the training input and generalises from it to novel data. In the case of this noisy, difficult data set, theta histograms could not provide sufficiently salient description of the analysed shapes and the 'what' channels, even when associated with it led to satisfactory, but quite rhapsodic system behaviour.

The size of the hidden layers employed in the neural networks did not appear to have a significant effect on overall performance. In all ANN–based tests, 1–2% increase in the mean classification accuracy was observed. ANOVA results quoted in previous sections showed that the variations in performance could be attributed to the presence of a hidden node effect only in the case of committee machines where a channel was the RF–based scale space descriptor. Contrary to the theta histogram–based tests that led to rhapsodic learning and generalisation, RF–based results followed considerably more consistent patterns in all experiments, therefore it seems that the latter ANOVA results can be considered to be a true indication of the presence of an effect. Stand–alone channels being trained and tested with low dimensional data in comparison with the collectives presented with concatenated channel data, the number of hidden nodes is indeed expected to affect generalisation, as it happened in the case of committees trained/tested with the 5–object data set. Since a classic cause of overfitting is the use of a large hidden layer in conjunction with small input dimensionality, these categorisers presented with single–channel data would be the most

susceptible to this phenomenon when the hidden layer becomes large. But for the reasonable and practical number of hidden nodes employed in these tests, overfitting was not observed. For 9 hidden nodes, the mean performance on the test set did not decrease compared to the one obtained for smaller hidden layer sizes.

In all tests, the performance of the system was significantly higher than the results obtained with body size data (Table 10.1.). The best classification accuracies obtained in the experiments were above 70%: 70% of herring, 90% of sprat, 70% of sandeel larvae and 100% of detritus were correctly classified by the system.

10.6. Summary

This chapter described the classification experiments carried out on a set of photomicrographs of fish larvae. The problems raised by this data set and the objectives of these experiments have been presented, together with the way in which the trials were set up. The tests performed on collective and committee classifiers and the results were described. The classification performance of the system, its variation with factors like training set size, channel configuration and hidden layer size were studied. The final section of this chapter discussed the results and draws conclusions on the behaviour of the system, the discriminatory power of data channels and compared these aspects with the results obtained in tests that used the 5–object computer–generated data set.

Chapter 11. Conclusions and future work

11.1. Introduction

Based on the previous discussions of test results, conclusions regarding the system's behaviour can be drawn. These conclusions, focusing on the issues of mean performance in various conditions, connections between channel configuration and performance, channel discriminatory power are introduced in the next section of the chapter. Following the summary of the system's achievements, a number of possible extensions and future directions of investigations are discussed. The coarse data channels, the scale–space representations and categoriser modules can be adapted for improving performance, making the system more adaptive and able to deal with multiple objects in the field of view. These issues are presented in the subsequent sections, indicating the main directions of study, in some cases outlines of algorithms for future developments of the object recognition system.

11.2. Synthesis of test results

In both cases of image data sets, the system achieved very satisfactory classification accuracies. When tested with computer–generated views of synthetic 3D objects, the observed amount of inter–category confusion generally followed the pattern of human perception of inter–category similarity. The system tested on photomicrographs of fish larvae (the Aberdeen data set) also achieved good recognition accuracies, considering the great similarity of larvae shapes and the poor quality of the images. Data from image analysis methods registering the geometrical measures of the natural objects used in experiments led to very poor results, the mean accuracy being 20%, as compared to the above 70% classification accuracy obtained by the proposed system on the same fish larvae data set.

The test objects had similar volume, their apparent size registering a less than 10% change with viewpoint. The majority of the objects' features were the same, in the case of the natural images the objects being difficult to be classified by human experts. In each test data set, a category was introduced, which was easily distinguishable from the other objects.

11.2.1. Limitations of the system

The system in its present configuration can analyse images with one object in the field of view. The simple categoriser module employed in the system can not complete missing feature information caused by occluding objects. Although the used scale–space representation's topology allows the description of multiple objects in the field of view, the way in which the connectivity trees' structures are coarse coded does not make possible the analysis of multiple (separated or occluding) shapes in the input.

The apparent size of the objects in the images must be similar, due to the fixed locations and number of the analysed scale planes. Significant changes in the apparent size of the objects can cause major alterations of the scale–space representation, since the position and number of local maxima in scale space will be affected. Possible ways of coping with this problem and the case of multiple objects in the scene are discussed in the final sections of this chapter.

The system uses a small number of data channels, therefore the variety of the features described by these channels is very limited at the moment. The junction, spatial frequency and texture density information, together with the coarse coded scale–space representation proved to be sufficient for a satisfactory recognition accuracy of a limited number of object categories. The number of analysed features is a potential limiting factor as far as system performance is concerned, in situations where very large number of categories have to be learnt and classified.

The size of the self–organising maps used in the 'what' channels is fixed. Also, the employed categoriser module does not allow flexible learning. Therefore the number of feature categories encoded by the SOM modules is limited and every time that a new category must be learnt by the system, it has to be re–trained on the full training data set. This is a drawback in practical situations, where fast assimilation of new knowledge is necessary. The final sections of this chapter deal with this problem.

11.2.2. Categorisers

Artificial neural network–based collectives proved to have advantages over discriminant analysis when poor quality data was presented to the system and the size of the training set was small. ANN collectives achieved better generalisation to novel views in these cases. Discriminant analysis

outranked ANN collectives on the synthetic 5–object data set, but only with a few percent difference that are likely to be due to the variable pruning that occurs during DA. This small advantage disappeared when the data set was small and the image quality together with the characteristics of the analysed shapes made the recognition task significantly more difficult.

The importance of classifying multi–channel data instead of feature data supplied by individual channels has been shown, competitive committees like the CMC machine having the greatest difficulties in accurately identifying the input. CSO machines, that used a basic method of combining the individual channel categorisers' outputs were slightly better. In general, collective machines were better than committees. It has been shown that the introduction of more coarse data channels leads to improvement in performance. ANOVAs have proven that differences in performance can be attributed to changes in channel configuration. Exceptions constituted the cases where the behaviour of the system was rhapsodic (the Aberdeen tests with theta histograms). It has been shown, that even the channel with lowest discriminatory power (texture density) increased the accuracy of the system when it was added to the other coarse data channels, while in the case of committees, it had no effect on the performance. These aspects provide a neat generalisation to the domain of 3D shape recognition of the studies carried out on classification of natural 2D shapes based on multiple coarse data channels (the DiCANN system).

Analysis of variance showed the effect of changes in training set size on performance, which was expected, especially in the case of the Aberdeen data set. In the latter case, contrary to the 5–object data set, no control over the training/testing viewpoints was possible and the training specimens could represent larvae in any of their developmental stages, making generalisation more difficult. Such clear dependence of performance on training set size has been observed in DiCANN studies when classifying plankton specimens. It is pleasing to see, that even with the morphological variations of the larvae specimens and their similarities, good results could be obtained.

With a preliminary optimisation of the network size, ANN–based categorisers can achieve good performance and this does not change significantly with minor adjustments of the size of the hidden layer. The experiments led to the conclusion that a significant increase in network complexity is needed to reach a state where the system performance falls due to overfitting. In a practical situation, without very sophisticated tuning of the network structure reliable categorisation can be obtained with the system, as tests carried out with very different data and data set sizes have shown.

Data categories that represent objects that are very different from the other objects in the data set were easily recognised with nearly 100% accuracy, even when the training set was very small. This could provide means for reliable rejection of insignificant data presented to the system. As an example, in the field of automatic marine biota classification, objects like detritus would be easily labelled by the system as a category to be ignored.

11.2.3. Coarse data channels

The scale–space channel playing the role of 'where' channel proved to have an important role in defining the behaviour of the system. Tests carried out on the Aberdeen data set have shown, that by substituting a scale–space descriptor with another (of different discriminatory power), the performance and overall behaviour of the system changes even in circumstances where all 3 'what' channels are present.

Theta histograms led to poorer performance in the tests based on the 5–object data set and even caused inconsistent, rhapsodic results in tests carried out on the Aberdeen data. The rho–theta receptive fields' activation patterns proved to be better shape descriptors. Despite the fact that the angle resolution is significantly lower in the case of RFs in comparison with theta histograms, the fuzzyness of the descriptor and the added scale–space distance information seemed to play an important part in improving the classification accuracies. The representation seems coarse enough to reduce the variations in distance information as the viewpoint changes alter the input shape. With the increase in RF grid size, hence angle and distance resolution, the classification accuracy in preliminary tests showed a significant increase. Therefore one could employ larger RF grids in the future, at a cost of increase in input data dimensionality.

The junction channel was a powerful 'what' channel in all tests, this fact being connected to the characteristics of the data. Object geometry in the case of the 5–object data set and the coterminations produced by visible details on the bodies of larvae seemed to have high saliency. The representation was sufficiently coarse, in all cases the junction channel leading to reliable high recognition rates.

The used simplified version of classic junction detection algorithm, when presented with realistic images, supplied multiple junction categories in areas where several regions meet, these not having the ideal one–pixel width like line drawings. Also, curves posed some problems in some stages of the algorithm that was designed to work with approximately straight lines. In these situ-

ations, the junction detector supplied usually multiple junction types for one area of connected high–resolution regions. Still, it proved to be a sufficiently robust descriptor of junctions, and led to very good results in tests that involved both synthetic and natural objects.

The spatial frequency descriptor and the texture density channels had less discriminatory power. The local spatial frequency contents characterised by the FFT spectra propagated through Kohonen maps were relatively good descriptors, while the texture channel only contributed to the performance of the system in collective machine trials. Since none of the test data sets contained categories with significant differences in surface texture, this is not surprising.

11.2.4. Conclusions

These aspects suggest several areas of applications. With the system's demonstrated tolerance towards poor quality images, automatic classification of marine biota seems to be an area in which the system can find immediate applicability. By not necessitating sophisticated, specialised hardware platform and being able to operate within relatively short time–frames, its use in laboratory conditions where a large number of field–collected images are to be classified seems to be very appropriate.

With the use specialised hardware, as the following section will point out, the processing time can be considerably reduced, bringing the system closer to the sphere of real–time applications. Having demonstrated its ability to classify both geometric and non–rigid natural objects, it could provide a versatile tool for recognition of natural or man–made objects.

11.3. Future work

This section describes possible extensions that can be brought to the system and its functionality.

11.3.1. Improvements to the image processing & analysis modules

A number of minor practical modifications can be brought to the system's preprocessing modules and coarse channels.

A minor, but necessary update of the preprocessing algorithms is the extension of the functions that perform region growing and store the representations of maxima and regions, used later in building the scale–space trees. The code used in the reported object classification experiments

can store a limited number (256) of regions on each analysed scale plane. Such an 8–bit representation of region labels/maps was suitable for the used test images, but in a situation where multiple objects are analysed, possibly on finer resolution layers of the representations, the number of regions can increase significantly. A general version of the maxima mapping and region growing functions would be able to store virtually unlimited number of regions and local maxima. The scale–space tree–building algorithms would also work with such extended–precision data. During the separation of the detritus images in the Aberdeen data set, a version of the code that works with 16–bit region labels and maxima tables has been developed for future applications, but the tree–building and coarse coding algorithms were not yet fully tested.

Another improvement can be brought to the preprocessing module that enhances contrast and eliminates low–frequency information (like illumination gradients). This part uses at the moment an omnidirectional Sobel filter (without thresholding the output). This enhances not only contrast, but also noise in the image. Future versions of the system, that operate on all 7 scale planes would be affected on high–resolution planes by this aspect. For the purpose of illumination normalisation and contrast enhancement, the system could contain a more sophisticated processing module. The ON–C/OFF–C networks used in the CORT–X filter, developed by Carpenter *et al.* (1989) constitute an attractive architecture that discounts the illuminant by enhancing contrasts in the image and normalising the illumination in image regions. Without using the component modules that carry out boundary extraction and completion, the on–centre/off–surround and off–centre/on–surround filters represent a computationally cheap solution to the problem of illumination gradient elimination. These use Gaussian kernels that have a smoothing effect during the process of local contrast enhancement and illumination normalisation, which makes the structure suitable for operating in noisy conditions (Bradski & Grossberg, 1995).

An important aspect of the image processing and feature coarse coding part of the system is the parallel nature of the processes that lead to the coarse–coded feature vectors presented to the categoriser module. As it was mentioned in the system's description, it was designed to run on widely available hardware platforms, the limited computational load leading to practically acceptable processing times. Still, in a future application, the use of a multi–processor architecture may become possible.

In this case, the way in which the processing modules currently employed in the system operate can be re–organised in order to exploit the parallelism of certain tasks. With current compilers

available for multi–processor platforms, it is easy to imagine a situation where executables compiled as multi–threaded applications can carry out the major processing operations on different processors. First of all, the maxima mapping algorithm can be split up. On each wavelet coefficient plane, local maxima and the positive coefficient regions can be extracted/generated in parallel with the similar processing occurring on the other considered wavelet planes. The gain in processing time can be significant. For example, the sequential algorithms described in section 5.2. (p. 72) take 7 seconds to run on a one of the 4 processors of a SUN Enterprise 3000 platform. By using a parallel algorithm and performing the maxim and region mapping on the 4 used wavelet planes on different processors of the same hardware platform, the runtime decreases under 2 seconds.

The operations performed by the coarse data channels can easily be made parallel. The channels themselves are parallel data streams, but in the current implementation, they carry out their processing one after another in an essentially sequential manner. Not only that the 4 or more data channels can be placed on separate processors, but also the operations carried out by coarse channels like the spatial frequency and texture channels can be further parallelised. The FFT spectra and the texture descriptors can be extracted in parallel on each wavelet plane.

In future applications, the number of the employed coarse data channels can be increased as generic or application–specific feature extractor and coarse coding modules are added to the system.

The already existing coarse data channels could be further improved. First of all, for better texture description, one could employ wavelet–based or other texture analysis methods. The obtained descriptors must be rotation–invariant, due to the arbitrary positions of the visible surfaces. It must be sufficiently coarse not to register the changes in texture caused by surface orientation. Possible methods would be the use of rotation–invariant texture analysis employing continuous wavelet transform (Van de Wouwer *et al.*, 1997), which also has a finer frequency–space resolution than descriptors obtained from discrete wavelet transforms. Such methods would obviously increase the computational load and processing time, but in the case of object sets where textures are salient features, the use of a sophisticated texture description may be important for the system's performance.

In the DiCANN system (Culverhouse *et al.*, 1996), Gabor filters were used for texture classification. For each considered frequency band, the outputs of the filters of different orientations have been combined into one texture descriptor with an n–dimensional Euclidian norm, thus rendering

the texture measure rotation–invariant. Although this method proved to be computationally expensive when used on the whole surface of visible shapes, it could provide a fast texture description method when used in relatively small processing windows centred around wavelet local maxima. The method could constitute a trade–off between computational load and coarseness of representation. It is expected that such, relatively simple texture measures would have considerably higher discriminatory power than the robust texture density measures employed at the present in the system.

11.3.2. Multiple objects in the field of view

By design, the wavelet maxima and link trees are able to register multiple objects in the input image. Any tree that has a root node on one of the wavelet coefficient planes and extends on several layers towards finer resolution potentially describes an object in the image. The system can be extended to cope with multiple objects and recognise in a sequential manner the shapes present in the input. With this, the scale–space channel not only provides information on the scale–space distribution of potentially relevant features on the surface of an analysed object view, but it becomes a 'where' channel that helps in localising the objects in the field of view. With this, one moves towards a what–and–where architecture in the conventional sense (Carpenter *et al.*, 1998). Before arriving at the issues of multiple objects, the aspect of arbitrary object size must be discussed, since in a real situation the presence of different objects with different sizes can not be avoided.

In the experiments carried out until now, the objects' size was similar, so major scale differences did not affect significantly the structure of the maxima and link trees. In situations where multiple objects are present, one would expect that such control over the size of visible objects is not possible. This essentially means that the use of a fixed number of scale planes is not feasible. In order to extend the system's functionality, a first step must be the modification of the tree–building algorithm and related procedures so that the scale–space representation becomes complete. Local maxima mapping and region growing should be carried out on all available wavelet planes, the maxima and link tree building algorithms then proceeding to the generation of data structures describing all scale planes. With the present processing algorithms, maxima trees on 7 scale planes and link trees of 6 layers would be generated from the maxima and region data.

Exceptions to the case where regions accurately describe separate objects in the field of view are

represented by situations where a large, elongated or asymmetric shape is present in the image and the regions on finer resolutions break up due to details of the shape that are distant relative to the centroid. In order to decide whether there are really multiple objects in the image or multiple trees are caused by such distant details, an additional module must carry out investigations on the way in which regions split up during the coarse–to–fine mapping. Also, on coarser resolutions a region can cover more than one distinct, relatively small objects and in such situations, the maxima and link trees must be analysed on higher resolution levels, when object separation occurs. These problems could be addressed by using additional coarse data channels that provide the system with more information on the scene.

The tree that has the lowest root node (i.e. it has a root node on a coarser resolution plane than the other trees) is the dominant tree. Since this is the structure that describes the largest object in the image, it is preferable to start the classification with this. The other tree structures, with root nodes situated on higher resolution planes would follow in a sequential manner.

This leads to a structure of the system, that is outlined below in Fig. 11.1. in the form of a block diagram.

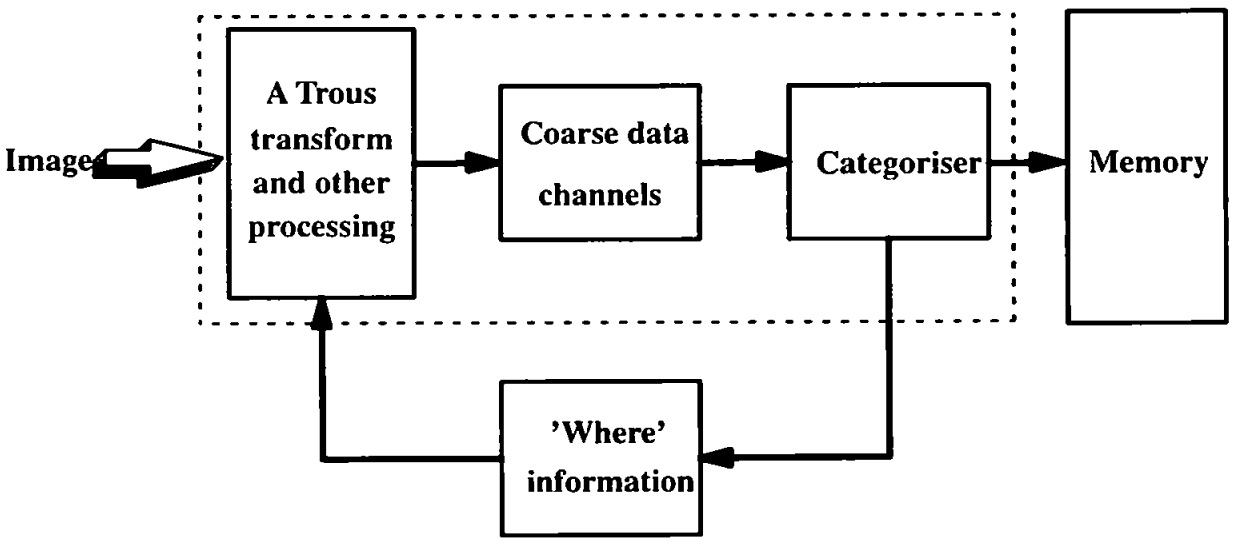


Fig. 11.1. Recognition system for multiple objects in the field of view.

The core of this extended system is constituted by the modules employed in the current version of the recognition system. Essentially, this provides the categoriser the 'what' information on objects: the scale–space channel (theta histograms or rho–theta receptive field activations) supply-

ing the information on the localisation of potentially relevant features on the surface of the analysed object view. The 'where' information in its conventional sense (Carpenter *et al.*, 1998) is supplied by the scale–space description, each root node of an object–maxlist being taken as the likely location of an object. The attention of the system is directed towards this, the coarse data channels extract and encode information on features and their distribution in scale–space. Basically, the loop defined by the 'where' information forces the system to perform a sequential processing of each shape that is present in the field of view. Also, in the case of occluded objects, the categoriser module (employing semantic nets, for example) and the scale–space description has a joint control over the feature extraction modules.

An algorithm for these operations is outlined below, using some of the notations introduced in chapter 5. It is considered that a number of k out of a total of N scale planes are used from the maxima trees. In the present system, k=4 and N=7. If the trees are shorter, i.e. have less populated layers, the top layers are left empty and ignored in further processing, as it will become apparent from the algorithm.

for each T_t **do** ; for each tree do

$q = \{p \le N \mid b \in T_{tp}\}$; get the index of the layer that contains

if $q + k - 1 > N$; the root node b of the current tree.

then $N \rightarrow N_{top}$

else $q + k - 1 \rightarrow N_{top}$

endif

$\{\} \rightarrow T_i \;,\; i = \overline{1,k}$; initialise a k–layer temporary structure

for $i = q$ **to** N_{top} **do** ; copy k (or less, if tree is shorter) layers

$T_{ti} \rightarrow T_{i-q+1}$; of maxima tree to temporary structure.

endfor

$\{m \mid m \in T_{N_{top}-q+1}\} \rightarrow M$; get set of local maxima located on

get_junction_histogram(M) $\rightarrow J$; highest populated layer of tree and

get_FFT_signature(M) $\rightarrow F$; obtain the 'what' channels' data.

get_texture_signature(M) $\rightarrow X$

build_link_tree(T) $\rightarrow L$; build connectivity tree and compute

get_RF_activations(L) → S ; scale–space descriptor.

Classify(S, J, F, X) → *object_label* ; categorise the chorus of coarse data.

endfor

One can imagine a situation where the 'what' channels are controlled by denser wavelet maxima situated on high resolution planes, when the image contains many small objects. This situation can arise when analysing natural images like the original photomicrographs of the Aberdeen data set. In this case, it seems feasible to control also the size of the processing windows used by the 'what' channels. As the algorithm moves upwards to short trees situated on high–resolution layers, the processing windows of the junction, texture and spatial frequency channels should be decreased. This adaptive measure is expected to lead to more accurate description of the small objects in the field of view, since with large, fixed–size windows like those employed by the texture channel, the risk exists that several small objects end up at least partially inside the processing window of such a channel.

During training, the system could be presented with several scaled versions of each view of the objects, the scaling factor being 2 or 1/2. This would lead to several variants of the scale–space descriptions and the outputs of the 'what' channels would also yield changing feature descriptions for different object sizes. One would expect that the system trained on these multi–channel descriptions of scaled versions of the input would be able to generalise to other descriptions of scaled object instances.

11.3.3. Improving the categoriser module

The developed system in a real–world situation would face the task of dealing with a large number of object categories. In learning stage, this would evidently mean large training sets that could significantly increase the processing time. Furthermore, learning large training sets with many categories means increasing the size of the network in order to build a general model that can describe in a satisfactory manner all of the categories. This could lead to impractical training times.

This on its own would not be an insurmountable problem, but with the presently used classifiers, the system must be re–trained every time that a new object category must be added to its knowledge. Practically, the novel object(s) must be added to the training set and the system's classifier

module is initialised, then trained with the new data set.

In these circumstances, the necessity of a classifier that can learn in a flexible manner becomes clear. On–line learning would be ideal, since every time a new category must be added to the system's knowledge base, the classifier can be trained on the novel data without presenting to it the already learnt data categories. New data would be assimilated as it is presented to the system.

A suitable neural network architecture for this purpose would be the adaptive resonance theory (ART) network (Carpenter *et al.*, 1991). The Fuzzy ART network, capable of learning analog data patterns (i.e. multidimensional data that contains real values) is a self–organising structure. The amount of category prototypes that are created in the network is influenced by the so–called vigilance parameter, that defines how well a prototype must match the input. For example, if this is set to maximum, the network will create a category template from each input pattern since no data variation is tolerated when comparing a new pattern to existing templates.

Such a Fuzzy ART module could replace the Kohonen self–organising maps used in the structure of the 'what' data channels. One would not have to worry about adjusting the size of the Kohonen map as the number of possible junction, texture etc. categories increase and above all, the map would not need re–training every time that new data is taught to the system. In testing stage, for each pattern (junction histogram, texture measure etc.) the ART module would supply as output the index of the category prototypes that are found to be the closest to the input. This can be used as a signature of the features, as in the case of the Kohonen map. The advantage of the ART module is also the fact that only one training epoch is necessary, unlike in the case of other network architectures used at present in the system.

The problem with this method is the variable number of categories. In the case of the Kohonen map, one had a fixed number of nodes and therefore the node activation signatures used as coarse data had fixed dimensionality. In the case of the Fuzzy ART module, the category templates being created dynamically in training stage, the number of such prototypes is not known *a priori*. A solution to the problem of providing a fixed–length coarse coded feature vector to the classifier module is histogramming of the prototype indexes. One can set up a sufficiently large histogram that would accumulate the outputs of the ART module (one for each junction histogram, spatial frequency spectrum or texture pattern detected in a processing window centred around a wavelet maximum). Empirically, one could train the ART module of each channel on an initially available training set of considerable size and assess the number of category templates created by the net-

works. The histogram size can be set to a multiple of this number, which would suffice, since realistically, one does not expect that the number of different junction, spatial contrast or texture pattern categories register a huge increase with the presentation of new objects to the system.

The classifier module of the system must also be replaced by an ART–based categoriser in order to make on–line learning possible. In this case, the classifier must allow supervised learning, since the target patterns specifying the object categories that the input views belong to must be presented to the system in training stage. Therefore an architecture like the Fuzzy ARTMAP (Carpenter *et al.*, 1992) should be used.

A potential problem with the use of the ARTMAP under these conditions is one related to the nature of the data presented to the categoriser. Having multiple data channels providing the coarse–coded feature descriptions, the input pattern to the categoriser contains information with different discriminatory power (as it has been shown in tests). Noisier or channels with less salient information should be given less attention in the update process. During learning, the ARTMAP's map field is able to adjust the ART modules' vigilance parameters only based on global judgements regarding the whole input pattern's predictive power. This could have a negative effect on performance.

A few preliminary investigations carried out in this direction showed indeed that the test performance of the Fuzzy ARTMAP, when trained on 1,2,3 and 4 channels' data did not improve. The map was trained with the feature vectors obtained from the computer–generated 5–object data set, the training and test sets were split in 1:1 ratio. After optimising the learning parameters, the mean classification accuracy stayed between 68% and 70% independent of the number of grouped channels. As new channels were added, the performance actually decreased with 1–2% compared to the one achieved by the Fuzzy ARTMAP trained on theta histogram data alone. Considering how prominently the performance increased on the same data set when using DA and feedforward ANN–based collectives, this trend seems to confirm the hypothesis that global adjustments of the vigilance parameter in a Fuzzy ARTMAP would be inappropriate.

Therefore a further improvement of the categoriser could be achieved, by the use of more adaptive network architectures. An ARTMAP–style network that is able to take into consideration the differences in discriminatory power between channels would keep the advantages enumerated above (on–line learning, reduced training time, optimised internal structure) and would possibly further improve performance. At present, solutions like the Fusion ARTMAP (Asfour *et al.*,

1993) seem to be suitable for such a task. This architecture employs an ART module for each input channel and their vigilance levels are individually adjusted during learning. It also has the advantage that it creates less weights and categories than the Fuzzy ARTMAP does in the same conditions – this feature can become important in the case of very large data sets.

But with the use of separate self–organising modules for each channel present in the input, the Fusion ARTMAP is not a collective machine. Essentially, it is a very sophisticated committee machine, where learning and decision making takes place based on the outputs of network modules associated with each input channel. Does one sacrifice the robustness and the superior performance of a collective machine? It remains to be seen and it is most certainly an interesting subject for future investigations, whether such a sophisticated, adaptive committee machine can actually outperform the collectives on noisy, difficult data sets.

A significantly more refined categoriser is required in the analysis of difficult scenes where multiple objects are partially occluding each other. One would expect that based on the constructed scale–space trees' structure information and the 'what' information associated with the nodes of these trees, complex neural network architectures could be taught to complete missing parts of maxima and/or connectivity trees and reach a decision regarding the nature of the objects. Likely candidates for this task are modules based on semantic nets (Winston, 1993). The knowledge base of the system would contain nets of nodes that describe multiscale features and the links would represent relationships between these.

One could imagine a situation where such a net stores the ways in which connectivity trees and the associated feature descriptors change in scale–space, partial occlusion making certain parts of the representation absent in the input. With the features, inter–relationships and restrictions on features stored in a semantic net, the system by inference would be able to reach a decision regarding the nature of an occluded object in these circumstances. Recent successes in the use of semantic nets combined with neural networks in image understanding and recognition of objects/object parts (Klusch & Napiwotzki, 1993; Robinson *et al.*, 1994; Brown *et al.*, 1997) suggest that such a direction of research could lead to promising results.

11.4. Publications

Much of the content of this thesis has been prepared and accepted for publication in the Image

& Vision Computing Journal:

Toth, L., Culverhouse, P. F., 3D object recognition from static 2D views using multiple coarse data channels, Image & Vision Computing Journal., (In press).

11.5. Summary

This final chapter summarised the aspects related to the object recognition system's behaviour in various test conditions and its achievements. The conclusions that could be drawn from the way in which the categoriser, the coarse data channels and image analysis modules functioned in different conditions provided the grounds for future directions of investigations. The last sections of this chapter outlined a number of methods and measures that could be taken for extending the system's functionality and improving its performance.

This work provided a novel system for 3D object recognition with immediate practical applicability. As a piece of research, it allowed the study of multi–channel feature representations in a framework inspired by biological vision and made possible generalisations to the problem of 3D object recognition of several aspects studied in the field of natural 2D shape recognition. Important conclusions could be drawn on the classification of multi–channel data in the context of 3D shape classification, based on results of tests that used similar shapes and difficult data sets. The validity of novel approaches in image analysis and feature extraction/representation have been proved. The novelty of this work lies mainly in the multiresolution analysis and representation scheme, the attention focusing mechanism coupled with unsupervised feature grouping. The developed system constitutes the basis for future schemes which would be able to deal with real, complex 3D scenes.

Chapter 12. References

Allen, R. L., Kamangar F. A., Stokely, E. M. (1993). Laplacian and orthogonal wavelet pyramid decompositions in coarse–to–fine registration. IEEE Transactions on signal processing, Vol. 41, No. 12, pp. 3536–3541.

Asfour, Y. R., Carpenter, G. A., Grossberg, S., Lesher, G. W. (1993). Fusion ARTMAP : An adaptive fuzzy network for multi–channel classification. Proceedings of the 3rd International Conference on Industrial Fuzzy Control & Intelligent Systems (IFIS), pp. 155–160. IEEE Service Center, NY.

Augusteijn, M. F. , Clemens, L. E. (1996). A neural–network approach to the detection of texture boundaries. Engineering Applications of Artificial Intelligence, Vol. 9, No.1, pp. 75–81.

Ballard, D.H. (1987). Generalizing the Hough transform to detect arbitrary shapes. In *Readings in computer vision: issues, problems, principles, and paradigms*, pp. 714–725. Eds. Fischler, M. A., Firschein, O., Morgan Kaufmann Publishers, Inc.

Ballard, D. H., Brown, C. M. (1982). *Computer vision*, p. 300, Prentice–Hall Inc., NJ.

Barrett, R., Ramsay, A., Sloman, A. (1986). *POP–11: A practical language for artificial intelligence*, Ellis Horwood Ltd., Chichester.

Battiti, R., Colla, A. M. (1994). Democracy in neural nets: voting schemes for classification. Neural Networks, Vol. 7, No. 4, pp. 691–707.

Biederman, I. (1987). Recognition by components: a theory of human image understanding. Psychological Review, Vol. 94, No.2, pp. 115–147.

Biederman, I. (1990). Higher–level vision. In *Visual cognition and action – An invitation to cognitive science,* Vol. 2, pp. 41–72. Eds. Osheban, D.N., Kosslyn, S.M., Hollerbach, J.M. The MIT Press, Cambridge.

Bijaoui, A., Starck, J.–L., Murtagh, F. (1994). Restauration des images multi–echelles par l'algorithme a trous. Traitement du Signal, Vol. 11, pp. 229–243.

Botha, E. C., Barnard, E., Barnard, C. J. (1996). Feature–based classification of aerospace radar targets using neural networks. Neural Networks, Vol. 9, No.1, pp.129–142.

Brady, M. (1987). Representing shape. In *Parallel architectures and computer vision workshop*, pp. 256–265. Somerville College, Oxford.

Bradski, G., Grossberg, S. (1995). Fast–learning VIEWNET architectures for recognizing three–dimensional objects from multiple two–dimensional views. Neural Networks, Vol. 8, No. 7/8, pp. 1053–1080.

Brown, M. S., McNittGray, M. F., Mankovich, N. J., Goldin, J. G., Hiller, J. (1997). Method for segmenting chest CT image data using an anatomical model : Preliminary results. IEEE Transactions on Medical Imaging, Vol. 16, No. 6, pp. 828–839.

Burt, P.J. (1988). Smart sensing within a pyramid vision machine. Proceedings of the IEEE, Vol. 76, No. 8, pp. 1006–1015.

Burt, P. J., Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. IEEE Transactions on Communications, Vol. 31, No.4, pp. 532–540.

Canny, J. (1987). A computational approach to edge detection. In *Readings in computer vision: issues, problems, principles and paradigms*, pp. 184–203. Eds. Fischler, M. A., Firschein, O., Morgan Kaufmann Publishers, Inc.

Carpenter, G. A., Grossberg, S., Lesher, G. W. (1998). The what–and–where filter – A spatial mapping neural network for object recognition and image understanding. Computer Vision and Image Understanding, Vol. 69, No. 1, pp. 1–22.

Carpenter, G. A., Grossberg, S., Menahian, C. (1989). Invariant recognition of cluttered scenes by a self–organizing ART architecture: CORT–X boundary segmentation. Neural Networks, Vol. 2, No. 3, pp. 169–181.

Carpenter, G. A., Grossberg, S., Rosen, D. B. (1991). Fuzzy ART : Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, Vol. 4, No. 6, pp. 759–771.

Carpenter, G. A., Grossberg, S., Markuson, N., Reynold, J. H., Rosen, D. B. (1992). Fuzzy ARTMAP : A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks, Vol. 3, No. 5, pp. 698–713.

Chandran, V., Carswell, B., Boashash, B., Elgar, S. (1997). Pattern recognition using invariants defined from higher–order spectra: 2–D image inputs. IEEE Transactions on Image Processing, Vol. 6, No. 5, pp. 703–712.

Chandrasekaran, V., Palaniswami, M., Caelli, T. M. (1995). Spatio–temporal feature maps using gated neuronal architecture. IEEE transactions on neural networks, Vol. 6, No.5, pp.1119–1130.

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, Vol. 20, No. 1, pp. 37–46.

Cohen, I., Raz, S., Malah, D. (1997). Orthonormal shift–invariant wavelet packet decomposition and representation. Signal Processing, Vol. 57, No. 3, pp. 251–270.

Culverhouse, P. F., Williams, R., Reguera, B., Ellis, R. and Parisini, T. (1996). Automatic categorisation of 23 species of Dinoflagellate by artificial neural network. Marine Ecology – Progress Series, Vol. 139, No. 1–3, pp.281–287.

Cutzu, F. , Edelman, S. (1994). Canonical views in object representation and recognition. Vision Research, Vol. 34, No. 22, pp. 3037–3056.

Cutzu, F., Edelman, S. (1996). Faithful representation of similarities among 3–dimensional shapes in human vision. Proceedings of the National Academy of Sciences of the United States of America, Vol. 93, No. 21, pp. 12046–12050.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. Communications in Pure and Applied Mathematics, Vol. 41, No. 7, pp.909–996.

Daubechies, I. (1989). Orthonormal bases of wavelets with finite support – connection with discrete filters. In *Wavelets: time–frequency methods and phase space*, pp. 38–66. Eds. Combes, J.M., Grossman, A., Tchamitchian, Ph. Springer–Verlag, Berlin.

Daugman, J.G. (1980). Two–dimensional spectral analysis of cortical receptive field profile. Vision Research, Vol. 20, pp. 847–856.

Daugman, J.G. (1988). An information–theoretic view of analog representation in striate cortex. In *Computational Neuroscience*, pp. 403–423. Ed. Schwartz, E.E., MIT Press, Cambridge, MA.

Deschenes, S., Sheng, Y. L., Chevrette, P. C. (1998). Three–dimensional object recognition from two–dimensional images using wavelet transforms and neural networks. Optical Engineering, Vol. 37, No. 3, pp. 763–770.

Dutilleux, P. (1989). An implementation of the algorithme a trous to compute the wavelet transform. In *Wavelets: Time–Frequency Methods and Phase–Space*, pp. 298–304. Eds. Combes, J.M., Grossman, A., Tchamitchian, Ph. , Springer–Verlag, Berlin.

Edelman, S. (1995a). Class similarity and viewpoint invariance in the recognition of 3D objects. Biological Cybernetics, Vol. 72, No. 3, pp. 207–220.

Edelman, S. (1995b). Representation, similarity and the Chorus of prototypes. Minds and Machines, Vol.5, No.1, pp.45–68.

Edelman, S., Bulthoff H.H. (1992). Orientation dependence in the recognition of familiar and novel views of 3D objects. Vision Research, Vol. 32, No. 12, pp.2385–2400.

Edelman, S., Duvdevani–Bar, S. (1997). A model of visual recognition and categorization. Philosophical transactions of the Royal Society of London – Biological Sciences, Vol. 352, No. 1358, pp. 1191–1202.

Edelman, S., Weinshall, D. (1991). A self–organising multiple–view representation of 3D objects. Biological Cybernetics, Vol. 64, No.3, pp. 209–219.

Ellis, R., Simpson, R., Culverhouse, P. F., Parisini, T.,Williams, R., Reguera, B.,Moore, B., and Lowe, D. (1994). Expert visual classification and neural networks: can general solutions be found?. In *Proceedings IEEE Oceans '94*, Brest, September 1994, pp. 330–334.

Ellis, R., Simpson, R., Culverhouse, P. F., Parisini, T. (1997). Committees, collectives and individuals: expert visual classification by neural network. Neural Computing & Applications, Vol.5, No. 2, pp. 99–105.

Elmes, D. G., Kantowitz, B. H., Roediger, H. L. III (1989). *Research methods in psychology.* West Publishing Company, St. Paul.

Eurich, C. W., Schwegler, H. (1997). Coarse coding: Calculation of the resolution achieved by a population of large receptive neurons. Biological Cybernetics, Vol. 76, No. 5, pp. 357–363.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Ann. Eugenics, Vol. 7, No. 2, pp. 179–188.

Fleiss, J. L. (1981). *Statistical methods for rates and proportions.*, pp. 212–236. John Wiley & Sons, Inc., NY.

Gaudart, L., Crebassa, J., Petrakian, J. P. (1993). Wavelet transform in human visual channels. Applied Optics, Vol. 32, No.22, pp. 4119–4127.

– (1988). *Discriminant analysis and clustering*. Eds. Gnanadesikan, R., Byke, R., Griffiths, P. A., Hackerman, N. National Academy Press, Washington DC.

Graps, A. (1995). An introduction to wavelets. IEEE Computational Science & Engineering, Vol.2, No. 2, pp. 50–61.

Hays, W. L. (1988). *Statistics*. 4th Edition. Holt, Rinehart and Winston, Inc., NY.

Hayward, W. G. (1998). Effects of outline shape in object recognition. Journal of Experimental Psychology – Human perception and performance, Vol. 24, No. 2, pp. 427–440.

Hebb, D. O. (1949). *The organisation of behavior*. Wiley, NY.

Helterbrand, J. D., Cressie, N., Davidson, J. L. (1994). A statistical approach to identifying closed object boundaries in images. Advances in applied probability, Vol. 26, No. 4, pp. 831–854.

Henkel, R. D. (1995). Segmentation in scale space. In *Proceedings of the 6th International Conference on Computer Analysis of Images and Pattern*, CAIP'95, Prague.

Hildreth, E. C., Ullman, S. (1989). The computational study of vision. In *Foundations of Cognitive Science*, pp. 581–630. Ed. Posner, M. I., The MIT Press, London.

Hirakura, Y., Yamaguchi, Y., Shimizu, H., Nagai, S. (1996). Dynamic linking among neural oscillators leads to flexible pattern recognition with figure–ground separation. Neural Networks, Vol. 9, No.1, pp. 189–209.

Holschneider, M., Martinet, R. K., Morlet, J., Tchamitchian, Ph. (1987). A real–time algorithm for signal analysis with the help of the wavelet–transform. In *Wavelets: Time–Frequency Methods and Phase–Space*, pp. 286–297. Eds. Combes, J.M., Grossman, A., Tchamitchian, Ph., Springer–Verlag, Berlin.

Howell, D. C. (1982). *Statistical methods for psychology*. Duxbury Press, Boston, MA.

Hsieh, J. W., Liao, H. Y. M., Fan, K. C., Ko, M. T., Hung, Y. P. (1997). Image registration using a new edge–based approach. Computer Vision and Image Understanding, Vol. 67, No. 2, pp. 112–130.

Hubel, D. H. (1982). Exploration of the primary visual cortex: 1955–1978. Nature, Vol. 299, No. 5883, pp. 515–524.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., Hinton, G. E. (1991). Adaptive mixtures of local experts. Neural Computation, Vol. 3, No. 1, pp. 79–87.

Jawerth, B., Sweldens, W. (1994). An overview of wavelet–based multiresolution analyses. SIAM Review, Vol. 36, No.3, pp. 377–412.

Julesz, B., Bergen, J. R. (1987). Textons, the fundamental elements in preattentive vision and perception of textures. In *Readings in computer vision: issues, problems, principles, and paradigms.*, pp. 243–256, Eds. Fischler, M. A., Firschein, O., Morgan Kaufmann Publishers, Inc.

Khotanzad, A., Liou, J. J–H. (1996). Recognition and pose estimation of unoccluded three–dimensional objects from a two–dimensional perspective view by banks of neural networks. IEEE Transactions on Neural Networks, Vol. 7, No. 4, pp. 897–906.

Klusch, M., Napiwotzki, R. (1993). HNS – A hybrid neural system and its use for the classification of stars. Astronomy and Astrophysics, Vol. 276, No. 1, pp. 309–319.

Kohonen, T. (1982). Self–organized formation of topologically correct feature maps. Biological Cybernetics, Vol. 43, No. 1, pp. 59–69.

Kohonen, T. (1987). *Self organisation and associative memory.* Springer Verlag, Berlin.

Laine, A., Fan, J. (1993). Texture classification by wavelet packet signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No.11, pp. 1186–1191.

Laine, A., Fan, J. (1996). Frame representations for texture segmentation. IEEE Transactions on Signal Processing, Vol. 5, No. 5, pp. 771–779.

Landis, R., Koch, G. G. (1977). The measurement of observer agreement for categorical data. Biometrics, Vol. 33, No. 3, pp. 159–174.

Lee, J.S. , Sun, Y.N., Chen, C.H. (1995). Multiscale corner detection by using wavelet transform. IEEE Transactions on Image Processing, Vol. 4, No.1, pp. 100–104.

Lettvin, J. Y., Maturana, H. R., McCulloch, W. S., Pitts, W. H. (1959). What the frog's eye tells the frog's brain. Proceedings of the Institute of Radio Engineering, Vol. 47, pp. 1940–1951.

Lin, W. H., Lee, J. S., Chen, C. H., Sun, Y. N. (1998). A new multiscale–based shape recognition method. Signal Processing, Vol. 65, No. 1, pp. 103–113.

Liu, J. Q., Yang, Y.–H. (1994). Multiresolution color image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 7, pp. 689–700.

Logothetis, N. K., Pauls, J., Bulthoff, H. H., Poggio, T. (1994). View–dependent object recognition by monkeys. Current Biology, Vol. 4, No. 5, pp. 401–414.

Lowe, D. G. (1987). The viewpoint–consistency constraint. International Journal of Computer Vision, Vol.1, No. 1, pp. 57–72.

Lu, C. S., Chung, P. C., Chen, C. F. (1997). Unsupervised texture segmentation via wavelet transform. Pattern Recognition, Vol. 30, No. 5, pp. 729–742.

Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, pp. 674–693.

Mallat, S., Hwang, W. L. (1992). Singularity detection and processing with wavelets. IEEE Transactions on Information Theory, Vol. 38, No. 2, pp.617–643.

Mallat, S. (1996). Wavelets for a Vision. Proceedings of the IEEE, Vol. 84, No.4, pp. 604–614.

Marr, D. (1982). *Vision – A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman, San Francisco.

Masters, T. (1993). *Practical neural network recipes in C++*. Academic Press, Inc., San Diego, CA.

Mel, B. W. (1997). SEEMORE: Combining color, shape and texture histogramming in a neurally inspired approach to visual object recognition. Neural Computation, Vol. 9, No. 4, pp. 777–804.

Minsky, M. (1975). A framework for representing knowledge. In *The Psychology of Computer Vision*, pp. 211–277. Ed. Winston, P.H., McGraw–Hill, NY.

Murtagh, F., Zeilinger, W., Starck, J.–L., Bijaoui, A. (1995). Object detection using multiresolution analysis. In *Astronomical Data Analysis Software and Systems IV.*, ASP, pp.260–263. Editors Shaws, D., Payne, H., Hayes, J.

Newell, F. N. (1998). Stimulus context and view dependence in object recognition. Perception, Vol. 27, No. 1, pp. 47–68.

Niemann, T., Lappe, M., Hoffmann, K.–P. (1996). Visual inspection of three–dimensional objects by human observers. Perception, Vol. 25, No. 9, pp. 1027–1042.

Palmer, S. E., Rosch, E., Chase, P. (1981). Canonical perspective and the perception of objects. In *Attention and Performance IX.*, pp. 135–151. Eds. Long, J., Baddeley, A. Erlbaum, Hilssdale, NJ.

Poggio, T., Edelman, S. (1990). A network that learns to recognize three–dimensional objects. Nature, Vol. 343, No. 6255, pp. 263–266.

Porat, M., Zeevi, Y. Y. (1988). The generalized Gabor scheme of image representation in biological and machine vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 4, pp. 452–467.

Ramsay, A., Barrett, R. (1982). *AI in practice : Examples in Pop–11.* Ellis Horwood Limited, Chichester.

Rao, C. R. (1952). *Advanced statistical methods in biometric research.* Wiley, NY.

Rao, R. P. N., Zelinsky, G. J., Hayhoe, M. M., Ballard, D. H. (1996). Modelling saccadic targeting in visual search. In *Advances in Neural Information Processing Systems*, Vol. 8, pp. 830–836. MIT Press, Cambridge.

Rattarangsi, A., Chin, R. T. (1992). Scale–based detection of corners of planar curves. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 4, pp. 430–449.

Reichel, F., Loffler, W. (1994). Optical space frequency–analysis for real–time pattern recognition. International Journal of Optoelectronics, Vol. 9, No. 1, pp. 99–109.

Ren, Z., Ameling, W., Jensch, P. (1990). An attributed tree data structure for representing the descriptions of object contours in images. Proceedings of the SPIE – The International Society for Optical Engineering, Vol. 1360, No. 2, pp. 956–969.

Robinson, G. P., Colchester, A. C. F., Griffin, L. D. (1994). Model–based recognition of anatomical objects from medical images. Image and Vision Computing, Vol. 12, No. 8, pp. 499–507.

Rosenfeld, A. (1987). Recognizing unexpected objects: a proposed approach. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 1, No.1, pp. 71–84.

Rue, F., Bijaoui, A. (1997). A multiscale vision model to analyse field astronomical images. Experimental Astronomy, Vol. 7, No. 3, pp. 129–160.

Rumelhart, D. E., McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1: Foundations. The MIT Press, Cambridge, MA.

Sabbah, D. (1985). Computing with connections in visual recognition of origami objects. Cognitive Science, Vol. 9, No. 1, pp. 25–50.

Sajda, P., Spence, C. D., Hsu, S., & Pearson, J. C. (1995). Integrating neural networks with image pyramids to learn target context. Neural Networks, Vol. 8, No. 7/8, pp.1143–1152.

Schiele, B., Crowley, J. L. (1997). Transinformation of object recognition and its application to viewpoint planning. Robotics and Autonomous Systems, Vol. 21, No. 1, pp. 95–106.

Seibert, M., Waxman, A. M. (1992). Learning and recognizing 3D objects from multiple views in a neural system. In *Neural Networks for Perception*, Vol. 1, pp.426–444. Academic Press,Inc.

Shensa, M. J. (1992). The Discrete Wavelet Transform: wedding the A Trous and Mallat algorithms. IEEE Transactions on Signal Processing, Vol.40, No.10, pp.2464–2482.

Shepard, R. N., Metzler, J. (1971). Mental rotation of three–dimensional objects. Science, Vol. 171, pp. 701–703.

Simpson, R. G. (1992). *Classification of complex two–dimensional images in a parallel distributed processing architecture.* PhD thesis, University of Plymouth.

Singh, M., Landau, B. (1998). Parts of visual shape as primitives for categorization. Behavioral and Brain Sciences, Vol. 21, No. 1, p. 36.

Sokal, R. R. (1974). Classification: purposes, principles, progress, prospects. Science, Vol. 185, No. 4157, pp. 1115–1123.

Spiegel, M. R. (1992). *Theory and problems of probability and statistics.* McGraw–Hill, NY.

Starck, J.–L., Murtagh, M., Bijaoui, A. (1995a). Multiresolution and astronomical image processing. In *Astronomical Data Analysis Software and Systems IV.*, pp.279–288. Eds. Shaws, D., Payne, H., Hayes, J.

Starck, J.–L., Murtagh, M., Bijaoui, A. (1995b). Multiresolution support applied to image filtering and restoration. Graphical Models and Image Processing, Vol. 57, No. 5, pp.420–431.

Strang, G., Nguyen, T. (1996). *Wavelets and filter banks.* Wellesley–Cambridge Press, Cambridge, MA.

Sugihara, T., Edelman, S., Tanaka, K. (1998). Representation of objective similarity among three–dimensional shapes in the monkey. Biological Cybernetics, Vol. 78, No. 1, pp. 1–7.

Sutherland, N. S. (1979). The representation of three–dimensional objects. Nature, Vol. 278, pp. 395–398.

Tarr, M. J., Pinker, S. (1989). Mental rotation and orientation–dependence in shape recognition. Cognitive Psychology, Vol. 21, No. 2, pp.233–282.

Ullman, S. (1984). Visual routines. Cognition, Vol. 18, No. 1–3, pp. 97–159.

Ullman, S. (1989). Aligning pictorial descriptions: an approach to object recognition. Cognition, Vol. 32, No. 3, pp. 193–254.

Ullman, S., Basri, R. (1991). Recognition by linear combination of models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 10, pp. 992–1005.

Ungerleider, L. G., Mishkin, M. (1982). Two cortical visual systems. In *Analysis of visual behaviour.*, Eds. Goodale, M. A., Mansfield, R. J. W.

Unser, M. (1995). Texture classification and segmentation using wavelet frames. IEEE Transactions on Image Processing, Vol. 4, No. 11, pp. 1549–1560.

Unser, M., Aldroubi, A. (1996). A Review of Wavelets in Biomedical Applications. Proceedings of the IEEE, Vol. 84, No.4, pp. 626–638.

Unser, M., Aldroubi, A., Eden, M. (1983). The L_2 polynomial spline pyramid. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 4, pp. 364–379.

Van Hulle, M. M., Tollenaere, T. (1993). A modular artificial neural network for texture processing. Neural Networks, Vol. 6, No. 1, pp. 7–32.

Van de Wouwer, G. (1998). *Wavelets for texture analysis.* PhD thesis, University of Antwerp.

Van de Wouwer, G., Vautrot, P., Scheunders , P., Livens, S., Van Dyck, D. (1997). Continuous wavelets for rotation–invariant texture classification and segmentation. In *Proceedings of the 2nd IEEE UK Symposium on Applications of Time–Frequency and Time–Scale Methods (TFTS'97)*, pp. 129–132, Conventry, UK.

Vetterli, M., Herley, C. (1992). Wavelets and filter banks : Theory and design. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 40, No. 9, pp. 2207–2232.

Wackerly, D. D., McClave, J. T., Rao, P. V. (1978). Measuring nominal scale agreement between a judge and a known standard. Psychometrika, Vol. 43, No. 2, pp. 213–223.

Warrington, E. K., Taylor, A. M. (1973). The contribution of the right parietal lobe to object recognition. Cortex, Vol. 9, pp. 152–164.

Waxman, A. M., Seibert, M. C., Gove, A., Fay, D. A., Bernardon, A. M., Lazott, C., Steele, W. R., Cunningham, R. K. (1995). Neural processing of targets in visible, multispectral IR and SAR imagery. Neural Networks, Vol. 8, No. 7/8, pp. 1029–1051.

Winston, P. H. (1993). *Artificial intelligence.* Third edition, pp. 15–45. Addison–Wesley, NY.

Wolpert, D. H. (1992). Stacked generalisations. Neural Networks, Vol. 5, No. 2, pp. 241–259.

Yin, H., Allinson, N.M. (1994). Unsupervised segmentation of textured images using a hierarchical neural structure. Electronics Letters, Vol. 30, No. 22, pp. 1842–1843.

Yoon, S. H., Kim, J. H., Alexander, W. E., Park, S. M., Sohn, K. H. (1998). An optimum solution for scale–invariant object recognition based on the multiresolution approximation. Pattern Recognition, Vol. 31, No. 7, pp. 889–908.

Yuille, A. L., & Ullman, S. (1990). Computational theories of low–level vision. In *Visual cognition and action – An invitation to cognitive science,* Vol. 2, pp.5–39. The MIT Press.

Zell, A., Mache, N., Sommer, T., Korb, T. (1991). Recent developments of the SNNS neural network simulator. Proceedings of the SPIE – Applications of Artificial Neural Networks, Vol. 1469, pp. 708–719. Ed. Rogers, S. K.

APPENDIX A.

C++ Source Code for Image Processing & Analysis

A1. The C++ class hierarchy

```
/*****************************************************************************************/
/*  Module: hierarchy.h                                                                  */
/*  Description: Class hierarchy.                                                        */
/*  (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996       */
/*****************************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/file.h>
#include <alloca.h>
#include <string.h>
#include <math.h>

#define UINT    unsigned int
#define BYTE    unsigned char
#define WORD    unsigned int

#define TRUE 255
#define FALSE 0


struct Node                 // used for stack
{
 int x,y;
 Node *next;
};

struct LocMax               //local maximum
 {
  int x,y;
  float magn;
  int index;
  int label;

  LocMax() {};
  LocMax(int, int, float,int,int);
 };

struct Polar                // a rho–theta link between two maxima
 {
  int src;
  float ro;
  float theta;
  int dest;

  Polar() {};
  Polar(int, float, float, int);
 };

struct RtEntry              // a link in ro–theta list
 {
  Polar* elem;
  RtEntry *next;
```

```
};

struct Element              //linked list's element, whose info member "elem"
 {                          //points to a local maximum structure
  LocMax* elem;
  Element* next;
 };

struct Object               // element of object list
 {
  Element* layers[6];       // linked list for every layer
  Object* next;
 };

struct RtList               //ro–theta list for every object
 {
  RtEntry* links[3];        // the 3 link levels in an object's r–t list
  RtList* next;
 };

class IMG
{
 public:

  WORD    cx,cy;            // dimx & dimy
  WORD    HeaderSize;       // image header size
  float  **pixels;          // pointer to the raster bits
  BYTE *header;
  char fname[120];

  void SetUpImg(void);      // allocate space for raster bits
  void DestroyImg(void);

  IMG() {};                  // this is called by derived class constructors
  IMG(IMG*);
  IMG(UINT,UINT,UINT);
  ~IMG();

};

class IMGRAST : public IMG
{

public:

   IMGRAST(char *);         // load image from file
                            // using only base class' destructor

};

class PROCIMG
{
protected:

  IMG *imgIn;
  IMG *imgOut;              // result image

public:

  PROCIMG (IMG*);
  ~PROCIMG();
};

class TRANSFIMG
{
 public:

  IMG* imgIn;
  UINT height,width;
```

```
  TRANSFIMG(IMG*);
  ~TRANSFIMG();
};

class PROCAREA : public PROCIMG
{
  protected:

   Node *stack;

   void Push(int , int );
   void Pop(int*, int*);

public:

  int* area;

  PROCAREA(IMG*);
  ~PROCAREA(void);

  IMG* Sobel(float);                    //sobel with a threshold
  IMG* Regions(int **, int, int*, float);  //region growing on member image
};

class ATROUS : public TRANSFIMG
{
 public:

   IMG *ci, *di[16];        //smoothed plane and detail planes
                            //as float 'images'
   int nrscales;
   int masksize;
   float* h;

   ATROUS(IMG *, int);
   ~ATROUS();

   void CompAtrous();
};

class MAPMAXATROUS : public ATROUS
{
  protected:

    Element *tmp,*last,*Head;

    void Append(LocMax*);
    void AppendToCurrentObj(LocMax*,Object*, int);
    Object* NewObject(LocMax* , int);
    void SplitRegCurrentObj(Object*, int);
    RtList* CreateRtList();
    void AppRoTheta(Polar*, int, RtList*);


  public:

   char fname[50];
   float **max[6];       // maxima 'images'
   Element *maxlist[6];    // maxima list on each layer
   Object *HeadObj;        // maxlist for each object; unknown nr. of objects!
   RtList *rothetalist;    // ro-theta lists for every object

   int nrnodes[16];        // how many maxima were found on each layer
   int nrobjects;          // nr. of objects

   MAPMAXATROUS(IMG*, int, char*);
   ~MAPMAXATROUS();

   void MapMaxima(int);
   void BuildTree(int**, char ***);
```

```
                        //receives array of region areas, one entry for each maximum,
                         // on each layer!   area[0][...] is for plane 6
                        // also receives region 'images', region[0] is plane 6
  double dist(int, int, int, int);          //distance - now just Euclidian
  void MapRoTheta(Element **);              // builds ro-theta tree (redundant fully conncted version!)
  void LogNormTree(RtList*);                // log-norm distances in ro-theta tree
};
```

A2. The class constructors/destructors

```
/*****************************************************************************/
/* Module:       construct.cpp                                               */
/* Description:  The constructors and destructors of base classes,           */
/*                          except ATROUS, MAPMAXATROUS.                     */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996 */
/*****************************************************************************/

#include "hierarchy.h"
#include <sys/types.h>
#include <sys/stat.h>

LocMax::LocMax(int xx, int yy, float m, int ind, int lab)
{
 x = xx;
 y = yy;
 index = ind;
 magn = m;
 label = lab;
}

Polar::Polar(int s, float r, float th, int d)  //ro-theta entry
{
 src = s;
 ro = r;
 theta = th;
 dest = d;
}

IMG::IMG(IMG* img)                  //copying img's data to data members
{
 int x,y;

 cx=img->cx;
 cy=img->cy;

 HeaderSize = img->HeaderSize;
 SetUpImg();

 for(y=0; y<cy; y++)
  for(x=0; x<cx; x++)
   pixels[y][x] = img->pixels[y][x];

 for (x=0; x<HeaderSize; x++)
   header[x] = img->header[x];

 strcpy(fname, img->fname);         //becomes the same name
}

IMG::IMG(UINT y, UINT x, UINT hs)  //allocate y by x pixel image
{
 cx = x;
 cy = y;
 HeaderSize = hs;
 SetUpImg();
}
```

```
IMG::~IMG(void)
{
 DestroyImg();
}

void IMG::SetUpImg(void)
{
  int i;

  header = new BYTE[HeaderSize];
  pixels = (float **) malloc(cy * sizeof(float*));
  for(i=0; i<cy; i++)
    pixels[i] = (float*) malloc(cx * sizeof(float));
}

void IMG::DestroyImg(void)
{
 int i;

 for(i=0; i<cy; i++)
  free(pixels[i]);

 free(pixels);
 delete header;
}

IMGRAST::IMGRAST(char *path)      //loads rasterfile
{
 UINT i,j;
 int errcode;
 BYTE stuff;
 FILE *fp;
 char path1[120];
 struct stat buff;

 UINT NSIZE = 256;                //(mast)rasterfiles : 256 x 256 pixels

 strcpy(path1, path);
 strcat(path1, ".bmp");

 if( ( errcode = stat(path1, &buff) ) == -1)  //get file info
  {
            printf("Image file error! \n");
            exit(1);
  }

 switch (buff.st_size)            //set up HeaderSize according to file format
 {
  case 66336:  HeaderSize = 800; break;  // Sun Raster with color table & header (768+32 bytes)
  case 65536:  HeaderSize = 0; break;     // no header at all
  case 65856:  HeaderSize = 320; break;   // MAST Rasterfile; just header, no colortable
  case 66614:  HeaderSize = 1078; break;  // BMP 256 colour, 256x256
 }

 fp=fopen(path1, "rt");
 if(fp==NULL) { printf("Image file open error! \n ");
          exit(1); }

 cy = cx = NSIZE;

 SetUpImg();              //alloc pixel array

 for(j=0; j<HeaderSize; j++)
 {
  fscanf(fp, "%c", &stuff);
  header[j]=stuff;        //read and store header
 }

 float Gray[256];         // grayscale colortable!!
```

```
for(j=54; j<1078; j+=4)   //colortable
  Gray[(j-54)/4] = header[j]*0.114 + header[j+1]*0.587 + header[j+2]*0.299 ;

for(j=0;j<NSIZE;j++ )
 for(i=0;i<NSIZE;i++)
 {
  fscanf(fp,"%c",&stuff);
  pixels[j][i] = Gray[stuff];              //convert to grayscale
 }

 for( j=54; j<1078; j+=4)
  { header[j] = header[j+1] = header[j+2] = (BYTE)((j-54)/4);   // 0..255 color entries
  header[j+3] = (BYTE) 0; }

 fclose(fp);
 strcpy(fname, path);
}

TRANSFIMG::TRANSFIMG(IMG* pOld)
{
 imgIn = pOld;
 height = pOld->cy;
 width = pOld->cx;
}

TRANSFIMG :: ~TRANSFIMG(void) { }

PROCIMG :: PROCIMG(IMG* pOld)        // create a clone image
{
  imgOut = new IMG(imgIn = pOld);
}

PROCIMG :: ~PROCIMG() { }

PROCAREA :: PROCAREA(IMG* pOld) : PROCIMG(pOld) { stack = (Node*) 0; }

PROCAREA::~PROCAREA(void) { delete area; }
```

A3. Dynamic list handling

```
/*****************************************************************************/
/*  Module:      list.cpp                                                    */
/*  Description:  Those member functions of MAPMAXATROUS, that               */
/*                        manipulate maxima, tree, ro-theta lists.           */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996 */
/*****************************************************************************/

#include "hierarchy.h"

void MAPMAXATROUS::Append(LocMax *p)      // append to maxlist
{

    if (Head == (Element*) 0)             // if empty list, create first entry as head
    {
      last=Head=new Element;
      last->next=(Element*) 0;
      last->elem=p;
    }
    else                                  // not empty, so look for last and append new
    {
      tmp=Head;
      while(tmp->next != (Element*) 0) tmp=tmp->next;

      last=tmp;
      tmp=new Element;
```

```
      last->next=tmp;
      tmp->next=(Element*) 0;
      tmp->elem=p;
      last=tmp;
    }
}

Object* MAPMAXATROUS::NewObject(LocMax* el, int nrlayer)
{
 int i;
 Object *tobj,*lastobj;

 if (HeadObj == (Object*) 0)            // no object yet in list, create first
 {
  tobj = HeadObj = new Object;
  HeadObj->next = (Object*) 0;
  for(i=0; i<4; HeadObj->layers[i++] = (Element*) 0);  // NULL layer sublists!
  Head = HeadObj->layers[nrlayer];
  Append(el);
  HeadObj->layers[nrlayer] = Head;
 }
 else                                   // append to end of list the new object
 {
  tobj = HeadObj;
  while( tobj->next != (Object*) 0) tobj = tobj->next;
  lastobj = tobj;
  tobj = new Object;
  for(i=0; i<4; tobj->layers[i++] = (Element*)0);   // NULL layer sublists!!
  lastobj->next = tobj;
  tobj->next = (Object*) 0;
  Head = tobj->layers[nrlayer];
  Append(el);
  tobj->layers[nrlayer] = Head;
 }
 return tobj;
}

void MAPMAXATROUS::AppendToCurrentObj(LocMax* el, Object* obj, int nrlayer)
{
 Element* HH;

 HH = Head = obj->layers[nrlayer];   // get object's <nrlayer> layer list head

 Append(el);                         //append new LocMax to on layer nr. <layer>

 if (HH == (Element*)0)
  obj->layers[nrlayer] = Head;       // if list was empty, write back head addr. !
}

void MAPMAXATROUS::SplitRegCurrentObj(Object* obj, int nrlayer)
{
 Element* HH;
 LocMax *temp;

 temp = new LocMax( 300,0,0,0,0);    //marking: x > 256 & region label = 0

 HH = Head = obj->layers[nrlayer];
 Append(temp);

 if(HH == (Element*)0)               // if list was empty, write back header addr.
   obj->layers[nrlayer] = Head;
}

RtList* MAPMAXATROUS::CreateRtList(void)
{
 int i;
 RtList *rthead, *rtp;

 rthead = rothetalist;
 if(rthead == (RtList*) 0)           //if no r-t list for any object, create first
```

```
 {
  rtp = new RtList;
  rtp->next = (RtList*) 0;
 for(i=0; i<3; i++) rtp->links[i] = (RtEntry*) 0;       // NULL links!!
  rothetalist = rtp;
 }
 else                                                   // list not empty, append new to end of list
 {
  while( rthead->next != (RtList*) 0) rthead = rthead->next;
  rtp = new RtList;
  rtp->next = (RtList*) 0;
 for(i=0; i<3; i++) rtp->links[i] = (RtEntry*) 0;       // NULL links!!
  rthead->next = rtp;
 }
 return rtp;
}

void MAPMAXATROUS::AppRoTheta(Polar *pol, int level, RtList* rtobj)
{
  RtEntry *temp, *pnt;

  temp = rtobj->links[level];              // get head of link list in object's r-t list
  if (temp == (RtEntry*) 0)
  {
   pnt = new RtEntry;
   pnt->next = (RtEntry*) 0;
   pnt->elem = pol;
   rtobj->links[level] = pnt;
  }
  else
  {
   while(temp->next != (RtEntry*) 0) temp = temp->next;

   pnt = new RtEntry;
   pnt->next = (RtEntry*) 0;
   pnt->elem = pol;
   temp->next = pnt;
  }
}
```

A4. A Trous transform and tree building

```
/*****************************************************************************************/
/*  Module :       atrous.cpp                                                            */
/*  Description:    ATROUS and MAPMAXATROUS classes' member functions,                   */
/*                              except the list manipulation functions.                  */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996        */
/*****************************************************************************************/

#include "hierarchy.h"

ATROUS::ATROUS(IMG* pim, int n) : TRANSFIMG(pim)
{
 int i;

 masksize = 5;
 h = new float[masksize];
 h[0] = h[4] = 1.0/16.0;
 h[1] = h[3] = 1.0/4.0;
 h[2] = 3.0/8.0;

 nrscales = n;

 for(i=0; i<nrscales; i++)
  di[i] = new IMG(height, width, 0);  //no header needed; saved in main.cpp
```

```
}

ATROUS::~ATROUS()
{
 int i;

 for(i=0; i<nrscales; i++)
  delete di[i];
}

void ATROUS::CompAtrous()
{
 IMG *c0,*c1,*temp;
 UINT i,j,k,m;
 UINT skip;
 int ind;
 float sum;
 float **t;

 c0 = new IMG(imgIn);          //alloc & copy input image
 c1 = new IMG(height, width, 0); //no header needed

 t = (float**) malloc( height * sizeof(float*));
 for(k=0; k<height; k++)
  t[k] = (float*) malloc( width * sizeof(float));

 for(i=0; i<nrscales; i++)
 {
  skip = 1<<i;
  for(j=0; j<height; j++)
   for(k=0; k<width; k++)
   {
    sum = 0.0;

    for(m=0; m<masksize; m++)
    {
     ind = (width + k + skip*(m-2))%width;
     sum += h[m]* c0->pixels[j][ind];
    }
    t[j][k] = sum;
   }

  for(k=0; k<width; k++)
   for(j=0; j<height; j++)
   {
    sum = 0.0;

    for(m=0; m<masksize; m++)
    {
     ind = (height + j + skip*(m-2))%height;
     sum += h[m]*t[ind][k];
    }
    c1->pixels[j][k] = sum;
   }

  for(j=0; j<height; j++)
   for(k=0; k<width; k++)
    di[i]->pixels[j][k] = c0->pixels[j][k] - c1->pixels[j][k];

  temp = c0; c0 = c1; c1 = temp;
 }
 ci = c0;

 delete c1;
 for(j=0; j<height; j++)
  free(t[j]);
 free(t);
}
```

```
MAPMAXATROUS::MAPMAXATROUS(IMG *im, int nn, char *fnm) : ATROUS(im, nn)
{
 int i,k;

 strcpy(fname, fnm);

 HeadObj = (Object*) 0;          // no objects in list

 for(i=0; i<6; i++)
 {
  maxlist[i] = (Element*)0;
  max[i] = (float**) malloc( im->cy * sizeof(float*));
  for(k=0; k<height; k++)
   max[i][k] = (float*) malloc( im->cx * sizeof(float));
 }

 rothetalist = (RtList *) 0;     // no ro-theta list
}

MAPMAXATROUS::~MAPMAXATROUS()
{
 int i,j;
 Element *pnt, *tmp;
 Object *obj, *tobj;
 RtEntry *pp, *tpp;
 RtList *rt, *trt;


 for(i=0; i<6; i++)
 {
   pnt = maxlist[i];
   while(pnt != (Element*)0)
   {
    tmp = pnt;
    pnt = pnt-> next;
    delete tmp->elem;
    delete tmp;
   }
 }

 obj = HeadObj;
 while(obj != (Object*)0)
 {
  for(i=0; i<4; i++)
  {
   pnt = obj->layers[i];
   while( pnt != (Element*)0)
   {
    tmp = pnt;
    pnt = pnt->next;
    delete tmp->elem;
    delete tmp;
   }
  }
  tobj = obj;
  obj= obj->next;
  delete tobj;
 }

 rt = rothetalist;
 while( rt != (RtList*) 0)
 {
  for (i=0; i<3; i++)
  {
   pp = rothetalist->links[i];
   while( pp != (RtEntry*) 0)
   {
    tpp = pp;
    pp = pp->next;
```

```
      delete tpp->elem;
      delete tpp;
     }
    }
    trt = rt;
    rt = rt->next;
    delete trt;
   }

  for(i=0; i<6; i++)
  {
    for(j=0; j<height; j++)
     free(max[i][j]);

    free(max[i]);
  }
}

void MAPMAXATROUS::MapMaxima(int fmax)   // fmax - flag: save float maxima as image?
{

  FILE *fp,*fls,*fout;
  int i,j,k,m,n;
  float fmx, fmn, pix, temp;
  int flag,size,equf,cx,cy, nrentries, nrn;
  char name[120], path1[120],path2[120], st[20];

  strcpy(path2, imgIn->fname); //max list
  strcat(path2, ".lst");

  fls = fopen(path2, "wt");
  printf("opening list file (%s) \n", path2);
  fprintf(fls, "[");   //open main list

  nrn = 0;

  for(i=6; i>2; i--)
  {
   nrnodes[i] = 0;              //first node will be 1, since nrnode is previously incr.

   fprintf(fls, "[");           //open sublist for this scale
   switch(i)
   {
     case 0: strcpy(st,"0"); break;
     case 1: strcpy(st,"1"); break;
     case 2: strcpy(st,"2"); break;
     case 3: strcpy(st,"3"); break;
     case 4: strcpy(st,"4"); break;
     case 5: strcpy(st,"5"); break;
     case 6: strcpy(st,"6"); break;
     case 7: strcpy(st,"7");
    }


    printf("looking for maxima at scale %d \n", i);
    for(j=0; j<height; j++)
     for(k=0; k<width; k++)
      max[i-2][j][k] = -99999.0; // a value that shouldn't show up

   size = 8;  //1<<i;  // second arg. => size = 2^arg.

   for(j=0; j<height; j++)
    for(k=0; k<width; k++)
    {
     flag = TRUE; //if not local max, then set false
     equf = TRUE; //if on local flat plain, leave true

// if centre pixel is not maxima then shift window and repeat test

      for(m=-size/2+1; m<=size/2-1; m++)
```

```
    {
     for(n=-size/2+1; n<=size/2-1; n++)
     {
      cx = (width+k+n)%width;
      cy = (height+j+m)%height;
// case 1: climbing hill
       if (di[i]->pixels[j][k] < di[i]->pixels[cy][cx] || (di[i]->pixels[j][k] == di[i]->pixels[cy][cx]
              && cy != j && cx != k)) //check if centre is biggest
       {
        flag = FALSE;       // there is something bigger then centre pixel's value
                            // break;
       }
// case 2: on flat plain
          if(di[i]->pixels[j][k] > di[i]->pixels[cy][cx] || di[i]->pixels[j][k] < di[i]->pixels[cy][cx] )
               equf = FALSE;

     } //end of n loop
     if (!flag) break;
    } //end of m loop
//end of tracking window loop

// case 3: found maximum
// now mark it
     if(flag && (!equf) && di[i]->pixels[j][k] > 0.0 )
      {  max[i-2][j][k] = di[i]->pixels[j][k]; }


  } // end k loop and end j loop

// now scale the values to pixel 0-255 range

    fmn = 500000.0;
    fmx = -500000.0;

     for(j=0; j<height; j++)
     for(k=0; k<width; k++)
      if(max[i-2][j][k] != -99999.0)  // we have a value with sense
      {
        if(fmx<max[i-2][j][k]) fmx = max[i-2][j][k];
        if(fmn > max[i-2][j][k]) fmn = max[i-2][j][k];
      }

// save maxima as image----------------------------------------

    if(fmax)
    {
    strcpy(name, imgIn->fname);
    strcat(name, ".bmp");
    strcat(name,st);        //saves as xxx.rasN.fmx !! (N is scale)
    strcat(name, ".fmx");
    fp = fopen(name, "wt");

    printf("writing maxima as image (%s) \n", name);

    for(j=0; j<imgIn->HeaderSize; j++)
     fprintf(fp, "%c", imgIn->header[j]);

    for(j=0; j<height; j++)
     for(k=0; k<width; k++)
     {
       if (max[i-2][j][k] == -99999.0) pix = 0.0;
         else pix = 255; //(max[i-2][j][k] - fmn)*255.0/(fmx-fmn);

       fprintf(fp, "%c", (BYTE)pix);
     }

    fclose(fp);
    }
```

```
 printf("save list of maxima for scale %d \n", i);

 //save maxima at this scale

  for(j=0; j<height; j++)
   for(k=0; k<width; k++)
    if(max[i-2][j][k]!=-99999.0)
    {
     nrnodes[i]++;
     nrn++;
     fprintf(fls,"[ %d %d %f %d ]",k,j,max[i-2][j][k],nrn); // x,y,magnitude
    }
   fprintf(fls, "]"); //close scale sublist
   fprintf(fls, "\n");

 } //i

 fprintf(fls, "]");  //close main list
 fclose(fls);

}

void MAPMAXATROUS::BuildTree(int **area, char **regions[])          //maxima tree
{
 int i,j,k, obj, nrnode, region, ind, xx,yy;
 LocMax *temploc, *item, *net, *newitem, *newnet, *node;
 Element *temp, *temp2;
 Object* object;

 obj = 0;
 nrobjects = 0;  //init data member

 nrnode = 1;   //startindex nodes

 for(i=6; i>2; i--)
 {
  Head = (Element*) 0;    // head of list is NULL for this layer

  for(j=0; j<256; j++)
   for(k=0; k<256; k++)
    if (max[i-2][j][k] != -99999.0 )
    {
     temploc = new LocMax(k,j,max[i-2][j][k], nrnode, 0);
     Append(temploc);
     nrnode++;
    }

  maxlist[6-i] = Head;    //layer 6 -> index 0!!
 }   //i

 temp = maxlist[0];
 while (temp != (Element*) 0)    // for every maximum in list , layer 1
 {
  item = temp->elem;
  temp->elem->label = 9999;    //mark it as processed

  j = item->y;
  k = item->x;

  region = regions[0][j][k];  // region label here on, layer 1

  newitem = new LocMax(item->x, item->y, item->magn, item->index, region);

  object = NewObject(newitem,0);
            //starts a new obj. sublist , with node on layer 1

  region = regions[1][j][k];  // region label above me

  if(region == 0)                // it's background => split region
     SplitRegCurrentObj(object,1);  //on layer 2
```

```
 else
 {
  temp2 = maxlist[1];  ind = 0;    // ind progresses as temp2 does!!!!!!
  while(temp2 != (Element*) 0)     // run through all maxima on layer2
  {
   net = temp2->elem;

   if(net->label != 9999)        // if wasn't marked
   {
     xx = net->x;
     yy = net->y;

     if(!(area[1][ind]<2 && area[1][ind] > 0))
      // small area, but non zero ! ( area = 0 marks a covered seedpoint)
     {
      if(regions[1][yy][xx] == region)
           // if this max. is in the same region as the up-projection
           // of the one in layer1, add to list
      {
        newnet = new LocMax(net->x, net->y, net->magn,
                        net->index, region);
       AppendToCurrentObj(newnet,object,1);    //layer2
       net->label = 9999;    //mark it as processed
      }
     }
   }

    temp2 = temp2->next;
    ind++;
  }    //while
 }     //if

 temp = temp->next;
}  //while

//now add layer 2 nodes that might be objects

 temp = maxlist[1]; ind =0;
 while(temp != (Element*) 0)
 {
  item = temp->elem;

  if (!((item->label == 9999) || (area[1][ind]<2 && area[1][ind]>0)) )
  {
     newitem = new LocMax(item->x, item->y, item->magn,
                     item->index, regions[1][item->y][item->x]);
     NewObject(newitem, 1);  // on layer 2 node, zip on layer 1 !!
  }
  ind++;
  temp = temp->next;
 }

 object = HeadObj;
 while(object != (Object*) 0)       // for every object in obj. list
 {
  temp = object->layers[1];
  while( temp != (Element*) 0)      //for every node on layer 2
  {
   node = temp->elem;
   if(node->x < 256)              // if not a split region node
   {
    region = regions[2][node->y][node->x];  // look on layer3

    if (region ==0)                 // if background, then
      SplitRegCurrentObj(object, 2);  //put split on layer 3
    else
    {
     temp2 = maxlist[2];  ind = 0;    // ind progresses as temp2 does!
     while(temp2 != (Element*) 0)     // run through all maxima on layer3
```

```
    {
      net = temp2->elem;

      if(net->label != 9999)
      {
        xx = net->x;
        yy = net->y;

        if(!(area[2][ind]<2 && area[2][ind] > 0) )
        {
          if(regions[2][yy][xx] == region)
          {
            newnet = new LocMax(net->x, net->y, net->magn,
                        net->index, region);
            AppendToCurrentObj(newnet,object,2);    //layer3

            net->label = 9999;   //mark it as processed
          }
        }
      } //if label != 9999

      temp2 = temp2->next;
      ind++;
    }    //while
   }    //if region==0
  }    //if not split region
  temp = temp->next;
 }    //while

object = object-> next;
} // for object in objects

// now add layer 3 nodes that might be objects

temp = maxlist[2]; ind =0;
while(temp != (Element*) 0)
{
  item = temp->elem;
  if (!((item->label == 9999) || (area[2][ind]<2 && area[2][ind]>0)) )
  {
     newitem = new LocMax(item->x, item->y, item->magn,
                  item->index, regions[2][item->y][item->x]);
     NewObject(newitem, 2); // on layer 3 node, zip on layer 1 and 2 !!
  }
  ind++;
  temp = temp->next;
}

object = HeadObj;

while(object != (Object*) 0)                  // for every object in obj. list
{
  temp = object->layers[2];
  while( temp != (Element*) 0)     //for every node on layer 3
  {
    node = temp->elem;
    if(node->x < 256)            // if not split region
    {
     region = regions[3][node->y][node->x]; // look on layer4

     if (region ==0)
       SplitRegCurrentObj(object, 3);     //put split on layer 4
     else
     {
      temp2 = maxlist[3]; ind = 0;    // ind progresses as temp2 does!
      while(temp2 != (Element*) 0)     // run through all maxima on layer4
      {
        net = temp2->elem;
        if(net->label != 9999)
```

```
      {
       xx = net->x;
       yy = net->y;

       if(!(area[3][ind]<2 && area[3][ind] > 0) )
       {
        if(regions[3][yy][xx] == region)
        {
         newnet = new LocMax(net->x, net->y, net->magn,
                             net->index, region);
         AppendToCurrentObj(newnet,object,3);    //layer4

         net->label = 9999;
        }
       }
      } //if
      temp2 = temp2->next;
      ind++;

     }    //while
    }     //if region==0
   }      //if not split region
   temp = temp->next;
  }   //while
  object = object->next;
 } // for object in objects
}

double MAPMAXATROUS::dist(int x1, int y1, int x0, int y0)
                        //x0,y0 - origin of vector
{
 return sqrt((x1-x0)*(x1-x0) + (y1-y0)*(y1-y0));
}

void MAPMAXATROUS::MapRoTheta(Element* layers[])    // redundant  fully connected rho-theta tree
              //works on the 4 layers of one object!
{

 int bg, fn, lxm2, lym2, lxm, lym, xm, ym, nrsrc, nrdest,count, i,j,jj;
 int index, indlargest;
 Element *tmp, *temp2;
 Polar* pol;
 float theta, ro, reftheta, magn, lmg, lmg2;
 RtList* rtobj;        // list head for a certain object's rotheta list

 rtobj = CreateRtList();  //create new ro-th list for new object

 //see on which layer the lists start & end (e.g., we can have [zip] (NULL)
 // on first and last layer!!

 bg=0;
 tmp = layers[0];
 while(tmp == (Element*) 0) tmp = layers[++bg];

 fn=3;
 tmp = layers[3];
 while(tmp == (Element*) 0) tmp = layers[--fn];

 for(i=bg; i<fn; i++)       //only on layers that have something (sublists)
 {
  tmp = layers[i];
  while(tmp != (Element*) 0)          // for every node on layer[i]
  {
   if(tmp->elem->x < 256)             // not split region node
   {
     xm = tmp->elem->x;
     ym = tmp->elem->y;
     magn = tmp->elem->magn;
    nrsrc = tmp->elem->index;         // becomes source node for link
```

```
  lmg = -10000.0;
  temp2 = layers[i+1];
  count = 0;

while(temp2 != (Element*) 0)        //scan next layer and get largest magn.
{
  if(lmg < temp2->elem->magn)
  {
    lmg = temp2->elem->magn;
    lym = temp2->elem->y;
    lxm = temp2->elem->x;
    indlargest = count;              // store number of the node that had largest
    }                                // magnitude in sublist for layer i+1
  count++;
  temp2 = temp2->next;
}

ro = dist(lxm, lym, xm, ym);         //get distance to this found node

if(ro == 0.0)                        // well, it's exactly above!
{
  temp2 = layers[i+1];
  for(j=0; j<indlargest; j++) temp2=temp2->next;   // so seek that node

  nrdest = temp2->elem->index;       //get its number in original maxlist

  pol = new Polar(nrsrc, 0.0, 0.0, nrdest);
  AppRoTheta(pol, i, rtobj);         // put it into ro-theta list with
                                     // ro=0, theta = 0 (convention)
  reftheta = 0.0;
  lmg2 = -10000.0;
  temp2 = layers[i+1];               // now search for second largest
  count = 0;                         // magn., if there are other nodes
                                     // on the layer above
  while(temp2 != (Element*) 0)
  {
   if(lmg2 < temp2->elem->magn && count != indlargest)
   {
     lmg2 = temp2->elem->magn;
     lym2 = temp2->elem->y;
     lxm2 = temp2->elem->x;
     index = count;      // keeep in mind the index of it
    }
    temp2 = temp2->next;
    }

  if(lmg2 != -10000.0)               // if !=, we found a second largest
  {                                  // (so there are more than one nodes)
     ro = dist( lxm2, lym2, xm, ym);
     if (ym-lym2 == 0.0 && xm-lxm2 == 0.0) reftheta = 0.0;
     else
        reftheta = atan2( ym-lym2, xm-lxm2)/3.1415926 * 180.0 + 180.0;

     temp2 = layers[i+1];
     for(j=0; j<index; j++) temp2=temp2->next;

     nrdest = temp2->elem->index;    // destination node for this link

     pol = new Polar(nrsrc, ro, 0.0, nrdest);
    AppRoTheta(pol, i, rtobj);          // put it in link list with theta = 0
    }
}
else     // ro != 0.0 => the node is not exactly above
{
    lxm2 = lxm; lym2 = lym;
    index = indlargest;                //so get ro, reference theta

    ro = dist( lxm2, lym2, xm, ym);
    if (ym-lym2 == 0.0 && xm-lxm2 == 0.0) reftheta = 0.0;
```

```
        else
          reftheta = atan2( ym-lym2, xm-lxm2)/3.1415926* 180.0 + 180.0;

        temp2 = layers[i+1];
        for(j=0; j<index; j++) temp2=temp2->next;

        nrdest = temp2->elem->index;  // destination node

        pol = new Polar(nrsrc, ro, 0.0, nrdest);   // ref. = 0.0 theta
        AppRoTheta(pol, i, rtobj);
    }

   temp2 = layers[i+1]; jj=0;              // ref. node being done, get others
    while(temp2 != (Element*) 0)
    {
      if( jj != index && jj != indlargest)
      {
        lym2 = temp2->elem->y;
        lxm2 = temp2->elem->x;

        ro = dist(lxm2, lym2, xm, ym);
        if (ym-lym2 == 0.0 && xm-lxm2 == 0.0) theta = 0.0;
        else
          theta = atan2(ym-lym2, xm-lxm2)/3.1415926 * 180.0 +180.0 - reftheta;

        if(theta < 0.0) theta += 360.0;
        if(theta > 360.0) theta -= 360.0;

        nrdest = temp2->elem->index;
        pol = new Polar(nrsrc, ro, theta, nrdest);
        AppRoTheta(pol, i, rtobj);
      }

      temp2= temp2->next;
      jj++;
    }
   } // if not split region
   tmp = tmp->next;
 }  //while
}   //for
}

void MAPMAXATROUS::LogNormTree(RtList* objrt)
{
 // log and normalise distances according to largest one in object tree
 int i,j;
 RtEntry *rt;
 Polar *pol;
 float maxlogro[3];

 for(i=0; i<3; i++)     //on every level of tree
 {
  rt= objrt->links[i];
  maxlogro[i] = -1000000.0;

  while (rt != (RtEntry*) 0)   // for every link on level
  {
   pol = rt->elem;
   printf("%f ", pol->ro);
   if (pol->ro != 0.0) rt->elem->ro = log(pol->ro);
     printf("%f \n", rt->elem->ro);
     if(maxlogro[i]<rt->elem->ro && rt->elem->ro != 0.0 ) maxlogro[i] = rt->elem->ro;

   rt=rt->next;
  }
 }

 //got overall nonzero maximum, do normalisation
 for(i=0; i<3; i++)     //on every level of tree
 {
```

```
    rt= objrt->links[i];
    while (rt != (RtEntry*) 0)   // for every link on level
    {
     pol = rt->elem;

     if (pol->ro != 0.0 && maxlogro[i] != 0.0) rt->elem->ro = pol->ro/maxlogro[i];

     rt=rt->next;
    }
   }
}
```

A5. Area processing

```
/*****************************************************************************************/
/* Module:       sobreg.cpp                                                              */
/* Description:  PROCAREA member functions - area processing.                            */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996        */
/*****************************************************************************************/

#include "hierarchy.h"

IMG* PROCAREA::Sobel(float thr)
{
  unsigned i,j,k,x,y;
  float res,strength;
  float sobmin, sobmax;

  for(y=1; y<255; y++)
   for(x=1; x<255; x++)
   {
    res = imgIn->pixels[y+1][x-1] + imgIn->pixels[y+1][x]*2 +
          imgIn->pixels[y+1][x+1] -
          imgIn->pixels[y-1][x-1] - imgIn->pixels[y-1][x]*2 -
          imgIn->pixels[y-1][x+1];

    strength = res<0.0 ? -res:res;

    res = imgIn->pixels[y-1][x+1] + imgIn->pixels[y][x+1]*2 +
          imgIn->pixels[y+1][x+1] -
          imgIn->pixels[y-1][x-1] - imgIn->pixels[y][x-1]*2 -
          imgIn->pixels[y+1][x-1];

    if (res<0.0) strength -= res;
     else strength += res;

   // thresholding only if threshold > 0; otherwise, scale everything down to BYTE
   if(thr > 0.0)
   {
    if (strength > thr) imgOut->pixels[y][x] = 255.0;
    else imgOut->pixels[y][x] = 0.0;
   }
   else imgOut->pixels[y][x] = strength;
  }

  if (thr == 0.0)   // scale it to BYTE
  {
    sobmin = 10000000.0;
    sobmax = -1000000.0;

    for(y=1; y<255; y++)
    for(x=1; x<255; x++)
    {
      if(sobmin > imgOut->pixels[y][x] ) sobmin = imgOut->pixels[y][x];
      if(sobmax <  imgOut->pixels[y][x] ) sobmax =  imgOut->pixels[y][x];
    }
```

```
    for(y=0; y<256; y++)
     for(x=0; x<256; x++)
     {
       if( x==0 || y==0 || x==255 || y==255)   //unconvolved border, clean it up
          imgOut->pixels[y][x] = 0.0;
       else    //not border, scale it properly
        imgOut->pixels[y][x] = 255.0*(imgOut->pixels[y][x] -
             sobmin)/(sobmax - sobmin);
     }
  }
  return imgOut;

}


void PROCAREA::Push(int x, int y)    //push a pixel position onto stack
{
  Node *p;

  if(stack==(Node*)0)
  {
   stack = new Node;
   stack->x = x;
   stack->y = y;
   stack->next = (Node*) 0;
  }
  else
  {
   p = new Node;
   p->x = x; p->y = y;
   p->next = stack;
   stack = p;
  }
}

void PROCAREA::Pop( int *x, int *y) //pop a pixel position from stack
{
  Node *p;

  p = stack;
  *x = p->x; *y = p->y;
  stack = p->next;
  delete p;
}

IMG* PROCAREA::Regions(int **seeds, int nrseeds, int *startlabel, float threshold)
{
  IMG* I;
  int i,x,y,X,Y, rnum,ofx,ofy,cx,cy;
  float thrs,maxmagn;

  I = new IMG(imgIn);   // copy to a new image, not to corrupt input
  rnum = *startlabel;

  for(y=0; y<256; y++)
   for(x=0; x<256; x++)
    imgOut->pixels[y][x] = 0.0;        // init region labels with 0 !

  area = new int[nrseeds];
  for(i=0; i<nrseeds; area[i++]=0);     // init areas with 0 for each seedpoint

  maxmagn = -100000.0;
  for(i=0; i<nrseeds; i++)
    if(maxmagn < I->pixels[seeds[i][1]][seeds[i][0]])
      maxmagn =  I->pixels[seeds[i][1]][seeds[i][0]];

  for( i=0; i<nrseeds; i++)
  {
    if( I->pixels[seeds[i][1]][seeds[i][0]] == -99999.0)
```

```
    {
      area[i] = 0;              // if seedpoint is marked, it means that it was covered
      rnum—;                    // by a previously born region; so mark this event with
    }                           // zero area, and keep region label constant !
  else
  {
    Push(seeds[i][0], seeds[i][1]);      //uncovered seedpoint, push it
    thrs=maxmagn*threshold;

    while (stack != (Node *) 0)
    {
      Pop( &X, &Y);
      area[i]++;
      if(imgIn->pixels[Y][X] > thrs)     //use ImgIn, since in I, this is marked as -999...
        imgOut->pixels[Y][X] = rnum;    //mark it on image only if it is relevant

      for(ofx = -1; ofx<2; ofx++)
       for(ofy = -1; ofy<2; ofy++)
       {
         cx = X+ofx;
         cy = Y+ofy;

          if(cx>=0 && cx<256 && cy>=0 && cy<256)
          if ( I->pixels[cy][cx] > thrs )        //was 0.0 , not thrs
          {
            Push(cx,cy);
            I->pixels[cy][cx] = -99999.0;     //mark it as processed
          }
       }
    }
   }
  rnum++;
 }

 delete I;
 *startlabel = rnum;        //return current region label value, so that
                            //it can be used by next call to this function
 return imgOut;

}
```

A6. The main module

```
/*****************************************************************************/
/* Module:    main.cpp                                                       */
/* Description: calls the functions that process the image, according to command line flags */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996 */
/*****************************************************************************/

#include "hierarchy.h"

void main(int argc, char* argv[])
{
 IMGRAST *Img;
 MAPMAXATROUS *Atr;
 PROCAREA* SobReg;

 char **regions[4];   // region maps on each layer
 int *area[4];        //areas on each layer, for each maxima

 FILE *fp,*fout;
 int i,j,k,m,n, indx, startreg;
 char name[120], path1[120], st[20], infname[120];
 float fmn, fmx, temp, thrs=0.0, regthrs = 0.0;
 int **seedlist;
 int sob=0,det=0,flt=0,fmax=0,reg=0, mapmx=0,sreg = 0, tree=0, roth=0;;     //flags
```

```
// ———————— printing help, if necessary ————————————————————

 if (argc < 2)
  {
   printf("Usage: \n mapmax fname option1 option2 ...\n ");
   printf("where options can be: \n");
   printf("  sob <threshold>   – does sobel on input image \n");
   printf("  flt             – saves float detail plane data as text\n");
   printf("  map               – maps maxima and saves list\n");
   printf("  fmx               – saves maxima as image (used only with map option!)\n");
   printf("  det               – saves detail planes as image\n");
   printf("  reg               – does region growing & saves images (used only with map!!\n");
   printf("  sreg              – saves region and area data ( region maps as bitmaps ) \n");
   printf("  thr               – threshold for region growing (optional)\n");
   printf("  tree              – does tree generation (used only with reg!)\n");
   printf("  roth              – ro – theta mapping (used only with tree!)\n");

   exit(1);
  }

// ———————— analysing options, setting flags ————————————————————
  i=2;
  while (i<argc)
  {
   if (strcmp(argv[i], "sob") == 0) { sob = 1; i++; thrs = atof(argv[i]); }
   else if (strcmp(argv[i],"thr") == 0) {i++; regthrs = atof(argv[i]);}
   else if (strcmp(argv[i], "det") == 0) det =1;
   else if  (strcmp(argv[i], "fmx") == 0) fmax = 1;
   else if (strcmp(argv[i], "flt") == 0) flt=1;
   else if (strcmp(argv[i], "reg") == 0) reg=1;
   else if (strcmp(argv[i], "sreg") == 0) sreg=1;
   else if (strcmp(argv[i], "map") == 0) mapmx=1;
   else if (strcmp(argv[i], "tree") == 0) tree=1;
   else if (strcmp(argv[i], "roth") == 0) roth=1;
   i++;
  }

// ———————————— load image ————————————————————————————————
  strcpy(infname,argv[1]);    //input image

  strcpy(name, infname);

  IMGRAST* inImg = new IMGRAST(name);

//———————————— exec according to flags ————————————————————

  if(tree)
   for(i=0; i<4; i++)
   {
    regions[i] = (char **) malloc(256 * sizeof(char*));
    for(j=0; j<256; j++)
     regions[i][j] = (char*) malloc(256 * sizeof(char));
   }

  if(sob)
  {
   printf("Computing Sobel\n");

   SobReg = new PROCAREA(inImg);
   IMG* img = SobReg->Sobel(thrs);

   strcat(infname, "sob");
   strcpy(path1, infname);
   strcat(path1, ".bmp");

   fout = fopen(path1, "wt");

   for(j=0; j<inImg->HeaderSize - 1024; j++)
    fprintf(fout, "%c", inImg->header[j]);    // initial image's header
```

```
  for(j=0; j<256; j++)
  {
    for( int jj =0; jj<3; jj++)
     fprintf(fout, "%c", (BYTE) j);
     fprintf(fout, "%c", (BYTE) 0);
  }  //write colortable 0——255, not messy !; white = 255

  for(j=0;j<256;j++ )
   for(k=0;k<256;k++)
    fprintf(fout, "%c", (BYTE)img->pixels[j][k]);

  fclose(fout);
  Img = (IMGRAST*)img;        // from now on, the filename has 'sob' in it!!
  strcat(Img->fname, "sob");

  delete SobReg;
  delete inImg;
 }
 else Img = inImg;

 printf("Computing Atrous transform \n");

 Atr= new MAPMAXATROUS(Img, 7, argv[1]);
 Atr->CompAtrous();
 if(mapmx) Atr->MapMaxima(fmax);

 startreg = 1;      // regions start label value

 for(i=6; i>=0; i--)
 {

  switch(i)
  {
    case 0: strcpy(st,"0"); break;
    case 1: strcpy(st,"1"); break;
    case 2: strcpy(st,"2"); break;
    case 3: strcpy(st,"3"); break;
    case 4: strcpy(st,"4"); break;
    case 5: strcpy(st,"5"); break;
    case 6: strcpy(st,"6"); break;
    case 7: strcpy(st,"7");
  }

 if(det)
 {
 strcpy(path1,infname);   //detail plane images
 strcat(path1, ".bmp");
 strcat(path1, st);

// save detail planes————————————————————————————

  printf("writing %s \n",path1);
  fout = fopen(path1, "wt");

  for(j=0; j<Img->HeaderSize; j++)
    fprintf(fout, "%c", Img->header[j]);

  fmx = -1000000.0;       //min, max for scaling
  fmn = 1000000.0;

  for(j=0;j<=255;j++)
   for(k=0;k<=255;k++)
   {
     temp = Atr->di[i]->pixels[j][k];

     if(fmx < temp) fmx = temp;
     if(fmn > temp) fmn = temp;
   }

  for(j=0;j<256;j++ )       //write scaled values to image file
```

```
 for(k=0;k<256;k++)
 {
   temp = (Atr->di[i]->pixels[j][k]-fmn)*255.0/(fmx-fmn);

   fprintf(fout, "%c", (BYTE) temp);
 }

 fclose(fout);
 }

if(flt)
{
 strcpy(path1, infname);
 strcat(path1, ".flt");
 strcat(path1, st);
 fout = fopen(path1, "wt");
 printf("Writing detail plane float data, plane %d\n", i);

 for(j=0;j<256;j++ )
  for(k=0;k<256;k++)
  {
   fprintf(fout, "%f ", Atr->di[i]->pixels[j][k]);
  }
 fclose(fout);
}

if(reg)          //region growing for current layer only
{
  if (!mapmx)
     { printf("maxima mapping should be performed!!!!\n"); exit(1); }
  else printf("Performing region growing\n");

 if(i>2)         //only for layers 1,2,3,4
 {
 seedlist = (int **) malloc( Atr->nrnodes[i] * sizeof(int*));
 for( j=0; j< Atr->nrnodes[i]; j++)
   seedlist[j] = (int*) malloc (2*sizeof(int));

 indx = 0;
 for(j=0; j<256; j++)      //prepare seedlist
  for(k=0; k<256; k++)
   if( Atr->max[i-2][j][k] != -99999.0)
    {
      seedlist[indx][0] = k;     //x of maximum
      seedlist[indx][1] = j;     //y of maximum
      indx++;
    }

 SobReg = new PROCAREA(Atr->di[i]);
 IMG* regs = SobReg->Regions(seedlist, Atr->nrnodes[i], &startreg, regthrs);

 area[6-i] = new int[Atr->nrnodes[i]];   //alloc area for current layer

 for(j=0; j<Atr->nrnodes[i]; j++)
              area[6-i][j] = SobReg->area[j];

 for(j=0;j<256;j++ )
  for(k=0;k<256;k++)
    if(tree) regions[6-i][j][k] = (BYTE) regs->pixels[j][k];

 // region growing threshold is default 0.0, if not set in command line
 if (sreg)
 {
 strcpy(path1, infname);
 strcat(path1, ".regs");
 strcat(path1, st);

 fout = fopen(path1, "wt");

 for(j=0; j<Img->HeaderSize; j++)
```

```
    fprintf(fout, "%c", Img->header[j]);    // initial image's header

// writes region map, if tree is true, copies them into big regions array
  for(j=0;j<256;j++ )
   for(k=0;k<256;k++)
   {
     fprintf(fout, "%c", (BYTE) regs->pixels[j][k]);
   }

  fclose(fout);

  strcpy(path1, infname);
  strcat(path1, ".areas");
  strcat(path1, st);

  fout = fopen(path1, "wt");              //write area data

  fprintf(fout, "%d " , Atr->nrnodes[i]);

  for(j=0; j<Atr->nrnodes[i]; j++)
    fprintf(fout,"%d ", area[6-i][j]);

  fclose(fout);
  } // save region data

   for(j=0; j<Atr->nrnodes[i]; j++)
     free(seedlist[j]);
   free(seedlist);

  delete regs;
  delete SobReg;

  } // if reg

 } // if i>2

 } //i - layer loop

 if(tree && reg)                          //reg must be done before tree!
 {
  Element  *tmp;
  LocMax* item;
  int count = 0;
  printf("Building tree\n");

  strcpy(path1, infname);
  strcat(path1, ".netxy");

 Atr->BuildTree(area,regions);            //for all 4 layers, using data
                                           //from region growing
  fout = fopen(path1, "wt");
  fprintf(fout, "[");

  Object* obj = Atr->HeadObj;
  while( obj != (Object*)0)
  {
    fprintf(fout, "[");                   //open object sublist

    for (i=0; i<4; i++)
    {
      tmp = obj->layers[i];
      fprintf(fout, "[");                 //open layer sublist
      if(tmp != (Element*) 0)
      {
        while(tmp != (Element*)0)
        {

         fprintf(fout, "[%d %d %f %d %d]",tmp->elem->x, tmp->elem->y, tmp->elem->magn,
tmp->elem->index,  tmp->elem->label);
         tmp= tmp->next;
```

```
      }
    }
    else
    {
                              // printf("layer %d zip\n", i);
      fprintf(fout, "[zip]");
    }

   fprintf(fout, "]");          //close layer sublist
   } //for
  obj= obj->next;
  count++;

  fprintf(fout, "]");           //close object sublist
 } //while

fprintf(fout, "]");
fclose(fout);

} // if

if (tree && reg && roth)
{
 RtEntry *rt;
 Polar *pol;
 RtList *rtlist;

 printf("Generating ro-theta map for each object\n");

 strcpy(path1, infname);
 strcat(path1, ".netrt");

 Object* object= Atr->HeadObj;

 while(object != (Object*)0)              //for each object, do rho-theta
 {
  Atr->MapRoTheta(object->layers);
  object = object -> next;
 }

 fout = fopen(path1, "wt");
 fprintf(fout, "[");

 rtlist = Atr->rothetalist;

  while(rtlist != (RtList*)0)
  {
   fprintf(fout, "[");          //open object sublist

   for(i=0; i<3; i++)
   {

    fprintf(fout, "[");         //open level sublist
    rt = rtlist->links[i];
    while (rt != (RtEntry*) 0)
    {
      pol = rt->elem;

       fprintf(fout,"[%d %f %f %d]",pol->src,pol->ro,pol->theta,pol->dest);
      rt=rt->next;
    }
    fprintf(fout,"]");          // close level sublist
   }
   rtlist = rtlist->next;
   fprintf(fout, "]");          // close object sublist
 } //while

fprintf(fout, "]");
fclose(fout);
```

```
 } // if tree, reg,roth

 printf("Deallocating memory\n");

 delete Atr;
 delete Img;

if (tree)
  for(i=0; i<4; i++)
  {
    for(j=0; j<256; j++)
      free(regions[i][j]);

    free(regions[i]);
    delete area[i];
  }

}
```

APPENDIX B.

Implementation of Coarse Data Channels

B1. Non–redundant connectivity tree building

```
function [rtlist] = centroid_tree(netxy)
%
% build centroid–mapped tree in order to calculate thetahisto
% it only works on 4 layers!!
% — Matlab v5.x —
%
% Input:   netxy structure from read_netxy
%
% Output:   centroid–mapped tree
%
% (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996
%

rtlist  = struct('src',[],'ro',[],'theta',[],'dest',[]);

REFTH = –10.0 ;

sznet = size(netxy);          % get size of all netxy struct

for obj = 1:sznet(1);         % for every subtree in netxy ( for every potential object )

indxx =  ones(1,5);           % start index memory for layers 1...

% get start and end layer in subtree  netxy(obj)

   bg = 0; fn = 0;

   for ll = 1:4 ;             % assume that very first node on a layer contains the [zip] label
    if netxy(obj, ll, 1).x ~= 0 | netxy(obj, ll, 1).y ~= 0 |
    netxy(obj, ll, 1).magn ~= 0.0 | netxy(obj, ll, 1).indx ~= 0 |
    netxy(obj, ll, 1).region ~= 0

        % then  it is not [zip]
        if bg == 0            % if bg was not set yet, set it – this is the beginning layer
           bg = ll;
        elseif  bg ~= 0       % if bg was already set and this is non–zip layer
           fn = ll;           % fn will remember the last non–zip layer => top of tree
        end;
    end;
   end;

   if fn == 0                 % if fn is still unset ( bg must be layer 4 then) , set fn equal to bg
      fn = bg;                % ergo:   one non–zip layer, usually happens on top one (4)
   end;



   indnode = 1;               % node indexes on future layers bg and bg+1 of tree
   indnode2 = 1;              % for present object


  if bg ~= fn & bg <= fn & fn > 1 & bg < 4

    maxx = –1000.0; ORIG = 1;

    lenlist = 0;
    % get nr. of nodes in layer bg of current tree
```

```
for linkindex = 1: sznet(3)          % for every potential link entry on layer bg
 if ~isempty(netxy(obj, bg, linkindex).x)
  lenlist = lenlist + 1;
 end;
end;


for l = 1:lenlist;          % get largest max on layer 1 ; scan all links

 if netxy(obj,bg,l).magn > maxx      %  if there are more than 1 maxima on layer 1...
    maxx = netxy(obj,bg,l).magn ;
    ORIG =netxy(obj,bg,l).indx ;
         Ox = netxy(obj,bg,l).x;
         Oy = netxy(obj,bg,l).y;
 end;

end;

if maxx ~= -1000.0                  %found node max

 maxx = -1000.0 ; indlargest = 1;

 lenlist2 = 0;                      % get length of list on layer bg+1
 for linkindex = 1: sznet(3)        % for every potential link entry on layer bg
  if ~isempty(netxy(obj, bg+1, linkindex).x)
   lenlist2 = lenlist2 + 1;
  end;
 end;

 for l = 1:lenlist2
  if netxy(obj, bg+1, l).magn > maxx              % if there is more than one...

     maxx = netxy(obj, bg+1, l).magn;
                        indlargest =  netxy(obj, bg+1, l).indx;
                        refx =  netxy(obj, bg+1, l).x;
                        refy =  netxy(obj, bg+1, l).y;
                      end;
 end;

 dx = refx - Ox;  dy = refy - Oy;
 ro = sqrt(dx*dx + dy*dy);

 if ro == 0.0   % exactly above root
  rtlist(obj, bg, indnode).src = ORIG;
  rtlist(obj, bg, indnode).ro = 0.0;
  rtlist(obj, bg, indnode).theta = 0.0;
  rtlist(obj, bg, indnode).dest = indlargest;
  indnode = indnode + 1;
  indxx(1) = indnode;               %correct start for this vector when it will be used

  if bg + 1 <= fn
   maxx = -1000.0;  indx2 = 1;

   for l = 1:lenlist2
    if netxy(obj, bg+1,l).magn > maxx  & netxy(obj,bg+1,l).indx ~= indlargest
     maxx = netxy(obj, bg+1, l).magn;
     indx2 = netxy(obj, bg+1, l).indx;
     refx = netxy(obj, bg+1, l).x;
     refy = netxy(obj, bg+1, l).y;
    end;
   end;

   if maxx ~= -1000.0               %found something not exactly above
    dx = refx - Ox; dy = refy - Oy;
    ro = sqrt(dx*dx + dy*dy);
    reftheta = (atan2(dy,dx) / pi) * 180.0 + 180.0 ;

    if REFTH < 0.0 &  bg == 1
       REFTH = reftheta;
    end;
```

```
    rtlist(obj, bg, indnode).src = ORIG;
    rtlist(obj, bg, indnode).ro = ro;
    rtlist(obj, bg, indnode).theta = 0.0;
    rtlist(obj, bg, indnode).dest = indx2;
    indnode = indnode+1;
    indxx(1) = indnode;              %correct start for indxx (when it will be used)

   else                              % nothing else there, so try above (one layer up)

    if bg+2 <= fn

     maxx = -1000.0;

     lenlist3 = 0;                   % get length of list on layer bg+2
     for linkindex = 1: sznet(3)     % for every potential link entry on layer bg
      if ~isempty(netxy(obj, bg+2, linkindex).x)
       lenlist3 = lenlist3 + 1;
      end;
     end;

     for l = 1:lenlist3
       if netxy(obj, bg+2, l).magn > maxx
       maxx = netxy(obj, bg+2, l).magn;
       indx2 = netxy(obj, bg+2, l).indx;
       refx = netxy(obj, bg+2, l).x;
       refy = netxy(obj, bg+2, l).y;
      end;
     end;

     dx = refx - Ox; dy = refy - Oy;
     ro = sqrt(dx*dx + dy*dy);
     reftheta = (atan2(dy,dx) / pi)*180.0 + 180.0 ;

     if REFTH < 0.0 & bg == 1
       REFTH = reftheta;
     end;

     rtlist(obj, bg+1, indnode2).src = ORIG;
     rtlist(obj, bg+1, indnode2).ro = ro;
     rtlist(obj, bg+1, indnode2).theta = 0.0;
     rtlist(obj, bg+1, indnode2).dest = indx2;
     indnode2 = indnode2 + 1;
     indxx(2) = indnode2;            % update index start in  indxx- when this will be used
                                     % to be correct start node

    end;  % if bg+2 <= fn
   end;   % if maxx ~= -1000.0
  end;    % if bg+1 <= fn

 else   % ro ~= 0.0

  reftheta = (atan2(dy, dx)/pi)*180.0 + 180.0;

  if REFTH < 0.0 & bg == 1
    REFTH = reftheta;
  end;

  indx2 = indlargest;

  rtlist(obj, bg, indnode).src = ORIG;
  rtlist(obj, bg, indnode).ro= ro;
  rtlist(obj, bg, indnode).theta = 0.0;
  rtlist(obj, bg, indnode).dest = indlargest;
  indnode = indnode + 1;
  indxx(1) = indnode;              % take from here index when indxx will be used,
                                   % don't write over some previous node

 end;   % if ro == 0.0

 for layer = bg+1 : fn
```

```
      lenlist = 0;
      % get nr. of nodes in layer 'layer' of current tree

      for linkindex = 1: sznet(3)          % for every potential link entry on 'layer'
        if ~isempty(netxy(obj, layer, linkindex))
          lenlist = lenlist + 1;
        end;
      end;


      for jj = 1:lenlist;

        if ~isempty(netxy(obj, layer, jj).indx)
          if netxy(obj, layer, jj).indx ~= indx2  &  netxy(obj, layer, jj).indx ~= indlargest

            dx = netxy(obj, layer, jj).x - Ox;
            dy = netxy(obj, layer, jj).y - Oy;

            ro = sqrt(dx*dx + dy*dy);

          if REFTH >= 0.0
            reftheta = REFTH;
          end;

          theta = (atan2(dy, dx)/pi)*180.0 + 180.0 - reftheta;
          if theta < 0.0
            theta = theta + 360.0;
          end;

          rtlist(obj, layer-1, indxx(layer-1)).src = ORIG;
          rtlist(obj, layer-1, indxx(layer-1)).ro = ro;
          rtlist(obj, layer-1, indxx(layer-1)).theta = theta;
          rtlist(obj, layer-1, indxx(layer-1)).dest = netxy(obj, layer, jj).indx;
          indxx(layer-1) = indxx(layer-1) + 1;

          end; %if
        end;  % if
      end ;   % for jj
    end;      % for layer

  end;   % if maxx ...
 end;    % if bg ~= ....

end;   % forobj
```

B2. Theta histogram computation

```
function [thisto] = thetahisto(centrt)
%
% compute theta histogram from centroid-mapped ro-theta tree
%
% Input:  one  netrt structure from centroid_tree function
%
% Output: one  theta histogram as 3 by 16 array ( 3 layers of links, 16 theta bins per layer)
% — Matlab v5.x —
% (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996
%

bins = 16;
phi = 360.0 / bins;

thisto = zeros(3, bins);

nsz = size(centrt);

for nrobj = 1:nsz(1);                          % every object ( every subtree, if there is any)
```

```
 for layer = 1:3;

  if ~isempty(centrt(nrobj, layer, 1).src)          % if layer not empty
                                                    % (if first node on layer not empty)

    lenlist = 0;
    for i = 1:nsz(3);                               % get length of non-empty node list
      if ~isempty(centrt(nrobj, layer, i).src)
        lenlist = lenlist+1;
      end;
    end;

    for i = 1:lenlist                               % for every node

      theta = centrt(nrobj, layer, i).theta;
      indx = round(theta/phi) + 1;

      if indx > bins    % just precaution - indx can not be > bins anyway, if all theta <= 360.0
        indx = bins;
      end;

      if centrt(nrobj, layer, i).dest ~= 0          % if not a split event , for example
        thisto(layer, indx) = thisto(layer, indx) + 1;
      end;

    end;
  end;
 end;
end;
```

B3. Rho–theta receptive fields

```
function [rfacts] = rothetamap(fname, layer, nr_ro, nr_theta, sigma)

%
% normalises all rho's with regard to a maximum ro found on a given layer
% and generates a map of Gaussian receptive-field activations.
% the RFs being placed on the norm(ro)–theta map of the given layer.
%
% input:   filename  - netxy file
%          layer     - where on the ro–theta tree to place RFs
%          nr_ro, nr_theta - nr. of RFs along norm(ro) and theta axes
%          sigma     - s.d. of Gaussian - can specify overlap for large values
%
% output:  array of nr_ro x nr_theta of float RF activations
% — Matlab v5.x —
% (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1997

netxy = read_netxy(fname);
centrt = centroid_tree(netxy);

rfacts = zeros(nr_ro, nr_theta);          % RF activations
centers = zeros(nr_ro, nr_theta, 2) ;     % RF grid points (centers of Gaussians)

% calculate Gaussians' centers

quantro = 1.0/nr_ro;
quantheta = 1.0/nr_theta;

for i = 1:nr_ro

  cntro = quantro/2.0 + quantro*(i-1);

  for j=1:nr_theta

    centers(i,j,1) = cntro;
    cntheta = quantheta/2.0 + quantheta*(j-1);
```

```
   centers(i,j,2) = cntheta;

  end;
end;

nsz = size(centrt);

for obj = 1:nsz(1)   % for every subtree

   if ~isempty(centrt(obj, layer, 1).src)
            % if layer not empty ( if first node on layer not empty – should do)

    lenlist = 0;
    for i = 1:nsz(3);                    % get length of non–empty node list
     if ~isempty(centrt(obj, layer, i).src)
      lenlist = lenlist+1;
     end;
    end;

    maxro = –1.0;
    for l = 1:lenlist                    % for every non–empty link entry
     if maxro < centrt(obj, layer, l).ro
       maxro =  centrt(obj, layer, l).ro;
     end;
    end;

    %normalise all ro's with maxro

    for l = 1:lenlist
     if maxro ~= 0.0
       centrt(obj, layer, l).ro = centrt(obj,layer, l).ro / maxro;
     end;
    end;

    % calculate RF activations

    for i= 1:nr_ro
     for j = 1:nr_theta;

       cntro = centers(i,j,1);           % center of Gaussian in norm(ro)–theta coords.
       cntheta = centers(i,j,2);

       for l =1:lenlist

         nodero = centrt(obj, layer,l).ro;
         nodetheta = centrt(obj, layer,l).theta;

         distro = nodero – cntro;        % distance from node to current hat's centre
         distheta = nodetheta/360.0 – cntheta;

          rfacts(i,j) = rfacts(i,j) + exp(–(distro*distro + distheta*distheta)/(sigma*sigma));

       end;
      end;
    end;

   end;  % if ~isempty
end;
```

B4. Junction extraction

```
;;; runs junction extraction on Aberdeen data
;;; using not edge map, but region map on seconf finest scale plane
;;; POPLOG–11
;;; (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996
```

```
5000000 -> popmemlim;


load junclib.p;              ;;; simplified junction detector from Ramsey & Barrett
load v1obj_lib.p

define JunctHisto( Im, Mx, My, pwsize ) -> histo;
lvars Im, Ox, Oy, pwsize, i,j, llist, lpic, junctypes;
lvars Mx, My, X,Y, prwnd, indx, juncts;

[ell tee fork kay arrow psi cross jn4 jn5 jn6 jn7 jn8 jn9 end]->junctypes;

newpwmrasterarray([ 1 ^(pwsize+1) 1 ^(pwsize+1)],8) -> prwnd;
newarray([ 1 ^(junctypes.length +1)], 0) -> histo;     ;;; types + 'other'

;;; work out offset in window sizes– where is the origin of the window
;;; centred on maximum (Mx, My) ??


for i from 1 to pwsize do;                ;;;copy window
  for j from 1 to pwsize do;

   i+Mx – pwsize/2 -> X;
   j+My – pwsize/2 -> Y;

   if X < 257 and X>0 and Y < 257 and Y>0 then
      Im(X,Y) -> prwnd(i,j);              ;;; if not inside image, leave 0!
   endif;

  endfor;
endfor;

findlines(prwnd) -> llist -> lpic;
findjunctions(llist, lpic) -> juncts;

for i from 1 to juncts.length do;         ;;; for every junction

  0 -> indx;
  for j from 1 to junctypes.length do;
    if juncts(i)(2) == junctypes(j) then  j -> indx;
    endif;
  endfor;

  ;;; if indx == 0  then jnN type junction, not in list —> 'other'
  ;;;  so indx  will be the index of junctypes.length +1 bin

  if indx == 0 then junctypes.length + 1 -> indx;
  endif;

  histo(indx) +1 -> histo(indx);

endfor;

enddefine;
;;; ——————————————————————————————————————————————————————

32 -> prwsize;    ;;; 32 by 32 pixel windows
4 -> nrlayer;     ;;; layer 4 maxima
'/weba/aberdeen/' -> datadir;

load /weba/aberdeen/rlist; -> rlist;      ;;;list of files

1 -> itemcount ;
for fname in rlist do;
  substring(1,2,fname) -> sss;
  fname==>;
  if sss =  'sp' then 'sprat/' -> fdir; endif;
  if sss = 'se' then 'seel/'->fdir;  endif;
  if sss ='he' then 'her/'->fdir; endif;

  sysobey('gunzip '><datadir><fdir><fname><'.ras.gz');
```

```
datadir><fdir><fname><'.ras' -> ffname;
mastrasterfile(ffname) -> Img -> header;
sysobey('gzip '><datadir><fdir><fname><'.ras');

 ReadObject(datadir><fdir><fname><'.v1') -> Object;
 Object(2)(2) -> netxy;

  outcharitem(discout(datadir><fdir><fname><'.junct4'))->charcon;
 if netxy(nrlayer) = [[zip]] then charcon(0);    ;;; special case: for [], write only 0
  else charcon(netxy(nrlayer).length);
 endif;

 charcon(space);                          ;;; nr. of histos
 charcon(15); charcon(newline);           ;;;bins

 if not(netxy(nrlayer) =  [[zip]]) then   ;;; only write if not empty layer
 for node in netxy(nrlayer) do;           ;;;for every maxima on layer nrlayer
  if node /= [zip] then

   JunctHisto(Img, node(1)+1, node(2)+1 , prwsize) -> jhisto;

   for kk from 1 to jhisto.length do;
    charcon(jhisto(kk));
    charcon(space);
   endfor;

   charcon(newline);
  endif;
 endfor;

 endif;
 charcon(termin);
 itemcount +1 -> itemcount ;
endfor;
```

B5. Spatial frequency extraction

```
#include <math.h>
#include <stdlib.h>
#include "hierarchy.h"

#define minim(a,b) ((a)<(b))?(a):(b)
#define maxim(a,b) ((a)>(b))?(a):(b)

extern void vrfft_alg(float ** , float ** );  // classic vector-radix 2D FFT

/*****************************************************************************************/
/* Module:    fourier.cpp                                                                */
/* Description: 2D FFT, collapse spectra                                                 */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996        */
/*****************************************************************************************/

class FOURIER : public TRANSFIMG
{
  protected:

   int specsize;          // the size of the spectrum
   int xfft, yfft;        // coords in the big input image where spect
                          //  will be calculated (centred on (xfft, yfft)!! )
   float max;             // max in power spectrum

   void SetupReIm();
   void DestroyReIm();

   void swapfloat( float*, float*);
```

```
public:

            float **DestRe;
            float **DestIm;
            float **spect;              // unscaled power spectrum

            FOURIER(IMG*, int);     // window size – class allows multiple spect calculation
            ~FOURIER();              // but not with different sizes!! then create other instance

            float** MyAllocFloat(UINT, UINT);
            int MyFreeFloat(float**, UINT);

            void VrFft(int , int);      // window's centre coordinates
            void CentreSpect(void);
            float* Collapse(float **); //collapses a 2D power spectrum

            void Spect();               //unscaled power spectrum ( 0 .. 255.0 )
            float **ScaleSpect();

};

float** FOURIER::MyAllocFloat(UINT y, UINT x)
{
 float**  buf=(float **)malloc(y*sizeof(float*));

 if(!buf)
   return (float **)NULL;

 for(int i=0; i<y; i++)
 {
   buf[i] = (float *) malloc(x*sizeof(float));
   if(!buf[i])
   {
    for(i-=1; i>=0; i—)
    free(buf[i]);
    return (float **)NULL;
   }
 }
 return buf;
}

int FOURIER::MyFreeFloat(float** buf, UINT y)

{
 for(UINT i=0;i<y;i++)
  free(buf[i]);

 free(buf);

 return 1;
}

void FOURIER::SetupReIm()
{
 int i,j;

  DestRe = MyAllocFloat(specsize, specsize);
  DestIm = MyAllocFloat(specsize, specsize);
}

void FOURIER::DestroyReIm()
{
  MyFreeFloat(DestRe, specsize);
  MyFreeFloat(DestIm, specsize);
}

void FOURIER::swapfloat(float *s,float *d)
{
  float t;
```

```
 t=*s; *s=*d; *d=t;
}

void FOURIER::CentreSpect(void)
{
 int i,j,cx,cy;

 cx = specsize/2; cy = specsize/2;

 for(i=0; i<cy/2; i++)
   {
    for(j=0; j<cx; j++)
     {
       swapfloat(&DestRe[i][j],&DestRe[cy-1-i][cx-1-j]);
       swapfloat(&DestIm[i][j],&DestIm[cy-1-i][cx-1-j]);
     }
    for(j=cx; j<specsize; j++)
     {
       swapfloat(&DestRe[i][j],&DestRe[cy-1-i][specsize+cx-1-j]);
       swapfloat(&DestIm[i][j],&DestIm[cy-1-i][specsize+cx-1-j]);
     }
   }

   for(i=cy; i<3*cy/2; i++)
   {
    for(j=0; j<cx; j++)
     {
       swapfloat(&DestRe[i][j],&DestRe[specsize+cy-1-i][cx-1-j]);
       swapfloat(&DestIm[i][j],&DestIm[specsize+cy-1-i][cx-1-j]);
     }

    for(j=cx; j<specsize; j++)
     {
       swapfloat(&DestRe[i][j],&DestRe[specsize+cy-1-i][specsize+cx-1-j]);
       swapfloat(&DestIm[i][j],&DestIm[specsize+cy-1-i][specsize+cx-1-j]);
     }
            }
}

void FOURIER::Spect()
{
 unsigned k,nc,nr,cols,rows;
 float temp, coef;
 float **mRe;
 unsigned long temp1;

 rows = cols = specsize;
 max=0.0;

 mRe = MyAllocFloat(specsize, specsize);

  for(nr=0;nr<rows;nr++)
   for(nc=0;nc<cols;nc++)
   {
     mRe[nr][nc] = temp = (float) sqrt(DestRe[nr][nc]*DestRe[nr][nc]+
                          DestIm[nr][nc]*DestIm[nr][nc] );

     if(max<temp)  max=temp;

   }


  for(nr=0;nr<rows;nr++)
  for(nc=0;nc<cols;nc++)
  {
   spect[nr][nc] = mRe[nr][nc];
  }
  MyFreeFloat(mRe, specsize);
```

```
}

float** FOURIER::ScaleSpect()
{
 int nr,nc;
 float **scp = MyAllocFloat(specsize,specsize);
 float coef=(float)255.0/(float)log((double)(1.0+max));

  for(nr=0;nr<specsize;nr++)
   for(nc=0;nc<specsize;nc++)
   {
     scp[nr][nc] = 0.5 + (float)log((double)(1.0+ spect[nr][nc]))*coef;
   }

  return scp;
}

void FOURIER::VrFft(int ccx, int ccy) // supports multiple calls for different (x,y)
{                                     // spectrum must be centred, and saved elsewhere
 unsigned i,j,xx,yy;                  // before another call to VrFft

 xfft = ccx; yfft = ccy;

 for(i=0; i<specsize; i++)
  for(j=0; j<specsize; j++)
  {
    xx = xfft+j-specsize/2;
    yy = yfft+i-specsize/2;

    if(xx >= 0 && xx<256 && yy>=0 && yy<256)
        DestRe[i][j] = (float) imgIn->pixels[yy][xx];
    else DestRe[i][j] = 0.0;   //clip outside image - put 0.0 instead of pixel

    DestIm[i][j] = 0.0;
  }

 vrfft_alg(DestRe, DestIm);
}

float* FOURIER:: Collapse(float **array)
{
   int i,xa, xz, ya, yz, xx, x, y, size, freqs;
   float *arrayout;

   freqs = specsize/2;
   arrayout = new float[freqs-1];        // no DC component
   for(i=0; i<freqs-1; i++) arrayout[i]=0.0;

   for(xx=1; xx<freqs; xx++)             // The first set
             {
    y =0; x= xx;                         //  of frequencies;
    while(x>=0)                          // ( top left corner)
    {
       arrayout[xx-1] = array[x][y] + arrayout[xx-1];
       y++; x--;
    }
   }

   for(xx = freqs+1; xx<specsize; xx++)  // The second set
   {
      y=0; x=xx;;             // of frequencies;
     while(x < specsize)                  // (top right corner)
     {
        arrayout[specsize-xx-1] = array[x][y] + arrayout[specsize-xx-1];
        y++; x++;
      }
   }

   for(xx=0; xx<freqs-1; xx++)           //The third set
   {
```

```
    y = specsize-1 ; x = xx;              // of frequencies;
    while(x>0)                            // (bottom left corner)
    {
      arrayout[xx] = array[x][y] + arrayout[xx];
      y—; x—;
    }
  }

  for(xx = freqs+2; xx<specsize; xx++)  // The fourth set
  {
    y = specsize-1; x = xx;               // of frequencies;
    while(x < specsize)                   // (bottom right corner)
    {
      arrayout[specsize-xx]  = array[x][y] + arrayout[specsize-xx];
      y—; x++;
    }
  }
  return arrayout;
}
```

B6. Texture density extraction

```
/*****************************************************************************************/
/* Module:     texture.cpp                                                                */
/* Description: extracts texture density measures based on calculated WT transform, local maxima map  */
/* (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1996        */
/*****************************************************************************************/

int *text = new int[16];              //texture histo
int nrlevels = 4;                     // levels from 0 upwards

spsize = winsz2;
nrcoords = 0;

  for  (rows=0; rows<256; rows ++)              //has as input an instance of the MAPMAXATROUS class
    for (cols=0; cols<256; cols ++)             // with the A Trous transform and local maxima maps
      {                                         // calculated previously.
        if (Atr->max[nplane2][rows][cols] != -99999.0)
        {
          nrcoords++;
        }
      }


  xx = new int[nrcoords];
  yy = new int[nrcoords];
  nrc = 0;

  for (rows=0; rows<256; rows ++)
      for (cols=0; cols<256; cols ++)
      {
        if (Atr->max[nplane2][rows][cols] != -99999.0)
        {
          xx[nrc] = cols;
             yy[nrc++] = rows;
        }
      }

    strcpy(path1, Img->fname);                  // the Img object contains the loaded image
     strcat(path1, ".text");

    fout = fopen(path1, "wt");
    fprintf(fout, "%d \n", nrcoords);

    int X,Y, dx, dy;
```

```
for (i=0; i<nrcoords; i++)                    // evaluate density of local maxima in fine scale layers
   {
    for (j=0; j<nrlevels; j++)
    {
        text[j] = 0;
        for(dx = -spsize/2+1 ; dx<=spsize/2-1; dx++)
          for(dy = -spsize/2+1; dy<=spsize/2-1; dy++)
          {
              X = xx[i] + dx; Y = yy[i] + dy;
              if (X<256 && Y<256 && X>=0 && Y>=0)
                { if (Atr->max[j][Y][X] != - 99999.0) text[j]++; }
          }
    }

    for (j=0; j<nrlevels ; j++)
      fprintf(fout, "%d ", text[j]);

    fprintf(fout, "\n");
   }

   delete xx;
   delete yy;

delete text;          // free coll, allocated inside Collapse funct.

fclose(fout);
```

APPENDIX C.

Implementation of Committee Machines

C1. CMC machine

```
% committee machine -
% most confident channel categoriser is chosen for decision.
% uses SNNS v4.1 result files from ANN testing.
% the present code exemplifies the committee machine implementation that uses
% all 4 channels, 3 split ratios and works on the 5-object synthetic data set.
% — Matlab v5.x —
% (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1997

clear all;

% define thetadir , juncdir , fftdir , texdir directories for coarse data files

Nruns = 20;

nrspl= [2 3 4];
nrhiddens = [4 6 8];

for splits = 1:3

 nrsplit = nrspl(splits);

 for hid = 1:3

 nrhid = int2str(nrhiddens(hid));

 trs = int2str(floor((1.0 - 1.0/nrsplit)*100));
 tes = int2str(floor(100* (1.0/nrsplit)));

tes
nrhid

 Ntr = floor(( 1.0 - 1.0/nrsplit)* 1800);          % how many training items
 Nte = floor(1800 * 1.0/nrsplit);                  % how many test

for i =1:Nruns ;

  categ = [];

  thetares = [ thetadir 'snns_' nrhid  'hte' tes 'shnotnorm_run' int2str(i) '.res'];
  juncres = [ juncdir 'snns_som' nrhid 'hte' tes 'shnotnorm_run' int2str(i) '.res'];
  fftres = [ fftdir 'snns_som' nrhid 'hte' tes 'shnotnorm_run' int2str(i) '.res'];
  texres = [ texdir 'snns_som' nrhid 'hte' tes 'shnotnorm_run' int2str(i) '.res'];

  unix(['/lisbon/public/gunzip ' thetares '.gz']);
  unix(['/lisbon/public/gunzip ' juncres '.gz']);
  unix(['/lisbon/public/gunzip ' fftres '.gz']);
  unix(['/lisbon/public/gunzip ' texres '.gz']);

  [ thetatargs, thetaouts ] = read_resfile(thetares, 48, Nte);
  [ junctargs, juncouts] = read_resfile(juncres, 30, Nte);
  [ ffttargs, fftouts] = read_resfile(fftres, 30, Nte);
  [ textargs, texouts] = read_resfile(texres, 30, Nte);

  unix(['/lisbon/public/gzip ' thetares]);
  unix(['/lisbon/public/gzip ' juncres]);
  unix(['/lisbon/public/gzip ' fftres]);
  unix(['/lisbon/public/gzip ' texres]);
```

```
% work out most confident

 for pat = 1:Nte;    % for every case

  % work out from target pattern the number of the object

   obj = find(thetatargs(pat,:) > 0);
   oo = find(junctargs(pat,:) > 0);
   ooo = find(ffttargs(pat,:) > 0);
   oooo = find(textargs(pat,:) > 0);

   if obj == oo & oo == ooo & ooo == oooo ;
    else
      disp('mismatch on object targets – file format error!');  exit;
   end;

   thout = thetaouts(pat,:);
  [val1, posth] = max(thout);          %largest activation and its pos
   thout(posth) = 0;                   % zero it
   val2 = max(thout);                  % second largest activation

   conftheta = val1–val2;

   jout = juncouts(pat,:);
   [val1, posjunc] = max(jout);        %largest activation and its pos
   jout(posjunc) = 0.0;                % zero it
   val2 = max(jout);                   % second largest activation

   confjunc = val1 – val2;

   ftout = fftouts(pat,:);
  [val1, posfft] = max(ftout);         %largest activation and its pos
   ftout(posfft) = 0.0;                % zero it
   val2 = max(ftout);                  % second largest activation

   confffft = val1 – val2;

   txout = texouts(pat,:);
  [val1, postex] = max(txout);         %largest activation and its pos
   ftout(postex) = 0.0;                % zero it
   val2 = max(txout);                  % second largest activation

   conftex = val1 – val2;


   % pick most confident and its verdict as final decision

   confid = [ conftheta confjunc confffft conftex];
   verdict = [ posth posjunc posfft postex];

  [highest, mostconfid] = max(confid);  % get highest confid and its poos in vector (which chan)

   decision = verdict(mostconfid);      % get corresponding verdict as final decision

   categ = [ categ;  obj decision mostconfid];  % save target, decision and nr. of most confident channel

  end;  % pat

  fff = fopen([texdir 'committee_decisions_' nrhid 'hte' tes '_run' int2str(i) '.dat'], 'w');
  for pat = 1:Nte
   fprintf( fff, '%d %d %d\n', categ(pat,1), categ(pat,2), categ(pat,3));
  end;
  fclose(fff);

end;   % Nruns

fclose('all');

 end;  % hid
end;  % splits
```

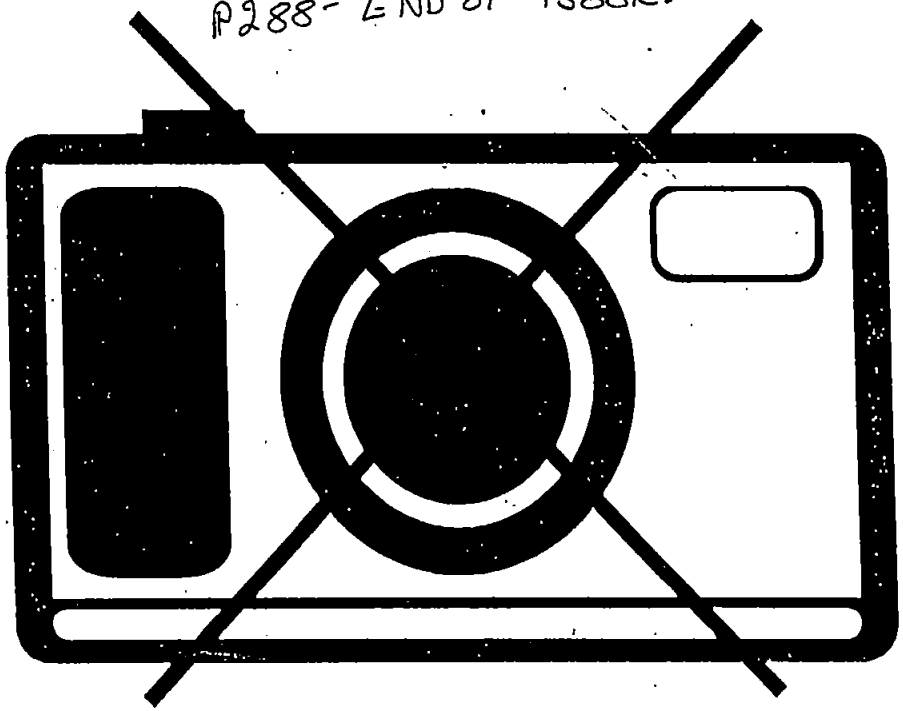
C2. CSO machine

```
% committee machine –
% sum of corresponding outputs of channel categorisers.
% uses all 4 channels, result files from SNNS simulations (tests) used as input.
% present version works on the 5–object synthetic data set.
% — Matlab v5.x —
% (C) Levente Toth, Centre for Intelligent Systems, University of Plymouth, 1997

clear all;

% define thetadir , juncdir , fftdir , texdir directories for coarse data files

Nruns = 20;

nrspl= [2 3 4];
nrhiddens = [4 6 8];

for splits = 1:3

 nrsplit = nrspl(splits);

 for hid = 1:3

 nrhid = int2str(nrhiddens(hid));

 trs = int2str(floor((1.0 – 1.0/nrsplit)*100));
 tes = int2str(floor(100* (1.0/nrsplit)));

tes
nrhid

 Ntr = floor(( 1.0 – 1.0/nrsplit)* 1800);          % how many training items
 Nte = floor(1800 * 1.0/nrsplit);                  % how many test

for i =1:Nruns ;

  categ = [];

  thetares = [ thetadir 'snns_' nrhid  'hte' tes 'shnotnorm_run' int2str(i) '.res'];
  juncres = [ juncdir 'snns_som' nrhid 'hte' tes 'shnotnorm_run' int2str(i) '.res'];
  fftres = [ fftdir 'snns_som' nrhid 'hte' tes 'shnotnorm_run' int2str(i) '.res'];
  texres = [ texdir 'snns_som' nrhid 'hte' tes 'shnotnorm_run' int2str(i) '.res'];

  unix(['/lisbon/public/gunzip ' thetares '.gz']);
  unix(['/lisbon/public/gunzip ' juncres '.gz']);
  unix(['/lisbon/public/gunzip ' fftres '.gz']);
  unix(['/lisbon/public/gunzip ' texres '.gz']);

  [ thetatargs, thetaouts ] = read_resfile(thetares, 48, Nte);
  [ junctargs, juncouts] = read_resfile(juncres, 30, Nte);
  [ ffttargs, fftouts] = read_resfile(fftres, 30, Nte);
  [ textargs, texouts] = read_resfile(texres, 30, Nte);

  unix(['/lisbon/public/gzip ' thetares]);
  unix(['/lisbon/public/gzip ' juncres]);
  unix(['/lisbon/public/gzip ' fftres]);
  unix(['/lisbon/public/gzip ' texres]);

 % work out sum of outputs and winner

  for pat = 1:Nte;    % for every case

   % work out from target pattern the number of the object

   obj = find(thetatargs(pat,:) > 0);
   oo = find(junctargs(pat,:) > 0);
```

```
  ooo = find(ffttargs(pat,:) > 0);
  oooo = find(textargs(pat,:) > 0);

  if obj == oo & oo == ooo & ooo == oooo ;
   else
     disp('mismatch on object targets');  exit(0);
  end;

  thout = thetaouts(pat,:);
  jout = juncouts(pat,:);
  ftout = fftouts(pat,:);
  txout = texouts(pat,:);

  sumouts = thout + jout + ftout + txout;

  % pick largest sum as decision

  [highest, decision] = max(sumouts); % get highest sum  and its pos in vector (which object)

  categ = [ categ;  obj decision ];        % save target, decision

 end;  % pat

 fff = fopen([texdir 'committee_sumout_decisions_' nrhid 'hte' tes '_run' int2str(i) '.dat'], 'w');
 for pat = 1:Nte
  fprintf( fff, '%d %d\n', categ(pat,1), categ(pat,2));
 end;
 fclose(fff);

end;   % Nruns

fclose('all');

end;  % hid
end;  % splits
```

PUBLISHED PAPERS NOT FILMED FOR COPYRIGHT REASONS

P288- END OF Book.



APPENDIX D.

Publications

3D OBJECT RECOGNITION FROM STATIC 2D VIEWS USING MULTIPLE COARSE DATA CHANNELS

Levente Toth
Phil F. Culverhouse

Centre for Intelligent Systems,
School of Electronic, Communication and Electrical Engineering,
University of Plymouth, Plymouth PL4 8AA, UK.
Tel., Fax: +44 1752 233517
email: pculverhouse@plymouth.ac.uk

Abstract

A 3D object recognition system is described that employs novel multiresolution representation and coarse encoding of feature information. Modifications are brought to classic feature extraction methods by proposing the use of wavelet transform maxima for directing the actions of feature extraction modules. The reasons behind the use of a multi–channel architecture are described, together with the description of the feature extraction and coarse coding modules. The targeted field of application being automatic categorisation of natural objects, the proposed system is designed to run on ordinary hardware platforms and to process an input in a short timeframe. The system has been evaluated on a variety of 2D views of a set of 5 synthetic objects designed to present various degrees of similarity, as being rated by a panel of human subjects. Parallels between these ratings and the system's behaviour are drawn. Additionally a small set of photomicrographs of fish larvae has been used to assess the system's performance when presented with very similar, non–rigid shapes. For comparison, the parameters extracted from each image were fed into two categorisers, discriminant analysis and multilayer feedforward neural network with backpropagation of error. Experimental evidence is presented which demonstrates the efficacy of the methods. The satisfactory categorisation performances of the system are reported, and conclusions are drawn about the system's behaviour.

Keywords: viewer–centred representation, A Trous transform, coarse coding, discriminant analysis, neural networks, self–organising maps.

1. Introduction

Computational studies of vision in the past decades have highlighted the complexity of processes involved in performing visual tasks and the inherent difficulty of building computational systems that perform 3D object recognition and scene comprehension. A series of computational studies and theories reviewed by Hildreth & Ullman [21] describe vision as a chain of processes that, based on the retinal image, yield increasingly complex representations of the visible world. Among the theories on visual information representation in biological systems (the work of Marr [28], Biederman [2], Ullman [43][45] and Edelman & Weinshall [16]), at the present the ones discussing viewer–centred representation seem to have more experimental evidence in their support (Tarr & Pinker [42], Edelman & Bulthoff [15]). The experimental results reviewed by Edelman [13] contradict also the theories centred around the idea of representation with reconstruction [28]. Hence there has been a gradual shift in vision research from theories postulating the necessity of using extremely detailed, often complete representations of the world (the work of Freuder, Tenenbaum & Barrow cited in [28]), towards more relaxed frameworks based on viewpoint–dependent encoding of views and storage of multiple views (as suggested by Ullman & Basri [45]).

Arriving at this fundamental problem of representing the visible world, an important task in the design of a recognition system is the selection of descriptors that would constitute the building blocks of the internal representation of the analysed objects. Still, the virtual impossibility of obtaining a universally valid set of features and categorisation criteria based on these has been pointed out by studies in the field of taxonomy. Sokal [41] highlighted the existence of individual differences in taxonomic judgement, since a group of human classifiers can arrive at correct categorisation of objects based on quite different sets of features considered to be salient.

In analysing images for object recognition purposes, researchers usually have tended to focus on object and image properties that are also salient to human enquiry. Thus texture descriptions [47], edge positions [8] and statistical descriptions of pixel densities [20] have all been used to segment images into their component parts. Categorisation follows, providing object recognition. Most of these methods rely on extracting very precise measurements of, for example, symmetries or shape description (as illustrated by methods proposed by Brady [6], Khotanzad & Liou [23]). None have proved reliable analysis tools for understanding natural images or images with noise and clutter obscuring the objects of interest. As an alternative, a method has been developed by Ellis *et al.* [17] that draws on the concept of Ullman's multiple visual routines [44]. The principle of operation is the registration at low resolution of multiple parameters that describe the object scene in an image. If many of these 'coarse channels' are analysed in concert a solution to the particular analysis may be found – one which may not be apparent when using high resolution data. This is similar in concept to finding a global minimum in a multidimensional descriptor space so often described in artificial neural network research (a good example being Rumelhart & McClelland's work [36]). The coarse channel principle has been applied successfully to the automatic categorisation of 23 species of field collected marine plankton, in a system developed by Culverhouse *et al.* [11]. It is also applied here to the task of three dimensional object recognition.

This approach of non–exact feature description and low–resolution encoding of features also constitutes the central concept of other recently developed systems that do not necessarily employ multiple data channels. Bradski and Grossberg [7] describe a system that uses in its preprocessing stage an array of Gaussian receptive fields in order to decrease the dimensionality of the data. The system developed by Mel [29] employs a large array of filters placed on the input image, the outputs being coarse coded as histograms for achieving viewpoint–invariance. In a conceptually related way, Schiele & Crowley [37][38] have used multidimensional receptive field histograms characterising 2D views of objects in classification and in determination of favourable viewpoints for recognition. Edelman's Chorus scheme [14] utilises a receptive field array that provides low–dimensional description of the input data. The main difference between these approaches and the authors' system is that the attention of the system is directed by a module employing Mallat's [26] multiresolution analysis (MRA) towards areas of the image that contain potentially relevant features for categorisation. Therefore it does not analyse the entire surface of the input image (e.g. by placing a large array of receptive fields on the image).

The proposed system constitutes an engineering solution, since the processing algorithms were designed to run on largely available hardware platforms and to perform analysis in a sufficiently short timeframe that makes it usable in laboratory conditions.

2. Overview of the system

The recognition system has three components: (i) a multi–resolution feature extractor that uses wavelet filter banks, (ii) a coarse channel feature analyser and (iii) an object categoriser. Features are defined in this context as areas of high contrast or high curvature, the extraction of these being directed by low–resolution information, following work on visual inspection through eye tracking by Niemann *et al.* [30] and Rao *et al.* [32]. The spatial organisation of these features is analysed through multiple low resolution data channels. In this paper the authors explore the superposition of three channels, which are described in depth in the next section of the paper. The first channel is based on spatial angle descriptions in scale–space that lead to so–called theta histograms; this channel uses a proposed multiscale representation of shapes. Another channel introduces descriptions of edge coterminations into the system, the junction data being collected from locations chosen automatically from MRA data. The resulting junction histograms, when propagated through a self–organising Kohonen map, produce a node activation pattern which is then used as feature data by the classifier. Local spatial contrast relationships are extracted in the third data channel,

in the form of FFT spectra, again propagated through a Kohonen map. Object labelling, or categorisation, completes the system. In this paper the performance reported by discriminant analysis is compared with the performance of feed forward neural networks trained with error backpropagation algorithm.

The general structure of the system is shown in Fig. 1.. Multiresolution analysis has been implemented by using wavelet transform. The classic discrete wavelet transform though, with all its disadvantages due to downsampling (most importantly, the lack of translation invariance) is not suitable in pattern recognition applications, a point supported by Mallat [27] as well. An alternative is offered by the A Trous transform described by Dutilleux [12] that does not employ downsampling of the data, it is translation invariant and it is an exact implementation of the discrete wavelet transform (as proven by Shensa [39]). The properties of the A Trous transform in comparison with other wavelet transforms are described in the next section of the paper.

——————— Fig. 1 to go here ———————

A multiscale coarse representation of 2D views of 3D objects is the core of the first coarse data channel. Hierarchic lists of link vectors represented in scale–space are calculated from sets of tree structures that record the relationships between regions of positive wavelet coefficients and wavelet local maxima. The information on the orientations in scale–space of links between the leaves situated on successive scale planes of maxima trees are coarse–coded into theta histograms.

This scale–space representation is related in its essence to other multiresolution tree structures found in the literature. Systems based on multiscale tracking of corners and edges (like the ones developed by Rattarangsi & Chin [33], Hsieh *et al.* [22]), contours or segments (Ren *et al.* [34], Liu & Yang [25]) all attempt to explicitly label events (e.g. singularities) in scale–space. The present approach takes a different route, by employing much more relaxed rules when generating the scale–space representation. Such generic wavelet maxima trees have been used recently for detecting and grouping of point source and extended astronomical objects, as the work of Rue & Bijaoui [35] illustrates. The method proposed in the present paper is designed for 3D objects, taking into account relationships between wavelet local maxima and regions of significant coefficients in order to encode, in multiple tree–like structures, the spatial organisation of points of interest in the image. The resulting representation is essentially a polar description of potentially relevant object features in scale–space.

The other coarse coding modules employed in the system extract and encode junction and spatial frequency information. As it has been pointed out by Biederman [3], the types of junctions created by edge coterminations are non–accidental properties that can provide information on object shape in a viewpoint–invariant way. Showing the feasibility of such recognition approaches based on junction types, these features have been extensively used in the interpretation of line drawings and views of origami objects. An example would be the work of Kanade (reported in Ballard & Brown [1]). In the proposed system, the lower resolution information available in the wavelet decomposition directs the extraction of junction information (after Ramsay & Barrett [31]). Instead of analysing the whole shape of a given object's view, these are being directed by low–resolution information obtained from the wavelet decomposition. Hence features are extracted only from areas centred on wavelet maxima detected on a sufficiently coarse coefficient plane. The spatial frequency channel works in a similar way, extracting FFT power spectrums in neighbourhoods of these wavelet maxima. These spectra are coarse coded into 1D feature vectors, that provide information on the spatial frequency content of the analysed areas. The junction and FFT data obtained from these areas are presented to self–organising maps. Hence for a given image, the succession of junction histograms, for example, leads to an activation pattern of the nodes of the considered map. This provides a signature vector for each of the analysed objects, used later in classification. This mechanism and the rationale behind it is detailed in the following sections of the paper.

In categorisation experiments, two different types of classifier have been used, discriminant analysis and feedforward artificial neural networks, which have allowed a robust evaluation of the preprocessing methods introduced above.

3. The preprocessing and coarse coding methods

In this section the mathematics and algorithms behind the feature extraction and coarse coding methods are described, with emphasis on the novel way of representing and encoding the scale–space topology of wavelet transform's local maxima.

3.1. The wavelet transform

The multiresolution information used by the first data channel for building the scale–space representation is provided by a module that computes the wavelet transform (WT) of the image. This module's output, as it has been mentioned before, is also used in other coarse data channels for directing the feature extraction process.

The employed wavelet transform, with regard to the targeted application area, had to have the following properties:

i.) *Translation invariance.* In any pattern analysis application using discrete wavelet transform (DWT), it is essential to have a transform that, when presented with an input pattern shifted with a certain amount relative to the pattern's original position, produces a set of coefficients shifted with the same amount (as it is described by Mallat [27]). If this condition is not satisfied, then the transform leads to an entirely different set of coefficients when presented with a translated version of its input. In mathematical terms, and in a 1D case, this condition can be written as

$$DWT_{j,k}[f(x-\tau)] \quad = \quad DWT_{j,k-\tau}[f(x)] \tag{1}$$

where $DWT_{j,k}[f]$ is the discrete wavelet transform of a function f at scale j and position k, τ being the amount of translation. The classic DWT used in multiresolution analysis (described by Mallat [26], Shensa [39]) does not exhibit this property, due to the downsampling of the data during the transform. But non–decimated versions [39], although redundant and overcomplete, do have this property. The A Trous algorithm described by Dutilleux [12], or wavelet packet transform (DWPT) with suitable choice for basis functions (Cohen *et al.* [10]) lead to transforms that increase the data storage requirements, but provide a translation–invariant decomposition.

ii.) *Correspondence between input pixels and wavelet coefficients.* Since the system analyses each wavelet (detail) plane individually, it was necessary to choose a wavelet transform that for a given resolution j generates only one detail coefficient for a given pixel in the input image. If this is not satisfied by the WT, then analysis of a particular detail plane is not possible without taking into account several other coefficient planes. Starting with an image plane, the classic DWT delivers 3, the DWPT produces 4 sets of coefficients on each detail plane. This makes close scrutiny of the coefficients difficult from a computational point of view, and accurate detection of events like singularities (e.g. local maxima) on a given resolution level becomes impossible, as it is pointed out in Bijaoui *et al.* [5]. The A Trous algorithm does meet this requirement, that is a one–to–one mapping between a pixel and a corresponding wavelet coefficient on a specific resolution plane is possible.

iii.) *Reasonable computational load.* The A Trous transform's mathematics allow the use of extremely relaxed design conditions for the involved smoothing and detail filters (as pointed out by Bijaoui *et al.* [5]), hence it is possible to reach an implementation that is acceptable for real–time processing.

Considering the above, the choice was to use the A Trous transform in the module responsible for multiresolution analysis. This transform leads to an exact sampled version of the continuous wavelet transform, without the necessity of initial approximations of the input data at the finest resolution plane, as it happens in the case of classic DWT (fact demonstrated by Shensa [39]).

The fundamental property of the A Trous algorithm is that the involved smoothing (low–pass) filter h is an interpolating filter. In the discrete case, as the corresponding scaling function dilates, zeros are inserted among the samples of the function. The mathematical implications of this and of the non–orthogonality of the transform are treated by Shensa [39]. The name of the algorithm ("with holes") comes exactly from the fact that the filter h interpolates between samples: leaves the even points on each scale fixed, and obtains the odd points by interpolating – hence no downsampling occurs. In all subsequent descriptions of

this section, the finest resolution (detail) plane will have the lowest index.

The continuous form of the transform can be written as

$$W_{j,k}[f(x)] = \frac{1}{\sqrt{2^j}} \int \overline{\psi}\left(\frac{x-k}{2^j}\right) f(x)dx \tag{2}$$

where j is the scale parameter and k the translation, $\overline{\psi}$ being the complex conjugate of the wavelet function. The scaling function used in the transform, that leads to the wavelet is

$$\phi_{j,k}(x) = \frac{1}{2^j}\phi\left(\frac{x-k}{2^j}\right) \tag{3}$$

Its essential property is that a set of h_m , $m \in Z$ coefficients exists, such that :

$$\frac{1}{2}\phi(\frac{x}{2}) = \sum_m h_m \phi(x-m) \tag{4}$$

This is the *dilation equation* – also called the two–scale difference equation, since it expresses the relationship between two scaling functions on two consecutive resolutions (scales). Having the scaling function known explicitly, the smoothing filter coefficients h_m can be obtained from the dilation equation.

From this, one can proceed to the calculation of the outputs of the smoothing filter on a given scale, using the data from the previous scale. The smoothed coefficients are calculated as it follows:

$$c_{j+1,k} = < f(x), \phi_{j+1,k} > = \sum_m h_m c_{j,k+2^j m} \tag{5}$$

In the case of the A Trous algorithm, the initial c_{0k} coefficients are equal to $f(x)$, without any initial approximation being necessary. The detail (wavelet) coefficients result in a similar fashion:

$$d_{j+1,k} = \sum_m g_m c_{j,k+2^j m} \tag{6}$$

where g_m is the impulse response of the detail filter. The latter is usually calculated as a quadrature mirror filter of the smoothing filter, but in the case of the A Trous algorithm (due to the relaxed filter design conditions) it can be calculated as in Bijaoui *et al.* [5]:

$$g_m = \delta_m - h_m \tag{7}$$

where δ_m is the discrete Dirac impulse (with $\delta_0 = 1$ and 0 in rest). Based on this equation, the detail filter coefficients can be calculated as a difference of smoothed coefficients of successive resolution planes, without using scalar product:

$$d_{j,k} = -\sum_{m \neq 0} h_m c_{j-1,k+2^j m} + (1-h_0)c_{j-1,k} = c_{j-1,k} - c_{j,k} \tag{8}$$

For the implementation of the 2D version of this A Trous algorithm, the 3rd–order B–spline was chosen

as scaling function, since it leads to computationally attractive smoothing and detail filters (as demonstrated in Bijaoui *et al.* [5][4]). The smoothing filter has a zero of order 4 at $z=-1$. The impulse response of both filters has only one positive maximum. One of the coarse coding methods being based on local maxima of detail filter responses, the latter characteristic proves to be essential in object detection and feature localisation. Having this choice for scaling function, the resulting wavelet in its 1D form shows attractive similarities with the response of X–type cells in monkey visual cortex, fact pointed out by Unser & Aldroubi [46]. The coefficients and the frequency characteristics of the smoothing and detail filters extended to 2D are shown in Fig. 2.

——————— Fig. 2 to go here ———————

3.2. Maxima trees and theta histograms

All images submitted to analysis are preprocessed to remove the DC component. This is followed by the computation of wavelet planes using the A Trous transform. The local maxima of the transform are detected on a number of coarse wavelet planes. Since positive coefficients mark areas of interest in the image, region growing is performed on these wavelet planes, using as seedpoints the detected local maxima and as decision criterion the sign of the coefficients. Having obtained a multiresolution map of the maxima and of the regions, a set of maxima trees are generated that record the way in which the configuration of maxima, in correlation with the region maps, changes from coarse to fine scale planes. Since large variations in a viewed object's size leads to important modifications in the structure of the maxima tree, this coarse–coding method will not exhibit complete scale invariance. A detailed description of the algorithm can be found below.

In the context of maxima trees, the first level of the trees corresponds to the coarsest detail plane, and it has the smallest index (j=1). The wavelet coefficient plane j is denoted W_j and the set of maxima detected on W_j is M_j. In this new context, the coarsest wavelet plane is W_1. The set of root nodes of the trees can be obtained according to equation (9):

$$B = \{m \mid m \in M_1\} \bigcup \{m \mid m \in R \wedge (P(m', W_j) \notin R,\ \forall m' \in M_{j-1}),\ \forall R \subset \Gamma_j\ ,\ \forall m \in M_j\ ;\ j = \overline{2, N_L}\ \} \tag{9}$$

where Γ_j is the region map on plane j, $P(m, W_j)$ is the orthogonal projection of a maximum m onto the wavelet coefficient plane W_j, N_L is the number of levels. The root nodes in B are the starting points for the algorithm that builds a tree for each root node. The tree T_t for each root is generated by the following algorithm:

```
t = 1;
for each b ∈ B
    for j = 1 to N_L
        if b ∉ W_j  then  T_tj = {φ}  else  T_tj = {b},  n = j
    endfor
    T_tj = T_tj ⋃ {m | m ∈ R ∧ (∃m' ∈ T_tj-1 : P(m', W_j) ∈ R), ∀R ⊂ Γ_j ,
                   ∀m ∈ M_j } ,  j = n+1, N_L
    t = t+1;
endfor
```

T_{tj} signifies level j of the tree T_t and $\{\phi\}$ denotes the empty set. Using the maxima trees, corresponding hierarchic link lists can be generated. For each tree, the algorithm moves from the root towards the leaves

on the finer resolution planes present in the considered tree. A set of link projection vectors are generated for each T_t tree, according to (10):

$$L_{t,j-1} = \begin{cases} \{\phi\} , & \forall q < j \ : \ b \notin T_{t,q} \\ \{\vec{l} = (a,\vec{m}) \mid a = P(b, W_j), \ \forall m \in T_{t,j} \subset W_j \ \}, & \exists q < j \ : \ b \in T_{t,q} \end{cases} \quad (10)$$

where b is the root of T_t, $j = \overline{2, N_L}$. The final step is to obtain the theta histogram; N_L–1 histograms of relative orientation angles of links found in all hierarchic L_t lists are computed. The reference orientation (θ^t_{ref}) is taken from the lists L_t. For a certain L_t, θ^t_{ref} is the orientation angle of the vector l that satisfies $\| \vec{l} \| \neq 0$ (non–zero norm in 2D Euclidian space) and links the projection of the root to the largest maximum found on the first level above the root of the corresponding T_t tree. At this stage, the relative θ values are collected from all link lists:

$$\Theta_j = \{ \arg(\vec{l}) - \theta^t_{ref} \mid \forall \vec{l} \in L_{t,j}, \forall t \ \}, \ j = \overline{1, N_L - 1} \quad (11)$$

On each Θ_j set a histogram is calculated. In the present tests, 16 bin histograms were used on each level of links (i.e. a resolution of 22.5 degrees) to coarse code angular information.

With regard to the 2D translation invariance of this coarse coding module's output, a crucial role is played by the translation invariance of the A Trous wavelet transform. The position on each scale plane of the local maxima of the transform will reflect the 2D translations of an object' view in the analysed frame. Hence the maxima trees calculated relative to the root nodes chosen according to equation (9) will exhibit translation invariance.

3.3 Junction histograms

The second coarse data channel provides information on edge junctions detected in the image. As opposite to classic approaches that perform an analysis of the whole shape presented as input, the extraction of junction information is restrained to the areas of interest in the image, identified by the MRA module described in section 3.1. For each neighbourhood of wavelet maxima (located on a sufficiently coarse coefficient plane), a junction histogram is produced.

The following classic junction types were taken into account: *L, T, K, arrow, Psi, fork, cross, end* (as termination of segment), *jnN* (N=4..9 – when N segments meet) and *'other'* for those that are not in any of the previously listed categories. An example of an edge map, the points of interest marked by wavelet maxima resulted from MRA of the image and the contents of the processing windows centred on these are shown in Fig. 3.

Fig. 3 to go here

Since the number of junction histograms extracted from each image differs due to the variable number of detected wavelet maxima on the selected coefficient plane, the junction histograms are not used directly as feature data. Instead, these are propagated through a 5x6 node Kohonen self–organising map and the summed activation patterns of the nodes are used as a fixed–length feature vector. The set of junction histograms obtained from each image of a model data set is used for training a Kohonen self–organising map. Following training, a histogram of node activations is generated by propagating through the trained map the junction data obtained from a set of test images. The algorithm of the training stage is described below, using the notations introduced in previous sections:

for each image I ∈ training_set

```
  input T_t                                          ; read all trees generated for this image
  for each m ∈ T_t,j, ∀t                             ; all maxima on layer j of each tree
    extract junctions around P(m, I)  →  H           ; obtain a junction histogram
    H  →  train_SOM                                  ; present histogram to Kohonen map
  endfor
endfor
```

In testing regime, when an image I is presented to the system for categorisation, the junction channel performs the following actions:

```
input T_t                                                          ; read all trees generated for test image
0 → ActivationSignature[30];                                       ; init node activation signature vector
for each m ∈ T_t,j, ∀t                                             ; all maxima on layer j of each tree
  extract junctions around P(m, I)  →  H                           ; obtain a junction histogram
  H → test_SOM → NodeActivations                                   ; obtain a map of node activations
  if NodeActivations[i] > 0.5 then ActivationSignature[i]++ , i = 1,30
endfor
```

The output vector named *ActivationSignature* is used as feature vector, and presented to the classifier module in chorus with the vectors supplied by the other coarse data channels. As it is apparent in the above pseudocode, only SOM node activations above a positive threshold are taken into account.

3.4 Spatial frequency channel

Rotation invariant encoding of FFT spectrums is used in the implementation of the the spatial frequency coarse data channel, in order to provide information on the spatial frequencies in areas of interest on the image. These FFT spectra are collected from locations marked by wavelet transform maxima, instead of the classical approach of computing a spectrum that characterises the whole object. The 2D shape of the object's view being viewpoint–variant, it is necessary to extract spatial frequency information in designated areas of the image that are likely to contain surface boundaries, edges etc.

The 2D power spectra calculated in the processing windows are collapsed into a feature vector, following work reported in Simpson *et al.* [40]. The training parameters were kept the same as in the case of the junction channel. The training stage and the way of obtaining the node activation signatures follow in a similar way the steps described in section 3.3. The node activation signatures are used as feature data characterising the object's view presented as input to the system.

4. Classification experiments

In order to evaluate the MRA/coarse–coded data channel image analyser, three test data sets were used. The results were fed into two categorisers for training and testing. An 8–object and a 5–object data set comprised computer–generated 2D views of 3D objects. The Aberdeen data set held multiple 2D views of natural images of fish larvae. Images in the Aberdeen set were typically of much poorer image quality than the first two sets of images. The 8–object data set was used to evaluate the theta histogram feature channel only. In conditions of relatively small variations in viewpoint (as it is described in section 4.1), this data set allowed a robust evaluation of the shape descriptor capabilities of the theta histogram channel. In tests involving wide variations of viewpoints, the 5–object and the Aberdeen data sets were used to evaluate the performance of the whole 3–channel system. In the sections below, the used data sets and the categorisation experiments are described.

4.1 The test images

Both synthetic data sets held 256x256 pixel images. The 3D scene contained omnidirectional and ambient light sources. The same light source and camera setup was used for all image rendering. In order to minimise the effect of scaling on the wavelet transform maxima and on the resulting connectivity tree, the objects were constrained to the same height and radius (where geometry allowed). All objects had the same matte surface texture. These constraints were to minimise the dissimilarities between objects (apart from their shapes). During preprocessing, the rendered views were converted to 8–bit grey level images that constituted the input to the system. Both data sets are available at *http://www.cis.plym.ac.uk/cis/3Darchive.html* for downloading.

The objects in the 8–object data set were rotated about the longitudinal (Z) axis with +/– 10 degrees relative to their reference position showed in Fig. 4.; a step of 2 degrees was used. The rotation about the Z axis produced self–occlusion of the asymmetric elements present in the structure of objects No. 7 and 8. The objects in their reference position were also rotated about the X and Y axis with +/– 10 degrees. Rotation in depth, hence foreshortening of the objects occurred; 15 views per object have been generated following this protocol.

——————— Fig. 4 to go here ———————

To assess the clustering of the feature vectors that encode theta histograms, only discriminant analysis (DA) was used, hence no training/test sets were generated based on this data set.

For the trials involving training and test data sets, a set of images of 5 synthetic objects was generated, all objects having elements that would present self–occlusion when viewed from certain camera positions. Two objects were designed to be very similar (objects No.1 and 2), so that discrimination would be very difficult from certain viewpoints. While having the same height and radius, the main difference between these two objects is the presence of a cavity in the case of object No.1. Object No. 5 was intentionally synthesised to be very different from the rest of the set. The 5 objects are shown in Fig. 5.

——————— Fig. 5 to go here ———————

The similarity of the 5 objects has been assessed by a panel of 10 human subjects who were presented with views of all 5 objects. They were asked to mark the similarities between objects on a 5–point scale (following Elmes *et al.* [18]). The mean ratings are listed in Table 1.

——————— Table 1. to go here ———————

A set of views was generated by animating a simulated camera that spanned the upper viewing hemisphere. The camera moved along 36 paths located at 10 degrees azimuth from each another, on each such path the step in elevation angle (spanning 0...90 degrees interval) was 10 degrees– hence 360 images per object were generated. This set of images has been split into two subsets for classification experiments involving training and test data. From the numbered sequence of views resulted from the camera animation, every 4th image became part of the test set, while the rest of the images were kept in the training set. Therefore 270 images per object were made available in the training set, and 90 images per object in the test set.

The Aberdeen data set contained photomicrographs of fish larvae. The images were of herring, sprat and sandeel larvae in 4 developmental stages, they have been chosen because of the particularly difficult task of discriminating between them. The specimens' similarity, their non–rigid elongated shapes, morphological variations occurring at different stages of their development and factors related to image quality present a series of problems for any classifier, whether it is an expert human taxonomist or an automatic system. The photomicrographs show the larvae in dorsal and lateral views, specimens also appearing twisted. Examples of each species of larvae are presented in Fig. 6.

Fig. 6 to go here

The images are corrupted by detritus present in the water sample, which in some cases appears to be attached to the larvae. A further problem is the presence of untypically shaped larvae in the data set, especially in their early stages of development (herring was particularly affected) and the shape of some specimens appears to be altered (e.g. older sandeels in a dorsal view). The images have uniform background, are histogram equalised and the images of detritus (various particles, planktons etc.) have been eliminated from the photomicrographs. These were stored as separate images, being labelled as a 4th category of objects (those that in a real–world situation must be recognised by the system as objects to be ignored). For each type of larvae, 50 images were available. With the above mentioned separation of detritus, 1562 images were obtained in the 4th category. All are 256x256 pixel 8–bit grey level images.

4.2 Tests on categorisation

In this section, the classification experiments involving statistical and artificial neural network–based categorisers are described. All tests involving discriminant analysis (DA), as statistical classifier, have been performed using the SPSS statistical software package. All experiments involving feedforward neural networks relied on the shareware Stuttgart Neural Network Simulator (SNNS).

The first feature extractor producing theta histograms has been used in preliminary tests meant to evaluate the ability of the maxima trees to record salient features of 2D views of 3D shapes and also, to test whether a simple classifier can discriminate between a set of 3D shapes by relying only on this data. The 8–object data set was suitable for this purpose, the views presenting phenomena like foreshortening and self–occlusion of the viewed objects. DA performed on the set of theta histograms obtained from the images in the 8–object data set has shown a 100% match between predicted and actual group memberships. This confirmed the expectations towards the theta histogramming method: it is able to provide distinguishable signatures for the 8 objects (some of them presenting geometric similarities) and also compensates for relatively small foreshortening (18% on vertical vertices) and partial self–occlusion.

Using all images of the 5–object data set, DA tests were performed in order to assess the clustering of data vectors provided by each coarse channel when the analysed sets of views capture essentially different aspects of the 5 objects. Some of these objects, image number 1 and 2 in Fig. 5 for example have distinctive features visible only from certain viewpoints. Hence the set of images has been split into 8 subsets according to 8 regions of the upper viewing hemisphere. Each region has been chosen to have a 90 degrees azimuth and 45 degrees elevation angle span. The 8 regions of the viewing hemisphere will be referred to as M.N, where M denotes the 90 degrees azimuth interval (I...IV) and N denotes the lower/upper 45 elevation angle interval (I,II).

The mean classification accuracy reported by DA in each of the 8 regions of the viewing hemisphere for each of the three coarse data channels is shown in Fig 7. The percentages of matches between the predicted and actual group memberships show that all of the employed coarse data channels can characterise with satisfactory accuracy the analysed shapes. For any given channel, the difference between the obtained mean recognition accuracy in regions x.I and x.II is less than 6%.

Fig. 7 to go here

The discriminant analysis trials employing model and test sets were run using within–groups covariance. The mean DA classification accuracies on 5 objects obtained based on the model and test images are summarised in Fig. 8. The performance of the statistical classifier presented with data provided by associated coarse channels registered significant improvement compared to the recognition accuracies obtained based on individual channels' data.

Fig. 8 to go here

For a comparison, recognition experiments were conducted with artificial neural network–based classifiers that employed 3–layer feedforward networks with 6 hidden nodes (chosen after optimising network

structure). The networks have been trained with backpropagation algorithm involving momentum term. Based on preliminary trials, learning rate was set to 0.01, momentum term to 0.96 and the flat spot elimination constant was 0.02. For each configuration of coarse data channels, 20 training/testing runs were performed, using 3000 training epochs in each run. In each of the 20 runs, the weights of the network have been initialised with random values. During each training epoch, the patterns in the training data set have been presented to the network in random order. Fig. 9 shows the minimum, maximum and mean recognition accuracies achieved on the test data set in the 20 runs.

Fig. 9 to go here

In order to assess the agreement between the ratings produced by the system and the groups (objects) the input data belongs to, the kappa statistic was used (after Cohen [9]). This provides a more refined measure of the system's performance, by taking into account both the observed agreement and the expected chance of agreement. Also, since in its calculation the non–diagonal terms of the confusion tables are included as well, performances that stand out (the case of object 5, for instance) do not introduce strong bias as it happens when reporting mean accuracy. The z–score associated with kappa gives a measure of kappa's significance (as described by Fleiss [19]), hence in the sets of 20 runs performed with neural networks, the mean overall kappa values across all categories and their corresponding z value have been recorded.

These values are reported in Table 2. Kappa values over 0.4 are considered good agreement beyond chance (after Landis & Koch [24]), hence Table 2. shows that the overall kappa values in all trials represented very good agreement beyond chance, with large z scores showing high statistical significance.

Table 2. to go here

The mean recognition rates in the case of each object, obtained by the neural network on the test data in 20 runs when using different coarse channel combinations are shown in Fig. 10.

Fig. 10 to go here

It is apparent, that the data channel containing junction information has a crucial role in improving the ability of the classifier to distinguish objects No. 1 and 2. The main difference between these two objects' shapes lies in the extra cavity in the case of the first object and the different geometry of the asymmetric element present in the structure of both objects. These differences led to distinguishable signatures in the junction data, while the other channels that characterised more general properties of the objects could only lead the classifier to a significantly poorer performance in discriminating the two objects. The use of the theta histogram channel led to a mean performance for objects No. 1 and 2 of 51.3% and 62.3%, respectively. The mean accuracy obtained by using only the FFT channel remained close to 50% (52.1% and 51.4% respectively). But in the case of the junction histogram channel, the ANN–based categoriser's performance in distinguishing between the two similar objects stayed above 65%. When associating this channel with the theta and FFT information, the system's mean performance for objects 1 and 2 increased with an additional 10% compared to the mean accuracy obtained when using junction channel only. Also, object No.5, exhibiting the most dissimilar shape in the considered set of objects, was easily identified by the classifier in all cases of channel combinations. The mean accuracy stayed above 90%. For an evaluation of the connections between object shape similarity (as assessed by human subjects) and the system's behaviour, the confusion tables obtained in 20 runs for the chorus of 3 channels were used. The mean percentages of confusion between the 5 objects (calculated from the sum of symmetric non–diagonal terms of the confusion tables) are listed in Table 3.

Table 3. to go here

It is apparent, that there is a strong connection between the highest similarity of objects 1 and 2 judged by humans and the extent of the system's confusion between these. Also, object 5 judged to be the most dissimilar is the least confused with any of the other objects. Objects No. 3 and 4, with similarities rated

around mid–range of the scale by human subjects did not produce such a consistent pattern of confusion when compared with Table 1.

The mean accuracies obtained with statistical and neural network–based categorisers (Fig. 7 and 8) show a less than 4% difference in the average performance of the two types of classifiers. Although the neural networks tend to slightly outperform the statistical classifier in the case of individual channels, the mean accuracy of the ANN–based categoriser falls 3–4% under the average performance of discriminant analysis when presented with grouped channels. No far reaching conclusions can be drawn from these minor differences, but the aspect of data pruning is to be noted. The DA–based categoriser automatically prunes histogram bins that are unpopulated in all of the presented cases, hence only the relevant data is kept in the analysis. With the increase in dimensionality of the input space (i.e. grouping of channels), the number of empty histogram bins can increase – it is likely that the performance of ANN–based classifiers trained on such data can be improved by employing weight pruning algorithms.

The above described tests allowed the evaluation of the role played by each data channel in the the system under conditions of controlled viewpoint changes, object shapes and scene parameters. Following these trials, the Aberdeen data set has been submitted to the system for classification. Due to the particularities of the data set (described in the previous section), a close–to–chance performance was expected. It was clear that with the pronounced morphological variations of the specimens in question, a small training set can only lead to poor performance in the context of a small data set (50 images per type of fish larvae). This behaviour in the case of such field–collected images of biota has been studied in Simpson *et al.* [40]. Hence the data set containing images of larvae was split into training and test sets in a 80/20 % ratio. This meant a number of 40 images of larvae in the training set and 10 in the test set. The same number of detritus images was kept in the training set, while the rest of 1522 images of detritus were inserted into the test set. The detritus has been included in the data set to have a clearly dissimilar group, in order to draw parallels with the synthetic images (there object No. 5 created a clear cluster). Since the detritus is very different in shape and size, it was expected from the system to perform well on this particular group even when presented with a very small training set compared to the number of images available for this group.

The mean classification accuracies (based on the diagonal percentages of the confusion tables) achieved by the statistical classifier (DA) on the model and test sets are shown in Fig. 11.

——— Fig. 11 to go here ———

As a comparison, discriminant analysis of data describing physical characteristics of the fish larvae (body size) reported a 20% mean accuracy on the test set (well below chance), when having the same 80%–20% split of data into model and test sets. The DA results in a form of a confusion table are given in Table 4.

——— Table 4. to go here ———

Categorisation trials using artificial neural networks were carried out on this data set, following the experimental protocol described above for the synthetic images. The mean accuracies obtained from 20 runs on the test data set are plotted in Fig. 12.

——— Fig. 12 to go here ———

In these experiments, too, the grouping of data channels had a positive effect on the system's performance. The ANN–based classifier clearly outperfomed the statistical method in this case. It can be noted, that the junction channel's discriminating power decreased in comparison with the test run on synthetic data. This behaviour did not constitute a surprise, since the object shapes were extremely similar in this case. The ANN–based classifier's mean performance across 20 runs for each of the fish larvae (herring, sprat, sandeel and detritus group) is shown in Fig. 13.

——— Fig. 13 to go here ———

Categorisation of herring larvae based on individual coarse channel data is approximately at chance level (25%), with the exception of the FFT channel. The strongest confusion occurred between herring and sprat, hence the recognition accuracy for the herring group in the 20 runs ranged between 10% and 40%. The detritus has been correctly identified in all cases of channel configuration, while the categorisation accuracy on sandeel and sprat larvae increased as channels have been added. Once again, in order to obtain a more accurate measure of performance than the mean accuracy (that can be biased by groups performing well above others and only takes into account the observed agreement), the kappa statistic was used. The mean overall kappa values and their significance measures are listed in Table 5.

——— Table 5. to go here ———

In the case of this data set, kappa actually assesses the agreement between the automatic system and the expert taxonomists who labelled each image as herring, sandeel, sprat or detritus. It can be seen that in the case of individual channels, there is satisfactory agreement between experts and the system – as more channels are added, the agreement improves. As it is now clearly reflected in the mean kappa scores, the junction channel delivers the poorest performance. This is probably due to the flexible nature of the specimens and the fact that all three fish larvae exhibit similar elongated shapes. Since tests were run on a small data set, an increase in performance would be expected with an increase in the size of the data set.

5. Conclusions

A three dimensional object recogniser has been presented that operates on coarse coded data obtained from multi–resolution analysis of 2D views of 3D objects. The system has been tested on a variety of synthetic objects, some of which present self occlusion during rotation. The implemented feature extraction and coarse coding techniques led to good results in classifying views of similar synthetic 3D objects, in conditions of wide variations of viewpoint. Also, in the case of a difficult set of field–collected natural images of fish larvae, the results were markedly better than discrimination based on body size measurements and the system exhibited a consistent behaviour compared to tests on synthetic data.

As a practical aspect, all processing stages in test regime for a single image took on average less than 4 minutes to run on a SUN Sparc 5 workstation (32 Mb RAM). This timeframe is adequate for practical situations, and due to the parallelism of the coarse data channels, the processing speed could be radically increased on parallel architectures.

Both statistical and artificial neural network–based categorisers achieved a mean test set recognition accuracy well above 60% when presented with individual channels' data, and above 80% when tested with grouped channels and on synthetic data. The obtained kappa scores and their corresponding statistical significance levels showed a very good agreement beyond chance between the system's output and the known categories. The obtained results have shown that the system is able to distinguish between dissimilar objects with a mean accuracy above 90% (as is the case of the dissimilarities between object No.5 and the rest in the 5–object data set or detritus and larvae images), while objects that presented pronounced geometrical similarity were classified with an accuracy that stayed well above chance – the best test performance in the case of objects 1 and 2 has been 84% and 77% respectively, when using all three channels in the trials conducted on synthetic data. In the case of the Aberdeen data set and a chorus of three data channels, the best performance on herring larvae was 40%, while sandeel and sprat larvae were classified with best accuracies of 90% and 80%, respectively.

The coarse data channels implemented so far have in common the fact that none of them encodes information that is related to the geometry of the objects only. By avoiding focus on geometric axes, symmetries or other similar characteristics of the objects, the automatic categoriser may be trained on images of natural objects. Even non–rigid objects have been amenable to categorisation using these methods.

6. Acknowledgements

The authors are grateful to Paul Rankine from the Marine Laboratory, Agriculture and Fisheries Depart-

ment, The Scottish Office, Aberdeen for providing the set of photomicrographs of fish larvae.

7. References

[1] Ballard D H, Brown C M, Computer vision (Prentice–Hall, Inc, NJ, 1982), p. 300.

[2] Biederman I, Recognition by components: a theory of human image understanding, Psychol. Review 94 (1987) 115–147.

[3] Biederman I, Higher–level vision, in: Visual cognition and action – An invitation to cognitive science, Vol. 2 (The MIT Press, Cambridge, 1990) 41–72 .

[4] Bijaoui A, Slezak E, Rue F, Lega E, Wavelets and the study of the distant Universe, Proceedings of the IEEE 84:4 (1996) 670–679.

[5] Bijaoui A, Starck J–L, Murtagh F, Restauration des images multi–echelles par l'algorithme a trous, Traitement du Signal 11 (1994) 229–243.

[6] Brady M, Representing shape, in: Parallel architectures and computer vision workshop (Somerville college, Oxford, 1987) 256–265.

[7] Bradski G, Grossberg S, Fast–learning VIEWNET architectures for recognizing three–dimensional objects from multiple two–dimensional views, Neural Networks 8:7/8 (1995) 1053–1080.

[8] Canny J, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 679–698.

[9] Cohen J, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1960) 37–46.

[10] Cohen I, Raz S, Malah D, Orthonormal shift–invariant wavelet packet decomposition and representation, Signal Processing 57:3 (1997) 251–270.

[11] Culverhouse P F, Williams R, Reguera B, Ellis R, Parisini T, Automatic categorisation of 23 species of Dinoflagellate by artificial neural network, Mar. Ecol. Prog. Ser. 139 (1996) 281–287.

[12] Dutilleux P, An implementation of the algorithme a trous to compute the wavelet transform, in: Wavelets: Time–Frequency Methods and Phase–Space (Springer, Berlin, 1989) 298–304.

[13] Edelman S, Representation without reconstruction, CVGIP–Image Understanding 60:1 (1994) 92–94.

[14] Edelman S, Representation, similarity and the Chorus of prototypes, Minds and Machines 5 (1995) 45–68.

[15] Edelman S, Bulthoff H H, Orientation dependence in the recognition of familiar and novel views of 3D objects, Vision Research 32 (1992) 2385–2400.

[16] Edelman S, Weinshall D, A self–organising multiple–view representation of 3D objects, Biological Cybernetics 64 (1991) 209–219.

[17] Ellis R, Simpson R, Culverhouse P F, Parisini T, Williams R, Reguera B, Moore B, Lowe D, Expert visual classification and neural networks: can general solutions be found? (IEEE Oceans '94, Brest, 1994) 330–334.

[18] Elmes D G, Kantowitz B H, Roediger H L III, Research methods in psychology (West Publishing Company, St. Paul, 1989).

[19] Fleiss J L, Statistical methods for rates and proportions (John Wiley & Sons, Inc, NY, 1981).

[20] Helterbrand J D, Cressie N, Davidson J L, A statistical approach to identifying closed object boundaries in images, Advances in applied probability 26:4 (1994) 831–854.

[21] Hildreth E C, Ullman S, The computational study of vision, in: Posner M I, ed., Foundations of Cognitive Science (The MIT Press, London, 1989) 581–630.

[22] Hsieh J W, Liao H Y M, Fan K C, Ko M T, Hung Y P, Image registration using a new edge–based approach, Computer Vision and Image Understanding 67:2 (1997) 112–130.

[23] Khotanzad A, Liou J J–H, Recognition and pose estimation of unoccluded three–dimensional objects from a two–dimensional perspective view by banks of neural networks, IEEE Transactions on Neural Networks 7:4 (1996) 897–906.

[24] Landis R, Koch G G, The measurement of observer agreement for categorical data, Biometrics 33 (1977) 159–174.

[25] Liu J, Yang Y–H, Multiresolution color image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 16:7 (1994) 689–700.

[26] Mallat S, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 11:7 (1989) 674–693.

[27] Mallat S, Wavelets for a vision, Proceedings of the IEEE 84:4 (1996) 604–614.

[28] Marr D, Vision – A computational investigation into the human representation and processing of visual information (Freeman, San Francisco, 1982).

[29] Mel B W, Combining color, shape and texture histogramming in a neurally inspired approach to visual object recognition, Neural Computation 9:4 (1997) 777–804.

[30] Niemann T, Lappe M, Hoffmann K–P, Visual inspection of three–dimensional objects by human observers, Perception 2 (1996) 1027–1042.

[31] Ramsay A, Barrett R, AI in practice : Examples in Pop–11 (Ellis Horwood Limited, Chichester, 1987).

[32] Rao R P N, Zelinsky G J, Hayhoe M M, Ballard D H, Modelling saccadic targeting in visual search, Advances in Neural Information Processing Systems 8 (1996) 830–836.

[33] Rattarangsi A, Chin R T, Scale–based detection of corners of planar curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 14:4 (1992) 430–449.

[34] Ren Z, Ameling W, Jensch P, An attributed tree data structure for representing the descriptions of object contours in images, Proceedings of the SPIE – The International Society for Optical Engineering 1360:2 (1990) 956–969.

[35] Rue F, Bijaoui A, A multiscale vision model to analyse field astronomical images, Experimental Astronomy 7:3 (1997) 129–160.

[36] Rumelhart D E, McClelland J L, Parallel distributed processing: Explorations in the microstructure of cognition Vol.1 (The MIT Press, 1986) .

[37] Schiele B, Crowley J L, Object recognition using multidimensional receptive field histograms, in: Buxton B, Cipolla R, eds., Proceedings ECCV'96, 1 (1996) 610–619.

[38] Schiele B, Crowley J L, Transinformation of object recognition and its application to viewpoint planning, Robotics and Autonomous Systems 21:1 (1997) 95–106.

[39] Shensa M J, The Discrete Wavelet Transform: Wedding the A Trous and Mallat algorithms, IEEE Transactions on Signal Processing 40:10 (1992) 2464–2482.

[40] Simpson R, Culverhouse P F, Williams R, Ellis R, Classification of Dinophyceae by artificial neural networks, in: Smayda T J, Shimizu Y, eds., Toxic Phytoplankton Blooms in the Sea (Elsevier Science Publishers B.V, 1993) 183–190.

[41] Sokal R R, Classification: purposes, principles, progress, prospects, Science 4157 (1974) 1115–1123.

[42] Tarr M, Pinker S, Mental rotation and orientation–dependence in shape recognition, Cognitive Psychology 2 (1989) 233–282.

[43] Ullman S, Aligning pictorial descriptions: an approach to object recognition, Cognition 32 (1989) 193–254.

[44] Ullman S, Visual Routines, Cognition 18 (1984) 97–159.

[45] Ullman S, Basri R, Recognition by linear combinations of models, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 992–1005.

[46] Unser M, Aldroubi A, A review of wavelets in biomedical applications, Proceedings of the IEEE 84:4 (1996) 626–638.

[47] Van Hulle M M, Tollenaere T, A modular artificial neural network for texture processing, Neural Networks 6:1 (1993) 7–32.

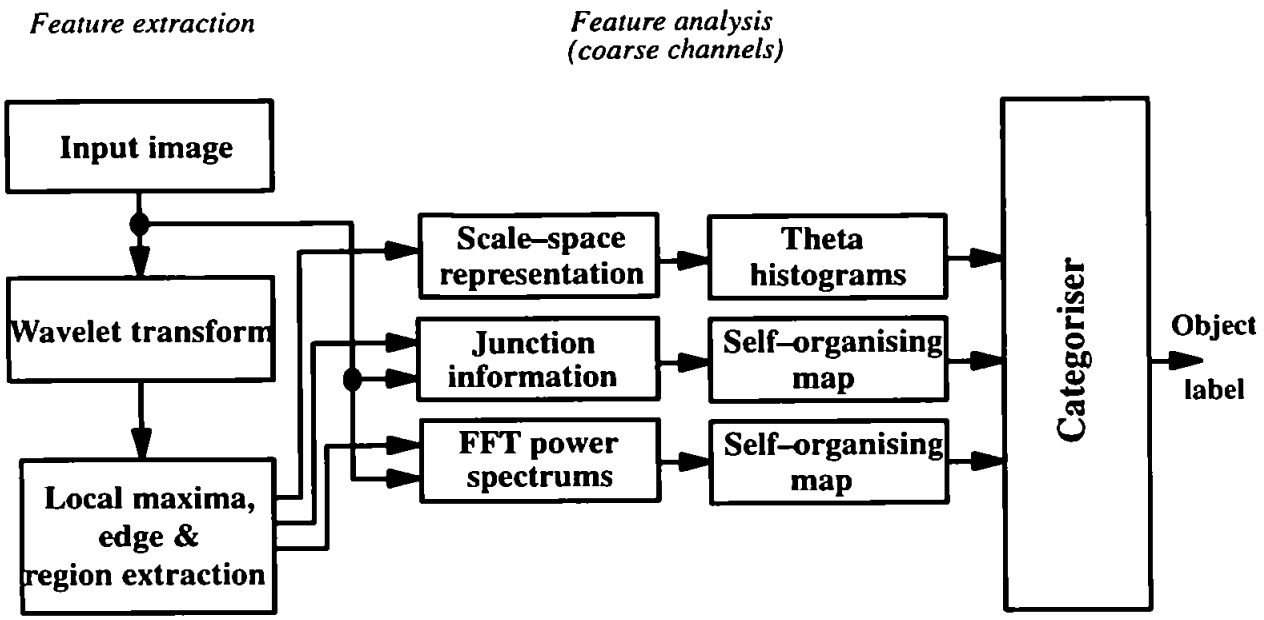


Fig. 1. Structure of the 3D object recognition system

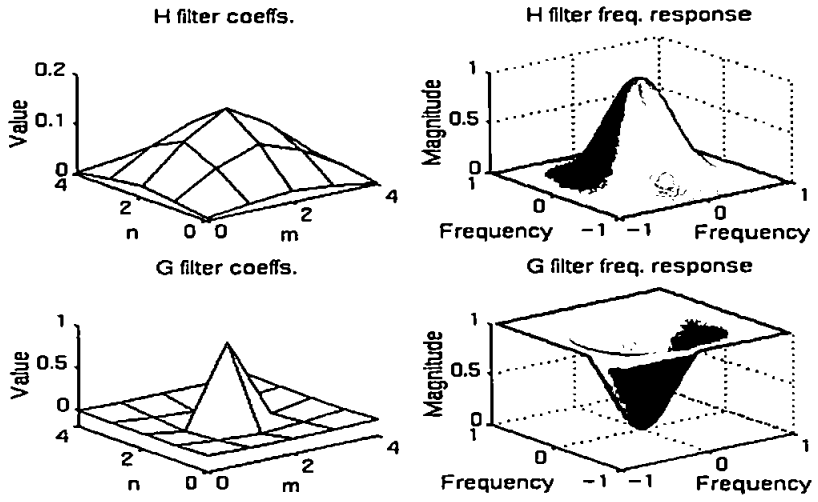


Fig. 2. Filter coefficients and frequency characteristics (normalised frequency, Nyquist = 1)

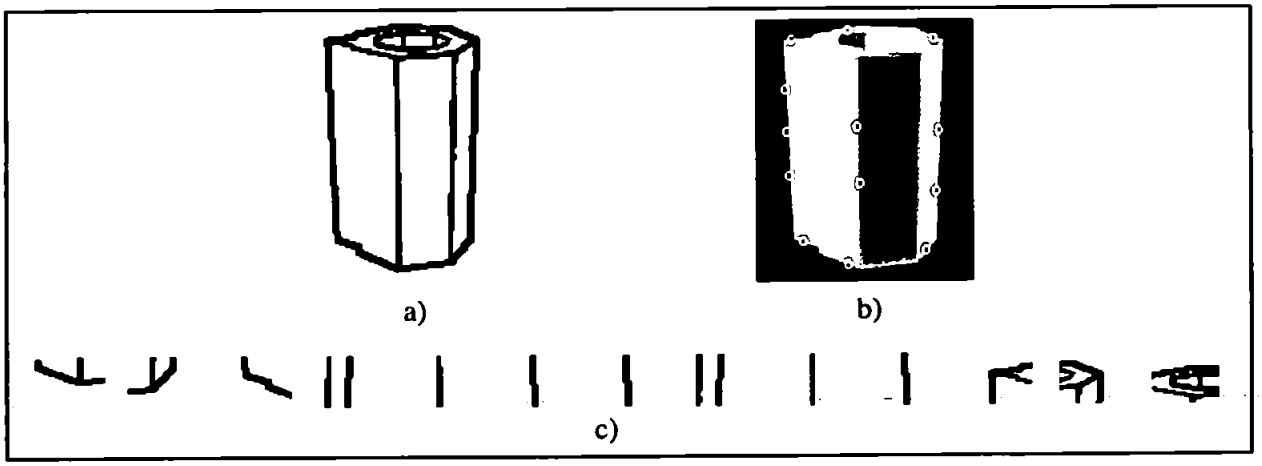


Fig. 3. Extracted junction data: a) Edge data b) Set of points of interest c) Contents of processing windows centred on wavelet maxima

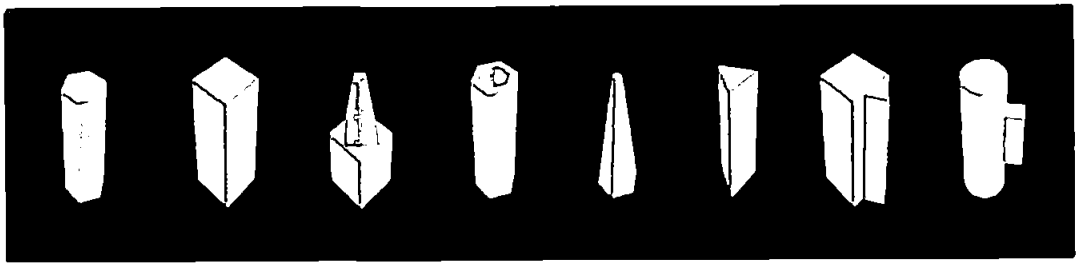


Fig. 4. The set of 8 synthetic objects

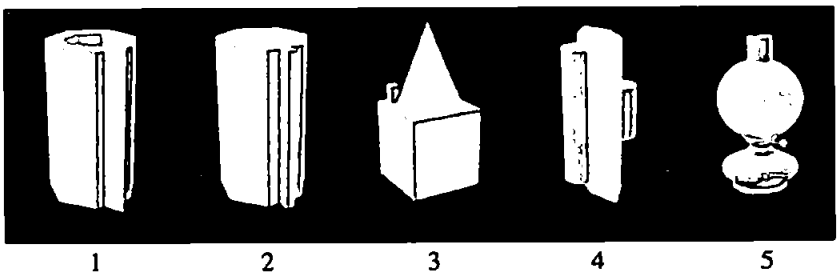


Fig. 5. The set of 5 synthetic objects

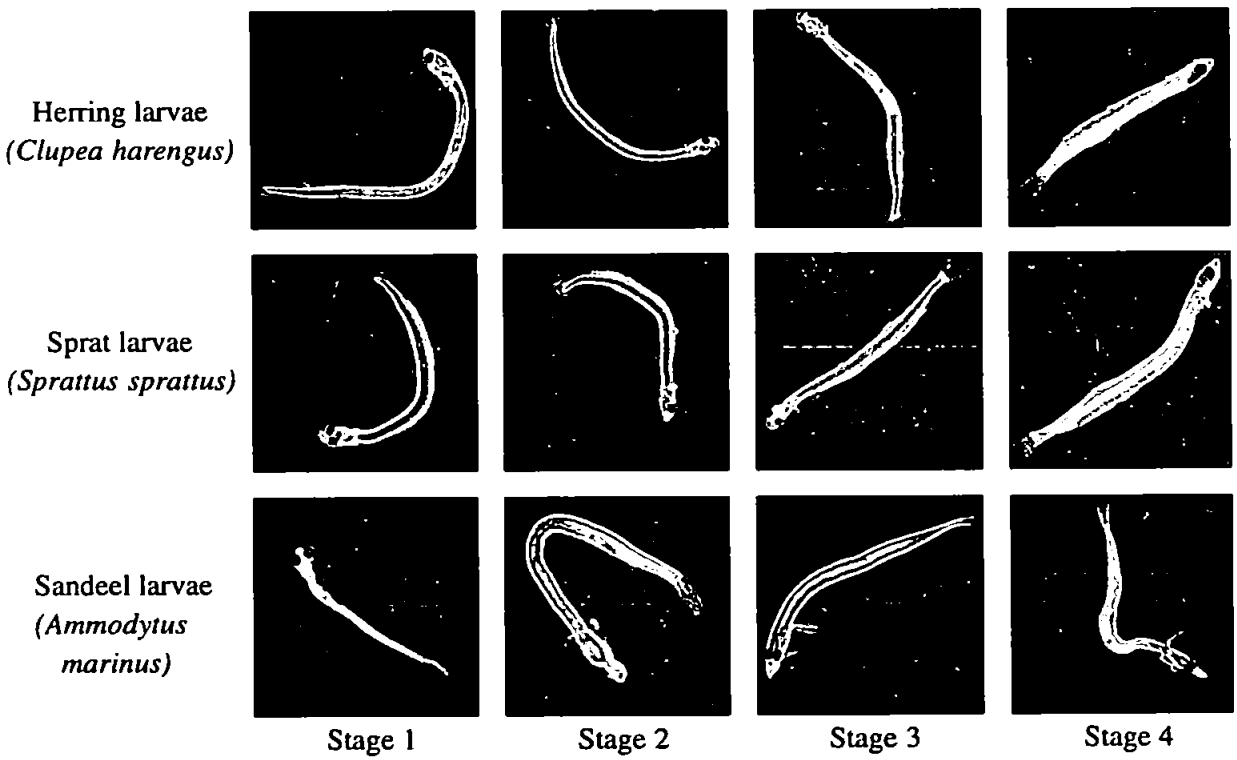


Fig. 6. The images of larvae in four developmental stages (Aberdeen data set).

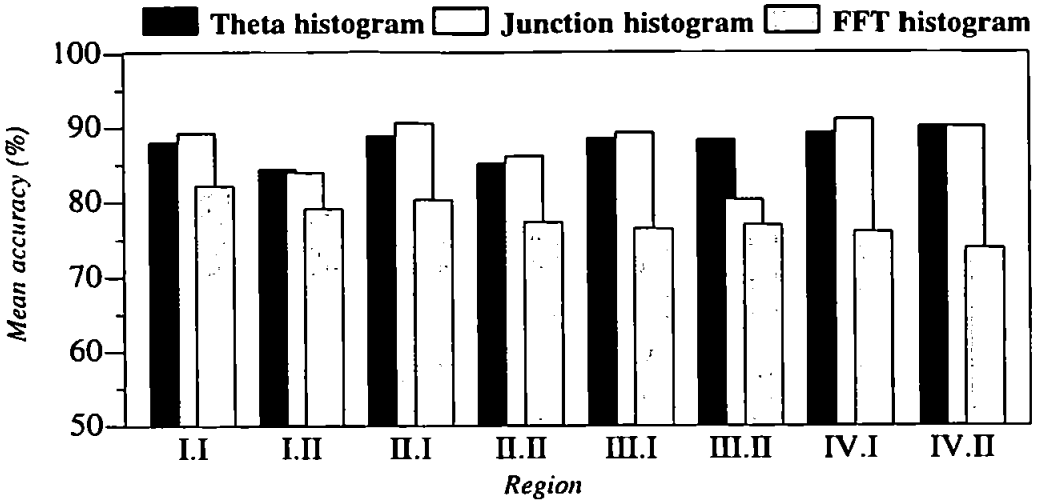


Fig. 7. DA mean classification accuracies for 8 subsets of the 5 object data set. Each viewpoint region has a 90 degrees azimuth and 45 degrees elevation angle span.

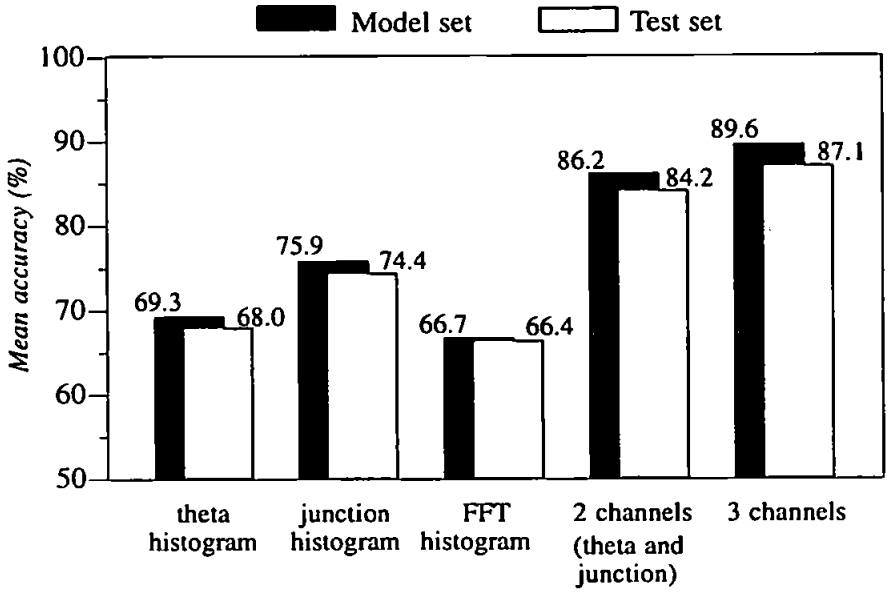


Fig. 8. Mean classification accuracies in DA trials for individual and grouped channels (5 object data set)

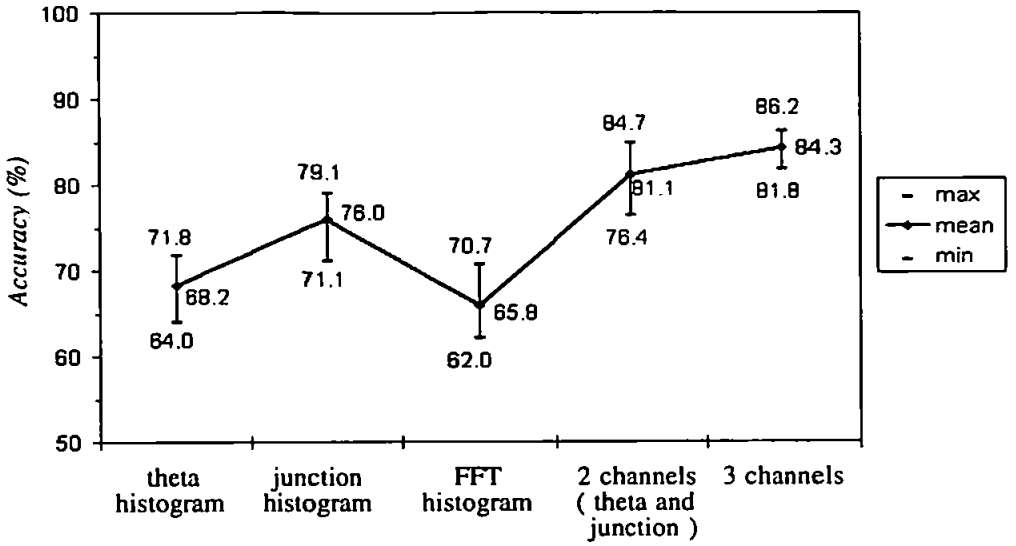


Fig. 9. Classification accuracies achieved on the 5–object test data set by feedforward neural network in 20 runs.

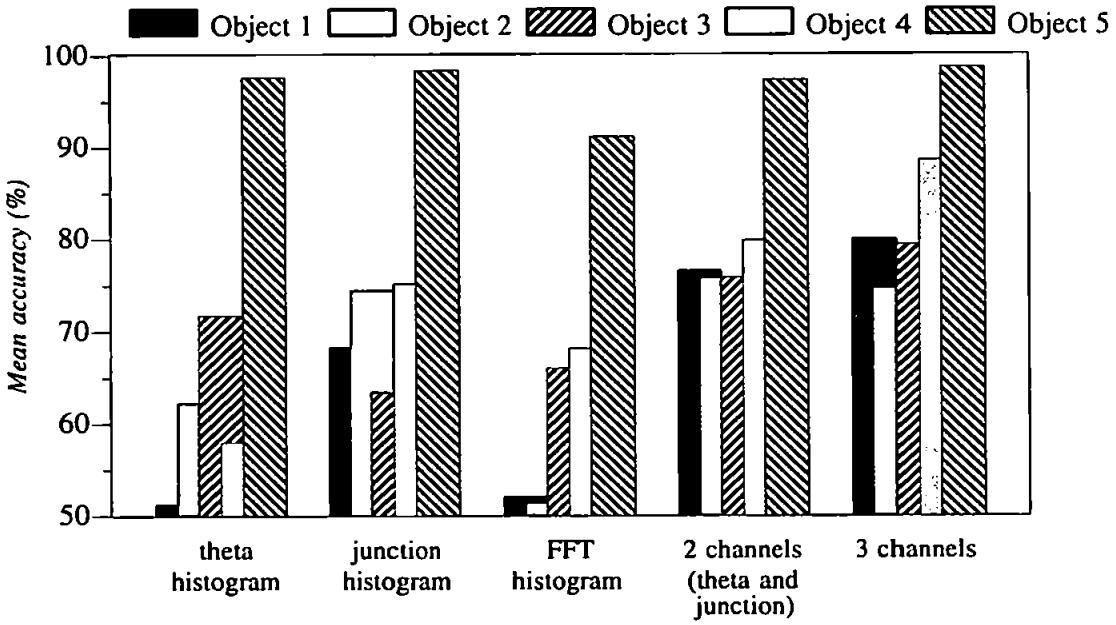


Fig. 10. Mean recognition accuracies obtained on test data by neural network in 20 runs, for each of the 5 synthetic objects

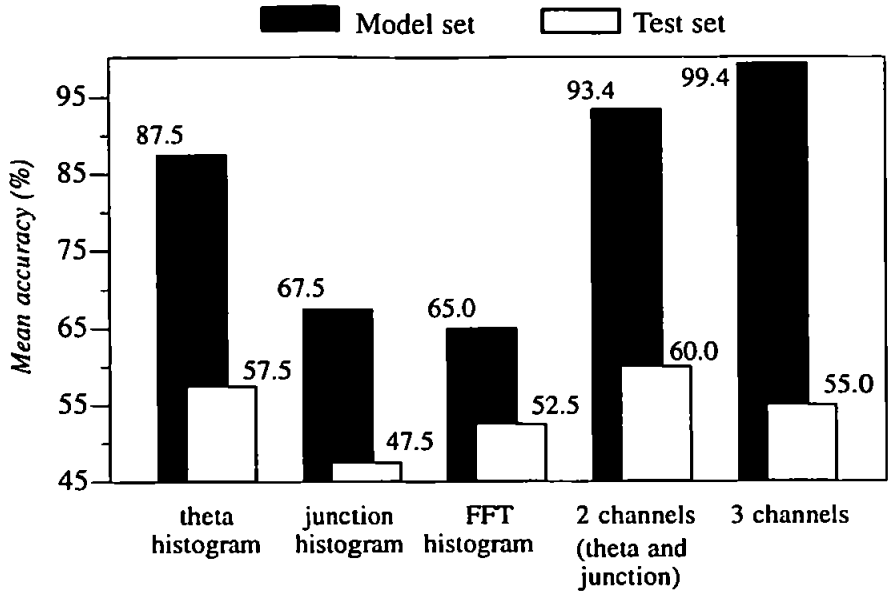


Fig. 11. DA mean classification accuracies for various channel configurations (Aberdeen data set)

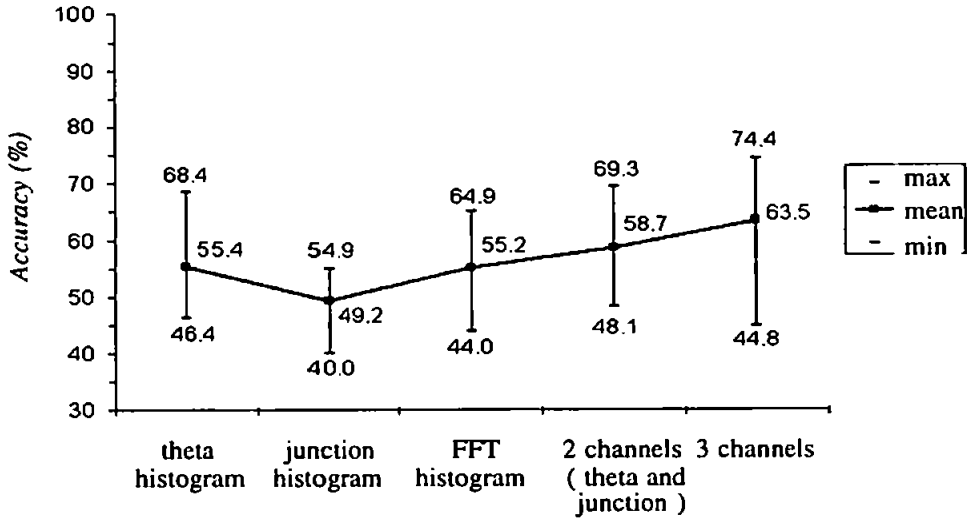


Fig. 12. Classification accuracies obtained from neural network in 20 runs, on larvae test images.

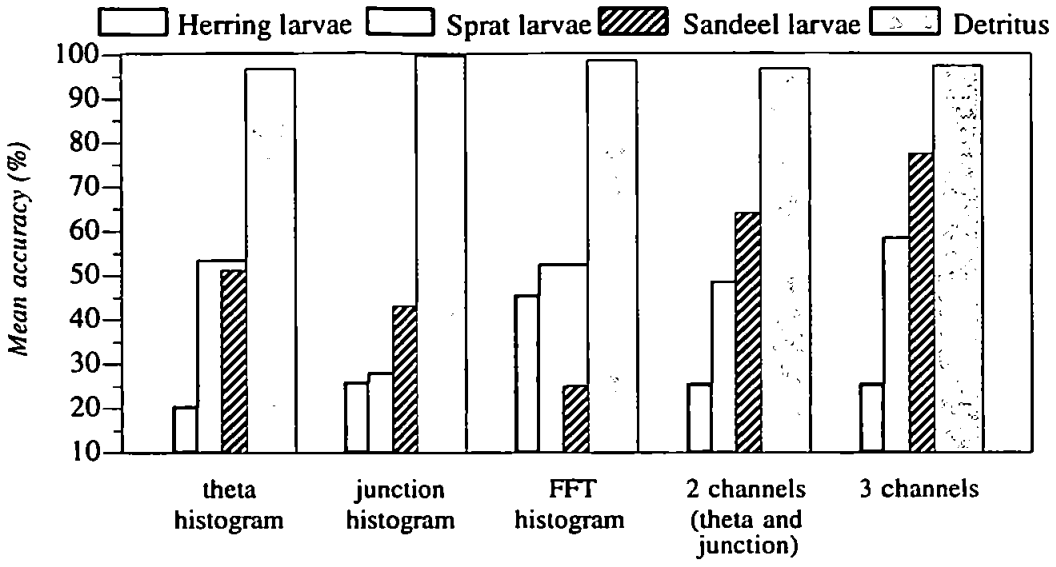


Fig. 13. Mean classification accuracies in ANN trials on Aberdeen test set, for each of the categories.

Table 1. Similarity of objects in the 5–object data set, as assessed by 10 human subjects (1 is very different, 5 is very similar)

Object	1	2	3	4	5
1	.	4.4	1.5	2.7	1.2
2	–	.	1.9	2.7	1.2
3	–	–	.	1.3	1.3
4	–	–	–	.	1.4
5	–	–	–	–	.

Table 2. The mean kappa and corresponding *z* score for each channel configuration, obtained in 20 neural network trials (5 object synthetic data set)

Channel configuration	Mean kappa	*z*
Theta histogram	0.602	25.619
Junction histogram	0.699	29.758
FFT histogram	0.572	24.474
2 channels (theta & junction)	0.764	32.418
3 channels	0.803	34.113

Table 3. Mean percentage of confusion between the 5 synthetic objects in 20 ANN runs					
Object	**1**	**2**	**3**	**4**	**5**
1	.	21.2	17.5	3.7	0.5
2	–	.	15.8	5.3	0.2
3	–	–	.	11.6	1.3
4	–	–	–	.	1.4
5	–	–	–	–	.

Table 4. Categorisation accuracy (%) of Aberdeen test data set, as reported by discriminant analysis run on body size data			
Species	**Herring**	**Sprat**	**Sandeel**
Herring	0.0	70.0	30.0
Sprat	20.0	60.0	20.0
Sandeel	90.0	10.0	0.0

Table 5. The mean kappa and corresponding *z* score for each channel configuration, obtained in 20 neural network trials (Aberdeen data set)		
Channel configuration	**Mean kappa**	*z*
Theta histogram	0.406	28.471
Junction histogram	0.320	23.534
FFT histogram	0.402	28.443
2 channels (theta & junction)	0.449	31.183
3 channels	0.513	35.759