USC-SIPI REPORT #188

Perceptual Grouping and Segmentation using Neural Networks

by

B.S. Manjunath

December 1991

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Electrical Engineering Building
University Park/MC-2564
Los Angeles, CA 90089 U.S.A.

Acknowledgements

I am very grateful to Professor Rama Chellappa, chairman of my thesis committee, for his excellent guidance, encouragement and support during the course of this research. I would like to thank my thesis committee members, Professors Bart Kosko and Gerard Medioni for many valuable suggestions. I thank Professor von der Malsburg, external member of my committee, for making himself available for many fruitful discussions despite his tight schedule. Thanks are also due to Professor Richard Leahy for being on my guidance committee. I would like to thank my former teachers at the Indian Institute of Science, Professors M. A. L. Thathachar who introduced me to the field of learning automata, K. R. Ramakrishnan and P. S. Sastry for their friendship and encouragement.

Dr. S. Chandrashekhar and and Dr. Q. Zheng deserve special credit for putting up with me for over three years. I benefitted a lot from many of those heated debates we had at PHE432. Thanks to Dr. A. Rangarajan and Dr. J. Buhmann for valuable discussions, and to Dr. T. Simchony for convincing me that vision is not just another branch of mathematics!. I would also like to thank Z. Liechtenstein, Dr. V. Venkateshwar, Dr. G. Young, Dr. Y. Zhou, and all fellow SIPI students who made this research a pleasant learning experience.

Special thanks to Dr. Allan Weber for his invaluable computer support, especially during those chaotic days in July 91. My thanks are also due to the SIPI staff, Toy Mayeda, Linda Varilla, Delsa Tan and Gloria Bullock.

This research would not have been possible without the encouragement, understanding, and patience of my parents, B. L. Suryanarayaniah and C. Lalithamba, and my brother B. S. Rajanikanth. I take this opportunity to express my deep gratitude to them.

This research is partially supported by AFOSR under the grants 86-0196, 90-0133.

Contents

List of Figures

List of Tables

Abstract

Grouping and segmentation occur at all levels in the visual information processing hierarchy. We address here their relevance during the early stages in vision, using neural networks as the computing paradigm. Neural networks provide a fresh perspective in solving many of the early vision problems: they provide a parallel and distributed computing environment with the associated fault tolerance; a homogeneous architecture which might help in integrating different visual cues, and in incorporating active interactions between different processing stages; and their ability to learn and self-organize in a continuously changing environment. Besides, they help in bridging the gap between computer vision and human vision research.

We first investigate the use of neural networks from a parallel computing perspective. Most low level vision problems such as model based texture segmentation can be formulated in an optimization framework. In the context of texture segmentation, we show how models based on Markov random fields naturally map onto networks for optimization. We develop a stochastic learning system which combines the speed of deterministic relaxation algorithms with the sustained exploration of search space, a characteristic of stochastic algorithms. We discuss extensions to unsupervised segmentation and the advantages of keeping estimation of parameters and segmentation separate.

While model based approaches to segmentation incorporate knowledge about the textures to obtain fairly accurate results, they do not provide much intuition about human texture perception. The second half of this research is biologically motivated and we try to model some of the early processing stages in vision. We study the problem of pre-attentive texture perception in a more general framework of boundary detection, and develop a unified approach to detecting intensity as well as texture edges, in addition to illusory contours. We clearly

demonstrate the role of end-inhibition, modeled by using local scale interactions, in texture boundary perception and in perceiving subjective contours. Visual illusions are a consequence of wired-in assumptions about the real world, which biological systems make full use of in order to process the enormous amount of visual data in real time. We suggest that the role of end-inhibited cells in the detection of illusory contours is not accidental. End-inhibition provides a robust model for feature detection and representation, and could be used in representing shape information. We demonstrate this in our development of a simple face recognition system, where object recognition is formulated as an inexact graph matching problem. Extensive experimental results are provided to illustrate the performance of the various algorithms.

Chapter 1

Introduction

We are so familiar with seeing, that it takes a leap of imagination to realize that there are problems to be solved

.... Richard Gregory.

Vision is about analyzing and interpreting images, and it continues to be a challenge for researchers in such diverse fields as Electrical Engineering, Computer Science, Mathematics, Physics, Neural Sciences and Psychology. Considerable progress has been made in understanding early human vision, as well as in the development of computer vision algorithms. A recent trend has been the emergence of neural networks as a computing paradigm for many vision applications. Besides the often cited biological motivation (the only existence proof of a complete vision system, the human visual system, is based on neural hardware), neural networks offer many advantages: they provide a parallel and distributed computing environment with the associated fault tolerance; a homogeneous architecture which might help in integrating different visual cues; and most importantly, their ability to learn and adapt to an ever changing visual environment using very simple mechanisms. This thesis is motivated by all the above mentioned aspects of neural computing with respect to vision research. In particular we consider perceptual organization of information during early stages of vision, and explore the use of neural networks in solving some of the associated problems.

During the past ten years a dominant paradigm in computer vision research

has been that of Marr's [73]. According to Marr, vision can be studied, *independently*, at three different levels: computational- where the goal of computation or *what* is to be computed, is defined precisely, algorithmic - explains *how* these computations can be carried out, and hardware- deals with specific implementations. Given the complexity of the visual world, there are very few tasks for which we know precisely what is to be computed. Even for early vision problems such as texture segmentation many different approaches exist, dictated by specific applications and prior assumptions about the data.

A topic of current interest among computer vision researchers is to bring in ideas and knowledge from psychophysical and anatomical studies of human visual system in the development of vision algorithms. However, the study of human vision is complicated because one can no longer identify or separate out the three different computational stages as suggested by Marr in the case of biological systems. Biological mechanisms have evolved over millions of years, and have been inherently opportunistic, and were not designed ! The only performance criteria for these systems is that they must work under the existing constraints (such as the hardware). This observation has been a motivating factor for an emerging new paradigm, the utilitarian theory of perception [88]. In spite of the complex nature of the problems, some progress has been made, particularly in the understanding of some of the pre-processing stages in the visual cortex. The second half of this thesis dealing with boundary detection tries to incorporate some of the available information about processing by cells in the visual cortex into a computational framework.

Before continuing we would like to make clear our use of some of the terminology. The word *neural* is used quite often throughout the thesis, but it is not intended to represent anything biological. The same is true with respect to our references to cells in the visual cortex. It is for convenience that we use the same terminology and one should be careful in making the interpretations. Most of our discussions are at a functional level. Although some of our work is motivated from biology, we are not developing any specific models for the visual cortex.

1.1 Why Neural Networks ?

Neural networks, as the name suggests, are networks of simple processing elements called neurons. A neuron receives signals form other neurons, processes the signal and transmits them. A neuron can influence the output of another neuron in two ways: (i) by providing a positive (or excitatory signal), or (ii) by providing a negative (or inhibitory) signal. The dynamics of the network is determined by these excitatory and inhibitory connections, and by the transfer characteristics of the neurons. Recently neural networks have received much attention in various applications such as associative memory, optimization, and vision. We discuss below some of the main reasons for the use of neural networks in our research:

1. Parallel and Distributed Computing: Neural networks offer a new paradigm for parallel and distributed computing. This aspect of computation has received a lot of attention, both in theory as well as in applications [45, 46, 47], and is particularly significant for many early vision problems. The amount of data that needs to be handled during the early stages in vision is quite enormous, even for modest image sizes. Many of these early vision problems can be formulated in an optimization framework [71, 109], and analog/digital neural networks can be designed to obtain good solutions. This, in our opinion, is among the most important of contributions to research in neural networks to vision to-date.

2. Fault tolerance and graceful degradation: The distributed nature of information processing by neural networks comes with an added advantage, that of fault tolerance and graceful degradation. This, in contrast to traditional computing where even the failure of a single component might affect the whole system, has been of considerable use in applications like associative memory [15, 46, 61, 62].

3. Learning and Self-Organization: Neural networks can learn by modifying their connection strengths (often referred to as synaptic strength in neural network literature), which represent in some sense information about the

environment. This ability to learn using such simple mechanisms as changing the synaptic strengths is an important factor which distinguishes neural networks from other traditional computational models. One of the early successful adaptive pattern recognition systems built, the Neocognitron [34] is based on neural architecture.

While the above attributes of neural architecture are of considerable importance, neural networks play a useful role from another perspective as well. As mentioned earlier, the only proof that the problem of vision is in fact open to solution is the existence of human visual system. Much can be gained from a proper understanding of the structure of visual cortex, as is evidenced by our work on boundary detection and shape recognition (see Chapters 4 and 5). The problems that we consider in this thesis are about detecting and grouping features during the early stages of visual processing in order to carry out scene segmentation, boundary perception and recognition. We explore the use of neural networks both from the parallel and distributed computing perspective as well as in developing models based on biological processing of these visual signals.

1.2 Perceptual Grouping and Segmentation

Perceptual grouping refers to combining regions in an image on the basis of simple cues, and in the process segmenting the scene into distinct objects [82]. This is one of the fundamental problems in computer vision, and plays a very important role in high level tasks such as object recognition.

Grouping and segmentation occur at all levels in the visual information processing hierarchy. Among the attributes that influence grouping during the early stages are color, size, orientation, brightness, and density. At a slightly higher level, symmetry, closure and continuity also play a role. It has been demonstrated that pre-attentive grouping (referring to grouping of features without attention) can occur only if the differences are in single features, and not in their conjunctions [94]. For example, one can segment a scene based on shape alone (such as a scene consisting of circles and squares), or brightness alone, but not a combination of shape and brightness. Such studies provide valuable information

about the underlying structure of human visual system.

In computer vision research also, the process of grouping, and in particular its role in perceptual organization, is receiving considerable interest. Lowe [66] discusses the development of a recognition system based on perceptual organization. Edges and lines form the basic primitives in his system. He demonstrates that it is not really necessary to have a 3-D description available for recognition, and that perceptual grouping of features in 2-D images are sufficient for developing a robust recognition system.

Perceptual grouping addresses the question of what are the characteristics of the features that help in the grouping process. In many cases, particularly in most natural images, the basic problem is what type of features are to be detected or computed from the raw image data. The choice of features undoubtedly will have an effect on the development of the algorithm itself. There is no reason why one should limit to features detected in early processing by biological systems. For example, if the visual environment is constrained and is known to have certain structure, random or otherwise, it would be wise to make full use of this knowledge. Our discussion on texture segmentation using Markov random field image model follows this line of thought. Statistical models, and in particular Markov random fields, have been effective in capturing information about many natural textures such as wood, wool, sand, water, grass etc., and have found many useful applications [90]. As they make use of some prior information about the textures, they are invariably more accurate than any model-free approach, such as algorithms for pre-attentive texture discrimination [67, 70]. Simple perceptual organization of features such as local curvature can provide a powerful mechanism for shape representation, as we illustrate in Chapter 5 in our development of a face recognition system.

1.3 Summary of Contributions

1. In a model based approach to texture segmentation, we formulate the problem in an optimization framework, and suggest a neural network implementation of a deterministic relaxation algorithm.

2. In order to overcome the dependency of the network on initial state, we combine techniques from stochastic learning automata with that of deterministic relaxation. This algorithm combines the speed of deterministic relaxation with sustained exploration of search space, a characteristic of stochastic algorithms, and its performance is comparable to computationally demanding techniques such as simulated annealing.

3. A general framework for boundary detection is suggested, based on a model for human perception. We include in this framework intensity edges as well as texture boundaries, and provide an analysis of the performance of the system in edge detection.

4. A model for end-inhibition (which refers to the property of hypercomplex cells in the visual cortex of mammals) is developed, and its role in texture boundary detection and in detecting illusory contours is demonstrated.

5. The role of end-inhibited cells in curvature detection and shape representation is further explored, and we develop an efficient system for human face recognition. Connections between shape perception and visual illusions are also discussed.

1.4 Thesis Outline

This thesis is divided into six chapters. The next chapter briefly summarizes background information on Markov Random Fields (MRF), neural networks and vision. Chapters 3, 4 and 5 form the core of the thesis. Boundary detection forms the basis of Chapters 3 and 4, although the approach they follow are very different from each other. In Chapter 3, a model based approach to texture classification and segmentation is taken. An MRF model is used in modeling the texture process, and the problem of classifying and segmenting a texture scene is formulated using an optimization frame work. We discuss how this problem can be mapped on to a neural network for obtaining fast solutions. A new algorithm is proposed which combines techniques from stochastic learning with deterministic relaxation in order to improve the performance. The results compare favorably with those

obtained using simulated annealing, a stochastic relaxation algorithm. We also discuss extensions to unsupervised segmentation where the model parameters are unknown. Unsupervised segmentation involves estimating model parameters and then performing the segmentation. We suggest that separating estimation from segmentation might be better in terms of developing fast algorithms. We also discuss some recent trends regarding unsupervised segmentation and put our work in perspective.

Chapter 4, while discussing the same problem, i.e., texture segmentation, takes a different approach. No prior models are assumed, and the development of the whole system is motivated from previous work on pre-attentive segmentation. However, unlike most other previously suggested models, we tackle a much more general problem, that of detecting perceptual boundaries in the images irrespective of whether they are intensity edges or texture discontinuities. We show how very simple grouping mechanisms can help in finding illusory contours as well. The discussion on illusory contours is continued in Chapter 5 as well as in Chapter 6.

While Chapter 4 looks at grouping from boundary detection point of view, Chapter 5 considers shape representation and the problem of recognition. We suggest that from the very early stages in information processing robust features are detected and grouped to form meaningful representations of shape. Perceptual organization of the features helps in developing efficient solutions to the recognition problem. We show that even such primitive features as local curvature can be used very effectively in many applications of practical interest. The proposed mechanism for identifying these features has been successfully applied in motion tracking and in image registration. We also develop a matching algorithm, motivated mainly from the work on dynamic link architecture [97, 99] to illustrate how organizations at even the lowest levels can help in solving problems such as face recognition.

Chapter 6 looks back at our work and summarizes the contributions. It also discusses some interesting aspects of visual illusions and how their study might provide useful information regarding image understanding. We discuss the emerging paradigm of utilitarian theory of perception, and its possible role

on computer vision research. We address issues related to dynamic binding of features, and its implications on research on associative memory and vision.

Chapter 2

Preliminaries

This chapter discusses some of the relevant work on Neural Networks, Markov Random Fields (MRF), and their applications to vision. Research on the applications of neural networks to vision has a long history. As mentioned in the previous chapter, neural networks offer a new paradigm for computation. Many of the low level vision algorithms benefit from the parallel computation aspect of these networks, and it continues to be the most active research area in vision and neural networks. The renewed interest in this field can be attributed to an interesting set of papers by Hopfield [45, 46, 47] exploring the role of neural networks in applications such as associative memory and optimization. A brief review of the Hopfield network follows:

2.1 Hopfield Networks and Optimization

Consider a network consisting of N *neurons*, with the ith neuron connected to the jth neuron with a synaptic strength T_{ij} (for simplicity assume that $T_{ii} = 0$). The input to a neuron i consists of contributions $T_{ij}v_j$ from other neurons and I_i from an external source, v_j representing the output of neuron j. Total input to i is then

$$u_i = \sum_j T_{ij}v_j + I_i \tag{2.1}$$

Assuming that the neurons are bistable devices taking on values 0/1, the input/output relation is given by

$$V_{ijl} = \begin{cases} 1 & \text{if } u_i \geq 0 \\ 0 & \text{if } u_i < 0 \end{cases} \tag{2.2}$$

The neurons change states at random time instants. It can be shown that the network has stable equilibrium points by considering the following Lyapunov or energy function:

$$E = -\frac{1}{2}\sum_{i \neq j} T_{ij} v_i v_j - \sum_i I_i v_i \tag{2.3}$$

The updating rule in (2.2) ensures that the above energy decreases during each iteration. Since the energy is bounded, the system eventually converges. Analog implementations of this network have also been suggested (see for example [46]). In general, discrete and analog versions of the same network might have different stable states corresponding to the same initial configuration.

Hopfield networks have been extensively used in vision applications. Zhou [109] discusses applications to image restoration, optical flow and stereo. It has been used also in surface interpolation [51] and motion computation [59, 43]. The underlying principle is to formulate these problems as optimization tasks. The key is to express the cost function to be optimized in terms of the parameters of the Hopfield network discussed above. Low level vision problems are computationally very intensive, and such network implementations, possibly on analog VLSI circuits, will be of significance for real time applications. MRF models for some of these problems are of particular interest as the cost function can be derived directly from these model parameters.

2.2 Neural Networks, Markov Random Fields and Vision

The renewed interest in neural networks coincided with another important development in vision, that of MRF image models. MRFs are an important class of stochastic models which have been applied to many image processing and low

level vision problems [19, 24, 40]. Geman and Geman [39] are mainly responsible for bringing to focus the relationship between MRFs and Gibbs distributions, and for providing an analytical treatment of the simulated annealing algorithm.

2.2.1 Markov Random Field

It is convenient to define a MRF on a graph, and the discussion here follows closely the one presented in [57]. Consider a graph $G = (S, E)$, where S denotes the set of nodes (also referred to as sites or vertices), and E the set of edges. An edge connects two *neighbors* in the graph. The set of neighbors for a site s is denoted by N_s. A value $Y_s \in \mathcal{I}$ is assigned to each site s in the graph. For example, in case of images, $\mathcal{I} = \{0, 1, \ldots, 255\}$ represents the set of possible intensity values. Let $\mathbf{y}$ represent one such assignment, and is called a *configuration.* and let $\mathcal{Y}$ represent the space of all possible configurations. We use Y_s to represent the random variable, and y_s for a specific value that this variable takes. A set of nodes $C \in S$ is called a *clique* if every pair of nodes in C are neighbors. A main attribute of a MRF is that the conditional distributions are local, and depend only on the neighborhood set.

$$P(y_s|y_r, r \neq s) = P(y_s|y_r, r \in N_s) \tag{2.4}$$

2.2.2 Gibbs Distribution

A probability density function of the form

$$P(y) = \frac{e^{U(y)}}{Z} \tag{2.5}$$

is called a Gibbs distribution. The function U is often termed as an energy function or a Gibbs measure, and Z is called the partition function. An important relationship between MRF and Gibbs distribution is the Gibbs-Markov equivalence (also referred to as the *Hammersley-Clifford* theorem), which states:

$\mathbf{Y}$ *is a MRF with respect to* G *if and only if it is a Gibbs distribution of the form*

$$P(\mathbf{y}) = \frac{e^{\sum_{c \in C} V_c(\mathbf{y})}}{Z} \tag{2.6}$$

where V_c is called the *potential* of clique c. The local conditional probabilities are given by

$$P(y_s | y_r, r \neq s) = \frac{e^{\sum_{c \in C} V_c(\mathbf{y})}}{\sum_{\mathbf{y}'} e^{\sum_{c \in C} V_c(\mathbf{y}')}} \tag{2.7}$$

Here $\mathbf{y}'$ includes all configurations which are identical to $\mathbf{y}$ except possibly at site s.

2.2.3 Early Vision and Markov Random Fields

Early vision problems deal with extracting higher order features from the raw intensity data. Intensity variations in an image are usually local, and reflect the physical constraints on the imaging process. An intuitive choice of variables can lead to powerful Markovian models, which in turn would help in extracting useful information from the image. An obvious choice for the variables is the intensity process itself, which is directly related to the observed data such as a 3-D object or a textured surface. The conditional distribution of these intensity variables can be expressed as a MRF, with the spatial neighbors of a pixel forming the neighborhood set. Very often, there is an unobserved set of variables, which we term as attribute variables, associated with the observed data. They are unobservable in the sense that they are not directly related to the intensity itself, but rather control the intensity process. For example, texture labels are the attribute variables for a texture intensity process. The distribution of the intensity directly depends on the particular texture, such as water or grass. In many low level vision problems such as texture segmentation, it is of considerable interest to extract these attribute variables. This is achieved by minimizing a suitable cost function, such as minimum mean square error or maximum a posteriori estimate.

This minimization problem, for most vision applications, is intractable, due to the non-convexity of the associated cost functions and the large dimensionality

of the data. Much research has gone into developing suitable algorithms. Among the stochastic algorithms, simulated annealing [39] offers a powerful relaxation technique which exploits the Markovian structure of the problem, and in theory at least, asymptotically converges to the globally optimal solution. However, this technique does not appear to be of much practical value for many, if not all, image processing and vision applications. In recent years several deterministic algorithms have been investigated. Besag [5] suggested the Iterated Conditional Modes (ICM) algorithm, initially as a sub-optimal solution to the MAP estimate, but later as a solution in its own right. ICM maximizes the local conditional density of each variable during each iteration, while keeping the others fixed. There is an immediate connection between this technique and the Hopfield network for optimization (discussed in section 2.1) and is further discussed in chapter 3. The Iterated Conditional Expectation (ICE) [37] on the other hand obtains a minimum variance estimate. It is in the development of such algorithms that neural networks have been of considerable interest. Besides providing a natural parallel architecture for distributed computing, essential for any real time solution, they have opened a whole range of techniques from statistical mechanics for pattern classification and clustering applications. For further discussion on neural networks as applied to Markovian models for low level vision applications we refer to [89].

2.3 Pattern Recognition and Neural Networks

2.3.1 Associative Memory

In traditional computers, the data is stored at specific address locations. To read this data, one has to address the correct location. In contrast, in associative memory, the data is retrieved by giving as an input an association. For example, the input can be the image of a human face and the retrieved information is the name of that person. Associative recall is the key to human perception as well. The goal of associative memory models is to capture this processing. There have been many associative memory models discussed in the literature. Among the prominent of them are Grossberg's ART [15], Kosko's BAM [62], Hopfield's

model [45] and Kohonen's CAM [60]. It is not our intention to review these models here, but instead consider some of the applications of these associative memory models in vision research.

An associative memory is a network of neurons, and typically stores information in the interconnections. Thus the connection matrix constitutes the Long Term Memory (LTM) of the system. Due to its distributed nature of information storage, failure of few elements or connections will not be catastrophic. The input defines the initial state of the system, and the corresponding association the final state. A typical application is in image restoration, where given a noisy or incomplete information, such as a partially occluded face, the system will retrieve the complete face [61]. Although many attempts have been made to apply these models to visual recognition, it is becoming increasingly clear that there are several serious shortcomings of these models, and they require some fundamental changes before even attempting to apply them to any realistic problems.

2.3.2 Structured Networks

An aspect which is usually ignored in applying associative memory to visual recognition is that of feature selection. Although feature extraction is part of the recognition problem, it has been generally ignored in the development of associative memory models. Some of the earlier applications of these models claim invariant object recognition (invariant to translation, rotation and scale), by simply applying global transforms such as Fourier or Mellin, to the intensity data, and then using the associative memory for recall [16, 103]. It is naive to assume that such global transformation techniques could lead to powerful models, or that they could provide an understanding of how the brain deals with invariances. This leads us to the issue of dynamic feature bindings, and one of the serious drawbacks of most current associative memory models – lack of flexible structure to represent syntactic information.

The main motivation for Malsburg's work on temporal correlations [97] as a means for feature binding is to overcome this deficiency of regular associative memory models. The novel idea is to use dynamic connections, varying at the same time scale as the neuronal activities, to bind features. Temporal

correlations play a very important role in establishing these connections. Although conceptually easy to visualize, computer implementations have turned out to be surprisingly complex. Many variations have been tried and we refer to [98, 99, 100, 101] for further details. In many of these, the recognition problem is reduced to finding suitable sub-graph isomorphism between the input and a stored model, both being represented in terms of topological graphs. Our development of human face recognition system is partly motivated by this approach to recognition.

A good example of a hierarchical structured network which systematically detects and groups features is Fukushima's Neocognitron [34, 35]. This feed-forward network was developed to recognize hand written characters. The connections between successive layers in the network are modified during the learning phase. Local features are extracted and grouped by the various layers in the network, with the cells in the final layer representing distinct patterns. The system performs well on hand written numerals and characters and tolerates a fair amount of distortion, scaling and translation. Neocognitron represents one of the few examples of neural models which, unlike the associative memory models, incorporate feature extraction into the network itself, and makes use of local spatial characteristics of the visual data.

2.4 Early Visual Processing in Biological Systems

The discussion so far has been on the applications of neural networks to vision from a purely computational point of view. In this context, it is probably not inappropriate to say that neural networks blur the distinction between the three computational levels proposed by Marr [74]. This is even more so when we try to understand the underlying structure of human visual system. Development of *neural algorithms* might further aid in bridging the gap between computer vision and research in biological vision systems. Considerable progress has been made in the understanding of visual cortex of mammals. Much of the early research in this field is due to the pioneering work of Hubel and Wiesel [48, 49, 50], who

identified and classified cells in the cortex responsible for early visual processing. These cells can be broadly grouped into three functional classes: simple, complex and hypercomplex. Simple cell receptive fields are sensitive to bars (lines) and step edges and their orientations, and can be modeled by even-symmetric (line detectors) and odd-symmetric (step edge detectors) filters. In addition the cells are also sensitive to direction of contrast. Complex cells respond to more complex patterns such as textures, and unlike simple cells, do not contain any phase information, are less sensitive to precise location but are tuned to respond to different specific orientations and direction of movement. These cells are usually modeled by summing the outputs of a group of simple cells of similar orientations. Hypercomplex cells in the cortex exhibit end-inhibition, in that they respond to small lines and edges, and their response decreases as the length increases [50]. These cells appear to play an important role in localizing line-ends and texture boundaries, and both simple and complex cells with this end-stopping behavior are known to exist.

Many attempts have been made at modeling the receptive fields of simple cells in the visual cortex. The response of these cells in the cortex can be characterized by convolution with an even or odd-symmetric filters and a non-linear transformation (such as a sigmoid non-linearity). These cells are tuned to specific spatial frequencies, and their bandwidth is about one octave [72]. Among the popular mathematical models used in capturing the receptive fields are the difference of Gaussians and the Gabor functions [25, 72]. Gabor functions are particularly attractive because of their theoretical information processing properties such as minimizing the joint two dimensional entropy [25]. This has been one of the motivating factors for our choice of Gabor wavelets in Chapter 4, in detecting features during the early stages of boundary perception.

2.4.1 Boundary Contour System

Grossberg and Mingolla's Boundary Contour System (BCS) [42] is one of the first attempts to model the early processing stages in the visual cortex. The BCS processes the intensity data and performs pre-attentive segmentation of the scene. The first stage of BCS consists of oriented contrast filters at various

scales and orientations and extracts the contrast information from the scene. The outputs of the filters are then fed to a two stage competitive network whose main goal is to generate end-cuts. Subsequently long range cooperative interactions and a positive feedback to the competitive stage help in boundary completion. The boundary detection takes place independently in different spatial channels. A detailed description of this model and its performance in texture grouping and in detecting illusory contours such as the Kanisza's square can be found in [42].

The BCS model provides a very general framework in which many of the current models for detecting boundaries can be included, though the details might differ. The model itself as detailed in [42] is quite complicated and computationally expensive to simulate on any real image examples. The BCS model does not account for even symmetric mechanisms which are useful in detecting lines and in texture discrimination [67]. The orientation contrast filters proposed as the first stage in the model consist of only odd-symmetric filters for detecting contrast information. A more serious problem with the BCS model, however, is the proposed end-cut mechanism, described below.

In the BCS model it is hypothesized that all line ends are illusory. In order to detect line ends, a two stage competitive network is proposed. Lateral inhibition between similar orientations in neighboring positions and between different orientations at the same spatial location result in the generation of end-cuts. As the authors point out, at the line ends, the simple and complex cell outputs are not strong and compensatory mechanisms are required for detecting these line ends. However, it is not clear if these competitive interactions by themselves are sufficient for end-cut generation, and if not, what additional mechanisms are needed. Further, as noted in [85], the two stage model hypothesizes that all cells involved in boundary completion are of hypercomplex type, which is clearly not true. We propose an alternate model for this using scale interactions in Chapter 4, and demonstrate the usefulness of these features in various applications including face recognition.

2.4.2 Hypercomplex Cells and Subjective Contours

It is only recently that researchers in computer vision are getting interested

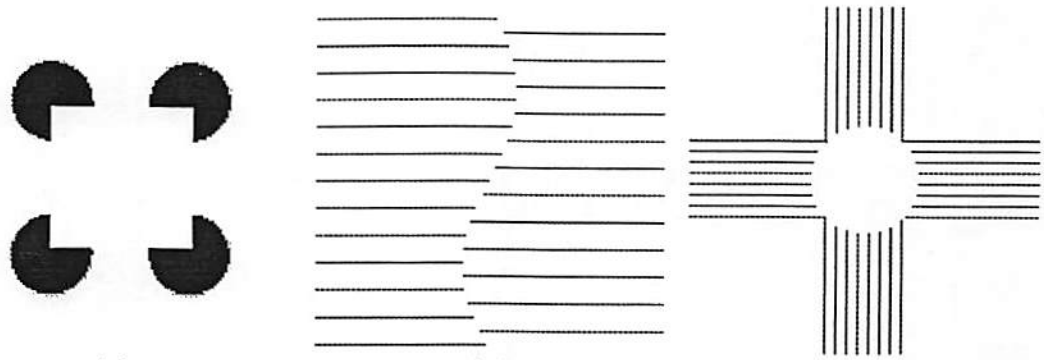


Figure 2.1: Some examples of illusory contours (a) Kanisza's square (b)-(c) subjective contours induced by line terminations.

in visual illusions, although this topic has been studied by psychologists for over fifty years. As Ramachandran [87] points out, visual illusions offer evidence regarding the *built-in* assumptions about the real world from the very early stages of visual processing, and provide valuable cues regarding the underlying structure of biological systems. Subjective contours are a class of visual illusions where boundaries are perceived in the absence of any physical causes. Figure 2.1 show some common examples of such illusory contours. A main motivation for our work on subjective contours is to gain a further understanding of the mechanisms involved, and to establish a relationship between their perception and other *normal* visual processing. Pioneering experiments by von der Heydt and Peterhans [85, 96] demonstrate that hypercomplex cells play a very active role in the perception of subjective contours. Our model for boundary perception (4) incorporates the key ideas from their study and show how simple bottom-up techniques could lead to the detection of contours such as those in Figure 2.1 (b)-(c). However, the main contribution is not in the detection of these contours as such, but in establishing the role of these hypercomplex cells in shape perception as well. We conclude that perception of many such subjective contours are a direct consequence of grouping mechanisms involved in the recognition process, and as mentioned above, reflect wired-in assumptions about the real world.

Chapter 3

Texture Classification and Segmentation

Textures form one of the most common attribute of natural scenes, and there is no precise definition yet of what a texture is. Nonetheless, for many applications segmentation of these textured scenes is a necessity, and during the past 10-15 years considerable work has been done on textured image segmentation. In this chapter we are primarily interested in a model based approach to this problem. The problem of pre-attentive segmentation, which refers to the ability of humans to segment textures without any sustained attention, will be dealt with in the next chapter.

In a model based approach, the texture process is characterized by a mathematical model. We assume that the underlying intensity process follows or is derived from a given model. Usually this is an approximation, as no real texture obeys any given model. In spite of this, good results have often been obtained in following this approach. In the literature there are hundreds of different models which have been proposed and shown to work on an equal number of textured images. Broadly, they can be identified as either statistical models or structural models. One should be careful to choose an appropriate model for the particular application. Structural models are useful when the patterns are highly repetitive, such as a brick wall. A statistical model is more appropriate when one is trying to capture information about random textures such as sand, wool or grass. By

choosing a model, the problem is reduced to estimating the model parameters, which are far fewer than the variations in the intensity that they represent. The model that we are using is statistical and is based on modeling the textured image as a Markov Random Field.

Choosing the model is only the first step in segmentation. The next step involves estimating the model parameters. In supervised segmentation, one assumes knowledge of different textures present in the image. The parameters of these textures are pre-computed from training date, and later used during the segmentation step. Thus, in this case segmentation is a result of classifying the image into different textures. Unsupervised segmentation is a much more difficult problem. Here one has to solve both parameter estimation, and segmentation. We first consider supervised segmentation, and then extend to the case of unsupervised segmentation. Our focus here is on the classification aspect rather than parameter estimation. We discuss and compare several relaxation schemes and their implementation on neural networks.

3.1 Previous Work

A textured image is characterized by two processes: the intensity process defines the intensity distribution within homogeneous regions, and the label process describes the distribution of different textures in the image. MRF models have been used in the past to characterize both these distributions [21, 27, 71]. The conditional distribution of the texture intensity process along with a prior model for the texture labels is then used to define a posterior distribution. The segmentation goal is then to maximize this posterior probability, i.e., *maximum aposteriori estimate* of the texture labels. In general, due to the non-convexity of this function, only an exhaustive search can guarantee the best solution. The associated computational complexity of this classification problem can be appreciated by the fact even for a small 128×128 image with two textures labels, the search for a correct classification involves a space of $2^{2^{14}}$!

Several algorithms, both deterministic as well as stochastic, have been studied to obtain reasonably good solutions without resorting to an exhaustive search of

the state space. While stochastic relaxation algorithms like simulated annealing are theoretically appealing, their implementation is however quite ad-hoc. This is due to the impractical nature of the cooling schedules that need to be followed [39] in order to guarantee conformity with the theoretical results. For a discussion on the use of simulated annealing for segmentation we refer to [71]. Derin and Eliott [27] use dynamic programming to optimize an alternate cost function (corresponding to pseudo-likelihood estimate). A hierarchical relaxation scheme is suggested by Cohen and Cooper in [21]. The main motivation of our study in this chapter is to develop parallel algorithms that can be implemented on neural networks. Such algorithms are desirable considering the complexity of the segmentation problem. The parameters of the network are obtained in terms of the cost function it is designed to optimize. For practical purposes networks with few interconnections are preferred because of the large number of processing units required in any image processing application. In this context MRF models play a useful role. They are typically characterized by local dependencies and symmetric interconnections which can be expressed in terms of energy functions using Gibbs-Markov equivalence.

The organization of this chapter is as follows: The next section describes the image model. A neural network algorithm is given in section 3.4. The use of stochastic learning automata in deterministic relaxation is discussed in 3.5. Experimental results and comparisons with some other algorithms are given in section 3.6. In 3.7 we discuss extensions to unsupervised segmentation, and make few observations about simultaneous parameter estimation and segmentation in 3.8.

3.2 Image Model

Let Ω denote such a set of grid points on an $M \times M$ lattice, i.e., $\Omega = \{(i,j)\ ,\ 1 \leq i,j \leq M\}$. Let $\{Y_s, s \in \Omega\ \}$ be a random process defined on this grid. In the following we use $\{L_s, s \in \Omega\}$ to denote the label process and $\{Y_s\ ,\ s \in \Omega\ \}$ for the zero mean intensity process.

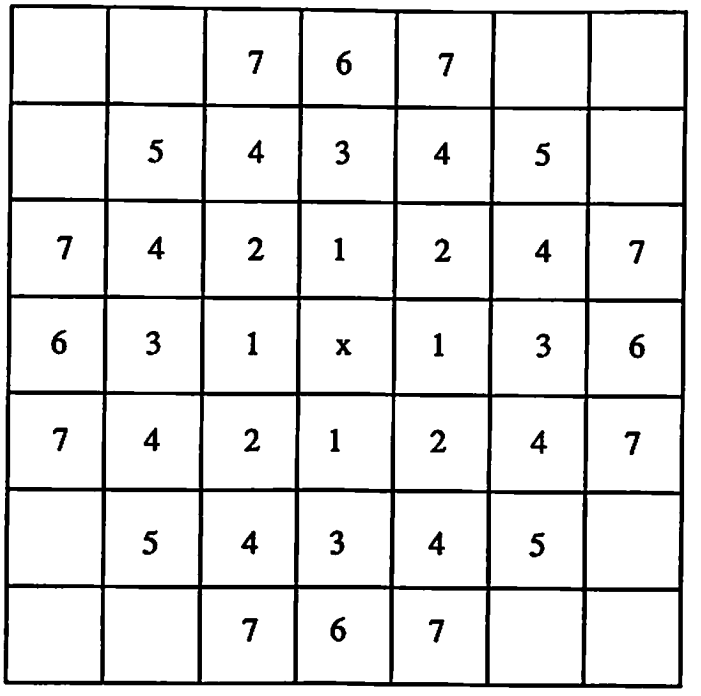


Figure 3.1: Structure of the GMRF model. The numbers indicate the order of the model relative to center location x [24].

3.2.1 Intensity Process

We model the intensity process $\{Y_s\}$ by a Gaussian Markov random field (GMRF). Depending on the neighborhood set one can construct a hierarchy of GMRF models as shown in Figure 3.1. The numbers indicate the order of the GMRF model relative to the center location x. Note that this defines a symmetric neighborhood set. We have noticed from our experience with many of the natural textures that a fourth order model captures most of the information regarding the intensity distribution. Increasing the order of the model only increases the amount of computation without having any significant effect on the nature of segmentation.

Let N_s denote the symmetric fourth order neighborhood of a site s. Let N^* be

the set of one sided shift vectors corresponding to the fourth order neighborhood.

$$\begin{aligned} N^* &= \{\tau_1, \tau_2, \tau_3, \cdots, \tau_{10}\} \\ &= \{(-1,0),(0,1),(-1,1),(1,1),(-2,0),(0,2),(-1,2),(1,2),(-2,1),(2,1)\} \end{aligned}$$

and

$$N_s = \{r : r = s \pm \tau, \tau \in N^*\} \tag{3.1}$$

where $s + \tau$ is defined as

$$s = (i,j), \tau = (x,y), s + \tau = (i + x, j + y)$$

Assuming that all the neighbors of s also have the same label as that of s, the conditional density of the intensity at the pixel s is

$$P(Y_s = y_s \mid Y_r = y_r, r \in N_s, L_s = l) = \frac{e^{-U(y_s \mid y_r, r \in N_s, l)}}{Z(l | y_r, r \in N_s)} \tag{3.2}$$

$$U(y_s \mid y_r, r \in N_s, l) = \frac{1}{2\sigma_l^2}(y_s^2 - 2 \sum_{r \in N_s} \Theta^l_{s,r} y_s y_r) \tag{3.3}$$

In (3.3), σ_l and Θ^l are the GMRF model parameters of the l-th texture class. A stationary GMRF model implies that the parameters satisfy $\Theta^l_{r,s} = \Theta^l_{r-s} = \Theta^l_{s-r} = \Theta^l_{\tau}$.

We view the image intensity array as composed of a set of overlapping $k \times k$ windows W_s, centered at each pixel $s \in \Omega$. In each of these windows we assume that the texture label L_s is homogeneous (all the pixels in the window belong to the same texture) and model the intensity distribution in the window by a fourth order stationary GMRF. Let $\mathbf{Y}^*_s$ denote the 2-D vector representing the zero mean intensity array in the window W_s. Using the Gibbs formulation and assuming a free boundary model, the joint probability density in the window W_s can be written as:

$$P(\mathbf{Y}^*_s = \mathbf{y}^*_s | L_s = l) = \frac{e^{-U_1(\mathbf{y}^*_s | l)}}{Z_1(l)} \tag{3.4}$$

where $Z_1(l)$ is the partition function and

$$U_1(y_s^*|l) = \frac{1}{2\,\sigma_l^2} \sum_{r \in W_s} \left\{ y_r^2 - \sum_{\tau \in N^* | r+\tau \in W_s} \Theta_\tau^l y_r (y_{r+\tau} + y_{r-\tau}) \right\} \tag{3.5}$$

Estimation of GMRF parameters

For supervised classification, the GMRF parameters are precomputed from training samples of the textures. There are several ways of estimating these parameters and a comparison of different schemes can be found in [19]. The method that we have used is based on the least square estimates.

Let Ω be the lattice under consideration and let Ω_I be the interior region of Ω, i.e.,

$$\Omega_I = \Omega - \Omega_B, \Omega_B = \{s = (i,j), s \in \Omega \text{ and } s \pm \tau \notin \Omega \text{ for at least some } \tau \in N^*\} \tag{3.6}$$

Let

$$Q_s = [y_{s+\tau_1} + y_{s-\tau_1}, \cdots, y_{s+\tau_{10}} + y_{s-\tau_{10}}]^T \tag{3.7}$$

Then the least square estimates of the parameters are [56]

$$\hat{\Theta} = [\sum_{\Omega_I} Q_s Q_s^T]^{-1} [\sum_{\Omega_I} Q_s y_s] \tag{3.8}$$

$$\hat{\sigma} = \frac{1}{M^2} \sum_{\Omega_I} [y_s - \Theta^T Q_s]^2 \tag{3.9}$$

One common problem with all GMRF parameter estimation schemes is that none of them can guarantee both consistency (estimates converging to the true values of the parameters) and stability (covariance matrix for the joint probability density must be positive definite) together, without additional constraints. One can enforce the stability constraint by reformulating the estimation problem as a constrained optimization problem [92].

3.2.2 Label Process

The texture labels are assumed to obey a first or second order discrete Markov model with a single parameter β, which measures the amount of clustering between adjacent pixels. If $\hat{N}_s$ denotes the appropriate neighborhood for the label field, then we can write the distribution function for the texture label at site s conditioned on the labels of the neighboring sites as:

$$P(L_s = l_s | L_r = l_r \ , \ r \in \hat{N}_s) = \frac{e^{-U_2(l_s \mid l_r)}}{Z_2} \tag{3.10}$$

where Z_2 is a normalizing constant and

$$U_2(l_s \mid l_r, \ r \in \hat{N}_s) = -\beta \sum_{r \in \hat{N}_s} \delta(l_s - l_r) \ , \ \beta > 0 \tag{3.11}$$

In (3.11), β determines the degree of clustering, and $\delta(i-j)$ is the Kronecker delta. From (3.4) and (3.10) we can write the following expression for the conditional posterior distribution of the labels:

$$P(L_s \mid \mathbf{Y}_s^*, \ L_r, \ r \in \hat{N}_s) = \frac{P(\mathbf{Y}_s^* \mid L_s) \ P(L_s \mid L_r, \ r \in \hat{N}_s)}{P(\mathbf{Y}_s^*)} \tag{3.12}$$

Since $\mathbf{Y}_s^*$ is known, the denominator in (3.12) is just a constant. The numerator is a product of two exponential functions and can be expressed as,

$$P(l_s \mid \mathbf{y}_s^*, \ l_r, \ r \in \hat{N}_s) = \frac{1}{Z_p} \ e^{-U_p(l_s \mid \mathbf{y}_s^*, \ l_r, \ r \in \hat{N}_s)} \tag{3.13}$$

where Z_p is the partition function and $U_p(\cdot)$ is the posterior energy:

$$U_p(l_s \mid \mathbf{y}_s^*, \ l_r, \ r \in \hat{N}_s) = w(l_s) + U_1(\mathbf{y}_s^* \mid l_s) + U_2(l_s \mid l_r, \ r \in \hat{N}_s) \tag{3.14}$$

Note that the second term in (3.14) relates the observed pixel intensities to the texture labels and the last term specifies the label distribution. The bias term $w(L_s) = \log Z_1(L_s)$ is dependent on the texture class and it can be explicitly evaluated for the GMRF model considered here using the toroidal assumption (the computations become very cumbersome if toroidal assumptions are not made).

An alternate approach is to estimate the bias from the histogram of the data as suggested by Geman and Graffigne [40].

3.3 Classification as Optimization

The goal of our computation is to extract the unobserved label process $\{L_s\}$ from the intensity data $\{y_s\}$. This is usually formulated as an optimization problem involving minimization of an objective function. The choice of the objective function has considerable effect on the nature of results obtained, and should take into consideration the applications.

3.3.1 Maximum *A Posteriori* Estimate

One of the commonly used criterion is to maximize the posterior probability, the maximum *a posteriori* (MAP) estimate:

$$P(\mathbf{L} \mid \mathbf{Y}^*) = \frac{P(\mathbf{Y}^* \mid \mathbf{L})\ P(\mathbf{L})}{P(\mathbf{Y}^*)} \tag{3.15}$$

We have used the notation $\mathbf{Y}^*$ instead of $\mathbf{Y}$ to denote the observed intensity image to make a distinction between the conditional probability $P(y_s|l_s)$, and the one that we are using in our computations $P(\mathrm{y}_s^*|l_s)$.

In practice, however, it is impossible to maximize the above function. Algorithms such as simulated annealing make use of local conditionals in (3.13) in the computations instead of evaluating the right hand side of (3.15). As noted earlier, however, almost all implementations of this algorithm have been quite ad-hoc, although they provide better results than most other known techniques. Instead of trying to optimize (3.15) we consider a cost function in section 3.4 whose local minima correspond to sub-optimal MAP estimates.

3.3.2 Maximum Posterior Marginal

Another objective function of considerable interest is the one which minimizes the expected percentage misclassification per pixel. A solution to this can be

obtained by maximizing the marginal posterior distribution of L_s given the observation y_s^* for each pixel s:

$$P(L_s = l_s|\mathbf{Y}_s^* = \mathbf{y}_s^*) \propto \sum_{\mathbf{l}|L_s=l_s} P(\mathbf{Y}_s^* = \mathbf{y}^*|\mathbf{l})\ P(\mathbf{L} = \mathbf{l}) \tag{3.16}$$

where the summation extends over all possible label configurations while keeping the label at site s constant. We refer to the result as the MPM solution. For a general discussion on MPM we refer to [75], and for details regarding its implementation in the context of texture classification we refer to [71]. For our discussion regarding MPM in this chapter, it is suffice to note that one can obtain an MPM solution by sampling out of the posterior distribution of the texture labels (3.13) and choose the class for each pixel that occurred more often than others.

3.4 Deterministic Relaxation Using a Neural Network

Here we discuss finding a sub-optimal solution to the MAP estimate using a neural network algorithm, and its relation to the Iterated Conditional Model (ICM) algorithm of Besag [5]. Figure 3.2 shows the network architecture used for classification. The energy function which the network minimizes is obtained from the image model discussed in the previous section. For convenience of notation let $U_1(i,j,l) = U_1(\mathbf{y}_s^*, L_s = l) + w(l)$ where $s = (i,j)$ denotes a pixel site and $U_1(\cdot)$ and $w(l)$ are as defined in (3.14). The network consists of K layers, each layer arranged as an $M \times M$ array, where K is the number of texture classes in the image and M is the dimension of the image. The elements (neurons) in the network are assumed to be binary and are indexed by (i,j,l) where $(i,j) = s$ refers to their position in the image and l refers to the layer. The (i,j,l)-th neuron is said to be ON if its output V_{ijl} is 1, indicating that the corresponding site $s = (i,j)$ in the image has the texture label l. Let $T_{ijl;i'j'l'}$ be the connection strength between the neurons (i,j,l) and (i',j',l') and I_{ijl} be the input bias current. Then a general form for the energy of the network is [47]

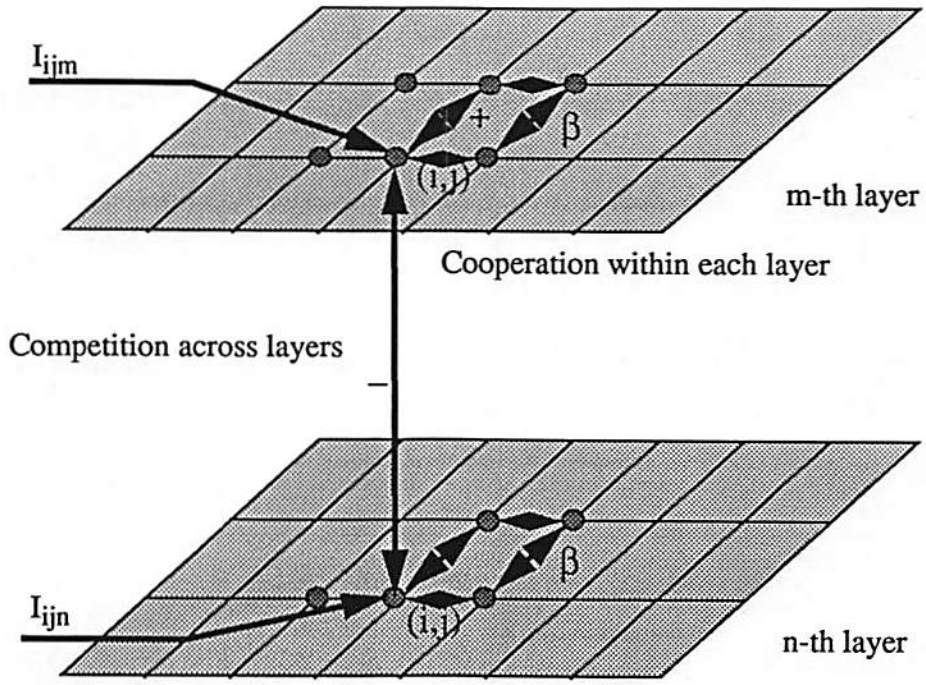


Figure 3.2: Network architecture

$$E = -\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{l=1}^{K}\sum_{i'=1}^{M}\sum_{j'=1}^{M}\sum_{l'=1}^{K} T_{ijl;i'j'l'} V_{ijl} V_{i'j'l'} - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{l=1}^{K} I_{ijl} V_{ijl} \qquad (3.17)$$

From our discussion in section 2 we note that the solution for the MAP estimate can be obtained by minimizing (3.15). Here we approximate the posterior energy by

$$U(\mathbf{l}|\mathbf{y}^*) = \sum_s \{U(\mathbf{y}_s^*|l_s) + w_{l_s} + U_2(l_s)\} \qquad (3.18)$$

and the corresponding energy to be minimized can be written as

$$E = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{l=1}^{K} U_1(i,j,l)V_{ijl} - \frac{\beta}{2}\sum_{l=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{(i',j')\in\hat{N}_{ij}} V_{i'j'l}V_{ijl} \tag{3.19}$$

where $\hat{N}_{ij}$ is the neighborhood of site (i,j) (same as the $\hat{N}_s$ in section 2). In (3.19), it is implicitly assumed that each pixel site has a unique label, i.e. only one neuron is active in each column of the network. This constraint can be implemented in different ways. For the deterministic relaxation algorithm described below, a simple method is to use a *winner-takes-all* circuit for each column so that the neuron receiving the maximum input is turned on and the others are turned off. Alternately a penalty term can be introduced in (3.19) to represent the constraint as in [47]. From (3.17) and (3.19) we can identify the parameters for the network,

$$T_{ijl;i'j'l'} = \begin{cases} \beta & \text{if } (i',j') \in \hat{N}_{ij}, \forall\, l = l' \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

and the bias current

$$I_{ijl} = -\,U_1(i,j,l) \tag{3.21}$$

3.4.1 Deterministic Relaxation

The above equations (3.20) and (3.21) relate the parameters of the network to that of the image model. The connection matrix for the above network is symmetric and there is no self feedback, i.e. $T_{ijl;ijl} = 0, \forall i,j,l$. Let u_{ijl} be the potential of neuron (i,j,l). (Note that l is the layer number corresponding to texture class l) , then

$$u_{ijl} = \sum_{i'=1}^{M}\sum_{j'=1}^{M}\sum_{l'=1}^{K} T_{ijl;i'j'l'}\, V_{i'j'l'} + I_{ijl} \tag{3.22}$$

In order to minimize (3.19), we use the following updating rule:

$$V_{ijl} = \begin{cases} 1 & \text{if } u_{ijl} = \min_{l'}\{u_{ijl'}\} \\ 0 & \text{otherwise} \end{cases} \tag{3.23}$$

This updating scheme ensures that at each stage the energy decreases. Since the energy is bounded, the convergence of the above system is assured but the stable state will in general be a local optimum.

This network model is a version of the ICM that we discussed in section 2.2.3. This algorithm maximizes the local conditional probability $P(L_s = l|\mathbf{y}_s^*, l_{s'}, s' \in \hat{N}_s)$ during each iteration . It is a local deterministic relaxation algorithm that is very easy to implement. We observe that in general any algorithm based on MRF models can be easily mapped on to neural networks with local interconnections. The main advantage of this deterministic relaxation algorithm is its simplicity, and it usually converges within 20-30 iterations.

3.5 Stochastic Learning and Deterministic Relaxation

The main advantage of deterministic relaxation is its speed and ease of implementation. In comparison, stochastic algorithms such as simulated annealing give better results at the expense of increasing the computations. Here we consider the possibility of improving the performance of deterministic relaxation by introducing a stochastic element in its search.

The following discussion is motivated from the theory of stochastic learning automata [80]. An automaton is a decision maker operating in a random environment. A stochastic automaton (Figure 3.3) can be defined by a quadruple (α, Q, T, R) where $\alpha = \{\alpha_1, \ldots, \alpha_N\}$ is the set of available actions to the automaton. The action selected at time t is denoted by $\alpha(t)$. $Q(t)$ is the state of the automaton at time t and consists of the action probability vector $\mathrm{p}(t) = [p_1(t), \ldots, p_N(t)]$ where $p_i(t) = \text{prob}\,(\alpha(t) = \alpha_i)$ and $\sum_i p_i(t) = 1\ \forall t$. The environment responds to the action $\alpha(t)$ with a $\lambda(t) \in R$, R being the set of

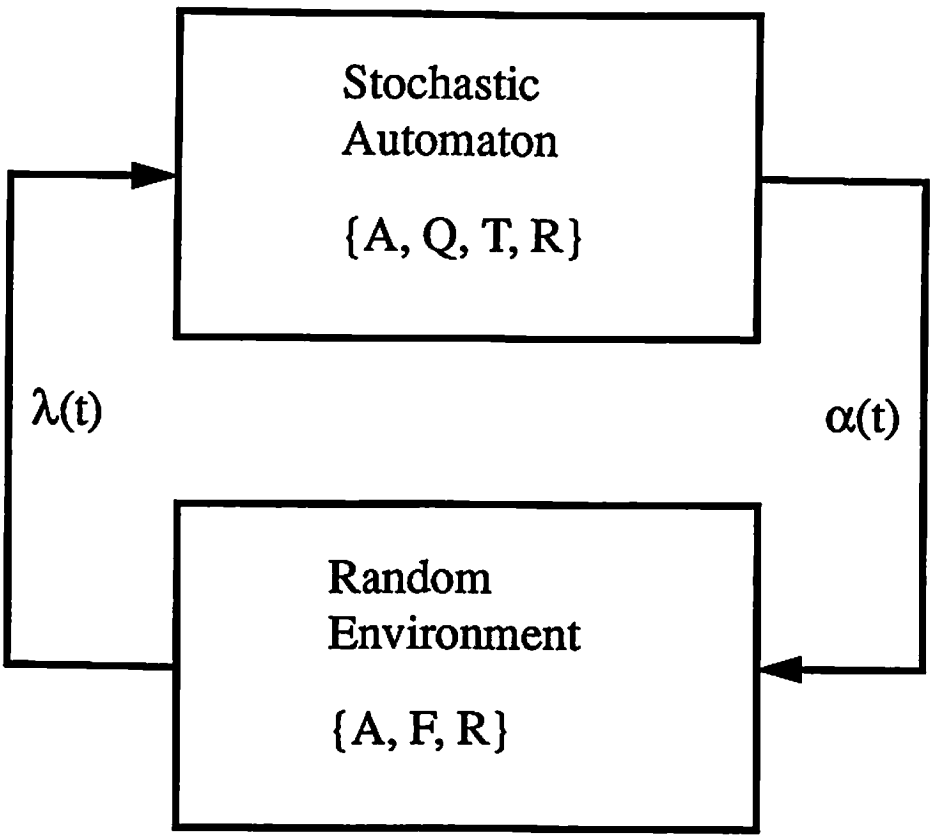


Figure 3.3: Stochastic learning automaton

environment's responses. The state transitions of the automaton are governed by the learning algorithm T, $Q(t+1) = T(Q(t), \alpha(t), \lambda(t))$. Without loss of generality it can be assumed that $R = [0, 1]$, i.e., the responses are normalized to lie in the interval [0,1], '1' indicating a complete *success* and '0', total *failure*. The goal of the automaton is to converge to the optimal action, i.e. the action which results in the maximum expected reward. Again without loss of generality let α_1 be the optimal action and $d_1 = E[\lambda(t) \mid \alpha_1] = \max_i\{E[\lambda(t) \mid \alpha_i]\}$. At present no learning algorithms exist which are optimal in the above sense. However we can choose the parameters of certain learning algorithms so as to realize a response as close to the optimum as desired. This condition is called ϵ-optimality.

If $M(t) \triangleq E[\lambda(t) \mid p(t)]$, then a learning algorithm is said to be ϵ-optimal if it results in a $M(t)$ such that

$$\lim_{t \to \infty} E[M(t)] > d_1 - \epsilon \tag{3.24}$$

for a suitable choice of parameters and for any $\epsilon > 0$. One of the simplest learning schemes is the Linear Reward-Inaction rule , L_{R-I} . Suppose at time t we have $\alpha(t) = \alpha_i$ and if $\lambda(t)$ is the response received then according to the L_{R-I} rule,

$$\begin{aligned} p_i(t+1) &= p_i(t) + a\ \lambda(t)\ [1 - p_i(t)] \\ p_j(t+1) &= p_j(t)[1 - a\ \lambda(t)],\ \forall j \neq i \end{aligned} \tag{3.25}$$

where a is a parameter of the algorithm controlling the learning rate. Typical values for a are in the range 0.01-0.1. It can be shown that this L_{R-I} rule is ϵ-optimal in all stationary environments i.e., there exists a value for the parameter a so that condition (3.24) is satisfied. Some properties of this algorithm are explored in the exercises and for a detailed discussion on this and other non-linear learning algorithms we refer the reader to the book by Lakshmivarahan[65]

Collective behavior of a group of automata has also been studied. Consider a team of N automata $A_i (i = 1, .., N)$ each having r_i actions $\alpha^i = \{\alpha_1^i \ldots \alpha_{r_i}^i\}$. At any instant t each member of the team makes a decision $\alpha^i(t)$. The environment responds to this by sending a reinforcement signal $\lambda(t)$ to all the automata in the group. This situation represents a co-operative game among a team of automata with an identical pay-off. All the automata update their action probability vectors according to (3.25) using the same learning rate and the process repeats. Local convergence results can be obtained in case of stationary random environments. Variations of this rule have been applied to complex problems like decentralized control of Markov Chains [104] and relaxation labelling [93]

The texture classification discussed in the previous sections can be treated as a relaxation labelling problem and stochastic automata can be used to learn the labels (texture class) for the pixels. A learning automaton is assigned to each of the pixel sites in the image. The actions of the automata correspond to selecting a label for the pixel site to which it is assigned. Thus for a K class problem each

automaton has K actions and a probability distribution over this action set. Initially the labels are assigned randomly with equal probability. Since the number of automata involved is very large, it is not practical to update the action probability vector at each iteration. Instead we combine the iterations of the neural network described in the previous section with the stochastic learning algorithm. This results in an iterative hill climbing type algorithm which combines the fast convergence of deterministic relaxation with the sustained exploration of the stochastic algorithm. The stochastic part prevents the algorithm from getting trapped in local minima and at the same time "learns" from the search by updating the state probabilities. However unlike simulated annealing, we cannot guarantee convergence to the global optimum. Each cycle now has two phases: The first phase consists of the deterministic relaxation network converging to a solution. The second phase consists of the learning network updating its state, the new state being determined by the equilibrium state of the relaxation network. A new initial state is generated by the learning network depending on its current state and the cycle repeats. Thus relaxation and learning alternate with each other. After each iteration the probability of the more stable states increases and because of the stochastic nature of the algorithm the possibility of getting trapped in undesirable local minima is reduced. The algorithm is summarized below.

3.5.1 Learning Algorithm

Let the pixel site be denoted by $s \in \Omega$ and the number of texture classes be K. Let A_s be the automaton assigned to site s and the action probability vector of A_s be $\mathrm{p}_s(t) = [p_{s,1}(t), \ldots, p_{s,K}(t)]$ and $\sum_i p_{s,i}(t) = 1 \forall s, t$, where $p_{s,l}(t) =$ prob(label of site $s = l$). The steps in the algorithm are,

1. Initialize the action probability vectors of all the automata

$$p_{s,l}(0) = 1/K, \ \forall s, l$$

Initialize the iteration counter to 0.

2. Choose an initial label configuration sampled from the distribution of these probability vectors.

3. Start the neural network of section 3 with this configuration.

4. Let l_s denote the label for site s at equilibrium. Let the current time (iteration number) be t. Then the action probabilities are updated as follows

$$\begin{aligned} p_{s,l_s}(t+1) &= p_{s,l_s}(t) + a\ \lambda(t)\ [1 - p_{s,l_s}(t)] \\ p_{s,j}(t+1) &= p_{s,j}(t)[1 - a\ \lambda(t)],\ \forall j \neq l_s \text{ and } \forall\ s \end{aligned} \tag{3.26}$$

The response $\lambda(t)$ is derived as follows: Suppose the present label configuration resulted in a lower energy state compared to the previous one then it results in a $\lambda(t) = \lambda_1$ and if the energy increases we have $\lambda(t) = \lambda_2$ with $\lambda_1 > \lambda_2$. In our simulations we used $\lambda_1 = 1$ and $\lambda_2 = 0.25$.

5. Generate a new configuration from this updated label probabilities, increment the iteration counter and go to step 3.

Thus the system consists of two layers, one for relaxation and the other for learning. The relaxation network is similar to the one considered in section 3, the only difference is that the initial state is decided by the learning network. The learning network consists of a team of automata and learning takes place at a much lower speed than the relaxation, with fewer number of updatings. The probabilities of the labels corresponding to the final state of the relaxation network are increased according to (3.26). Using these new probabilities a new configuration is generated. Since the response does not depend on time, this corresponds to a stationary environment and as we have noted before this L_{R-I} algorithm can be shown to converge to a stationary point, not necessarily to the global optimum.

	leather	grass	pig skin	sand	wool	wood
θ_1	0.5689	0.5667	0.3795	0.5341	0.4341	0.5508
θ_2	0.2135	0.3780	0.4528	0.4135	0.2182	0.2498
θ_3	-0.1287	-0.2047	-0.1117	-0.1831	-0.0980	-0.1164
θ_4	-0.0574	-0.1920	-0.1548	-0.2050	-0.0006	-0.1405
θ_5	-0.1403	-0.1368	-0.0566	-0.1229	-0.0836	-0.0517
θ_6	-0.0063	-0.0387	-0.0494	-0.0432	0.0592	0.0139
θ_7	-0.0052	0.0158	-0.0037	0.0120	-0.0302	-0.0085
θ_8	-0.0153	0.0075	0.0098	0.0111	-0.0407	-0.0058
θ_9	0.0467	0.0505	0.0086	0.0362	0.0406	-0.0008
θ_{10}	0.0190	0.0496	0.0233	0.0442	-0.0001	0.0091
σ^2	217.08	474.72	79.33	91.44	126.22	14.44

Table 3.1: GMRF texture parameters

3.6 Experimental Results

The segmentation results using the above algorithms are given on two examples. The parameters σ_l and Θ_l corresponding to the fourth order GMRF for each texture class were pre-computed from 64×64 images of the textures. The local mean (in a 11×11 window) was first subtracted to obtain the zero mean texture and the least square estimates [20] of the parameters were then computed from the interior of the image. The parameter values for the different textures used in our experiments is given in Table 3.1.

The first step in the segmentation process involves computing the Gibbs energies $U_1(\mathbf{y}_s^*|L_s)$ in (3.5). This is done for each texture class and the results are stored. For computational convenience these $U_1(\cdot)$ values are normalized by dividing by k^2, where k is the size of the window. To ignore the boundary effects, we set $U_1 = 0$ at the boundaries. We have experimented with different window sizes and larger windows result in more homogeneous texture patches but the boundaries between the textures are distorted. The results reported here are based on windows of size 11×11 pixels. The bias term $w(l_s)$ can be estimated using the histogram of the image data [40] but we obtained these values by trial and error.

The choice of β plays an important role in the segmentation process and its value depends on the magnitude of the energy function $U_1(\cdot)$. Various values

of β ranging from 0.2-3.0 were used in the experiments. In the deterministic algorithm it is preferable to start with a small β and increase it gradually. Large values of beta usually degrade the performance.

The nature of the segmentation results depends on the order of the label model. It is preferable to choose the first order model for the stochastic algorithms if we know apriori that the boundaries are either horizontal or vertical. However for the deterministic rule and the learning scheme the second order model results in more homogeneous classification.

The performance of the deterministic relaxation rule of section 3 also depends on the initial state and we have investigated two different initial conditions. The first one starts with a label configuration $\mathbf{L}$ such that $L_s = l_s$ if $U_1(\mathbf{y}_s^* | l_s) = \min_{l_k}\{U_1(\mathbf{y}_s^* \mid l_k)\}$. This corresponds to maximizing the probability $P(\mathbf{y}^* | l)$. The second choice for the initial configuration is a randomly generated label set. Results for both the cases are provided and we observe that the random choice often leads to better results.

Finally a note on implementing the relaxation itself: during updating we assume that neighboring pixels are not updated simultaneously. Otherwise it is no longer possible to guarantee that the energy decreases during each change of state. This is not of much concern in analog network as the time variable there is continuous, and a simultaneous change of state of neighbors is physically a rare event. For synchronous updating, consider the energy function of the following form [1]:

$$E(t) = -\sum_{i,j,j\neq i} T_{ij} V_i(t) V_j(t-1) \tag{3.27}$$

where t refers to time instants. This energy function decreases monotonically at every instant, except when it has reached an equilibrium point, or a limit cycle with alternating states. The equilibrium states are the same as for sequential updating (when neighbors are not updated simultaneously). However the limit cycles do not have any analogues in the sequential (or asynchronous updating) case, and might result in some undesirable states [29]. In particular, local minima of the energy function in (3.27) might be a maximum in the corresponding energy function for asynchronous updating.

In the examples below the following learning parameters were used: Learning

rate $a = 0.05$, reward/penalty parameters $\lambda_1 = 0.25$ and $\lambda_2 = 1.0$. In the following MLE refers to the classification result obtained by maximizing $P(y_s^*|L_s)$ at each location s.

Example 1: This is a two class problem consisting of grass and leather textures. The image is of size 128 × 128 and is shown in Figure 3.4(a). Figure 3.4(b)the deterministic algorithm discussed in section 3.4 is shown with the MLE as the initial state. Figure 3.4(c) shows the result with a random initial state. Notice that in this case the final result has fewer misclassified regions than in 3.4(b) and this was observed to be true in general. The result of the learning algorithm is shown in Figure 3.4(d), and for comparison we also give the results of MAP using simulated annealing and the MPM solution (Figures 3.4(e)-(f)). In all the cases we used $\beta = 0.6$.

Example 2: This is a 256 × 256 image (Figure 3.5(a)) having six textures: calf, grass, wool, wood, pig skin and sand. This is a difficult problem in the sense that three of the textures (wool, pig skin and sand) have almost identical characteristics and are not easily distinguishable even by the human eye. The MLE solution is shown in Figure 3.5(b) and Figure 3.5(c) shows the result obtained by the deterministic relaxation network MLE as initial condition. Figure 3.5(d) gives the result with random initial configuration. Introducing learning improves the the performance of deterministic relaxation (Figure 3.5(e)), and Figure 3.5(f) shows the results of hierarchical implementation of the learning scheme (see section 3.6.1). Again, for comparison we show the results of MAP using simulated annealing and MPM solutions in Figure 3.5(g)-(h).

The accuracy of classification for different algorithms is shown in Table 3.2. As is evident from the table, introducing learning considerably improves the performance of the network. With learning, the results compare favorably with those of computationally expensive schemes like simulated annealing and the MPM algorithm.

3.6.1 Hierarchical Segmentation

The various segmentation algorithms described in the previous sections can be easily extended to hierarchical structures wherein the segmentation is carried out

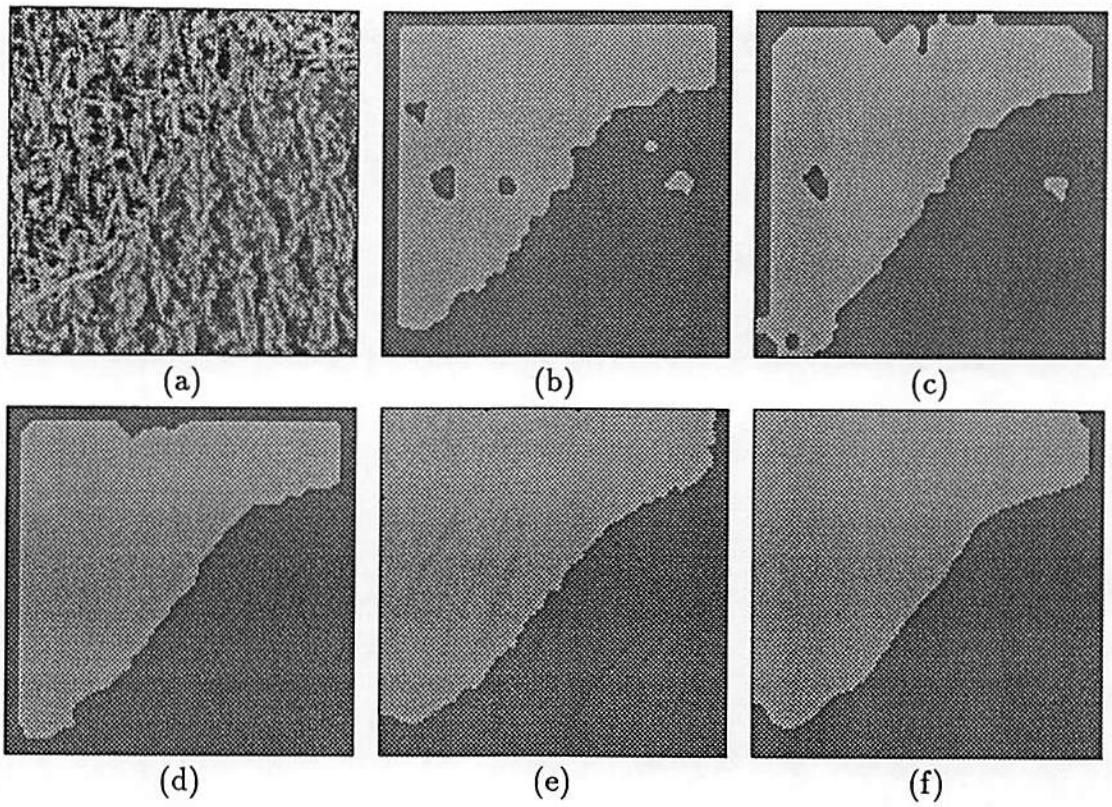


Figure 3.4: Two class segmentation problem, (a) original image consisting of two textures. Classification results obtained using different algorithms are shown in (b)-(f), (b) deterministic relaxation with maximum likelihood solution as initial condition, (c) with random initial condition, (d) network with stochastic learning (e) MAP estimate using simulated annealing and (e) MPM solution.

at different levels - from coarse to fine. The energy functions are modified to take care of the coupling between the adjacent resolutions of the system. Consider a K-stage hierarchical system, with stage 0 representing the maximum resolution level and stage $K-1$ being the coarsest level. The energy corresponding to the k-th stage is denoted by $U_1^k(s,l)$ and $U_2^k(s)$ (equations 3.5 and 3.11). The size of the window used in computing the joint energy potential $U_1^k(\cdot)$ increases with the index k. The potential U_2 is modified to take care of the coupling as below,

$$U_2^k(s) = -\beta \sum_{t' \in D_s^k} \delta\left(L^k(s) - L^k(t')\right) + \beta_k \left(\delta(L^k(s) - L^{k+1}(s)\right) + w\left(L(s)\right) \quad (3.28)$$

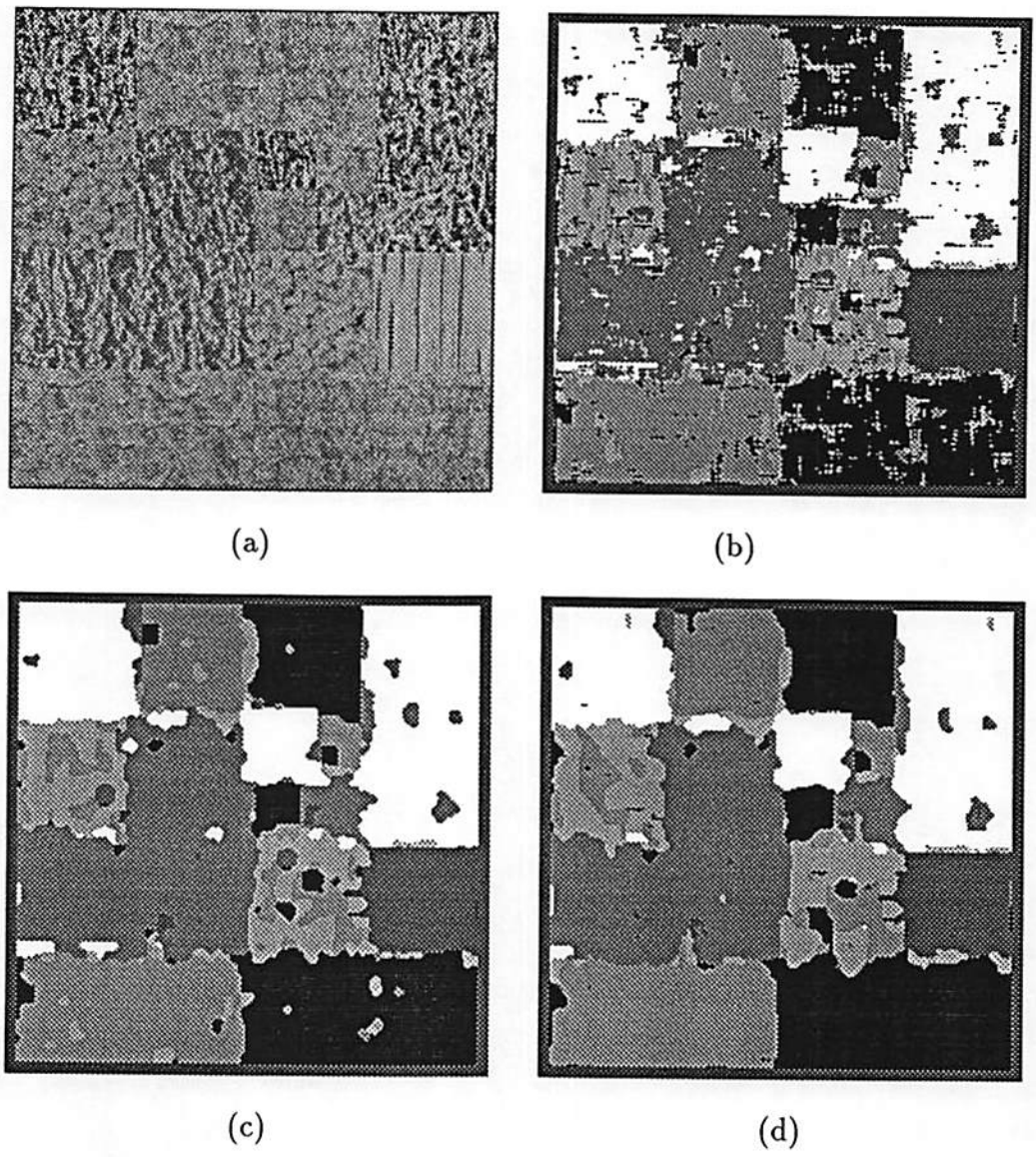
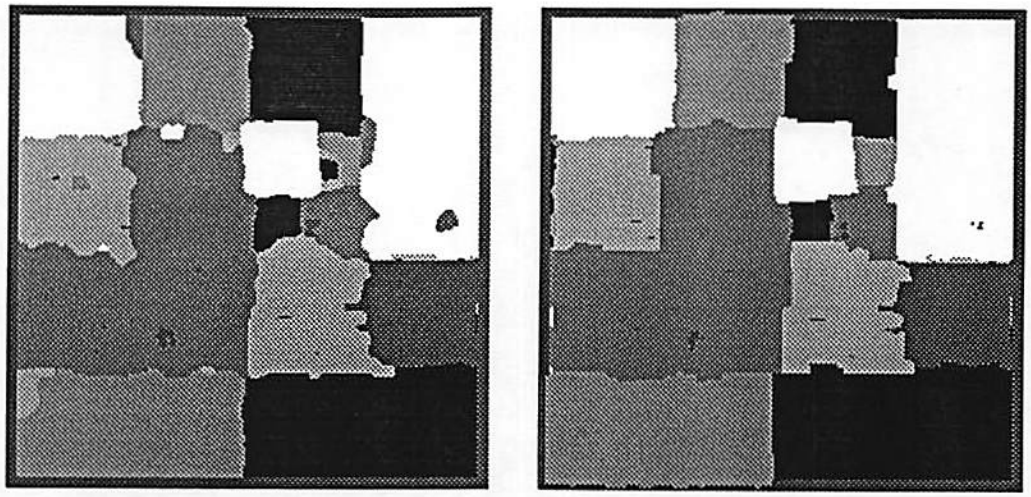
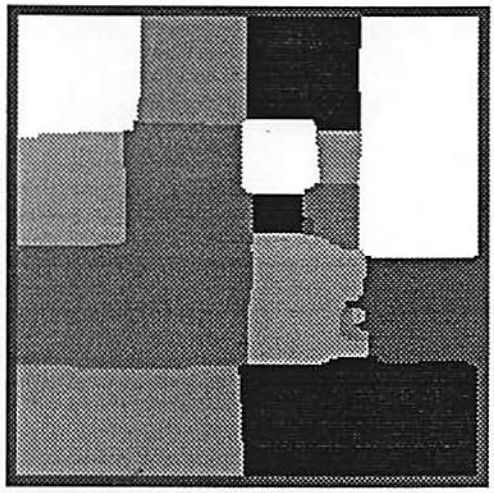


Figure 3.5: (a) Original image with six textures, (b) MLE solution (c) deterministic relaxation solution with (b) as initial condition, (d) deterministic relaxation with random initial condition

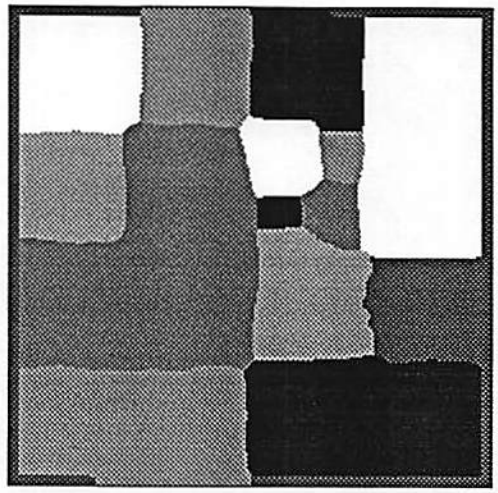


(e)

(f)



(g)



(h)

Figure 3.5: (contd.) (e) deterministic relaxation with learning (f) hierarchical implementation (g) simulated annealing and (h) MPM solution.

Algorithm	Percentage Error
Maximum likelihood estimate	22.17
Neural network (MLE as initial state)	16.25
Neural network (random initial state)	14.74
Simulated annealing	6.72
MPM	7.05
Neural network with learning	8.70
Hierarchical network	8.21

Table 3.2: Percentage misclassification for example 2 (six class problem).

where $0 \leq k < K-1$, and $L^k(s)$ is the label for the site s in the k-th stage, β_k is the coupling coefficient between the stages $k+1$ and k. $D^k(s)$ is the appropriate neighborhood set for the k-th stage. The result of segmentation on the six class problem with $K = 2$ and using the learning algorithm is shown in Figure 3.5(f).

3.7 Extensions to Unsupervised Segmentation

We consider extending the results to the case when neither the model parameters nor the number of textures in the image are known. For computational simplicity, we use the second order GMRF model for the image intensity instead of the fourth order model discussed in section 3.2. The given image is first partitioned into a number of small non-overlapping sub-images and the GMRF parameters are computed for each of these regions. A simple clustering method is then used to merge these regions. From these clustered regions, the parameters are recomputed and the new values are used in the relaxation schemes discussed in the previous sections.

We noted in 3.2.1 that an optimization algorithm is normally used to obtain stable estimates. The parameters computed in supervised learning used one such procedure to make sure that the covariance matrix is positive definite. The same least square technique is used here also for estimating the GMRF parameters [56]. However we do not check for the stability of the estimates obtained. The reason being, the parameters are used in obtaining certain measures for segmentation, and is appropriate to use computationally less demanding schemes which can provide reasonably good estimates. The particular choice of least-squares

estimate is motivated by this simplicity-stability trade-off.

3.7.1 Clustering

The given image is divided into a number of non-overlapping subimages. For each of these subimages the corresponding feature vectors are estimated as described in the previous section. It is assumed that all these subimages are homogeneous. A normalized Eucledian distance measure is defined for these vectors as

$$d(F^i, F^j) = \sum_k \frac{(f_k^i - f_k^j)^2}{(f_k^i)^2 + (f_k^j)^2} \tag{3.29}$$

A simple clustering is done based on this distance measure. First the maximum distance between any two regions in the image is found as

$$d_{max} = \max_{i,j} d(F^i, F^j)$$

The regions are now grouped such that any two subimages i and j belonging to the same class satisfy

$$d(F^i, F^j) < \rho \;\; d_{max} \tag{3.30}$$

where ρ is a clustering parameter. Since ρ affects the number of clusters that are formed, a good guess of ρ should be based on the knowledge about the approximate number of classes present in the image. In our experiments we used a simple heuristic $\rho = 1/(\text{approximate number of classes})$. In the above clustering process all isolated regions are marked as ambiguous. Also all those regions which satisfy the criterion (3.30) for two different classes should be labelled ambiguous. Usually the boundary regions which have more than one texture fall into this class. Note that alternate schemes like k-mean clustering can also be used in obtaining such a coarse segmentation, and we believe that the choice of distance measure is not a critical issue here.

From the clustered regions so obtained, the parameters are recomputed. These values are then used in the relaxation algorithms discussed in the previous sections to obtain finer segmentation.

3.7.2 Experimental Results

In the experiments described below, the subimage size was chosen to be 32x32. The value of the clustering parameter depends on the number of texture classes present. To eliminate very small isolated regions one can use a penalty function in the relaxation algorithm which prohibits small clusters from being formed [38]. We found it convenient to use a smoothing filter (size 3x3 or 5x5) to do the same. An interesting observation is that with this kind of "post-processing", the performance of both the deterministic and stochastic relaxation algorithms is comparable. The results given below correspond to those obtained after performing the smoothing. However the boundaries obtained by the stochastic algorithms are more accurate.

Example 1: (Grass and leather texture) Figure 3.6 shows this mosaic and is a 128 x 128 image and $\rho = 0.5$. Figure 3.6(b) shows the result of coarse clustering described in section 3.7.1. Figure 3.6(c) is the result of the deterministic relaxation. This normally takes about 10-20 iterations. The result of using learning in the deterministic relaxation is shown in Figure 3.6(d). About 10 learning cycles are used in this experiment. Figure 3.6(e) gives the result for the MPM algorithm after about 500 iterations. The boundary obtained by the MPM is the most accurate and also there are no misclassifications inside the homogeneous regions.

Example 2: (Grass, Raffia and Wood) Figure 3.7(a) shows this image and Figure 3.7(b) gives the coarse clustering obtained using $\rho = 0.3$. Note the presence of an ambiguous region (dark region at the top), which could not be classified into any of the other classes. The results of the various algorithms are shown in Figures 3.7(c)- 3.7(e). Here again the best result is obtained by the MPM algorithm.

3.8 Discussions and Conclusions

Unsupervised segmentation is a difficult problem. The approach presented here is based on assuming a prior model for the texture data. Then the image is segmented by computing the difference between these assumed model parameters.

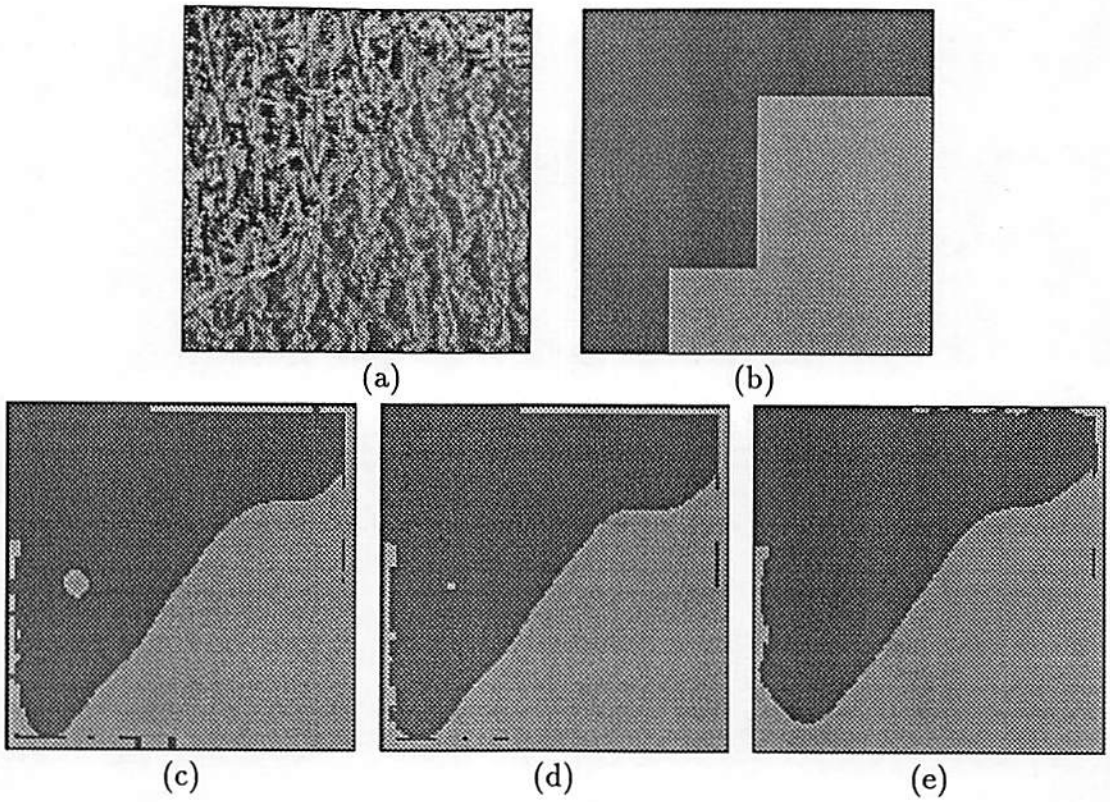


Figure 3.6: Unsupervised segmentation of an image consisting of two textures (grass and leather) (a) original image (b) coarse clustering (c) deterministic relaxation (d) stochastic learning (e) MPM result

Since at the beginning of segmentation no information is available regarding the distribution of homogeneous regions, we considered blocks of square pixels for estimating model parameters, and obtained a coarse segmentation first. Later these parameters are re-computed and used in pixel based relaxation schemes to obtain more accurate boundaries.

Geman et al. [38] follow a similar approach but they compute the boundaries at the pixel level directly. Also they do not assume any specific model for the intensity data, and instead compute various statistics from the raw data as well as from its transformations. Then they use the *Kolmogorov-Smirov* distance to compute the disparity between blocks of pixels, which is used in the segmentation process. Additional constraints are also added to the cost function to eliminate small isolated regions in the segmented image.

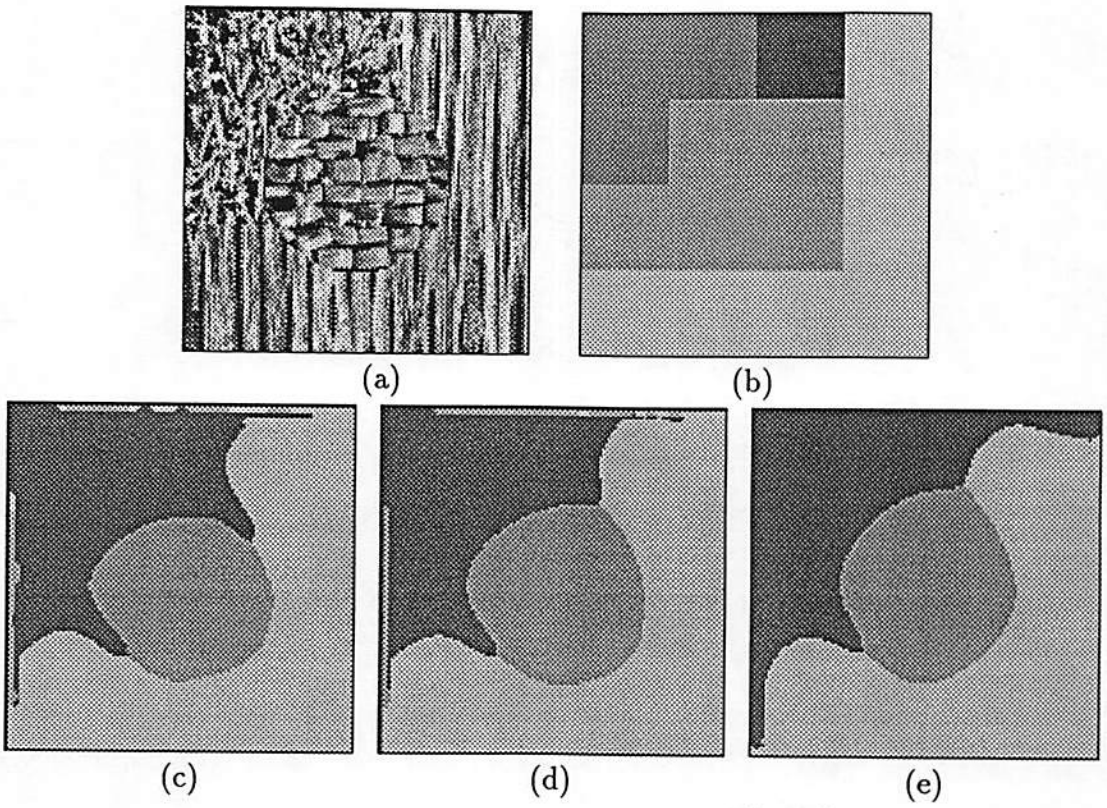


Figure 3.7: Unsupervised segmentation of an image having three textures (Grass, Raffia and Wood). (a) original image (b) coarse clustering. Note the presence of an ambiguous region (darkest region at the top) (c) deterministic relaxation (d) stochastic learning (e) MPM result

Separating estimation from segmentation simplifies the problem and enables computationally manageable algorithms. However such a procedure does not lend itself easily to rigorous analysis. Recently Lakshmanan and Derin [64] have proposed an interesting algorithm which combines parameter estimation with segmentation. We give a brief description of their algorithm, followed by some simple experiments to illustrate that prior assumptions about the data is important in the design of appropriate algorithms.

3.8.1 Simultaneous Parameter Estimation and Segmentation

In [64], a simpler model based on GRF is used to model the intensity process. This model can be summarized as:

$$Y_{ij} = X_{ij} + W_{ij} \tag{3.31}$$

where Y_{ij} is the observed intensity at location (i,j), X_{ij} is the true intensity and W_{ij} is an independent identically distributed zero mean Gaussian noise and it is assumed that its variance is known. Further, $X_{ij} \in \{s_1, \ldots, s_N\}$, s_i being the intensity of the i-th region, and N are assumed to be known. The process X is modeled as a MRF taking one of these N values. The joint distribution of X can be written as a Gibbs distribution and the particular form of this used in [64] is called a Multilevel Logistic (MLL) distribution. Hence the parameters correspond to this MLL distribution. A maximum likelihood estimate of the parameters of the MLL distribution are computed and combined with simulated annealing to obtain an optimum segmentation of the scene. A convergence result is also proved for this adaptive segmentation scheme.

In the analysis of the algorithm, the assumptions made play an important role. Even with these simple assumptions, due to computational difficulties further approximations have to be made. For example in the above scheme, a pseudo-likelihood algorithm is used to approximate the MLEs to avoid the computational burden involved in the estimation of MLE. The use of simulated annealing makes the algorithm computationally demanding. Further, if any of the assumptions made above (eg., known number of regions, their intensity values or known noise parameters) are relaxed [106], the resulting convergence may not be even to a local optimum. Thus, even though the principle of simultaneous parameter estimation and segmentation could be used in more general cases like the textured images considered in section 3.6, it is not clear if it has any advantages compared to the scheme detailed in section 3.7 where we first estimate the parameters from windows to obtain a coarse segmentation and then use pixel based schemes for finer segmentation.

A simple nearest-neighbor classification scheme

Adaptive segmentation should be data driven, but at the same time we should make use of whatever information that is available about the data in the design of such algorithms. To further illustrate the usefulness of the prior knowledge about data, we give below a simple classification scheme which makes the same assumptions as in the adaptive segmentation scheme of [64] and does not need expensive algorithms like simulated annealing to obtain comparable results. The data is the same as the one used in [64]. Also the information about the noise variance is not used in this segmentation operation. For obtaining the segmented image from a noisy version of it, we used the following ad-hoc scheme:

1. At each pixel site (i,j), compute the average μ_{ij} in a small window (of size 3x3 in our case) around the pixel (i,j)

2. Then the intensity of the pixel is estimated as:

$$\hat{x}_{ij} = s_k = \min_l |s_l - \mu_{ij}| \tag{3.32}$$

3. The resulting segmentation is processed through a smoothing filter (similar to the one described in section 3.7) to eliminate small isolated regions.

Figure 3.8 shows the performance of this scheme on a two region hand drawn image. Figure 3.8(a) is the original image with the two intensity levels being 100 and 150. This is one of the images used in [64]. Figure 3.8(b) is the noisy version with the noise being additive i.i.d. zero mean gaussian with standard deviation 25 (Signal to noise ratio (SNR) of 2). The classification result we obtained is shown in Figure 3.8(c) with the classification error of 1.73%. Figures 3.8(d) and 3.8(e) show the results when the noise deviation is 50 (SNR 1).

Corresponding results for the four region case (with intensity values 100, 150, 200 and 250) are shown in Figure 3.9. The maximum difference in the performance of the nearest neighbor classification rule to that reported in [64] is for the four region case with SNR 2, where we obtained an error of 2.21% compared to 0.4% reported in [64]. Table 3.3 compares the performance of this nearest neighbor classification scheme with the adaptive segmentation algorithm

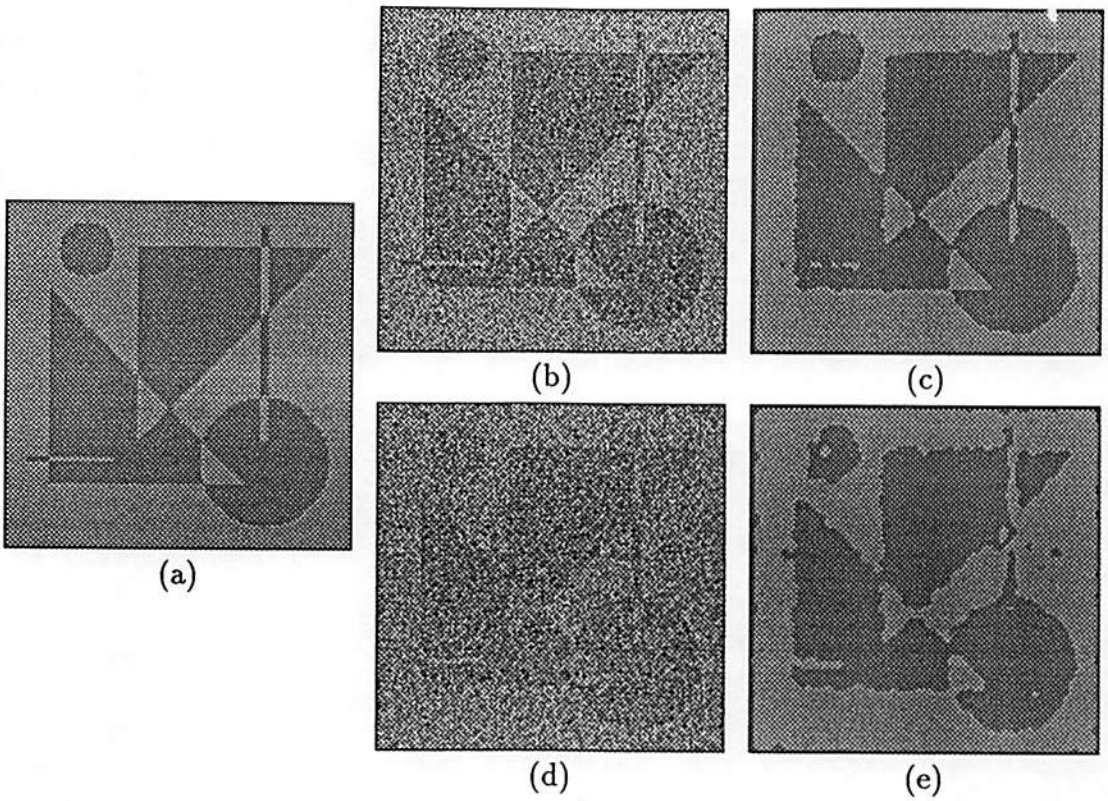


Figure 3.8: Segmentation of two region hand drawn image using a nearest neighbor classification rule, (a) original image, (b) with SNR 2 (standard deviation 25) (c) segmented image from (b), (d) with SNR 1 (standard deviation 50) (e) segmented image from (d).

of [64]. As far as the computation time required, this clustering technique takes few seconds of CPU time (for the 128x128 images, on a SUN-3 workstation) compared to 15-30 minutes (on VAX 8600) reported in [64].

3.8.2 Conclusions

As can be seen from these experiments, the complexity of the segmentation algorithms can be greatly reduced by a proper use of prior information about the assumed models. The texture model considered in section 3.2 is more complicated than the one discussed in this section and the only explicit assumption made was that the textures can be modeled by a second order GMRF. Depending on the textures, this may or may not be a valid assumption. However from our

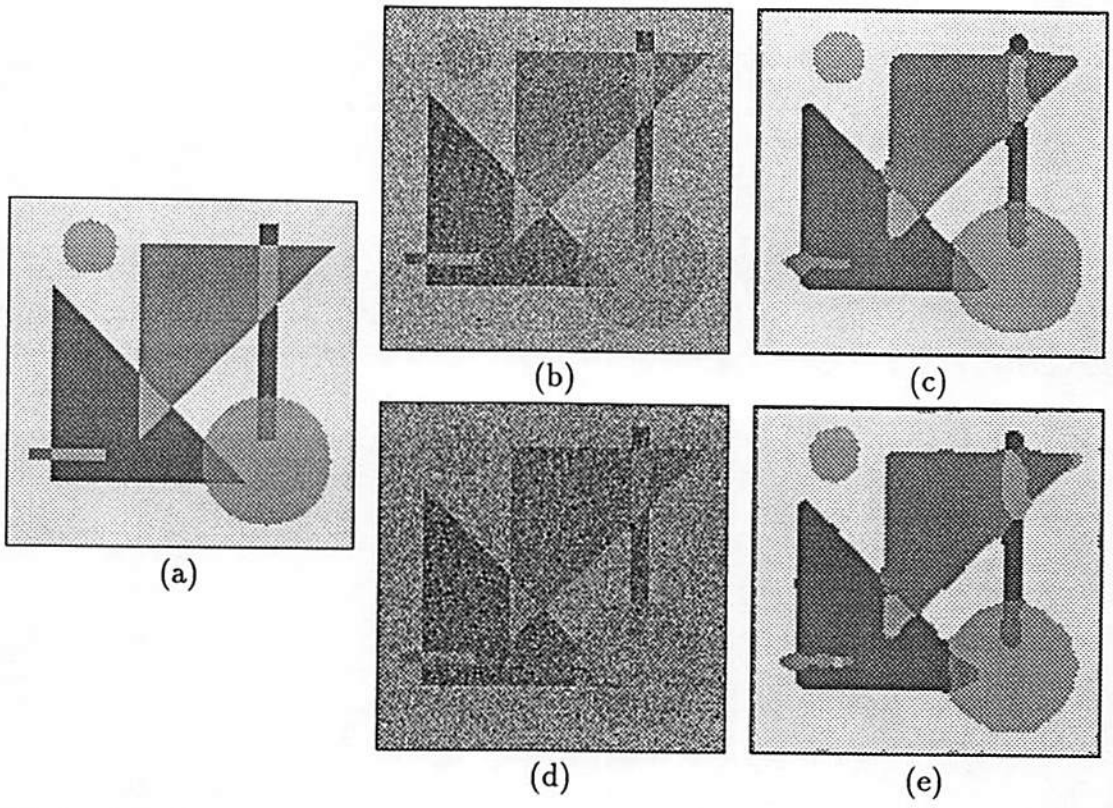


Figure 3.9: Segmentation of a four region hand drawn image using a nearest neighbor classification algorithm. (a) original image, same as the one used in [1]. (b) with SNR 2 (standard deviation 25) (c) segmented image from (b), (d) with SNR 1 (standard deviation 50) (e) segmented image from (d).

experience with different real textures like wood, wool, water etc., this appears to be a good approximation and our experimental results also support this fact. From the computational view-point it is better to separate estimation and segmentation stages. As mentioned earlier however, by doing this the algorithm will not lend itself to easy theoretical analysis.

Algorithm	2 Regions		4 Regions	
	SNR 2	SNR 1	SNR 2	SNR 1
Nearest-Neighbor classification	1.73	4.60	2.21	3.19
Adaptive Segmentation (from [1])	0.96	3.88	0.40	1.98

Table 3.3: Comparison of the adaptive segmentation algorithm in [1] with the nearest neighbor classification scheme. The numbers indicate percentage classification errors. SNR 2 corresponds to a noise standard deviation of 25 and SNR 1 corresponds to a deviation of 50.

Chapter 4

Boundary Detection

In the previous chapter we considered model based texture segmentation. Here we extend the discussion to a more general problem of boundary detection, including texture boundaries as well as intensity discontinuities. Regarding texture boundary detection, the approach taken here differs from that of the previous chapter in that no models are assumed. The development here is similar to modeling human pre-attentive texture discrimination, which is one of the topics of much current interest. Pre-attentive texture perception refers to the ability of the human visual system to discriminate patterns without any conscious effort.

The development of our system is biologically motivated as we model some of the early processing stages in the visual cortex using multiscale orientation sensitive filters and local nonlinear interactions. An interesting feature of this system is the end-inhibition property, referring to the property of feature detectors which are sensitive to small lines and edges in the image. End-inhibition plays a very important role in early processing tasks such as localizing texture boundaries, and in curvature detection. This property is further exploited in our face recognition system discussed in the next chapter. We also discuss their role in perceiving illusory contours.

4.1 Introduction

We suggest a simple biologically motivated approach to detecting image boundaries. Biological vision systems, especially those of mammals and in particular human's, are extremely adept at processing the vast amount of intensity data projecting from the three dimensional external world on to the two dimensional retina. Recent research in psychophysics and neurophysiology has begun to shed light on some of the basic mechanisms that are used in interpreting this information. The initial stages of this visual processing are very important in this respect as they detect and group various types of salient features, and transform the intensity information to a more suitable representation convenient for further processing. These stages are responsible for preliminary processing of stereo, texture and motion, which further aid in performing one of the fundamental tasks in image understanding, namely boundary perception and scene segmentation.

In the 3-D world the objects are separated from the background (as well as other objects) by depth discontinuities, which usually manifest as intensity discontinuities in 2-D images. Intensity changes also result from occlusion of objects, sharp changes in surface orientation, changes in reflectance properties or illumination. As these intensity changes are a rich source of information, detecting them is an important problem both in computer vision as well as in human vision. Among the most commonly used edge detection algorithms are the zero crossings of the Laplacian of the Gaussian [74] and Canny's edge detector [14]. Textures form another important class of natural scenes and like intensity edges provide useful information regarding shape and motion. We have discussed a model based approach for detecting texture boundaries in Chapter 3. Computational models for human texture perception have also been extensively studied [3, 4, 53, 67]. Though intensity edges and textures are fundamental to image understanding, only recently some work has been done in integrating the detection of these features. In [30], a composite model is proposed for detecting both the intensity as well as texture edges. A random field model is proposed for a general boundary detection scheme in [38], where the problem of segmentation is formulated as an optimization process and relaxation algorithms are used to obtain the segmentation.

In addition to intensity edges and textures, human vision system can perceive object boundaries where none physically exist, giving rise to what are generally referred to as illusory contours. This perception is a consequence of the mechanisms involved in interpreting incomplete information such as those due to occlusion, which is very common in the real world. The mechanisms themselves are not well understood and surprisingly not much attention has been given to this problem in computer vision research. The problem of understanding the perception of such contours is complicated because it is difficult to separate the role of high level (or contextual) knowledge from the low level mechanisms which actually complete the boundary. Our discussion in this chapter regarding such contours, hence, is limited to very simple examples such as the ones induced by line terminations (see Figure 4.10).

A schematic diagram of our model is shown in Figure 4.1. The input image is first processed through a bank of orientation selective bandpass filters at various spatial frequencies. Our choice of Gabor functions to model these filters has been mainly due to mathematical convenience and their important theoretical properties concerning localization in space and frequency. Gabor functions are modulated Gaussians having an even symmetric real part and an odd symmetric imaginary part. They have been used in many vision applications such as optical flow computations [44], image coding [26, 86], pattern recognition [10], and texture analysis [9]. The convolution of the image with these filters yields a representation which is localized in space as well as in frequency. The filter parameters determine the exact nature of this representation. A special class of this decomposition is the wavelet transformation where the filter profiles are all self-similar. Wavelets are families of basis functions obtained through dilations and translations of a *basic wavelet* and such a decomposition provides a compact data structure for representing information. In our case the basic wavelet is a Gabor function and we refer to this decomposition as the Gabor wavelet transformation (in [86] the term Gabor pyramid is used instead). A Gabor wavelet decomposition can be interpreted as extracting salient features in the image at different scales and orientations and the local maxima in their *energy* (see section 4.4.1) correspond to the intensity edges in the image.

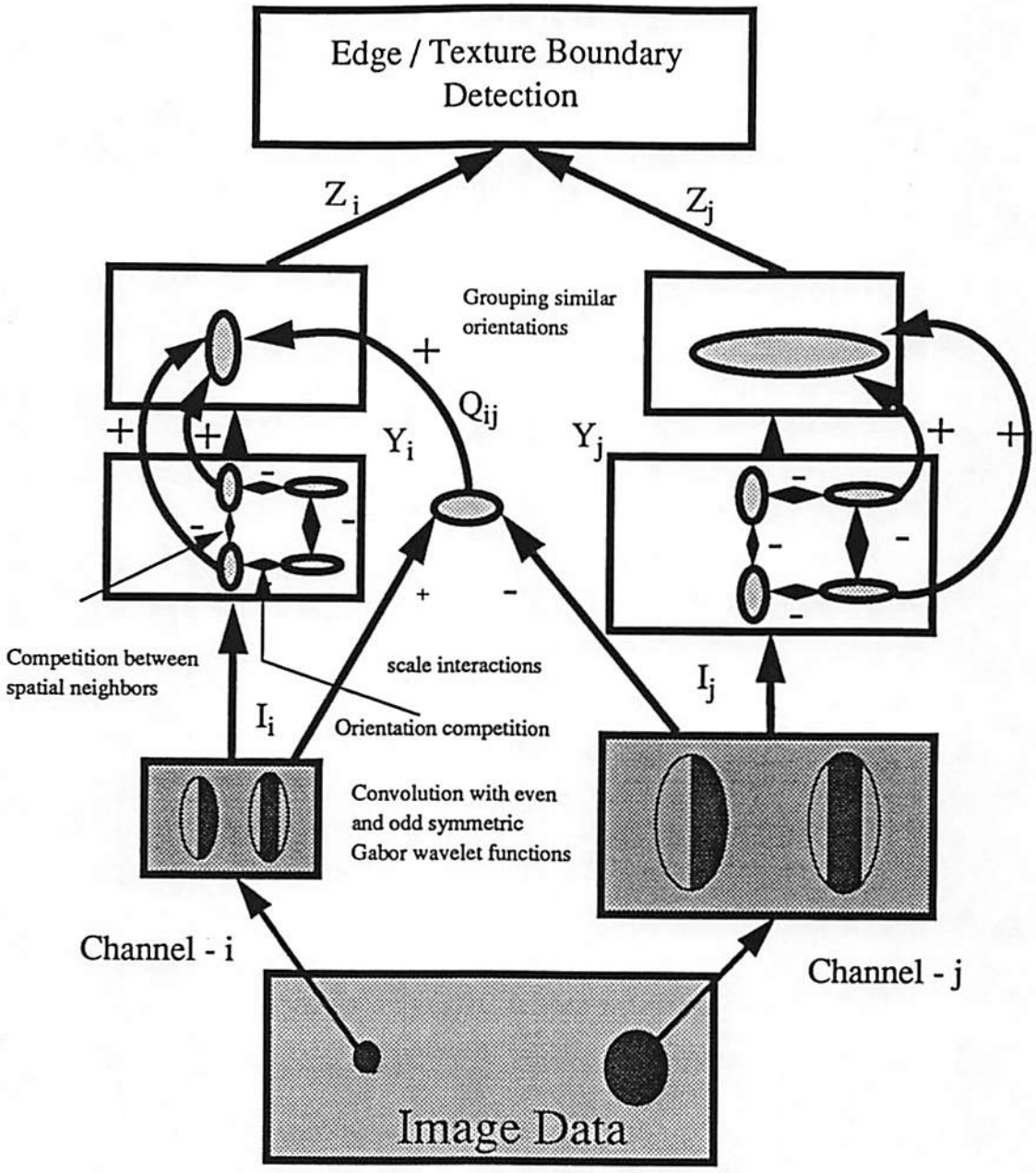


Figure 4.1: Schematic diagram of the model. The input image is first processed through a wavelet transform based on Gabor functions. In the next stage local competitive interactions are introduced in each of the frequency channels. Inter-scale interactions help in localizing line ends. In the final stage outputs from like oriented cells are grouped to complete boundaries. Edges are located at the local maxima in Z and texture boundaries correspond to local maxima in the gradient field of Z.

Following the wavelet decomposition we introduce local feature interactions. Three distinct types of interactions are considered: competition between spatial neighbors in each orientation channel, competition between orientations at each spatial location, and interscale interactions. These interactions are shown in Figure 4.1. One extreme form of this type of interactions is the *winner-take-all* case where the dominant feature suppresses all the others, and this has been used in [42, 67]. Interscale interactions are used in localizing line ends and play an important role in boundary detection. The second stage of interactions groups similar features in the neighborhood. This cooperative processing helps in the boundary completion process. The receptive fields of the cells in this stage have the same orientation selectivity as their inputs and have a larger receptive field and the filter profiles are modeled by oriented Gaussians. From a neurophysiological perspective the Gabor wavelet decomposition can thus be identified with processing by simple cells and local interactions with those of complex and hypercomplex cells.

The final step in the model involves identifying the boundaries in the image. Let the output after the grouping stage be denoted by Z_i, where i corresponds to the ith frequency channel. Features such as intensity discontinuities and illusory contours can now be located at the local maxima in Z_i and textural boundaries correspond to the local maxima in the gradient of Z_i. Experimental results on several images are presented to illustrate the performance of this model in detecting these features.

The organization of this chapter is as follows: Section 4.2 discusses some related work on edge detection, pre-attentive segmentation and illusory contour perception. Section 4.3 gives a brief introduction to wavelets and Gabor functions. Section 4.4 describes the different stages in our model which includes Gabor wavelet transformation to extract features and local feature interactions for segmentation and grouping. A brief analysis of the Gabor wavelets in edge detection is also given. Experimental results in detecting edges and texture boundaries in a variety of images are provided in section 4.5.

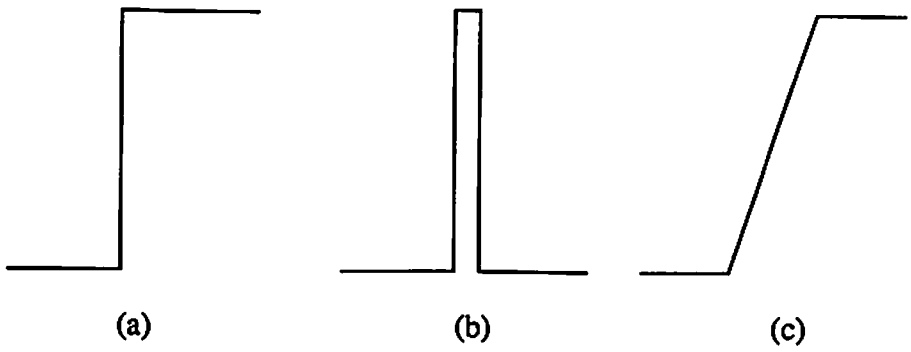


Figure 4.2: (a) Step edge (b) bar or line edge (c) ramp

4.2 Review of Previous Work

4.2.1 Energy Features and Edge Detection

Almost all techniques for edge detection in computer vision literature have been developed for detecting step edges. Consequently their performance is poor on other edges such as lines (or bars) and ramps (Figure 4.2). As mentioned earlier, the popular techniques include locating edges at the zero crossings of the Laplacian of the Gaussian convolved with the image [74], or at the local maxima of the outputs of convolution with directionally selective odd-symmetric filters [14]. The limitations of these methods are well known [76, 79, 91]. Some important observations are that feature detection/localization by any type of linear filtering operation is not adequate, and in particular zero crossings of the result of applying any linear operator to the image do not capture all significant features in the image. Secondly, there is a need for directionally selective quadrature filter pairs. The outputs of these filters cannot be analyzed separately (except when either the step edges alone or the line edges alone are present). In general no linear filtering operation will be able to detect and localize composite edges accurately

[79, 91]. As an alternative, Morrone and Burr [79] suggested the use of energy measures in edge detection. They show that locations in an image where the Fourier components have zero phase difference constitute perceptually significant features (such as the different types of edges mentioned above), and these could be detected at the local maxima in an appropriate energy measure. This energy model for edge detection is also used to explain perceptions of Mach bands[1] and several other visual illusions [11]. Perona and Malik [83] have provided analysis on the performance of energy features in composite edge detection. Their filters are derived from Gaussian function and its derivatives. They address such issues as signal to noise ratio and localization error, similar to the ones used by Canny [14]. Note that energy is one of the measures that can be used in combining the outputs of the even and odd symmetric filters and a discussion of other means of combining the information and their relative advantages can be found in [91].

4.2.2 Pre-attentive Segmentation

Pre-attentive segmentation refers to the ability of humans to perceive textures without any sustained attention. Among the basic primitives that strongly influence the perception of textural scenes are color, brightness, orientation and size [3]. In addition, distribution of features also play an important role. For example, Figure 4.9 (a) show randomly oriented L and + texture. The two regions are easily discriminable, although there is no statistical difference in either orientation or scale of the micropatterns. What they differ is in the distribution of higher order features such as corners and intersections. Central to solving this problem are the issues of what features need to be computed and what kind of processing of these features is required for texture discrimination.

Some of the early work in this field can be attributed to Julesz [52] for his theory of textons as basic textural elements. Spatial filtering approach has been used by many researchers for detecting texture boundaries not clearly explained by the texton theory [4]. Recently an elegant computational model for preattentive texture discrimination is proposed by Malik and Perona [67]. Their three

[1]Mach bands are bright and dark lines that are perceived close to the transition regions of a blurred edge.

[79, 91]. As an alternative, Morrone and Burr [79] suggested the use of energy measures in edge detection. They show that locations in an image where the Fourier components have zero phase difference constitute perceptually significant features (such as the different types of edges mentioned above), and these could be detected at the local maxima in an appropriate energy measure. This energy model for edge detection is also used to explain perceptions of Mach bands[1] and several other visual illusions [11]. Perona and Malik [83] have provided analysis on the performance of energy features in composite edge detection. Their filters are derived from Gaussian function and its derivatives. They address such issues as signal to noise ratio and localization error, similar to the ones used by Canny [14]. Note that energy is one of the measures that can be used in combining the outputs of the even and odd symmetric filters and a discussion of other means of combining the information and their relative advantages can be found in [91].

4.2.2 Pre-attentive Segmentation

Pre-attentive segmentation refers to the ability of humans to perceive textures without any sustained attention. Among the basic primitives that strongly influence the perception of textural scenes are color, brightness, orientation and size [3]. In addition, distribution of features also play an important role. For example, Figure 4.9 (a) show randomly oriented L and + texture. The two regions are easily discriminable, although there is no statistical difference in either orientation or scale of the micropatterns. What they differ is in the distribution of higher order features such as corners and intersections. Central to solving this problem are the issues of what features need to be computed and what kind of processing of these features is required for texture discrimination.

Some of the early work in this field can be attributed to Julesz [52] for his theory of textons as basic textural elements. Spatial filtering approach has been used by many researchers for detecting texture boundaries not clearly explained by the texton theory [4]. Recently an elegant computational model for preattentive texture discrimination is proposed by Malik and Perona [67]. Their three

[1]Mach bands are bright and dark lines that are perceived close to the transition regions of a blurred edge.

stage model involves convolutions with even symmetric filters followed by half wave rectification, local inhibition, and texture boundary detection using odd symmetric filters. They present very convincing arguments regarding the necessity for each of these stages, and provide quantitative measures regarding the performance of their algorithm to establish consistency between their results and human perception of textures. Although impressive, their approach still does not clearly suggest *why* their model is able to detect certain textures such as those in Figure 4.9 (a). What their model suggests is that, by using a large number of filters followed by non linear transformations, one can detect these texture boundaries.

4.3 Multiscale Representation and Wavelets

The multiscale approach provides an elegant hierarchical framework for image analysis. The features of interest in an image are generally present in various sizes. An efficient way to analyze such features is to have a multiscale decomposition of the image. Laplacian pyramid [13] is one of the early schemes developed for such applications, though initially it was proposed for compact image coding applications. Multiscale representation also helps in parallel processing as the different channels can now be analyzed independently (at least initially). Multiscale approach has also been used in robust detection of step edges [74] and Witkin [105] describes how information at different levels can be related. The presence of parallel visual pathways consisting of cells with varying receptive field sizes and orientations are indicative of a multiscale feature extraction in biological systems as well. However, the role of interactions that exist between different scales in these systems is not well understood. In section 4.4.3 we suggest a simple model which uses such interactions in detecting line ends and corners, and its possible biological significance.

There has been a growing interest in the use of wavelets for multiscale representation of the image data [68]. Wavelets are families of basis functions generated by dilations and translations of a *basic wavelet*. The wavelet transform is thus

a decomposition of the function (image intensity) in terms of these basis functions. One of the objectives of such a transformation is to provide a simultaneous description of the data in frequency and spatial domains.

Let us first consider the one dimensional case. Let $g(x)$ be a wavelet, $x \in \mathbf{R}$ ($\mathbf{R}$ denotes the set of real numbers). Then the the family of basis functions corresponding to $g(x)$ can be generated by translations ($g(x-s)$) and dilations ($g(\alpha x)$), where s and α are the translation and scale parameters respectively. Let this family be denoted by $(g(\alpha(x-s)))$, $(\alpha, s) \in \mathbf{R}^2$. The wavelet transform of a function $f(x)$ (assuming that $f(x)$ is square integrable) is defined by

$$W_f(\alpha, s) = \int_{-\infty}^{\infty} f(x)\, g^*(\alpha(x-s))\, dx \tag{4.1}$$

The (*) indicates complex conjugate. Wavelets can be discretized by suitable sampling of the parameters α and s. For example we can write the scale parameter as α^j where $j \in \mathbf{Z}$, $\mathbf{Z}$ being the set of integers. This results in a class of discrete wavelets represented by $g(\alpha^j x - n), (j, n) \in \mathbf{Z}^2$. A function $f(x)$ can then be expanded in terms of the basis functions $g(\cdot)$ as

$$f(x) = \sum_{i,j} c_{ij} g(\alpha^j x - i)$$

The Laplacian pyramid [13] mentioned earlier is a wavelet decomposition based on Difference of Gaussian (DOG) wavelet and has found many applications in image processing [12]. Orthogonal wavelets are a special family of discrete wavelets corresponding to $\alpha = 2$, where the basis functions are mutually orthogonal, i.e., $\int g(x)\, g(2^j x - k)dx = 0$ for $((j,k) \in \mathbf{Z}^2)$. A discussion on orthogonal wavelets and their applications to image processing can be found in [68]. An important feature of orthogonal wavelets is that the information at different resolutions is uncorrelated. Orthogonality, in general, is a strong condition, and is difficult to achieve if arbitrary orientation selectivity is desired. Further, it is harder to give a frequency domain interpretation of the features so extracted by the decomposition. In the following we consider a transformation based on non-orthogonal Gabor basis functions and discuss its usefulness for image processing applications.

4.3.1 Gabor Functions and Wavelets

Gabor functions are Gaussians modulated by complex sinusoids. In its general form, the 2-D Gabor function and its Fourier transform can be written as [25],

$$g(x, y; u_0, v_0) = exp(-[x^2/2\sigma_x^2 + y^2/2\sigma_y^2] + 2\pi i[u_0\, x + v_0\, y]) \qquad (4.2)$$

$$G(u, v) = exp(-2\pi^2(\sigma_x^2(u - u_0)^2 + \sigma_y^2(v - v_0)^2)) \qquad (4.3)$$

σ_x and σ_y define the widths of the Gaussian in the spatial domain and (u_0, v_0) is the frequency of the complex sinusoid. A well known property of these functions is that they achieve the minimum possible joint resolution in space and frequency domains [25]. A signal such as a delta function which is concentrated at a point in space has no frequency localization. Likewise, a function concentrated in frequency has no spatial localization. A good measure of localization in the two domains is the product of the bandwidths in space and frequency. The effective bandwidth of a signal is defined as the square root of the variance of the energy of the signal. Let δx and δy be the effective widths of the signal in the horizontal and vertical directions in space respectively and δu, δv denote the corresponding widths in frequency. Then the following inequalities (also called the uncertainty relations) hold: (a) $\delta x\ \delta u \geq 1/4\pi$ and (b) $\delta y\ \delta v \geq 1/4\pi$. Gabor family of functions are unique in attaining the minimum possible value of this joint uncertainty. This localization property has received considerable attention among vision researchers and has led to many applications [9, 10, 26, 86].

The Gabor functions form a complete but non-orthogonal basis set and any given function $f(x, y)$ can be expanded in terms of these basis functions. Such an expansion provides a localized frequency description and has been used in image compression [26] and texture analysis [9]. Local frequency analysis, however, is not suitable for feature representation as it requires a fixed window width in space and consequently the frequency bandwidth is constant on a linear scale. However, in order to optimally detect and localize features at various scales, filters with varying support rather than a fixed one are required. This would suggest a transformation similar to wavelet decomposition rather than a local Fourier transform. We now consider such a wavelet transform where the *basic*

wavelet is a Gabor function of the form

$$g_\lambda(x, y, \theta) = e^{-(\lambda^2 x'^2 + y'^2) + i\pi x'} \tag{4.4}$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

where λ is the spatial aspect ratio, θ is the preferred orientation. To simplify the notation, we drop the subscript λ and unless otherwise stated assume that $\lambda = 1$. For practical applications, discretization of the parameters is necessary. The discretized parameters must cover the entire frequency spectrum of interest. Let the orientation range $[0, \pi]$ be discretized into N intervals and the scale parameter α be sampled exponentially as α^j. This results in the wavelet family

$$(g(\alpha^j(x - x_0, y - y_0), \theta_k)), \alpha \in \mathbf{R},\ j = \{0, -1, -2, ..\} \tag{4.5}$$

where $\theta_k = k\pi/N$. The Gabor wavelet transform is then defined by

$$W_j(x, y, \theta) = \int f(x_1, y_1)\ g^*(\alpha^j(x_1 - x, y_1 - y), \theta)\ dx_1 dy_1 \tag{4.6}$$

At each resolution in the representation hierarchy these wavelets localize the information content in both frequency and spatial domains simultaneously. Further any desired orientation selectivity can be obtained by controlling the parameter θ. The Gabor wavelet decomposition also has an important physical interpretation of the type of features detected and this is further discussed in section 4.4.1 and has been used in applications such as image coding [26, 86] and pattern recognition [10]. Section 4.4.1 discusses in detail the usefulness of this representation in detecting perceptually significant features in the image.

4.4 Stages in Boundary Detection

We now discuss the various processing stages in our model shown in Figure 4.1. We begin with a brief analysis of the Gabor wavelets as edge detectors.

4.4.1 Line and Edge Detectors

The wavelet decomposition using Gabor functions has an important physical interpretation. The complex Gabor function has an even-symmetric (cosine) real part and an odd-symmetric (sine) imaginary part, which respond maximally to *line edges (or bars)* and *step* edges (of appropriate sizes and orientations), respectively, in the image. This wavelet decomposition can be viewed as obtaining a primal sketch of the raw intensity data by detecting perceptually significant features at different scales. These features can be detected at the local maxima in their *energy* [79]. If R_i and I_i represent the response from the even and odd symmetric feature detectors at a position i, then the local energy E_i at i is given by $E_i = \sqrt{R_i^2 + I_i^2}$.

We now provide a brief analysis of the performance of such energy detectors based on Gabor wavelets.

Performance Analysis

In the following we assume 1-D functions for simplicity and give an analysis of the signal to noise ratio (SNR) and localization properties analogous to the one given in [14]. A similar analysis for energy feature detectors based on derivatives of the Gaussians can be found in [83]. The SNR is defined as the ratio of the signal power output to the noise power at the true location of the edge. The localization gives a measure of the performance of the detector in accurately localizing the edge in the presence of noise and is defined as the inverse of the square root of the variance in this deviation. In [14], the product of SNR and localization is maximized in deriving an appropriate filter for detecting step edges. Notice that in general these two criteria contradict, as better localization implies poorer SNR.

Here we derive the SNR and the localization error for the two cases (step and line edges) in 1-D. Let the true location of the edge $e(x)$ be at the origin $x = 0$ and due to the presence of noise $\eta(x)$ the observed maximum in the energy $E(x)$ is at x_0. Let $y(x) = f(x) + n(x)$ be the response of the complex filter $g(x)$ due

to the noisy input $e(x) + \eta(x)$, where

$$g(x) = g_r(x) + \imath g_i(x) = \exp(-\alpha^2 x^2/2 + \imath \pi \alpha x)$$

and $f(x)$ and $n(x)$ denote the signal and noise terms respectively at the output. The output noise energy for a white noise input is

$$\int_{-\infty}^{\infty} g(x)g^*(x)dx = \int_{-\infty}^{\infty} exp(-\alpha^2 x^2)dx = \sqrt{\pi}/\alpha \tag{4.7}$$

If the edge feature $e(x)$ is centered at the origin, the SNR at the true location of the edge is given by the ratio of the signal power to the noise power at the origin:

$$SNR = \frac{|f(0)|}{\sqrt{\sqrt{\pi}/\alpha}} \tag{4.8}$$

The edges are located at the local maxima of the energy $E(x) = y(x)y^*(x)$ and $E'(x_0) = 0$. Further, assuming that the noise power is small compared to the signal power, we can approximate the energy by neglecting the terms containing $n^2(x)$, $E(x) \approx ff^*(x) + 2(f_r(x)n_r(x) + f_i(x)n_i(x))$, where $f_r(x) = g_r * e(x)$, $f_i(x) = g_i * e(x)$. Similarly n_r and n_i denote the real and complex parts of the output noise signal. Now,

$$E'(x_0) \approx (ff^*)'(x_0) + 2(f_r n_r + f_i n_i)'(x_0) = 0 \tag{4.9}$$

Expanding the first term in a Taylor's series around the origin,

$$(ff^*)'(x_0) = (ff^*)'(0) + x_0(ff^*)''(0) + \mathcal{O}(x_0^2) \tag{4.10}$$

Noting that $(ff^*)'(0) = 0$ and substituting (4.10) in (4.9) and ignoring higher order terms, we get

$$x_0(ff^*)''(0) + 2(f_r n_r + f_i n_i)'(x_0) = 0 \tag{4.11}$$

As in [14], we use the inverse of the variance of x_0, $(\mathcal{E}(x_0^2))^{-1}$ as a measure of

localization, and from (4.11) we get,

$$\mathrm{L} = [\mathcal{E}(x_0^2)]^{-\frac{1}{2}} \approx \frac{|[(ff^*)''(0)]|}{2\sqrt{\mathcal{E}[(f_r n_r + f_i n_i)'(x_0)]^2}} \tag{4.12}$$

We now evaluate the SNR and localization for the cases of line and step edges. For a line edge, $e(x) = \delta(x)$

$$\mathrm{SNR(line)} = |f(0)|/\sqrt{\sqrt{\pi}/\alpha} = 0.7551\sqrt{\alpha} \tag{4.13}$$

$$\mathcal{E}[(f_r n_r + f_i n_i)'(x_0)]^2 \approx \mathcal{E}[(f_r n_r)'(x_0)]^2 \approx f_r^2(x_0)\mathcal{E}[(n_r)'(x_0)]^2$$

Further, it can be shown that

$$\mathcal{E}[(n_r)'(x_0)]^2 = \mathcal{E}[(n_i)'(x_0)]^2 \approx 9.19\alpha$$

$$f_r(x_0) = (g_r * \delta)(x_0) = exp(-\alpha^2 x_0^2/2)\cos(\pi\alpha x_0)$$

To evaluate the numerator in (4.12),

$$(ff^*)'' = f''f^* + f(f^*)'' + 2f'(f^*)' \tag{4.14}$$

and for the line edge, $f(x) = \delta(x) * g(x) = g(x)$, $f'(0) = g'(0) = \imath\pi\alpha$ and $f''(0) = g''(0) = -\alpha^2(1+\pi^2)$. Substituting these values in (4.14), $(ff^*)''(0) = 2\alpha^2$, and finally from (4.12)

$$\mathrm{L(line)} = \frac{2\alpha^2}{2exp(-\alpha^2 x_0^2/2)\cos(\pi\alpha x_0)\sqrt{9.19\alpha}} \approx 0.33\alpha\sqrt{\alpha} \tag{4.15}$$

For the step edge we have $e(x) = \int_{-\infty}^{x}\delta(x')dx'$, $f(x) = \int_{-\infty}^{x} g(x')dx'$, $|f(0)| = |\int_{-\infty}^{0} g(x')dx'| \approx 0.3694/\alpha$, and

$$\mathrm{SNR(step)} = |f(0)|/\sqrt{\sqrt{\pi}/\alpha} \approx 0.2774/\sqrt{\alpha} \tag{4.16}$$

Further we have $f'(x) = g(x)$, $f''(x) = g'(x)$ and from (4.14)

$$(ff^*)''(0) = g'(0)\int_{-\infty}^{0} g^*(x)dx + (g^*)'(0)\int_{-\infty}^{0} g(x)dx + 2g(0)g^*(0) = 4.32$$

and

$$\mathcal{E}[(f_r n_r + f_i n_i)'(x_0)]^2 \approx \mathcal{E}[(f_i n_i)'(x_0)]^2 \approx f_i^2(x_0)\mathcal{E}[(n_i)'(x_0)]^2 \approx 9.19\alpha(0.3693/\alpha)^2$$

Substituting all these values in (4.12)

$$\text{L(step)} = 1.93\sqrt{\alpha} \tag{4.17}$$

In general (except for the special case of line edge), the SNR improves with increasing filter width whereas localization deteriorates. The SNR (for step edges) is poor compared to the first derivative of Gaussian used in [14], but its overall performance in the presence of composite edges is better. Note that both the Marr-Hildreth [74] and Canny [14] operators will fail to detect line edges at their true locations. In fact it is easy to see that at the true locations of line edges the SNR is zero for these operators.

4.4.2 Local Spatial Interactions

Following feature extraction using Gabor wavelets, we now consider local competitive and cooperative processing of these features. Competitive interactions help in noise suppression, and in reducing the effects of illumination.

These interactions are modeled by non-linear lateral inhibition between features. Two types of such interactions are distinguished. The first type includes competition between different orientations at each spatial position and the second between spatial neighbors within each orientation and scale. Figure 4.1 shows the various interactions in two frequency channels in the system. For simplicity the transfer function $g(x)$ of all feature detectors is assumed to be the same. The following notation is used in explaining the interactions: The output of a cell at position $s = (x, y)$ in the ith spatial frequency channel with a preferred orientation θ is denoted by $Y_i(s, \theta)$, with $I_i(s, \theta)$ being the excitatory input to that

cell from the previous processing stage. For example $I_i(s,\theta)$ could be the energy in the filter output corresponding to feature (s,θ) in the ith frequency channel. For convenience we will drop the subscript i indicating the frequency channel whenever there is no ambiguity. Let N_s be the local spatial neighborhood of s. The competitive dynamics is represented by

$$\dot{X}(s,\theta) = -a_{s,\theta}X(s,\theta) + I(s,\theta) - \sum_{s' \in N_s} b_{s,s'}Y(s',\theta) - \sum_{\theta' \neq \theta} c_{\theta,\theta'}Y(s,\theta') \quad (4.18)$$

$$Y(s,\theta) = g(X(s,\theta)) \quad (4.19)$$

and (a,b,c) are positive constants. In our experiments we have used a sigmoid non-linearity of the form $g(x) = 1/(1+exp(-\beta x))$. The dynamics of (4.18) can be visualized as follows : At each location within a single frequency channel, the corresponding cell receives an excitatory input from a similarly oriented feature detector (of the same spatial frequency). Further it also receives inhibitory signals from the neighboring cells within the same channel. We assume that all these interactions are symmetric ($b_{s,s'} = b_{s',s}$ and $c_{\theta,\theta'} = c_{\theta',\theta}$). The competitive dynamics of the above system can be shown to be stable. The Lyapunov function for the system [22, 47] can be written as

$$\begin{aligned} E(Y) \;=\; & \frac{1}{2}\sum_{s,s'} b_{s,s'}Y(s,\theta)Y(s',\theta) + \frac{1}{2}\sum_{\theta,\theta'} c_{\theta,\theta'}Y(s,\theta)Y(s,\theta') \\ & + \sum_{s,\theta} \int_0^{Y(s,\theta)} \left(a_{s,\theta}g^{-1}(y) - I(s,\theta)\right) dy \end{aligned} \quad (4.20)$$

Under the assumptions that the interactive synapses are symmetric and that $g(\cdot)$ is monotone non-decreasing, the time derivative of E is negative and the system represented by (4.18) always converges.

The specific form of the dynamics such as the one in (4.18) is not very critical, as long as there is some form of local inhibition to suppress weak responses. For example, one can use the inhibition scheme proposed by Malik and Perona in [67], or the one suggested in Grossberg and Mingolla's BCS [42]. In [42] the orientation competition is separated from the local spatial competition between

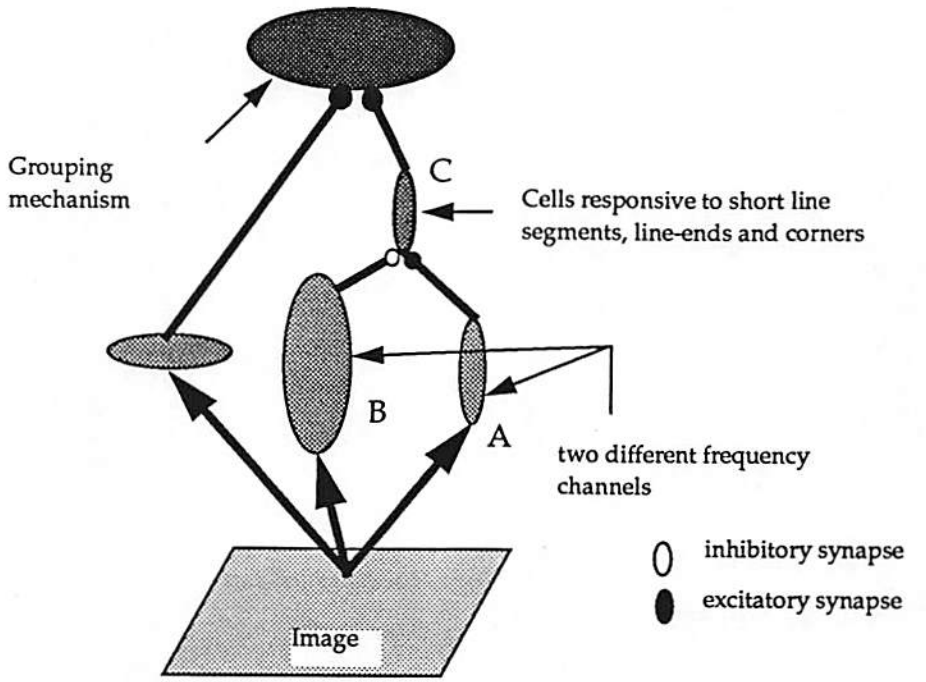


Figure 4.3: Interscale interactions: Cells with larger receptive profiles (B) inhibit those with shorter receptive fields (C), which also receive excitatory inputs from similar sized cells (A). Due to these interactions cell C exhibits end-inhibition, and in turn cooperates with orthogonal orientations in grouping the edges.

neighbors. An advantage of our method as well as that of BCS is that since the dynamics is expressed in terms of differential equations, it is amenable for analog implementations. The main difference between our model and that of BCS is in the generation of end-inhibition, discussed below.

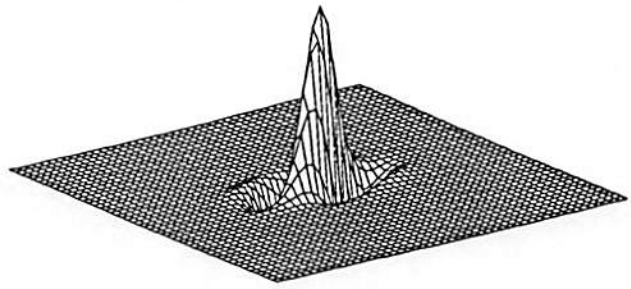
4.4.3 Local Scale Interactions

We now suggest a simple mechanism to model the behavior of hypercomplex cells. The hypercomplex cell receptive field must have inhibitory end zones along the preferred orientation. Such a profile can be generated either by modifying the profile of the simple cell itself or through interscale interactions, discussed

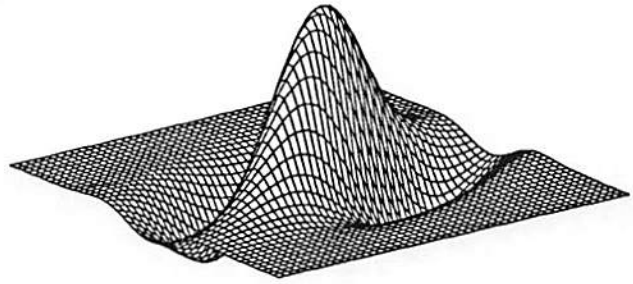
below. The fact that both simple and complex cells often exhibit this end-stopping behavior further suggests that both these mechanisms are utilized in the visual cortex. A schematic diagram of the model which utilizes interscale interactions is shown in Figure 4.3. If $Q_{ij}(x, y, \theta)$ denotes the output of the cell C at position (x, y) receiving inputs from two frequency channels i and j $(\alpha^i < \alpha^j)$ with preferred orientation θ, then

$$Q_{ij}(x, y, \theta) = g(||W_i(x, y, \theta) - \gamma W_j(x, y, \theta)||) \tag{4.21}$$

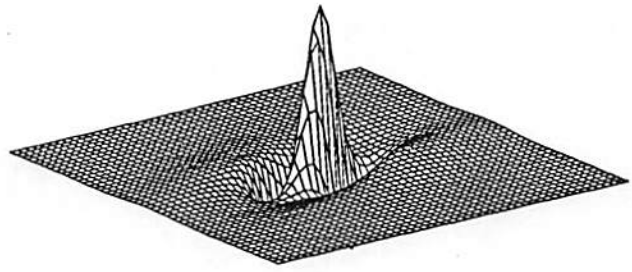
where $\gamma = \alpha^{-2(i-j)}$ is the normalizing factor. Figure 4.4 shows a typical receptive field profile of such an end-inhibited cell, corresponding to an even symmetric receptive field, and without taking into account the sigmoid non-linearity $g(\cdot)$. The parameter values used are $\alpha = \sqrt{2}, i = -2, j = -5$. In Figure 4.3, unit C represents the hypercomplex cell, which receives an excitatory input from unit A and an inhibitory input from unit B. All three units A, B and C have the same orientation preference and unit B has a larger receptive field profile compared to A. Unit C thus responds to only line ends and short line segments and its response decreases as the output of B increases for larger line segments. The logic behind this is simple. At line ends, cells with shorter receptive fields will have a stronger response than those with larger fields, and consequently will be able to excite the hypercomplex cells. At other points along the line, both small and large receptive field cells are equally excited and in the process the response of the hypercomplex cells is inhibited. It appears that such scale interactions to generate end inhibition do exist in the visual cortex. Bolz and Gilbert [8] observe that connections between layers 6 and 4 in the cat striate cortex play a role in generating end inhibition. The cells in layer 4 are of hypercomplex type exhibiting end inhibition. Layer 6 cells have large receptive fields and require long bars (or lines) to activate them. In addition, cells in both layers showed orientation selectivity. Inactivating layer 6 cells resulted in the loss of end-inhibition property of layer 4 cells, while preserving other properties such as orientation selectivity. Thus, in the absence of layer 6 activity, cells in layer 4 could be excited by short bars and their response did not decrease as the bar lengths increased, suggesting that layer 6 cells have inhibitory effect on the cells of layer 4.



(a)



(b)



(c)

Figure 4.4: Even symmetric cell's receptive fields (a) for scale $\alpha^i = 1/2$, (b) for $\alpha^i = 1/2\sqrt{2}$, and (c) Profile generated due to interactions between the above two fields.

The model suggested in Figure 4.3 is one of the ways of generating end-inhibition (and probably the most simple one) through scale interactions. The original idea of using such interactions dates back to the early work of Hubel and Wiesel [49]. A similar model is also suggested in [28], where the role of end-inhibition in detecting curvature is also discussed. It has generally been suggested that hypercomplex cells help in localizing texture boundaries. We provide here for the first time a demonstration of their usefulness in detecting texture boundaries (see example 4 in section 4.5). von der Heydt and Peterhans [85, 96] were the first to clearly demonstrate that these cells play a major role in the perception of illusory contours, and some of their observations are used in our model for boundary detection. Before getting into the details of their role in the perception of illusory contours, it would be interesting to consider again the type of features that these cells represent. The inhibitory end-zones of these cells helps in making them respond to local curvature changes in the input intensity data (see also [28]). That these cells respond to line-ends is nothing but one extreme example. These cells thus form the first stage in extracting meaningful shape information. Appropriate grouping of these cells help in detecting texture boundaries, shape recognition or illusory contours, depending on the context. The use of these cells in shape representation is further explored in Chapter 5 on face recognition.

4.4.4 Grouping and Boundary Detection

The final stage involves grouping similar orientations. The grouping process receives inputs both from the competitive stage (4.18) and from the end-detectors (hypercomplex cells) described in section 4.4.3. Note that the orientation of the activating end-detector is orthogonal to the actual orientation of the grouping process. This incorporates the observation made in [85, 96] that hypercomplex cells are responsible for detecting illusory contours. Abrupt line endings signal an occluding boundary almost orthogonal to the edge orientation, and this is represented by these end-inhibited cells providing input to the grouping process nearly orthogonal in their orientation preference. If $Z_i(s, \theta)$ represents the output

of this process, then

$$Z_i(s,\theta) = g\left(\int d_i(s-s',\theta)(Y_i(s',\theta) + Q_{ij}(s',\theta')ds'\right) \tag{4.22}$$

$d_i(s,\theta)$ represents the receptive field of $Z_i(s,\theta)$ and in our experiments we have used

$$d(s=(x,y),\theta) = \exp(-(2\sigma^2)^{-2}[\lambda^2(x\cos\theta + y\sin\theta)^2 + (-x\sin\theta + y\cos\theta)^2]) \tag{4.23}$$

where θ is the preferred orientation, θ' is the corresponding orthogonal direction, and λ is the aspect ratio of the Gaussian. The Z cells thus integrate the information from similar oriented cells within each frequency channel and from hypercomplex cells of appropriate orientation, and thus help in grouping the features and in boundary completion. Since the various frequency channels are sampled, the effective standard deviation of the Gaussian is σ/α^i, where α^i is the scale parameter for channel i.

To summarize, this approach consists of three distinct steps (a) Feature detection using Gabor wavelets, (b) Local interactions between features and (c) Scale interactions to generate end-inhibition. We now explain how to use the output $Z(\cdot)$ from different frequency channels to detect edges and texture boundaries.

Intensity edges and illusory contours

In section 4.4.1 the usefulness of energy detectors in localizing image features was discussed. In detecting the intensity edges in the image we used the energy features as input to the competitive stage. Thus the input to a cell in the competitive stage at a position (x,y) in the ith frequency channel is given by

$$I_i(x,y,\theta) = ||W_i(x,y,\theta)|| \tag{4.24}$$

where $W(\cdot)$ is as in (4.6), and $i = \{0,-1,-2,-3,\ldots\}$ and $\theta = k\pi/N, k = \{0,1,\ldots,N-1\}$, N is the number of discrete orientations. The edges are located at the local maxima of the $Z(\cdot)$ field in (4.22). These energy features are also used in our experiments to detect the line-ends through scale interactions. The

perceptual boundaries for the examples in Figure 4.10 are marked at the local maxima of the $Z(\cdot)$ field.

Texture boundaries

The information extracted by the wavelets can be used in several ways to detect textures, though the results reported here are obtained using the energy measure. Texture boundaries are located at the local maxima of the gradient of the Z field. Scale interactions also play an important role in texture boundary detection as is evident from the example in Figure 4.9 where the two regions differ only in the distribution of intersections and corners.

4.5 Experimental Results

The performance of the model is illustrated on several images. The following parameter values were used in our experiments described here: $\beta = 4.0$ in the transfer function $g(\cdot)$. The strengths of the inhibitory synapses in (4.18) are $b_{s,s'} = 1/||N_s||$ and $c = 1/N$, where $||N_s||$ is the cardinality of the neighborhood set and N is the number of discrete orientations used. Unless otherwise stated, $N = 4$ and N_s consists of the four nearest neighbors of s. The aspect ratio of the Gaussians in both the Gabor wavelets (4.4) and in the receptive field of Z cells (4.23) is set to 0.5. If more than one channel is mentioned then the result shown is a superposition of the boundaries detected in the individual channels. Since the various frequency channels are sampled, in order to bring them to the original image size the output of the grouping stage are first convolved with appropriate size Gaussian smoothing filters, and then the boundaries are detected.

Regarding the implementation of the dynamics of competition, we used a simple gradient descent on the corresponding energy function (4.20) instead of solving the set of differential equations. The equilibrium points in general for these two methods will be different, but gradient descent on E in (4.20) will be much faster (typically it takes less than 50 iterations to converge on a 256×256 image).

Example 1 (intensity edges): Figure 4.5 shows two examples of edge detection using

the energy measures. Figures 4.5(a) and (c) show the original 256×256 images. The edges shown in Figure 4.5(b) are detected in channels $\alpha^i = \{1/\sqrt{2}, 1/2\}$ and in (d) they correspond to the channel $\alpha^i = 1/\sqrt{2}$. In both cases σ is set to 1.

Example 2 (natural textures): Figure 4.6 shows the boundaries detected in an image consisting of four textures, grass, water, wood and raffia. The wood texture is present at two regions at different orientations. The parameter values used are $\alpha^i = \{1/2, 1/2\sqrt{2}, 1/4\}$ and $\sigma = 5.0$.

Example 3 (LT-T texture): Figure 4.7 shows the results on a synthetic texture which is often used in psychophysical experiments. The boundary between L and Ts is not easily perceived where as that between straight and oriented Ts clearly stands out. This boundary can be easily detected in almost all frequency channels, and the parameters values used in this example are the same as in the previous example.

Example 4: Figure 4.8(a) shows a texture consisting of micropatterns which differ in their sign. This particular texture is generated by adding to a constant intensity background (intensity value 120 on a 0-255 scale) patterns formed of bright (intensity 200) and dark (intensity 40) regions. Malik and Perona [67] correctly observe that the two regions in such textures can not be distinguished using energy measures. This was one of the motivations for using halfwave rectification in their model, and is based on the assumption that the filters are exactly zero mean and no non-linear transformations of the intensity prior to filtering. The cosine component of the Gabor wavelet filters used in here is not exactly zero mean (though very close to zero). The non-linearities following filtering enhance the differences at the boundaries as is illustrated by the boundary detected (Figure 4.8(b)). Slight bias in the patterns towards one of the grey levels as in Figure 4.8(c) (which has a background level of 150, brighter region at 200 and the darker region at 80) significantly influences the strength of the boundary (Figure 4.8(d)).

Example 5 (L-Plus texture): This example illustrates the importance of end-inhibition in texture boundary detection. Figure 4.9 shows another of commonly used texture consisting of randomly oriented Ls and +s. Unlike the previous example, orientation information can not be used for segmentation. The line

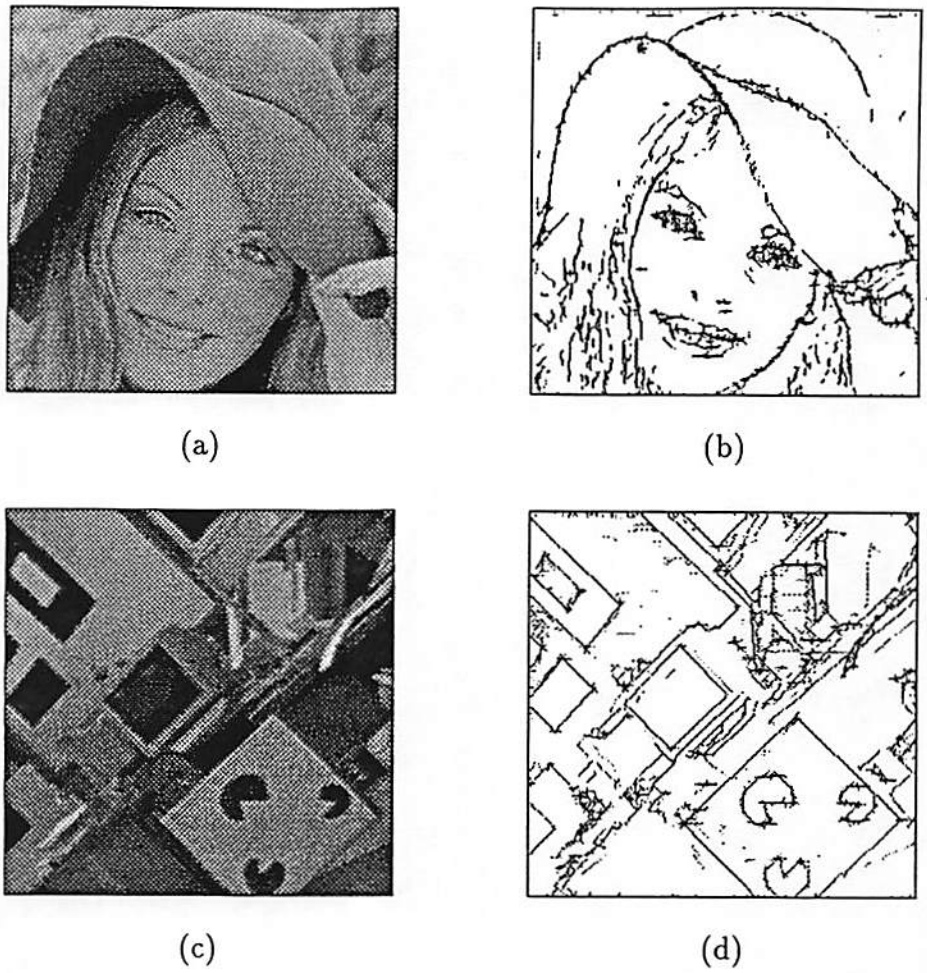


Figure 4.5: (a) and (c) show two 256×256 images and the corresponding edges detected are shown in (b) and (d). In (b) the edges are from two channels $\alpha^i = \{1/\sqrt{2}, 1/2\}$ and in (d) $\alpha^i = 1/\sqrt{2}$. For both examples $\sigma = 1$.

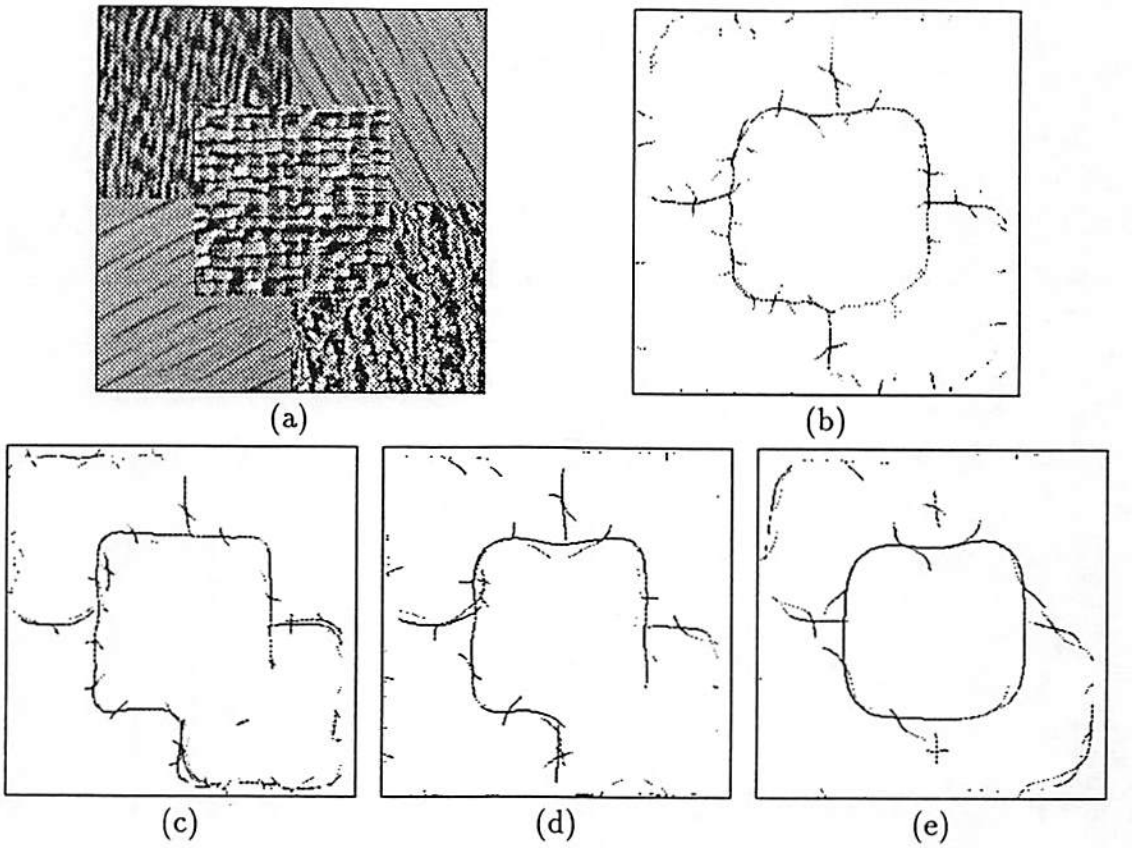
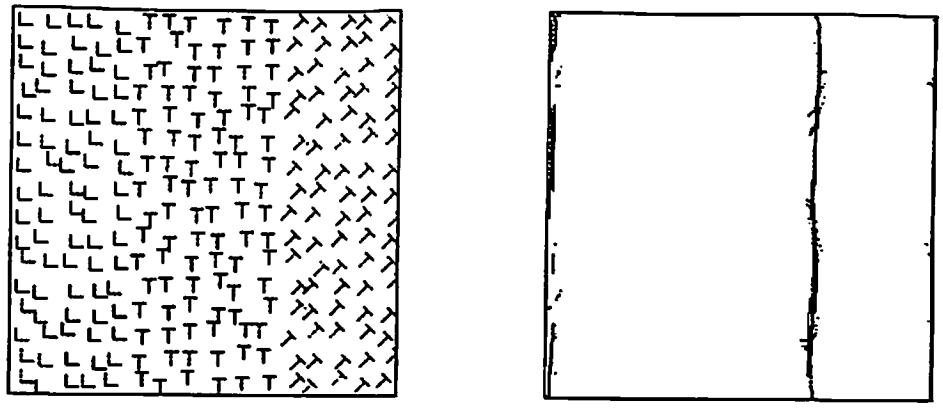


Figure 4.6: (a) Image consisting of four natural textures, water, wood (in two regions at different orientations), raffia and grass. (b) texture boundary detected using the scales $\alpha^i = \{1/2, 1/2\sqrt{2}, 1/4\}$ and $\sigma = 5$ pixels. (c),(d) and (e) show the texture boundaries detected in each of these individual frequency channels separately. The result in (b) is obtained by superimposing the boundaries in (c)-(e), filtering using a Gaussian filter (to smoothly combine the boundaries) and thresholding. The filter used has a standard deviation of 4 pixels.



(a) (b)

Figure 4.7: Texture consisting of three regions, L, T and tilted-Ts. The boundary between L and T s can not be easily detected. However the orientation difference between the two T regions is enough to discriminate between the two regions in almost all frequency channels. The boundary shown in (b) corresponds to the combined output from channels $\alpha^i = \{1/2, 1/2\sqrt{2}, 1/4\}$ and $\sigma = 5$ pixels.

segments forming Ls and +s have the same length (7 pixels). The two regions differ in the distribution of corners, line-ends and intersections. As we discussed in section 4.4.3, scale interactions play an important role in detecting these features. None of the scales by themselves contain enough information to segment the two regions, but using these interscale interactions the boundary between the Ls and +s can be detected (Figure 4.9(b)). The boundary shown is for the case of using the interactions between scales corresponding to $\{1/2, 1/4\}$ with a $\sigma = 16$. In this context it is interesting to note the observation by Bergen and Adelson [4] that the L+ texture can be discriminated by simple linear filtering followed by rectification, where they used size tuned center-surround filters. These filters are the simplest case of filters having inhibitory end-zones and as such respond to *blobs* of certain size. Hence in a sense, they are sensitive to the distribution of line ends and corners, though the authors comment that "...this discrimination might be based on the density of such features as terminators, corners, and intersections within the patterns. We note here that simpler, low level mechanisms tuned for size may be sufficient to explain this discrimination".

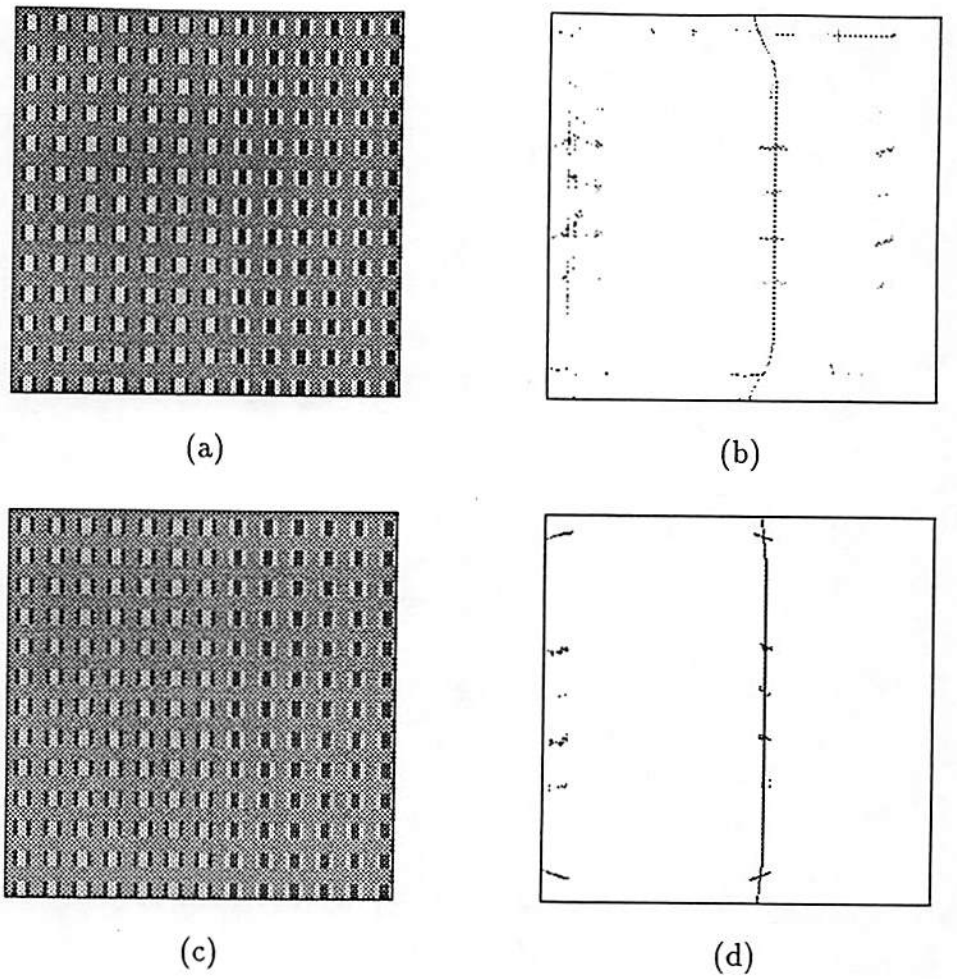


Figure 4.8: (a) Primitives in this texture are zero mean (i.e., mean equals the background intensity level) patterns, with the intensity levels of the background, brighter and darker regions respectively at 120, 200 and 40 (on a 0-255 scale). We were able to detect the boundary between the two regions using the energy measure and (b) shows the result for $\alpha^i = 1/2\sqrt{2}, \sigma = 5$ pixels. Even a slight offset in the mean of the patterns can result in significant increase in the strength of the boundary. In (c) the intensity levels are adjusted to be non-zero mean (at 150,80 and 200 respectively for the background, darker and brighter regions, a net difference of 10 intensity levels between the background and the patterns) and the boundary detected in (d) is twice as strong.

Filters which have such inhibitory end zones include Laplacian of the Gaussians and the DOGs, and have been used in texture discrimination of L+ patterns in [67, 102].

Example 6 (Illusory contours): The usefulness of scale interactions in detecting line endings and their subsequent grouping to detect illusory contours is illustrated in Figure 4.10. For the line (Figure 4.10(d)) and sine wave (Figure 4.10(e)) contours the results shown are for $\alpha^i = \{1/2, 1/4\}$, $\sigma = 8$. For the circle (Figure 4.10(f)) $\alpha^i = \{1/\sqrt{2}, 1/2\}$ and $\sigma = 2$.

4.6 Conclusions

In this chapter we have developed a common framework for detecting perceptually significant features such as edges, textures and illusory contours. We have suggested a simple model based on detecting oriented features at different spatial scales and on local interactions between features. Interaction between frequency channels is used in generating end-inhibition which plays an important role in boundary perception. Several examples are provided to illustrate the performance of this approach in detecting different types of boundaries. The next chapter develops on the end-inhibition topic, its role in curvature detection and shape representation, and presents a simple system for recognizing human faces from their images.

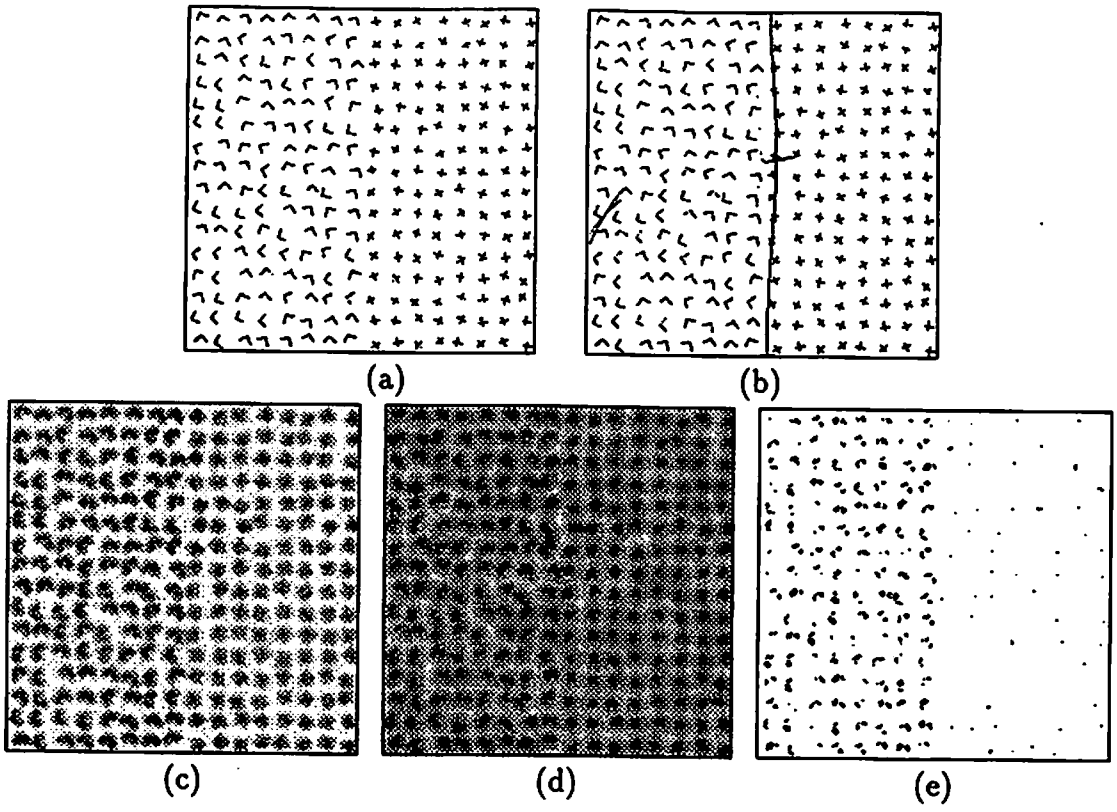


Figure 4.9: Texture consisting of randomly oriented L and +. The line segments of the primitives are 7 pixels wide and the image is 256 × 256 pixels. The two regions differ in the distribution of line-ends, intersections and corners. The boundary shown in (b) (superimposed on the original texture) is detected using the output of the scale interactions with $\sigma = 16$. The scales used in this example are $\alpha^i = \{1/2, 1/4\}$, and figures (c) and (d) show the result of convolution and (e) shows the output after the interactions.

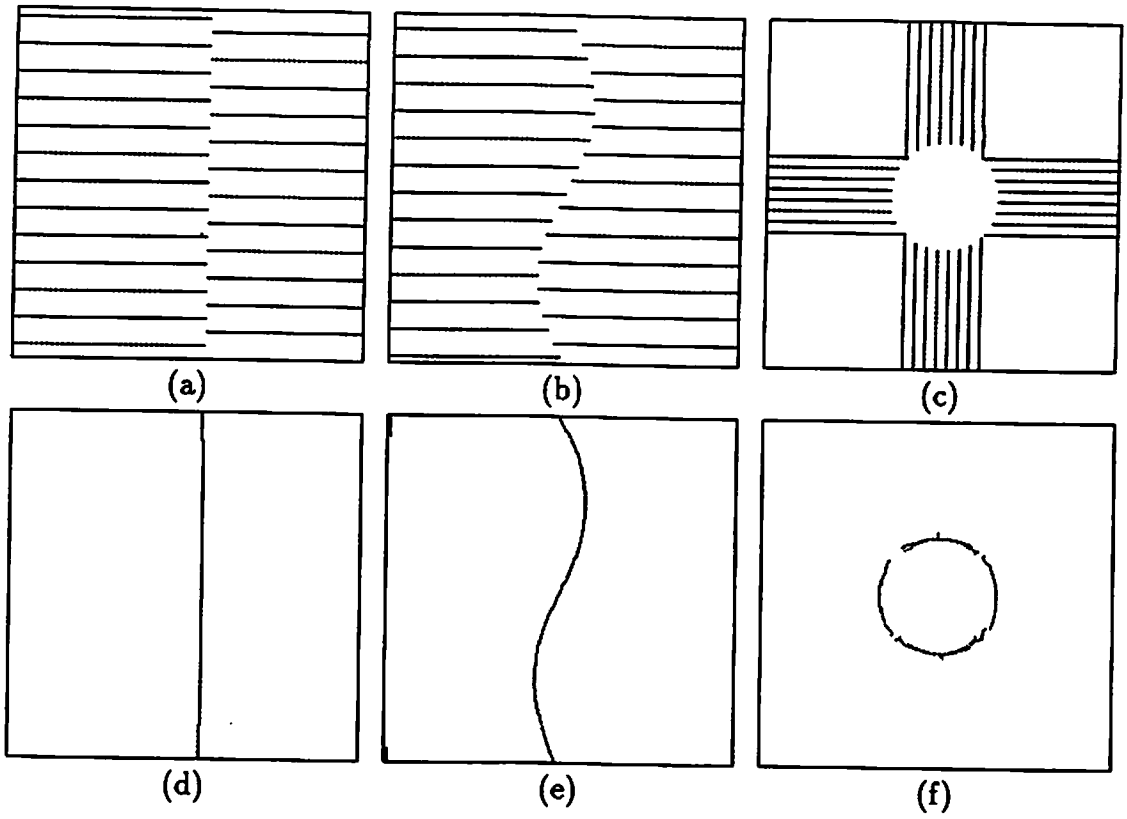


Figure 4.10: Some examples of illusory contours formed by line terminations ((a),(b), and (c)) and the detected contours (d) and (e) correspond to the interaction between scales $\alpha^i = \{1/2, 1/4\}$ and $\sigma = 8$. In (f) the scales are $\alpha^i = \{1/\sqrt{2}, 1/2\}$ and $\sigma = 2$

Chapter 5

Feature Detection and Representation: Application to Human Face Recognition

In this chapter we continue our discussion on the role of end-inhibition in early visual processing. We suggest that the activities of cells with this end-inhibition property can be used to represent salient features in the image, and in some sense encode local shape information. The feature localization scheme described here has been successfully used in applications such as motion tracking and in image registration. Here we demonstrate the role of these end-inhibited cells in representing shape information as part of a simple human face recognition system. This system further illustrates that,

- Recognition does not necessarily need detailed 3-D representations.
- From the very early stages in vision, robust representations based on 2-D features can be built to facilitate the task of recognition.

It is of particular interest to note that the mechanisms that are used in representing shape information are also involved in the perception of visual illusions such as illusory contours, as we discussed in Chapter 4.

5.1 Introduction

Feature detection and representation is a basic issue in most intermediate level vision problems such as stereo, motion correspondence, image registration etc.. In the literature there are many algorithms which detect salient features in the image and use information in the neighborhood of these feature points for further applications [58, 78]. Typical features detected include sharp corners, points with significant curvature discontinuities and points with large local intensity variations. Here we propose to use the scale interaction model for end-inhibition developed in the previous chapter as a robust method for feature selection and representation. Unlike many other methods such as the Moravec's operator [78], our scheme directly provides a means for quantitatively representing feature information which can be used as a basis for representing shape information as well. This we illustrate in our development of a simple face recognition system. At the heart of recognition are two issues: *representation* and *matching*. Representation involves detecting useful features, and representing information about the objects in an appropriate data structure. Matching involves finding which one of the stored representations, if any, identify with the image.

5.1.1 Shape Representation

The current work on representation can be grouped into two categories: the first one involves creating a detailed 3-D representation of the object, and using this to find appropriate transformations that can lead to the recognition of an unknown projection. The second approach involves using features extracted from multiple views of the object. Creating a detailed 3-D representation of objects is an expensive task, both in terms of storage requirements and computations. Besides, it needs in the first place accurate depth information about objects, which is hard to obtain from images. The current techniques for extracting 3-D information from 2-D images, such as stereo, shape from shading etc., do not provide such a description. Not surprisingly, most of the current work on 3-D object recognition uses range data as input. Since our interest is in recognizing objects from intensity images, we will not discuss further the work on range

images, and for some of the recent research on this topic we refer to [6, 31, 81].

Our approach closely parallels the arguments made in [66] that it is not necessary to create a detailed 3-D reconstruction as an intermediate step for object recognition. In other words, it is not necessary that all aspects of an object to be recognized are known. Rather it involves efficiently representing salient features extracted from multiple views of the 3-D objects. For example in the SCERPO vision system [66], Lowe uses edges and lines as the basic features. Asada and Brady [2] suggest that curvature changes are a very useful source of information, and provide a description (for 2-D shapes) in terms of these curvature changes. They call their representation a *curvature primal sketch.* Recognition can also be based on 3-D volumetric primitives such as blocks, cylinders and cones, detected from 2-D images. Biederman [7] enumerates a set of such primitives, which he calls *geons*, for representing complex objects, and refers to his approach as recognition-by-components. He suggests that perceptual organization of 2-D features such as curvature, collinearity, symmetry, parallelism and cotermination can be used to compute these geons. Representation of objects in terms of relations between these geons would be robust to variations in viewpoint over a wide range. Although quite appealing in theory, it is not clear how easy or difficult it is to detect these geons from real images. Further, a system based on a qualitative description of geons might be able to differentiate between objects belonging to two different generic classes, say a man's face from that of a horse, but will not provide a sufficiently strong description to identify or recognize objects within the same class, such as recognizing human faces.

There may not be a unique solution to the problem of representation, and different types of features might be involved in different contexts. As mentioned in the previous paragraph, while high level descriptions such as geons might be useful in classifying objects belonging to different categories, a lower level description such as local curvature would provide a more quantitative description in applications such as face recognition. The fact that in the visual cortex there are specialized regions for tasks such as face recognition supports this view [84].

5.1.2 Matching

Matching refers to obtaining correspondence between an image and one of its stored representations. It involves searching through the database in order to obtain the required correspondence, and is a computationally expensive task. Almost all model based approaches require finding suitable affine transformations that best align the image with one of the stored models. Lowe [66] identifies two aspects of this search: computing an appropriate viewpoint, and then searching for the object which best matches this viewpoint. Perceptual organization plays a very important role in the search process, and helps in limiting the search space. Reduction in search space is obtained by segmenting the scene into perceptually significant features, by deriving 3-D interpretations from these features, and identifying only a subset of objects from the database for further consideration. One of the features of the SCERPO system developed by Lowe is that it is able to distinguish accidental relations between features from non-accidental ones, and is central to the use of perceptual organization in grouping the features.

More recently there have been many attempts at formulating matching as an optimization problem. Since one can use a data structure such as a topological graph for representing relational information between features, matching can be thought of as a problem of matching an input graph with that of a stored one. Paul Cooper in his Tinker Toy project [23] uses Hopfield network for this graph matching problem. As we discussed in Chapter 2, networks such as Hopfield are ill-suited for expressing syntactical relationships. The cost that one has to pay for using them is in terms of encoding these relations exhaustively, as by using additional neurons to represent the bindings between features. Malsburg and Bienenstock [100] instead suggest the use of dynamic connections to represent bindings and formulate a cost function in terms of these variable for graph matching. Their cost function includes terms which count the number of four cycles in the matched system, and constraints on number of matches per node. A variant of this is is used in [10, 63] in their work on face recognition. This idea is also used by Mjolsness et al. [77] in their formulation of object recognition as an optimization problem.

5.2 Previous Work on Face Recognition

Human faces provide a very good example of a class of natural objects which do not lend themselves to simple geometrical representations, and yet the human visual system does an excellent job in efficiently recognizing these images. Considerable research has been done in developing algorithms to solve this problem. Kanade [54] describes one of the early systems built for this task. The system automatically localizes features such as corners of the eyes, nostrils, mouth etc.. Then a set of sixteen facial parameters corresponding to these features is computed. They correspond to ratios of distance and area, and angles to compensate for scaling differences. A simple Eucledian distance measure is then used to compute the similarity between a test face and a stored face. The best case performance of the system was 15 correct identifications out of 20 test faces. The test data differed from the training data in that there was a period of one month between the acquisition of the samples, and in both cases the full frontal view was obtained.

More recently Turk and Pentland [95] describe a real time working system for face recognition. Their system tracks a person's head and identifies the face by comparing its features with a known database. The basic idea is to find a low dimensional feature space to represent the intensity data, and they make use of principal component analysis. Since intensity data is directly used in the recognition process, such a system will be prone to local fluctuations in the image. Their database consists of 2500 images of 16 persons taken over different lighting conditions, image size and orientation. They report classification accuracy of 96% over lighting variations, 85% over orientation variations and 64% over size variation. This approach to recognition is similar to many other earlier attempts in transforming a 3-D recognition problem to a 2-D matching, without detecting any perceptually significant features. See for example Kohonen's associative memory for face detection [61], and Fuch and Haken's associative memory [32, 33].

Kanade's, and Turk and Pentland's approaches reflect two extremes in solving this problem. In a different context, and as an example to illustrate the principle of dynamic link architecture, Martin et al. [63] also provide results of their

experiments on face recognition. In their case, the basic features are the Gabor coefficients obtained by convolving the image with a bank of Gabor filters at multiple scales and orientations. As we discussed in Chapter 4, these features correspond to edges and lines in the image. The authors report an impressive 100% classification over a data set of about 80 faces, with the training and test sets differing in the orientations of faces. However, their procedure does not appear to be completely automatic, and human intervention is needed in the selection of "feature points". Also, their approach appears to be quite sensitive to illumination changes.

Our development here continues the discussion in the previous chapter of the role of end-inhibited cells in early processing. The activities of these cells represent local curvature information, and this is used as the basis for representing shape.

5.3 Feature Detection and Localization

We discussed in the previous chapter the role of hypercomplex cells in texture grouping and in the detection of illusory contours. The inhibitory end zones of these cells make them selective to local curvature, in addition to being responsive to short lines and line endings. This was observed by Hubel and Wiesel [49], and more recently demonstrated by Dobbins et al. [28]. The activities of these cells represent curvature changes at different spatial scales, thus representing in some sense a *curvature primal sketch.* This is further illustrated in Figure 5.1. The first step is to localize these curvature changes. Locations (x, y) in the image which are identified as feature locations satisfy the following:

$$Q_{ij}(x,y) = \max_{(x',y') \in N_{xy}} Q_{ij}(x',y') \tag{5.1}$$

where

$$Q_{ij}(x,y) = \max_{\theta} Q_{ij}(x,y,\theta)$$

and $Q_{ij}(x,y,\theta)$ is given by (4.21). N_{xy} represents a local neighborhood of (x,y) within which the search is conducted, and in our experiments we set this to a

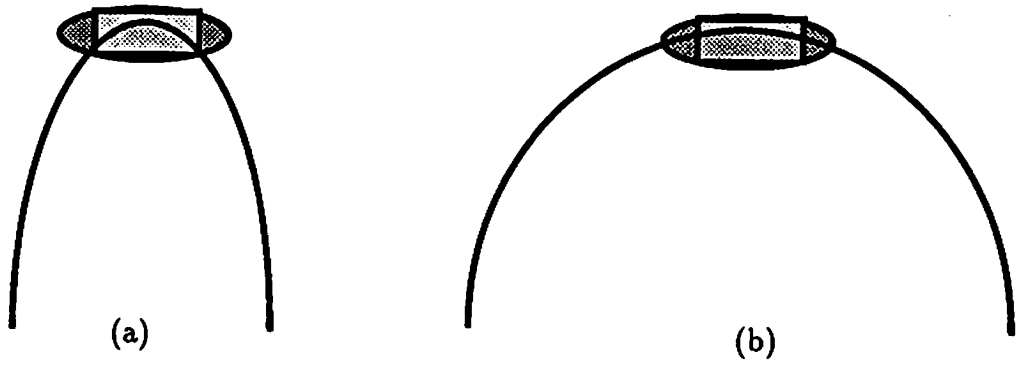


Figure 5.1: Illustrating the selectivity of end-inhibited cells to curvature changes. In (a) the inhibitory end zones of such a cell are not activated, and the cell in turn responds strongly to the local curvature. In (b) the same cell is not activated as the inhibitory end zones suppress its activity. In our model these inhibitory end zones are simulated through the interactions between simple cells at different scales.

circle of radius equal to the standard deviation of the Gaussian of the coarser of the two scales used in generating end-inhibition.

We have applied this method of detecting features to various applications, including recognition. Figure 5.2 illustrates the observation that the features located correspond to local curvature changes Figure 5.3 shows the type of features that are detected on face images. Information at these locations is used in the recognition process and will be discussed in detail in the following.

5.3.1 Applications to Motion Tracking and Image Registration

This feature detection scheme has been successfully applied to some practical problems. The first example is illustrated in Figure 5.4. The figure shows the first frame in a motion sequence. The goal is to detect and track features, and to compute motion and structure parameters. Also shown in Figure 5.5 are

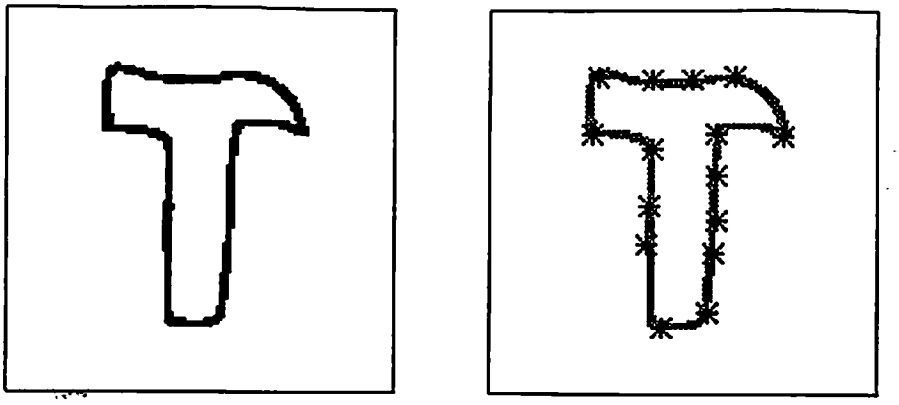


Figure 5.2: Salient features detected by the system. For the hand drawn hammer image, all the feature locations correspond to significant changes in curvature. The particular scale-pair used in this example is $i = 0$, $j = -6$, with $\alpha = \sqrt{2}$. These parameter values correspond to the highest and the lowest spatial frequencies in our system, corresponding to 1 and 8 pixel standard deviations of the Gaussians, respectively.

the feature points detected using the scheme outlined in the previous section. Some of these points are them tracked over successive frames. For tracking, the information contained in the Gabor wavelet transformation (discussed in the previous chapter) is used, and the correspondence problem is integrated with that of motion estimation. The details are beyond the scope of this chapter and we refer to [18, 17]. The tracking results are shown in Figure 5.5.

Another application where this feature detection scheme has been quite effective is in obtaining correspondence in aerial images [108]. These aerial images do not contain any easily perceivable structures which can be detected and used in obtaining correspondence. The problem is of considerable interest as its solution would be a part of a planned future mission to Mars involving estimation of surface wind velocities. As part of the project, cameras strapped to a balloon would take pictures of the surface of Mars, and the wind velocity is estimated by computing the motion of the balloon. Central to this is solving the correspondence problem. Figure 5.6 shows two images in the sequence, and in Figure 5.7 the features identified are shown. The key to the matching problem is the idea of Zheng [108] to use techniques from shape from shading algorithms to estimate

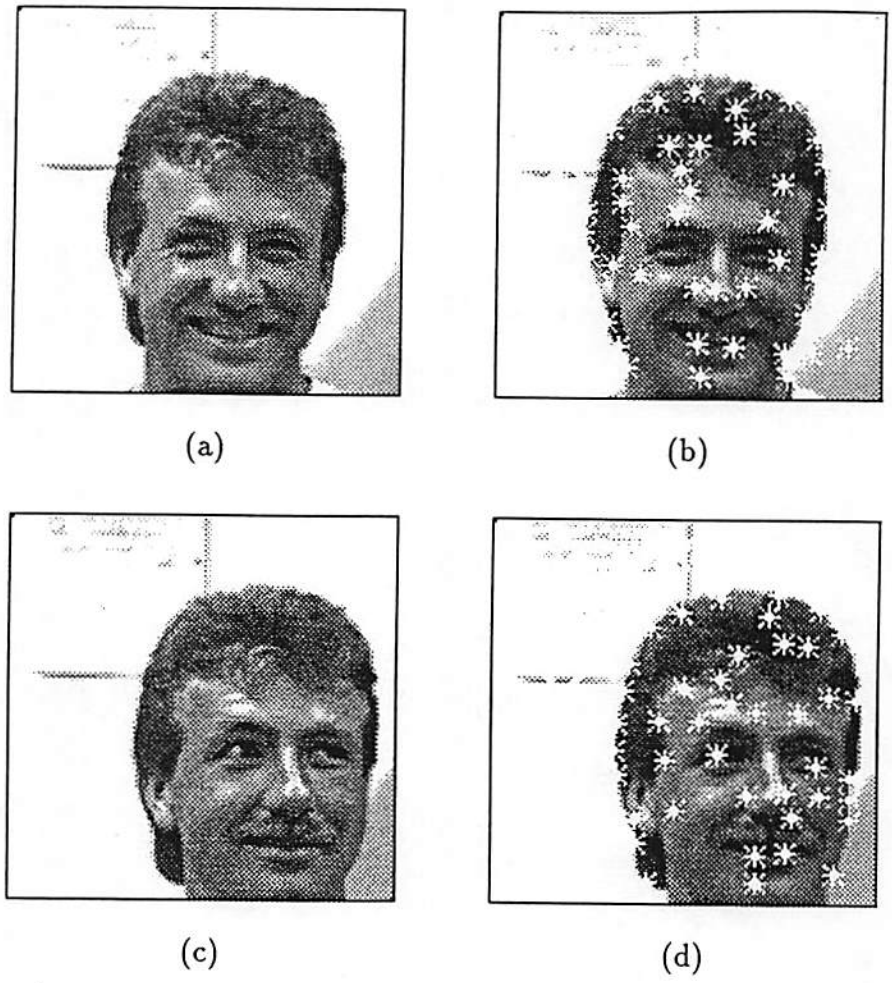


Figure 5.3: Feature locations marked for the face images. The scales used in this case correspond to $i = -2, j = -5$ ($\alpha = \sqrt{2}$). Information at the feature locations is stored and used during the recognition process. The two faces shown here are matched to each other from a database of over 300 images. In general, to detect features at various scales, multiple scale interactions need to be considered.

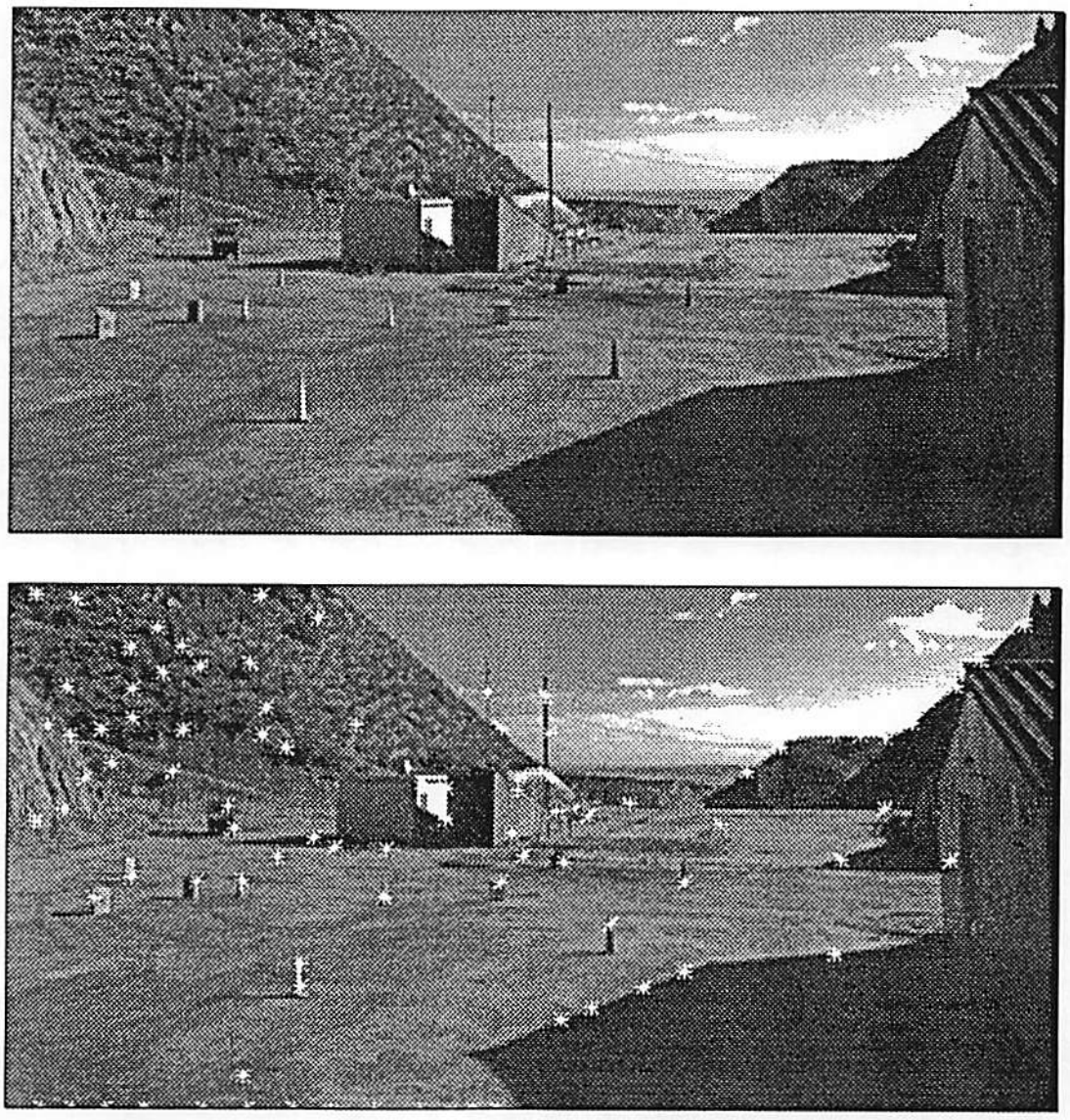


Figure 5.4: First image in the ROCKET sequence (courtesy UMASS CS department) and the feature points detected.

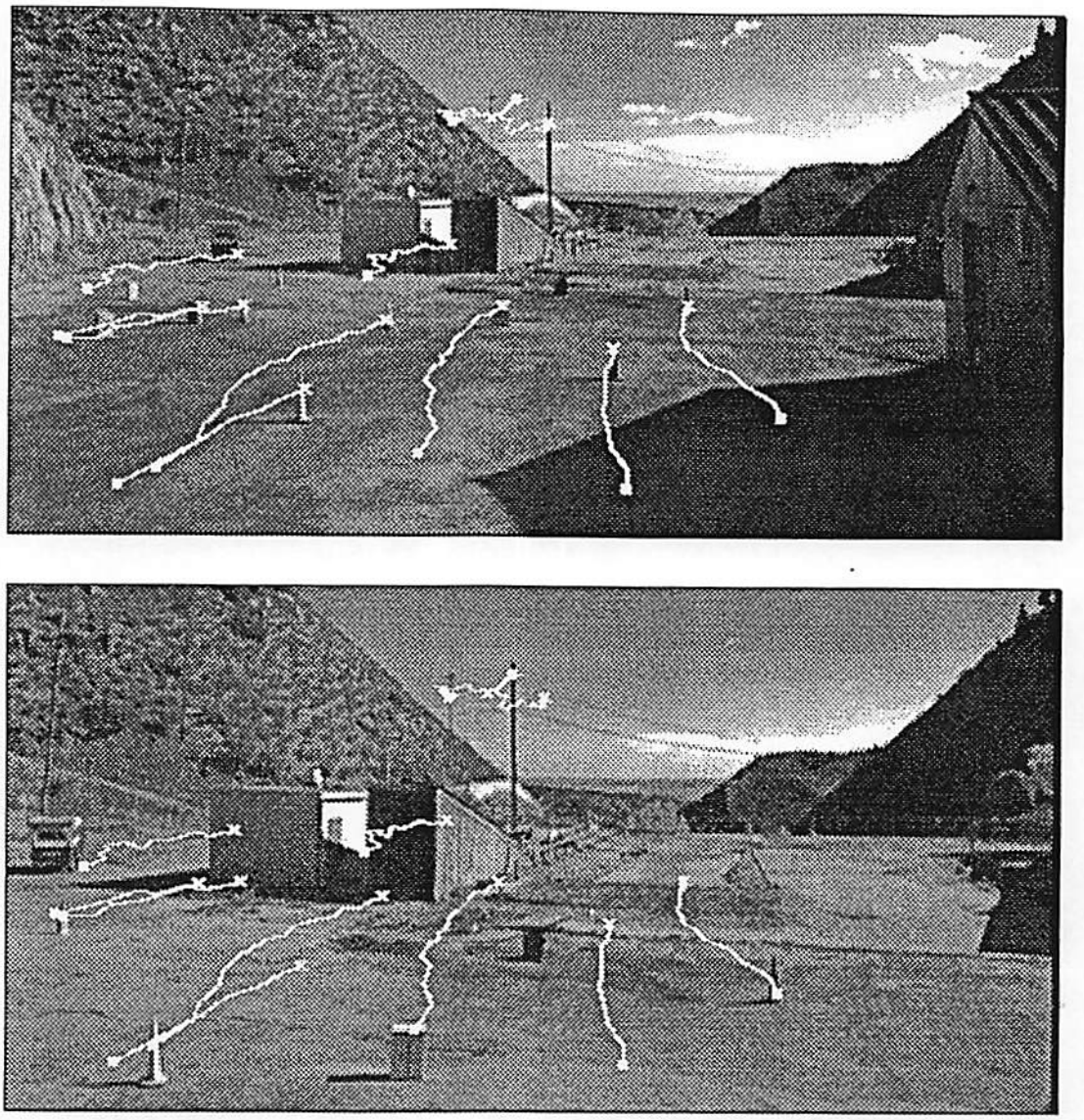


Figure 5.5: Features selected for tracking. Initial positions are marked by $*$ and final positions by solid squares. Final positions are after 16 frames. The trajectories are shown superimposed on the first and the sixteenth frame in the sequence. (courtesy S. Chandrashekhar [18]).

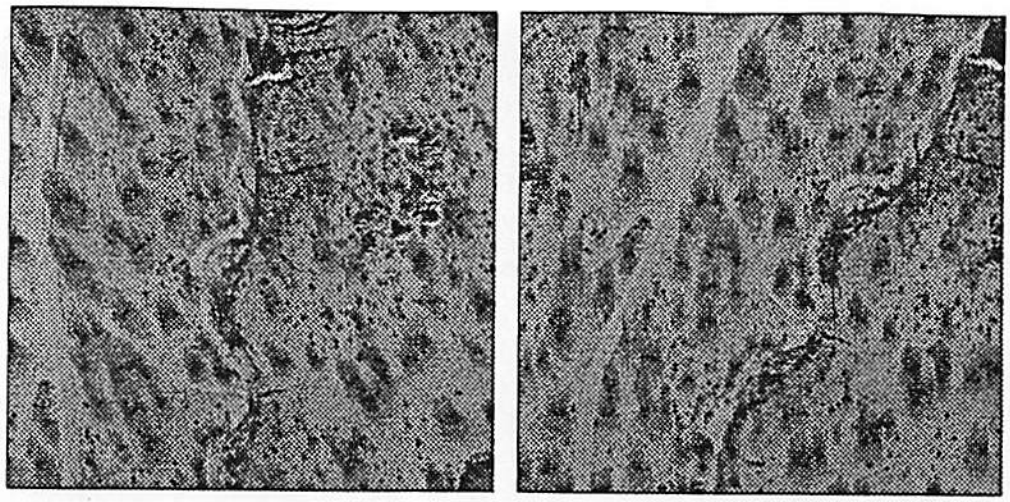


Figure 5.6: Two successive images from a motion sequence (courtesy: Peter Kroger of JPL).

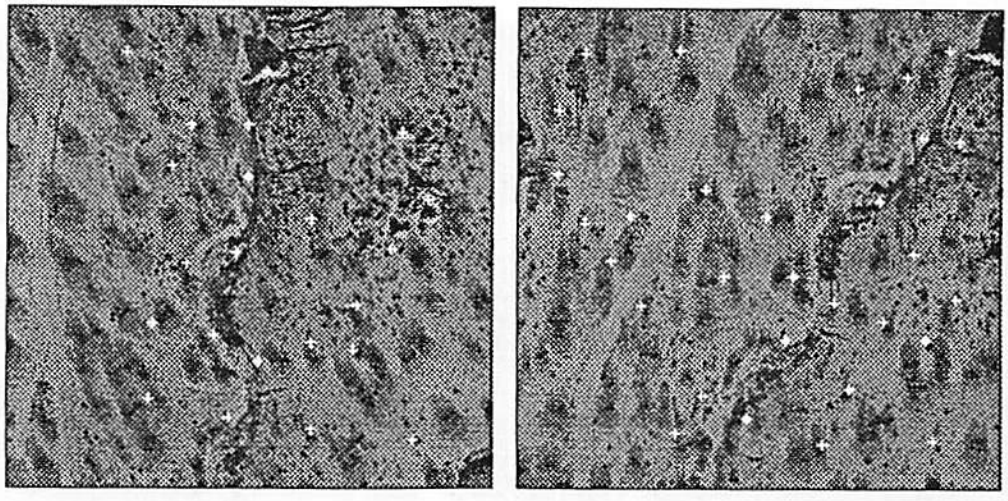


Figure 5.7: Features detected using our model.

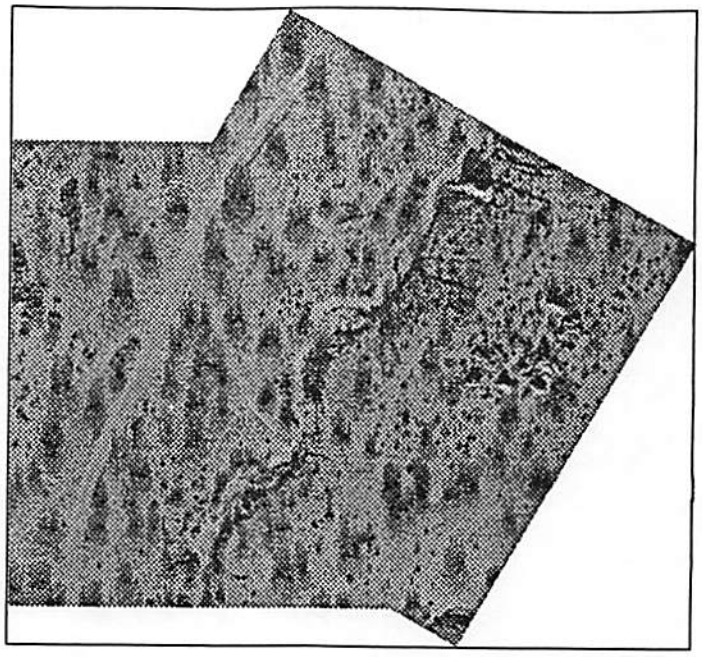


Figure 5.8: An exhaustive search is performed to match the corresponding features. The two images are superimposed (after performing the required affine transformations) to show overlapping regions. (courtesy Q. Zheng [108]).

the rotation between the images. Once rotation is estimated, the intensity information around the feature points are then used in an exhaustive search scheme, the details of which can be found in [108]. The results are described in Figure 5.8.

5.4 Application to Face Recognition

5.4.1 Representing Shape Using Graphs

Topological graphs are used in our recognition scheme to represent relationships between features. The only measure we use here is the distances between the features. For convenience the features detected in a given image are numbered as {1,2,..} (in any arbitrary, but consistent way). The nodes V_i in the graph correspond to the feature points, and are characterized by $\{S, \mathbf{q}\}$, where S represents

information about the spatial location, and

$$\mathbf{q}_i = [Q_i(x, y, \theta_1), \ldots, Q_i(x, y, \theta_N)] \tag{5.2}$$

is the feature vector corresponding to the ith feature. Let N_i denote the set of neighbors of ith node. Directional edges connect the neighbors in the graph (i.e., the neighborhood is not symmetric). Neighborhood of a node is determined by taking into account both the maximum number of neighbors allowed as well as the distance between them. The Eucledian distance between two nodes V_i and V_j is denoted by d_{ij}.

5.4.2 Graph Matching

To identify the input graph with that of a stored one (which is most similar to the input one based on certain criteria) efficiently is another important issue, and has received considerable attention recently. We describe below a very simple algorithm which only involves local search, is deterministic in nature and extremely fast. The algorithm, however, does not guarantee optimizing the criterion function. In spite of this, the recognition rate is comparable to most face recognition schemes that we are aware of, demonstrating further the robustness of our feature extraction. Our implementation of the matching algorithm is given below:

In the following, subscripts i, j refer to nodes in the input graph $\mathcal{I}$, and i', j', m', n' correspond to nodes in the stored graph $\mathcal{O}$.

1. The input graph $\mathcal{I}$ is spatially aligned with the stored graph $\mathcal{O}$ by matching the centroids of the features $\{V_i\}$ and $\{V_i'\}$.

2. Let N_i be the spatial neighborhood for the ith feature in the input graph, over which a search is conducted to find the best matching feature node $V_{i'}$ in the stored graph, such that

$$S_{ii'} = 1 - \frac{\mathbf{q}_i \cdot \mathbf{q}_{i'}}{||\mathbf{q}_i||\ ||\mathbf{q}_{i'}||} = \min_{m' \in N_i} S_{im'} \tag{5.3}$$

3. After all the individual features are matched, total cost is computed by

taking into account the topology of the matched graphs. Let the nodes i and j match i' and j' respectively, and further let $j \in N_i$ (i.e., V_j is a neighbor of V_i). Let $\rho_{ii'j'j} = \min\{d_{ij}/d_{i'j'}, d_{i'j'}/d_{ij}\}$. Then the topology cost for this particular pair of nodes is computed as

$$T_{ii'j'j} = 1 - \rho_{ii'j'j} \tag{5.4}$$

Note that if the match is perfect, $d_{ij} = d_{i'j'}$ and $T_{ii'j'j} = 0$.

4. The total cost for matching input graph $\mathcal{I}$ to a stored graph $\mathcal{O}$ is then given by

$$C_1(\mathcal{I}, \mathcal{O}) = \sum_i S_{ii'} + \lambda_t \sum_i \sum_{j \in N_i} T_{ii'j'j} \tag{5.5}$$

where λ_t is a scaling parameter which controls the relative importance of the two cost functions.

5. The total cost is then scaled appropriately to reflect the the difference in the number of features between the input and stored graphs. If $n_{\mathcal{I}}, n_{\mathcal{O}}$ denote the number of feature nodes in the input and stored graphs respectively, then the scaling factor $s_f = \max\{n_{\mathcal{I}}/n_{\mathcal{O}}, n_{\mathcal{O}}/n_{\mathcal{I}}\}$, and the scaled total cost $C(\mathcal{I}, \mathcal{O}) = s_f C_1(\mathcal{I}, \mathcal{O})$.

6. The best candidate match $\mathcal{O}^*$ then satisfies

$$C(\mathcal{I}, \mathcal{O}^*) = \min_{\mathcal{O}'} C(\mathcal{I}, \mathcal{O}') \tag{5.6}$$

Note that the above algorithm does not take into account the topology cost during the matching process. The topology cost is computed only after the features are matched. The advantage is that there are no iterations, and no stochastic elements invoked in the search, resulting in a very fast algorithm for matching.

5.5 Experimental Results

We have implemented a simple face recognition system based on the above principles. The input is 128 × 128 image, having very little background noise. In our current implementation, the responses of hypercomplex cells are computed at only one scale, corresponding to the scale parameters $\alpha = \sqrt{2}, i = -2, j = -5$ in (4.21). Typical number of feature points detected in a face image using (5.1) vary from 35 to 50. Number of discrete orientations used was $N = 4$ (in (5.2), corresponding to $\theta = \{0, 45, 90, 135\}$. One byte of information is stored for each of the components in the feature vector, or approximately about 200 bytes of information per face. This constitutes an order of magnitude savings in memory, from a 16K raw intensity data.

The database we have used has face images of 86 persons, with two to four images per person, taken with different facial expressions, and/or orientations. Often there is a small amount of translation and scaling as well. The neighborhood set N_i of a feature node i consists of its five nearest neighbors. Note that this set is not necessarily symmetric. The parameter λ_t in (5.5) is set to 0.2, so as to have equal contributions to the total cost from the similarity measure and topological cost (as the summation over j is over the neighbors, which in our case total five). The graph matching steps 1 through 5 discussed in section 5.4.2 typically takes less than 0.5 seconds for each graph (on a SUN-Sparc workstation). Some typical results of successful matches as well as failures are shown in Figures 5.9, 5.9 and 5.11, and Table 1 summarizes the results. These results show that the system tolerates a fair amount of rotation and distortion. In cases of failures, often the best match has the same orientation or facial expression as the input. We observed that by increasing the number of images per person in the database from 2 to 4 (by adding additional orientations and/or facial expressions), the performance improved by about 10%. Since the storage requirements of the systems are quite low, in a practical application one can store 8-12 different views for every person in the database, and matching can be implemented in a parallel computer to achieve almost real time recognition.

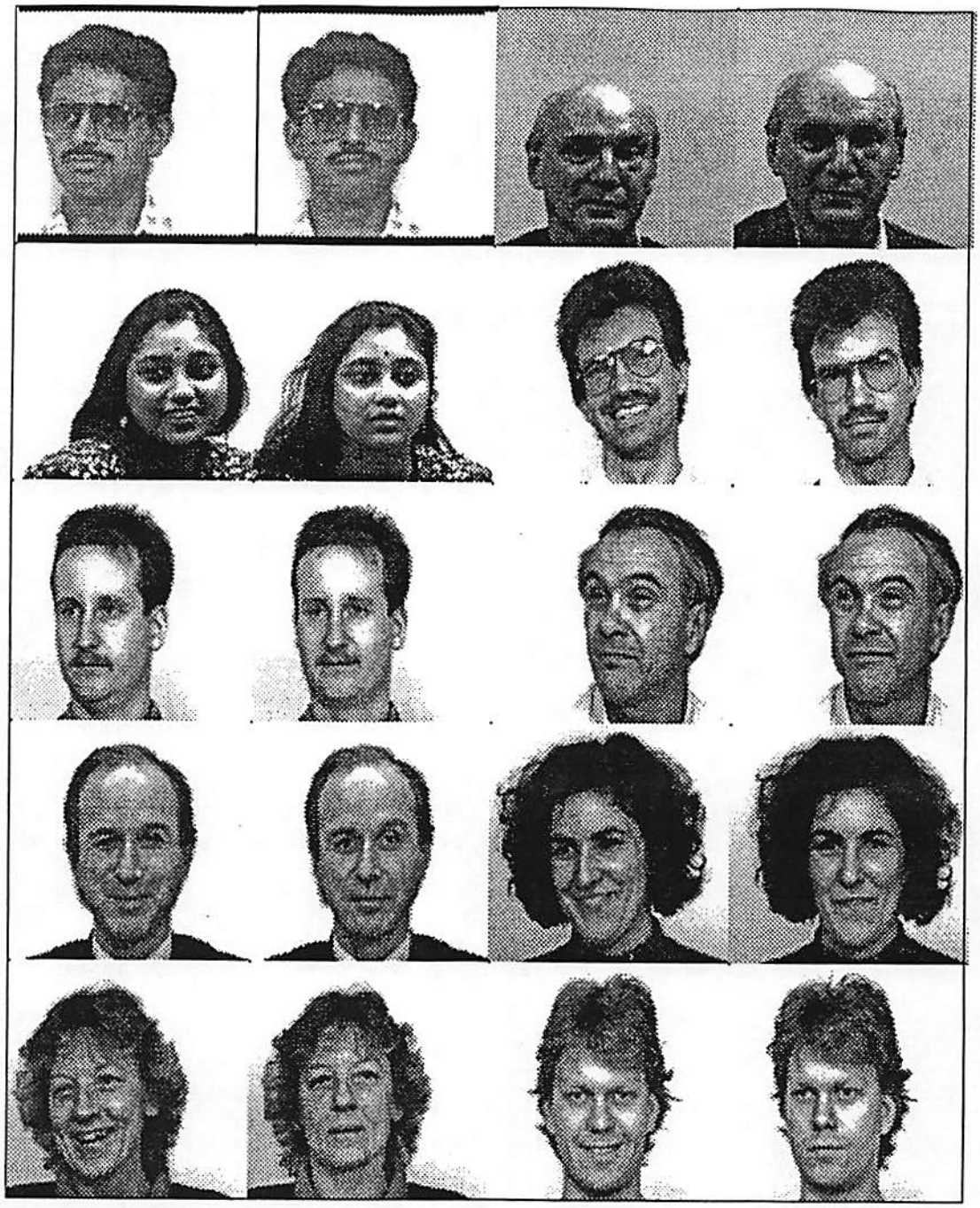


Figure 5.9: First and the third columns show input images to the recognition system. The database has over 300 images, and the second and fourth columns show the stored images in the database that were found closest to the input images.

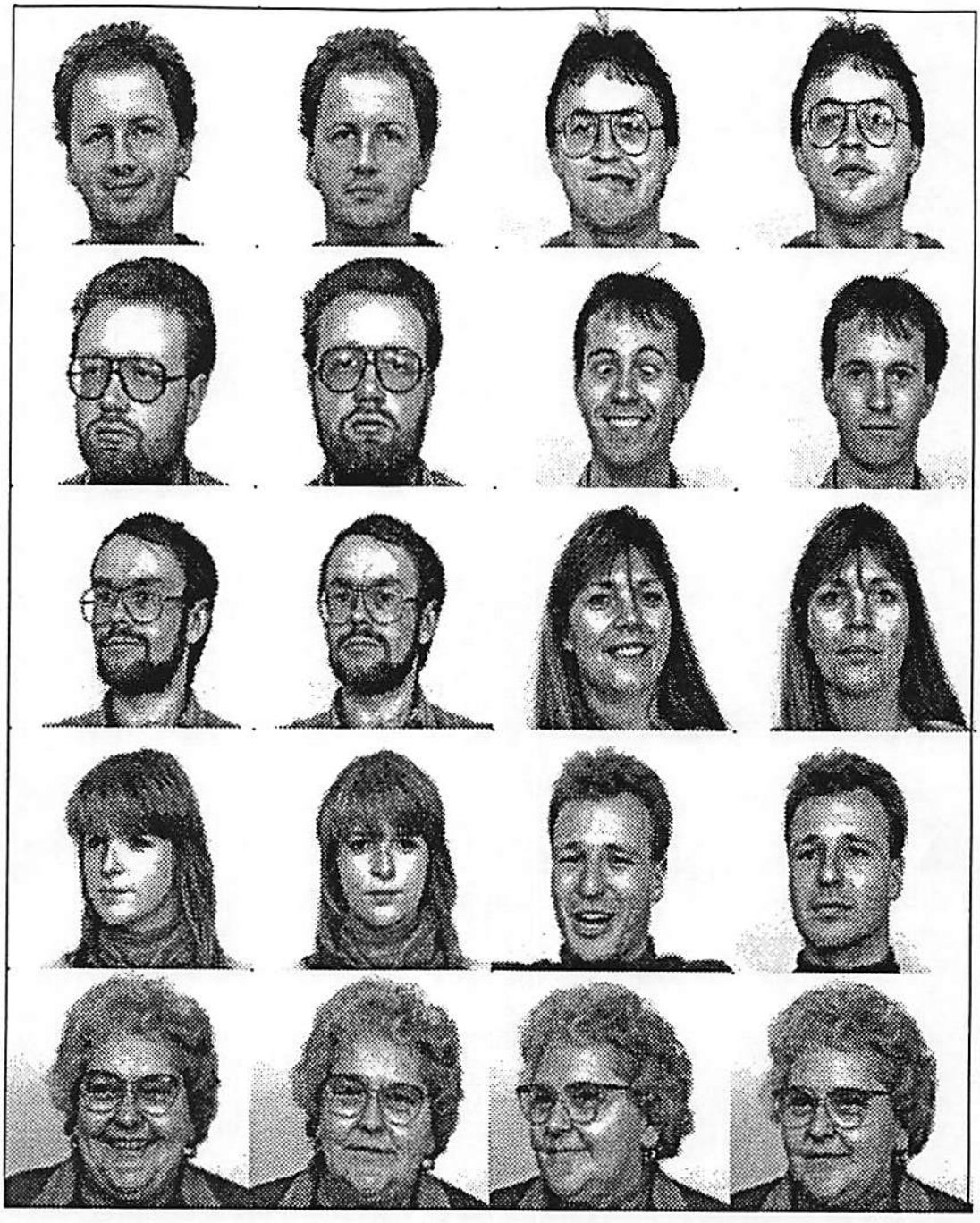


Figure 5.9: (continued).

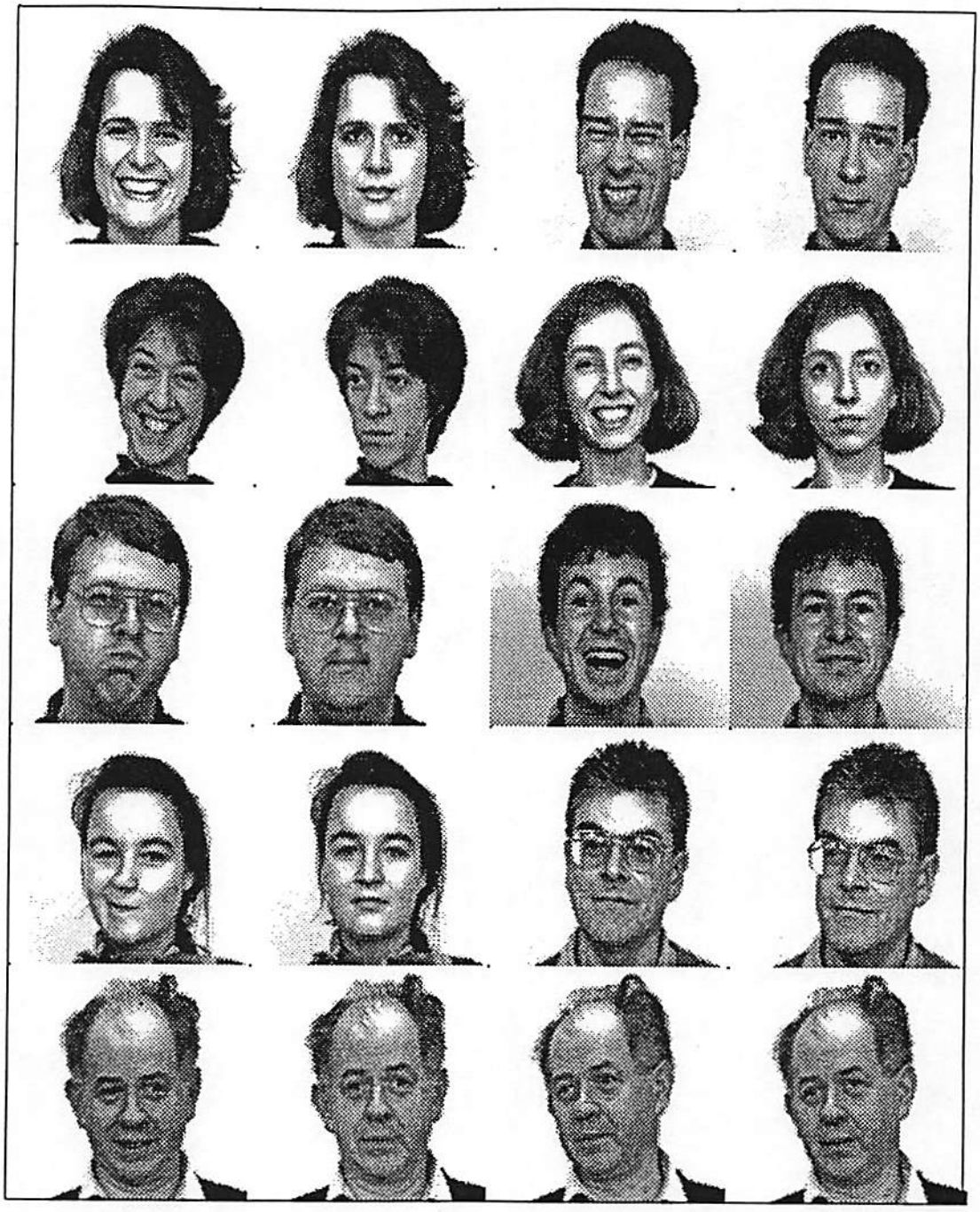


Figure 5.9: (continued).

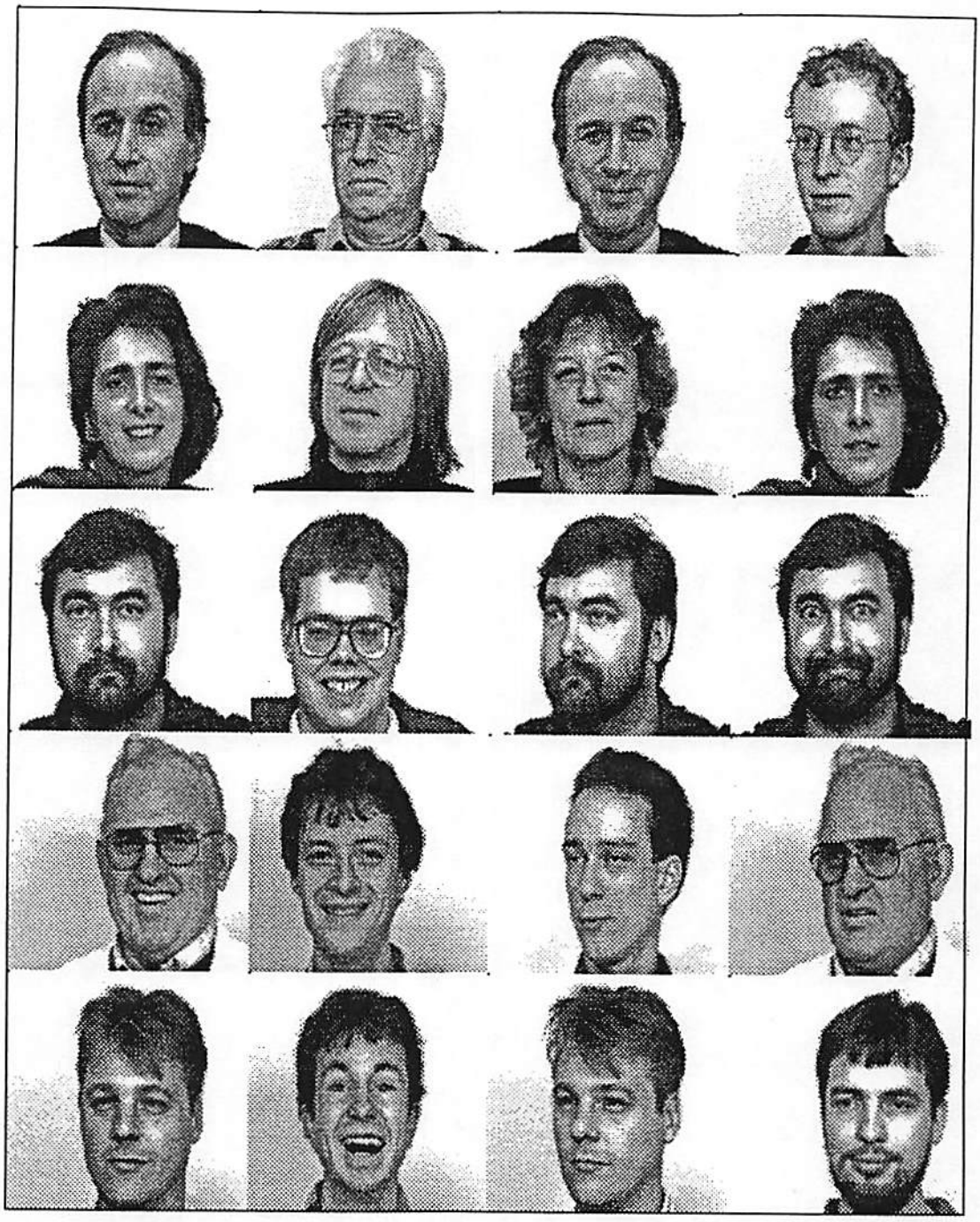


Figure 5.10: Here we show some cases where the best match was not the correct one, but the correct match was in the top three matches. First column shows the input face image, and second through fourth columns show the top three matches, from the best to the third best, respectively

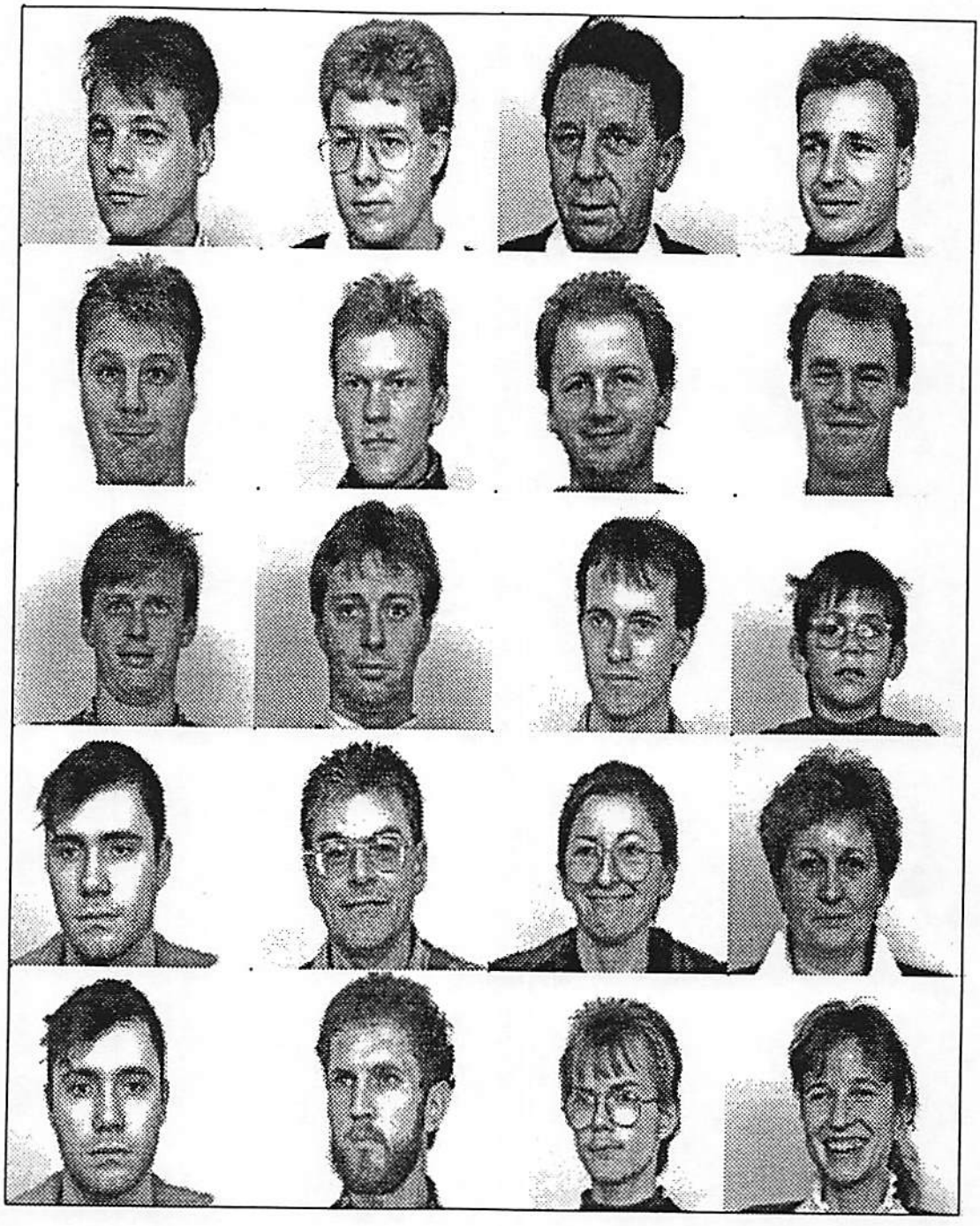


Figure 5.11: Some complete failures. Again as in the previous figure, the first column shows the input, and the following three columns show the top three matches

Number of persons	86
Number of Images in the database	303
Number of correct matches	261
Number of correct matches in top 3	286

Table 5.1: Statistics of the face recognition system: The success rate is 86% for the best match being correct, and 94% for the correct match being in the top 3 candidates.

5.6 Discussions

We described in this chapter a very simple scheme for representing and recognizing shapes, and illustrated the performance in recognizing human faces. Recognition involves two fundamental steps: creating an efficient representation of the input data, and then identifying this representation with a stored one. From a *neural* perspective, these two steps correspond to (a) creating a Short Term Memory (STM) representation and (b) identifying the STM activities with the Long Term Memory (LTM) states, respectively. In the literature, these two problems have been treated more or less independently, and probably is the main reason for the deficiencies of current *neural* techniques, which we discussed in section 2.3.1. The lack of flexible structure to represent syntactic bindings make them unsuitable for recognition applications. For example, consider the following extension of the face recognition problem. In addition to identifying the face, the system has to decide whether is the person is smiling or not (for simplicity assume that there are no other facial distortions). The only way this can be done in a standard associative memory is to add additional *neurons* to the system for *each* person to represent this conjunction of events. The problem gets more complicated if one wishes to have more than one face in a given scene. In order to overcome this fundamental problem, the dynamic link architecture was introduced by von der Malsburg [97]. The main difference between this architecture and other associative memory models is the way in which data is organized. In the dynamic link architecture temporal correlations between neurons play a very important role, in addition to the neuronal activities themselves, and help

in representing syntactical bindings. An excellent discussion on this and other advantages of DLA can be found in [97, 99, 98].

Although a biologically feasible and computationally efficient system to implement the principles of dynamic links is yet to be designed, it has helped in the development of useful alternate computer implementations. One such implementation for face recognition can be found in [10], and has influenced to a considerable extent our own graph matching algorithm. The main advantages of our scheme is that it avoids a stochastic search, and for this reason is extremely fast. It also is more efficient in terms of memory requirements. The reasons for these two advantages can be traced to our automatic construction of the feature graphs, unlike the ones considered in [10] where the spatial positions of the graph nodes are fixed. Further, the features used are also completely different: We have used higher order features (the curvature discontinuities) as opposed to the wavelets themselves (which represent information about local edges in the image). Possible extensions of this include using features at multiple scales, which we believe would improve the recognition process further.

Chapter 6

Discussions and Directions for Future Research

6.1 Summary

We have studied the use of neural networks in several grouping and segmentation applications. The first topic discussed is that of texture segmentation based on a Markovian model for the texture process. We developed a parallel deterministic relaxation network to compute the texture labels. In order to overcome the dependence of the network on initial state, we suggested the use of stochastic learning in the iterations of the network. The modified network combines the speed of deterministic relaxation with the sustained exploration of search space characteristic of the stochastic algorithms. The classification/segmentation results compare favorably with techniques such as simulated annealing. Our algorithms have found useful applications in segmenting synthetic aperture radar images [90]. We also discussed extensions to unsupervised segmentation involving parameter estimation and segmentation. Our conclusion is that it is often advantageous to separate estimation from segmentation, in contrast to some recent studies [64]. Another important observation is that a careful choice of the model, based on reasonable assumptions about the nature of the available data will help significantly in developing appropriate algorithms, as is clearly demonstrated in section 3.8.1.

While model based approaches offer quite accurate results, they do not provide a general theory for texture perception. Very often a natural scene contains various types of discontinuities. A problem of considerable interest is to provide a unified framework for detecting such boundaries. We have suggested one such approach to solving this problem in Chapter 4. Both intensity discontinuities and texture discontinuities are treated in this framework. Several examples of boundary detection, including natural as well as synthetic textures are provided. One of the important contributions is in clearly identifying the role of hypercomplex cells in texture boundary perception, besides providing a model for their receptive field profiles using local scale interactions.

A simple face recognition system is developed based on the observation that from the very early stages in visual processing, organization of features such as local curvature can help in developing robust representation mechanisms. Hypercomplex cells are useful in localizing curvature information, and dynamic binding of these cells activities results in recognition. Our work on face recognition is based on representing information about face images using the activities of these cells and formulating the recognition problem as an inexact graph matching process involving identifying an input representation with a stored representation. This also further illustrates previous observations by other researchers, notably Lowe [66] that perceptual organization of features extracted from 2-D images can be an alternative to elaborate 3-D reconstructions. The role of hypercomplex cells in perceiving illusory contours also can be attributed to their role in shape representation.

We briefly touched upon the topic of illusory contours in Chapter 4. Visual illusions are a rich source of information regarding the underlying structure of the visual cortex [87]. von der Heydt and Peterhans [96, 85] have demonstrated the role of hypercomplex cells in perceiving illusory contours. Hypercomplex cells form the first stage in shape representation in the visual cortex. They act as local curvature detectors and appropriate bindings of the activities of these cells help in a robust representation of shape from the very early stages itself. Thus it comes as no surprise that they are also involved in the perception of illusory contours, as such perceptions are a direct consequence of the dynamics governing the feature

binding process. Here again prior knowledge in the form of built-in assumptions plays an important role. Biological mechanisms which have evolved over millions of years make efficient use of non-accidental relationships between features in their hardware structure in order to process visual signals in real time. For example, abrupt discontinuities in images are either due to surface discontinuities, or due to occlusion. Our visual systems have developed appropriate mechanisms to handle such cases. Thus strong activations of hypercomplex cells at line ends might signal the percept of a boundary in an orthogonal direction. When such line endings are sufficiently close, the grouping of these activities can occur during the early stages itself as is demonstrated in our model in Chapter 4. It is of interest to note that people belonging to the south african Zulu culture do not perceive many of the illusions that are associated with sharp corners and edges, as they are brought up in an environment consisting mostly of circular or smoothly shaped objects [41]. The role of knowledge in the perception of illusory contours is further illustrated in Figure 6.1.

6.2 Directions for Further Research

Integrating different sources of information

Very often it is necessary to combine information from different visual cues in order to arrive at reliable and robust interpretations of the data. It is only recently that this problem is receiving attention [36, 107]. Coupled Markov random fields provide convenient tools for achieving this, and together with the homogeneous architecture of neural networks, it might be possible to develop parallel algorithms for integrating information from cues such as shape from shading, shape from stereo etc.

Robust feature detection

In many problems such as depth from stereo, a main issue is to extract reliable features for obtaining correspondence. The scheme that we described in Chapter 5 based on a model for the response of hypercomplex cells appears to be quite robust, as illustrated in our applications to motion correspondence and image

Interactions between knowledge and feature detection

Vision is an active process, and prior knowledge, context and goal all play critical roles. Our discussion on boundary detection was data driven, with no consideration given to the appropriate choice of scales. Here active interactions with contextual knowledge is necessary, and this remains as one of the open issues in computer vision. The homogeneous architecture of neural networks should prove useful in developing such interactions using active feedback.

How to realize the concept of dynamic links ?

The use of dynamic links [97] in the neural architecture might be a solution to many of the existing problems in using associative memory to vision applications. They provide a natural basis for feature binding, and hence incorporating into the dynamics the role of context and prior knowledge. Since dynamic binding eliminates the need for more neurons to represent syntactic structure, it also reduces the learning time, which is a serious problem with most neural architectures. However, as pointed out in our discussions in Chapter 5, many of these advantages of this architecture, although very promising, are yet to be fully realized.

The computational paradigm of Marr has dominated research in vision for over a decade now. However, despite the progress that has been made during this time, most of the fundamental problems in vision still remain unsolved. Hence it comes as no surprise that more researchers are turning towards biology in the hope of getting some clues. But, as Ramachandran [88] points out in support of his utilitarian theory of vision, nature had millions of years to experiment and design its own mechanisms to solve the problems. The mechanisms evolved under a different set of constraints, with a more goal oriented approach. The recent trend in vision, towards action oriented perception, is in this spirit. There is a shift from general paradigms to solving specific problems, to a more purposive approach, with the emphasis on real time computations. Neural networks will play a key role from the perspective of parallel computations, and by serving as a natural medium for interactions between researchers in computer and human vision. In this thesis we have explored the utility of both these aspects of neural

registration, and can be used in various other applications requiring detection of consistent set of features. In fact the image registration problem [108] is a much more difficult problem than one normally encountered in matching features in a stereo pair.

Another problem of considerable interest in semiconductor industry is in automating the defect detection process in semiconductor wafers. Except for highly regular circuits such as in memory chips (where one can use techniques such as those derived from fourier analysis), automating the process has proved to be quite difficult. The currently available alternatives include matching with templates, an extremely time consuming process requiring precise alignment of patterns, or manual inspection. Some preliminary studies have been made regarding the suitability of neural networks for this application [69], where intensity patterns around a defective region are used in training the networks. A better way to approach this problem is to detect features representative of such defects, and by proper choice of parameters, it is possible to develop a system based on our feature detection algorithm. Since the system can be easily fabricated on analog networks using VLSI techniques, real time defect detection is quite possible.

Face recognition

The approach to face recognition presented in Chapter 5 is based on low level features. One straight forward extension is to consider multiple scale features, and perform the matching hierarchically. This would improve the performance of the system, both in terms of speed and accuracy.

It is also useful to characterize the faces in terms of facial features such as eyes, nose, mouth etc.. For example it might be of interest to derive observations such as whether a person is smiling or not, or if he has a long nose. This requires development of a hierarchical feature extraction system which can group or bind features at successive stages. One can a teach the network to develop appropriate bindings of the activities of hypercomplex cells to represent various facial features.

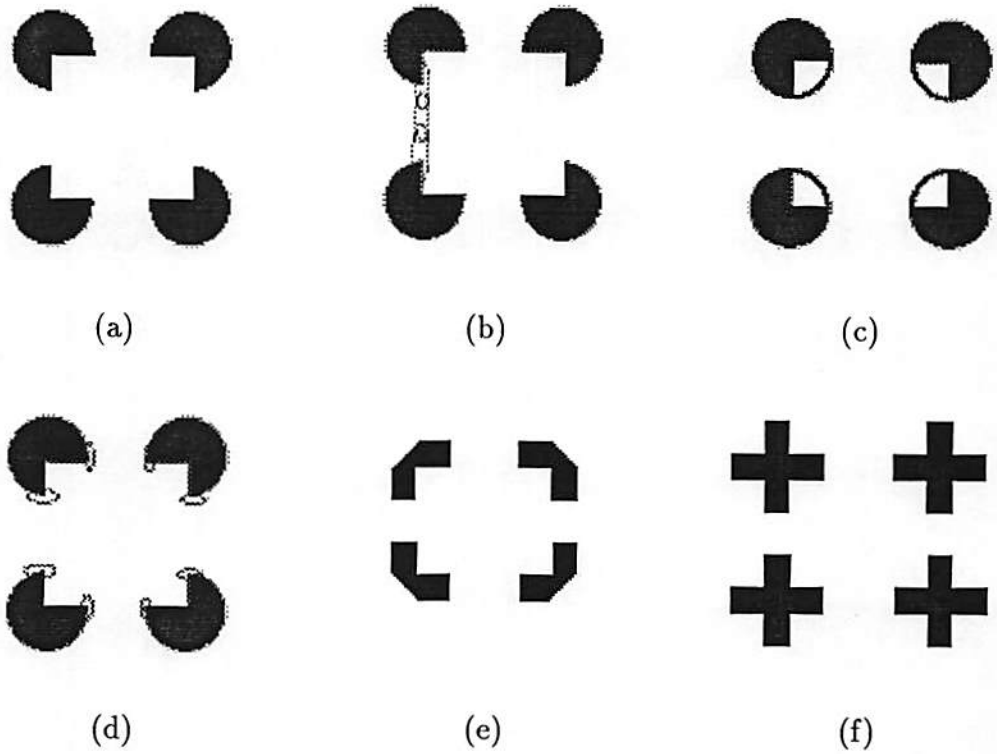


Figure 6.1: (a) Kanisza's illusory square formed by four pac-man figures. Our visual system discounts the accidental alignments of the pac-man figures to infer closed circles partly occluded by a white square, (b) Grossberg's model for detecting the square using overlapping receptive fields sending positive feedback to lower grouping mechanisms to complete the boundaries of the square, (c) completing the pac-man's contours should not affect the boundary completion process in (b), but the illusion is lost [85], (d) von der Heydt and Peterhans propose an alternate model based on the role of hypercomplex cells, grouping of whose activities help in perceiving the contour, (e)-(f) illustrate the role of prior knowledge of the figures in the perception of illusions (adopted from Kanisza [55]). Although the perception of the illusory square in (e) is as not strong as in (a), it is still more perceivable than in (f), where we tend to see the completed figures of crosses. Note that the interior of the square is not changes, hence the responses of the hypercomplex cells will be the same in both cases. However, the grouping is done differently, with the preference given to completed figures in (f), rather than to the illusion of the square as in (e).

computation in the context of some early vision problems. There is much potential for the use of these networks in tasks such as invariant object recognition and associative memory, and in developing adaptive vision systems.

References

[1] D. Amit, *Modeling Brain Function*, Cambridge: Cambridge University Press, 1989.

[2] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, pp. 2–14, January 1986.

[3] J. Beck, K. Prazdny, and A. Rosenfeld, "A Theory of textural segmentation," in *Human and Machine Vision* (J. Beck, B. Hope, and A. Rosenfeld, eds.), pp. 1–38, Academic Press, 1983.

[4] J. R. Bergen and E. H. Adelson, "Early vision and texture perception," *Nature*, Vol. 333, pp. 363–364, May 1988.

[5] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Statist. Soc. B*, Vol. 48, pp. 259–302, 1986.

[6] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *Computing Surveys*, Vol. 17, pp. 75–145, March 1985.

[7] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, Vol. 94, pp. 115–147, 1987.

[8] J. Bolz and C. D. Gilbert, "Generation of end-inhibition in the visual cortex via interlaminar connections," *Nature*, Vol. 320, pp. 362–365, March 1986.

[9] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-12, pp. 55–73, January 1990.

[10] J. Buhmann, J. Lange, and C. von der Malsburg, "Distortion invariant object recognition by matching hierarchically labelled graphs," in *Proc. Int. Joint Conf. on Neural Networks*, vol. 1, (Washington D.C.), pp. 155–159, July 1989.

[11] D. C. Burr and M. C. Morrone, "Feature detection in biological and artificial visual systems," in *Vision: coding and efficiency* (C. Blakemore, ed.), Cambridge University Press, 1990.

[12] P. J. Burt, "The Pyramid as a structure for efficient computation," in *Multiresolution image processing and analysis* (A. Rosenfeld, ed.), pp. 6–35, Springer-Verlag, 1984.

[13] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, Vol. COM-31, pp. 532–540, April 1983.

[14] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, pp. 679–698, November 1986.

[15] G. Carpenter and S. Grossberg, "A Massively parallel architecture for a self-organizing neuaral pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, pp. 54–115, January 1987.

[16] G. Carpenter, S. Grossberg, and C. Mehanian, "Invariant recognition of cluttered scenes by a self-organizing ART architecture :CORT-X boundary segmentation," *Neural Networks*, pp. 169–181, 1989.

[17] S. Chandrashekhar, *Motion analysis and passive navigation using long image sequences*, Ph.D. dissertation, University of Southern California, Department of Electrical Engineering, July 1991.

[18] S. Chandrashekhar and R. Chellappa, "Passive navigation in a partially known environment," in *Proc. of Workshop on Visual Motion*, (Princeton, NJ), October 1991. Accepted for presentation.

[19] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition 2*, (Ed. L.N. Kanal and A. Rosenfeld, Elsevier Science Publishers, North-Holland), pp. 79–112, 1985.

[20] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian-Markov random fields," *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-33, pp. 959–963, August 1985.

[21] F. Cohen and D. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian fields," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-9, pp. 195–219, March 1987.

[22] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. Syst. Man Cybern.*, pp. 815–825, September 1983.

[23] P. R. Cooper, "Parallel object recognition from structure," Tech. Rep. 301, Computer Science Department, University of Rochester, July 1989.

[24] G. Cross and A. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 25–39, January 1983.

[25] J. G. Daugman, "Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, Vol. 2, pp. 1160–1169, 1985.

[26] J. G. Daugman, "Relaxation neural network for non-orthogonal image transforms," in *Proc. Int. Conf. on Neural Networks*, vol. 1, (San Diego, CA), pp. 547–560, June 1988.

[27] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-9, pp. 39–55, January 1987.

[28] A. Dobbins, S. W. Zucker, and M. S. Cynader, "Endstopped neurons in the visual cortex as a substrate for calculating curvature," *Nature*, Vol. 329, pp. 438–441, October 1987.

[29] E.Goles and G. Vichniac, "Lyapunov Function For Parallel Neural Networks," in *Neural Networks for Computing*, (Ed. John S. Denker,Snowbird, UT), 1986.

[30] K. B. Eom and R. Kashyap, "Composite Edge Detection with Random Field Models," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC-20, pp. 81–93, Jan 1990.

[31] T. J. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-D objects using surface descriptions," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, pp. 1140–1157, 1989.

[32] A. Fuchs and H. Haken, "Pattern recognition and associative memory as dynamical processes in a synergetic system I," *Biological Cybernetics*, Vol. 60, pp. 17–22, 1988.

[33] A. Fuchs and H. Haken, "Pattern recognition and associative memory as dynamical processes in a synergetic system II," *Biological Cybernetics*, Vol. 60, pp. 107–109, 1988.

[34] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, Vol. 36, pp. 193–202, 1980.

[35] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, Vol. 1, pp. 119–130, 1988.

[36] E. Gamble, D. Geiger, T. Poggio, and D. Weinshall, "Integration of vision modules and labelling of surface discontinuities," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC-19, pp. 1576–1581, November 1989.

[37] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRF's: Surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-13, pp. 401–411, May 1991.

[38] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-12, pp. 609–628, July 1990.

[39] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, pp. 721–741, November 1984.

[40] S. Geman and C. Graffigne, "Markov random fields image models and their application to computer vision," in *Proc. of the Int. Congress of Mathematicians 1986*, (Ed. A.M. Gleason, American Mathematical Society ,Providence), 1987.

[41] R. L. Gregory, *Eye and Brain: The Psychology of seeing*, Princeton, New Jersey: Princeton University Press, 1990.

[42] S. Grossberg and E. Mingolla, "Neural dynamics of surface perception: Boundary webs, illuminants, and shape-from-shading," *Computer Vision, Graphics, and Image Processing*, pp. 116–165, January 1987.

[43] N. M. Grzywacx and A. L. Yuille, "Motion correspondence and analog networks," in *Proc. of Neural Networks for Computing*, (Ed. J. Denker, Snowbird, Utah), pp. 200–205, 1986.

[44] D. Heeger, "Optical Flow from Spatiotemporal Filters," in *Proc. Int. Conf. Computer Vision*, (London,England), pp. 181–190, June 1987.

[45] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, Vol. 79, pp. 2554–2558, April 1982.

[46] J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci.*, Vol. 81, pp. 3088–3092, May 1984.

[47] J. Hopfield and D. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, Vol. 52, pp. 114–152, 1985.

[48] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, Vol. 160, pp. 106–154, January 1962.

[49] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas(18 and 19) of the cat," *Journal of Neurophysiology*, Vol. 28, pp. 229–289, March 1965.

[50] D. H. Hubel and T. N. Wiesel, "Functional architecture of macaque monkey visual cortex," *Proceedings of Royal Society of London (B)*, Vol. 198, pp. 1–59, 1977.

[51] J. Hutchinson and C. Koch, "Simple analog and hybrid networks for surface interpolation," in *Proc. of Neural Networks for Computing*, (Ed. J. Denker, Snowbird, Utah), pp. 235–240, April 1986.

[52] B. Julesz, "Textons, the elements of texture perception and their interactions," *Nature*, Vol. 290, pp. 91–97, March 1981.

[53] B. Julesz and B. Krose, "Features and spatial filters," *Nature*, Vol. 333, pp. 302–303, May 1988.

[54] T. Kanade, *Picture processing system by computer complex and recognition of human faces*, Ph.D. dissertation, Kyoto University, Department of Information Science, November 1973.

[55] G. Kanisza, "Subjective contours," *Scientific American*, Vol. 234, pp. 48–52, April 1976.

[56] R. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial interaction models of images," *IEEE Trans. Information Theory*, Vol. IT-29, pp. 60–72, January 1983.

[57] R. Kinderman and J. L. Snell, *Markov Random Fields and their applications*, Providence: American Mathematical Society, 1980.

[58] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, pp. 95–102, December 1982.

[59] C. Koch, J. Luo, C. Mead, and J. Hutchinson, "Computation motion using resistive networks," in *Proc. of Neural Information Processing Systems*, (Ed. D.Z. Anderson, Denver, Colorado), 1987.

[60] T. Kohonen, "Content addressable memory," *IEEE Trans. Computers*, Vol. C-21, pp. 353–359, April 1972.

[61] T. Kohonen, *Self-Organization and associative memory*, New York: Springer-Verlag, 1989.

[62] B. Kosko, "Adaptive bidirectional associative memories," *Applied Optics*, Vol. 26, pp. 4947–4960, December 1987.

[63] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. Computers*, 1991. submitted for publication.

[64] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, pp. 799–813, August 1989.

[65] S. Lakshmivarahan, *Learning algorithms theory and applications*, New York: Springer-Verlag, 1981.

[66] D. G. Lowe, *Perceptual organization and visual recognition*, Massachusetts: Kulwer academic publishers, 1985.

[67] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Opt. Soc. Am. A*, Vol. 7, pp. 923–932, May 1990.

[68] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, pp. 674–693, July 1989.

[69] B. Manjunath, P. B. Chou, and R. S. Jaffe, "Optical detection of defects using artificial neural networks." Unpublished IBM internal report, August 1990.

[70] B. S. Manjunath and R. Chellappa, "A computational model for boundary detection," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Maui, Hawaii), pp. 358–363, June 1991.

[71] B. S. Manjunath, T. Simchony, and R. Chellappa, "Stochastic and deterministic networks for texture segmentation," *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-38, pp. 1039–1049, June 1990.

[72] S. Marcelja, "Mathematical description of the responses of simple cortical cells," *J. Opt. Soc. Am.*, Vol. 70, pp. 1297–1300, November 1980.

[73] D. Marr, *Vision*, San Francisco, CA: W. H. Freeman, 1982.

[74] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of Royal Society of London (B)*, pp. 187–217, 1980.

[75] J. Marroquin, *Probabilistic solution of inverse problems*, Ph.D. dissertation, M.I.T, Artificial Intelligence Laboratory, September 1985.

[76] E. D. Micheli, B. Caprile, P. Ottonello, and V. Torre, "Localization and noise in edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, pp. 1106–1117, October 1989.

[77] E. Mjolsness, G. Gindi, and P. Anandan, "Optimization in model matching and perceptual organization," *Neural Computation*, Vol. 1, pp. 218–229, 1989.

[78] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proc. 5th Int. Joint Conf. Artificial Intell.*, (Cambridge, MA), p. 584, August 1977.

[79] M. C. Morrone and D. C. Burr, "Feature detection in human vision: a phase dependent energy model," *Proceedings of Royal Society of London (B)*, Vol. 235, pp. 221–245, 1988.

[80] K. Narendra and M. Thathachar, *Learning Automata*, New York: Prentice-Hall, 1989.

[81] M. Oshima and Y. Shirai, "Object recognition using three-dimensional information," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 353–361, July 1983.

[82] S. E. Palmer, "The psychology of perceptual organization," in *Human and Machine Vision* (J. Beck, B. Hope, and A. Rosenfeld, eds.), pp. 269–339, Academic Press, 1983.

[83] P. Perona and J. Malik, "Detecting and localizing edges composed of steps, peaks and roofs," in *Proc. Intl. Conf. Computer Vision 90*, (Tokyo,Japan), pp. 52–57, December 1990.

[84] D. I. Perrett, A. J. Mistlin, and A. J. Chitty, "Visual neurones responsive to faces," *Trends in Neural Sciences*, Vol. 10, pp. 358–364, 1987.

[85] E. Peterhans and R. von der Heydt, "Mechanisms of contour perception in monkey visual cortex. II. Contour bridging gaps," *Journal of Neuroscience*, Vol. 9, pp. 1749–1763, May 1989.

[86] M. Porat and Y. A. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-10, pp. 452–468, July 1988.

[87] V. S. Ramachandran, "Capture of stereopsis and apparent motion by illusory contours," *Perception and Psychophysics*, Vol. 39, pp. 361–373, 1986.

[88] V. S. Ramachandran, "Interactions between motion, depth, color and form: the utilitarian theory of perception," in *Vision: coding and efficiency* (C. Blakemore, ed.), Cambridge University Press, 1990.

[89] A. Rangarajan, R. Chellappa, and B. S. Manjunath, "Markov random fields and neural networks with applications to early vision problems," in *Artificial neural networks and statistical pattern recognition* (I. Sethi, ed.), Elsevier Science, 1991.

[90] E. Rignot and R. Chellappa, "Segmentation of synthetic aperture radar complex data," *J. Opt. Soc. Am. A*, September 1991.

[91] C. Ronse, "A twofold model of edge and feature detection." pre-print, September 1990.

[92] T. Simchony, R. Chellappa, and Z. Lichtenstein, "Relaxation algorithms for MAP estimation of grey level images with multiplicative noise," *IEEE Trans. Information Theory*, Vol. IT-36, pp. 608–613, May 1990.

[93] M. Thathachar and P. Sastry, "Relaxation labelling with learning automata," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, pp. 256–268, March 1986.

[94] A. Treisman, "Preattentive processing in vision," *Computer vision and Image processing*, Vol. 31, pp. 156–177, August 1985.

[95] M. Turk and A. Pentland, "Face recognition using Eigen faces," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Maui, Hawaii), pp. 586–591, June 1991.

[96] R. von der Heydt and E. Peterhans, "Mechanisms of contour perception in monkey visual cortex. I. Lines of pattern discontinuity," *Journal of Neuroscience*, Vol. 9, pp. 1731–1748, May 1989.

[97] C. von der Malsburg, "The correlation theory of brain function," Tech. Rep. 81-2, Department of Neurobiology, Max-Planck-Institute for Biophysical chemistry, 1981.

[98] C. von der Malsburg, "Pattern recognition by labelled graph matching," *Neural Networks*, pp. 141–148, 1988.

[99] C. von der Malsburg and E. Bienenstock, "Statistical coding and short-term synaptic plasticity: A scheme for knowledge representation in the brain," in *Disordered systems and biological organization* (E. B. et al, ed.), pp. 61–70, Springer-Verlag, 1986.

[100] C. von der Malsburg and E. Bienenstock, "A neural network for invariant pattern recognition," *Europhysics Letters*, pp. 121–126, July 1987.

[101] C. von der Malsburg and E. Bienenstock, "A neural network for the retrieval of superimposed connection patterns," *Europhysics Letters*, pp. 1243–1249, June 1987.

[102] H. Voorhees and T. Poggio, "Computing texture boundaries from images," *Nature*, Vol. 333, pp. 364–367, May 1988.

[103] H. Wechsler and L. Zimmerman, "2D invariant object recognition using distributed associative memories," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-10, pp. 811–821, 1988.

[104] R. Wheeler, Jr., and K. Narendra, "Decentralized learning in finite Markov chains," *IEEE Trans. Automatic Control*, Vol. AC-31, pp. 519–526, June 1986.

[105] A. Witkin, "Scale-space filtering," in *Int. Joint Conf. Artificial Intelligence*, (Karlsruhe, West Germany), pp. 1019–1021, 1983.

[106] C. S. Won and H. Derin, "Unsupervised image segmentation using Markov random fields - part I: Noisy images." preprint.

[107] Q. Zheng, *Robust algorithms for estimation of illuminant and shape from shading*, Ph.D. dissertation, University of Southern California, Department Electrical Engineering, July 1991.

[108] Q. Zheng, R. Chellappa, and B. S. Manjunath, "Balloon motion estimation using two frames," in *Proc. 25th Asilomar Conf. on Systems and Signals*, November 1991. (invited paper).

[109] Y. T. Zhou, *Artificial neural network algorithms for some computer vision problems*, Ph.D. dissertation, Univ. of Southern California, Electrical Engineering-Systems, June 1989.