

NEURAL NETWORK ARCHITECTURES FOR PREATTENTIVE VISUAL PROCESSING

A THESIS

submitted for the award of the degree

of

MASTER OF SCIENCE

in

COMPUTER SCIENCE AND ENGINEERING

by

A. ARUL VALAN



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

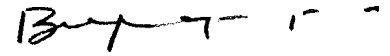
INDIAN INSTITUTE OF TECHNOLOGY

MADRAS-600 036, INDIA

MARCH 1993

CERTIFICATE

This is to certify that the thesis entitled "**NEURAL NETWORK ARCHITECTURES FOR PREATTENTIVE VISUAL PROCESSING**" is the bonafide work of Mr. **A.Arul Valan**, carried out under my guidance and supervision, in the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, for the award of the degree of Master of Science in Computer Science and Engineering.



(B.Yegnanarayana)

ACKNOWLEDGEMENTS

I thank my research guide **Prof. B.Yegnanarayana** for introducing me into the area of Neural Networks. I have learnt from him not only the attitude and aptitude to do research but also a balanced viewpoint. I have also learnt from him the essentials of a professional life: systematic ways of writing, thinking, planning and working towards a specific goal.

Ravi has taught me the fundamental tools for thinking through a number of discussions. I thank him for teaching me how to plan during crisis. **Ramaseshan** has given me moral support and advice in abundant measure. I thank him for being with me during some of my difficult times in research. I thank **Rajendran, Uday, Ramachandran** and **Madhukumar** for their free help and advice in the course of my research. I thank **Dr. H.M.Chouhan** for giving me enthusiasm and helping me to direct my research efforts. I thank **Hema Murthy, Chandra Sekhar, Sundar, Ramana Rao** and **Alwar** for many useful discussions.

I thank my friends **Sanjay, Srikanth, Sreenivasan, Sriram, Sundar, Sethu, Shankar, Arun, Indu, Poongodi, Sudha** and **Raghu**, who made my stay at IIT enjoyable.

I thank my parents for providing continuous **moral** support throughout the course **of** research.

CONTENTS

ABSTRACT

Chapter 1: INTRODUCTION	1
1.1 Motivation for visual pattern recognition research	1
1.2 Pattern recognition approach	3
1.3 Neural networks approach to pattern recognition	4
1.4 Overview of the thesis	6
1.4.1 Objective of current research	6
1.4.2 Motivation for current research	7
1.4.3 Scope of the work	7
1.4.4 Overview of the research	8
1.4.5 Organization of the thesis	8
 Chapter 2: APPROACHES FOR PREATTENTIVE VISUAL PROCESSING	 9
2.1 Introduction	9
2.2 Background	9
2.2.1 Pattern recognition approaches	11
2.2.2 Computational framework for visual processing	13
2.2.2.1 Low level versus high level processing	13
2.2.2.2 Serial versus parallel processing	14
2.2.2.3 Automatic versus selective processing	14
2.2.2.4 Signal versus symbols	15
2.2.3 Eye movements and visual pattern perception	15
2.2.4 Results from neurophysiology for visual processing	16
2.3 Review of neural network architecture and principles for visual pattern recognition	18

2.3.1	Neocognitron: An architecture for visual pattern recognition	19
2.4	Preattentive Visual Processing: Issues and approaches	21
2.5	Summary	24

Chapter 3: ORIENTED FILTERING AND INTEGRATION NETWORK FOR STRUCTURAL FEATURE EXTRACTION 25

3.1	Introduction	25
3.2	Oriented filtering and integration network(ORFIN)	25
3.2.1	Design of oriented filtering network	27
3.2.2	Design of integration network	32
3.3	Application of ORFIN for isolated word recognition	35
3.3.1	Design of isolated word recognition system	39
3.3.2	Data preparation	47
3.3.3	Implementation details and results	49
3.4	Summary	54

Chapter 4: DIRECTED SPREADING ACTIVATION LAYERS FOR LOCATING MAXIMUM INFORMATION POINTS 55

4.1	Introduction	55
4.2	Spreading activation layers	56
4.2.1	Activity diffusion and centroid detection	57
4.2.2	Feature extraction in spreading activation layers	59
4.2.3	Centers of focus of attention	62
4.3	Motivation for directed spreading	63
4.3.1	Drawbacks of the spreading activation layers for low level feature extraction	63
4.3.2	Basis for directed spreading activation model	66
4.4	Directed spreading activation (DSA) layers	67

4.4.1	Organization of DSA layers	68
4.4.2	Design of DSA layers	70
4.5	Implementation details and examples	75
4.6	Application of DSA layers to low level feature extraction from machine fonts	80
4.7	Application of DSA layers to recognition of transformation invariant binary pattern recognition	83
4.8	Application of DSA layers to images of formant contour patterns	86
4.9	Summary	93
Chapter 5:	SUMMARY AND CONCLUSIONS	94
LIST OF FIGURES		97
LIST OF TABLES		100
PUBLICATION		100
REFERENCES		101

ABSTRACT

Visual pattern recognition such as reading handwritten characters or distinguishing shapes is easily accomplished by human beings. When attempted to design information processors to do the same, it presents significant difficulties. There have been two approaches for machine implementation of visual pattern recognition. The first approach considers vision as an abstract problem and attempts to design computational algorithms. The second approach attempts to study the biological visual system and model its behavior for engineering applications.

Artificial neural networks which are reminiscent of the neurons in the brain attempt modeling the function of the biological system. They are characterized by their nonsymbolic, distributed, fault tolerant computing which are very useful for pattern recognition tasks.

In visual pattern recognition there is a natural factoring part of the process that extract information about the geometry of the visual pattern and the process that recognizes the familiar objects. Preattentive visual processing is a parallel, automatic and data driven **processing** which extracts geometric properties of the input pattern without using the detailed knowledge of the domain. In this work we have attempted to develop neural network architectures for

automatic, data driven extraction of geometric properties like straight lines, corners and contour termination points from binary images. We also show how these architectures can be used in some engineering applications.

Based on the observations about some aspects of the visual perception in the biological visual system, we propose two approaches for processing binary images. In the first approach, the structural properties of the input pattern like straight lines are extracted from the input image. This approach is implemented as an *oriented filtering and integration network*, motivated by the orientation specificity shown by certain cells in the visual cortex.

In the second approach, the maximum information points of the binary image are located. In the case of simple geometric contours these points coincide with the points of maximum inflection. In this study, points of **maximum** information are obtained using *directed spreading activation layers*.

We describe two applications of these architectures. The first application is recognizing isolated utterances of words from images of formant contour patterns. For this we use oriented filtering and integration network. The second application deals with recognition of objects in binary images invariant to translation, rotation and scale. For this the directed spreading activation neural architecture is used to extract the maximum information points from the input pattern.

A log-polar transformation is described which derives an invariant representation from the maximum information points. This invariant representation can be used for recognition using standard methods.

INTRODUCTION

1.1 MOTIVATION FOR VISUAL PATTERN RECOGNITION RESEARCH

Since the advent of digital computer there has been an effort to expand the domain of computer applications. Some of the motivation for this effort comes from important practical needs to find more efficient ways of doing things. At present, the ability of machines to perceive their environment is very limited. A variety of transducers are available for converting light, sound, temperature etc., to electrical signals. When the environment is carefully controlled and the signals have a simple interpretation, as is the case with the standard computer devices, the perceptual problems become trivial. But as we move beyond having a computer read punch cards or magnetic tapes to having it read hand-printed characters or analyze biomedical photographs, we move from problems of sensing the data to much more difficult problems of interpreting the data. Of the various problem areas, the domain of visual pattern recognition has received by far the most attention.

There are three basic motivations for trying to achieve automatic recognition of visual patterns. The first is simply intellectual curiosity. How can machines be organized to designate a particular presentation as belonging to the same

class **that** a human would specify? This raises intriguing questions of systems analysis and design, and it leads to sharper appraisal of how living systems process information. The second purpose is to provide intelligent aids. There is great utility in machines which can process optical information more quickly or accurately or safely or cheaply than people. The automatic reading of postal addresses, classification of weather-satellite photographs and terrain maps, recognition of bubble chamber tracks, diagnosis of biological cells, and monitoring of cardiac **performance** can substantially relieve human drudgery and provide economic advantage. Still other uses are in prosthetic aid - for example, in reading and mobility devices for the blind. The third reason for developing machines which recognize optical patterns is to obtain more effective man-machine interfaces. It is becoming increasingly important to provide computers with fluency in **man's** natural languages. With more direct communication between man and machine, important gains in flexibility and efficiency can be obtained.

In Section 1.2 a pattern recognition approach to visual pattern recognition is discussed. In Section 1.3, advantages of neural networks approach to pattern recognition problems is discussed. In Section 1.3 an overview of the thesis is presented.

1.2 PATTERN RECOGNITION APPROACH

The term pattern recognition was introduced in the early 1960s, and it originally meant detection of simple visual patterns like handwritten characters, weather maps and speech spectra. Later the domain of application of pattern **recognition** is expanded to almost all disciplines of engineering and science. Of the various problem areas in pattern recognition research, the domain of visual pattern recognition has attracted much attention. Since the human experience of vision is effortless, quick and adaptable studies have been made on biological visual system. Neurophysiological and psychological studies have given us several **interesting** facts about visual perception. But no understanding has been sufficient to duplicate their **performance** by computer. This has resulted in a lack of complete theory of vision.

The lack of complete theory has not deterred people from attempting modest problems. Many of these involve pattern classification - the assignment of a physical object or event to one of several prespecified categories. Extensive study of classification problems has **led to** an abstract mathematical model that provides the theoretical basis for classifier design. Even though abstract mathematical model is available, in any specific application one ultimately must come to grips with the special characteristics of the problem in hand. These models are applied successfully to the recognition of

handwritten characters, chromosome types, printed characters, Chinese characters, aircraft, machine parts, circuit boards, maps, and lung radiographs.

1.3 NEURAL NETWORKS APPROACH TO PATTERN RECOGNITION

Though pattern recognition research focussed on solutions for modest problems, the ambitious objective has all the time been to implement artificial perception, that is, to imitate the functions of the biological sensory systems in their most complete forms. The first experiments around 1960 were indeed based on elementary neural networks, known by names like **perceptron**[44], **Adaline**[53] and **Learning Matrix**[50], respectively. But it was soon realized that the performance of the biological sensory system is very difficult to reach. Even high computing capacity, achievable by parallel computing circuits, did not solve the problems. For example, in image analysis there exists requirements which are very difficult to fulfill: Invariance of detection with respect to translation, rotation, scale, perspective, partial occlusion and modest marring of the objects.

Artificial neural networks are massively parallel interconnected networks of simple adaptive elements. These elements are arranged in a hierarchical manner to interact with the objects of the real world in the same way as biological neural systems do. These simple neuron like elements connected together show powerful learning,

memorization, associative recall capabilities and self organization for pattern formatted **information**[36]. Apart from these properties, they have number of other advantages. The computation is distributed, fault tolerant and has the ability to tolerate distortions in the input pattern. This neural network approach differs significantly from the earlier approaches by its nonsymbolic processing and distributed representation.

Since these neural networks are conceptually compatible with the biological neural networks it is possible to derive inspiration from neurobiological or psychological studies, even though the objective might be engineering. When the engineering model performance mirrors human performance, similar model **might** be applied to biological neural net and mutually useful hints can be obtained in this manner.

Neural network architectures are generally meant to learn and recognize the input patterns. **But there** are certain neural mechanisms in the initial stages of **animal** visual and auditory system. These neural mechanisms possess very little domain specific knowledge and essentially act as data adaptive filters. In this work we attempt to design such neural architectures for processing visual input patterns.

1.4 OVERVIEW OF THE THESIS

In this section we introduce the specific research problem addressed. In Section 1.4.1 discuss the objective of the thesis. Section 1.4.2 discusses the motivation of this work and Section 1.4.3 discusses the scope of the study. Section 1.4.4 presents the overview of research and Section 1.4.5 discusses the organization of the rest of the thesis.

1.4.1 Objective of Current Research

Visual pattern recognition can be considered as consisting of two stages: (i) A low level analysis concerning extraction of geometric properties of the input pattern and generation of a description of the **pattern**[32] and (ii) a higher level analysis which uses the description together with the knowledge of the domain to perform the recognition task. Our preattentive visual **processing**[14] is a parallel, automatic and data driven processing which extracts properties of the input pattern based on local data. Artificial neural networks, with their collective **nonsymbolic** computational capabilities, are useful to achieve the preattentive visual processing. The objective of this thesis is to develop neural architectures for automatic extraction of geometric properties like straight lines, corners and contour termination points from binary input image patterns. We also show how these architectures can be used in some engineering applications.

1.4.2 Motivation for Current Research

There are two different approaches for machine vision. The first approach is computational vision approach. In this approach vision is studied abstractly independent of any particular domain. Pattern recognition and Artificial Intelligence follow this approach and attempted to develop computational algorithms for vision. The other approach is to study the human visual system. Since the human vision is rapid and effortless, the objective had been to study human vision and design engineering models for practical applications. Here, reports from psychological and neurophysiological studies on biological visual system are used to design engineering models. In this work the design of neural architectures for preattentive visual processing is motivated by some aspects of the visual perceptual process in biological visual system.

1.4.3 Scope of the Work

The focus of the work is on neural network architectures for data driven extraction of geometric properties. We assume that the input pattern is clean and has a noise free boundary contour shape. The issue of pattern recognition is not addressed in detail, although in all these cases recognition studies have been made using standard neural architectures.

1.4.4 Overview of the Research

In this work, we have proposed two approaches for processing binary images. We have developed two neural network architectures based on these approaches. The first approach is implemented through an oriented filtering and integration network. The second approach is implemented using directed spreading activation layers. We also describe two applications of these architectures.

1.4.5 Organization of the Thesis

Chapter 2 discusses the motivation and proposes two approaches to preattentive visual processing. Chapter 3 discusses the design of Oriented filtering and Integration Network and the application of **this** architecture for isolated word recognition. Chapter 4 discusses the directed spreading activation neural architecture and proposes a methodology for recognizing transformation invariant binary pattern recognition. Chapter 5 concludes the thesis with a summary of the work.

APPROACHES FOR PREATTENTIVE VISUAL PROCESSING

2.1 INTRODUCTION

Numerous approaches are proposed in the literature for preprocessing the visual patterns. In Section 2.2, we categorize these approaches into four classes and briefly review these approaches. Visual pattern recognition has been attempted by neural networks also. In Section 2.3 we review some of the neural principles and architectures for visual pattern recognition. In Section 2.4 we discuss approaches adopted in this work for preattentive visual processing.

2.2 BACKGROUND

Visual pattern recognition deals with the analysis of visual patterns in order to achieve results similar to those obtained by man. A simplified machine paradigm for visual pattern recognition consists of two computational stages. The first stage is concerned with low level techniques and referred in the literature as picture processing or preprocessing. When neural networks are used for such initial processing it is called preattentive visual **processing**[14]. The second stage is referred as picture interpretation or pattern matching or recognition stage. The focus of this work is on the first stage using neural networks.

Low level **analysis** involves aggregation of imperfect edge data in the two-dimensional image projection. Here, shape attributes of collection of edges are computed and a description consisting of the shape attributes and their spatial locations are generated. This description serves as input to a subsequent process of high level organization and understanding.

There exist many theories of visual pattern or shape description and recognition, each attempting to explain some specific aspect of the problem. This is so because it is possible to conceptualize visual pattern as a high level perceptual function. Since there is very little neurophysiological evidence about its nature and the basic constituents are not known, the field has been open to freewheeling hypothesization. These theories can be broadly categorized as **follows**[52]: correlation techniques, computational approaches, neurophysiological and sensory-motor **approaches**[33,34]. Among these correlation and computational approaches are engineering approaches. The other two approaches are motivated by the studies from neurophysiology and visual perception research. These studies are especially useful to design artificial neural networks. In this section we briefly review these four theories.

Among the four categories the correlation technique is followed in the pattern recognition research. In Section 2.2.1 we summarize techniques proposed in pattern recognition research for visual pattern description and recognition.

Any visual pattern recognition task must be implemented in an algorithm form. Implementation of such algorithm requires a computational framework for representing the algorithm. In Section 2.2.2 we discuss a framework for computational visual processing.

The sensory-motor approach to visual processing is modeled after the oculomotor movements of the eye. In Section 2.2.3 we briefly describe the oculomotor movements of eye and its role in visual perception.

The biological visual perception is carried out by the neural mechanisms in visual cortex and superior **colliculus** of the brain. In Section 2.2.4 we present **some** of the reports from neurophysiology about visual cortex.

2.2.1 Pattern Recognition Approach

Pattern Recognition techniques for preprocessing binary images can be broadly classified into two approaches, spatial domain approach and scalar transform approach. Spatial domain approach focuses on aggregating edge data and transform the input image into an alternative spatial domain representation, The input images are transformed into a representative graph which portrays the two-dimensional shape. Subsequent recognition of the shapes is accomplished by means of syntactic or structural analysis. Among spatial domain techniques there have been two approaches. The first approach uses a collection of fixed templates of geometric

features like straight line segments of different orientations, corners and T-shapes; The input image is scanned for these patterns and a representative graph which portrays the two-dimensional shape is generated.

The other approach is based on information theoretic point of view suggested by **Attneave[1]**. He suggested that a shape is **segmented** by means of dominant points which coincide with points of maximum inflection along its contour. Pattern recognition has proposed a number of techniques for extracting dominant points in the input **pattern[49]**. These techniques are mostly an outgrowth of interest in specific applications, the most common being the recognition of handwritten characters and chromosome types.

Scalar transform techniques map the image into an attribute vector description. The objective here is to transform the boundary data into a new representation, one in which object translation, rotation, and size are no longer factors. The method of moments offer such a possibility. There have been many applications of this methodology to pattern recognition problems. These have included printed characters and **numerals[3]**, hand-printed **characters[7]**, chest **x-rays[18]**, aircraft **identification[10]**, and ship **recognition[48]**. Categorization of shapes with this approach is usually achieved by means of classical pattern recognition.

2.2.2 Computational Framework for Visual Processing

Since vision is an interdisciplinary research field large number of theories are proposed in other disciplines like neurophysiology and perceptual psychology. If we want to develop artificial visual systems, these theories developed in the other disciplines must be tested rigorously. For rigorous testing, they must be converted into algorithms. Expressing visual theories as algorithms leads to the development of computational models. In creating computational models, several important issues must be addressed. In this section we discuss a framework for computational visual processing and isolate functional characteristics of an architecture for preprocessing.

2.2.2.1 Low level versus high level visual processing

A useful conceptual simplification is to divide the visual process into two levels: low level visual processing and high level visual processing. Low level processing deals directly with the incoming visual stimuli. Simple features may be extracted and simple patterns recognized. The high level visual processing is concerned with cognitive processing and makes use of the knowledge about the world when processing the visual information. Which visual cues are to be chosen by the lowest levels is an important consideration, as all further processing depends on **how** well this initial stage is carried out.

2.2.2.2 Serial versus parallel processing

It is useful to distinguish between the type of processing used by high and low level visual processes in terms of serial versus parallel processing. The low level visual processing is primarily performed in parallel. Evidence for this assumption comes from four different areas namely neurophysiology, psychophysics, machine vision and computational theories. Serial processing is more likely to occur at the high levels of visual processing.

2.2.2.3 Automatic versus selective processing

Low level visual processing involves parallel **computations** performed simultaneously at many locations on the image. **Much** of this processing is performed automatically without intervention from higher levels. High level processing is more likely to be serial and require flexible **control** of the operations to be **performed**. Another way to discuss the automatic versus selective issue is in terms of bottom-up versus top-down processing. Automatic **processing** can be performed bottom-up without using information from higher levels. On the other hand, selective processing might **require** top-down processing where there is feedback between the different stages of processing. At the low level, bottom-up processing can be done in parallel, automatically without flexible control and efficiently.

2.2.2.4 Signal versus symbols

Low level processing is closely tied to the image, or the visual signal. By contrast, high level processing deals with cognitive symbols rather than visual signals. The main task of the early stages of visual processing is to extract meaningful information from the total visual information and to pass it on to the higher levels of **processing**. The problem is in deciding how the information should be represented. There **are two** possibilities, either the useful visual features could be labeled and that information transmitted symbolically, or else a scheme not requiring the explicit labeling of features could be employed.

2.2.3 Eye Movements and Visual Pattern Perception

The sensory-motor theory of visual pattern description and recognition is motivated by the oculomotor **movement** of eye. In this section we briefly review the role of eye-movement for visual pattern perception.

The interaction with the world around relies to a major extent on the ability to actively look, visually scan, and selectively pick up information on the basis of which effective, **visually guided** action can be deployed. Such visual scanning and deployment of goal-directed behavior in turn requires spatial as well as temporal coordination between sensory and motor processes. Spatially what is required in

sensory-motor coupling is that the outer world be projected systematically onto a motor map of the body; Much of this sensory-motor coupling is **reflexive**[4,54]. The visual perceptual cycle is characterized by (1) the directing of sensory apparatus to (2) selectively pick up information which serves to (3) **modify** and update the schemata that in turn direct the further pick-up of **information**[40].

The rapid movement of the sensory apparatus to pick up information is called '**saccade**'. The saccades are driven between points of interest in the visual field and play an essential role in human visual processing, particularly in the establishment of spatial **relations**[35,54]. Saccades are controlled by a complex set of interrelationships between low level and high level cues. The superior colliculus of the brain, which receives both retinal and cortical projections, directs the **saccades**[4].

2.2.4. Results from Neurophysiology for Visual Processing

The neurophysiology approach for visual pattern description is motivated by the reports from the results of biological neural mechanisms for vision. In this section we review the neural mechanisms for visual perception.

The neural mechanisms involved in the visual perception seems to be superior colliculus and visual **cortex**[4]. The superior colliculus is involved in localizing and detecting the presence of a visual stimulus which may be potentially

informative and behaviorally **significant**[4]. However, it is not involved in the detailed qualitative analysis or identification **of the** stimulus. By contrast, the visual cortex seems to be involved primarily in the localization of a stimulus and in analyzing its qualitative and figural aspects.

In the visual cortex four classes of cells are distinguished in a series of ascending **complexity**[23]. These are termed as '**circularly symmetric**', '**simple**', '**complex**' and '**hypercomplex**'. Circularly symmetric cells show no preference to any particular orientation of lines and act as contrast detectors. Simple cells are the first in the hierarchy to orientation specificity. A simple cell responds to an optimally oriented line in some narrowly defined position, even a slight displacement of the line to a new position without change in orientation renders the line ineffective. A complex cell, on the contrary, is as specific in its orientation requirements as the simple cell, but is far less particular **about** the exact positioning of the line. Such a cell will respond wherever a line is projected within a rectangle. Hypercomplex cells respond to more specific types of stimuli than either **simple** or complex cells. They respond maximally to edges, **corners**, curves and angles of particular sizes.

In the literature neural architectures are reported simulating some of the properties of the visual cortex, and used in visual pattern recognition **systems**[12,15,24].

2.3 REVIEW OF NEURAL NETWORK ARCHITECTURES FOR VISUAL PATTERN RECOGNITION

Theoretical neurodynamic approaches in cognitive sciences seek to replace symbol-manipulating formal computational rules with a short yet powerful list of elementary neural principles[17]: 1.Competition 2.Cooperation 3.Shunting inhibition 4.Adaptive feedback 5.Resonance. This short list pf neural principles are the basis of diverse phenomena encountered in the cognitive sciences and neurosciences. The large number of computational neural models reported in the literature[6,20,29,30,36] are found to have based on these elementary neural principles.

These elementary neural principles give raise to some interesting neural properties like associative recall, self organization, adaptive resonance and competitive learning. Number of architectures are proposed in the literature demonstrating these properties. These architectures include Hopfield Net[20,21,22], Hamming net[30], Adaptive Resonance Theory[6], Self organizing Maps[29], Boltzman machine[36], perceptron[30] and back propagation[36].

Various neural architectures for visual pattern recognition tasks are reported in the literature[12,15,24]. These architectures are designed to solve specific visual pattern recognition problems like handwritten character recognition, recognition of silhouettes etc. In visual pattern recognition, in general, the feature distribution of

the input is not identical with that of the stored template. Hence a mechanism which can resolve the differences is necessary. There have been two approaches to this problem. The first one is to incorporate the mechanism into feature extracting stages as neocognitron **does**[12]. The second approach regards the feature extraction and pattern matching as separate **stages**[55].

In the following section we briefly review the **neocognitron**[12] architecture which follows the first approach. There are other architectures for visual pattern recognition which follow the second **approach**[55]. These architectures use geometrical or analytical methods to extract features from the input pattern. These architectures use standard neural architectures like multilayer perceptron for recognition. We do not review these architectures here.

23.1 Neocognitron: An Architecture for Visual Pattern Recognition

Fukushima proposed the **cognitron**[13] model for pattern recognition. This model does not have the capability to correctly recognize the position-shifted or shape-distorted patterns, **Neocognitron** which is an improved version of the conventional **cognitron** and has the capability to recognize stimulus patterns correctly, even if the patterns are shifted in position or distorted in shape. It has a hierarchical structure. The **information** of the stimulus pattern given to

the input layer of the neocognitron is processed step by step in each stage of the multilayered network. A cell in the deeper stage generally has a tendency to respond selectively to a more complicated feature of the stimulus patterns. At the same time it has a larger receptive field and is less sensitive to shifts in position of the input pattern. Thus, each cell in the deepest stage responds only to a specific stimulus pattern without being affected by the position or the size of the stimulus patterns.

Neocognitron handles shifts by replicating the receptive field of a feature to cover the entire visual field. Distortions are tolerated by integrating the response from overlapping receptive fields of the previous stages in the subsequent stages. The successful performance of neocognitron is due to the gradual steps with which this replicating and integrating process is done. However, when this network is applied to other problem domains it poses a number of problems.

Since the inner layers of neocognitron are trained for specific patterns, it falls short of the general purpose vision system. Each new pattern to be learnt is to be manually segmented and trained to various layers of the network. This is a comparatively easy task in the case of numerals for which neocognitron was shown. But designing such network for a pattern which has curves and lines as features, like in the case of images of formant contour patterns in speech, becomes extremely cumbersome. Moreover the training patterns

like Arabic numerals themselves do not have any noise. If noise itself is part of the pattern then the first stage of neocognitron itself filters out such information and cannot be used by subsequent stages.

2.4 PREATTENTIVE VISUAL PROCESSING: ISSUES AND APPROACHES

There are two issues to be addressed in the design of neural architectures for preattentive visual processing. The first issue is to identify different types of preattentive visual processing. The second issue is to find neural principles useful for the design of neural architectures. In this section we discuss these issues.

The biological visual process can be functionally segregated into visual perception and visual **cognition**[4]. The visual perceptual process extracts information about the geometry of the visual world and the visual cognitive process concerns with the recognition of familiar objects. The visual perception in biological visual system seems to be automatic, does not use any detailed knowledge of the visual patterns, and extracts properties of the visual input which are not immediately used for recognition.

The visual perception is based on two interrelated processes: parallel processing of visual information carried out automatically by mechanisms determined by neuronal organization of the retina, lateral geniculate nucleus, and visual cortex; and sequential processing is related to image

recognition mechanisms and is controlled by **attention**[27]. In the first process, detector properties of single neurons and local neuron nets are of primary importance. Here, orientation of edges and contour elements of the input image are extracted by these neurons. In the second process, eye movements are considered to be an essential factor. As a result of these movements, the most informative parts of the image are sequentially projected onto the fovea for fine **processing**[5,54].

Therefore, an adequate computer system **for the** processing and analysis of visual information should include a preprocessor with a neural network architecture, simulating parallel information processing at low levels of the visual system, and a sequential type neural system tuning the preprocessor to obtain necessary information for image recognition. Development of the neural network preprocessor should be preceded by a study on **neuronal** organization of low level structures of the visual system, their mathematical modeling and computer simulation.

Based on the observations about the visual perception, we have considered two possible approaches for processing binary images: The first approach **extracts** primitive line segments from an input pattern and retains the spatial relationship between the features. In this processing the detector properties of individual neurons and their spatial locations are important. This architecture is implemented as an oriented filtering and integration network. In Chapter 3

we discuss this architecture and its application for isolated word recognition. In the second approach the maximum information points from the input pattern are located. This is implemented using directed spreading activation layers. Spreading activation layers reported in the **literature**[37] uses isotropic spreading of activation to carry out early vision tasks like feature clustering and feature centroid determination. The directed spreading activation layers proposed in Chapter 4 uses anisotropic or directed spreading of activation followed by maxima detection to locate maximum information points from the input image.

Since preattentive visual processing is parallel, the neural network architectures have in their input stage two dimensional array of neurons and the input pattern is fed directly to this array. Also, since preattentive visual processing is purely data driven and does not use any detailed knowledge **about** the patterns, the neural computations must be from local data, **i.e.**, each neuron receives its input from local data only. Apart from the **computations** from local data, it is possible to have lateral interactions between neurons. In this work we show how **neurocomputations** from purely local data extract structural features, and local data computations with lateral interactions between neurons give rise to an architecture which extracts maximum information points.

2.5 SUMMARY

In this chapter we have discussed four theories about visual pattern description and recognition. We have reviewed some of the neural network principles and architectures for visual pattern recognition. Some aspects of the visual perception are presented. Based on these observations we have considered two approaches to **preattentive** visual processing. In the following chapters we discuss two neural architectures and applications based on these two architectures.

ORIENTED FILTERING AND INTEGRATION NETWORK FOR STRUCTURAL FEATURE EXTRACTION

3.1 INTRODUCTION

In this chapter we present the design of the oriented filtering and integration network. This architecture extracts the structural features like straight line segments from the input image. This is similar to the first stage of **neocognitron**[12], but differs in the implementation of the integrating network. We show how this network can be applied for recognizing isolated utterances of words from the images of formant contour patterns. Section 3.2 discusses the structural organization and functional characteristics of the oriented filtering and integration network. In Section 3.3 we describe the design of a neural architecture for recognizing isolated utterances of words.

3.2 ORIENTED FILTERING AND INTEGRATION NETWORK (ORFIN)

This is a two stage hierarchical network as shown in Fig.3.1. Each stage consists of a number of two dimensional array of neurons and these neurons are of analog type, *i.e.*, the input output signals of the cells take

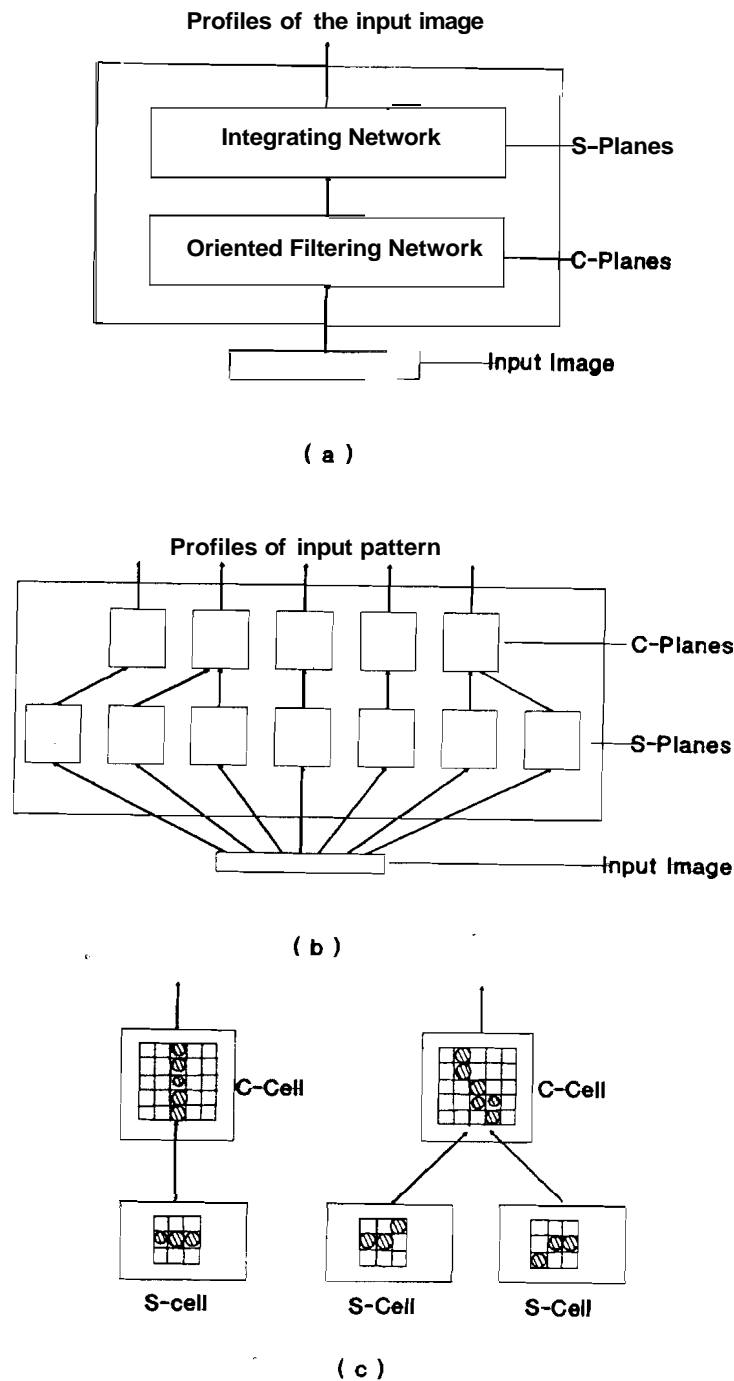


Fig.3.1 This figure illustrates the structural organization of **ORFIN**. (a) shows the block diagram of **ORFIN** and (b) illustrates the interconnection between S-planes and C-planes. Outputs of two of the S-planes which have the same orientation of stimuli but trained differently are fed to corresponding C-planes. This is shown as outputs from two S-planes converging into a single C-plane. (c) illustrates examples of S-cells whose outputs are fed to corresponding C-cells.

nonnegative analog values. The first stage is an oriented filtering network (also referred to as S-layer) which extracts line segments from the input pattern. The second stage is an integrating network (also referred to as C-layer) which integrates responses from overlapping fields of the output of the first stage. The computation in the second stage allows small variations in the positions of the line segments.

In this architecture all the computations are carried out from local data only. These two stages are motivated by the orientation specificity shown by simple and complex cells in the visual **cortex**[23].

Functional characteristics and structural organization of this network are described in detail in the following sections.

3.2.1 Design of Oriented Filtering Network

This network extracts line segments from the input pattern by filtering through a **number** of planes called **S-planes**. Each one of the S-planes consist of two-dimensional array of cells and each cell favors a specific orientation of preferred **stimuli**. There are two types of cells in the S-plane, called S-cells and **V_s -cells**. The S-cells receive input **from** either excitatory or inhibitory input terminals. If the cell receives signals from excitatory input **terminals** the output of the cell will increase. On the other hand, a signal from inhibitory input terminal will suppress the **output**. Each

input terminal has its own interconnection coefficient whose values are positive. These values determine the preference of the orientation of the cell. The output of the S-cell goes to a number of input terminals of next C-layer.

The schematic diagram illustrating the interconnections converging to a S-cell is summarized in Fig.3.2. Each one of the S-cells receives its inhibitory signal from the V_s -cell which causes the shunting effect. All the S-cells in the given S-plane are trained to respond for a specific orientation of stimuli. The V_s -cells are trained to recognize the absence of the specific orientation of stimuli. So if the input stimuli is exactly similar to the trained stimuli, then S-cells respond to its maximum and V_s -cells respond to its minimum. On the other hand, if the input stimuli is completely different then the S-cells respond to its minimum and V_s -cells respond to its maximum.

Both S-cell and V_s -cell receive input interconnection from the same spatial distribution. All the other cells in the same cell-plane have input interconnection from the same spatial distribution and only the positions of the input cells to which their terminals are connected are shifted in parallel from cell to cell. Fig.3.3 is a schematic diagram illustrating the interconnections from this stage to the second stage. In this diagram for the sake of simplicity, only one cell is shown in each cell-plane. Each of these cells receive input interconnection from the cells within the area enclosed by circle in its preceding layer.

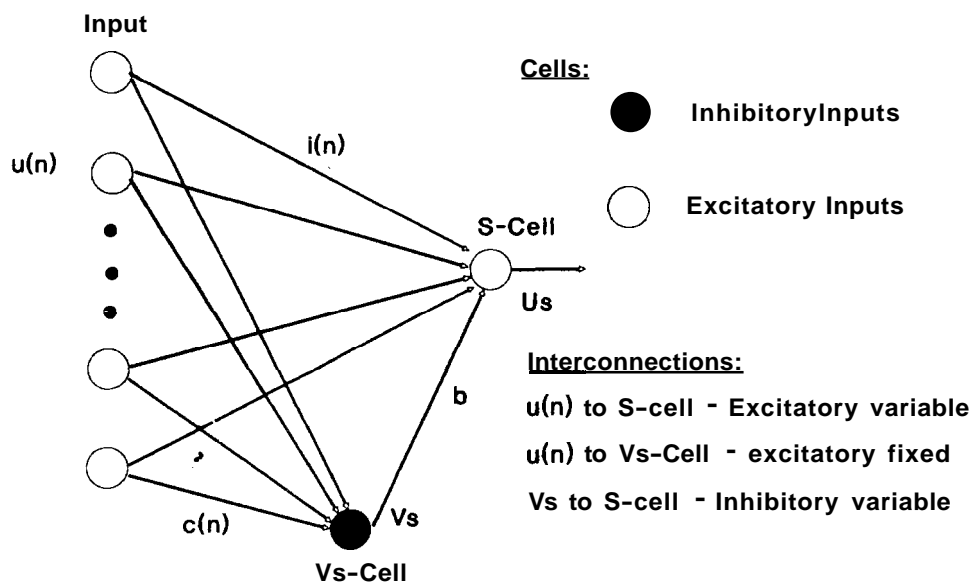


Fig. 3.2 Interconnections converging to a S-cell.

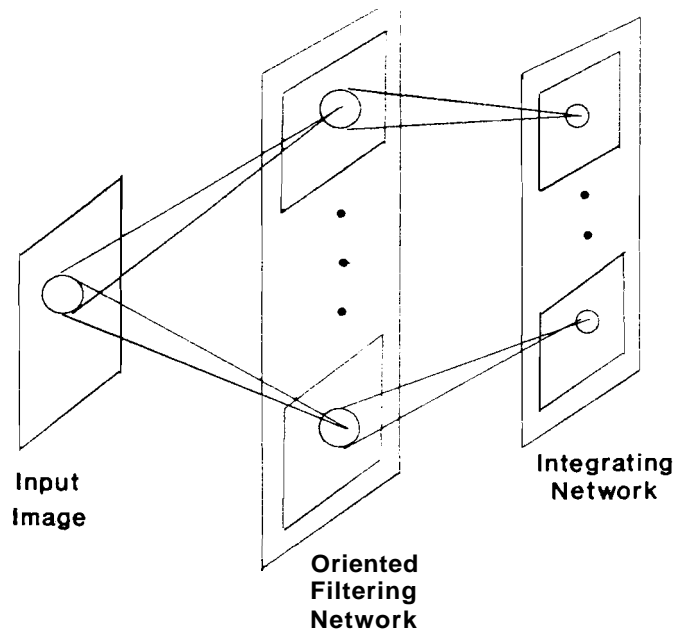


Fig. 3.3 Schematic diagram illustrating the interconnections between the two stages.

Let $u(1), u(2), \dots, u(N)$ be the **excitatory** inputs and V_s be the inhibitory input. Then the S-cell output is computed using the following equation:

$$U_s = r * \varphi \left[\frac{\left\{ 1 + \sum_{n=1}^N u(n) * i(n) \right\}}{1 + \frac{r}{(r+1)} * b * V_s} - 1 \right] \quad (3.1)$$

where $u(n)$ and b represent the excitatory and inhibitory coefficients respectively, $i(n)$ is the fixed weight pattern, V_s is the output of the V_s cells and r is a constant. The characteristic behavior of S-cell is summarized in Fig.3.4. The function $\varphi()$ is defined by the following equation:

$$\varphi(x) = \begin{cases} \frac{x}{(\alpha+x)}, & \text{for } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

where α is a positive **constant** which determines the degree of saturation of the output.

The output of V_s -cell is computed using the following equation:

$$V_s = \left[\sum_{n=1}^N c(n) * u^2(n) \right]^{1/2} \quad (3.3)$$

The fixed values of $c(n)$ are determined so as to decrease monotonically with respect to the center and to satisfy $c(n) = 1.0$. (Though $c(n)$ is a two dimensional array for notational convenience it is denoted as a single dimensional array).

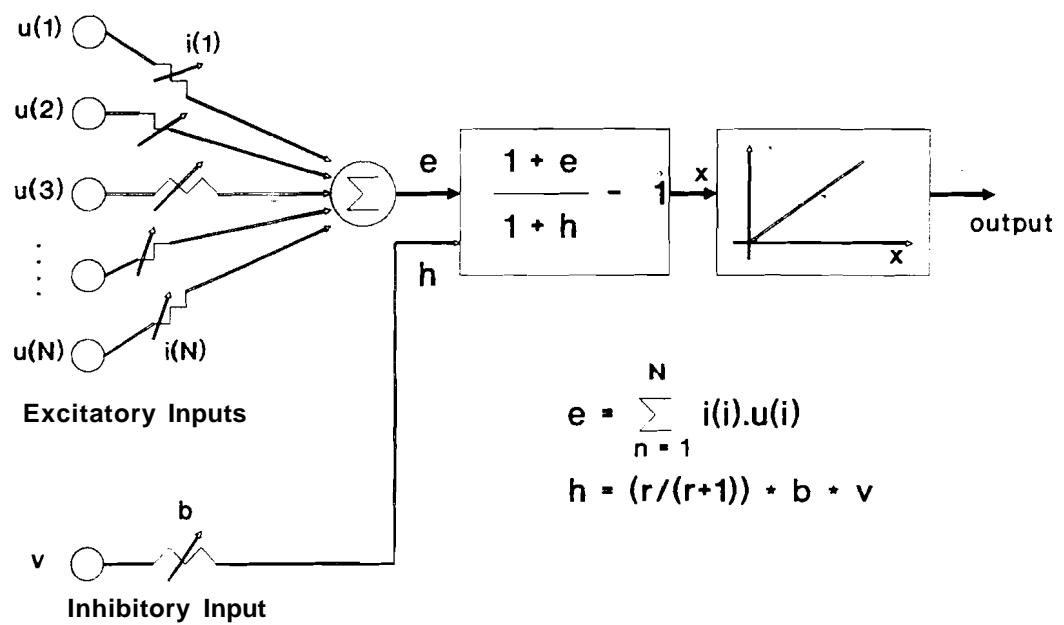


Fig. 3.4 Input-to-output characteristics of a S-cell.

The input area for a S-cell is taken from a 3x3 array. In the 3x3 array twelve orientations are possible. These twelve orientations are shown in Fig.3.5.

3.2.2 Design of Integration Network

This network integrates responses from overlapping fields of the output of the first stage to tolerate small variations in the positions of the line segments. This network also consists of a number of planes called C-planes, and each C-plane consists of two dimensional array of cells. There are two types of cells, C-cells and V_c -cells in the C-planes. Both C-cells and V_c -cells receive input from the S-plane. C-cells receive inputs from S-cells and V_c -cells. Each C-cell has input interconnections leading from a group of S-cells and these interconnections are fixed and unmodifiable. All the **S-cells** in the C-cells' connecting area extract the same stimulus feature from a slightly different positions on the input layer. The values of the interconnection between S-cells and C-cells are determined such a way that the C-cell will be activated whenever at least one of these S-cells is active. V_c -cells average the input from S-cells which have same orientation but trained differently. Fig.3.6 shows **some** examples of the interconnection topology. Even if a stimulus pattern which has given a large response from the C-cell is shifted a little in position, the C-cell will still keep responding as before.

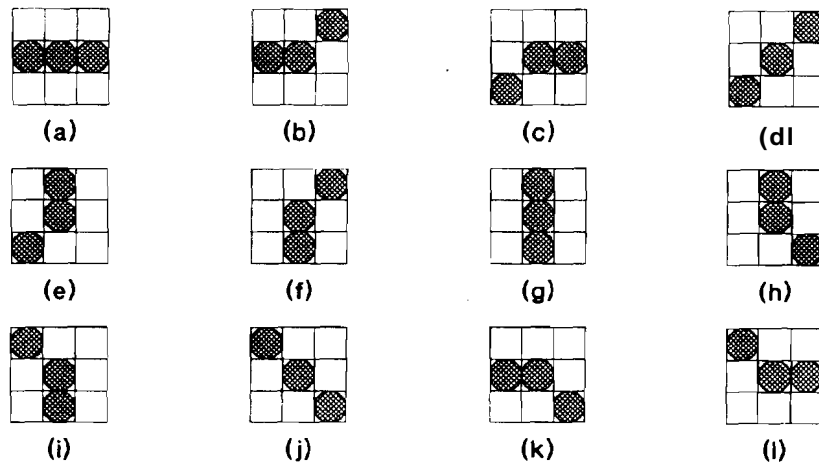


Fig. 3.5 Twelve line segments used to train S-cells.

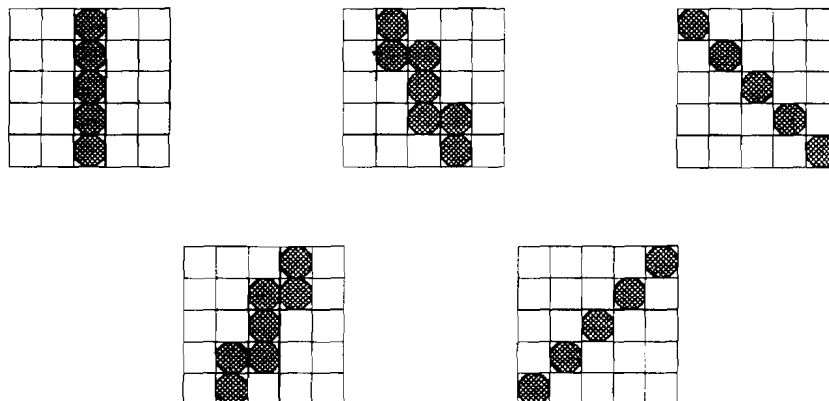


Fig. 3.6 Fixed weight pattern between S-cells and C-cells. This pattern is responsible for handling small shifts in the input visual pattern.

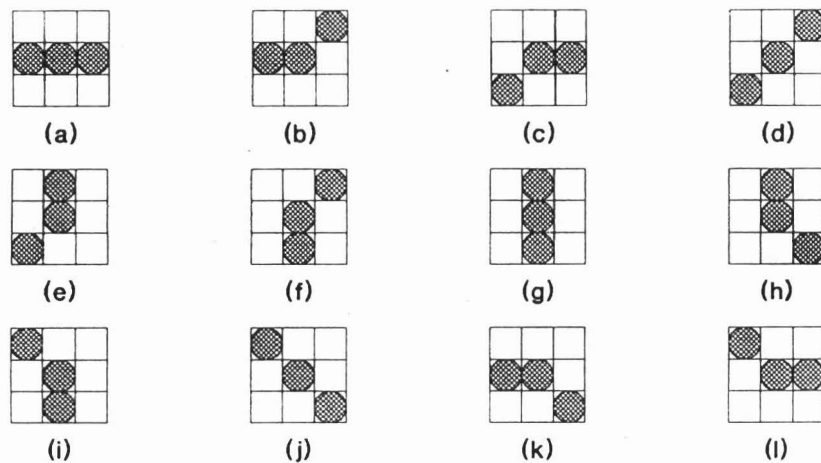


Fig. 3.5 Twelve line segments used to train S-cells.

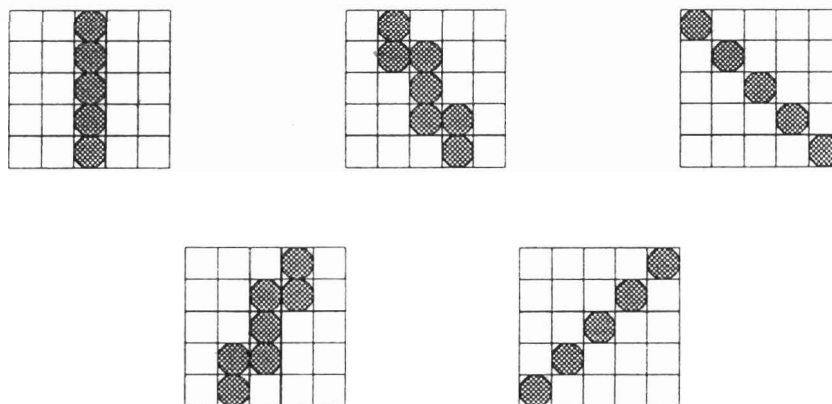


Fig. 3.6 Fixed weight pattern between S-cells and C-cells. This pattern is responsible for handling small shifts in the input visual pattern.

In other words, a C-cell responds to the same stimulus feature as the S-cells, but is less sensitive to the position of the stimulus feature.

There are twelve orientations of stimuli in the S-layer. These are connected to the eight C-planes in the C-layer. The S-planes which have same orientation of stimuli but trained differently are fed to a single C-plane. This is illustrated in **Fig.3.1b** and **3.1c**. This figure is illustrated for seven S-planes and five C-planes. **Fig.3.1b** shows how the interconnections between S-planes and C- planes are arranged. Some examples of **S-cells'** outputs feeding C-cells are shown in **Fig.3.1c**.

The output values of C-cells are computed using the following equation:

$$U_c = \varphi \left(\frac{1 + \sum_{n=1}^N d(n) * U_s(n)}{V_c} - 1 \right) \quad (3.4)$$

where $\varphi()$ is a function defined by **eqn(3.2)**, $d(n)$ denotes the values of the interconnection **topology**, and V_c is the output of **Vc-cells**. In this implementation $d(n)$ is assigned a constant value. (In neocognitron[12] $d(n)$ is assigned monotonically decreasing values with respect to the center. The **eqn(3.4)** is also simplified and differs from neocognitron.) V_c is computed using the following equation:

$$V_c = \frac{1}{k} \sum_{n=1}^N d(n) U_s(n) \quad (3.5)$$

where k is the number of S-planes connected to a C-plane.

To summarize the functional behavior, **ORFIN** extracts straight line segments with tolerance in their positions while retaining the spatial relationship between them. This generates profiles of the input pattern which can be used for recognition. In the following section we show how this preprocessing is useful for recognizing isolated utterances of words from the images of formant contour patterns.

3.3 APPLICATION OF ORFIN FOR ISOLATED WORD RECOGNITION

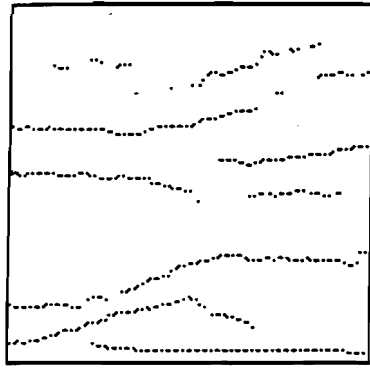
The isolated word **recognition(IWR)** systems reported in the literature consider **parameters[42]** like spectral coefficients, discrete **Fourier transformed(DFT)** spectrum, linear prediction **coefficients(LPC)** etc. as input for recognition. These parameters are extracted from the speech signal from the patterns and these patterns, are then matched by template matching techniques. The nonlinear temporal changes in these patterns are handled by using dynamic programming techniques like dynamic time **warping[25,46]** and probabilistic models like hidden markov **models[43]**. The success of the **IWR** systems depends on the choice of parameters and the technique adopted to match these parameters.

Speaker independent isolated word recognition with these parameters has been attempted with partial **success[47]**. The reason for the partial success of the parametric representation used for speaker independent IWR systems can

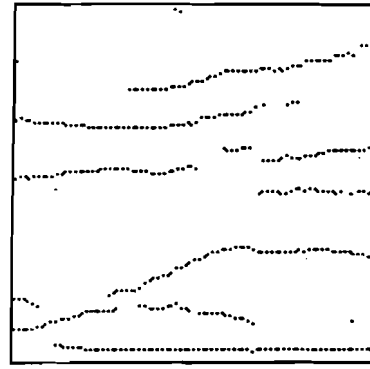
be ascribed to the parameters' inability to capture the features of the word. The utterances of the same word by two speakers show little similarity in the parametric form. But the same words show significant similarity in the gross features level in the spectrogram. Though there is a relative shift in the features depending on the speaker, there are common features between them.

Formants are resonances of the vocal tract system. These formant values vary slowly and continuously with time. The formants carry information relating to the identification of the speech sounds. Changes in the formant values with time can be traced to obtain a formant contour. This formant contour reflects the movements of the articulators positioned in sequence. Even though different speakers utter the word, the articulatory movements need to be the same. Such formant contour represents the speech signal in the form of an image. Fig. 3.7 shows some examples of the images of formant contour patterns. In this work the images of formant contour patterns extracted from the speech signal are considered as input to the isolated word recognition system.

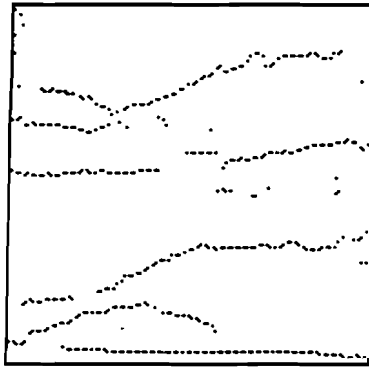
In the images of the formant contours, the features are simple lines and curves and they undergo distortions and shifts depending on the utterance and speaker. Even for the same speaker these formant contour patterns show variations. Here, both the absolute location and the relative arrangement



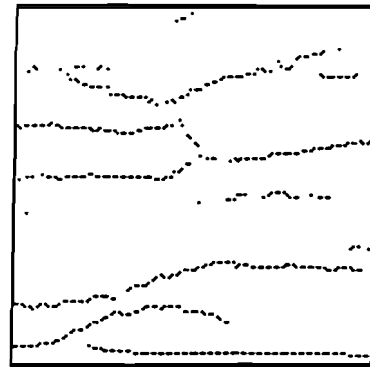
(a) Formant Plot: One



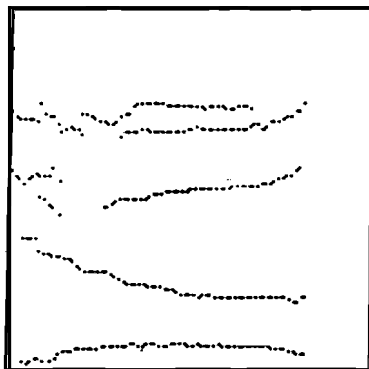
(b) Formant Plot: One



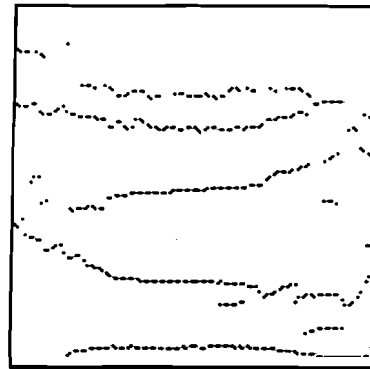
(c) Formant Plot: One



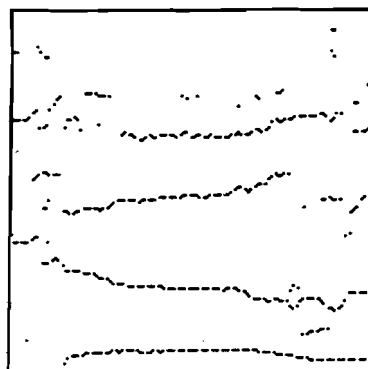
(d) Formant Plot: One



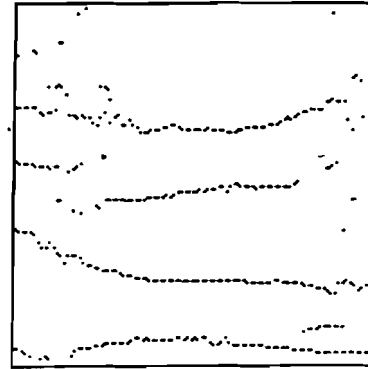
(e) Formant Plot: Two



(f) Formant Plot: Two



(g) Formant Plot: Two



(h) Formant Plot: Two

Fig. 3.7 Some examples of images of formant contour patterns are shown.

of the features are significant. For example, depending on the vowel, the positions of the lines representing **F1** and F2 formant frequencies change.

The formant contour of the same word undergoes changes in both time scale and frequency scale. This is reflected in changes in the shapes and lengths of the curves and straight lines. This results in a significant change in the binary pattern and a drastic change in the physical location of the pixels. So the image of the formant contour pattern cannot be used for simple template matching. However, at a higher level the curves and straight lines exist as specific features of the utterance.

The approach adapted in this work attempts to preprocess the images of the formant contour to get an invariant **representation**. The distortions and shifts in the input pattern are processed by the preprocessing technique. Here we have attempted to use the oriented filtering and integration network for preprocessing the images of formant contour patterns.

In the following section we describe a neural architecture for recognizing isolated utterances of words from the images of formant contours.

3.3.1 Design of Isolated Word Recognition System

The organization of the neural architecture proposed is shown in the Fig.3.8. This system consists of two stages. The first stage is called Feature Extraction **stage(FE)** and the second stage is called Pattern **Matching(PM)** stage. Oriented filtering and integration network is used as FE stage. The small distortions and shifts of the features of the formant contours are preprocessed by this network to get an invariant representation.

Since the formant contour image does not have any lines with angles above 45° , all the orientations of stimuli above 45° need not be considered. This eliminates five of the twelve orientations. So the number of S-planes in the S-layer in this system is seven responding to seven different orientations. The outputs of these S-planes are fed to five C-planes. The FE stage generates different profiles from the input image which are the outputs of C-planes of the integration network.

The five C-planes generate five different profiles. These profiles are input to the PM stage (Fig. 3.9). The PM stage is a hierarchical Adaptive Resonance **Architecture[6]**. It consists of two stages of **ARTs** in a hierarchy. First stage consists of five Simple Adaptive Classifiers called SAC-1. Each SAC-1 receives one of the profiles as input. It classifies the profile into a category. Each SAC-1 makes its

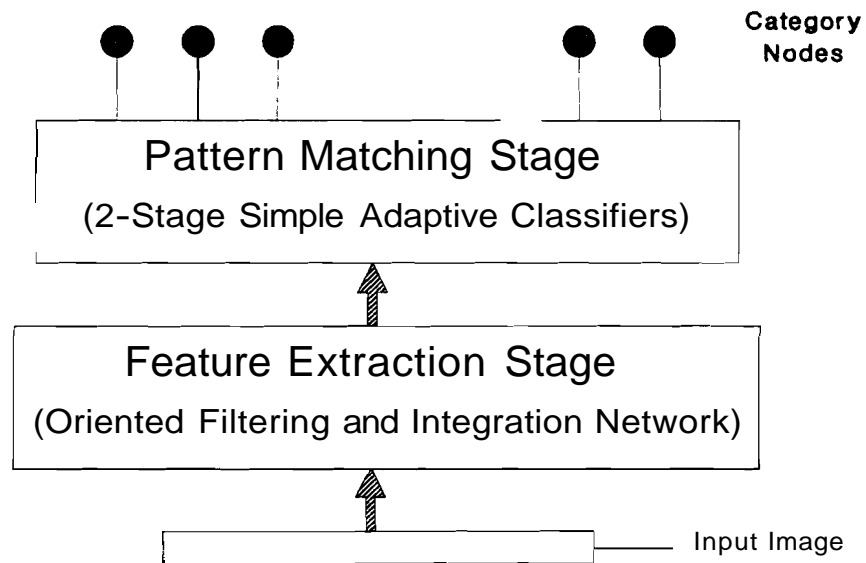


Fig. 3.8 Neural architecture for recognizing isolated utterances of words. First stage extracts structural features using **ORFIN**. Second stage implements two-stage **Simple** Adaptive Classifiers for recognition.

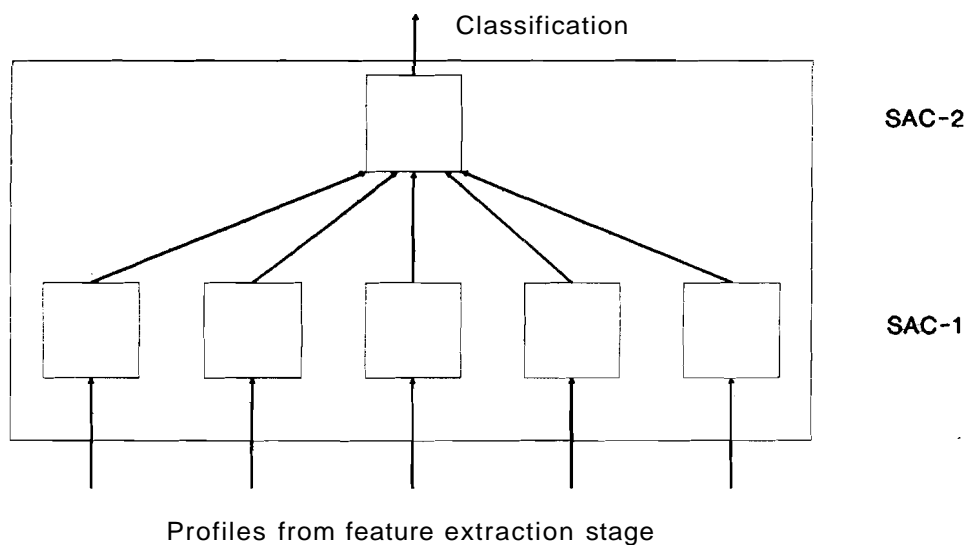


Fig. 3.9 Pattern Matching Stage is a hierarchical adaptive **resonance** architecture. SAC-1 categorizes the profiles. SAC-2 **classifies based on the** categorization done by SAC-1.

decisions based purely on the specific profile it receives. The second stage is also a Simple Adaptive Classifier, called SAC-2. All the outputs of the SAC-1 are fed to SAC-2 and it merges the classification done by SAC-1 and identifies the input pattern. The Simple Adaptive Classifiers follow the adaptive resonance architecture (**Fig.3.10**) and the salient points of this architecture are summarized below.

The main feature of adaptive resonance architecture is the adaptive resonance that occurs between the current input and learned expectations. In ART the system which carries out the adaptive resonance is called attentional subsystem, which consists of bottom-up and top-down adaptive filters. These filters are contained in pathways from a feature representation field (F1) to a category representation field (F2) whose nodes undergo competitive-cooperative interactions.

An auxiliary orienting subsystem controls the self organizing and recognizing capability of ART. When a new input is added at any time, the system would search the established categories. If an adequate match is found on the initial search cycle, the bottom-up weights would be refined if necessary to incorporate the new pattern. If no match is found and the full coding capacity is not exhausted a new category would be formed with previously uncommitted F2 nodes encoding the new input pattern.

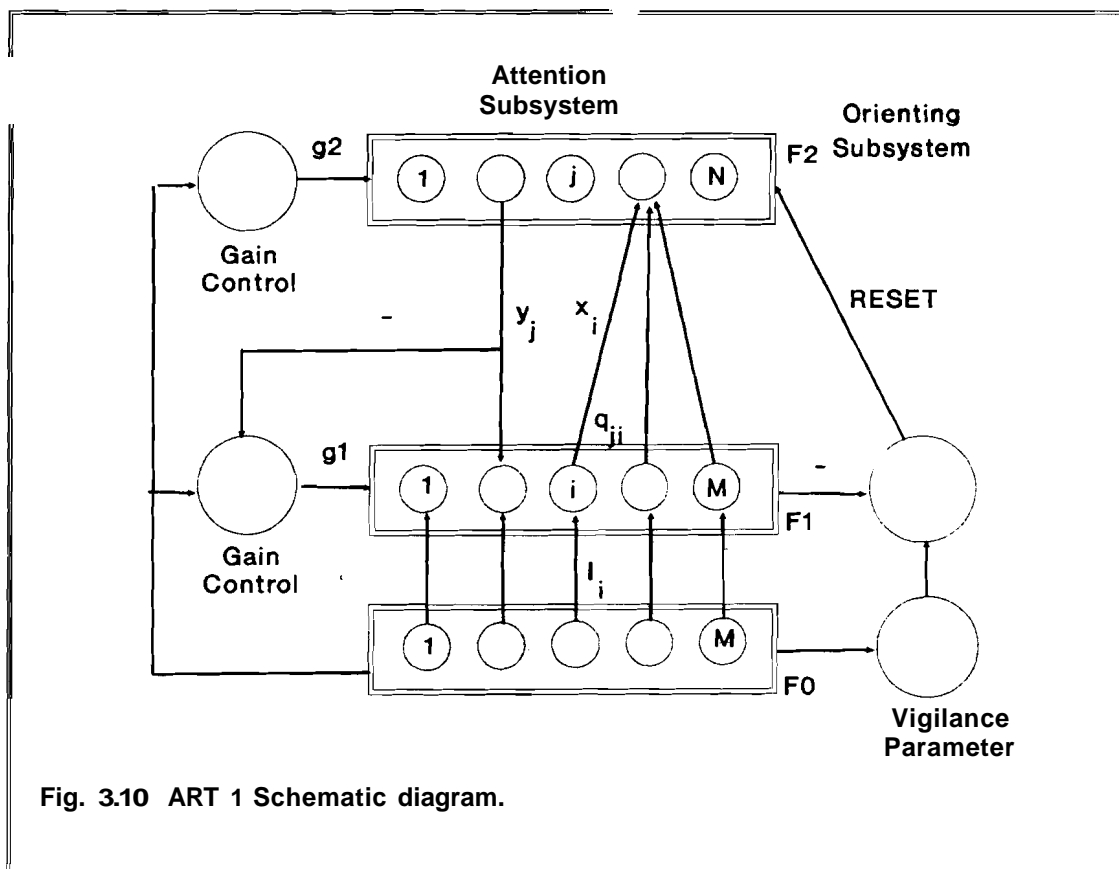


Fig. 3.10 ART 1 Schematic diagram.

The auxiliary orienting subsystem becomes active when a bottom-up input to $F1$ fails to match the learned top-down expectation read-out by the active category representation at $F2$. In this case, the orienting subsystem is activated and causes rapid reset of the active category representation at $F2$. This reset event automatically induces the attentional subsystem to proceed with a parallel search. Alternative categories are tested until either an adequate match is found or a new category is established. The search proceeds rapidly relative to the learning rate. Thus significant changes in the bottom-up and top-down adaptive filters occur only when a search ends and a matched $F1$ pattern resonates within the system.

The criterion for an adequate match **between** an input pattern and a chosen category template is adjustable in an ART architecture. The matching criterion is determined by a vigilance parameter that controls activation of the orienting system. All other things being equal, higher vigilance imposes a stricter matching criterion, which in turn partitions the input set into finer categories. Lower vigilance tolerates greater **top-down/bottom-up** mismatches at $F1$, leading in turn to coarser categories.

Fig.3.10 illustrates the main components of ART module in detail. Field $F1$ of M nodes, with output vector $X = (x_1, x_2, \dots, x_M)$, registers the input vector $I = (I_1, I_2, \dots, I_M)$. The bottom-up weights are denoted by q_{ij} and top-down weights are

denoted by z_{ij} . The index i is used for the feature representation nodes of the field $F1$ and the index j is used for category nodes in the field $F2$. In the current implementation the input feature **vector** I is a two dimensional vector for both **SAC-1** and SAC-2. This is denoted as a single dimensional vector for convenience. The size of M for SAC-1 is taken to be 32x32 and for SAC-2 is taken to be 5x15.

Each $F1$ node can receive input from three sources: the bottom-up input, nonspecific gain control signals which is received by all the nodes at $F1$ at the same time, and the top-down signals from the N nodes of $F2$ via an top-down adaptive filter. The nonspecific gain signals in SAC-2 are activated only after SAC-1 stabilizes the resonance activity. Therefore SAC-2 is inactive when SAC-1 is active. A node in $F1$ is said to be active if it generates an output signal equal to 1. Output from inactive nodes equals 0. The 2/3rule[6] is realized in its simplest, dimensionless form as follows:

2/3 Rule Matching: The i^{th} $F1$ node is active if its net input exceeds a fixed threshold. Specifically,

$$x_i = \begin{cases} 1 & \text{if } I_i + g_1 + \sum_{j=1}^M y_j z_{ji} > 1+k \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

where term I_i is the binary input, term g_1 is the binary nonspecific $F1$ gain control signal, term $\sum_{j=1}^N y_j z_{ji}$ is the sum of top-down signals y_j via pathways with adaptive weights z_{ji} , and

k is a constant such that $0 < k < 1$. In this implementation k is chosen to be 0.23 which is the least value computed by the C-cells of the integrating network.

F1 gain control: The $F1$ gain control signal g_1 is defined by

$$g_1 = \begin{cases} 1 & \text{if } F0 \text{ and } F2 \text{ are active} \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

Since $F2$ activity inhibits $F1$ gain

$$x_i = \begin{cases} 1 & \text{if } I_i = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

If only one of the $F2$ nodes are active eqn(3.6) reduces to the single term z_j , so

$$x_i = \begin{cases} 1 & \text{if } I_i = 1 \text{ and } z_{ji} > k \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

The case where two $F2$ nodes are active at the same time has not occurred during our simulation.

F2 Choice: Let T_j denote the total input from $F1$ to j^{th} $F2$ node, given by

$$T_j = \sum_{i=1}^N x_i z_{ji} \quad (3.10)$$

where the z_{ji} denote the bottom-up adaptive weights. If some $T_j > 0$, define the $F2$ choice index J by

$$T_J = \max(T_j; j = 1, \dots, N)$$

In the typical case, J is uniquely defined. Then the $F2$ output vector $y = (y_1, y_2, \dots, y_N)$ obeys

$$y_j = \begin{cases} 1 & \text{if } j=J \\ 0 & \text{otherwise} \end{cases} \quad (3.11)$$

If two or more **indices** j share maximal input, then they equally share the total activity. In the simulation this situation also never arose because of the nature of the distinct categories of isolated words.

Learning Laws: The adaptive weights reach their new asymptote on each input presentation. The learning is gated by $F1$ activity: that is, the adaptive weights z_{ji} and q_{ji} can change only when the j^{th} $F2$ node is active.

Top-down learning: When the y_i gate opens then learning of top-down weights z_{ji} begins and z_{ji} is attracted towards x_i . This is called **outstar learning rule**[17]. Initially all z_{ji} are set to 1. The $F2$ activity vector can be described as

$$x = \begin{cases} I & \text{if } F2 \text{ is inactive} \\ I + Z_J & \text{if the } j^{th} \text{ node is active} \end{cases} \quad (3.12)$$

When node I is active, learning causes $z_j = I + z_j(old) - 1$ where $z_j(old)$ denotes z_j at the start of the input presentation. The first time an $F2$ node J becomes active, it is said to be uncommitted. In this case $z_j = I$ during learning. Thereafter node is said to be committed.

Bottom up learning: In simulations it is convenient to assign initial values to the bottom-up adaptive weights q_{ji} in such a way that $F2$ nodes first become active in the order

$j = 1, 2, \dots, N$. This is done by choosing the bottom-up weights small but decreasing order. This is accomplished by letting $q_{ji} = \alpha_i$ where $\alpha_1, \alpha_2, \dots, \alpha_N$.

Like the top-down weights vector z_j , the bottom-up weight vector q_j also becomes proportional to the $F2$ output vector x when the $F2$ node J is active. In addition the bottom-up weights are scaled inversely to $|x|$, so that $q_{ij} = \frac{x_i}{(\beta + |x|)}$ where $\beta > 0$. During learning q_j is computed by

$$q_j = \frac{(I + z_j(\text{old}) - 1)}{\beta + |I + z_j(\text{old}) - 1|} \quad (3.13)$$

Since learning depends on the few samples provided in the initial stages of the training the network, it is possible that from the training set provided it may not be possible for the system to **generalize** for correct recognition. Hence the network is allowed to learn continuously even during the recognition phase. To facilitate such learning possible, the vigilance parameters are adjusted during recognition.

3.3.2 Data Preparation

A number of approaches are proposed to extract formant contours from the speech signal. Some of the approaches proposed extract the formant frequencies by linear prediction analysis or from cepstrum. Another approach to extract formant frequency from speech signal is using group delay function [19]

which is the negative derivative of the Fourier transform phase. The group delay function derived from the Fourier transform phase of a signal has two important properties, namely, additive and high resolution. Hema[19] has proposed a technique for formant extraction from group delay function using these properties. From the group delay the formant frequencies are picked using a simple peak picking method. In this work the formant contour is extracted from the speech signal using the above technique.

The speech signal is sampled at 10,000 samples per second. These samples are grouped into blocks of 256 samples. Each block is processed through the group delay formant extraction technique. The next block is chosen by shifting 32 samples. This processing generates the image of the formant contour. This image should be preprocessed before feeding into the proposed system. There are number of issues to be addressed for preprocessing the images.

The first issue is to normalize the temporal variations in the image. Depending on the time taken for uttering the word the length of the x-axis of the image changes. Since the input to the proposed system is a fixed two-dimensional array of visual pattern, the formant contour should be normalized before feeding into the system. This essentially involves normalizing the duration of the uttered speech signal. In this work we have used a simple normalizing technique. The time expansion and compression is carried out in vowel regions of the uttered signal. The vowel region in

the formant contours contains nearly horizontal lines. In these locations the formant contours are compressed or expanded and normalized to specific size of the input.

The second issue is to remove the noisy peaks in the image. A simple support point technique is used to remove the noisy pixels of the image. In this technique each point in the image is retained only if there are atleast 20 neighboring points. The other issue is to process the discontinuities in the image. The same support point technique which is used above automatically corrects the discontinuities.

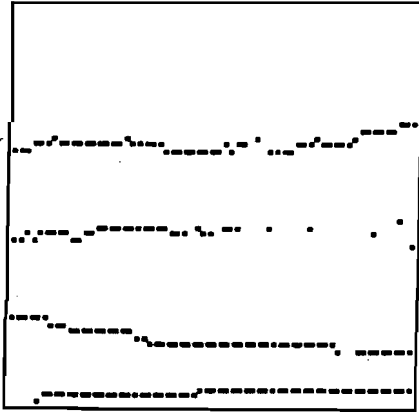
3.3.3 Implementation Details and Results

In the current implementation the S-layer in the FE stage consists of seven S-planes. The S-cells in these S-planes are tuned to seven different orientations. Each S-plane consists of 64x64 array of S-cells. The orientation for which each S-plane responds is already trained and the values are hard-coded into the program. Each pattern is a 3x3 array as shown in Fig. 3.5. Each S-cell receives its input from a window of size 3x3. The adjacent S-cell receives the input from an overlapping window. A number of parameters are used in eqn(3.1) and (3.2) for computing the outputs of S-cells. These parameters are fine tuned for a good performance. The value of r is taken to be 1.7, $b = 1$, $\beta = 0.5$ and $\alpha = 0.333018$.

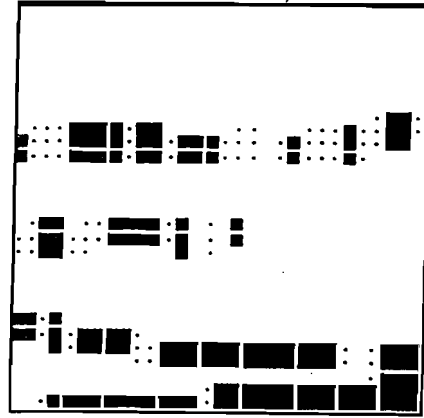
The C-layer in the FE stage consists of five C-planes. Each C-plane consists of 32x32 array of C-cells. The outputs of the S-planes are connected to the C-planes through the interconnection topology as shown in Fig.3.6. This topology is a 5x5 matrix for each C-plane and hardcoded into the program. Each C-cell receives its input from output of the S-plane having a window of size 5x5. This feature extraction phase finally generates five different profiles each of size 32x32. These profiles are fed to the PM stage. An example of the outputs of C-planes for the utterance TWO are shown in Fig.3.11.

Field **F1** of SAC-1 is an array of size 32x32. Field F2 has 15 category nodes for classification. All the five SAC-1 classifiers together generate a two dimensional array of values of size 15x5 which is fed as input for SAC-2. Hence, in SAC-2 the field F2 has an array 15x5 input nodes. There are 10 category nodes in field F2 of SAC-2.

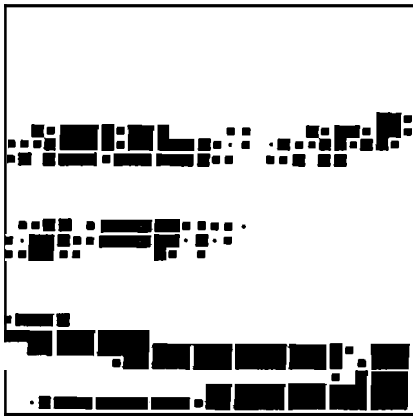
There are two isolated word recognition tests conducted on this system. We have selected utterances of the digits for recognition. In the first test the system is tested with the utterances of a single speaker. The recognition results of the system for a single speaker with 20 utterances of each digit, are shown in Table 3.1. The system was trained with three utterances of each word.



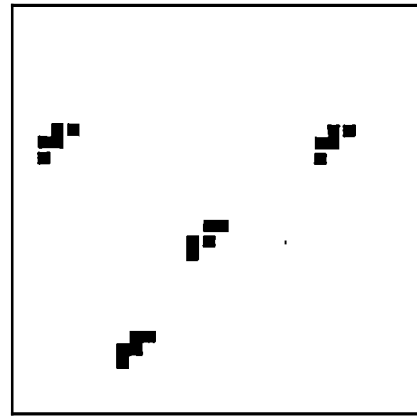
(a) Compressed Plot: Two



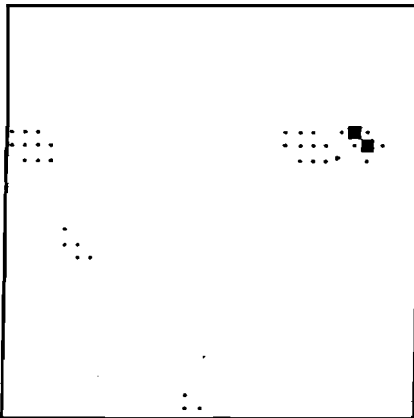
(b) Output of C-Plane id:0



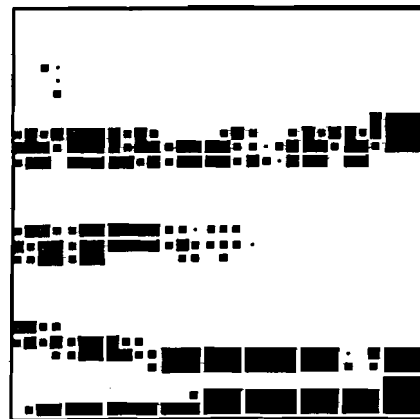
(c) Output of C-Plane id:1



(d) Output of C-Plane id:2



(e) Output of C-Plane id:3



(f) output of C-Plane id:4

Fig. 3.11 Image of a formant contour pattern campressed into 64x64 array is shown in (a). The **output** values of five C-Planes are shown for the example input pattern. The size of the block in (b)-(f) indicates the value of C-cell at that point.

Table 3.1. Isolated Word Recognition System Test results for a single speaker

Words (20 each)	Correctly recognized	Unclassified	Misclassified
Zero	18	2	-
One	20	-	-
Two	14	6	-
Three	20	-	-
Four	18	2	-
Five	16	-	4 (As Eight)
Seven	17	3	-
Eight	17	3	-
Nine	14	2	4 (As Five)

In the second test the system is tested with isolated utterances of digits from two American speakers. The system is trained with two utterances each of the two speakers and tested with five utterances of each speaker. The results are shown in Table-2 and Table-3.

Table 3.2. Isolated word recognition system test results for two speakers: Speaker-1

Words (5 each)	Correctly recognized	Unclassified	Misclassified
Zero	5	-	-
One	5	-	-
Two	3	2	-
Three	4	1	-
Four	5	-	-
Five	5	-	-
Seven	4	1	-
Eight	5	-	-
Nine	3	2	-

Table 3.3 Isolated word recognition system test results for two speakers: Speaker-2

Words (5 each)	Correctly recognized	Unclassified	Misclassified
Zero	5	-	-
One	2	3	-
Two	4	1	-
Three	4	1	-
Four	5	-	-
Five	5	-	-
Seven	2	1	2 (As Five)
Eight	2	2	1 (As Five)
Nine	4	-	1 (As Five)

From the tests conducted we observe that the system performs well for a single speaker for distinct words. Words like FIVE, EIGHT and NINE have the same dominant vowels and formant contour image for these words show similar horizontal lines. The system attempts to locate the distinct features of these words for classification and shown good results, for example 16 out of 20 instances of FIVE are identified correctly. The system misclassifies these words in some cases. This may be attributed to the limitation of using iamges of formant contour patterns which capture only the resonances of the system properly. In the second test also we observe **misclassifications** in those words where there is vowel domination.

3.4 SUMMARY

In this chapter we have presented the design of the oriented filtering and integrating network for structural feature extraction. We have also described an application of this architecture. A neural architecture for recognition of utterances of isolated words from the images of the formant contour patterns is **presented**. We have described the implementation details of the neural architecture' and also presented the test results.

DIRECTED SPREADING ACTIVATION LAYERS FOR LOCATING MAXIMUM INFORMATION POINTS

4.1 INTRODUCTION

In this chapter we present the design of directed spreading activation layers. This architecture extracts the maximum information points in the input image. We describe two applications of this architecture. In the first application we show how low level features can be extracted from the machine fonts. In the second application we show how transformation invariant binary pattern recognition can be achieved using the maximum **information** points generated by this architecture.

Spreading activation **layers**[37] has been used to carry out early vision tasks like feature clustering and feature centroid determination. However, studies reported in the literature use isotropic spreading of activation. In this chapter we discuss the drawbacks of the spreading activation layers for locating maximum information **points** and propose a new directed spreading activation model. In Section 4.2 we describe the spreading activation layers. Section 4.3 discusses the motivation for the directed spreading and Section 4.4 describes the design of the directed spreading

activation model. We discuss the implementation details and examples in Section 4.5. In Section 4.6 and 4.7 we show some applications of these architectures.

4.2 SPREADING ACTIVATION LAYERS

Evidence for rapid diffusion like phenomena are found in the brightness and color domains of stabilized image experiments. Compelling evidence is provided by **Yarbus's**[54] experiments, in which color from the surround rapidly fills regions in which stabilized images have faded. These evidences are reported in the brightness domain. **But the** diffusion-like phenomena are used in both high level information processing **models**[2,26] and low level visual processing models **also**[14,15,37]. Spreading activation layers use this diffusion like phenomena for early vision tasks.

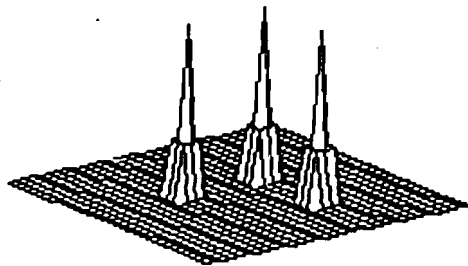
Diffusion enhancement is a low level computational model which has been used in building a neural network vision **system**[37]. This model is used for learning and recognizing two-dimensional binary patterns invariant of their location, orientation and scale. The processing is divided into layers, each of which encompass many levels of neuron-like processing cells. This low level processing model carries out early vision tasks like feature extraction, feature clustering and feature centroid determination. In the following **sections** we summarize the salient features of the spreading activation layers.

4.2.1 Activity Diffusion and Centroid Detection

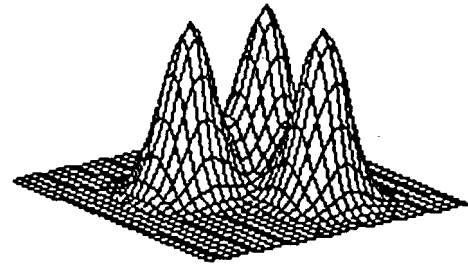
Consider a region R and an activation function $A(R)$ defined over it at an initial time t_0 . Let the function $A(R)$ be binary values at t_0 , either A_{sat} or 0 , corresponding to locations where maximum information or the low level features on the binary image have been detected. The maximum **information** points are the high curvature points detected by a technique proposed by **Rosenfeld**[28]. Now let the activation diffuses locally through the region according to the classical diffusion equation:

$$\frac{dA}{dt} = \nabla [k(R) \cdot \nabla A(R)] \quad (4.1)$$

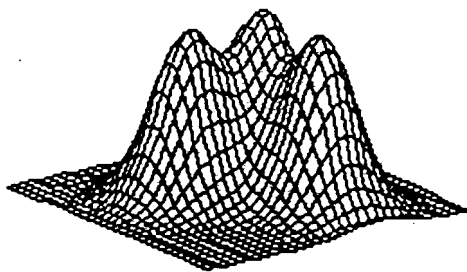
where $k(R)$ accounts for the density and conductivity of the region. If $k(R) = k$ this reduces to $dA/dt = k^2 A(R)$ a constant. If the total activation is held constant, then the locations with initial activation A_{sat} begin to lose activation, while adjacent locations begin to gain activation. Due to superposition, areas near activation-rich locations gain activation more quickly than areas far from the activation-rich locations. Fig.4.1 plots the activity distribution surface as it spreads by the simple diffusion as described above. Activity spreads as the time progresses from t_0 (**Fig.4.1a**) until a global activity maximum emerges (**Fig.4.1d**), indicating the geometric centroid of the features. At an intermediate time various local maxima can



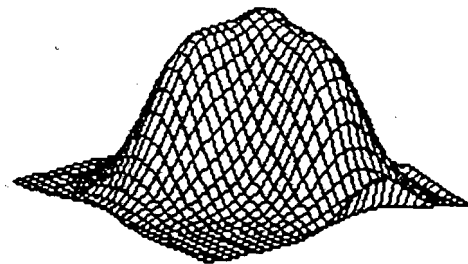
(a)



(b)

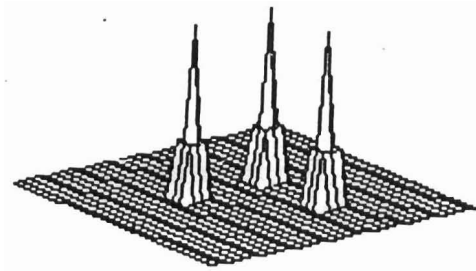


(c)

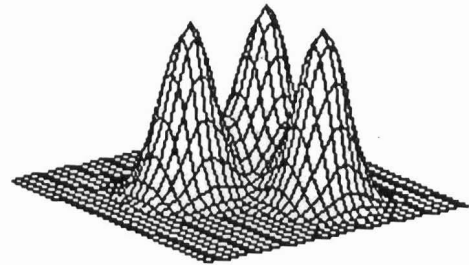


(d)

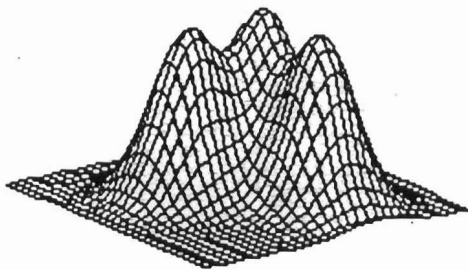
Fig. 4.1 The activity distribution of the spreading activation layer is plotted in three dimensions at four times: (a) at t_0 as diffusion begins; (b) at t_0^+ , after a short time; and much later in (c) and (d). In (d) the peak is located at the geometric centroid of the three features as shown in (a).



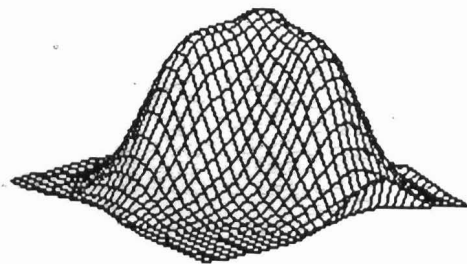
(a)



(b)



(c)



(d)

Fig. 4.1 The activity distribution of the **spreading** activation layer is plotted in three dimensions at four times: (a) at t_0 as diffusion begins; (b) at t_0^+ , after a short time; and much later in (c) and (d). In (d) the peak is located at the geometric centroid of the three features as shown in (a).

be located. Fig.4.2 shows the time sequence of two feature locations spreading, superimposing their tails, and finally merging at the centroid. This example is shown for one dimensional spreading.

The activation distribution in the diffusion level defines a surface over a 2D plane. Extrema of activity are found in areas of positive curvature of the surface. The maximum is computed in neural networks by self-activation and competition. Using lateral inhibition, each element suppresses its neighbors according to its activation, while feeding back an excitatory activation to itself. This is accomplished using an **on-center/off-surround** recurrent receptive-field for each element. Among other properties, this type of network **enhances**[16] the contrast of the activity distribution, or in the extreme case, leaves only the maximally activated element on. This type of network along with spreading activation layers locates 'the feature centroid of the given feature points.

4.22 Feature Extraction in Spreading Activation Layers

Curvature along contours are useful for recognition of shapes from 2D images. Spreading activation layers may be used in locating the curvature **along** contours. Fig.4.3 shows the result of using spreading activation layers for locating a corner. The figure shows that the areas near high curvature points along the contour are easily found, since they receive superimposed activation from a greater number of locations

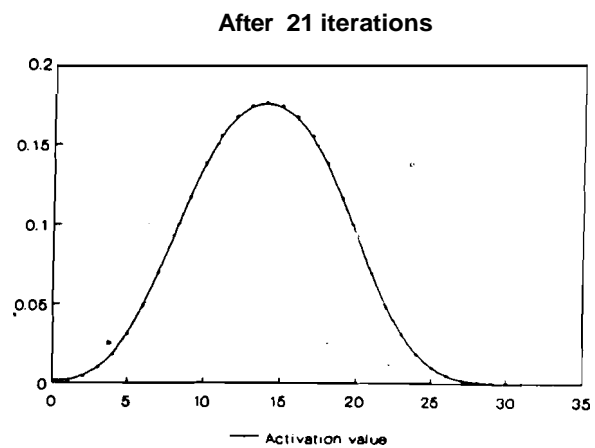
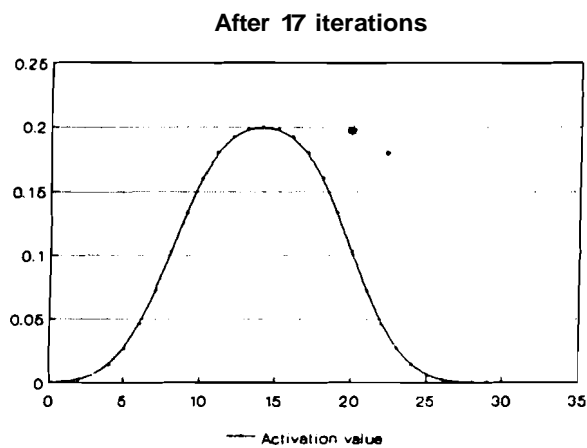
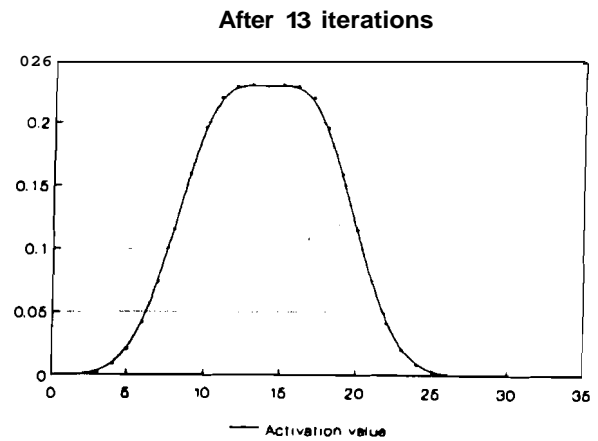
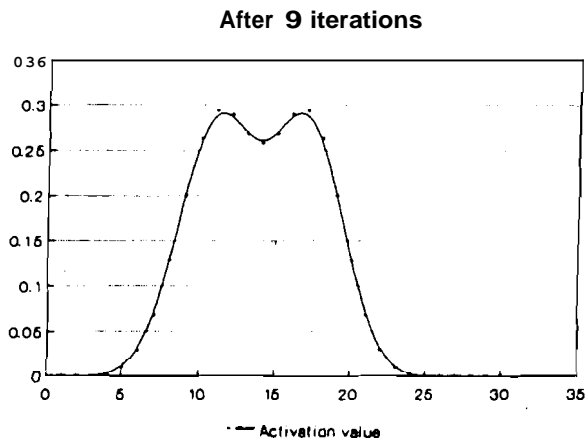
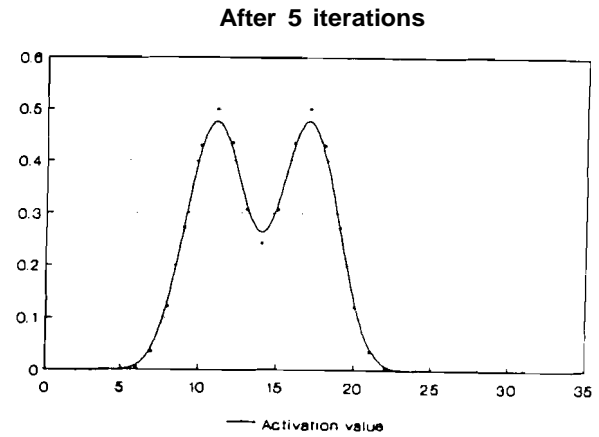
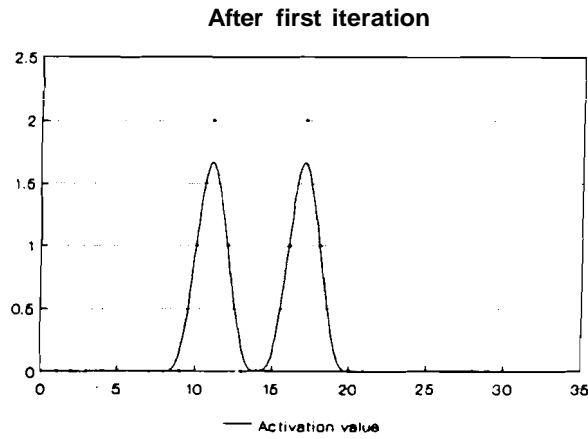
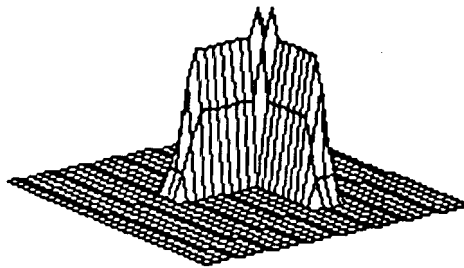
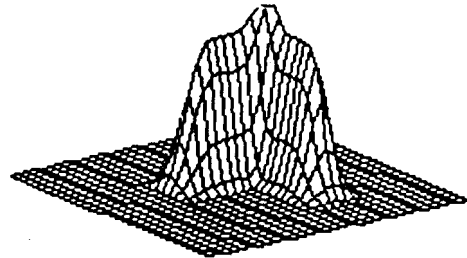


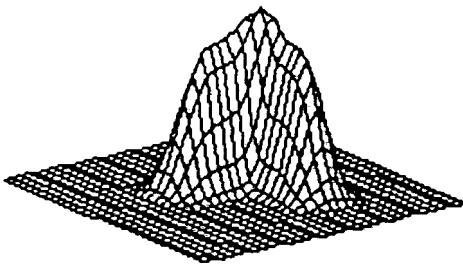
Fig. 4.2 As time progresses (a) to (f), the activity distributions initially due to two features spread. As activity spreads the local maxima moves toward the centroid. The global maxima is stable at that point.



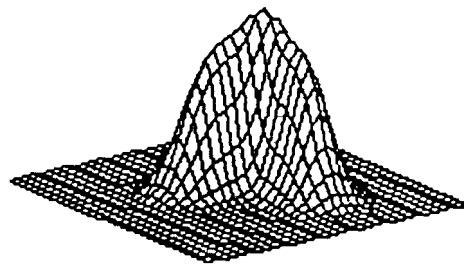
(a)



(b)



(c)



(d)

Fig. 4.3 A contour with a **corner** shown in (a) is diffused in (b)-(d). Activity accumulates more **quickly** where the average distance to the features is least. As the **diffusion** progresses the activity **maxima** moves to the global centroid. Since maxima moves continuously it is difficult to determine when to stop diffusion to locate the peak at the **corner**.

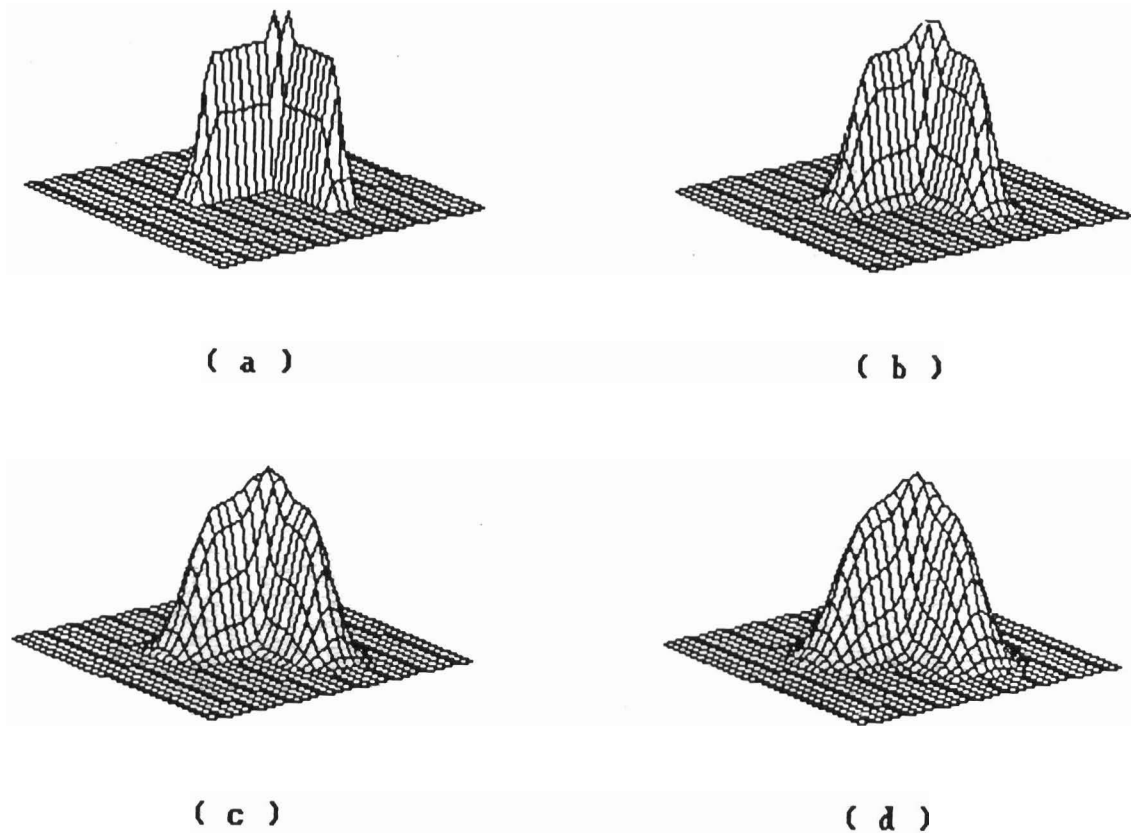


Fig. 4.3 A contour with a **corner** shown in (a) is diffused in (b)-(d). Activity accumulates more **quickly** where the average distance to the features is least. As the diffusion progresses the **activity** maxima moves to the global centroid. Since **maxima** moves continuously it is difficult to determine when to stop diffusion to locate the **peak** at the **corner**.

than areas near straight contours. But a certain amount of care is required in using diffusion as a corner and contour termination points detector. If the diffusion is too short on a coarsely sampled image, then maxima will be detected for a short time. If the diffusion is too long, as the diffusion progresses, the maxima points merge together with real corners, and corners located around small features merge together.

4.2.3 Centers of Focus of Attention

Since initial activation function corresponds to locations where features have been located, the diffusion as it progresses form feature clusters. These feature clusters can be used as a center of focus of the saccadic controller of any visual system. Since the activation level of each maxima point depends on the density of features nearby, it may be used to prioritize the importance of feature area as a fixation point. The level of detail, and thus the size of the feature cluster, can be controlled by the extent in time of the diffusion process. For instance, if the diffusion results can be sampled before extensive feature clustering occurs, they will reflect small feature clusters and a high level of detail. If recognition using the clusters found at this fine level of detail is incomplete, the diffusion may be allowed to proceed, creating larger feature clusters. Hence, small scale organization emerges before large scale organization in a natural way.

Fig.4.4 shows an example of the feature clustering in a binary image. The figure illustrates how small scale organization arises naturally before a large scale organization. These small local clusters are shown in different stages of spreading.

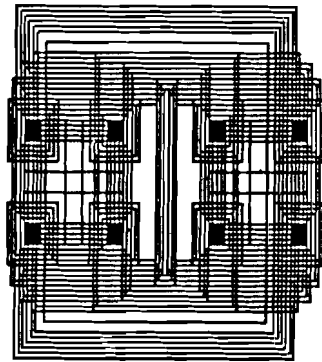
Features can be separated from each other by merging into different activity groups. These different groups emerging as a function of time can be processed individually leading to piecewise support for recognizing a complete object, even in the presence of noise or occlusions.

4.3 MOTIVATION FOR DIRECTED SPREADING

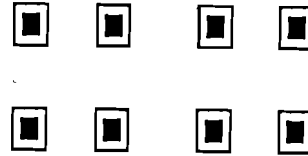
4.3.1 Drawbacks of the Spreading Activation Layers for Low level Feature Extraction

The objective here is to use spreading activation layers for low level features or maximum information points extraction. In this section we discuss the drawbacks of using spreading activation layers for extracting low level features.

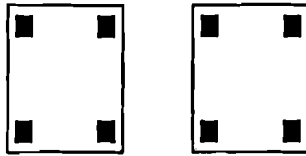
The spreading activation layers is essentially employs an averaging process. When the input pattern is directly presented to the spreading activation layers, as the time progresses, the activation values of the individual neurons reflect the averaging process which takes place over two dimensional space. This kind of averaging is unconstrained



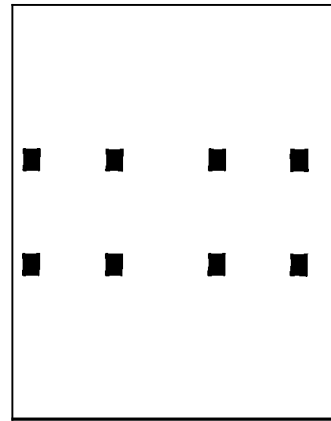
(a)



(b)

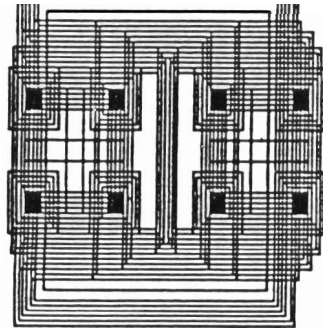


(c)



(d)

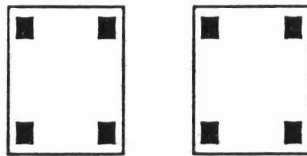
Fig. 4.4 Small scale organization of feature clusters emerges before large scale organization. (a) shows the continuous process of feature clustering. (b)-(d) shows different snapshots of feature clustering at different times.



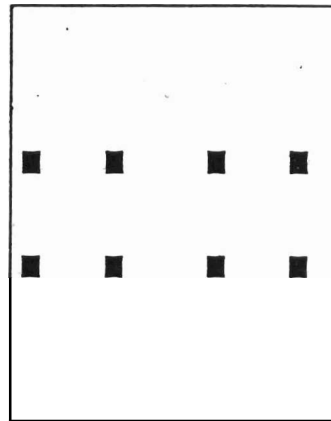
(a)



(b)



(c)



(d)

Fig. 4.4 Small scale organization of feature clusters emerges before large scale organization. (a) shows the continuous process of feature clustering. (b)-(d) shows different snapshots of feature clustering at different times.

because there is neither a limiting factor nor a complementary mechanism to constrain the spreading of activation in both time and space. The local maxima formed as time progresses, represent various features and feature clusters in the image. As there is no constraint in the spreading it is very difficult to determine '**a priori**' when to stop the spreading process and identify features or feature clusters, since the peaks which are formed during the spreading slowly drift away towards the global centroid. Hence the main problem in using spreading activation layers for feature extraction is identifying the temporal event for stopping the spreading.

The location of quasi-static **points**[37] during the spreading activation process has been proposed as a temporal event for determining the feature clusters. This quasi-static point method cannot be adopted to the low level feature extraction directly as the feature maxima tend to move continuously towards the global centroid. To overcome this problem the feature extraction phase and feature cluster identification phase are isolated in spreading activation layers. The feature points are detected by nonneural techniques and the feature map is considered as input for spreading instead of the direct input pattern. But the lines, curves and contour termination points which are not retained are very useful and significant as they contain information useful for invariant pattern recognition. When the **eye/camera** movement is used to identify the features located at the maxima points, the lines and contour termination points

will be missed. Even though spreading activation layers is not successful in low level feature extraction, it can be successfully used for saccadic movement, once the maximum information points on the binary images are located.

4.3.2 Basis for Directed Spreading Activation Model

This drawback of the spreading activation layers' inability to detect the low level features like line segments, corners, curves and contour termination points correctly as part of the low level feature extraction can be attributed to mainly the unconstrained nature of spreading both temporally and spatially. In this section we discuss the basis for directed spreading which constrains the spreading spatially. The spreading takes place in specific predetermined directions and the directions specified by the input pattern. The directed spreading activation model locates the midpoints of lines of different lengths, curves and edge **termination** points in a purely datadriven manner.

When the input binary pattern is subjected to unconstrained spreading, the maxima points are formed at the line segments, corners, curves and contour termination points. If the diffusion is too short then these feature maxima are not formed correctly. On the other hand, if the diffusion is long then they move towards each other and merge. The nonstationary nature of the feature maxima is due to the lateral influence of the adjacent feature maxima.

The straight line segments and the corners may be considered as complementary features. Since the spreading is unconstrained these complementary feature peaks spread fast and become nonstationary. To avoid this lateral influence it is necessary to separate these complementary features. In this directed spreading activation model there are two surfaces which work in parallel and locate complementary features. One layer of neurons is sensitive to lines of different orientations and acts similar to Boundary Contour **System(BCS)** proposed by Grossberg[14]. The second parallel layer of neurons is similar to Feature Contour **System(FCS)** and is sensitive to curves and contour terminations. By proposing constrained spreading activation simultaneously taking place in two functionally complementary neuron nets, we isolate the complementary features and hence prevent the lateral influence of the feature maxima points.

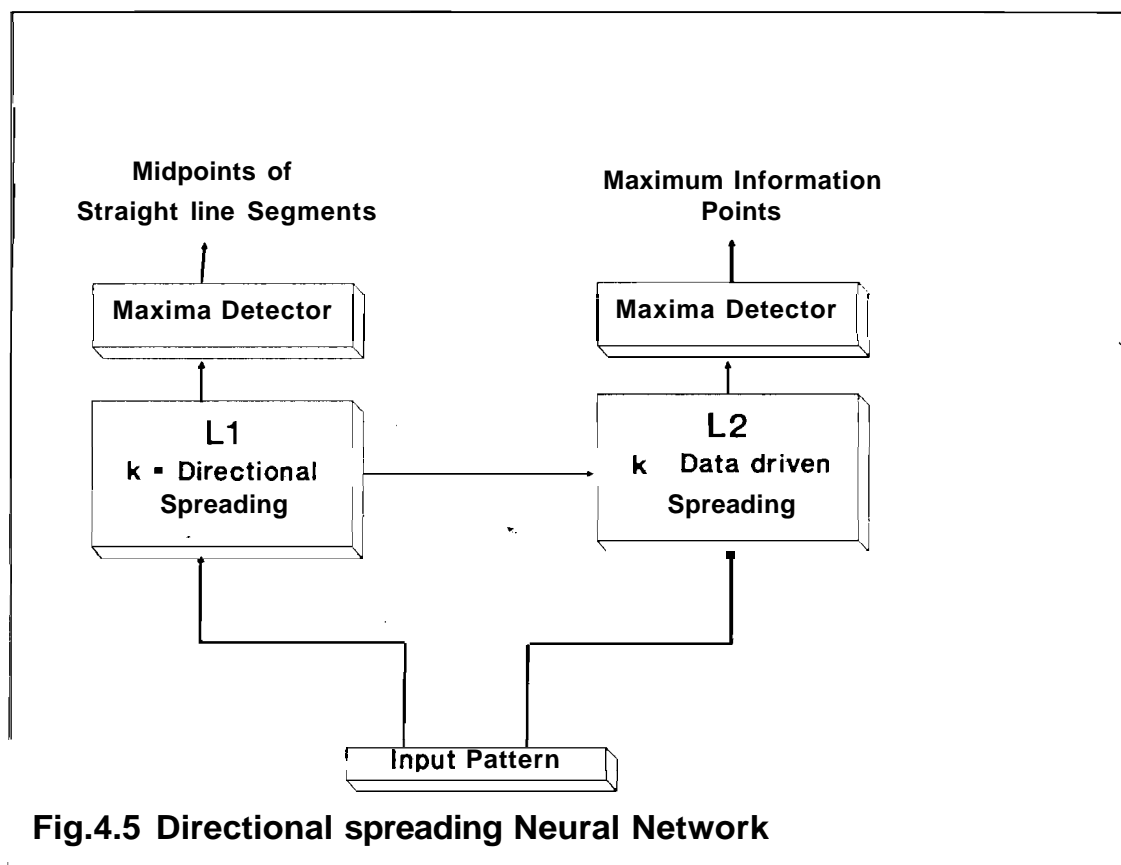
4.4 DIRECTED SPREADING ACTIVATION (DSA) LAYERS

In the directed spreading activation layers discussed in this section there are two layers each with different characteristic $k(R)$. The first layer has $k(R)$ defined for specific directions and spreading takes place only in these directions. It locates the midpoints of the line segments. The second layer receives its input from the first layer and the input binary pattern. In the second layer the spreading activation takes place in the direction specified by the activation values of the adjacent neurons. Hence the

conductivity function $k(R)$ of the region is directed by the data. This second layer detects curve centroids of all curvatures and contour terminations. Since the spreading in these two layers is spatially constrained there is no lateral influence between peaks, hence these peaks are always stationary and the movement is restricted to the directions specified within a layer. These two layers along with their maxima detectors locate midpoints of lines, curves, corners and contour terminations in a purely data-driven manner which can be used for **eye/camera** movement.

4.4.1 Organization of DSA Layers

The functional organization of the directed spreading activation layers is shown in Fig.4.5. It consists of two layers called **L1** and **L2** each of which consists of two dimensional array of neurons. In the case of **ORFIN** the layers are arranged in a hierarchy. In DSA both the layers receive the input simultaneously and send their outputs to a two dimensional array of neurons which locate the maxima points. The layer **L2** also receives input from **L1**. These two layers with their maxima locating network locate the complementary features in the input image. **L1** locates the midpoints of line segments and **L2** locates other maximum information points like corners, curve segments and contour termination points.



4.4.2 Design of DSA Layers

The first layer **L1**, consists of two dimensional array of **hypercolumns**[23]. A **hypercolumn** is a collection of orientation specific cells. Each cell in a hypercolumn responds to a specific orientation. The collection of cells is such that cells responding to all the orientations are available in a hypercolumn. In the current implementation each hypercolumn consists of a twelve directional detector neurons which respond to twelve different directions. A hypercolumn with twelve directional detectors is shown in Fig.4.6. These hypercolumns receive their input from the input binary pattern. The outputs of all the directional detector neurons are totally connected and these links have a small negative value. Hence when the input is presented each hypercolumn act like a winner take all network as shown in Fig.4.7. As a result, even though the directional detectors respond to partial line segments, the one which has the maximum response survives. All the directional detectors belonging to a hypercolumn receive **their** input from a fixed window of the input pattern. Adjacent hypercolumns receive their input from overlapping windows.

The general structure of the directional detectors is essentially the same as that of the S-cells described in Section 3.2.1. Each directional detector has two types of cells, excitatory cells (**ECs**) and the inhibitory cells (**ICs**) that occur in pairs. Each pair receives the same input set.

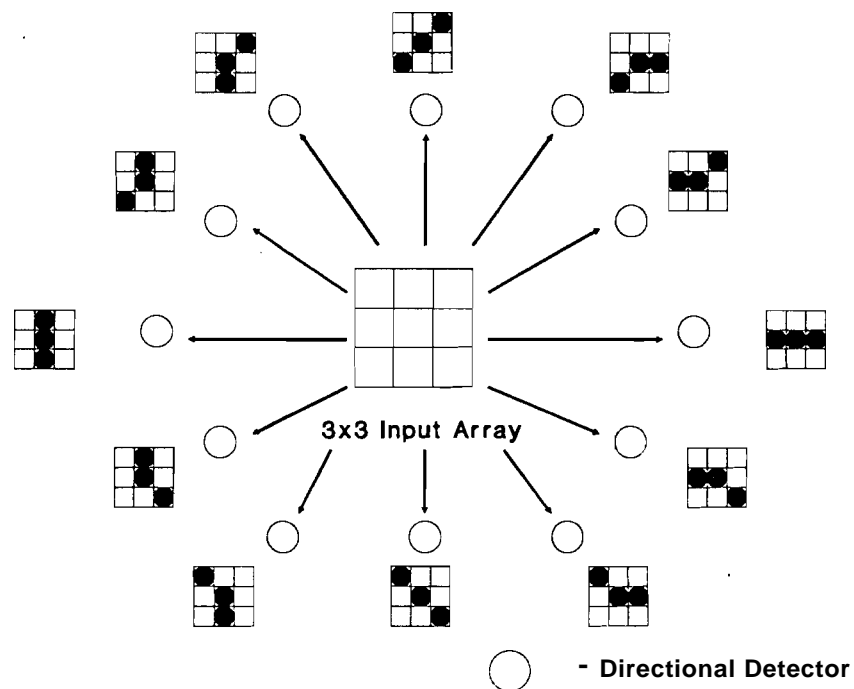


Fig.4.6 Layer one: Hypercolumn Input

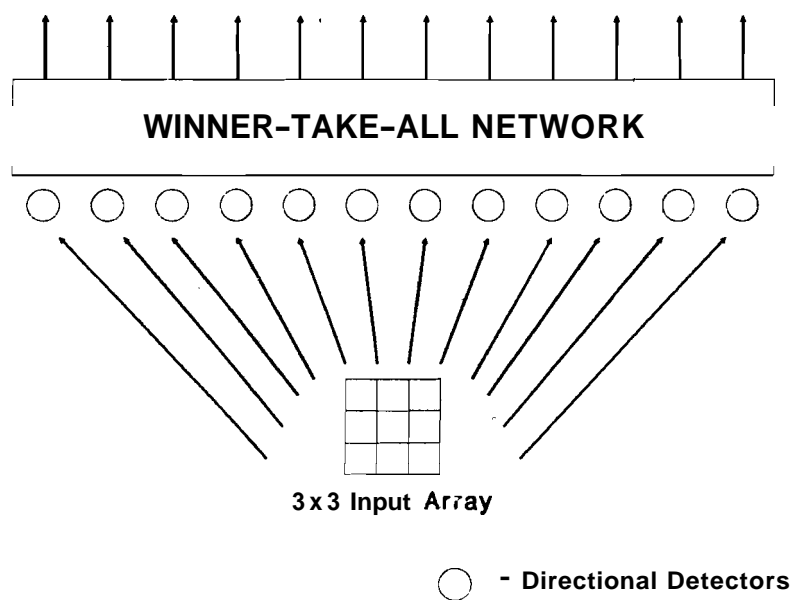


Fig.4.7 Layer one: Hypercolumn output

The **ICs** have fixed excitatory weights with values such that the output of the **ICs** is proportional to the mean intensity value over the input. The activation function of the **ICs** that produces this mean value is a simple weighted sum:

$$v_l = \sum_{i=1}^N c_l(i) I(i) \quad (4.2)$$

where the $c_l(i)$ values are determined by a function that decreases monotonically with distance from the center of the connectable area and sums to 1. The mean value v_l is used as inhibition to the paired EC, which generates an output according to the equation:

$$u_l = r * \varphi \left[\frac{1 + \sum_{i=1}^N a_l(i) * u(i)}{1 + \frac{r}{(r+1)} * b * v_l} - 1 \right] \quad (4.3)$$

where the weights a_l and b are modifiable weights, r represents the efficacy of the inhibitory **synapse** and the transfer function is a piecewise linear function according to:

$$\varphi(x) = \begin{cases} x/(\alpha+x) & \text{if } (x>0) \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

The functional characteristics of directional detector is summarized in Fig.4.8.

The directional detectors which have the same directional sensitivity of neighboring hypercolumns are connected by a link. An example of the hypercolumns connected through the links is illustrated in Fig. 4.9. In the illustration six hypercolumns with each hypercolumn having only four directional detectors are shown. The directed spreading takes

place through these links. Hence the $k(R)$ defined for **L1** is sensitive to the direction. The output of the layer **L1** is connected to the maxima detector. This network is a simple **on-center/off-surround** network to detect maxima. Each maxima detector cell suppresses the neighboring neurons according to its activation and feeds back excitatory activation to itself.

The second layer **L2** also consists of two-dimensional array of neurons. These cells are connected to all their neighbors by links. Each neuron receives its activation from the input and the first layer according to the following equation:

$$L2_{x,y} = I_{x,y} - L1_{x,y} \quad (4.5)$$

where $L2_{x,y}$ is the activation value fed to the neuron of **L2**, $I_{x,y}$ is the input binary pattern and $L1_{x,y}$ is the activation values of **L1**. From the equation it is clear that the second layer receives complement of the first layer output over the input binary pattern. All the inputs and outputs of a single neuron in **L2** is shown in Fig.4.10. Since the first layer detects all the lines and diffuses them, the second layer receives activations at corners, curves of all curvatures other than straight lines and contour terminations. In layer **L2**, the spreading takes place between only the active neighboring neurons. So the corner, curve and contour

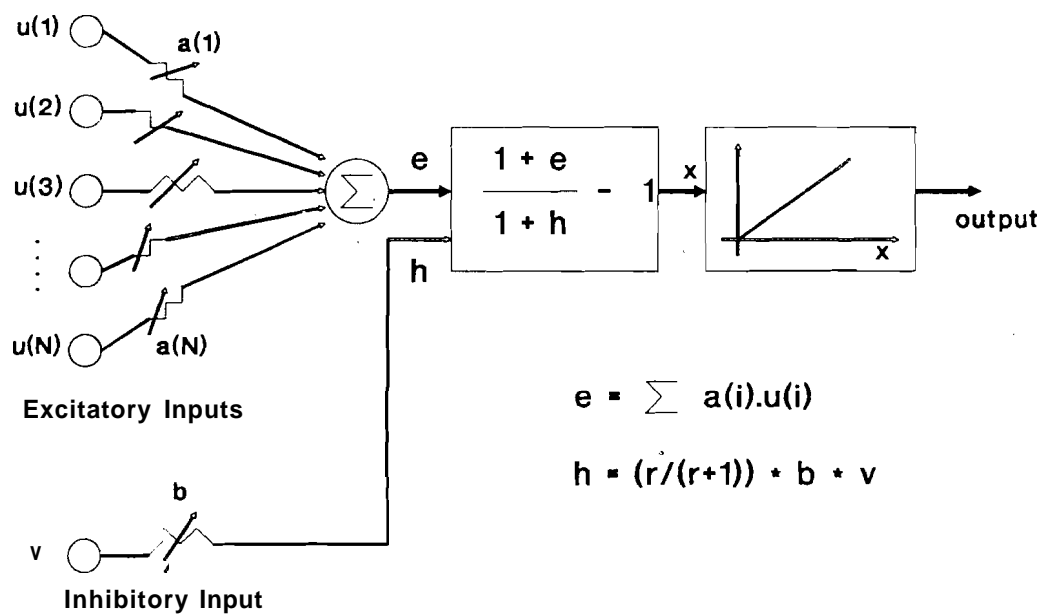


Fig. 4.8 Characteristics of a Directional detector

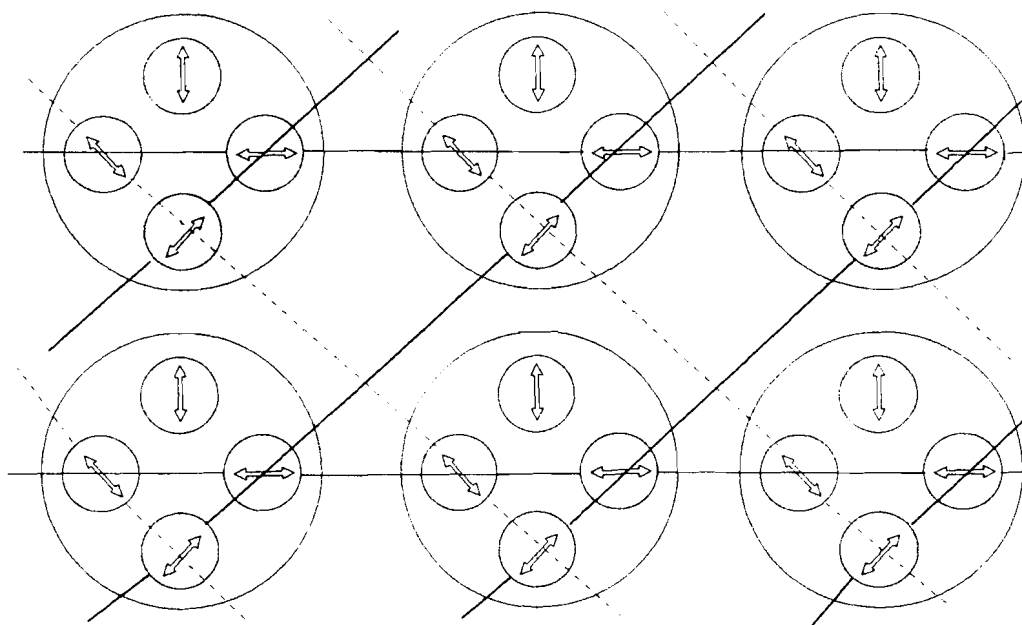


Fig. 4.9 Illustration of links between hypercolumns

termination centroids are enhanced. The output of L2 is fed to the maxima detector and the maxima detector locates the enhanced peaks of L2.

Rapid eye movements(saccades) driven by the locations of maximum information points play an important role in the establishment of spatial relations. The absolute and relative positions of the peaks located by L1 and L2 of this system can be considered as bottom-up cues for the **eye/camera** movement to establish the spatial relationships. The peak strength shows the length of a line or a curve at that position. The 'on' pixels around the fixed window of the peak is useful for identification of the feature at the peaks.

4.5 IMPLEMENTATION DETAILS AND EXAMPLES

The input visual pattern is a 32x32 two-dimensional array of binary values. There are twelve directional detectors in the **hypercolumn** structure as shown in Fig.4.6. These directional detectors compute their activation values following the eqn(4.3) . The parameters for the directional detectors are fine tuned and these values are $r = 1.7$ and $b = 1.0$.

The **L1** layer receives the maximum value of each hypercolumn. This **L1** layer is implemented in an array of size 31x31, giving an offset of one for computing the directional detectors. The directional spreading takes place in **L1** layer.

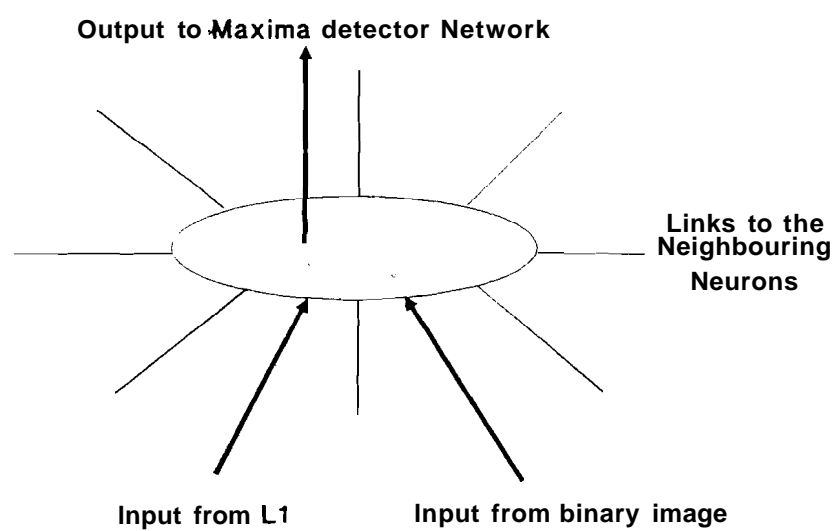
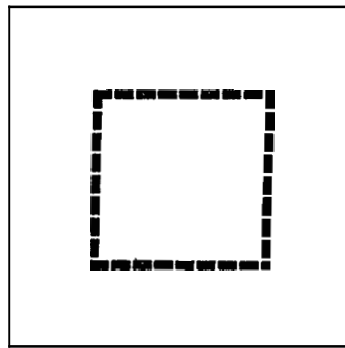


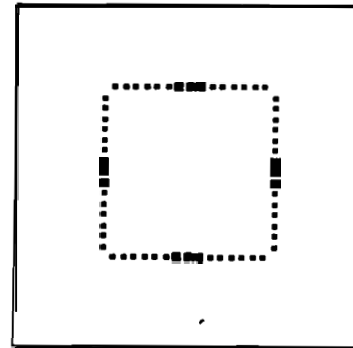
Fig. 4.10 Input/Output of a neuron in L2.

The spreading activation coefficient k is taken to be 0.005. The L2 layer receives the complement of **L1** over the input array.

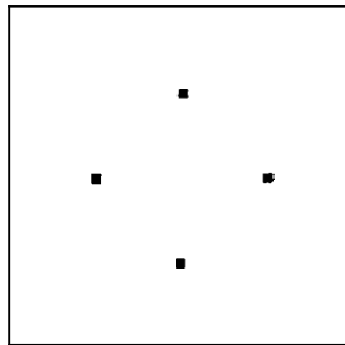
Fig.4.11 shows an example of the input binary pattern for which maximum information points are generated. Figs.4.11b to 4.11e show the outputs of different layers. Fig.4.11a shows the input pattern for which maximum information points are to be located. **Fig.4.11b** shows the spreading taken place in specific directions. The centers of the line segments have the maximum activation which is shown in Fig.4.11 c. Fig.4.11d shows the complementary of **L1** values to the input image. Since the adjacent values to these corners are very large in **L1** layer, the complement becomes too small and hence the adjacent values are not seen in **Fig.4.11d**. In this binary pattern the maximum information points are the corners. These points are automatically located by the architecture and is shown in Fig.4.11e. It can be observed that even though this architecture does not have any corner or any other template, it locates the corners and other maximum information points automatically, This is an advantage for locating low level features from machine fonts which is illustrated in the next section. Fig.4.12 shows another example of low level feature extraction from another binary pattern.



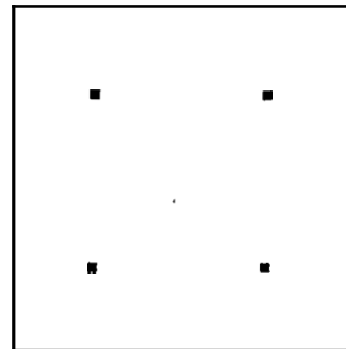
(a) Input Binary Image



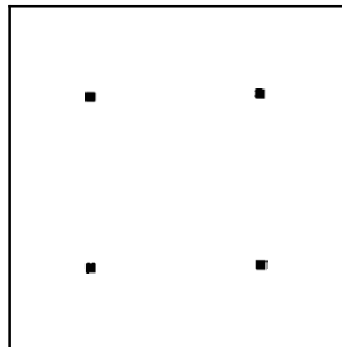
(b) Layer L1 output values



(c) Maxima points in L1

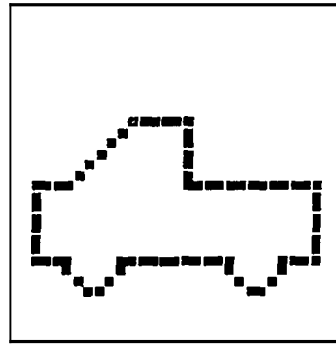


(d) Layer L2 output values

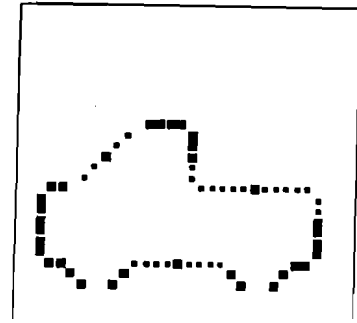


(e) Maximum information points for the input image

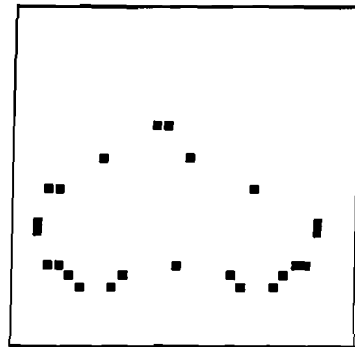
Fig. 4.11 Example-I. Outputs of different stages of directed spreading activation layers for a square.



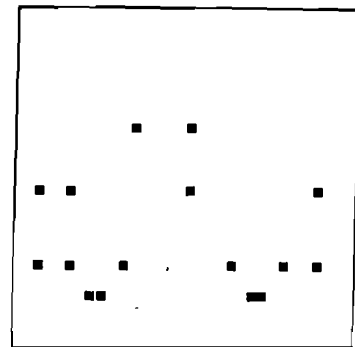
(a) Input Binary Image



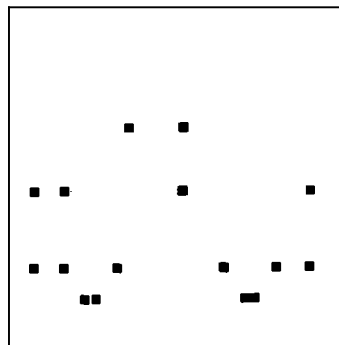
(b) Layer L1 output values



(c) Maxima points in L1



(d) Layer L2 output values



(e) ~~Maximum~~ information points for the input image

Fig. 4.12 Example-2. Outputs of different stages of directed spreading activation layers for the jeep shown in (a).

4.6 APPLICATION OF DSA LAYERS TO LOW LEVEL FEATURE EXTRACTION FROM MACHINE FONTS

In this section an application of the directed spreading activation layers is discussed. The application considered is low level feature extraction from machine printed fonts for recognition. This is one of the cases of visual patterns where low level feature extraction can carry out significant amount of data reduction in a purely data-driven manner.

Machine recognition of characters continues to be a problem even when the number of characters is limited, and the characters are restricted to machine printed characters. When the machine printed character set involves different fonts, it becomes very difficult to design a recognition system which works for all the fonts. The brute force approach to this problem could be to store all possible characters of all fonts in the long term memory and compare them with the test input one by one. This not only requires a large amount of long term memory but also the comparison time increases exponentially as the number of fonts to be recognized increases.

In all the previous approaches for machine font recognition, the low level features are fixed 'a priori'. In other words the feature extraction phase is model driven. Generally these features are small straight line and curve segments. Since these low level features are fixed, significant amount of information is lost in the feature

extraction phase resulting in the reduction of recognition accuracy. Attempting to extract all the features with this approach involves not only manual extraction of low level features from all the fonts but also large amount of storage space and comparison time. The ideal case would be to find a mechanism to evolve these features from the data itself. Then all the features can be captured without any loss.

The directed spreading activation discussed in the last section could be used successfully for this problem. Directed spreading activation layers locate the low level features like straight lines, curves, corners and contour termination points in a purely data-driven manner. From these locations the low level features can be extracted. There are other advantages to the low level feature extraction by the directed spreading activation. The low level feature extraction by **directed** spreading activation layers is translation invariant. Hence the low level features from fonts located at any part of the input visual pattern can be extracted. Some examples of extracting low level features from printed alphabets are shown in Fig.4.13.

The feature map which is generated from the locations of the low level features can be used for invariant representation. Since the feature map is a compressed representation of each one of the characters, this can be used for scale, rotation and translation invariant representation of the character and can be used for recognition.

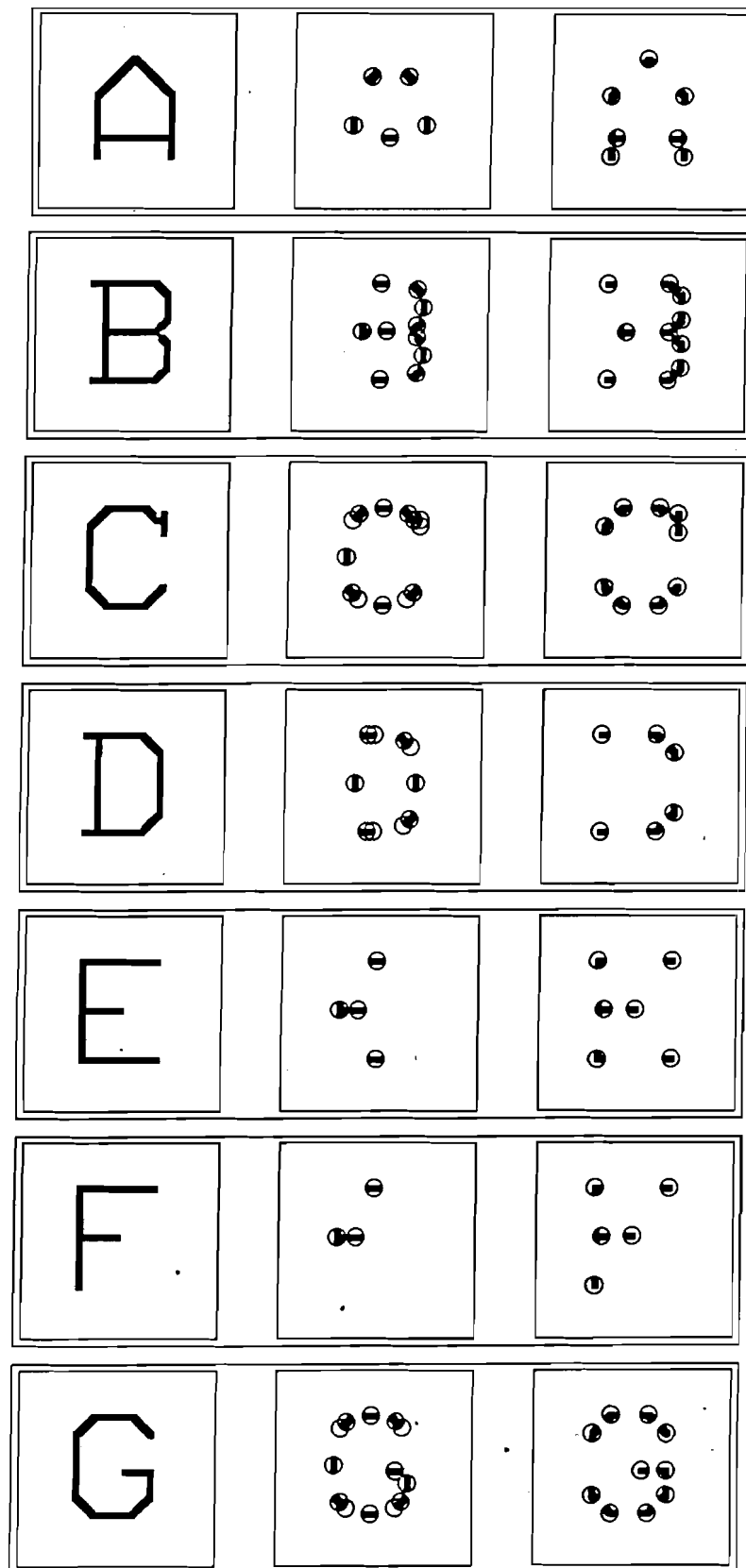


Fig. 4.13 This figure shows examples of low-level **features** located for machine fonts. In each row the first block shows the input character. The second and third blocks show the outputs of layers L1 and L2.

4.7 APPLICATION OF DSA LAYERS FOR TRANSFORMATION INVARIANT BINARY PATTERN RECOGNITION

The maximum information points simplify the analysis of images by drastically reducing the amount of data to be processed while at the same time preserving important information about the **object**[1]. In this section we describe a method to recognize the binary image patterns subjected to affine transforms, from the maximum information points generated by the directed spreading activation layers.

The maximum information points in the image space are denoted using complex notation as

$$Z = (x-x_c)+i(y-y_c) \text{ or } Z = ae^{i\theta} \quad (4.6)$$

where (X_c, Y_c) is the location of the centroid, $a = |Z|$, and $\theta = \arg Z$. The conformal mapping, $\ln Z$, has the effect of transforming both rotation and scale effects to translations in the transformed space. If Z is multiplied by a scale factor a , then $\ln Z = (\ln \sigma) + (\ln a) + i\theta$ and if Z is rotated through an angle β , then

$$\ln Z = \ln (ae^{i(\theta+\beta)}) = \ln \sigma + i(\theta+\beta).$$

Rotation around the centroid on the visual field becomes translation with respect to θ . Scaling becomes translation with respect to $\ln a$. Thus, if the centroid in the θ and $\ln a$ directions can be determined, then the effects of scaling and rotation can be eliminated by translating the log-polar space with respect to the centroid. The logarithmic

transformation also has the desirable side effect of emphasizing the importance of the central visual area by compressing radially distant features when the input is uniformly sampled. The centroid in the log-polar space moves along with the features so that it stays in the same place with respect to the features.

This transformation generates a two dimensional array of points which can be used for recognition using simple pattern matching technique. The pattern matching technique must take care of the small positional errors which arise due to the quantization errors. A possible limitation is that, as a result of rotation, features may move off the left or right edges of the log-polar map. A wrapping may occur due to the 2π periodicity of the log-polar mapping.

Fig.4.14 illustrates the log-polar mapping for scaled and rotated binary images. **Figs.4.14a** to **4.14c** show the normal, scaled and rotated binary images. The corresponding maximum information points located by DSA layers are shown in **Figs.4.14d** to **4.14f**. **Figs.4.14g** to **4.14i** show the log-polar mapping for the maximum information points. In these three images the x-axis is θ and y-axis is $\ln r$. In **Fig.4.14h** it can be observed that this output is translated in the y-axis since the input image is scaled. Similar translation can be observed in **Fig.4.14i** where the input image is rotated. Here the translation has taken place with respect to θ . **Fig.4.14j** to **4.14l** show the log-polar transformed images in which the effects of scaling and rotation are eliminated by translation

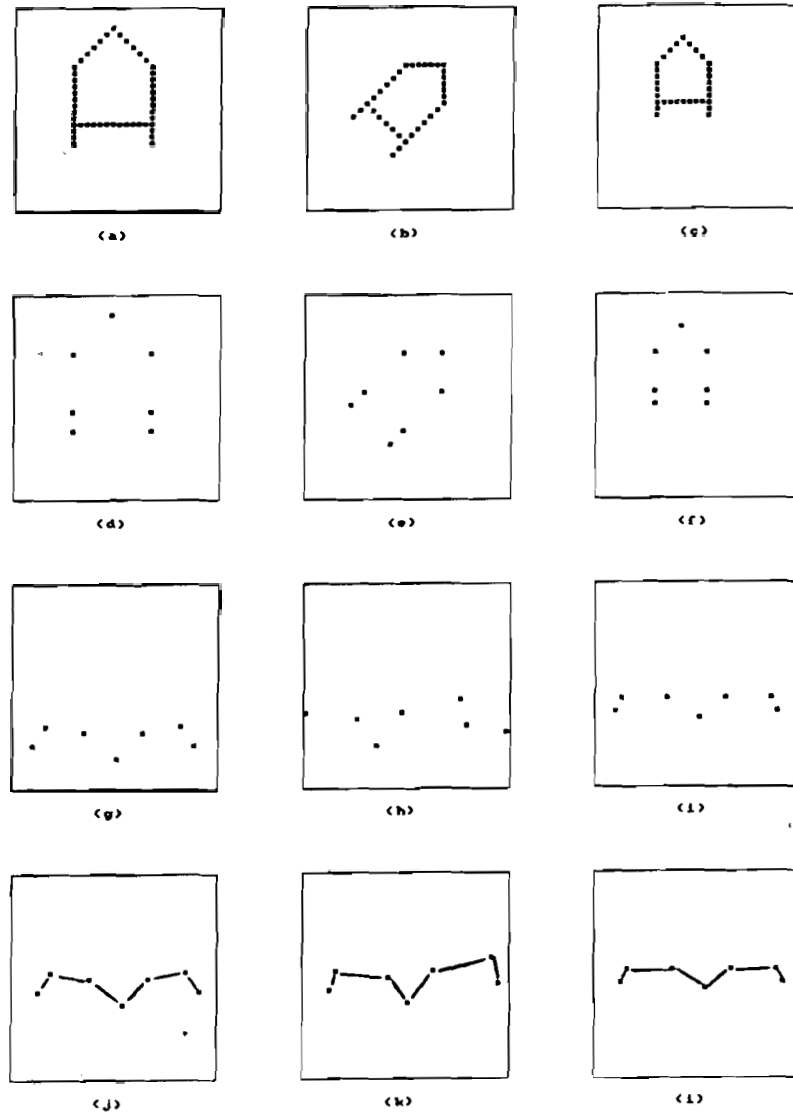


Fig. 4.14 This figure shows the scale and rotation invariance achieved by the log-polar mapping. (a)-(c) show the normal, rotated and scaled binary images. The corresponding maximum information points generated by directed spreading activation layers are shown in (d)-(f). The log-polar mapping for the maximum information points of (d)-(f) is shown in (g)-(i). In (g)-(i) the x-axis is θ and y-axis is $\ln r$. (j)-(l) show the images in which the effects of scaling and rotation are eliminated.

in the corresponding axes. Figs.4.15 to 4.17 show some more examples of log-polar transformed binary images. It can be observed that there are some distortions in the final translated image. This arises not only due to distortions in the shape of the input image and but also due to the quantization errors.

4.8 APPLICATION OF DSA LAYERS TO IMAGES OF FORMANT CONTOUR PATTERNS

DSA layers perform well in the class of patterns where the low level features are characterized well. In these situations the output of **DSA** layers can be used directly for recognition. On the other hand, for the class of patterns where the features are not characterized well, for example images of formant contour patterns, the output generated by **DSA** layers **cannot** be immediately used for recognition. The other knowledge source about the patterns are required.

Fig.4.18 shows the output of **DSA** layers for an image of a formant contour pattern. Fig.4.18a shows the formant contour pattern for an isolated utterance of the word TWO. Fig.4.18b shows the midpoints of the straight line segments in the input pattern and Fig.18c showsthemaximum information points located in the input pattern. It can be observed that processing carried out by **DSA** layers reduces the initial information significantly. But this processing has a number of disadvantages. The spurious formant values in the input pattern gets reflected as maximum information points **which**

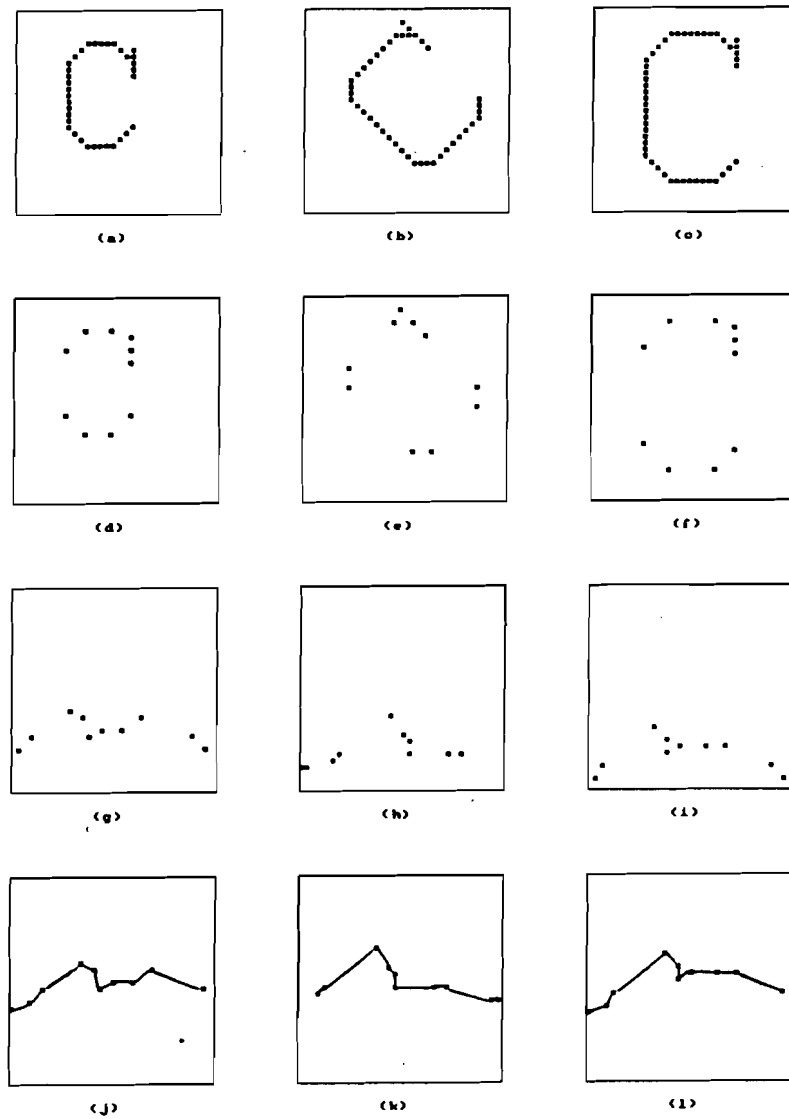


Fig. 4.15 Log-polar mapping for a binary image: Example-2. (a)-(c) show the normal, rotated and scaled images and (d)-(e) show the corresponding maximum information points located by DSA layers. (f)-(h) show the log-polar mapping for the maximum information points. (j)-(l) show the log-polar domain images where the effects of scaling and rotation are eliminated.

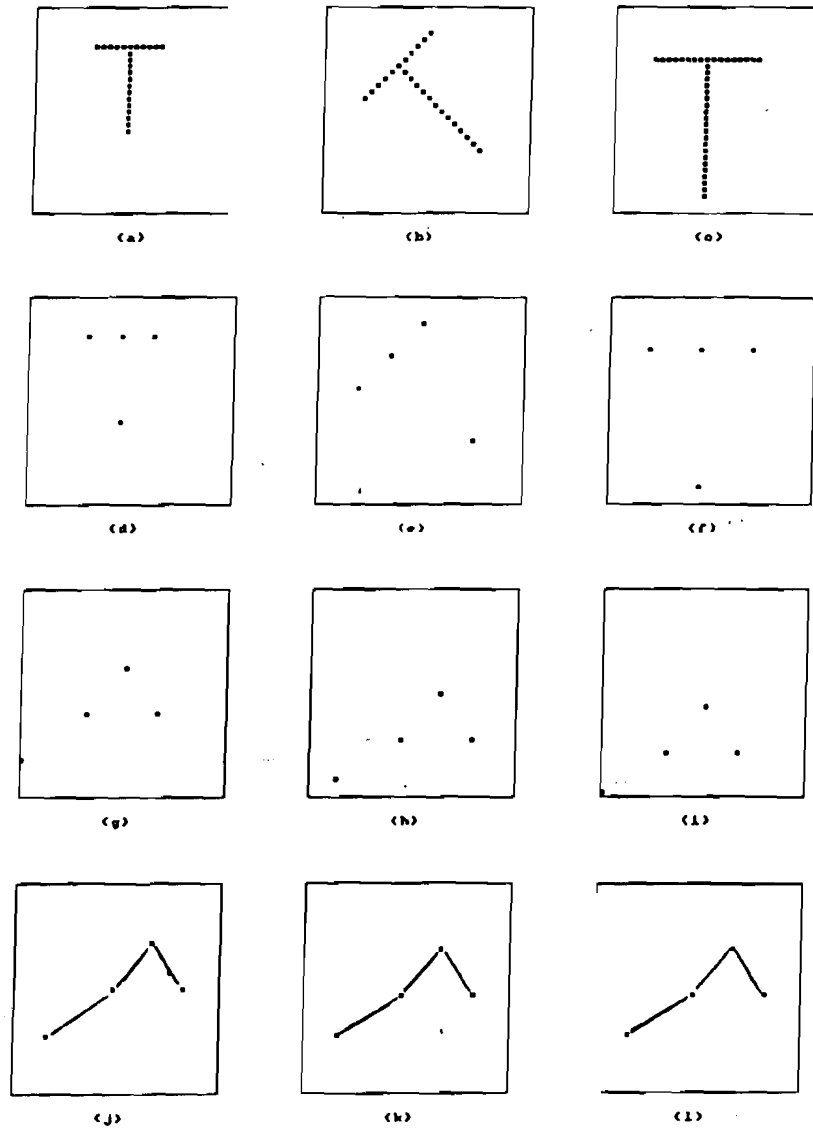


Fig. 4.16 Log-polar mapping for a binary image: Example-3. (a)-(c) show the **normal,rotated** and scaled images and (d)-(e) show the corresponding maximum information points located by DSA layers. (f)-(h) show the log-polar mapping for the maximum information points. (j)-(l) show the log-polar domain images where the effects of scaling and rotation are eliminated.

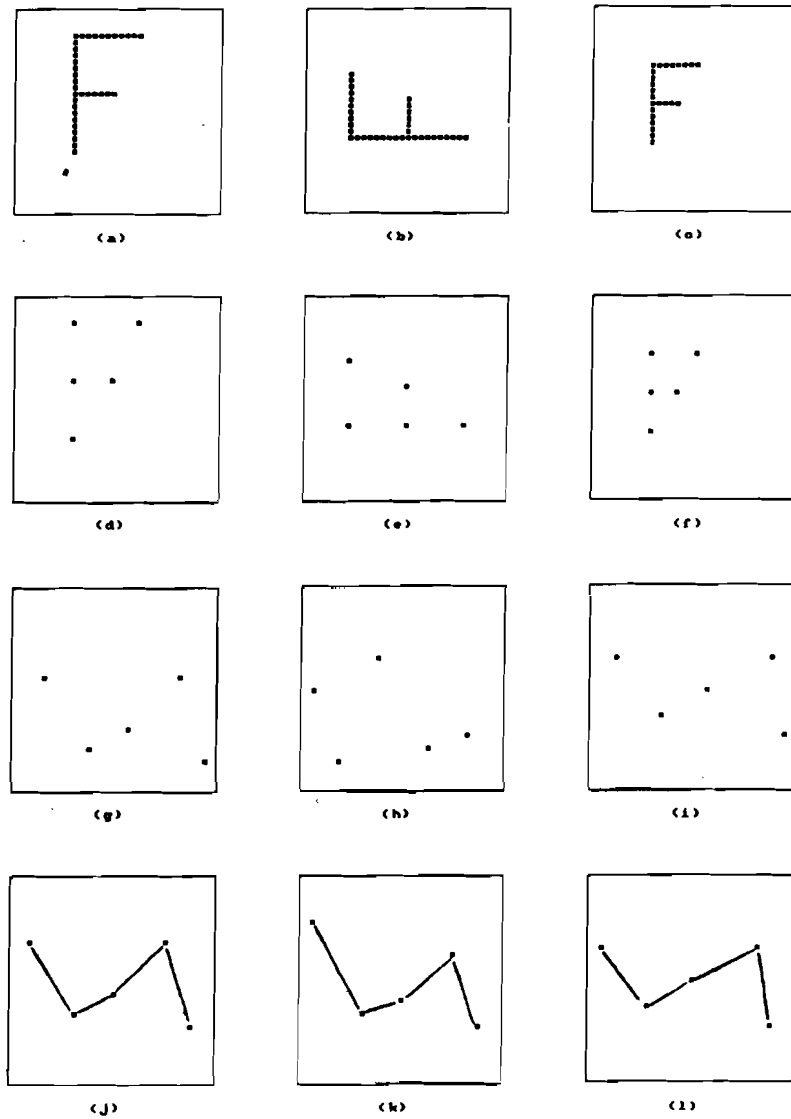
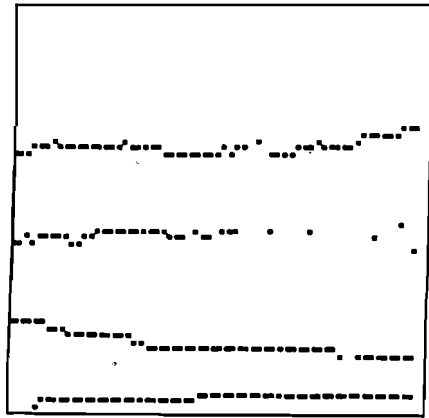
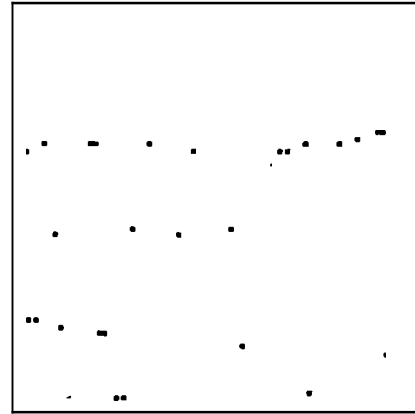


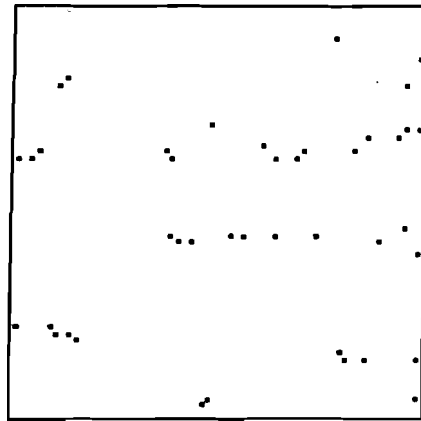
Fig. 4.17 Log-polar mapping for a binary image: Example-4. (a)-(c) show the normal, rotated and scaled images and (d)-(e) show the corresponding maximum information points located by DSA layers. (f)-(h) show the log-polar mapping for the maximum information points. (j)-(l) show the log-polar domain images where the effects of scaling and rotation are eliminated.



(a) Formant Plot: Two



(b) Output of Layer L1



(c) Output of Layer L2

Fig. 4.18 Output of DSA layers for formant contour pattern is shown. (a) shows the formant contour pattern for the utterance TWO. (b) shows the midpoints of the straight line segments and (c) shows the maximum information points located by DSA layers.

are difficult to eliminate in the later stages. The discontinuities in the line segments due to missing formant also affect significantly leading to multiple maximum information points. Hence it is necessary to get a formant contour pattern which is free from noise and discontinuities. This requires some issues which need to be addressed at the time of extraction of formant contour itself.

The problem in the extraction of formants from the speech signal can be attributed to the block processing. Block processing processes the speech signal by blocks containing fixed number of samples. This block processing assumes that the input signal is stationary which implies that the formants do not change within the block. But speech signal is nonstationary and formant frequencies undergo change in a much shorter time especially during formant transitions. Using smaller block size leads to poor frequency resolution which affects the accuracy of the extracted formant frequency. The other approach is to use pitch synchronous analysis. Here we assume that the signal is stationary for one pitch period. But reliable pitch extraction is also very difficult.

In this work we have attempted processing synthetic formant contour patterns. Fig.4.19 shows the processing of synthetic formant contour patterns using DSA layers. Figs.4.19a to 4.19c show some synthetic formant contour patterns. Figs.4.19d to 4.19e show the midpoints of straight line segments and Figs.4.19f to 4.19h show the maximum

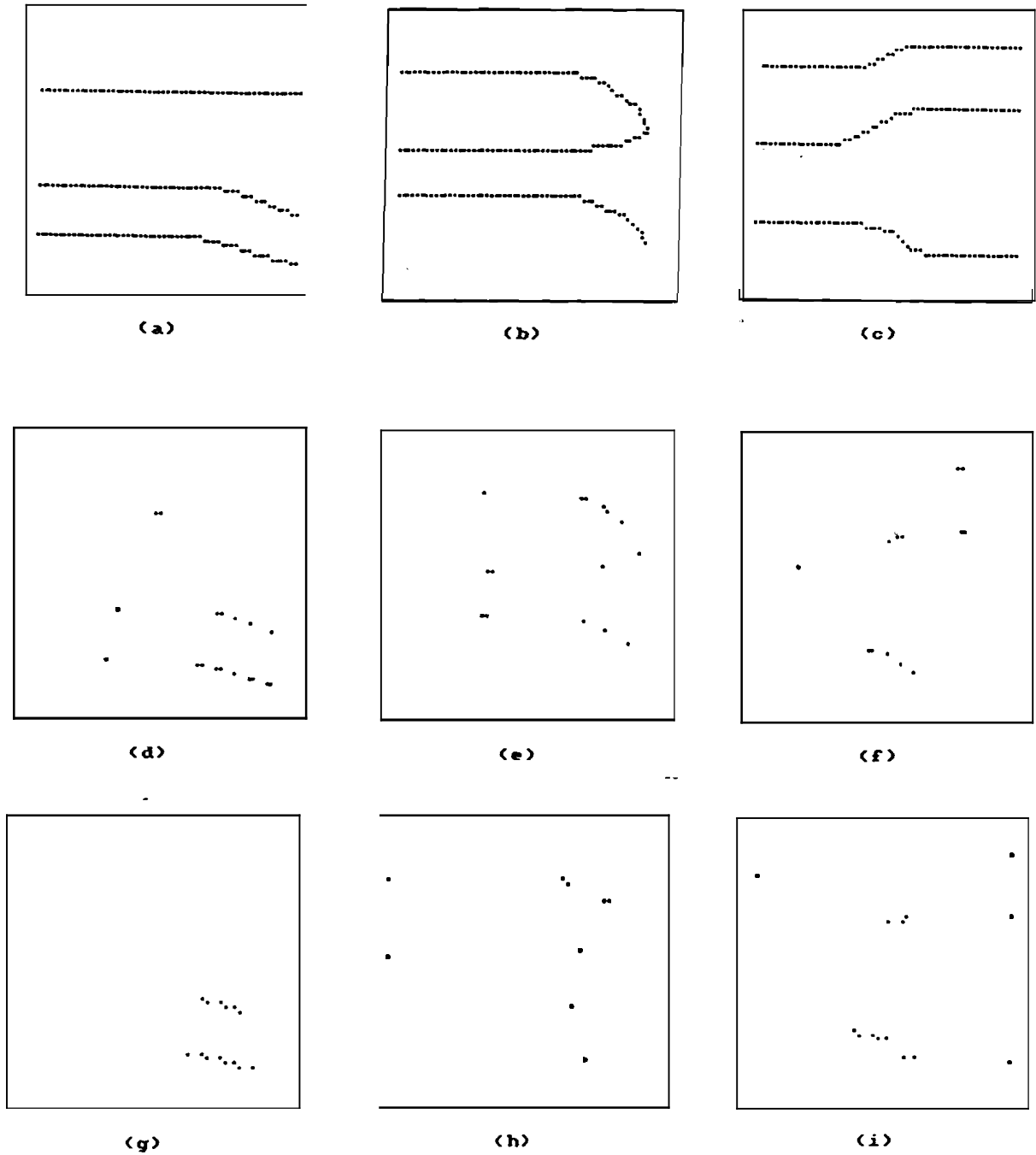


Fig. 4.19 This figure shows the output of DSA layers for processing synthetic formant contour patterns. (a)-(c) show examples of synthetic formant contour patterns. (d)-(e) show the midpoints of straight line segments and (f)-(h) show the maximum information points generated from these synthetic formant patterns. These **information** points may be used as features for developing a recognizer.

information points generated from the synthetic formant pattern. These information points may be used as features for developing a recognizer.

4.9 SUMMARY

In this chapter we have proposed a new neural architecture for automatic location of maximum information points. The spreading activation layers and its utility for early vision tasks were discussed. The difficulty in using the spreading activation layers for low level feature extraction was explained. We have pointed out the importance of the directed spreading for low level feature extraction, and described a new directed spreading activation model. Two applications of the model, low level feature extraction for machine fonts and transformation invariant binary pattern recognition were also discussed.

SUMMARY AND CONCLUSIONS

Visual pattern recognition can be considered as consisting of two stages: (i) a preprocessing stage which primarily concerns with the data reduction by extracting geometric properties like straight lines and corners and (ii) a recognition stage which recognizes the **familiar objects**. Preattentive visual processing concerns with extracting geometric properties from input image with parallel, automatic and data-driven mechanisms. Artificial neural networks with their parallel, **nonsymbolic**, fault tolerant computing are useful to achieve preattentive visual processing. In this thesis we have developed neural network architectures for extracting geometric properties from binary images.

Since artificial neural networks are reminiscent of biological neural mechanisms they attempt to derive motivation from biological systems and model their structural and functional characteristics. **Neurophysiological** and visual perceptual evidences reported in the literature are useful for such modelling. The main motivation of our design of the neural networks for preattentive visual processing has come from some aspects of visual perception in biological visual system. Based on the observations we have proposed two approaches for preattentive visual processing and we have

proposed two architectures based on these two approaches. First approach extracts straight line segments from the input image and this was implemented in an oriented filtering and integration(**ORFIN**) network. Second approach extracts the maximum information points from the input image and this was implemented in directed spreading **activation(DSA)** layers.

Though these two architectures are functionally totally different we have shown in this thesis that these two differ in a simple neurocomputing property like lateral interaction. **ORFIN** is a two-stage hierarchical network which does not have any lateral interaction or cooperation between adjacent neurons. DSA layers have two-stages which receive the input parallelly and have lateral interaction between adjacent neurons. One of the layers has lateral interaction depending on the direction and the other layer has lateral interaction depending on the input data.

We have considered some applications for these architectures. The first application is recognizing the isolated utterances of words from the images of formant contour patterns. Here the problem is to get an invariant representation from the input binary image. We have used **ORFIN** to preprocess to get an invariant representation from the input image. The recognition in this case is implemented using a two-stage hierarchical adaptive resonance architecture. We have tested the system with isolated

utterances of digits. We have conducted two tests, with a single speaker and with multiple speakers and the test results are shown.

The other application uses the **DSA** layers. **DSA** layers extract **maximum** information points from the input image and at these points there are low level features like corners, curves and **cotour** termination points. Hence the output of **DSA** layers can be used in two ways. One way is to use the low level features for recognizing the input pattern. This approach works successfully in images where the low level features are characterized well, for example machine fonts. We have shown examples of such low level feature extraction from machine fonts. The other way is to use the spatial relationship between different parts of the image by using only the locations of the maximum information points. We have also shown in this thesis a methodology to recognize affine transformed images from the maximum information points generated by the **DSA** layers.

LIST OF FIGURES

- Fig. 3.1 This figure illustrates the structural organization of **ORFIN**. (a) shows the block diagram of **ORFIN** and (b) illustrates the interconnection between S-planes and C-planes. Outputs of two of the S-planes which have the same orientation of stimuli but trained differently are fed to corresponding C-planes. This is shown as outputs from two S-planes converging into a single C-plane. (c) illustrates examples of S-cells whose outputs are fed to corresponding C-cells.
- Fig. 3.2 Interconnections **converging to** a S-cell.
- Fig. 3.3 Schematic diagram illustrating the interconnections between the two stages.
- Fig. 3.4 Input-to-output characteristics of a S-cell.
- Fig. 3.5 Twelve line segments used to train S-cells.
- Fig. 3.6 Fixed weight pattern between S-cells and C-cells. This pattern is responsible for handling small shifts in the input visual pattern.
- Fig. 3.7 Some examples of images of formant contour patterns.
- Fig. 3.8 Neural architecture for recognizing isolated utterances of words. First stage extracts structural features using **ORFIN**. Second stage implements two-stage Simple Adaptive Classifiers for recognition.
- Fig. 3.9, Pattern Matching stage is a hierarchical adaptive resonance network. SAC-1 categorizes the profiles. SAC-2 classifies based on the categorization done by SAC-1.
- Fig. 3.10 ART 1 Schematic diagram
- Fig. 3.11 Image of a formant contour pattern compressed into 64x64 array is shown in (a). The output values of the five C- planes are shown for the example input pattern. The size of the block in (b)-(f) indicates the value of C-cell at that point.

- Fig. 4.1** The activity distribution of the spreading activation layers is plotted in three dimensions at four times: (a) at t_0 +after a short time; and much later in (c) and (d). In (d) the peak is located at the geometric centroid of the three features as shown in (a).
- Fig. 4.2** As time progresses (a) to (f), the activity distributions due to two features spread. As activity spreads the local maxima moves toward the centroid. The global maxima is stable at that point.
- Fig. 4.3** A contour with a corner shown in (a) is diffused in (b)-(d). Activity accumulates more quickly where the average distance to the features is least. As the diffusion progresses the activity maxima moves to the global centroid. Since maxima moves continuously it is difficult to determine when to stop locate the peak at the corner.
- Fig. 4.4** Small scale organization of feature clusters emerges before large scale organization. (a) shows the continuous process of feature clustering. (b)-(d) shows different snapshots of feature clustering at different times.
- Fig. 4.5** Directional spreading neural network
- Fig. 4.6** Layer one: Hypercolumn Input
- Fig. 4.7** Layer one: Hypercolumn Output
- Fig. 4.8** Characteristics of a Directional detector
- Fig. 4.9** Illustration of links between hypercolumns
- Fig. 4.10** Input/output of a neuron in L2.
- Fig. 4.11** Example- 1: Outputs of different stages of directed spreading activation layers for a square.
- Fig. 4.12** Example-2: Outputs of different stages of directed spreading activation layers for a jeep shown in (a).

- Fig. 4.13 This figure shows examples of low-level features located for machine fonts. In each row the first block shows the input character. The second and third blocks show the outputs of layers **L1** and **L2**.
- Fig. 4.14 This figure shows the scale and rotation invariance achieved by the log-polar mapping. (a)-(c) show the normal, rotated and scaled binary images. The corresponding maximum information points generated by directed spreading **activation** layers are shown in (d)-(f). (g)-(i) show the log-polar mapping for the maximum information points of (d)-(f). In (g)-(i) the x-axis is θ and y-axis is $\ln r$. (j)-(l) show the images in which the effects of scaling and rotation are eliminated.
- Fig. 4.15 Log-polar mapping for a binary image: Example-2. (a)- (c) show the **normal,rotated** and scaled images and (d)-(e) show the corresponding maximum information points located by **DSA** layers. (f)-(h) show the log-polar mapping for the maximum information points. (j)-(l) show the log-polar domain images where the effects of scaling and rotation are eliminated.
- Fig. 4.16 Log-polar mapping for a binary image: Example-3. (a)- (c) show the **normal,rotated** and scaled images and (d)-(e) show the corresponding maximum **information** points located by **DSA** layers. (f)-(h) show the log-polar mapping for the maximum information points. (j)-(l) show the log-polar domain images where the effects of scaling and rotation are eliminated.
- Fig. 4.17 Log-polar mapping for a binary image: Example-4. (a)- (c) show the **normal,rotated** and scaled images and (d)-(e) show the corresponding maximum information points located by **DSA** layers. (f)-(h) show the log-polar mapping for the maximum information points. (j)-(l) show the log-polar domain images where the effects of scaling and rotation are eliminated.
- Fig. 4.18 Output of **DSA** layers for formant contour pattern is shown. (a) shows the formant contour pattern for the utterance TWO. (b) shows the midpoints of the straight line segments and (c) shows the maximum information points located by **DSA** layers.

Fig. 4.19 This figure shows the output of DSA layers for processing synthetic formant contour patterns. (a)-(c) show examples of synthetic formant contour patterns. (d)-(e) show the midpoints of straight line segments and (f)-(h) show the maximum information points generated from these synthetic formant patterns. These information points may be used as features for developing a recognizer.

LIST OF TABLES

- Table 3.1 Isolated Word Recognition System test results for a single speaker.
- Table 3.2 Isolated Word Recognition System test results for two speakers: Speaker-1.
- Table 3.3 Isolated Word Recognition System test results for two speakers: Speaker-2.

PUBLICATION

- [1] A.Arul Valan and B.Yegnanarayana, "Directed Spreading Activation in Multiple layers For Low level Feature Extraction," Proceedings of International symposium on information theory and its applications, Singapore, pp.563-567, November, 1992.

REFERENCES

- [1] Attneave, F., "Some informational Aspects of Visual Perception," Psychological Review, vol.61, no.3, pp. 183- 193, 1954.
- [2] Allan, M. C. and Elizabeth, F. L., "A Spreading Activation theory of semantic processing," Psychological review, Vol. 82, pp. 407-428, 1975.
- [3] Alt, F., "Digital Pattern Recognition by moments," Journal of the ACM, Vol. 9, pp. 240-258, 1962.
- [4] Breitmeyer, B.G., "Eye movements and visual pattern perception," In E. C. Schwab & H. C. Nusbaum (Eds.) Pattern recognition by humans and machines, Vol.2., pp.65-86, New York: Academic Press, 1986.
- [5] Burt, P.J., "Smart sensing within pyramidal vision machine," Proceedings of the IEEE, Vol. 76, pp. 970-981, 1988.
- [6] Carpenter, A., and Grossberg, S., "ART 2: Self- organization of stable category recognition codes for analog input pattern," Applied optics, Vol. 26, No. 23, pp.4919-4930, December 1987.
- [7] Casey, R.G., "Moment normalization of Handprinted Characters," IBM Journal of Research and Development, pp. 548-557, September 1970.
- [8] Deborah, K. W. Walters, "A Computer vision model based on Psychophysical experiments," In E. C. Schwab & H. C. Nusbaum (Eds.) Pattern recognition by humans and machines, Vol.2. pp.87- 120, New York: Academic Press, 1986.
- [9] Didday, R.L., and Arbib, M.A. "Eye movements and Visual Perception: A two visual system model," International Journal of Man-Machine Studies, Vol. 7, pp. 547-569, 1975.
- [10] Dudani, S.A., Breeding, K.J., and McGhee, R.B., "Aircraft Identification by Moment Invariants," IEEE Transactions on Computers, Vol. C-26, No.1, pp.39-46, January 1977.
- [11] Eugene, C.F., "Knowledge-Mediated Perception," In E. C. Schwab & H. C. Nusbaum (Eds.) Pattern recognition by humans and machines, Vol.2., pp.219-236, New York: Academic Press, 1986.

- [12] Fukushima, K., and Miyake S., "NEOCOGNITRON: A new algorithm tolerant of deformations and shifts in position," Pattern recognition, Vol.15, No.6, pp. 445-469, 1982.
- [13] Fukushima, K., "Cognitron: a self-organizing multilayered neural network," Biological Cybernetics, Vol. 20, pp. 121-136, 1975.
- [14] Grossberg, S., Mingolla, E., and Todovoric, D., "A Neural network architecture for preattentive vision," IEEE Transactions on Biomedical Engineering, Vol. 36, No. 1, pp. 65- 84, January 1989.
- [15] Grossberg, S., "Cortical dynamics of three- dimensional form, color, and brightness perception: I. Monocular Theory," Perception and Psychophysics, 41(2), pp. 87-116, 1987.
- [16] Grossberg, S., "Contour enhancement, short time memory, and constancies in reverberating neural networks," Studies in Applied Mathematics, 52, pp. 217-257, 1973.
- [17] Grossberg, S., The Adaptive Brain, Vol. II, Amsterdam: NorthHolland, 1987.
- [18] Hall, E.L., Crawford, W.O., and Robert, F.E., "Computer Classification of Pneumoconiosis from Radiographs of Coal workers," IEEE Transactions on Biomedical Engineering, Vol. BME-22, No.6, pp. 518-527, November 1975.
- [19] Hema, A. M. and Yegnanarayana, B., "Formant extraction from group delay functions," Speech Communication, vol.10, pp.209-221, Aug. 1991.
- [20] Hopfield, J.J., "Neural networks and physical systems with emergent collective Computational Abilities," Proc. Natl., Acad. Sci. USA, Vol.79, pp. 2554-2558, April 1982.
- [21] Hopfield, J.J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," Proc. Natl. Acad. Sci. USA, Vol. 81, pp. 3088-3092, May 1984.
- [22] Hopfield, J.J., and Tank D.W., "Computing with Neural Circuits: A Model," Science, Vol. 233, pp. 625-633, August 1986.

- [23] **Hubel**, D. H., and Wiesel, T. N., "Functional architecture of macaque monkey visual cortex," Proceedings of the Royal Society of London, Series B, 198, pp. 1-59, 1977.
- [24] Ilya, A. R., Natalia, A. S. et al., "A Visual Cortex Domain Model and its use for Visual Information Processing," Neural Networks, Vol.4, pp.3-13, 1991.
- [25] Itakura, F., "Minimum prediction residual principle applied to speech recognition," IEEE Trans. on ASSP, Vol. ASSP-23, pp. 67-72, 1975.
- [26] John, R. A, "A spreading activation theory of memory," in **Allan Collins** and **Edward E. Smith** (Eds.) Readings in Cognitive Science: A perspective from psychology and Artificial Intelligence, Morgan Kaufman Publishers, 1988.
- [27] Julesz, **B.**, "Experiments in the visual perception of texture," Scientific American, 232, pp. 34-43, 1975.
- [28] Kitchen, L., and Rosenfeld A., "Gray level corner detection," Pattern Recognition letters, 1(2), pp. 95-102, 1982.
- [29] Kohonen, T., Self-organization and Associative Memory, Springer-verlog, Berlin, 1984.
- [30] Lippman, **R.P.**, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, pp.4-22, April 1982.
- [31] Margaret, M., Cognition, CBS College Publishing, 1983.
- [32] Marr, D., "Early Processing of Visual Information," Philosophical Transactions of the Royal Society, London, ser. B, vol. 275, pp.483-524, 1976.
- [33] Martin, A.F., and Firschein O., Intelligence: The Eye, the Brain, and the Computer, Addison-Wesley Publishing Company, Inc., 1987.
- [34] Martin, D. L., Vision in Man and Machine, **McGraw-Hill** Publishing, 1985.
- [35] Mays, L.E., and Sparks, D.L., "Saccades are spatially, not retinotopically coded," Science, Vol. 208, pp. 1163-1165, 1980.

- [36] **McClelland**, J.L., Rumelhart, D.E., and PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: **MIT Press**, 1986.
- [37] Michael, S. and Allen M. W., "Spreading Activation Layers, Visual saccades, and invariant representations for Neural Pattern recognition systems," *Neural Networks*, Vol.2, pp.9-27, 1989.
- [38] **Min-Hong**, H., and Dongsig, J., "The use of maximum curvature points for the recognition of partially occluded objects," *Pattern Recognition*, Vol. 23, No. 1/2, pp. 21-33, 1990.
- [39] **Mishkin**, M., Ungerleider, L.G., and Macko, K.A., "Object vision and spatial vision: two cortical pathways," *Trends in Neuroscience*, 6, pp. 414-417, 1988.
- [40] Neisser, U., *Cognition and Reality*, San Francisco: **Freeman**, 1976.
- [41] Rabiner, L.R., Rosenberg, A.E., and Levinson, S.E., "Considerations in dynamic time warping algorithm for isolated word recognition," *IEEE Trans. on ASSP*, Vol. ASSP-26, pp. 575- 582, Dec. 1978.
- [42] Rabiner, L.R., "On creating reference templates for speaker independent **recognition** of isolated words," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-26, pp.34-42, Feb. 1978.
- [43] Rabiner, L.R., **Levinson**, S.E., and Sondhi, M.M., "On the use of hidden Markov models to speaker-independent recognition of isolated words from a medium-size vocabulary," *AT&T Tech. J.*, Vol. 63, No. 4, pp. 627-642, April, 1984.
- [44] Rosenblatt, F., "The perceptron: A probabilistic model for information storage and organization in brain," *Psychoanalytic Review*, 65, pp. 386-408, 1958.
- [45] Rosenblatt, R., *Principles of Neurodynamics*, Newyork, **Spartan Books**, 1959.
- [46] Sako, H., and Chiba, S., "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on ASSP*, Vol. ASSP-26, pp. 43-49, Feb. 1978.
- [47] Sambur, M.R., and Rabiner, L.R., "A speaker independent digit recognition system," *Bell. system Tech. J.*, Vol.54, pp.81-102, Jan. 1975.

- [48] Smith, F.W., and Wright, M.A., "Automatic ship photo interpretation by the method of moments," IEEE Transactions on Computers, Vol. C-20, no.9, pp.1089-1095, September 1971.
- [49] Song-Tyang, L., and Wen-Hsiang, T., "Moment preserving corner detection," Pattern Recognition, Vol. 23, No. 5, pp. 441-460, 1990.
- [50] Steinbuch, "Die lernmatrix," Kybernetik, 1, pp.36-45, 1961.
- [51] Steven, P., Visual cognition, The MIT Press, 1986.
- [52] Uttal, W.R., "An autocorrelation theory of form detection," Lawrence Erlbaum Associates, Hillsdale, N.J., 1975.
- [53] Widrow and Hoff, "Adaptive switching Circuits," WESCON Convention, Record Part IV, pp. 96-104, 1960.
- [54] Yarbus, A.L., "The role of eye movements in vision process," Moscow, USSR: Nauka, 1969.
- [55] Yuzo, H. and Yasuyuki, T., "Position independent Neuro pattern matching and its application to Handwritten numerical character recognition," IJCNN, Vol. III, pp.695-702, 1990.