NOTE TO USERS

A HYBRID MULTIRESOLUTION TEXTURE SEGMENTATION AND ITS APPLICATION TO IKONOS SATELLITE IMAGERY

By
Nezamoddin Nezamoddini-Kachouie
Bachelor of Computer Science
Isfahan University of Technology
Isfahan, Iran, 1989

A thesis
Submitted to Ryerson University
in partial fulfillment of the
requirement for the degree of

Master of Applied Science

in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2004

Abstract

In this thesis, a method for segmenting textured images using Gabor filters is presented. One of the most recent approaches for texture segmentation and analysis is multi-channel filtering. There are several applicable choices as filter banks which are used for textured images. Gaussian filters modulated by exponential or by sinusoidal filters, known as Gabor filters, have been proven to be very useful for texture analysis for the images containing specific frequency and orientation characteristics.

Resembling the human visual cortical cells, Gabor function is a popular sub-band filter for multi-channel decomposition. Optimum joint spatial/spatial-frequency uncertainty principle and its ability to recognize and pass specific frequencies and orientations are attributes of Gabor filter that make it more attractive. Gabor function with these attributes could simulate the task of simple visual cells in the cortex. Gabor function has several parameters that determine the sub-band Gabor filter and must be determined accurately to extract the features precisely for texture discrimination. A wide selection range for each parameter exists and many combinations of these parameters are possible. Accurate selection and combination of values for the parameters are of crucial importance. Hence a difficult goal is minimizing the number of filters.

On the other hand a variety of approaches of texture analysis and recognition have been presented in remote sensing applications, including land cover/land use classification and urban scene segmentation. With the availability of very high-resolution commercial satellite imagery such as IKONOS, it is possible to obtain detailed information on urban land use and change detection that are of particular interest to urban and regional planners.

In this thesis considering the attributes of human visual system, a hybrid algorithm is implemented using multi-channel decomposition by Gabor filter bank for feature extraction in conjunction with Artificial Neural Networks for both feature reduction and texture segmentation. Three approaches are implemented to optimize Gabor filter bank for image segmentation. Eventually the proposed method is successfully applied for segmentation of IKONOS satellite images.

Acknowledgments

Supervisors are guiding lights for graduate students to reach the goal. Dr. Javad Alirezaie's technical knowledge and practical information provided a strong foundation for my thesis work. I would like to thank him, for his support, understanding and advice throughout my thesis that helped me to promote and to successfully complete of my research that would not have been otherwise possible.

Further I am grateful to Dr. Jonathan Li from Civil Engineering Department with whom I have had numerous discussions on satellite image processing that have resulted in joint publications and his support to provide IKONOS images that helped to improve this thesis considerably. I would like to thank Dr. Kaamran Raahemifar for valuable discussions on optimization methods and appreciate for his continuous support and his great attitude as an example professor. I also would like to take this opportunity to thank Dr. Sri Krishnan, the chair of Electrical and Computer Engineering who opened the door of signal processing world to me and showed me how an example professor could be, if I would have the opportunity to teach in future. I would like to thank Dawn Wright, program assistant of Electrical and Computer Engineering who generously and always helped me.

Our research group and other grad students made a perfect research environment both socially and professionally. I would like to thank all, especially my officemates Mahdi, Hoda and Ali. I also would like to thank Ontario Graduate Scholarships in Science and Technology (OGSST) and Natural Sciences and Engineering Research Council of Canada (NSERC) that partially supported funding for this research.

I thank my mom for her continuous support of my dreams. Special thanks to my wife, Sholeh, for her love, patience and understanding that was my major support for successful completion of my thesis.

Dedications

I would like to dedicate my research to my father in law who always encouraged me and left us just before I complete my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Several remote sensing satellites are circling the Earth, each acquiring a very specific type of imagery. In this section a brief introduction to remote sensing and its applications will be given. The most important remote sensing concept is what a satellite image actually is and how it is acquired. Almost all commercial remote sensing satellites acquire images with digital sensors which operate like the new digital cameras that have recently become popular and on the same principles. A satellite sensor like a digital camera has no film but it has thousands of tiny detectors that measure the amount of energy (i.e. electromagnetic radiation) reflecting from the Earth's surface known as spectral measurements and objects on it. These spectral reflectance values are recorded as digital numbers. To create an image that looks like a photograph, these numbers should transmit back to the Earth and convert into colors or grey scale brightness levels by computers. Sensors measure reflectance of energy in the visible, near-infrared, short-wave infrared, thermal infrared, and microwave portions of the electromagnetic spectrum depending on the sensitivity of the detectors. Most of remote sensing satellites measure energy in well-defined and very specific wavelengths of the spectrum. Created images from reflectance measurements provide extremely accurate representation of what objects and surface features on the ground look like to the naked eye according to shape, size, color and overall visual appearance. This representation is known as the spatial content of the image. Digital images not only show those spatial details, but more than that those reflectance measurements can help reveal thousands of invisible

details known as the spectral content of a satellite image such as the physical composition of buildings, the mineral content of rocks, the health of vegetation and the moisture of soil. Since digital sensors measure the energy reflectance this spectral information is visible to them. All object's characteristics such as chemical make-up, density, water content and other unseen conditions and characteristics of a particular surface feature influence how energy in various spectra or wavelengths interact with that feature and reflect off it. Insight to those invisible conditions and characteristics is provided by spectral interaction which, in turn, is measured by the digital sensor. Interpretation of digital images is the most important task to extract useful spatial and spectral information that are reflected in them [1].

Image Interpretation

Spectral values are only part of the information contained in satellite images and each pixel has both spectral and spatial information that means we can identify objects and ground features visually by their physical appearance. For instance, a round football field will look round or a square building will appear square. The color intensity of the image reveals spectral information too. For example if the color red has been chosen to represent near-infrared reflectance in the image, green objects such as vegetation which reflect energy in that wavelength completely will be appeared very bright red. Image interpretation can range from simple visual inspection of the image to the use of sophisticated image processing systems which analyze and classify based on the digital value of the spectral signatures that are reflected from ground features. The analysis and classification of ground features can be done by image processing and mapping softwares much more accurately than what the human eye can. The following products could be derived by the analysis of these digital images:

- Classification Maps which are called thematic, morphology or clutter maps as well: In these maps land areas have been clustered and classified into groups of similar land cover or land use. It may be a broad classification such as urban, forested, open fields and water or it may be more specific such as differentiating wheat, soybean, beet and

corn fields [2].

- Digital Elevation Model (DEM) are also referred to as digital terrain model [3]: DEM are used extensively in three dimensional modelling and visualization software packages commonly utilized in geologic mapping, flight simulation and civil engineering. In these data sets photogrammetric processes are applied to overlapping stereo image pairs to derive terrain elevation measurements.

- Mosaics: When one satellite image scene does not cover the entire project area, two or more adjacent scenes can be ordered and stitched together using complex computer algorithms that balance colors and match the edges of the scenes exactly to create a large seamless area dataset.

- Merging satellite images: To create a hybrid product, two different types of satellite images can be merged by computer to gain the benefits of both images. The most common is to merge a high-resolution panchromatic with a lower resolution such as merging a SPOT 10-meter image with a SPOT 20-meter or Landsat 30-meter multi-spectral image. The resulting merged image has both the spectral content of the multi-spectral data and the spatial detail of the panchromatic image.

- Change detection of the same scene: Applying specialized algorithms on two satellite images of the same area which are acquired at different times creates a change detection image.

Here an overview of satellite imagery, its applications and high resolution satellite imagery, texture, texture analysis and the importance of texture analysis to interpret satellite images are presented as following.

1.1 Applications of Satellite Imagery

A variety of objects with different shapes and sizes can be observed in aerial and satellite images. Two groups of identifiable and non identifiable objects may be recognized depend-

ing on our individual experience and perception. At the time of data acquisition a detailed record of features on the ground is provided. These recorded features in conjunction with other supporting materials such as field observation reports and maps are used to interpret the images. Satellite images may be interpreted and processed with different levels of complexity from object recognition on the earth's surface to the interactions among earth surface and subsurfaces. There are numerous applications which would benefit from a reliable and true processing and interpretation of satellite imagery. Some of the most popular ones are discussed in the next section.

Land Use/Land Cover Mapping

For many management and planning activities, the knowledge of land use and land cover is important and is considered an essential element for modeling and understanding the earth as a system. Presently land cover maps are being developed from local to national to global scales. Land cover explains and relates to the type of feature that is present on the surface of the earth such as corn fields, maple trees, concrete highways and lakes. Economic function or human activities associated with a specific piece of land is defined by land use. For example a tract of land on an urban area that is used for single family housing could be described as single family residential use, residential use or urban use regarding the term of land use and level of mapping detail. Land cover consisting of trees, grass, pavement and roof would be considered for the same tract of land [1].

Geologic and Soil Mapping

By geologic and soil mapping through visual image interpretation, the earth materials and structures can be identified and evaluated. Although geologic and soil mapping requires a considerable amount of field exploration, through the use of visual image interpretation the mapping process could be greatly facilitated.

Forestry Applications

Forestry applications focus on forest management for wildlife, water, wood and recreation. The main concern of forestry is maintenance, management and improvement of existing forest fields and fire control. Image interpretation is a feasible means to monitor forests conditions such as tree species identification, harvested areas studies and disease and insect detection. Tree species identification is generally a more difficult task in comparison with agricultural crop identification. A forest land area is usually occupied by a mixture of different tree species in contrast with relatively uniform agricultural fields. Tree species can be identified through the elimination process. In the first step the species whose presence in an area is impossible or improbable because of location or climate are eliminated. Ir the second step based on knowledge of the common species associations and their requirements, groups of species may occur in the area are established and finally using basic image interpretation principles individual tree species are identified.

1.2 High Resolution Satellite Imagery

To achieve higher resolution satellite imagery than available moderate and low resolution satellite imagery, numerous high resolution land satellites are in development phase or have been lunched already. Some of the most widely available sources of high resolution satellite images are IKONOS, SPIN-2 and QuickBird.

IKONOS Satellite Images

IKONOS is developed by Space Imaging and was successfully lunched in September 1999 after the first lunch has been failed in April 1999. IKONOS has occupied a 682 km sun-synchronous orbit having an equatorial crossing time of 10:30 A.M. Although the ground track of the system repeats every 11 days, based on tilt and latitude of the system that are selected to acquire any given image the revisit time for imaging is less than 11 days. The system is capable of collecting data at angles of 45° from vertical in the both across and along track directions. This provides the opportunity for frequently covering a given

area. A typical image size is 11 by 11 km but user specified image strips can be collected. Linear array technology is employed in IKONOS and data in four multi spectral bands including blue (0.45 to 0.52 μm), green (0.51 to 0.60 μm), red (0.63 to 0.70 μm) and near IR (0.76 to 0.85 μm) at 4m nominal ground resolution are collected. An effective 1 m spatial resolution can be provided by pan-sharpened multi-spectral imagery. Pan-sharpened imagery is concluded by combining one-meter panchromatic imagery and the multi-spectral bands. A data set with 1 m spatial resolution pan-sharpened IKONOS satellite images of an urban residential area is employed for applying to the proposed algorithm. The 1 m resolution IKONOS satellite imagery used in this study covers a complex urban residential area mixing roads, paths, houses, and trees. It is located in the northwest of the Greater Toronto Area (GTA), Ontario, Canada, near the Pearson International Airport and was acquired in August 2001. The IKONOS imagery was geometrically corrected in two dimensions (X and Y) in the Universal Transverse Mercator (UTM) projection coordinate system, zone 17 in the Geodetic Reference System of 1980 (GRS80) and North American Horizontal Datum of 1983 (NAD83) [1]. The roads in the image are continuous regions which form a network. Houses are very dense in the image, some roofs of which have similar spectral characteristics as roads. Trees along the road are located close to houses.

1.3 Digital Processing of Satellite Images

As humans, we are adept at visually interpreting data and can distinguish a huge number of different colors and several gray levels. Although humans have a demonstrated ability to identify vegetation, water and urban areas on several different types of imagery, digital process of remotely sensed data is needed.

- Our ability to distinguish small differences in color are limited. Human's vision is limited especially in resolution of shades of gray. Although gray level data are usually collected using 256 gray levels or more, human eye can distinguish 8 to 64 shades of gray. The exact number depends much on the arrangement of the shades and a great amount of information is potentially lost [4]. Precise data mostly outpace the

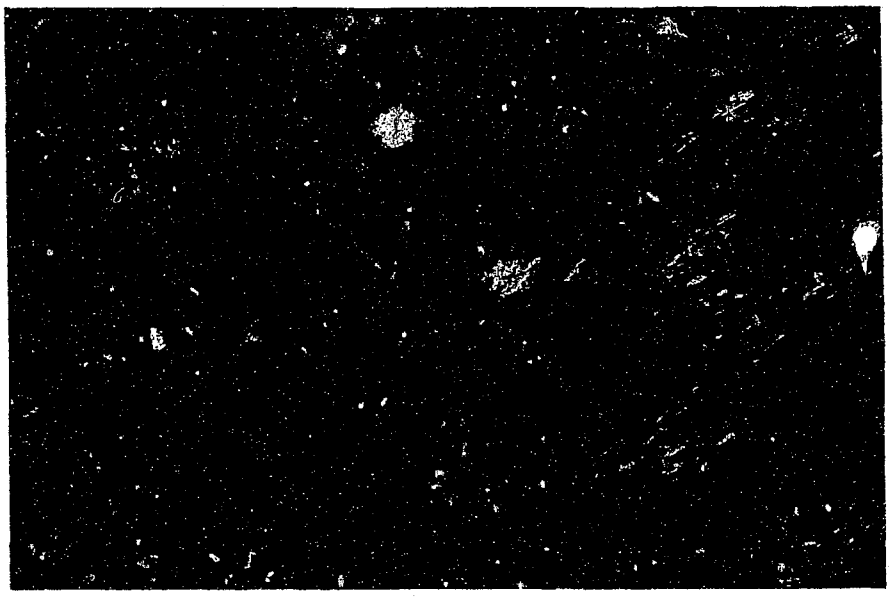


Figure 1.1: Satellite imagery taken by IKONOS, showing an urban area near the International Pearson Airport in the Grater Toronto Area in Canada

Figure 1.2: An IKONOS imagery with 1m resolution covering a part of Markham in Grater Toronto Area in Canada

human vision interpreter. However computers do not have trouble to distinguish 256 gray levels and recognize each one individually. On the other hand the computer's presentation of the data can be controlled by an analyst and he can group or extract a portion of it, combine, compare and contrast data with more ease and precision than if it were left to human vision alone.

- Since our vision interpretations are highly subjective, they are not perfectly repeatable. Generated results by computer are usually repeatable in converse.
- Computers may be better suited to managing the large amount of detailed and tedious data.

The process of digital and manual image interpretation are similar in many ways and the goals of analysis are often the same, but the routes may vary. One of the most important tasks to interpret images is image segmentation and is required to be accomplished prior to the image interpretation. Texture segmentation is happening in human vision system to segment and interpret images in real time. On the other hand in digital routes for image processing, researchers try to mimic and imitate human vision system. Since texture recognition is a fundamental part in human image segmentation, using textures and developing some algorithms to recognize and segment textures have been noticed recently by a lot of researchers [5, 6, 7].

1.4 Texture and Texture Analysis

Simplifying assumptions about the uniformity of intensities in local image regions are considered in many image processing and machine vision techniques. However, natural images and real objects usually do not exhibit regions of uniform intensities. For example, the image of a natural surface has variations of intensities and is not uniform. These variations of intensities form certain repeated patterns known as visual texture. The patterns may be the result of reflectance differences such as the surface and color or can be the result of physical surface properties which often have a tactile quality such as softness, roughness and oriented

strands. Although when we see texture we recognize it, defining it is very difficult. The large number of different texture definitions, which are presented by image and vision researchers demonstrate this difficulty. The following definition for texture is given by Oxford Dictionary [8]:
The way a surface, substance or fabric feels when you touch it, for example, how rough, smooth, hard or soft it is.

More complete definition for texture is given by Webster Dictionary [9]:

- Function: noun
- Etymology: Latin textura, from textus, past participle of texere
- Description
  - something composed of closely interwoven elements; specifically : woven cloth
  - the structure formed by the threads of a fabric
  - essential part : SUBSTANCE
  - identifying quality : CHARACTER
  - the disposition or manner of union of the particles of a body or substance
  - the visual or tactile surface characteristics and appearance of something; the texture of an oil painting

A catalogue of texture definitions in the computer vision literature is collected by Coggins [10]. Some examples are presented here.

1. Texture may be regarded as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule.

2. A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic.

3. The image texture which is considered is non figurative and cellular. An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of its (tonal) primitives.

4. A fundamental characteristic of texture:

   (a) It cannot be analyzed without a frame of reference of tonal primitive being stated or implied.

   (b) For any smooth gray-tone surface, there exists a scale such that when the surface is examined, it has no texture.

   (c) Then as resolution increases, it takes on a fine texture and then a coarse texture.

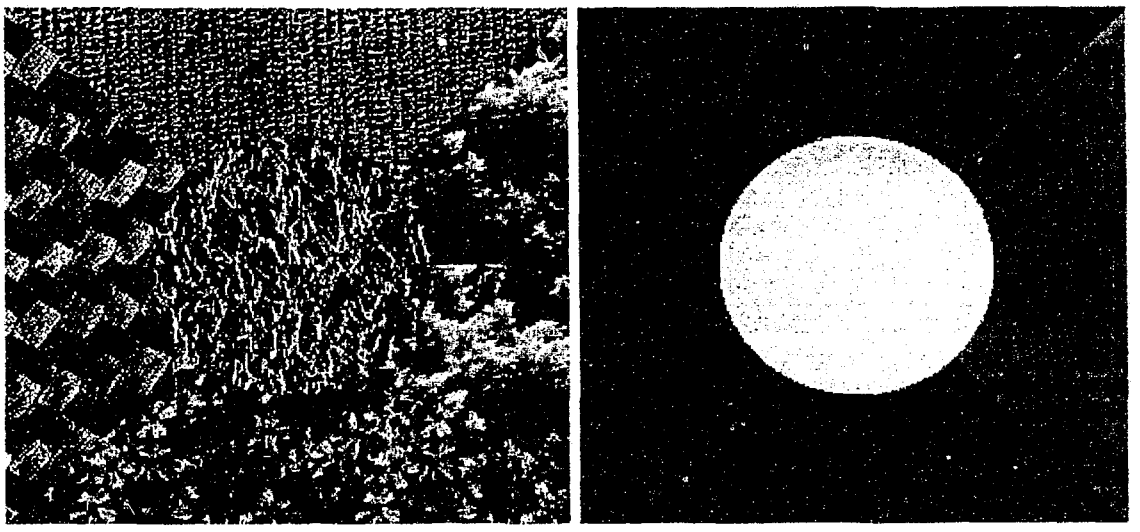
5. The notion of texture appears to depend upon three ingredients:

   (a) The parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.

   (b) The order consists in the non random arrangement of elementary parts.

   (c) Some local 'order' is repeated over a region which is large in comparison to the order's size.

Almost all of the natural surfaces and patterns are textured. Most of the artificial surfaces and patterns are textured as well. Because of the importance of textures, recognizing different textures is a crucial stage in most of computer vision systems. The problem in texture segmentation is partitioning the different regions with similar textures like segmented image depicted in Fig.6.2. In this chapter different approaches of texture analysis are discussed in Section 1.5 and various applications of texture analysis are given in Section 1.6.



(a) (b)

Figure 1.3: (a)Original image (b)True segmentation results

1.5 Various Approaches for Texture Analysis

There are a wide range of different methods to approach texture analysis, feature extraction and texture recognition. Overviews of major categories are investigated by several authors and are available in different books and articles. These categories could be divided in five major methodologies [11, 12, 13].

Geometrical techniques: such as Voroni tessellation.
Structural techniques: such as structural features.
Statistical techniques: such as autocorrelation and Co-occurrence features.
Model based techniques: such as fractal parameters and random field.
Signal processing techniques: such as spatial and spatial/frequency filter banks, wavelet transforms and wavelet packets.
Among them statistical and signal processing techniques seem are more commonly applied. On the other hand the geometrical and structural techniques need strict assumptions on

images as priori. They have limited practical applications and are less commonly used in comparison with two former techniques for texture recognition.

1.6 Applications of Texture Analysis

In this section an overview of some applications of texture analysis and recognition systems that are more important is presented.

Satellite Remote Sensing: Measurement of specific properties of objects from far distance is called remote sensing. The number of Earth observation satellites that are in operation is rising every year. These satellites carry a diverse spectrum of radar and optical sensors capable of acquiring imagery that utilizes for a broad range of applications. There are a variety of different remote sensing techniques including very high resolution (VHR) satellite imagery, synthetic aperture radar (SAR), satellite multi-spectral imagery, seismic surveys, sonar surveys, etc. A variety of numerous approaches for texture analysis and recognition have been presented in remote sensing. Applications include land cover/land use classification [14], urban area and facility segmentation, road network map updating, cloud classification [15], sea floor mapping, seismic pattern recognition, sea ice segmentation, mapping, urban planning, civil engineering, defence, agriculture, environmental science, land management and a lot of other applications [1, 14, 16, 17].

Medical imaging: Medical images are usually obtained by non-intrusive methods such as x-ray, photography, Magnetic Resonance Imagery (MRI), ultrasound, tomography, CT scan, etc. On the other hand, some medical images such as microscopic photography of biopsies are obtained by intrusive methods. Textural properties in most of these types of images are important to diagnose the disease. A wide variety of different diseases like cancer are often recognized and characterized by characteristics of textures obtained by different medical imaging methods. Most effective way to reduce mortality of some diseases is early detection of the disease. For example the Breast cancer among women is one of the leading

cause of cancer deaths [18]. For early detection of this cancer, mammography screening is currently the best method. In mammography images lesion detection is successfully accomplished by applying texture recognition techniques [19]. The other applications of medical image segmentation are lung and liver cancer diagnosis. Electron microscope images of thin cell slices of the liver and lung are taken to diagnose the lesion by texture analysis and the texture characteristics of the lesion would be extracted. Also for this kind of images, texture techniques have been applied successfully for cancer detection [20].

Visual content based retrieval: Since manual labelling of the images is a tedious task and in some cases is not feasible at all, content-based retrieval methods to search the visual data are hot area in the computer vision. Retrieving an image with some specific contents is a difficult task on the internet and numerous number of technics have been developed to search and manage images in visual data banks. Due to the nature of images in a specific data bank, the content types usually have some texture characteristics that are very useful to search and extract the image. Hence, texture analysis and recognition methods are in high demand for visual retrieval systems. Using texture analysis, automatically feature extraction and databases search is possible.

Industrial defect inspection: Defect detection of raw materials and manufactured products is very important in industrial processes. Automatic visual inspection systems are preferred since manual inspection not only is a tedious task but also is not enough accurate. Texture recognition technics are useful to analyze the surface characteristic for defect detection and surface qualification. Applications of automatic inspection are growing and a lot of approaches using different texture analysis technics are proposed [21, 22, 23].

Document segmentation: In printed documents like papers text could be considered as a homogeneous texture that is written in specific language, font and size. Texture analysis techniques would be necessary as a preprocessing task to segment these documents in

separate regions of text and graphic before accomplishing the major processing on the region of interest. Some implemented Optical Character Recognition (OCR) systems employs texture segmentation techniques to optimize the character recognition task. These techniques could be also useful for document digitization and save the disk space by converting the text regions from graphic to text format.

1.7 Thesis Objective

The objective of this thesis are:

- Optimizing Gabor filter bank for texture analysis
- Utilizing Artificial Neural Networks for feature dimension reduction
- Developing Artificial Neural Networks for segmentation of extracted features
- Application of developed techniques for satellite image segmentation

1.8 Thesis Outline

The focus of this thesis is mainly on using of the filter bank model and multi-channel decomposition for feature extraction and artificial neural networks for feature reduction and segmentation of satellite images. This thesis is organized as follows:

Chapter 2 describes texture segmentation and multi-channel decomposition by filter bank model.

Chapter 3 describes feature extraction using Gabor fiilter bank.

Chapter 4 explains neural networks for feature reduction and segmentation.

Chapter 5 discusses about practical aspects of neural networks and the other optimization methods that are implemented to optimize Gabor filter bank.

Chapter 6 presents results.

Chapter 7 discusses the results and presents conclusion.

Chapter 2

Texture Segmentation and Filter Bank Model

2.1 Segmentation

The shape of an object can be described in terms of: (1) its boundary, and (2) the region that is occupied by the object. The first one can be achieved by edge detection of the image and the second one requires segmenting the image into homogeneous regions using homogeneous characteristics of image regions such as intensity and texture. Segmentation is the process of distinguishing objects in the data set from their surroundings so as to facilitate the creation of geometric models. To segment an image, intensity provides less robust classification information than texture. Intensity does not identify different homogeneous regions consistently and it provides poor directional information. Finding regions that represent objects or meaningful parts of objects is the goal of image segmentation. As it is mentioned in chapter one, there are numerous applications for satellite image segmentation. To accomplish these applications, some important features are obtained by image segmentation. As a result satellite image segmentation can facilitate the data extraction from satellite imagery by:

- Measuring features such as: shape, surface area and volume
- Partitioning regions such as: land cover, urban area and vegetation
- Mapping the satellite imagery and existing maps and databases.

Texture segmentation

Texture segmentation and analysis are important aspects of pattern recognition and digital image processing. In image analysis, textures have been used to perform scene segmentation, object, shape and region recognition and surface classification. In texture analysis attributes which are useful to discriminate, recognize and segment different texture types should be identified. Texture segmentation involves accurately partitioning an image into sections according to the textured regions or by recognizing the borders between different textures in the scene or image.

2.2 Human Visual System

To discover the mysteries of human visual system and improve our knowledge about biological vision, animal models are used by scientists. By improving our understanding about the human visual system, more accurate artificial vision systems can be implemented and used as comparison basis to measure the performance of machine vision models. Julesz [4] explained pre-attentive aspects of human visual perception and developed the concept of textons. Although the concept of texton was first proposed by Julesz twenty years ago, a clear definition is still in debate. Textons are suspected to have positional relationship with human visual system and to direct it to identify the patterns in the scene. To avoid the complexities of human visual system, an abbreviate explanation of the interconnections is presented. Electrical impulses are passed from receptor cells that are located on the back inner surface of the eyes to the visual cortex placed in the brain. To reach the visual cortex they are passed through various nervous pathways and the optic nerve. Scientists discovered that various cells in the visual cortex perform different processing tasks on the incoming signals. Responses of the visual cortex in cats were tested by Hubel and Wiesel [24] and they deduced that simple cells are tuned to specific orientations. It means individual simple cells respond to a certain range of frequencies. Campbell [25] explained that human vision system is sensitive to both orientation and spatial frequency and presented a model made up of a

number of independent detectors. Detectors are preceded by relatively narrow band filters that are tuned to different frequencies and each filter-detector pairs would be considered as a separate channel. In the other experiments Pollen and Ronner [26] have shown that the frequency bandwidth of simple cells in the visual cortex is about one octave.

Recent study on the human vision system indicates that the spatial-frequency representation is adequate to characterize texture properties. On the other hand, experiments on visual system of some mammals like cats show that their visual system is similar to that of humans [27]. In these mammals the image is decomposed to several filtered images by the visual system. The decomposition filters have different radial frequencies and orientations and generate tuned filtered images to the specified frequencies and orientations [28]. Researchers have been motivated by this experiments and observations to develop multi-resolution texture models for texture segmentation and analysis.

2.3 Filter Bank Model

One of the most recent approaches for texture segmentation and analysis is Multi-channel filtering. In this method an image is processed using a bank of filters with multiple resolution. Having the ability to decompose the image, the filter bank extracts the relevant texture features from the image.

Several researchers in this field [5, 7, 11, 6] have proposed texture segmentation and analysis methods based on a filter bank model which is motivated by the human vision system's (HVS) unique capabilities for texture segmentation [26, 29]. In this model, a set of filters in the frequency domain (or a set of masks in spatial domain) are applied in parallel to an input image to decompose it into a set of filtered images accordingly. The set of filtered images are used directly as feature images or they are processed and/or combined to extract the features. Eventually extracted features are used for the segmentation of the input image. Each individual filter in the bank is designed to focus on a specific local spatial region and a specific range of frequency to make spatial/spatial-frequency decomposition possible. The joint spatial/spatial-frequency decomposition, unlike Fourier analysis considers the local

frequency variations that vary with position in the image. Textured images that are encoded in narrow spatial frequencies and orientation channels can be recognized and segmented by filtered images that carry out the texture features. These features describe local spatial-frequencies and orientations to which the channels are tuned. Therefore texture can be regarded as a region information carrier and the local structure of a texture can be explained by the frequencies and orientations of the carrier that is contained in the channel outputs. There are several applicable choices as filter banks which are used for textured images. From a practical point of view, some filters may be more useful for specific segmentation tasks but not for the others. Gaussian filters modulated by exponential or by sinusoidal filters, known as Gabor filters, have been proven to be very useful for texture analysis for the images containing specific frequency and orientation characteristics [30, 7, 31, 32]. While different filter banks can perform joint spatial/spatial-frequency decomposition, Gabor filter bank using a Gabor base function is one of the most attractive ones. This set of filters has an optimal localization in the joint spatial/spatial-frequency domain according to the uncertainty principle. These selective band pass filters with different radial spatial frequency and orientation have optimum resolution in the time and frequency domains that resemble the simple visual cortical cell characteristics [26, 29].

2.4 Previous Work

Considerable research has been performed using Gabor filters for texture segmentation and analysis. The research involving the use of Gabor filters for texture segmentation can be divided into three major disciplines:

1. Investigating the best frequencies and orientations (bandwidth) according to the characteristics of a specific textured image.

2. Inventing robust feature extraction and feature reduction methods.

3. Employing the best classification and segmentation methods to devote to the optimal extracted features.

Bovik and Clark [5] in their research have proposed a computational approach for analyzing visible textures. In their method they detect boundaries between textures by comparing the channel amplitude responses and detecting discontinuities in the texture phase by locating large variations in the channel phase responses. Dunn and Higgins have investigated designing the optimal Gabor filters [7] and argued that Gabor filter outputs can be presented as a Rician model and developed an algorithm to select optimal filter parameters to discriminate texture pairs. Jain and Farrokhnia proposed a filter selection method [6] based on reconstruction of the input image from the filtered images. They also proposed the optimum radial frequencies and orientations for different channels that have been used widely by many researchers. Unser and Eden described an unsupervised texture segmentation method [32] using Karhunen-Loeve transform on the resulting features of Gabor filters to reduce the feature vector dimension. Weldon et. al. presented a method [31] to design a single Gabor filter for multi-textured image segmentation. Clausi and Jernigan investigated and compared different techniques [33] used to extract texture features and Jain and Karu proposed a neural network texture classification method [34] as a generalization of the multi-channel filtering method.

2.5 Principles of Multi-Resolution Texture Analysis

An overview of the elements of the general multi-channel filtering model will be explained before proceeding with the description of Gabor filter approach. Consider two simple textures generated by sinusoids and composed as a synthetic image. One is low frequency and the other is a summation of the same low frequency and a high frequency sinusoids as depicted in Fig. 2.1. The first step in the decomposition model depicted in Fig. 2.2 is filtering the input textured image. If the filter passes the high frequency sinusoid and stops the low frequency one, the filter output for this textured image would be like Fig. 2.3. Filter output has a low energy response for the low frequency texture and high response for the other. Classification of the image by the filter response is still erroneous and will result in a high classification error. To reduce the error, applying nonlinearity and smoothing functions to

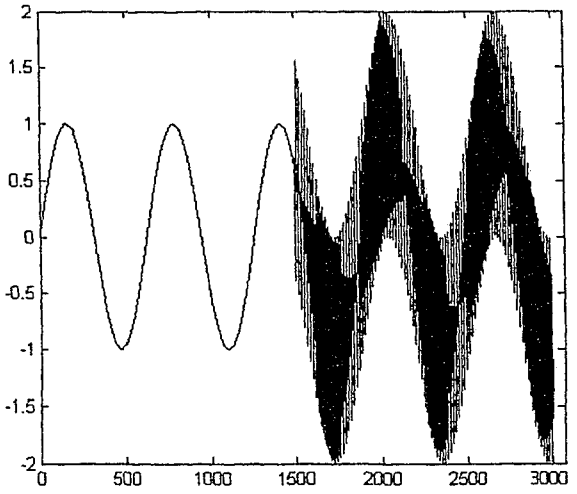


Figure 2.1: Simple Sinusoid Textures

the filter response for local energy measurement are very important. The filtered response after applying nonlinearity and smoothing would be like Fig. 2.4 and Fig. 2.5, respectively.

2.5.1 Local Energy Measurement

To estimate the local energy in the filter response, local energy function would be applied. Since each filter in the filter bank has its specified frequency and orientation, local energy function passes regions with strong local sub-band energy in each channel. Supporting high spatial resolution and high spatial-frequency resolution at the same time is not possible since these are conflicting goals. To localize the edges accurately, high spatial resolution should be provided and accurate estimation of the local energy, needs high spatial-frequency resolution. The objective of smoothing function is to smooth out the variations in the filter response after applying the nonlinearity. One of the well known and widely used smoothing filters is low pass Gaussian filter. This low pass filter has optimum joint resolution in spatial and spatial-frequency domain. The spatial-frequency of the channel is used to determine the size of the smoothing filter. For low spatial-frequency channels, the low pass filter must be wide to accurate estimate of the local energy of the channel. On the other hand local energy

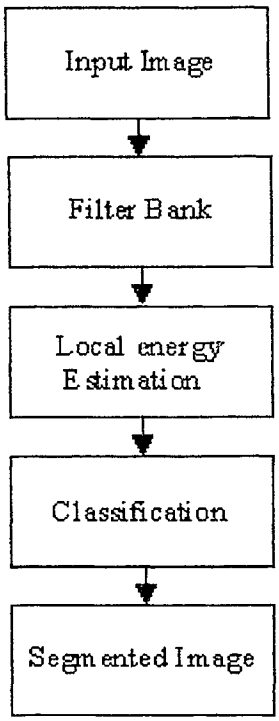


Figure 2.2: Multi-Channel Decomposition

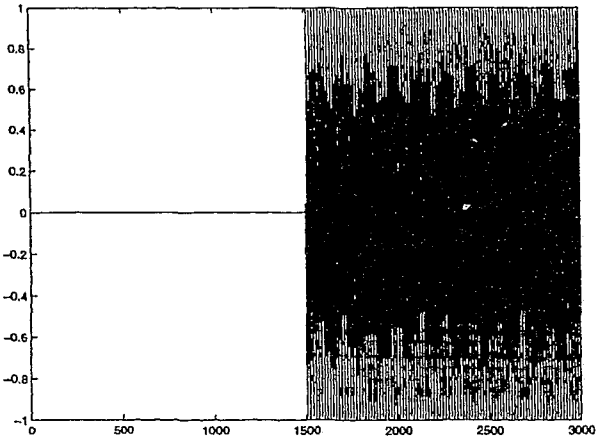


Figure 2.3: Simple Sinusoid Textures after Filtering

estimation of the high spatial-frequency channels requires a narrow low pass filter.

2.6 Gabor Filter Bank

Resembling the human visual cortical cells Gabor function is a popular sub-band filter for multi-channel decomposition. Optimum joint spatial/spatial-frequency uncertainty principle and its ability to recognize and pass specific frequencies and orientations are attributes of Gabor filter that make it more attractive. Gabor function with these attributes could simulate the task of simple visual cells in the cortex. Gabor base function is a Gaussian function modulated with exponential or sinusoidal function that is defined in terms of the product of a Gaussian and an exponential. Two dimensional (2D) Gabor functions $h(x, y)$ can be written as:

$$h(x, y) = g(x, y) \cdot exp^{-2\pi j f_r x} , \tag{2.1}$$

where

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot exp^{\frac{-1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} . \tag{2.2}$$

and its frequency response $H(u, v)$ is:

$$H(u, v) = G(u - f_r, v) = exp^{-2\pi^2[(u-f_r)^2\sigma_x^2+v^2\sigma_y^2]} , \tag{2.3}$$

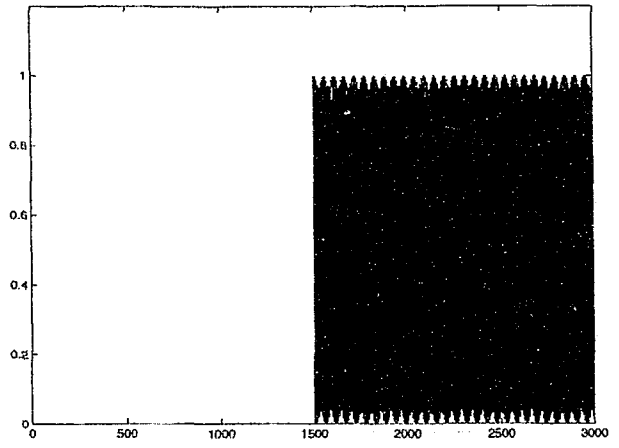


Figure 2.4: Filtering Results after Applying Nonlinearity

where

$$f_r^2 = u_r^2 + v_r^2 \ , \tag{2.4}$$

and

$$\theta = tan^{-1}(v_r/u_r) \ . \tag{2.5}$$

Gabor functions are bandpass filters which are Gaussians, centered on (f_r,θ) in the spatial-frequency domain. The parameters f_r, θ, σ_x and σ_y determine the sub-band Gabor filter. f_r and θ are center frequency and orientation and σ_x and σ_y are the bandwidth of the filter. Equation 2.2 defines a complete Gabor function consisting of both real and imaginary (or even and odd) components. Rotation by θ in the spatial domain ($x - y$ plane) or in the spatial-frequency domain ($u - v$ plane) provides selective arbitrary orientation for different channels. A Gabor filter bank by using only even-symmetric or real components can be implemented as suggested by Jain and Farrokhnia [6] and can be represented by:

$$h(x, y) = g(x, y).cos(2\pi f_r x) \ , \tag{2.6}$$

and $H(u, v)$ is:

$$H(u, v) = G(u - f_r, v) + G(u + f_r, v) \ . \tag{2.7}$$

That is composed of two Gaussians in the spatial-frequency domain. Resolution in spatial and spatial-frequency domain is determined by the bandwidth of the filter (i.e., σ_x and σ_y).

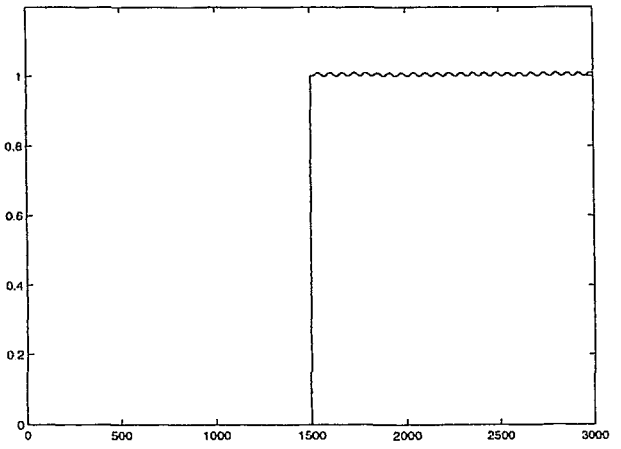


Figure 2.5: Filtering Results after Applying Smoothing

High spatial resolution requires small spatial extent or small values for x and y. On the other hand to support high spatial-frequency resolution, small frequency bandwidth or high values of σ_x and σ_y is necessary. It means for image segmentation small intervals in spatial domain are preferred for better discrimination of different textures in the region boundaries. However to recognize characteristics of specific textures, small frequency bandwidth is preferred. Narrow spatial-frequency bandwidth (for high spatial-frequency resolution) and small spatial extent (for high spatial resolution) could not be satisfied at the same time, since they have inverse mathematical relation and conflict with each other. Joint spatial/spatial-frequency resolution is known as uncertainty principle and is defined as:

$$(\Delta_x)(\Delta_u) \geq \frac{1}{4\pi} , \tag{2.8}$$

where Δ_x and Δ_u are spatial resolution and spatial-frequency resolution, respectively. Gabor function has the optimum uncertainty principle and the equality is achieved by this function. It means Gabor filter has the optimum spatial/spatial-frequency resolution.

2.7 Implementation of Gabor Filter

A typical Gabor filter in spatial and spatial frequency domain and its frequency response are depicted in Fig. 2.6, 2.7 and 2.8, respectively. Gabor function has six parameters

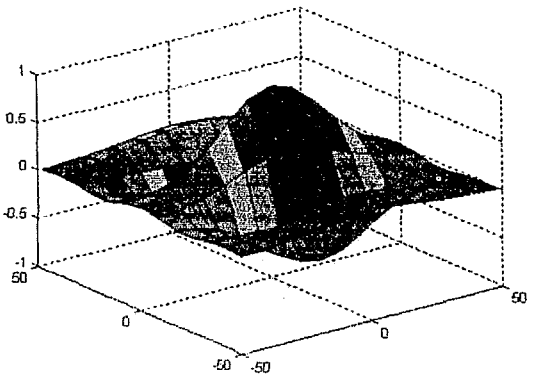


Figure 2.6: Gabor Filter in Spatial Domain

that determine the sub-band Gabor filter and must be determined accurately to extract the features precisely for texture discrimination. Gabor filter parameters are defined as following: center frequency (f_r), orientation (θ), spatial bandwidths in X and Y directions (σ_x and σ_y), angular and frequency bandwidths (D_θ and D_f). A wide selection range for each parameter exists and many combinations of these parameters are possible. Accurate selection and combination of values for f_r and θ have a crucial importance. Hence an important and difficult challenging goal is minimizing the number of filters. Many researchers have been working to design an optimal Gabor filter bank with minimum number of filters. Suggested radial frequencies by Jain and Farrokhnia [6] are $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$, $64\sqrt{2}$ and 4 orientations, 0° , 45° , 90° and 135° for each center frequency are used. On the other hand, D_θ and D_f must be set accurately to cover spatial-frequency domain properly. Having f_r, θ, D_θ and D_f as known parameters, two equations are necessary to determine σ_x and σ_y. Calculating the half bandwidth where $v = 0$, i.e., Cutoff Frequency equal to $-6dB$ yields:

$$H(u,0) = exp^{-2\pi^2(u-f_r)^2\sigma_x^2} \tag{2.9}$$

set $H(u,0) = 1/2$ and solving for u:

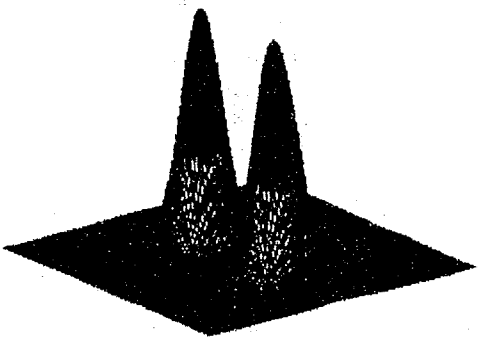


Figure 2.7: Gabor Filter in Spatial Frequency Domain

$$u = f_r \pm \frac{\sqrt{\ln 2}}{\sqrt{2}\pi\sigma_x} \tag{2.10}$$

To obtain the radial bandwidth or distance between two frequencies in terms of octave, the log_2 of their ratios would be:

$$D_f = \log_2\left(\frac{f_r\pi\sigma_x + \sqrt{\frac{(\ln 2)}{2}}}{f_r\pi\sigma_x - \sqrt{\frac{(\ln 2)}{2}}}\right) \tag{2.11}$$

would yield the σ_x as:

$$\sigma_x = \frac{\sqrt{\ln 2}(2^{D_f}+1)}{\sqrt{\ln 2}(2^{D_f}-1)} \tag{2.12}$$

The second equation where u is set to f_r is:

$$H(f_r, v) = exp^{-2\pi^2 v^2 \sigma_v^2} \tag{2.13}$$

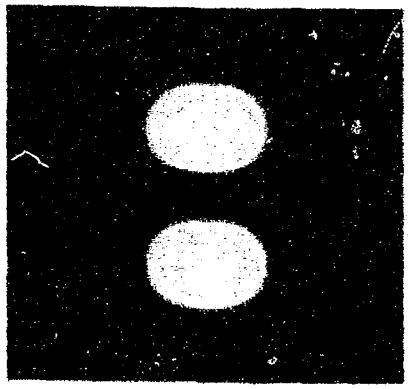


Figure 2.8: Frequency Response of Gabor Filter

v is solved as:

$$v = \pm \frac{\sqrt{\ln 2}}{\sqrt{2}\pi\sigma_y} \tag{2.14}$$

The tangent of the angular half bandwidth could be computed as:

$$D_\theta = 2\tan^{-1}\left(\frac{\ln 2}{f_r\sqrt{2}\pi\sigma_y}\right) \tag{2.15}$$

and as a result σ_y is:

$$\sigma_y = \frac{\sqrt{\ln 2}}{f_r\sqrt{2}\pi\tan(\frac{D_\theta}{2})} \tag{2.16}$$

2.8 Other Multi-Channel Decomposition Methods

In almost all of the filtering approaches it is assumed that a texture could be identified by energy distribution in the frequency domain. Hence, by decomposing the frequency domain of the image into a sufficient number of sub-band channels the different textures would be identified by different signatures in the frequency spectrum.

2.8.1 Laws filter masks

Laws introduced a separable filter bank for texture segmentation as one of the premiers to use filtering approach to identify different textures. The suggested filter bank by Laws [35] including 25 filters, i.e., five filters in each dimension that are:

$$L5 = [\ 1,\ 4,\ 6,\ 4,\ 1\ ]$$

$$E5 = [-1, -2,\ 0,\ 2,\ 1\ ]$$

$$S5 = [-1,\ 0,\ 2,\ 0, -1\ ]$$

$$W5 = [-1,\ 2,\ 0, -2,\ 1\ ]$$

$$R5 = [\ 1, -4,\ 6, -4,\ 1\ ]$$

One dimensional (1D) separable Laws filter masks are depicted in Fig. 2.9. By applying the filter bank to the input image 25 filtered images are produced and as a result feature vectors consist of 25 features. These mnemonics (L, E, S, W and R) stand for Level, Edge, Spot, Wave and Ripple. Note that all kernels except L5 have zero sum. Laws has also presented convolution kernels of length three and seven in his research, and has discussed the relationship between different sets of kernels. Using these 1D convolution kernels, by convolving a vertical kernel with a horizontal kernel, 25 different 2D convolution kernels can

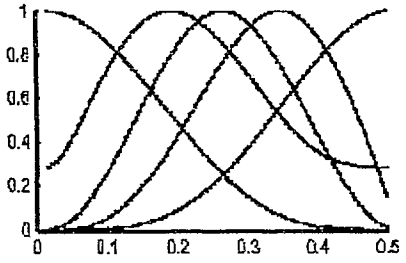


Figure 2.9: 1D separable Laws filter masks

be generated [35]. As an example, a 2D kernel L5E5 would be obtained by convolving a vertical kernel L5 with a horizontal kernel E5. Twenty five 2D convolution kernels can be generated from the 1D kernels which 24 of them have zero sum and only the sum of L5L5 kernel is not zero. A list of all 5x5 kernels is given below:

L5L5 E5L5 S5L5 W5L5 R5L5

L5E5 E5E5 S5E5 W5E5 R5E5

L5S5 E5S5 S5S5 W5S5 R5S5

L5W5 E5W5 S5W5 W5W5 R5W5

L5R5 E5R5 S5R5 W5R5 R5R5

2.8.2 Discrete Cosine Transform

DCT transforms a signal from a spatial representation into a frequency representation. Because of the fast implementation and good results, DCT not only is widely used in image compression algorithms such as JPEG compression standard [36], but also in image segmentation approaches [37].

DCT for image segmentation

Discrete cosine transform (DCT) helps separate the image into parts known as spectral sub-bands of differing importance with respect to the image's visual quality. Since DCT transforms a signal or image from the spatial domain to the frequency domain, it is similar to the discrete Fourier transform. The DCT input is an N by N matrix of integers. This matrix contains each pixel's gray scale level (from 0 to 255 for 8 bit per pixel representation). For most images, much of the signal energy lies at low frequencies; these will appear in the upper left corner of the DCT and the lower right values represent higher frequencies, and are often smaller values.

2D DCT

With an input image S, using the formula for a 2D DCT, the coefficients of the output image F are obtained by:

$$A_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0; \\ 1 & \text{otherwise} \end{cases}$$

$$A_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } v = 0; \\ 1 & \text{otherwise} \end{cases}$$

$$F_{vu} = \frac{1}{4} A_v A_u \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} S_{yx} \cos\left(v\pi \frac{2y+1}{2N}\right) \cos\left(u\pi \frac{2x+1}{2N}\right) \tag{2.17}$$

The input image is N pixels wide by N pixels high, $S(x, y)$ is the intensity of the pixel in row i and column j and $F(u, v)$ is the DCT coefficient in row u and column v of the DCT matrix. All DCT multiplications are real. This lowers the number of required multiplications as compared to the discrete Fourier transform.

Inverse Discrete Cosine Transform (IDCT)

To rebuild an image in the spatial domain from the frequencies obtained above, the IDCT formulae can be used:

$$A_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0; \\ 1 & \text{otherwise} \end{cases}$$

$$A_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } v = 0; \\ 1 & \text{otherwise} \end{cases}$$

$$S_{yx} = \frac{1}{4} \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} A_v A_u F_{vu} \cos\left(v\pi \frac{2y+1}{2N}\right) \cos\left(u\pi \frac{2x+1}{2N}\right) \tag{2.18}$$

Mathematically, the DCT is perfectly reversible and no image definition is lost. Lower frequencies are more obvious in an image than higher frequencies, hence in image compression algorithms by transforming an image into its frequency components and throwing away less important information according to the image quality (i.e., throw away some higher frequency coefficients), the amount of data that is needed to describe the image without sacrificing too much image quality can be reduced.

Optimization

A 2D DCT can be evaluated more quickly using a series of 1D DCTs:

$$F_u = \frac{1}{2} A_u \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} S_x \cos\left(u\pi \frac{2x+1}{2N}\right) \tag{2.19}$$

This means that instead of performing a 2D DCT, a 1D DCT can be performed on each row and then on each column of the processing block.

Feature Extraction

Ng et al. [37] introduced a 3 by 3 DCT to extract features from textured images. They suggested using eight masks excluding the low frequency component of DCT. The suggested filter bank is separable and the masks are:

$$h1 = [1, 1, 1]$$

$$h2 = [1, 0, -1]$$

$$h3 = [1, -2, 1]$$

Applying the suggested filter bank to the image generates eight filtered images, therefore, eight-dimensional (8D) feature vectors will be obtained.

Chapter 3

Feature Extraction by Gabor Filter Bank

The proposed method, CMLP-GFHF, is a hybrid algorithm using Competitive layer and Multi-Layer Perceptron (CMLP) in conjunction with Gabor Filter bank and High Frequency component (GFHF). The segmentation algorithm is described in the following chapters. Multi-channel decomposition is explained in this chapter and Artificial Neural Networks will be discussed in the next chapter. The following sections are explained in this chapter: Multichannel decomposition, High and low frequency component, Implementation of the Gabor filter and Computing features.

3.1 Multi-Channel Decomposition

Multi-channel decomposition is achieved by applying the filter bank to generate filtered images and local energy estimation for feature extraction as shown in Fig. 3.1. In the first step a textured input image is decomposed into filtered images. In the second stage, to estimate the local energy, a function consisting of nonlinearity and smoothing is applied to the output of Gabor filter bank for feature extraction. In this step each channel corresponding to a different filter is tuned to a different radial frequency or orientation to capture local characteristics of different textures such as spatial-frequency in the input image [38]. After the second stage, a set of feature images or a feature vector corresponding to each pixel in the input image is generated. The dimension of feature vectors is equal to (or multiplied by

an integer) the number of filters in the filter bank.

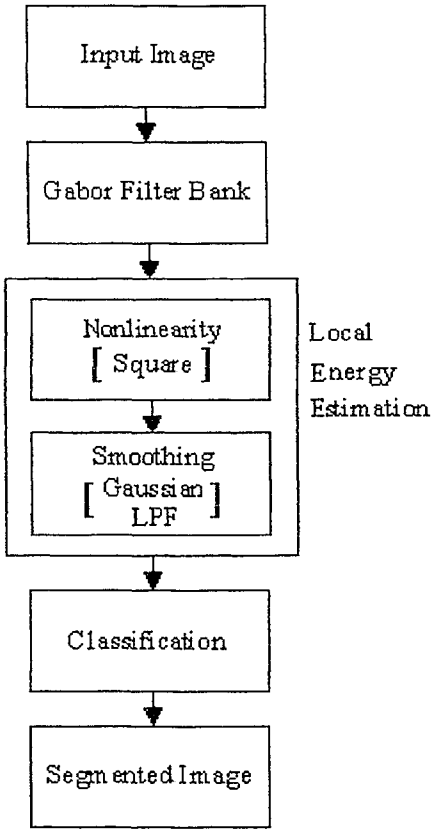


Figure 3.1: Multi-Channel Decomposition

In the first stage, to decompose the input image three sets of Gabor filters composed of 20, 30 and 40 filters are designed. For all sets the same five radial frequencies suggested by Jain and Farokhnia [6], are used:

$$4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2} \tag{3.1}$$

The radial frequency bandwidth is one octave, thus the frequency difference of f_1 and f_2, is given by $\log_2(f_2/f_1)$, is equal to 1. In the first bank for each radial frequency four orientations 0°, 45°, 90° and 135° are used to generate the total number of 20 channels. In the second

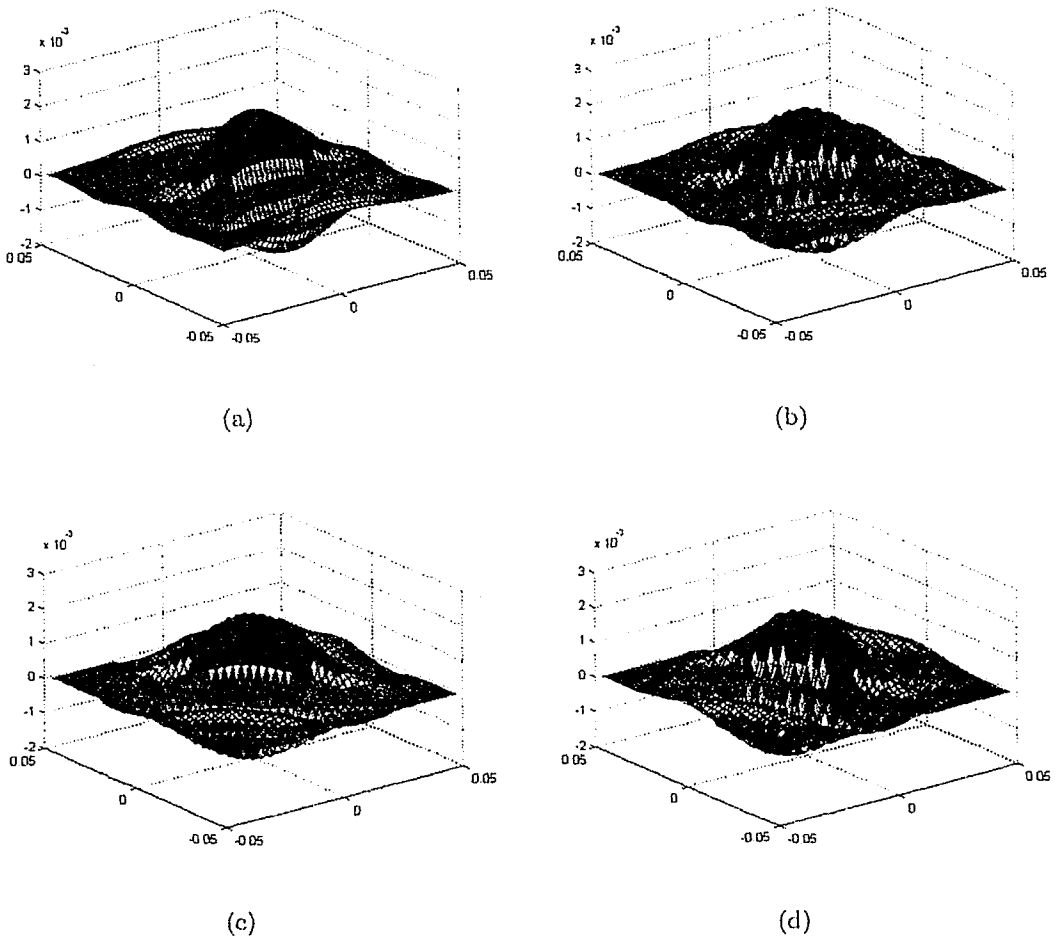
bank using six orientations 0°, 30°, 60°, 90°, 120° and 150° will generate a bank of 30 filters and the third filter bank is composed of 40 filters obtained by eight orientations for each radial frequency, i.e., 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135° and 157.5°. Filter parameters for three mentioned filter banks are presented in Table 3.1.

Table 3.1: Filter parameters for three filter banks

Filter Bank	Radial Frequency	Orientation
1	$4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2}$	0°, 45°, 90°, 135°
2	$4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2}$	0°, 30°, 60°, 90°, 120°, 150°
3	$4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2}$	0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°

The filter banks consisting of 20 and 30 filters using 45° and 30° orientations, respectively were used widely in previous research [6, 33, 11, 34]. Here the third filter bank using 22.5° angular bandwidths and orientations is introduced. In several experiments with a variety of different textures using all three filter banks, the preserved energies in the filtered images are computed. By comparing and sorting the filtered images according to the preserved energies, it was observed that by dividing the angular bandwidth by two, i.e., 22.5° in place of 45° and covering the frequency domain by this angular bandwidth, not only the textures with 45° orientations i.e., 0°, 45°, 90° and 135°, could be caught but also textures with 22.5° orientations i.e., 22.5°, 67.5°, 112.5° and 157.5°, could be better discriminated and classified. Gabor filters in spatial domain with 22.5° orientations are depicted in Fig. 3.2 and Fig. 3.3.

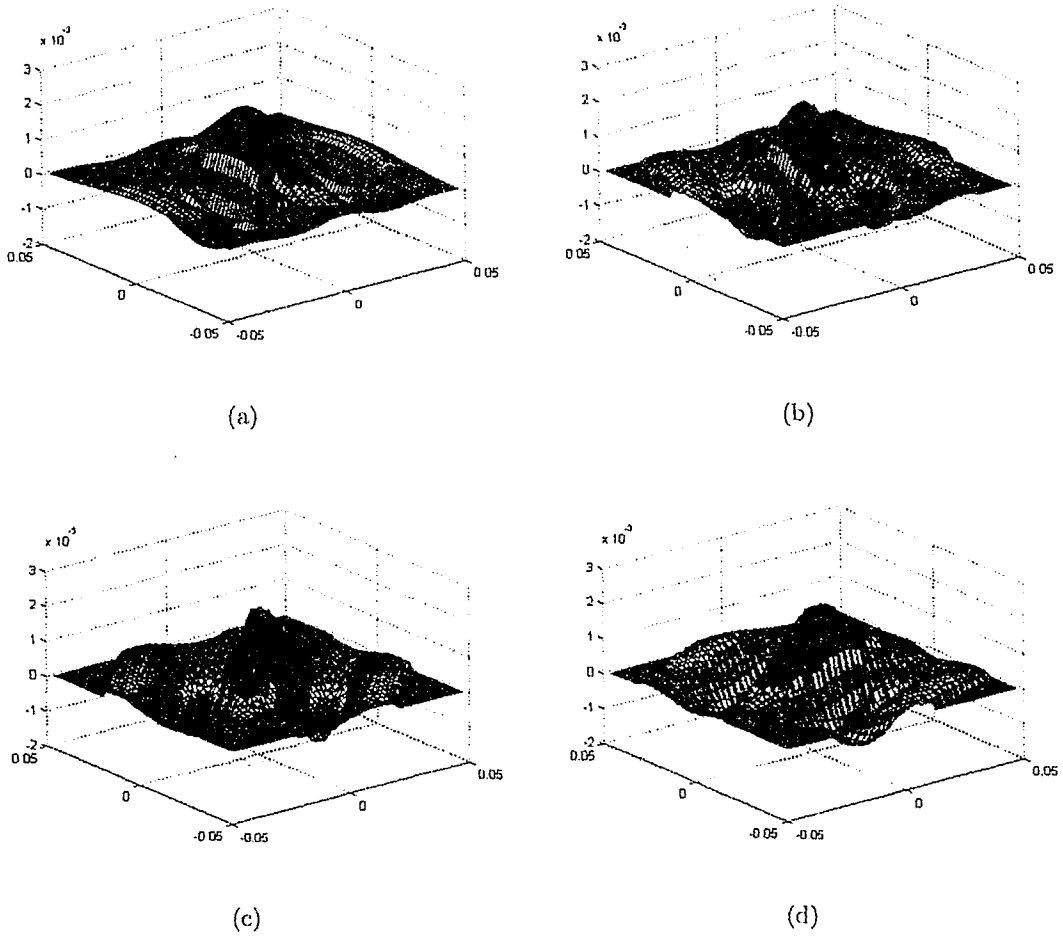
It was noticed that for some textures, maximum energy is preserved by filters with 45° orientations (and 22.5° angular bandwidths) and filters with 22.5° orientations (and 22.5° angular bandwidths) would preserve the maximum energy for the other textures. By applying



(a) (b)

(c) (d)

Figure 3.2: Gabor filters in spatial domain with orientations: (a) 0° (b) 22.5° (c) 45° (d) 67.5°



(a) (b)

(c) (d)

Figure 3.3: Gabor filters in spatial domain with orientations: (a) 90° (b) 112.5° (c) 135° (d) 157.5°

22.5° orientations for different radial frequencies and sorting the filtered images regarding preserved energies, it was observed that some of the filtered images with these orientations are among the ten best filters respecting the preserved energy of the original image. Hence for this filter bank, eight equally spaced orientations 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135° and 157.5° are used. To ensure proper coverage of the spatial-frequency domain, the angular bandwidth is set to 22.5°. Orientations and radial frequencies of the ten best filters using three mentioned filter banks are presented in Table 3.2. The best filters are selected and sorted descending regarding the preserved energies of synthetic image depicted in Fig. 3.4.

Figure 3.4: A sample texture image

In the second step a square function is used as nonlinearity and it causes the sinusoidal modulations in the output of a filter bank to be transformed to a square modulation. To smooth out the fluctuations in the specific texture or noise in the image, a Gaussian low pass filter is applied to the output of the filter bank. The size of the smoothing function is determined according to the size of the Gabor band pass filter. The impulse response of the Gaussian filter is:

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot exp\{\frac{-1}{2}(\frac{x^2}{\sigma^2})\} \ , \tag{3.2}$$

where

$$\sigma = \frac{1}{2\sqrt{2}f_r} \ , \tag{3.3}$$

Table 3.2: Orientations and Radial frequencies of the ten best filters using three filter banks

Angular Bandwidth	Radial Frequency and Orientation of the 10 best filters
45°	$64\sqrt{2}-90°, 64\sqrt{2}-135°, 64\sqrt{2}-0°, 64\sqrt{2}-45°, 32\sqrt{2}-90°,$ $32\sqrt{2}-0°, 32\sqrt{2}-135°, 32\sqrt{2}-45°, 16\sqrt{2}-0°, 16\sqrt{2}-45°$
30°	$64\sqrt{2}-90°, 64\sqrt{2}-150°, 64\sqrt{2}-0°, 64\sqrt{2}-120°, 64\sqrt{2}-60°,$ $32\sqrt{2}-90°, 64\sqrt{2}-30°, 32\sqrt{2}-0°, 32\sqrt{2}-60°, 32\sqrt{2}-150°$
22.5°	$64\sqrt{2}-90°, 64\sqrt{2}-157.5°, 64\sqrt{2}-135°, 64\sqrt{2}-67.5°, 64\sqrt{2}-112.5°,$ $64\sqrt{2}-0°, 32\sqrt{2}-90°, 64\sqrt{2}-45°, 64\sqrt{2}-22.5°, 32\sqrt{2}-67.5°$

and f_r is the radial frequency of band pass Gabor filter.

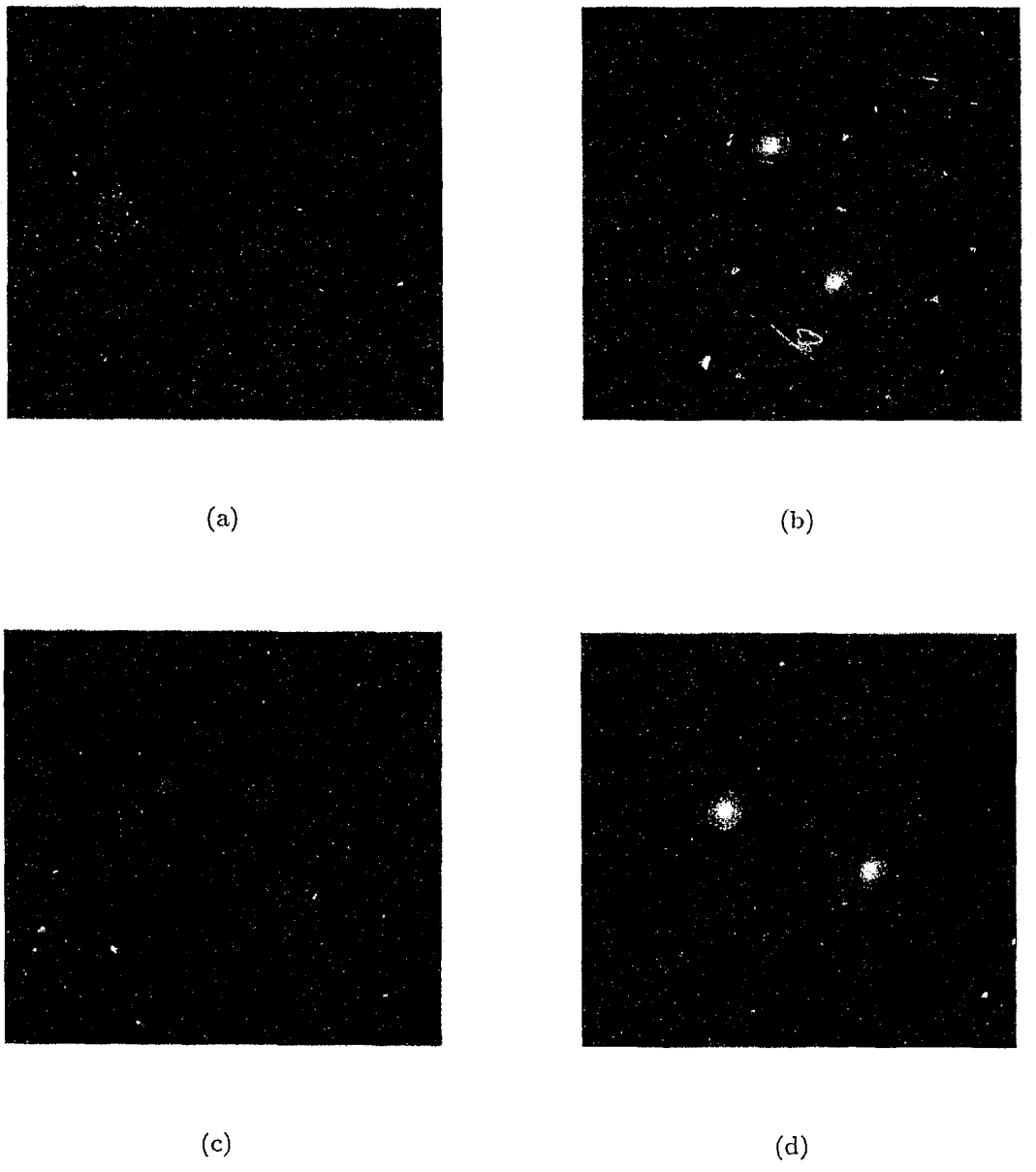
3.2 High and Low Frequency Component Extraction

Neither this filter bank nor the two others completely cover the corners of the frequency domain along the diagonals as are showed in Fig. 3.5 and Fig. 3.6. Since the coverage of the filter bank is not completely flat there are some regions that are not covered (very high frequency) and some other regions without a flat response (very low frequency). In the proposed method, the spatial implementation for the Gabor filter bank is optimized considering the diagonal high frequency and very low frequency components.

The low frequency component that is not included in Gabor filter bank is obtained by the equation 3.4 where radial frequency f_r is equal to zero and concludes a Gaussian low pass filter as:

$$h(x,y) = g(x,y).cos(2\pi f_r x) = g(x,y)\ . \tag{3.4}$$

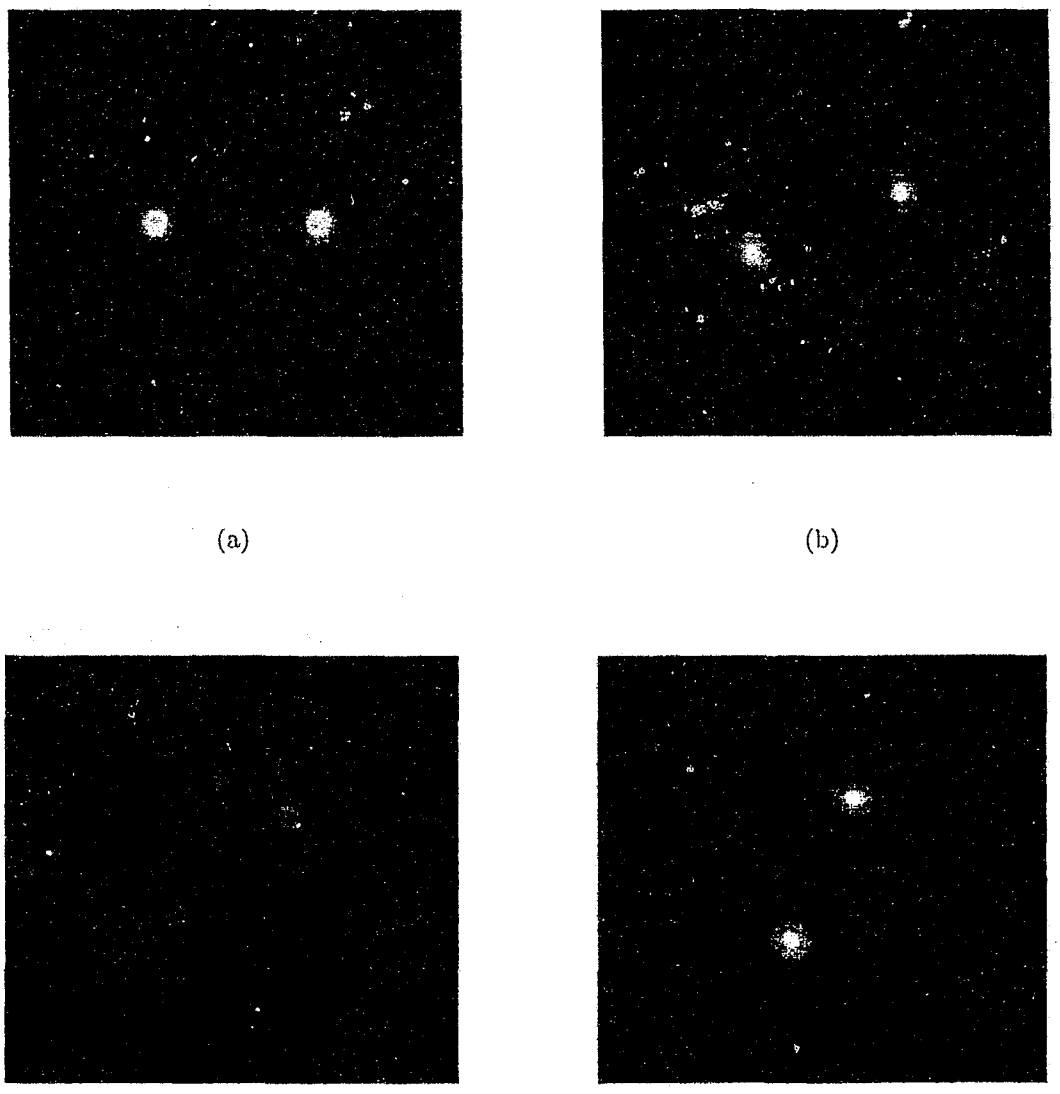
The high frequency component is computed by using the Gabor filter bank and Gaussian low pass filter as depicted in Fig. 3.7. The optimized filter bank which employed this high



(a) (b)

(c) (d)

Figure 3.5: Gabor filters in frequency domain with $64\sqrt{2}$ radial frequency and with orientations: (a) 0° (b) 22.5° (c) 45° (d) 67.5°



(a) (b)

(c) (d)

Figure 3.6: Gabor filters in frequency domain with $64\sqrt{2}$ radial frequency and with orientations: (a) 90° (b) 112.5° (c) 135° (d) 157.5°

frequency component is showed in Fig. 3.8. The high frequency component is computed by:

$$h_d(x,y) = im(x,y) * \delta(x,y) - \sum_{1}^{L} h_1(x,y) * im(x,y) - h_2(x,y) * im(x,y) \ , \tag{3.5}$$

where " * " indicates convolution and $h_d(x,y)$ is diagonal high frequency component and is equal to:

$$h_d(x,y) = h(x,y) * im(x,y) \ . \tag{3.6}$$

h_1 is Gabor filter impulse response, L is the number of filters in the filter bank and h_2 is Gaussian low pass impulse response.

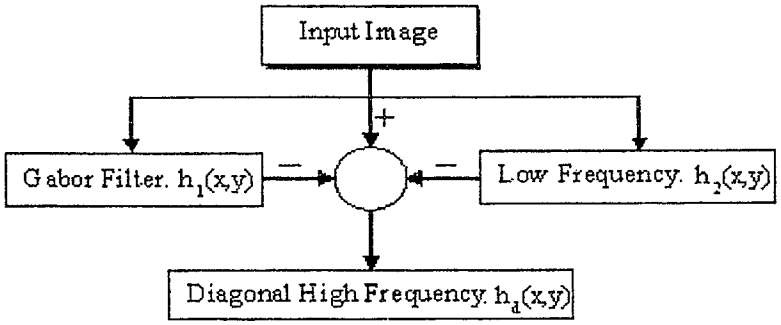


Figure 3.7: Diagonal High Frequency Component

3.3 Implementation of the Gabor Filter

The two dimensional (2D) Gabor function is composed of a sinusoidal plane wave of a particular frequency and orientation, and a Gaussian function. Equation 3.7 shows a complex 2D Gabor filter. The implemented even symmetric real component of the complex 2D Gabor filter is depicted in Equation 3.8:

$$h(x,y,f_r,\theta) = g(x,y,\theta) \cdot exp^{-2\pi j f_r x_\theta} \ , \tag{3.7}$$

$$h(x,y,f_r,\theta) = g(x,y,\theta).cos(2\pi f_r x_\theta) \ , \tag{3.8}$$

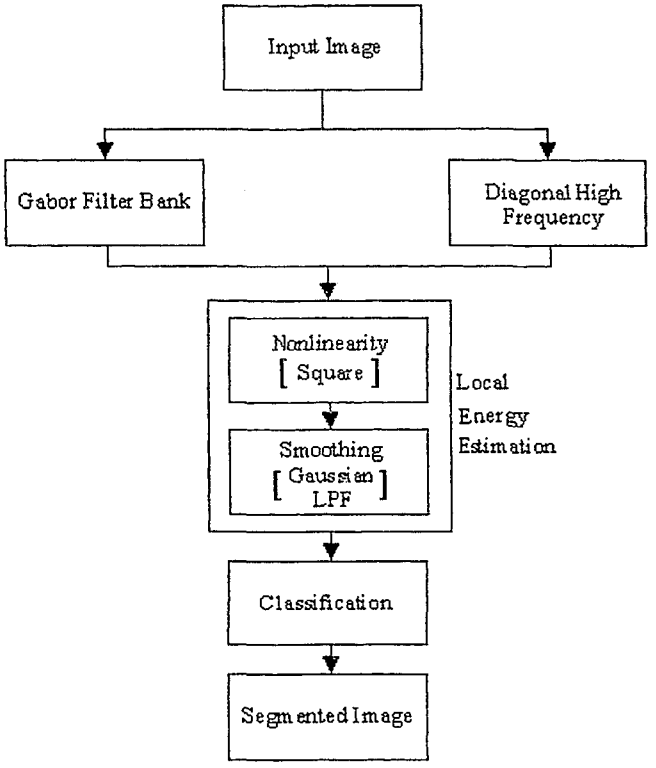


Figure 3.8: Optimized Filter Bank

and

$$g(x,y,\theta) = \frac{1}{2\pi\sigma_x\sigma_y} exp^{\frac{-1}{2}(\frac{x_\theta^2}{\sigma_x^2}+\frac{y_\theta^2}{\sigma_y^2})} , \tag{3.9}$$

where, x_θ and y_θ are:

$$x_\theta = xcos\theta + ysin\theta , \tag{3.10}$$

$$y_\theta = -xsin\theta + ycos\theta , \tag{3.11}$$

and modulation is specified by spatial-frequency and orientation:

$$f_r^2 = u_r^2 + v_r^2 , \tag{3.12}$$

$$\theta = tan^{-1}(v_r/u_r) . \tag{3.13}$$

σ_x and σ_y are the standard deviations of the Gaussian envelope along x-axis and y-axis, respectively. The 2D Gabor kernel is separable, that is, it can be decomposed and represented as two orthogonal 1D components, one parallel and the other perpendicular to the orientation θ:

$$h_x(x, f_r, \theta) = \frac{1}{\sqrt{2\pi}\sigma_x} exp^{\frac{-1}{2}(\frac{x_\theta^2}{\sigma_x^2})}.cos(2\pi f_r x_\theta) \ , \tag{3.14}$$

$$h_y(y, \theta) = \frac{1}{\sqrt{2\pi}\sigma_y} exp^{\frac{-1}{2}(\frac{y_\theta^2}{\sigma_y^2})} \ , \tag{3.15}$$

and

$$h = h_x.h_y \ . \tag{3.16}$$

The first component is an even symmetric real part 1D Gabor function and the second is a Gaussian function. As a result convolution of an even symmetric 2D Gabor with an image can be computed separately. Convolving a given image I with 2D Gabor function h will result:

$$I * h = I * (h_x * h_y^T) = (I * h_x) * h_y^T \ , \tag{3.17}$$

where h_x and h_y are horizontal and vertical Gabor kernels, respectively. By separating the 2D convolution into two 1D convolutions, the computation is reduced from $O(N^2M^2)$ to $O(2N^2M)$ for an $M \times M$ Gabor kernel and an $N \times N$ image.

3.4 Computing Features

In this step each channel corresponding to a different filter is tuned to a different radial frequency or orientation to capture local characteristics of different textures in the input image such as spatial-frequency, edge intensity and direction [38]. After applying the filter bank, local energy is estimated to compute the features. To estimate the local energy, square function is used as nonlinearity and low pass gaussian filter is used for smoothing. To determine the size of smoothing function, as it can be observed in Fig. 3.9 by increasing the radial frequency the Gabor function will be narrower, and a smaller smoothing function can be used. Conversely for the lower radial frequencies, wider Gabor function is obtained and

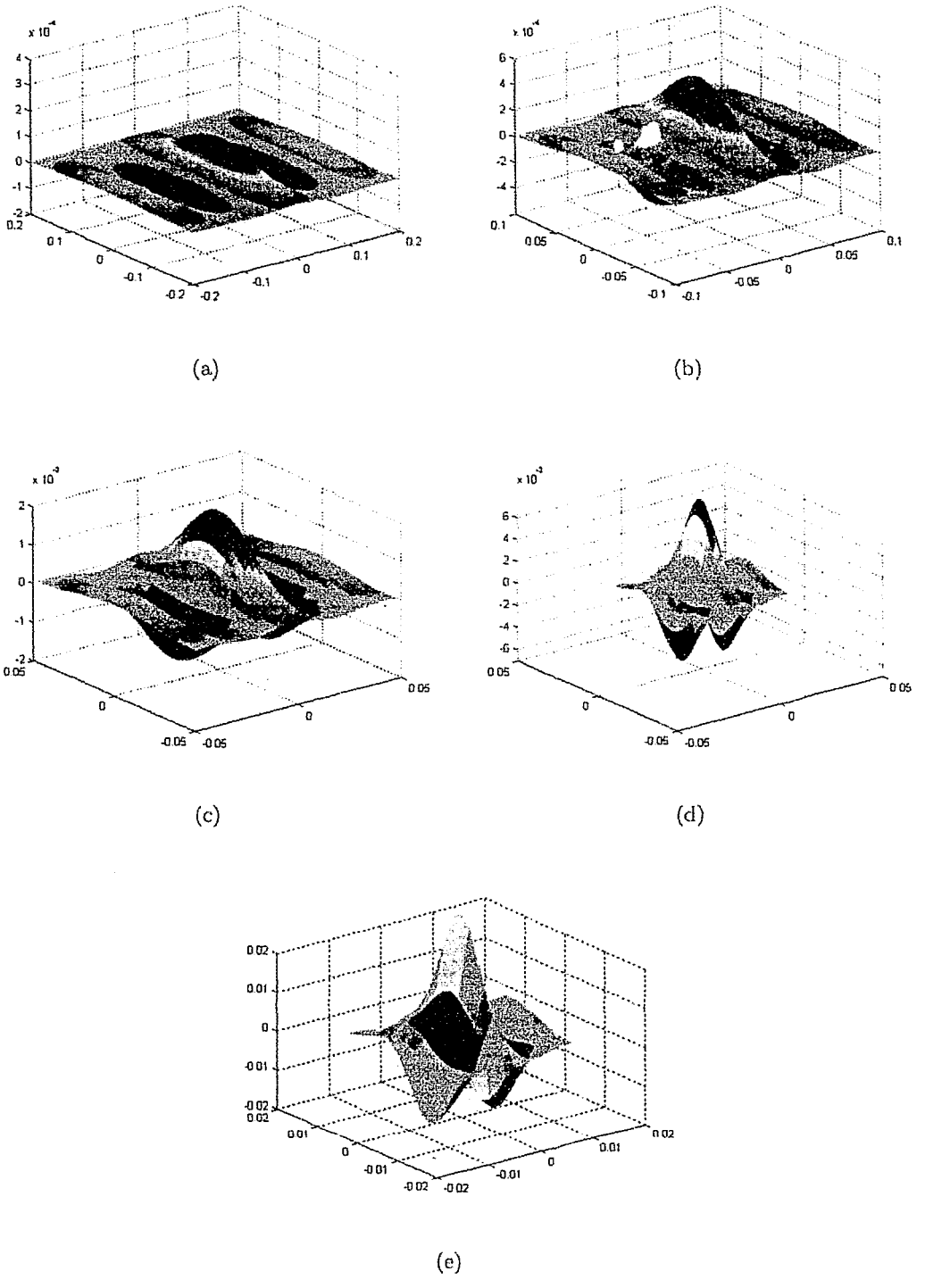


Figure 3.9: Gabor filters in spatial domain with orientation 90° and radial frequencies: (a) $4\sqrt{2}$ (b) $8\sqrt{2}$ (c) $16\sqrt{2}$ (d) $32\sqrt{2}$ (e) $64\sqrt{2}$

a larger smoothing function is required. After the second stage, a set of feature images or a feature vector corresponding to each pixel in the input image is generated. The dimension of feature vectors is equal to (or multiplied by an integer) the number of filters in the filter bank. Two methods to construct the feature vectors are used. In the first method, the feature vector for each pixel of the input image is composed of respected pixels from all feature images as depicted in Fig. 3.10. Having 42 filters in the filter bank results in 42 featured images that are used to form feature vectors. Each feature image corresponds to one element in the feature vectors and each feature vector corresponds to one image pixel. As a result the features are obtained for each pixel are 42 or the feature vectors are 42 dimensional.

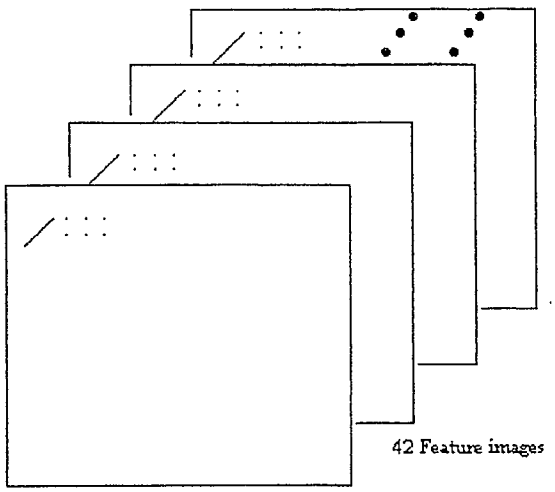


Figure 3.10: Feature vector for each pixel is formed by respected pixels from all feature images.

In the second method, the mean and variance are computed as the feature vector elements. The feature vector for each pixel of the input image is constructed by the mean and variance which are computed over the neighborhood window of the pixel in all feature images [39]. According to the periodicity of the textures in the input image the size of neighborhood window could change. For textured images with smaller periodicity the window centered by the regarded pixel would be smaller and would select the larger window for the textures

with larger periodicity. Filter bank is applied to the input image and filtered images are squared and smoothed in turn. To construct the feature vector for each pixel in the input image, the mean and variance are processed over the neighborhood window on all resultant images. Having 42 filters in the filter bank results in 42 filtered images and by computing mean and variance 84 features will be obtained for each pixel (i.e., the feature vectors are 84 dimensions).

Chapter 4

Segmentation by Artificial Neural Networks

In this chapter proposed technique for texture segmentation based on Neural Networks is presented. Extracted features by Gabor filter bank are segmented by Artificial Neural Networks. In the following sections feature reduction and segmentation are explained. The following ANNs concepts including (1) Competitive Networks, (2) Feature vector reduction, (3) Multi-Layer Perceptron and (4) Classification are discussed.

4.1 Introduction to Artificial Neural Networks

Although the field of neural networks has a history of some five decades, this field has found solid application only in the past fifteen years, and it is still developing rapidly. Hence, this field is distinctly different from the fields of optimization or control systems where the basic mathematics, and design procedures have been firmly established and applied for a long time. ANNs are a simulation of a real human nervous system that contains a collection of neurons which communicate with each other via connections known as axons. Such a model resembles axons and dendrites in the human nervous system. In neural networks, simple elements operate in parallel. The connections between elements determine the network function, as in nature. To perform a particular function, a neural network can be trained by adjusting the weights of the connections between elements. It is estimated that the human brain contains over 100 billion (10^{11}) neurons and 10^{14} synapses in the human nervous system. Studies of

brain anatomy on the neurons indicate more than 1000 connections (synapses) on the input and output of each neuron. Although the neuron's switch time is about a million times slower than current computer elements (a few milliseconds), their connectivity is a thousand times greater than today's supercomputers. To lead particular inputs to a specific target output, neural networks are trained and weights are adjusted. The first basic model for neural networks was proposed by McCulloch and Pitts in 1943 in terms of a computational model of "nervous activity". The proposed neuron was a binary device and each neuron has fixed threshold logic. This model leads the works of the other researchers such as Jhon von Neumann, Marvin Minsky, Frank Rosenblatt, and many others [40, 41, 42, 43].

4.1.1 Neural Networks Categories

The neural networks are commonly classified into three categories in terms of their corresponding training algorithms and how the synaptic weights are obtained:

1. Fixed-weights networks
2. Unsupervised networks
3. Supervised networks

The synaptic weights can be considered as the memory of a neural network that may be either pre-stored or adaptively trained by a learning mechanism. There is no learning required for the fixed-weight networks, so a learning mode is supervised or unsupervised. Associative memory networks are an instance of fixed-weight networks. They are mostly designed for the best recovering of the original pattern from an incomplete or distorted signal. The main characteristic of the fixed-weight association networks is that the synaptic weights are precomputed and pre-stored. Since they cannot adapt according to the change of the input environment, the fixed-weight models have limited applications.

Supervised Learning Rules

The mainstream of neural network model development has begun with supervised learning networks. Since the training data consist of many pairs of input/output training patterns, the learning will benefit from the assistance of the teacher. The weights may be updated by giving a new training pattern, $(t+1)^{th}$, as follows:

$$w_{ji}^{(t+1)} = w_{ji}^{t} + \Delta w_{ji}^{(t+1)} \tag{4.1}$$

An schematic diagram of a supervised model is showed in Fig. 4.1. Hebbian learning rule is

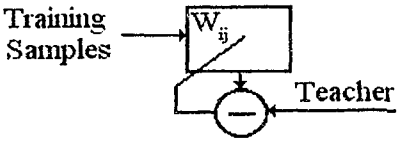


Figure 4.1: Supervised Learning

a typical example of supervised learning [44].

Unsupervised Learning Rules

The training set consists only of input training patterns in an unsupervised learning rule. Hence, the network will be trained without benefit of any teacher. Based on the experiences collected through the previous training patterns the network is trained and adapted. Fig. 4.2 shows typical schema for an unsupervised model. Typical example is the Hebbian based

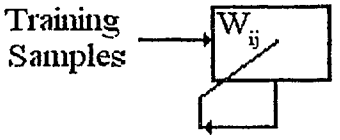


Figure 4.2: Unsupervised Learning

competitive learning rule. As an example, when a new pattern is determined to belong to a cluster, it will be included to that cluster and the representation of the cluster will be

affected by inclusion of the new pattern for example by changing the centroid of the cluster and in turn this will change the weights characterizing the classification network.

4.1.2 Neural Networks Characteristics

The functional descriptions of the connection network and the network activation can characterize a neural network model.

Activation Function

Each processing unit (neural cell) has a neuron value a_j which is propagated to other cells in the network through a network of unidirectional connections. There is a synaptic weight denoted by W_{ji} associated with each connection that determines the effect of the i^{th} cell to the j^{th} cell. To yield the net value n_j, the inputs to the j^{th} cell from other cells are accumulated together and mapped by a basis function to yield a new activation value n_j as depicted in Fig. 4.3. The net value is expressed by the basis function, $n_j(p, w)$, where p is

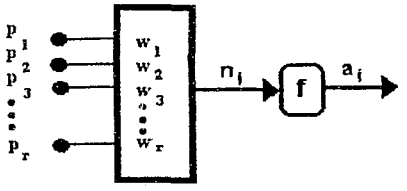


Figure 4.3: A sample neuron

input vector and w is (a row of) weight matrix:

$$n_j(p, w) = \sum_{1}^{r} w_{ji} p_i \ . \tag{4.2}$$

The final output a_j can be expressed as a function of the input and the weights. A nonlinear activation function of the neuron will receive and transfer the net value to the final output. Sigmoid function (f) is the most common activation function.

Connections

Connections can be divided into two major types:

1. Feedforward Connections: For all the neural network models, data are propagated from neurons of a lower layer forward to neurons of an upper layer via feedforward connections networks.

2. Feedback Connections: Data are brought from neurons of an upper layer back to neurons of a lower layer by feedback connections.

Neural Networks Size

There are one or more layers of hidden neurons between the input and output layers in a feedforward multi-layer neural network. The size of network depends on the number of layers and the number of hidden units per layer. The capabilities of the network is directly related to the number of hidden units. An optimal number of hidden units must properly be determined for the best network performance. In the literature sometimes the output layer is considered as another hidden layer, so to keep the concept uniform, the total number of layers are referred wherever is necessary in this text.

4.1.3 Neural Networks Applications

The strength of neural networks from an application driven perspective are adaptive, non-linear and parallel processing. Neural networks are used in many successful applications such as signal & image processing, computer vision, speech & character recognition, expert systems, remote sensing, medical image analysis, robotic processing, industrial inspection and scientific exploration. Some application domains of neural networks can be categorized as following:

1. Clustering

2. Classification

3. Association

4. Pattern completion

5. Optimization

Clustering and Classification

In this application, input patterns or signals are to be classified or recognized. A classifier should be trained such that when a slightly distorted version of pattern is presented it can still be correctly recognized. The two most important categories are [40]:

- Unsupervised Clustering: The synaptic weights of the network are trained using an unsupervised learning rule in this paradigm. Therefore the network adapts the weights and verifies the result exclusively based on the input patterns.
- Supervised Classification: This one is more suitable for the problems which involve a number of classes with more complex separating borders. For this application the training data consist of pairs of input pattern and teacher and the weights of the network are adopted based on the training pairs.

4.2 Dimension Reduction of Extracted Features

4.2.1 Principal Component Analysis

Feature extraction is a common problem in statistical pattern recognition. This problem introduces a process to transform data space to feature space. By reducing the dimension of data set, the data is represented by a reduced number of features that retain most of important information of the data. Assume that r-dimensional vector p is transformed to l-dimensional vector a, where $l < r$. The easiest way to reduce the dimension is truncating the vector p to reach l-dimensionality. Such dimension reduction causes a mean-square error (MSE) equal to the sum of the eliminated element's variances. If an invertible linear transformation T can be found such that the truncated vector T_p has the optimum mean square error, most of the intrinsic information content of data p would have be retained in feature a. To maintain an optimum mean square error, some of the components in transformation T should have low variance. Principal Component Analysis (PCA) optimizes

the rate of variance decreasing. The self-organized learning that is a Hebian-based learning algorithm can perform principal component analysis. Suppose P is a r-dimensional random vector representing the input space and S is the projection of P on a r-dimensional unit vector q. This projection is defined by [41]:

$$S = P^T q = q^T P \ , \tag{4.3}$$

where

$$\|q\| = (q^T q)^{\frac{1}{2}} = 1 \ . \tag{4.4}$$

The unit vectors q for which the variance is optimum can be obtained by:

$$Cq = \lambda q \ , \tag{4.5}$$

where C is the $r - by - r$ correlation matrix of the random vector P and is shown by:

$$C = E[PP^T] \ , \tag{4.6}$$

that is expectation of outer product of the vector P. The variance σ^2 of projection S is a function of unit vector q:

$$\sigma^2 = q^T C q \ . \tag{4.7}$$

The Eq. 4.7 is known as the eigenvalue problem and has nonzero ($q \ = 0$) solutions for some special values of λ that are defined as eigenvalues of the matrix C. Associated unit vectors with eigenvalues are eigenvectors of correlation matrix C. Denoting the eigenvalues of C by $\lambda_1, \lambda_2, ..., \lambda_r$ and associated eigenvectors by $q_1, q_2, ..., q_r$, respectively, it can be written:

$$Cq_j = \lambda_j q_j, \ j = 1, 2, ..., r \ . \tag{4.8}$$

Corresponding eigenvalues are arranged in descending order:

$$\lambda_1 > \lambda_2 > ... > \lambda_r \tag{4.9}$$

so, $\lambda_1 = \lambda_{max}$ and by define $Q = [q_1, q_2, ..., q_r]$, it would be:

$$CQ = Q\Lambda \ , \tag{4.10}$$

where Λ is :

$$\Lambda = diag[\lambda_1, \lambda_2, ..., \lambda_r] \ . \tag{4.11}$$

By decomposing the correlation matrix C to its eigenvectors will conclude:

$$C = \sum_{i=1}^{r} \lambda_i q_i q_i^T \tag{4.12}$$

Eigendecomposition of matrix C is principal component analysis and the following result is made from eigenstructure of principal component analysis: eigenvectors of the correlation matrix C define the unit vectors q_j that represent the principal directions along which the variance has its optimal value:

$$p = Qs = \sum_{j=1}^{r} s_j q_j \ , \tag{4.13}$$

where

$$s_j = q_j^T p, \ j = 1, 2, ..., r \ , \tag{4.14}$$

or:

$$s = Q^T p \ . \tag{4.15}$$

4.2.2 Hebbian Based PCA

The statistical method of PCA has a close correspondence with self-organized neural networks. A single linear neuron can evolve into the first principal component of the input data. Synaptic weights will be adapted with a Hebbian-type learning rule [41]. Consider the simple neuron depicted in Fig. 4.4. Since the model is linear, the output is a linear combination of its inputs. An R dimensional vector $p = [p_1, p_2, ..., p_r]$ is received in the input and pass through the corresponding synaptic weight vector $W = [w_1, w_2, ..., w_r]$. The output a is defined by:

$$a = \sum_{i=1}^{r} w_i p_i \ . \tag{4.16}$$

According to the Hebbian learning, synaptic weights vary with time and grow strong as follows:

$$w_i(t+1) = w_i(t) + \eta a(t) p_i(t), \ i = 1, 2, ..., r \tag{4.17}$$

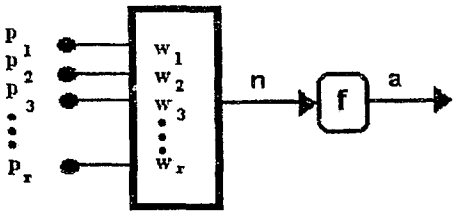


Figure 4.4: A single linear neuron

where η is the learning rate parameter. This basic form of the learning rule may lead to unlimited growth of the synaptic weights $w_i,\ i = 1, 2, ..., r$ that is unacceptable on physical grounds. By using some saturation or normalization form this problem would be solved as:

$$w_i(t+1) = w_i(t) + \eta a(t)[p_i(t) - a(t)w_i(t)],\ i = 1, 2, ..., r \tag{4.18}$$

The single linear neuron with the Hebbian based learning role extracts the first principle component of the input. This model can be expanded to a single layer feedforward network of linear neurons to principal component analysis purpose [41].

4.2.3 Competitive Networks

Competitive Learning Networks are unsupervised networks whose synaptic weights adapt according to unsupervised learning rules. These models can learn in the absence of teacher's guidance and exclusively based on the input training patterns [43, 40]. A basic competitive learning network has one feedforward excitatory layer of input neurons. An input pattern p is a sample point in the r-dimensional real or binary vector space. Unsupervised classification procedures are often based on some kind of clustering strategy, which forms groups of similar patterns. The clustering technique is very useful for pattern classification problems. Furthermore, it plays a basic role in many competitive learning networks. For a clustering procedure, it is necessary to define a similarity measure to be used for evaluating how close the patterns are. Some popular measures are: (1) Inner Product, (2) Euclidean distance and (3) Weighted Euclidean Distance. Among them the most common is the Euclidean distance. When several neurons are grouped into a single layer and a competitive network is formed,

each neuron that is a single processing element, shares its processing functions with other neurons and responds to a group of input vectors in a different cluster maximally. This layer of neurons can classify any input vector, since the neuron with the strongest response for a given input identifies the cluster to which the input vector most likely belongs to. In this layer no external judge is to decide which neuron must win, hence the neurons must decide who is the winner themselves and decision making process needs communication among all neurons in the layer [40].

To reduce the feature vector dimension a competitive network is used. In a competitive layer the neurons are distributed in order to recognize frequently presented input vectors in an unsupervised manner. The competitive transfer function accepts a net input for each neuron in the layer and returns neuron outputs of 0 for all neurons except for the winner, which outputs 1. In a competitive layer each neuron competes to respond to an input vector p, the neuron whose weight vector is closest to p gets the highest net input and, therefore, wins the competition and outputs 1 and all other neurons output 0. Considering the single layer feedforward network depicted in Fig. 4.5 and making the following assumptions:

1. Neurons in the layer are linear ,
2. Network has r inputs and l neurons that generate l outputs ,
3. $l < r$, it means outputs of network are fewer than inputs ,

the output $a_i(t)$ is calculated as:

$$a_i(t) = \sum_{i=1}^{r} w_{ji}(t)p_i(t), \quad j = 1, 2, ..., l \quad . \tag{4.19}$$

The set of synaptic weights w_{ji} are subject to training and adjusted during the training phase. In this phase to adjust the winner so as to move it closer to the input the weights of neurons are adjusted by:

$$\Delta w_{ji}(t) = \eta a_j[p_i(t) - \sum_{k=1}^{j} w_{ki}(t)a_k(t)], \quad i = 1, 2, ..., r \ \ \& \ \ j = 1, 2, ..., l \ , \tag{4.20}$$

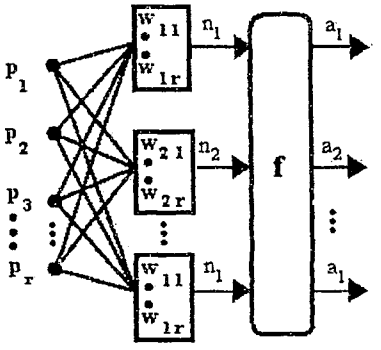


Figure 4.5: A Feedforward network with a single layer of processing neurons

or in matrix notation as:

$$\Delta w_j(t) = \eta a_j p'(t) - \eta a_j^2(t) w_j(t), \quad j = 1, 2, ..., l \ , \tag{4.21}$$

and:

$$p'(t) = p(t) - \sum_{k=1}^{j-1} w_k(t) a_k(t) \ , \tag{4.22}$$

where p, w and η are an input vector, a weight vector and a learning rate respectively. Since this learning rule allows the weights to learn an input vector, it is useful in recognition applications. Thus, the neuron whose weight vector is closest to the input vector will be updated to be even closer. As a result the winning neuron is more likely to win the competition when a similar vector is provided, and less likely to win if a very different input vector is provided. During training, as more inputs are presented, each neuron in the layer closest to a group of input vectors adjusts its weights to more closely resemble the inputs. Eventually, the competitive network will categorize the input vectors and every cluster of similar input vectors will have a neuron that identifies the presented vector if it belongs to that cluster [41]. According to the equation 4.21 the following results are made:

1. The first principal component of the input p is discovered by the first neuron in the Fig. 4.5, since $p'(t) = p(t), \quad j = 1$.

2. The second neuron converges to the first component of $p'(t)$ or equivalently to the second principal of $p(t)$. The reason is that the first principal component or the first

eigenvector of the correlation matrix C has been extracted by the first neuron by:

$$p'(t) = p(t) - w_1(t)a_1(t), \quad j = 2 \ . \tag{4.23}$$

3. With the same reasoning the third neuron converges to the third principal component of p(t) or the first principal component of the $p'(t)$ where $p'(t)$ is equal to:

$$p'(t) = p(t) - w_1(t)a_1(t) - w_2(t)a_2(t), \quad j = 3 \ . \tag{4.24}$$

4. The remaining neurons of the competitive feedforward network of Fig. 4.5 proceed in the same way and converge to the rest of the eigenvectors of the correlation matrix C by decreasing eigenvalue order.

4.2.4 Dimension Reduction

The important practical application of PCA is providing an effective method to reduce the dimension of input data space. By eliminating those linear combinations in Equation 4.13 that have small variances the number of necessary features for effective data representation can be reduced. To approximate the data vector p by expanding Equation 4.13 and truncating after l terms:

$$\hat{p} = \Sigma_{j=1}^{l} s_j q_j = [q_1, q_2, ..., q_l] \begin{bmatrix} s_1 \\ s_2 \\ ... \\ s_l \end{bmatrix}, l < r \ , \tag{4.25}$$

having the input vector p, the retaining principal components are:

$$\begin{bmatrix} s_1 \\ s_2 \\ ... \\ s_l \end{bmatrix} = \begin{bmatrix} q_1^T \\ q_2^T \\ ... \\ q_l^T \end{bmatrix} p \tag{4.26}$$

that is the mapping from data space to feature space or linear projection from R^r to R^l. To reduce the dimension of extracted features by narrow band Gabor filter bank composing of 42 filters, competitive layer is trained by applying the training samples that are selected

randomly from different regions of the image. To select the number of neurons in the network that is equal to the number of most important eigenvectors, 42 neurons that is the total number of possible eigenvectors are selected as the initial value and the layer is trained with training samples equal to 6% of input data and the classification error for the network is computed. By pruning the neurons in the network, repeating the same approach and computing the classification error, the appropriate number of neurons in the layer is chosen. After selecting the appropriate number of neurons the competitive layer will be trained. After competitive network is trained and the network converged, the estimated eigenvectors that are weight vectors of the network are obtained and regarded as quantizing mask coefficients. These coefficients are used to apply on the extracted feature vectors by Gabor filter bank correspond to pixels in the input image. The resultant weight vectors of a competitive layer are applied to the features prior to feed in the classifier and will reduce the dimensionality of feature vectors as depicted in Fig. 4.6. In order to train competitive network, sample vectors are selected by random among the extracted feature vectors obtained by Gabor filter bank in the previous step. According to the following criterions; the classification error and dimension of quantized feature vectors, 21 neurons are selected for competitive network as the optimum number of neurons.

4.3 Image Segmentation using Extracted Features in Previous Phase

4.3.1 Supervised Network Models

Main stream of the development in neural networks is represented by supervised learning networks. Adaline & Madaline networks, perceptron network, and various multi-layer networks are examples of well known pioneering supervised networks. Supervised learning network involves two phases:

- Learning phase
- Retrieving phase

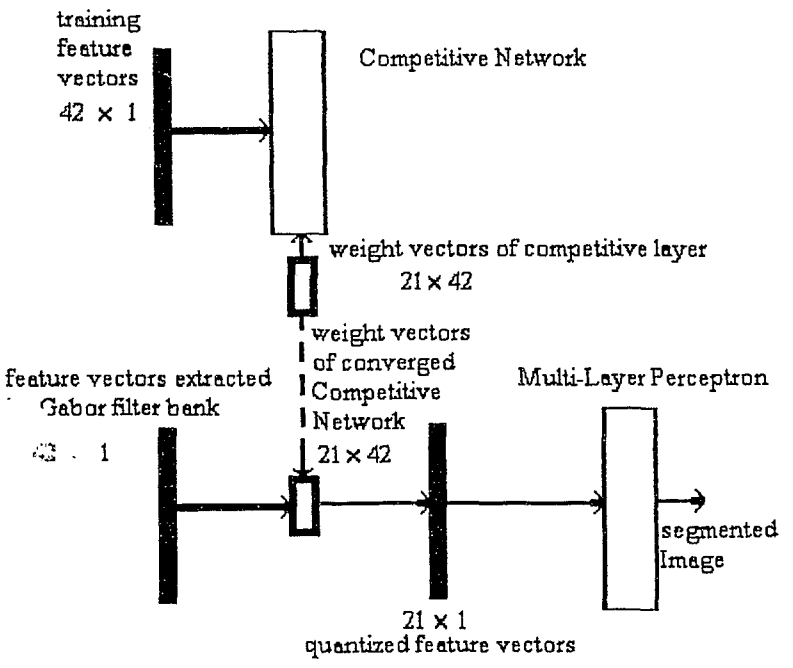


Figure 4.6: Feature reduction and segmentation by CPMLP-GFHF

The training patterns are provided as input-teacher pattern pairs in supervised training:

$$[X, T] = [x_1, t_1], [x_2, t_2], ..., [x_m, t_m] \tag{4.27}$$

where m is the number of training pairs. The supervised learning method is based on the optimization of a training cost criterion and depends on the information that is provided by the teacher. The most important case is represented based on the least square error approximation.

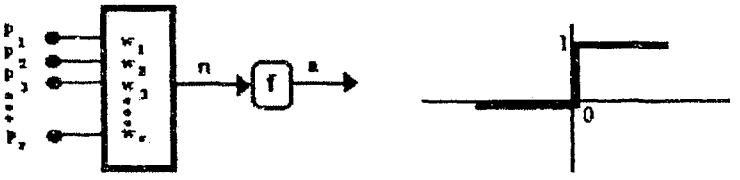
4.3.2 Perceptron

Training is a procedure whereby a network is adjusted to do a particular task. A learning method determines how next changes that might be made in a network would be driven. In this section supervised learning rules are introduced. One of the simplest perceptrons that was created by Rosenblatt was a single-layer perceptron whose weights could be adjusted

during training to produce a correct desired vector when the corresponding input vector is presented. The perceptron learning rule is used as training technique. Due to its ability to generalize from its training vectors and learn from initial random samples, perceptron has generated great interest. Perceptrons are especially suited for simple problems in pattern classification. For the problems they can solve, perceptrons are fast and reliable networks [41, 40, 36].

Linear Perceptron for Binary Classification

A perceptron neuron, which uses the hard-limit transfer function is shown in Fig. 4.7. Each external input is weighted with an appropriate weight w_i, and the sum of the weighted inputs is sent to the hard-limit transfer function. If the net input into the transfer function



(a) (b)

Figure 4.7: (a) A single linear neuron (b) Linear activation function

is equal to or greater than 0, the perceptron neuron produces a 1; otherwise it produces a 0. A perceptron having the hard-limit transfer function is able to classify input vectors by dividing the input space into two regions. Outputs will be 0 if the net input n is less than 0, or 1 if the net input n is 0 or greater. A binery linear separable input space is depicted in Fig. 4.8.

4.3.3 Linear Perceptron for Multiple Classification

To classify input space to more than two regions, the perceptron network consists of a single layer of l perceptron neurons connected to r inputs through a set of weights $w_{i,j}$ as depicted in Fig. 4.9. This perceptron layer can classify input vectors by dividing the input space

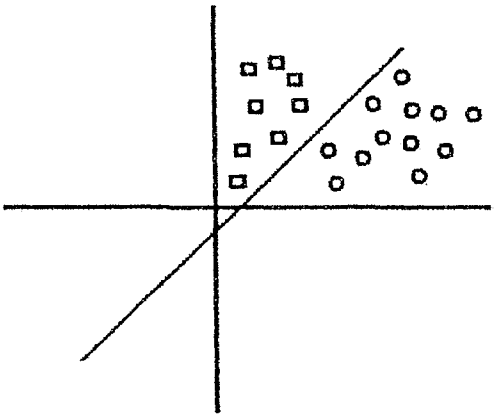


Figure 4.8: A linear separable problem

into l regions according to the number of neurons in the layer. The network indices i and j indicate that $w_{i,j}$ is the strength of the connection from the j^{th} input to the i^{th} neuron.

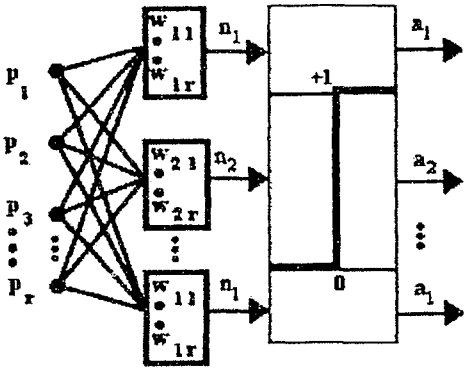


Figure 4.9: A single layer of linear perceptrons

Nonseparable Clusters and Nonlinear perceptron

The linear Perceptron is applicable only when the classes of patterns are known to be separable by linear decision boundaries. If patterns from different classes overlap in the border areas, some probabilistic criteria can be adopted to handle this situation. Thus, use a nonlinear decision boundary would be effective. Having a nonlinear decision boundary, the

nonlinear perceptron offers a much greater domain of practical applications. A single-layer feedforward network of l sigmoid neurons having r inputs is shown in Fig. 4.10.

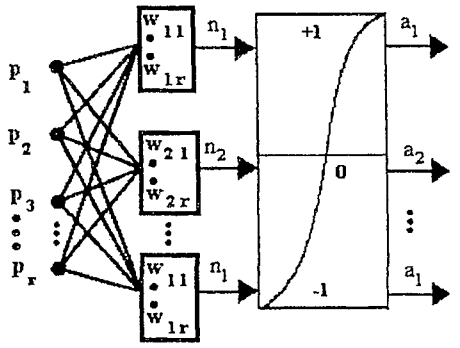


Figure 4.10: A single layer feedforward of sigmoid neurons

4.3.4 Nonlinear Multilayer Backpropagation Networks

A Multi-Layer Perceptron (MLP) has multiple layers with nonlinear transfer functions to learn nonlinear relationships between input and output vectors as depicted in Fig. 4.11. The benefits of cascading multiple layers of nonlinear units can be given as:

1. Linear approximation networks are too restrictive and nonlinear approximation networks offer much greater capacity.

2. In order to enhance the approximation capabilities, it is critical to expand a single-layer structure to a multilayer network.

A three-layer network is depicted in Fig. 4.11. Feedforward networks often have one or more hidden layers of sigmoid neurons. MLP is trained by adjusting the weights using Least Square Error (LSE) that minimizes the mean square error as shown by Equation 4.28:

$$E = \sum_{1}^{N} (R_k - O_K)^2 \ . \tag{4.28}$$

The total square error between the desired R_k class and the actual output O_k is calculated for N neurons in output layer K. To train the neural network, the gradient is determined

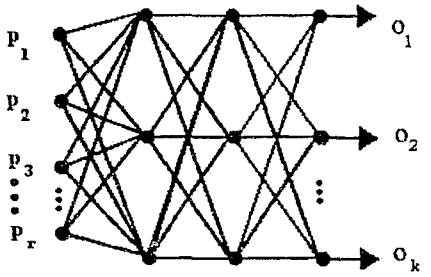


Figure 4.11: A feedforward multi layers of sigmoid neurons

by using a backpropagation technique which involves performing computations backwards through the network. After the backpropagation network is trained properly, it typically provides reasonable answers when presented with inputs that it has never seen. Commonly, a new input vector similar to past inputs used during training, leads to an output similar to correct values.

4.3.5 Backpropagation Algorithm

The Backpropagation algorithm offers an effective approach to the computation of the gradients and can be applied to any optimization formulation. The activation function usually is a sigmoid function. The objective of this algorithm is to train the weights w_{ji} so as to minimize error E. The basic gradient learning formula is:

$$w_{ji}^{(t+1)} = w_{ji}^{t} + \Delta w_{ji}^{(t+1)} \tag{4.29}$$

In order to derive the backpropagation learning rule, the chain rule is used to rewrite the gradient error for each pattern as the product of two partial derivatives. The change in error as a function of the net input is reflected by the first partial derivative, and by the second partial derivative, the effect of a weight change on a change in the net input is reflected [36, 41]. Therefore, the error becomes:

$$\partial E/\partial w_{ji} = [\partial E/\partial net_j][\partial net_j/\partial w_{ji}] \ . \tag{4.30}$$

Since the equation for the net input to a unit is known, the second partial derivative can be calculated directly:

$$\partial net_j/\partial w_{ji} = \partial(\sum w_{ji}o_i)/w_{ji} = o_j \ . \tag{4.31}$$

If δ be defined as the negative of the first partial derivative:

$$\delta_j = -\partial E/\partial net_j \ . \tag{4.32}$$

Thus, the appropriate change in the weight w_{ji} with respect to the error E can be written as:

$$\Delta w_{ji} = \eta \delta_j o_j \ , \tag{4.33}$$

where η is the learning rate. The next step to derive the backpropagation learning rule is determining δ_j for each unit in the network. It turns out that δ can be calculated recursively by backpropagating the error from the outputs to the inputs. However, the chain rule should be applied again to compute δ_j. The derivative error δ_j can be written as the product of two partial derivatives:

$$\delta_j = -(\partial E/\partial o_j)(\partial o_j/\partial net_j) \ . \tag{4.34}$$

At first the calculation of the second factor is considered:

$$o_j = f(net_j) \ , \tag{4.35}$$

then:

$$\partial o_j/\partial net_j = f'(net_j) \ . \tag{4.36}$$

The sigmoid function has an elegant derivative and it's derivative with respect to the net input simplifies to:

$$f'(net_j) = f(net_j)(1 - f(net_j)) \ . \tag{4.37}$$

The first partial derivative can be calculated considering two cases:

1. If u_j is an output unit of the network, then it follows directly from the definition of E that:

$$\partial E/\partial o_j = -2(t_j - o_j) \ . \tag{4.38}$$

Substitute this for δ_j back into the equation, will obtain:

$$\delta_j = 2(t_j - o_j)f'(net_j) \ . \tag{4.39}$$

2. If u_j is not an output unit of the network, the chain rule is used again to write:

$$\begin{aligned} \partial E/\partial o_j &= \sum[\partial E/\partial net_k][\partial net_k/\partial o_j] \\ &= \sum[\partial E/\partial net_k][\partial(\sum w_{kj}o_j)/\partial o_j] \\ &= \sum[\partial E/\partial net_k]w_{kj} \\ &= -\sum \delta_k w_{kj} \end{aligned} \tag{4.40}$$

Combining two mentioned cases to obtain δ_j for all neurons of the network provides a recursive procedure which is used to update weights of the network. The backpropagation learning algorithm is summarized as follows. The weight on each connection is changed by an amount during learning, that is proportional to the derivative error δ. The weight change for the weight connecting unit u_i to u_j using gradient descent on the error E is given by:

$$\Delta w_{ji} = \eta \delta_j a_j \tag{4.41}$$

and w_{ji}^{t+1} is computed by:

$$w_{ji}^{(t+1)} = w_{ji}^{t} + \Delta w_{ji}^{(t+1)}$$

where η is the learning rate. The error signal δ_j according to u_j is given by case 1 or 2 (if u_j is output neuron, case 1, otherwise case 2).

4.3.6 Segmentation by a 3-Layer Perceptron

A 3-layer perceptron, which is used to accomplish the segmentation task, is depicted in Fig. 4.12. The proposed MLP uses the sigmoid transfer function in all three layers. During training, random selected quantized feature vectors are assigned to proper classes. Although the extracted feature dimension is 42 or 84 according to the the feature computing method, the

quantized feature dimension is 21. After MLP is trained, quantized feature vectors corresponds to each pixel of the image are segmented to proper regions. The learning method is backpropagation and least mean square error is employed as the performance of the network. After a large number of epoches the performance criteria will converge and the network will be fed with the test data. A typical training curve of the proposed MLP is shown in Fig. 4.13. While it is possible to get excellent fits to the training data, the application of

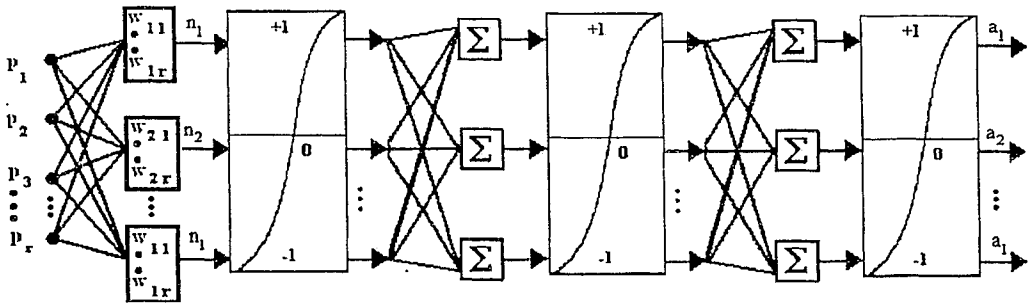


Figure 4.12: Proposed MLP for segmentation

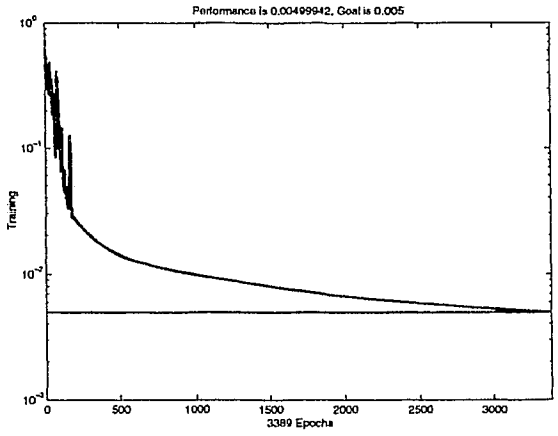


Figure 4.13: A typical convergence curve using backpropagation learning

backpropagation may face some difficulties in terms of computation. Some practical aspects are considered to get good results when applying MLP for segmentation. These practical aspects are discussed in Chapter 5 when results are discussed.

Chapter 5

Implementation Aspects of MLP and Gabor Filters

In this chapter implementation aspects of MLP and Gabor filter bank are presented.

5.1 Practical Aspects of MLP

Some practical aspects are considered and presented here to get good results when implementing MLP for segmentation. These aspects are discussed in following section and include selecting the number of layers and the number of neurons in each layer.

5.1.1 Selecting the Number of Layers

In a wide variety of applications generalized linear models are adequate and hidden layers may not be needed at all. Cascading multiple layers of linear networks has no real benefit in linear systems and the product of the weight matrices of different layers provides an equivalent weight matrix for the network. Having nonlinear neurons in hidden layers makes a quite different situation and more layers are used, more powerful networks are created. However, a simple linear model may provide better generalization than a complicated nonlinear model. When the function we want to approximate is mildly nonlinear or enough data is not available to estimate the nonlinearities by the network accurately, a simple linear model is preferred. Although excessive number of layers provides more complex function approximation, in practice it may not be productive and slower convergence may be caused

in the backpropagation learning. Therefore, identifying the proper number of layers in the network is essential. In general, to approximate any nonlinear function, two layer networks are adequate and for any polyhedral separable decision regions, three layer networks are sufficed. As a result, for most applications two or three layers are adequate [45].

To select the right number of layers for the proposed MLP we should consider the nature of our problem that is image segmentation. Since classifying different textures in the given image is a nonlinear model, a nonlinear function should be approximated. Therefore a single layer MLP is not appropriate and proposed MLP must have two or more layers including one output layer and at least one hidden layer. As it was mentioned before, for universal approximation of any nonlinear function, two layer network should be adequate. On the other hand a three layer network suffices for polyhedral separable decision region. As a result a MLP with two or three layers should be adequate for our application. To select the right number of layers between two and three layer perceptrons, both MLPs were implemented and fed with the feature vectors. According to the training convergence curve, the required time (number of epoches) for training and classification error, the number of layers for MLP are chosen. The results are discussed as follows. In Fig. 5.1 the performance goal i.e. the estimated error computed by least mean square is equal 0.01 and the maximum number of epoches is set to 3000. As we can observe in Fig. 5.1 according to the number of epochs the convergence curve of the 3 layer perceptron converge faster than the 2 layer perceptron to the same performance goal. The classification error of 3 layer perceptron in this case is less than that of 2 layer perceptron. It was 11.6 percent for 3 layer and 12.33 for 2 layer perceptron. On the other hand as depicted in Fig. 5.2 after passing the same number of epoches, the achieved performance of 3 layer perceptron is better than that of 2 layer perceptron. This performance is 0.0024 for 3 layer perceptron and 0.0045 for 2 layer as well. The measured classification error is 8.6 percent for 3 layer and 9.28 for 2 layer, therefor achieved performance is better for 3 layer again in this case. In Fig. 5.3, although the performance of 3 layer is better than 2 layer i.e. 0.0018 for 3 layer perceptron and 0.0040 for 2 layer, the observed classification error is 10.87 for 3 layer and 10.01 for 2 layer. Just in

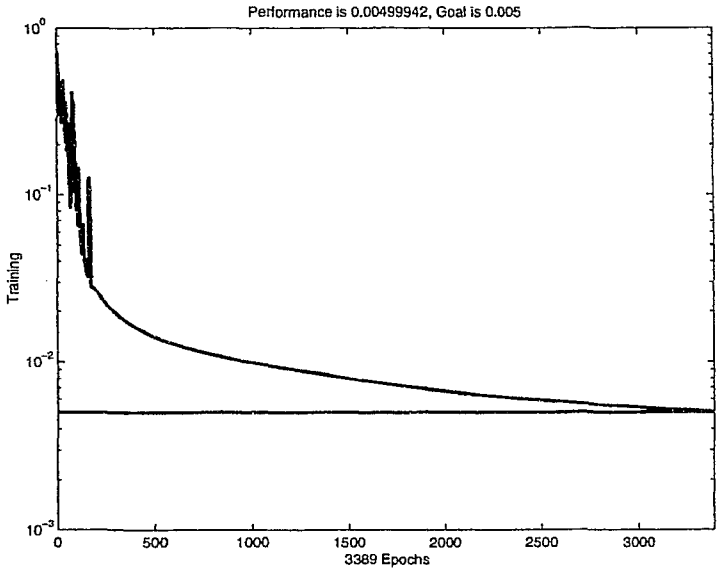
such cases the measured classification of 2 layer is better than that of 3 layer. The reason of having bigger classification error despite achieving better performance goal in training could be explained by overfitting.

5.1.2 Overfitting

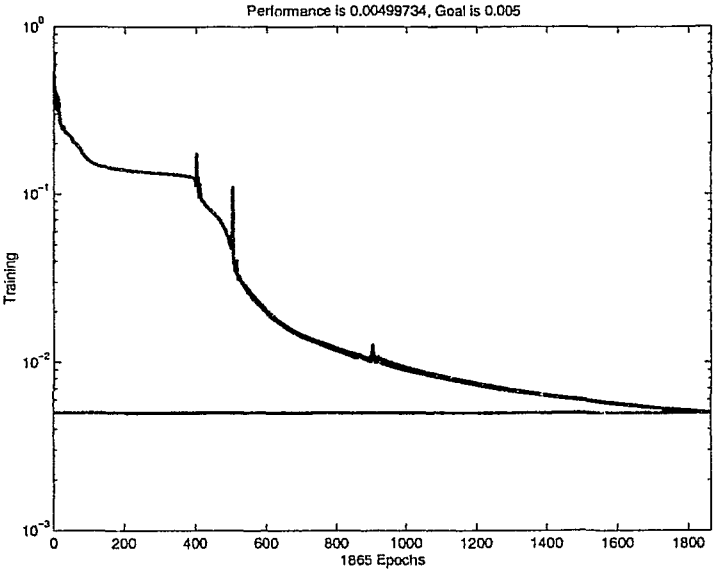
Overfitting can be explained as getting large error in test phase in spite of driving the error on the training set to a very small value. It happens when the network has memorized the training examples, but it has not learned to generalize the new situations. There are some methods to improve network generalization and to prevent from overfitting. One of these methods is to use a network that is just large enough to provide an adequate fit. By employing a larger network, the estimated function by the network is more complex. If we use a small enough network, it is less probable to overfit the data. Since in most cases the proposed 3 layer perceptron has better performance and better classification error than the 2 layer perceptron, we prefer to keep the number of layers equal to 3. Another method to prevent overfitting is driving the error on the training set to a small enough value to maintain both a good performance and prevent overfitting or achieve better generalization. As we can observe in Fig. 5.3(a) which 2 layer perceptron reached the better classification error than 3 layer one, the performance goal is obtained by 3 layer is smaller than that of 2 layer and is the smallest value that was achieved. As a result, we set the performance to a small enough value for better generalization.

5.1.3 Selecting the Number of Neurons in Layers

For a MLP with only one hidden layer, a large number of hidden units is required for a good approximation of hills and valleys. Although one hill or valley can be represented easily when there is one unit in the second hidden layer (output layer) of the network, for learning two hills or valleys having only one unit in the second hidden layer is of little benefit. Using two units in the second hidden layer enables the network to approximate two hills or valleys easily. Each additional unit in the last hidden layer enables the network to learn another

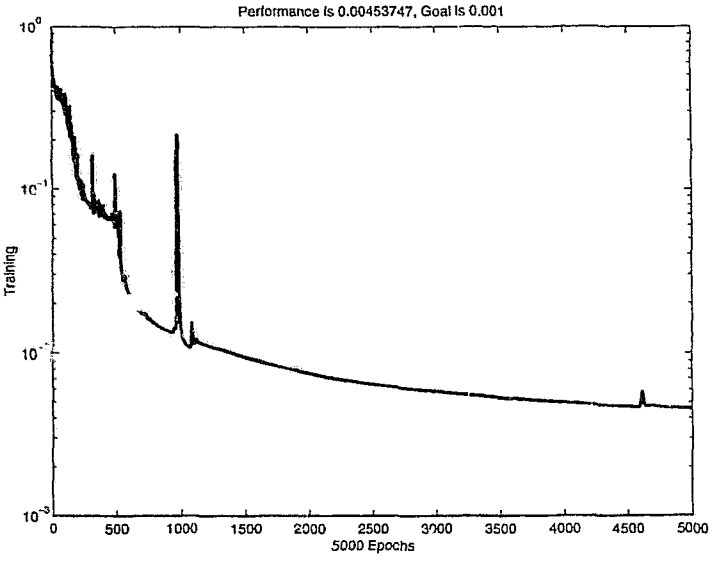


(a)

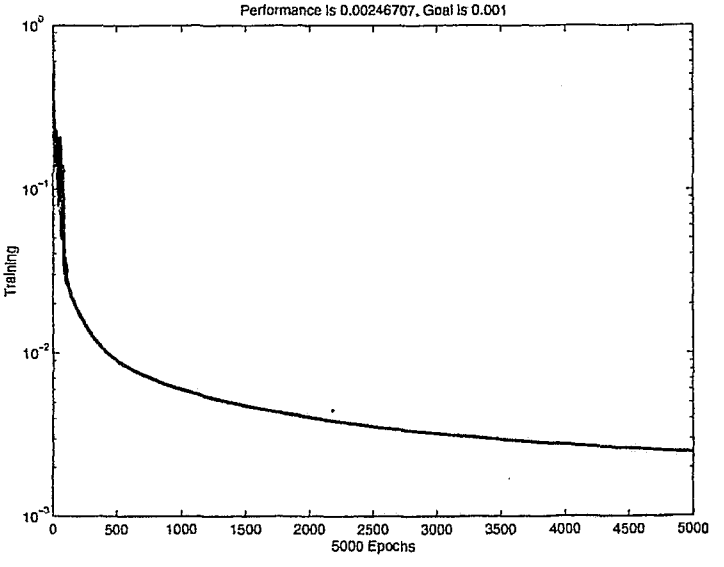


(b)

Figure 5.1: Convergence of the networks toward the performance goal equal to 0.01 calculated by the least mean square: (a) MLP with 2 sigmoid layers (b) MLP with 3 sigmoid layers

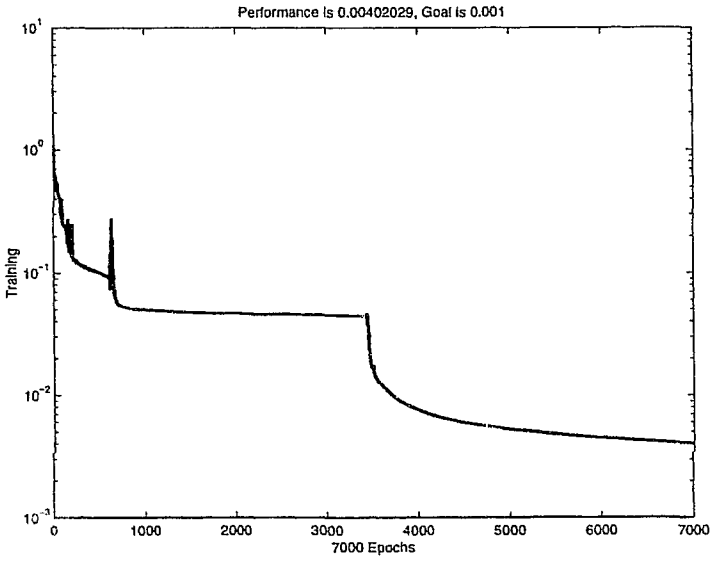


(a)

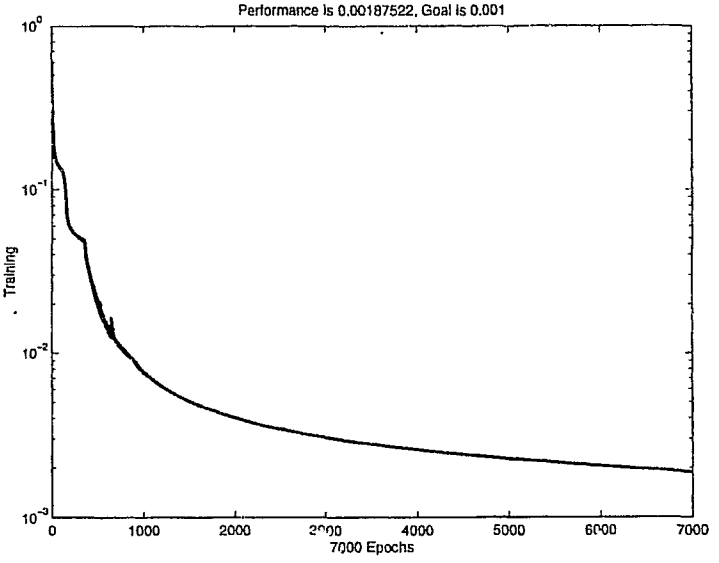


(b)

Figure 5.2: Convergence of the networks for the same number of epoches: (a) MLP with 2 sigmoid layers (b) MLP with 3 sigmoid layers



(a)



(b)

Figure 5.3: A sample of better performance goal but more classification error (Overfitting): (a) MLP with 2 sigmoid layers (b) MLP with 3 sigmoid layers

hill or valley with a relatively small number of units in the previous hidden layer [46]. For more complicated target functions, especially those with several hills or valleys, it is useful to have several units in the last hidden layer. A separate hill or valley can be approximated by each unit in the last hidden layer. A MLP with two hidden layers can often yield an accurate approximation with fewer weights than a MLP with one hidden layer and a MLP with three hidden layers can yield an approximation with fewer weights than a MLP with two hidden layers.

In the experiments, the number of neurons in the last hidden layer (or output layer) is considered as equal to the number of textures in the image. Thus MLP can learn several hills and valleys equal to the number of textures. On the other hand the number of neurons in the input layer (first hidden layer) must be equal to the dimension of extracted feature vectors and it is known. To Set the number of neurons in the first and last hidden layers or in the input and output layers, we should determine the number of neurons in the second or intermediate hidden layer. A simple role to select the number of neurons in the intermediate layers of MLPs is taking average on the number of neurons in the previous and next layers of that layer. Since proposed MLP is formed of 3 layers and the number of neurons in the input and output layers are known, the number of neurons in the intermediate hidden layer would be calculated by taking average on input and output layers. As it was mentioned before in this section, the higher the number of neurons the more complex function would be approximated. Hence, we set the number of neurons in the middle layer to different numbers in a range from half of the obtained average to two times of that number. Considering this range for the number of neurons and measuring the performance goal in training phase and classification error in the test phase the number of neurons is chosen. To get the better results and have a variety of choices a small range of numbers are proposed to select the number of neurons in the intermediate layer. In Fig. 5.4 the performance of the network according to the different number of neurons in the middle layer are depicted. As we can observe, the performance of the network in the Fig. 5.4(a) which the number of neurons is equal half the number of average of first and third layer is equal to 0.00578 and the classification error is

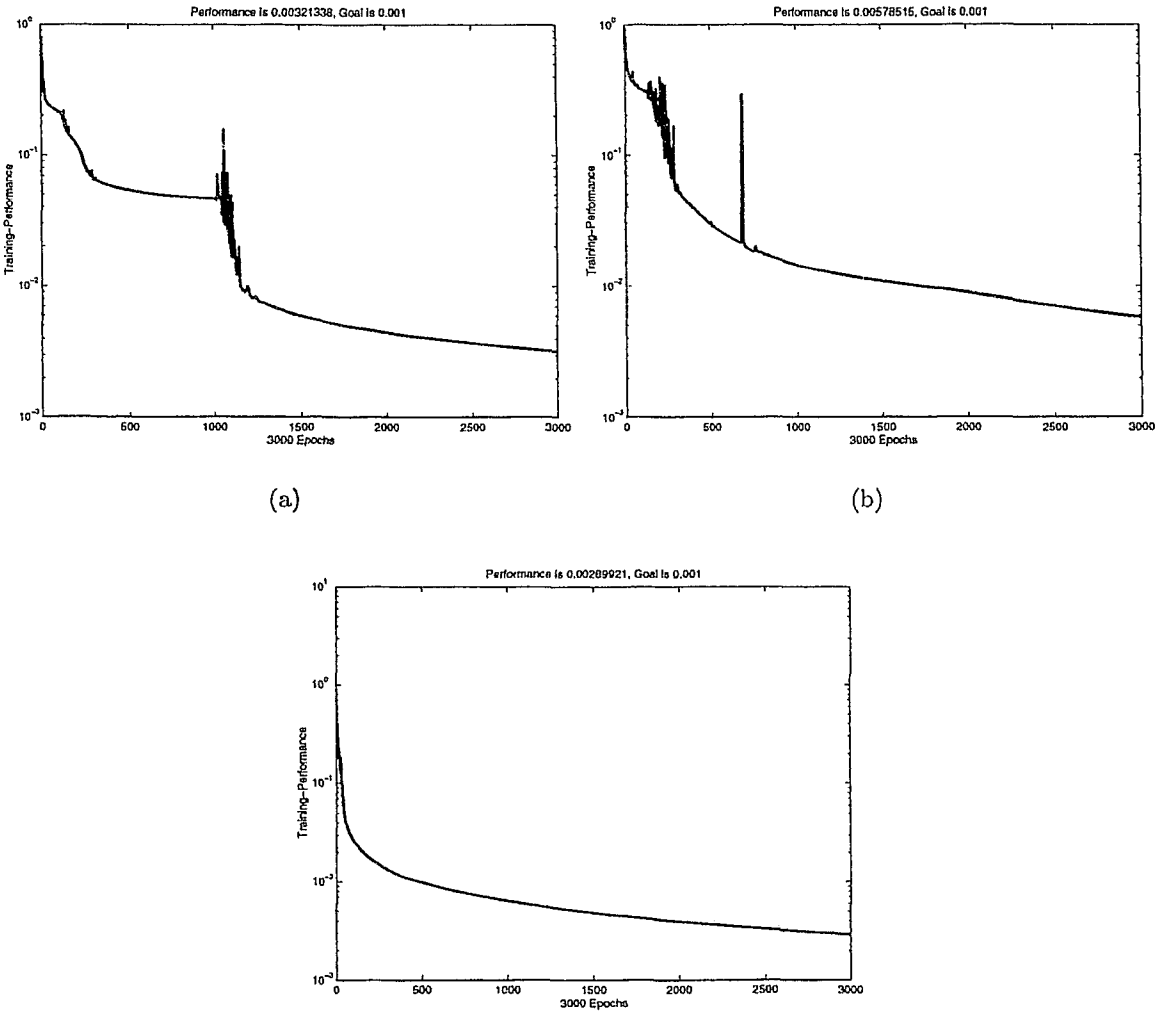


Figure 5.4: Convergence of the network toward the performance goal calculated by the least mean square: (a) MLP with 7 neurons in the intermediate layer (half the average number on first and last layer) (b) MLP with 13 neurons in the intermediate layer (The average number on first and last layer) (c) MLP with 26 neurons in the intermediate layer (Twice the average number on first and last layer)

10.68 percent. The performance goal is 0.00321 and classification error is equal to 9.973 as depicted in Fig. 5.4(b) where the number of neurons is equal to the average of neurons in the first and the third layer and the performance and classification error are equal to 0.00289 and 10.13 percent respectively for the network that the number of neurons in middle layer is twice of the average. As we can observe the best classification error belongs to the network which the number of neurons in it's middle layer is equal to the average of the first and last layer.

5.1.4 Error Surface of MLP

A backpropagation network follows the contour of an error surface because it uses a gradient descent procedure. The weight updates moving backpropagation network in the direction of steepest descent in the error surface. For two layer networks the error surface is bowl shaped and there is not a problem using gradient descent to minimize the error so that the network will always find an errorless solution at the bottom of the bowl. Such solutions are called global minima since they are errorless. However, to solve more difficult problems we must increase the number of layers. For example when an extra hidden layer is added, error surface may become more complex and may contain several minima. In such cases some minima could be deeper than the others and it is possible that a global minima wouldn't be found by gradient descent. In such cases the network fall into local minima and would represent suboptimal solutions. When we train a backpropagation network we prefer to avoid local minima and this may be difficult to do in some cases. As a general rule, The more hidden units you have in a network the less likely you are to encounter a local minimum during training. Although additional hidden units increase the complexity of the error surface, the number of possible escape routes will be increased by the extra dimensionality. Since more complex networks may cause the overfitting and increasing the error in the test phase, we prefer to keep the network small enough. Using some other techniques such as adaptive learning rate and momentum can speed up backpropagation learning and made it less sensitive to small features in the error surface such as shallow local minima .

5.1.5 The Learning Rate and Local Minima

There is no easy way of choosing a good learning rate for nonlinear multi-layer networks unlike linear networks. Choosing a too large learning rate leads to unstable learning and a learning rate that is too small results in long training times conversely. The problem is that nonlinear transfer functions in multi-layer networks introduce several local minima in the error surface. As gradient descent is performed on the error surface it is possible for the network solution to become trapped in one of these local minima. Trapping in a local minimum may be good or bad. It depends on how low an error is required and how close the local minimum is to the global minimum. Networks are also sensitive to the number of neurons in their hidden layers. Too few neurons can lead to underfitting. Too many neurons can contribute to overfitting. The selection of a learning rate is an importance issue in finding the true global minimum of the error distance. Backpropagation training with too small learning rate will make incredibly slow progress. Too large a learning rate will proceed much faster, but may simply produce oscillations between relatively poor solutions. Typical values for the learning rate parameter are numbers between zero and one:

$$0.01 < \eta < 0.75 \tag{5.1}$$

Using the largest learning rate that still converges to the minimum is preferred.

5.1.6 Momentum

To speed up the convergence in the backpropagation algorithm and avoiding possible local minima, I use a fraction of previous weight change called momentum to revise weights in the backpropagation learning. The idea is stabilizing the weight change and smooth updating by means of combining a fraction of the previous weight change with the gradient decreasing term:

$$\Delta w_{ji}(n) = \eta d_{pj} a_{pi} + \alpha \Delta w_{ji}(n-1) \tag{5.2}$$

where α is taken $0 \leq \alpha \leq 0.9$, and n is the index of the current weight change. This helps the network to avoid local minima since weight vector tends to continue moving in the same

direction unless the gradient term causes an opposite direction. The momentum has the following effects:

- The weight changes may be smoothed by momentum and oscillations across the error valley are suppressed
- If weight changes are all in the same direction, the learning rate will be amplified by the momentum and causes a faster convergence as a result
- It enables to escape from local minima on the error surface solution

For stable learning small learning rates are generally required using the gradient descent algorithm and so it will be slow. The momentum variation is usually faster than simple gradient descent, since it allows higher learning rates while maintaining stability.

5.2 Practical Aspects to Optimize Gabor Filter Bank

In this section some aspects to optimize Gabor filter bank and two implemented methods are discussed.

5.2.1 A bank of Gabor and DCT filters (GDCT)

The following attributes make DCT filters attractive to be considered as a joint filter bank for image segmentation algorithms.

1. Providing a consistent average good results for different kind of textures.
2. Fast implementation as separable filters.
3. Small size of some eight filters in the filter bank.

The mentioned attributes are the motivation toward using DCT filters as a complimentary filter bank for the bank of 20 Gabor filters with 45° angular bandwidth and following radial frequencies:

$$4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2} \tag{5.3}$$

Separable implemented DCT filters are:

$$h1 = [1, 1, 1]$$

$$h2 = [1, 0, -1]$$

$$h3 = [1, -2, 1]$$

where h1, h2 and h3 are 1D horizontal and vertical kernels. Using these 1D convolution kernels, by convolving a vertical 1D kernel with a horizontal 1D kernel we can generate 9 different 2D convolution kernels. As an example, a 2D kernel h1h2 would be obtained by convolving a vertical kernel h1 with a horizontal kernel h2. From the 1D kernels we can generate 9, 2D convolution kernels, excluding the low frequency component, 8 remaining 2D masks are obtained. A list of all 3x3 kernels is given below:

h1h1 h2h1 h3h1

h1h2 h2h2 h3h2

h1h3 h2h3 h3h3

As depicted in Fig. 5.5, both filter banks are applied to the textured image to generate a total number of 28 filtered images including 20 images generated by Gabor filter bank and the 8 others provided by DCT. The resultant feature vectors have 28 elements for each pixel of the image.

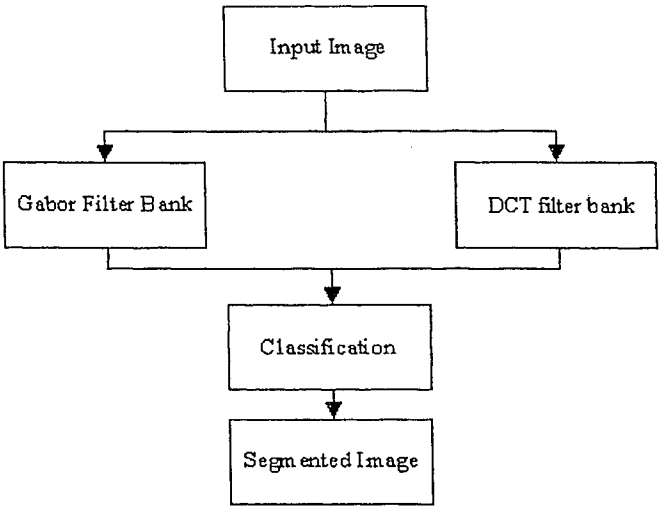


Figure 5.5: Hybrid Gabor and DCT Filter Bank

5.2.2 Adaptive selected filters from 40 filters in Gabor Filter Bank (AGFB)

In this approach a set of Gabor filters are selected as the best Gabor filters, so that each single Gabor filter in this small set of selected Gabor filters can discriminate a pair of textures in a multiple textured image. Assume that an input image I consists of T different textural classes T_c, $c = 1, 2, ..., T$ and I_c a sub-image of I, be a training sample of texture class T_c. Let us assume E_i as the energy of texture class i to select the optimal set of filters, so that the selection objective for the optimal Gabor filter set is based on the energy ratio of any two distinct textures T_i and T_j, $i \neq j$:

$$\frac{E_i}{E_j}, \quad i = 1, 2, ..., T-1 \tag{5.4}$$

$$j = i+1, i+2, ..., T$$

Therefore, the optimal filter set is selected by the following maximum principle:

$$Maximum\{\frac{E_i}{E_j}\}, \tag{5.5}$$

$$i = 1, 2, ..., T-1$$

$$j = i+1, i+2, ..., T$$

In this approach the best Gabor filters are selected so that the corresponding energy is maximum for each pair of distinct textures in a multiple textured image. For instance, considering an image composing of two different texture classes $T_i = 1$ and $T_j = 2$, the filter set includes one filter which has the maximum ratio to distinct these two textures:

$$Maximum\{\frac{E_1}{E_2}\} \tag{5.6}$$

To choose the best filter by obtaining the maximum principle, the energy ratio $\frac{E_1}{E_2}$ is calculated for all 40 Gabor filters with 22.5° angular bandwidth and the following radial frequencies:

$$4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}, 64\sqrt{2} \tag{5.7}$$

The 40 obtained energy ratios are sorted accordingly and the filter that has the maximum ratio among them is selected:

$$\{\frac{E_1}{E_2}\}_f, \quad f = 1, 2, \cdots, 40 \tag{5.8}$$

$$\{\frac{E_1}{E_2}\}_i > \{\frac{E_1}{E_2}\}_j > \cdots > \{\frac{E_1}{E_2}\}_k, \tag{5.9}$$

where,

$$i, \quad j, \quad k = 1, 2, ..., 40$$

$$i \neq j, \quad i \neq k \quad and \quad j \neq k$$

For an image containing T different textures as depicted in Fig. 5.6, the energy ratio $\frac{E_i}{E_j}$ of any two texture classes T_i and T_j, $i \neq j$, is calculated for each filter in the filter bank and sorted in decreasing order so as the maxim principle according to Equation 5.5 is obtained. The larger the value of ratio $\frac{E_i}{E_j}$, the better discrimination between any two distinct textured image. Thus the best discriminative filter for each pair of different textures according to the

maximum energy ratio is selected to be included in the subset filter bank:

$$f_{index} \in SFB \ (SubsetFilterBank),$$

$$if \quad \{\frac{E_i}{E_j}\}_{index} > \{\frac{E_i}{E_j}\}_k$$

$$\forall\, k \in 1, 2, ..., 40 \ \& \ index \neq k$$

$$for \ \ i = 1, 2, ..., T-1$$

$$j = i+1, i+2, ..., T$$

Therefore the subset filter is composed of N selected filters:

$$N = \frac{(T-1)^2 + (T-1)}{2} \tag{5.10}$$

For instance the size of subset filter for an image containing 4 different textures is:

$$T = 4, \quad N = \frac{(4-1)^2+(4-1)}{2} = 6$$

Three optimization methods are implemented and after applying the textured images, the segmentation results are obtained. Implemented filter banks are as following:

1. A bank of 42 filters composed of 40 Gabor filters with 22.5° angular bandwidth. Two low frequency and high frequency filters are added to improve the filter bank that reach the total number of filters in this bank to 42. The increase in the feature dimension using 42 filters in comparison with 20 filters in widely used Gabor filter bank is compensated by feature quantization using competitive layer.

2. A bank of 28 filters composed of 20 Gabor filters with 45° bandwidth and 8 DCT filters.

3. Adaptive Gabor filter bank which composed of adaptively selected filters from the Gabor filter bank with 22.5° angular bandwidth. The best Gabor filters are selected

for this filter subset so that the energy ratio of any two distinct textures is maximum in a multiple textured image. The number of filters in this subset is variable and depends on the number of textures in the image.

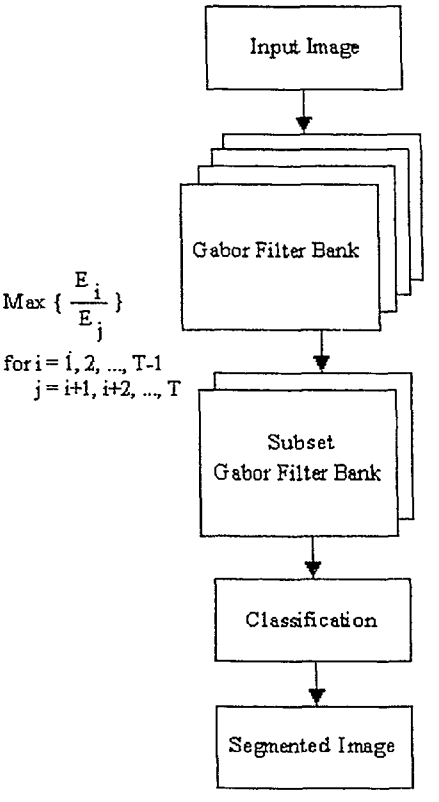


Figure 5.6: Adaptive Gabor Subset Filter

Filter parameters for the mentioned filter banks are presented in Table 5.1. The obtained results using mentioned methods will be presented and discussed in the next chapter.

Table 5.1: Filter parameters for three optimized filter banks

Filter Bank	No. of filters	Filter Bank Parameters
1	42	22.5° angular bandwidth
2	20 Gabor and 8 DCT filters	45° angular bandwidth
3	Different no. of filters according to the no. of textures in the image	22.5° angular bandwidth

Chapter 6

Results and Discussion

In this section, some results obtained by applying the proposed approach and two widely used approaches for texture segmentation, Discrete Cosine Transform (DCT) and Laws filters are presented. As it is mentioned before in Chapter 2, because of the fast implementation and good results, DCT not only is used in image compression algorithms such as JPEG compression standard but also in image segmentation algorithms such as 3 by 3 DCT that is introduced by Ng et. al. [37] to extract features from textured images. This filter bank has the same 3 filters both in vertical and horizontal directions which results in a bank of 9 filters. Meanwhile separable filter bank introduced by Laws [35] to identify different textures generates good results for texture segmentation. This filter bank is composed of five filters in each dimension which concludes a group of 25 filters in the filter bank. The results are obtained by applying the proposed method, DCT and Laws filters to two different types of images. The first set formed of synthetic textured images and the second one is IKONOS satellite images. The segmentation results are compared for both data types. The results are presented in the following sections.

6.1 Synthetic Textured Images

The first set of images formed of synthetic textured images from Brodatz album [47], MIT vision and modeling database [48] and MeasTex image texture database [49]. The combinations of different number of textures are used as test images. The sample synthetic images

consist of D17, D24, D53, D55, D77 and D84 selected from Brodatz album, Fabric.0000, Fabric.0017, Flowers.0002, Leaves.0006 and Leaves.0013 selected from [48] and Grass.0002, Misc.0002 and Rock.0005 selected from [49].

The image in Fig. 6.1(a) is a textured image consisting of Fabric.0000 from [48] and Rock.0005 selected from [49] and Fig. 6.1(b) shows the true segmentation reference image. Fig. 6.1(c) shows the segmented result by CPMLP-GFHF (Competitive layer and Multi-Layer Perceptron in conjunction with Gabor Filter bank and High Frequency component) method and the result obtained by GDCT (bank od Gabor and DCT filters) is depicted in Fig. 6.1(d). Fig. 6.1(e) and Fig. 6.1(f) are the segmented results applying AGFB (Adaptive selected Gabor Filter Bank) with 2 and 3 filters respectively. Fig. 6.1(g) and Fig. 6.1(h) show the results using DCT and Laws filters respectively. The classification error using CMLP-GFHF, GDCT and AGFB are 0.59%, 0.43% and 2.57% (having 3 filters) respectively. The improvement of results in comparison with DCT and Laws filters is presented in Table 6.1. As it can be observed in this table, CPMLP-GFHF has improved the classification results by 80% and 91% in comparison with DCT and Laws filters respectively. This amounts are 85% and 93% for GDCT and AGFB using just 3 filters improved the segmentation by 12% and 58% comparing with DCT and Laws respectively. Segmentation results obtained by CPMLP-GFHF and GDCT are better than the other results. These approaches with 0.59% and 0.43% misclassification concludes almost a perfect classification for Fig. 6.1(a). On the other hand AGFB using just 3 filters, gives better segmentation than DCT and Laws. I experinced that to get good results using AGFB when the image composed of just two textures, selecting only one filter which has the best energy ratio doesn't provide reasonable results and at least the two best ratios should be considered to have the minimum number of two filters in the filter subset. Having three filters with the three best energy ratios in AGFB subset provides better results than DCT and Laws despite the number of filters that is one third and one eighth comparing DCT and Laws respectively. Laws filters performed the worst segmentation results that have 2.4 times more classification error than AGFB and 14 times

more than GDCT.

Table 6.1: Segmentation improvement regarding Fig. 6.1

Filter Bank	In comparison with DCT	In comparison with Laws filters
CPMLP-GFHF	80%	91%
GDCT	85%	93%
AGFB	12%	58%

The image in Fig. 6.2(a) is a textured image consisting of Fabric.0000, Fabric.0017, Flowers.0002, Leaves.0006 and Leaves.0013 selected from [48]. Fig. 6.2(b) is true segmentation reference image. Fig. 6.2(c) shows the segmented result by CPMLP-GFHF method and in Fig. 6.2(d) the segmented results after applying median filter is provided. The result obtained by GDCT is depicted in Fig. 6.2(e) and Fig. 6.2(f) shows the segmented results using AGFB. In Fig. 6.2(g) and Fig. 6.2(h) the results obtained by DCT and Laws filters are depicted respectively. The classification error using CMLP-GFHF, GDCT and AGFB are 8.37%, 5.61% and 12.62%, respectively. The improvement of results in comparison with DCT and Laws filters are presented in Table 6.2. As it can be observed in this table, CPMLP-GFHF has improved the classification results by 45% and 79% in comparison with DCT and Laws filters, respectively. This amounts are 63% and 86% for GDCT and AGFB improved the segmentation by 18% and 69% comparing with DCT and Laws, respectively. Segmentation results obtained by GDCT and CPMLP-GFHF with 5.61% and 8.37% are better than the other results.

According to the results obtained in this study for synthetic images the proposed approaches provided better classification performance in comparison with DCT and Laws filter bank and reduced misclassification by 12% to 93% respectively.

Table 6.2: Segmentation improvement regarding Fig. 6.2

Filter Bank	In comparison with DCT	In comparison with Laws filters
CPMLP-GFHF	45%	79%
GDCT	63%	86%
AGFB	18%	69%

6.2 IKONOS Satellite Images

The continuous growth and expansion of urban areas is occurring on all megacities in the world like the Greater Toronto Area (GTA), Canada. The 2001 Census in Canada counted a population of 5.1 million in the GTA, an increase of 9.8% from 1996. The GTA accounted for 44.5% of Ontario's population in 2001, up from 43% in 1996. The main factor behind the population growth in the GTA was immigration to Toronto. Between the 1996 and 2001 Census, Toronto received 69% and 53% of immigrants to the GTA and Ontario, respectively (Ministry of Finance Ontario, 2003) [50]. Like other North American megacities, the GTA is expending rapidly into adjacent landscape. This change in land-use and land-cover is of concern to urban and regional planners and metropolitan government agencies, because this growth has profound impacts on the available water resources, crop land and limited remaining space. A spatio-temporal analysis of growth patterns is essential in order to develop sufficient infrastructure to support the growth. Of particular interest to urban and regional planners are the available tools and information that can be used to monitor the spatial growth of the urban areas. Conventional surveying and mapping methods cannot deliver the necessary information in a timely and cost-effective mode. With the availability of commercial very high-resolution satellite imagery such as IKONOS and QuickBird (up to 61 cm spatial resolution and 1 - 3 days temporal resolution), it is possible to obtain detailed

information on urban land use and change detection. Urban growth analyses became more sophisticated with the use of a variety of image processing techniques with varying degrees of success, such as principal component analysis [51], V-I-S analysis [52], post-classification feature extraction [53], post-classification comparison [54], integration of remote sensing and GIS [55], normalized difference built-up index [56], combined fuzzy pixel-based and object-based classification [57], texture analysis [58], spatial metrics and texture measures [59]. However, classification and segmentation of satellite images of complex urban environments is a very difficult task due to mixed pixels, spectral similarity, and the textured appearance of many land-use and land-cover types in urban areas. Conventional methods for pixel-based classification of multi-spectral satellite imagery such as parallelepiped, minimum distance from means, and maximum likelihood [1]only utilize spectral information and consequently have limited success in classifying high-resolution urban multi-spectral satellite images [60]. As many urban land-use and land-cover types, such as roads, buildings, parking lots, etc., are spectrally similar, spatial information such as texture and context must be exploited to produce more accurate classification maps [61].

The second set of images is 1 m spatial resolution pan-sharpened IKONOS satellite images of an urban residential area. The imagery used in this study covers a complex urban residential area mixing roads, paths, houses, and trees. It is located in the northwest of the GTA, Ontario, Canada, near the Pearson International Airport. The IKONOS imagery was geometrically corrected in two dimensions (X and Y) in the Universal Transverse Mercator (UTM) projection coordinate system, zone 17 in the Geodetic Reference System of 1980 (GRS80) and North American Horizontal Datum of 1983 (NAD83). The roads in the image are continuous regions which form a network. Houses are very dense in the image, some roofs of which have similar spectral characteristics as roads. Trees along the road are located close to houses. The results are obtained by applying the proposed CMLP-GFHF method, DCT and Laws filters to different sets of IKONOS satellite images. The segmentation results obtained using the proposed CMLP-GFHF method and the comparison with the DCT and Laws methods are shown and explained as following.

The images in Fig. 6.3(a) and Fig. 6.4(a) are sub-images of a large IKONOS satellite image of an urban residential area. The sub-images cover a complex urban residential area mixing roads, houses, and trees. The segmentation results obtained by applying CPMLP-GFHF, DCT and Laws to Fig. 6.3(a) are depicted in Fig. 6.3(b), Fig. 6.3(c) and Fig. 6.3(d) respectively. Another IKONOS sub-image of an urban residential area near international pearson airport is segmented by proposed method. Fig. 6.4(b) shows the segmentation results obtained by applying CPMLP-GFHF to Fig. 6.4(a). The segmentation results obtained by DCT and Laws are depicted in Fig. 6.4(c) and Fig. 6.4(d) respectively. As it can be visually observed, the segmentation results of CPMLP-GFHF is much better than DCT and Laws filter bank. CMLP-GFHF method has shown superior results for dealing with the complex urban residential situations. For example, vegetation (e.g., trees) and non-vegetation (e.g., buildings, roads, etc.) are clearly classified in the segmented images obtained using the CMLP-GFHF method.

6.2.1 Refinement the Results

To improve the segmentation results obtained by applying the proposed method to the satellite imagery, the results are refined according to a refinement method that is presented. The proposed refinement method is a greedy point method with some attributes inherited from region growing and split and merge methods. Boundary based segmentation is based on finding the boundaries between regions which are based on discontinuities in gray level or using thresholds based on the distribution of pixel properties. Conversely region based segmentation methods are based on directly finding regions to achieve the goal that is partitioning the image to the regions.

Greedy Point Method According to Region Growing and Split & Merge

If the entire image is represented by region G, region growing method partitions G into R subregions, $G_1, G_2, ..., G_R$, such that:

- Pixels in a region must be connected: G_j is a connected region for $j = 1, 2, ..., R$

- The pixels in segmented region must satisfy some properties: When $P(G_j)$ is a logical predicate defined over the points in set G_j, $P(G_j) = True$ for $j = 1, 2, ..., R$
- According to the predicate P regions are different: $P(G_j \cup G_k) = False,\ j \neq k$
- Each pixel must belong to a region and segmentation must be complete: $\bigcup_{j=1}^{R} G_j = G$
- The region must be disjoint: $G_j \cup G_k = \emptyset,\ j \neq k$

Region growing based on some predefined criteria groups pixel or sub regions larger. With a set of seed points the approach is started and grow region by appending those neighboring pixels to each seed that have properties similar to the seed. For the gray level images region analysis could be performed based on gray levels and spatial properties such as texture or moments. If the assumption of a model of expected results is available, some additional criteria would improve a region growing algorithm based on the available assumptions which may utilize the concept of the shape, size, likeness between a candidate pixel and the pixels grown in the region so far, such as a comparison of the average gray level of the grown region and the gray level of the candidate pixel.

On the other hand, split and merge algorithm initially subdivide an image to disjoint regions and then merge the regions if some properties are satisfies by the adjacent regions. This method iteratively split the image to G_j distinct partitions and splitting will be continued for any region that $P(G_j) = FALSE$. When the property $P(G_j) = TRUE$ is satisfied by the pixels in the region, the process stops splitting that region. Merge process starts as soon as all regions satisfy the property $P(G_j) = TRUE$. All adjacent regions G_j and G_k iteratively will be merged if $P(G_j \cup G_k) = TRUE$. When no further merging or splitting is possible the algorithm will stop.

Since the image is segmented, there are a few number of gray tones according to the number of classes so that pixels with the same gray level are in the same class. In the first

step of this greedy point method all of the pixels belong to the same class (possibly located in different regions) in the segmented image are recognized and registered to the same clusters. Then boundary pixels will be adjusted to the neighboring cluster of this pixel according to the inter cluster distance criteria. The inter class distance of clusters is defined as:

$$Distance = \sum_{f=1}^{F} (\mu_{(G_j,f)} - \mu_{(G_k,f)})^2 \qquad (6.1)$$
$$For j, k = 1, 2, ..., C,\ j \neq k$$

where, F is the number of the filters of the Gabor filter bank, C is the number of classes, G_j and G_k are the j^{th} and k^{th} classes respectively and $\mu_{(G_j,f)}$ is mean of class j over the output of f^{th} filter of the Gabor filter bank. If by adjusting the boundary pixel to the neighbor class of this pixel the inter class distance increase due to this adjustment, the pixel will be adjusted to the neighbor class. Otherwise the boundary pixel remains in the same class and will not be adjusted that means by adjusting this pixel to the neighbor class the inter class distance will decrease. The process will continue by following on from left to right and top to bottom of the segmented image pixel by pixel. This method is iterative and will be ended after some iterations.

There are some regions that are very small and may not be considered as a region in the segmented image. Thus next step of the refinement algorithm starts by recognizing all distinct regions of the segmented result. Since the image is segmented, there are some limited number of gray tones according to the number of classes. The homogeneity criteria in this step is gray level of the pixels in the segmented result. After all adjacent pixels that satisfy the mentioned property are grouped to the proper regions, the regions among them which have very small sizes are recognized and will be merged with the adjacent regions. After all distinct regions satisfy the minimum acceptable size the merging process stops. The refinement algorithm can be abstracted as:

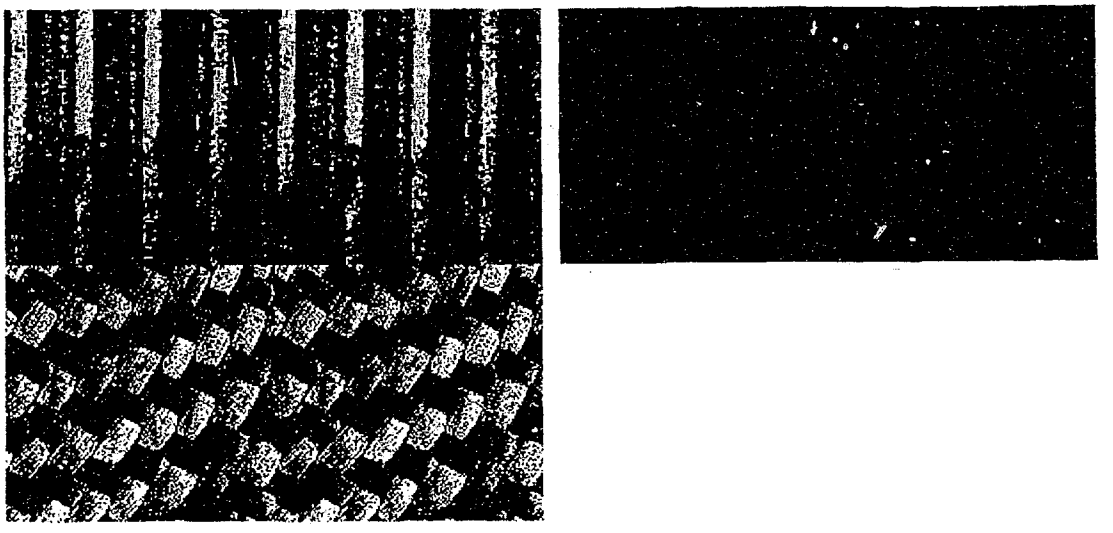
- Boundary Adjustment

  - Row Adjustment: Consider the pixel and its neighbor in the same row for adjustment according to the maximum distance.
  - Column Adjustment: Consider the pixel and its neighbor in the same column for adjustment according to the maximum distance.

- Merge: Small regions are merged with the neighboring class to erase very small and meaningless regions and guarantee that regions have a minimum acceptable size.

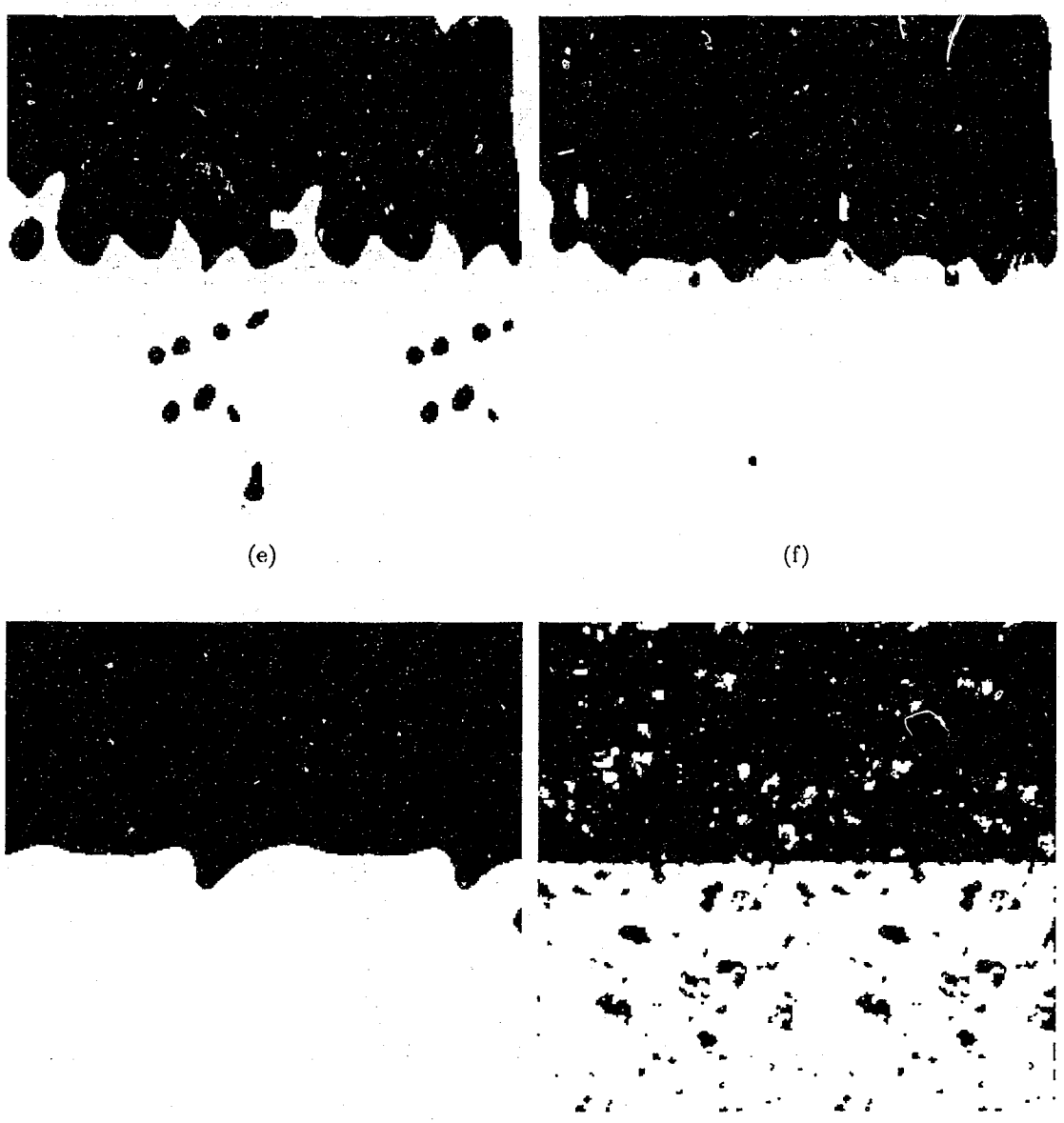
The refinement algorithm is applied to the segmentation results obtained by CPMLP-GFHF. Fig. 6.3(e) shows the result that is provided by refining the Fig. 6.3(b). The segmented result of Fig. 6.4(b) after refinement is depicted in Fig. 6.4(e).

The proposed method provided better results both for synthetic texture dataset images and IKONOS satellite images. In case of synthetic images this method reduced the classification error by 12% to 93% comparing the results obtained by DCT and Laws, respectively. Applying CPMLP-GFHF to the satellite images, it can be visually observed that the segmentation results generated by this method are much better than DCT and Laws. To compare the results statistically, IKONOS sub-images are manually segmented and by counting false-positive and true-negative pixels in each class and sum them up, segmentation results are compared with DCT and Laws filters. The misclassification is reduced by proposed method by 43% to 87% in comparison with DCT and Laws filters, respectively.



(a) (b)

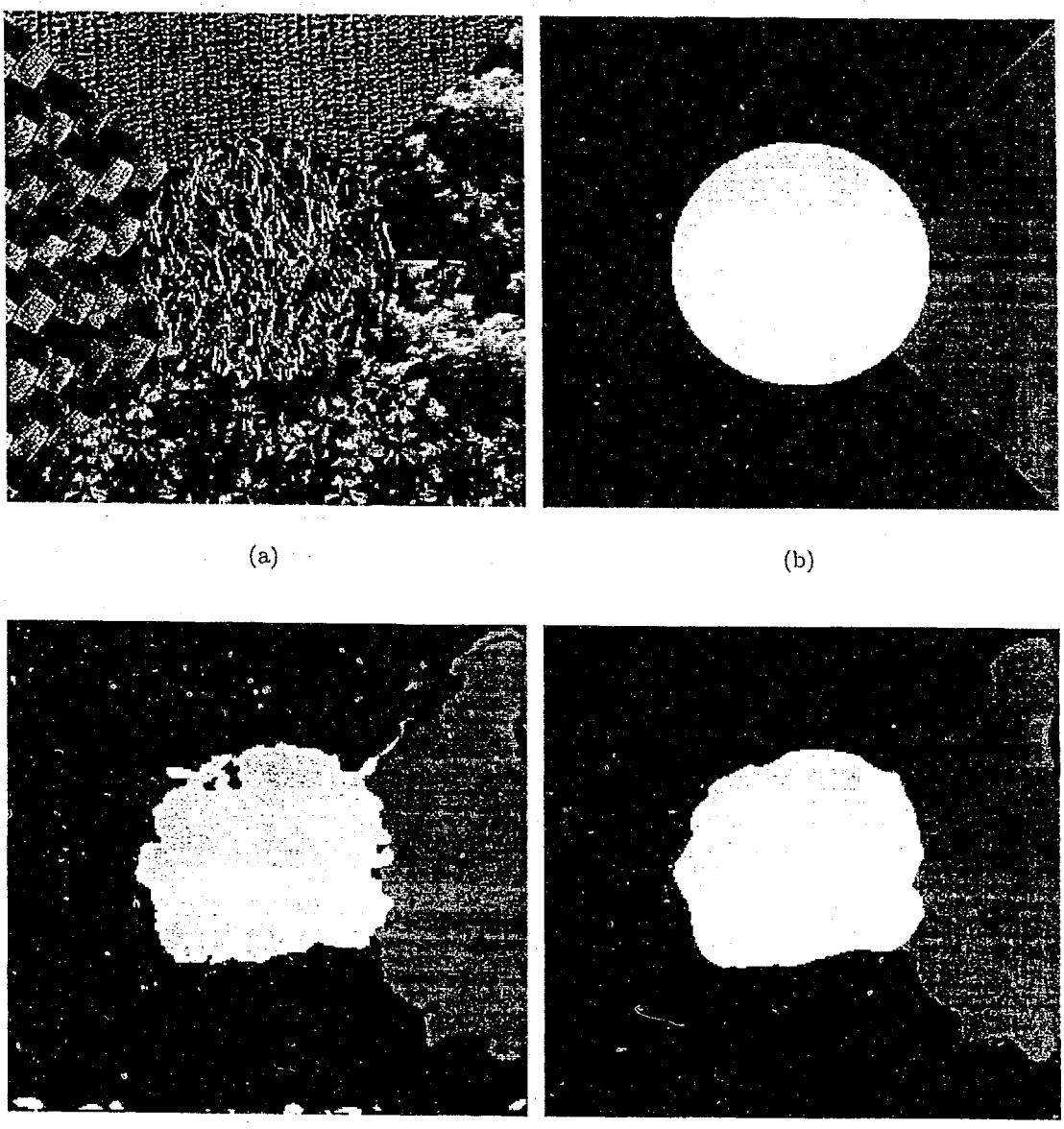
(c) (d)



(e) (f)

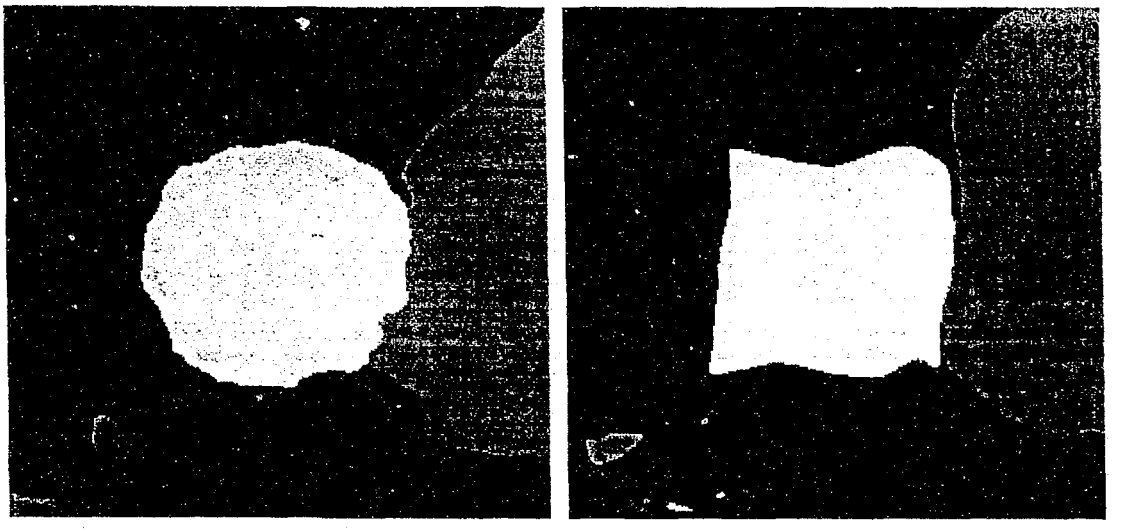
(g) (h)

Figure 6.1: (a) Textured image consisting of Fabric.0000 and Rock.0005 (b) True segmentation reference image (c) Segmented result by CPMLP-GFHF (d) Segmented result by GDCT (e) Segmented results applying AGFB (2 filters) (f) Segmented results by AGFB (3 filters) (g) Segmentation result using DCT (h) Segmentation obtained by Laws filters

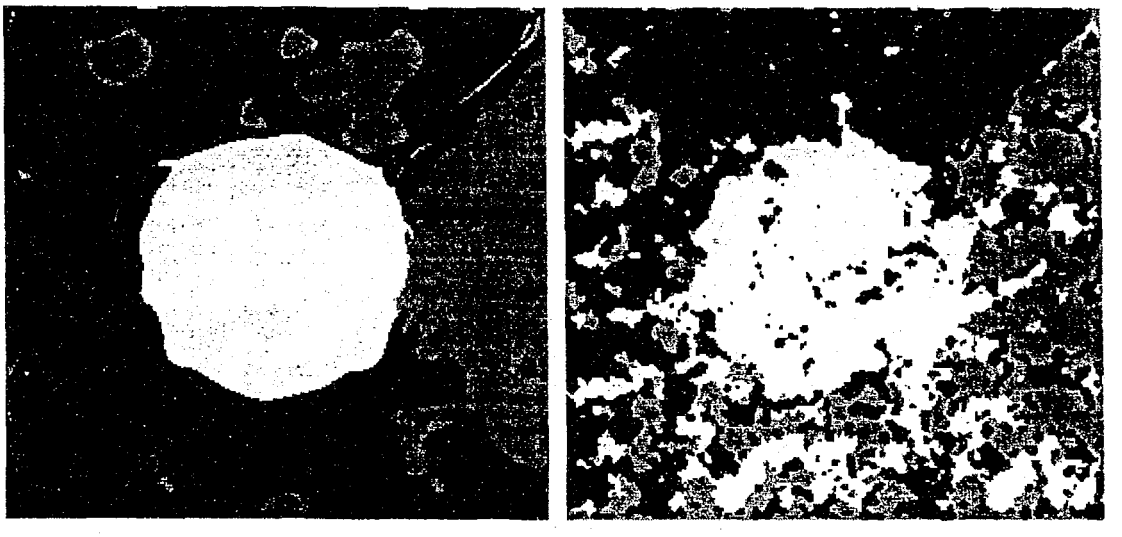


(a) (b)

(c) (d)

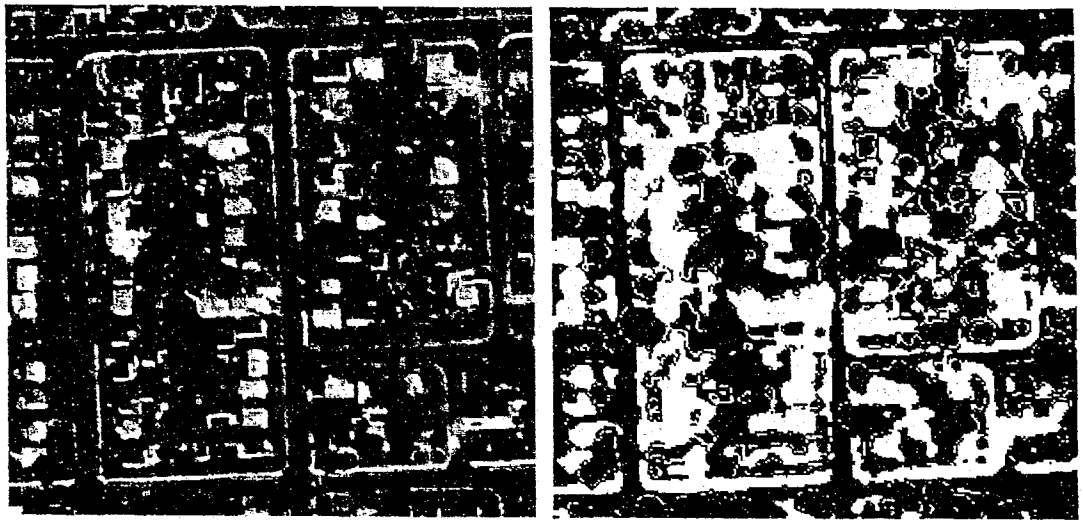


(e) (f)

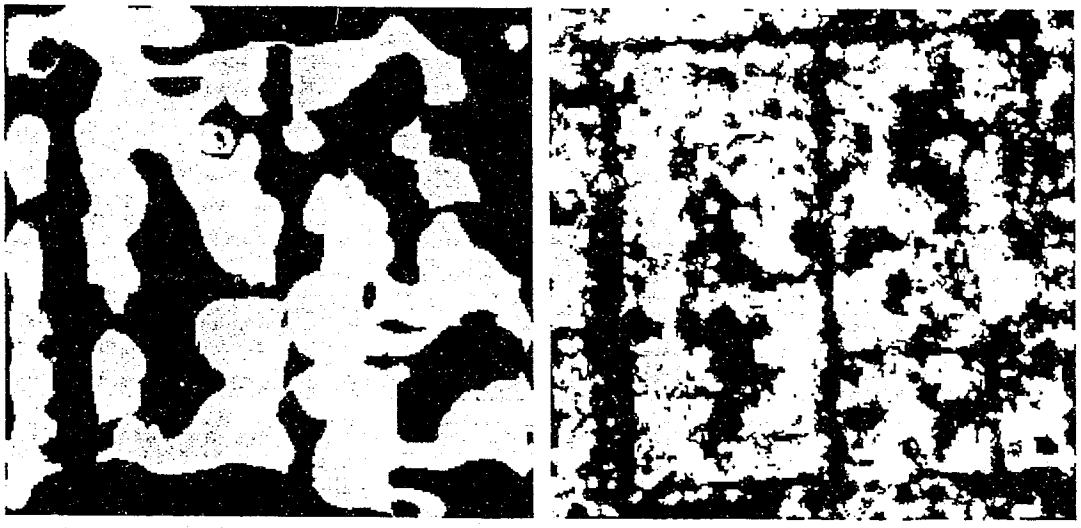


(g) (h)

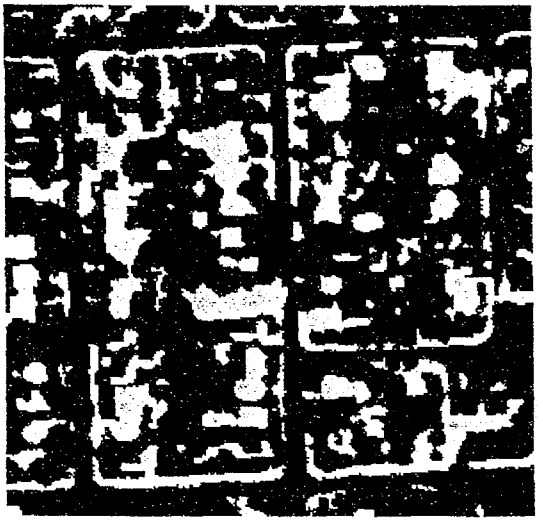
Figure 6.2: (a) Textured image consisting of Fabric.0000, Fabric.0017, Flowers.0002, Leaves.0006 and Leaves.0013 (b) True segmentation reference image (c) Segmented result by CPMLP-GFHF (d) Segmented result by CPMLP-GFHF after applying median filter (e) Segmented result by GDCT (f) Segmented results applying AGFB (g) Segmentation result using DCT (h) Segmentation obtained by Laws filters



(a) (b)

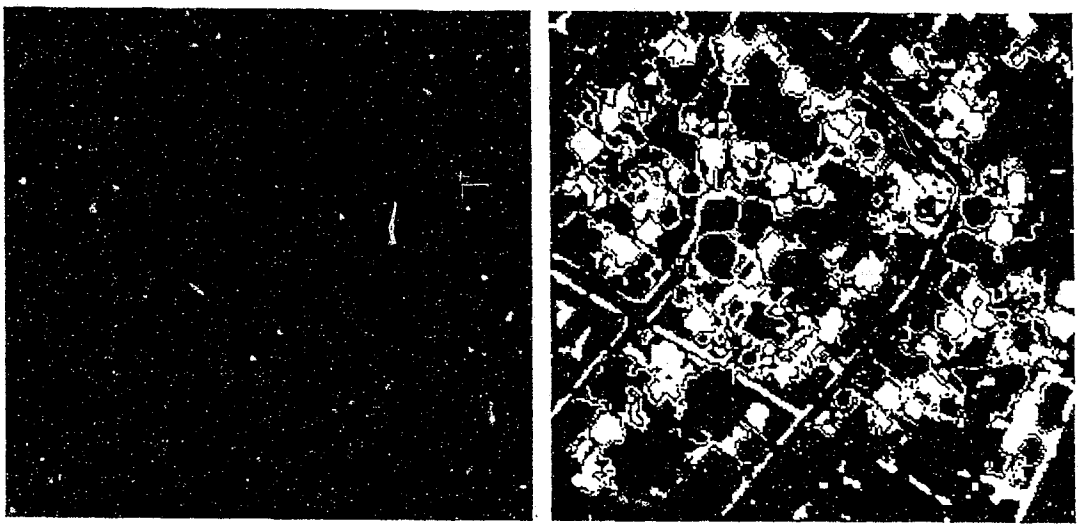


(c) (d)

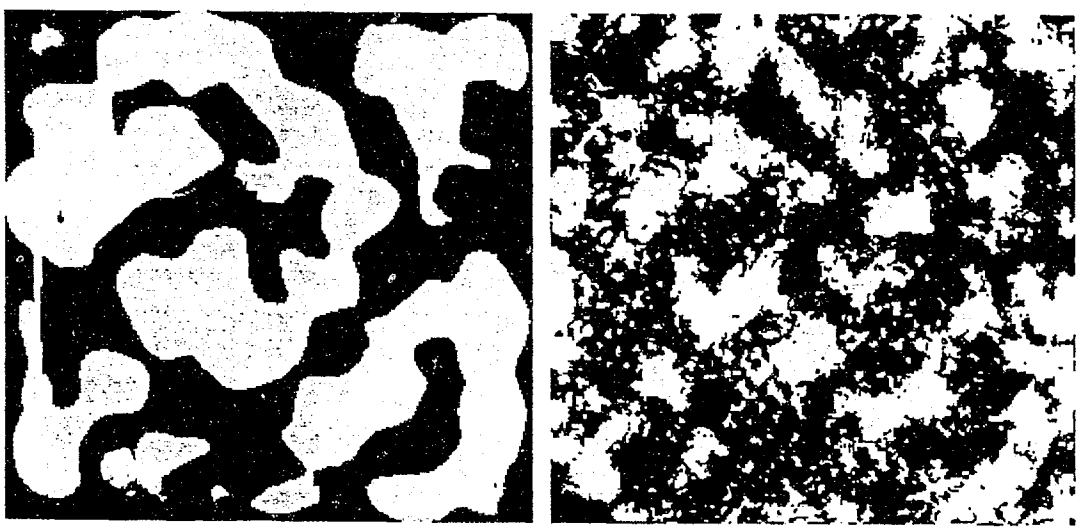


(e)

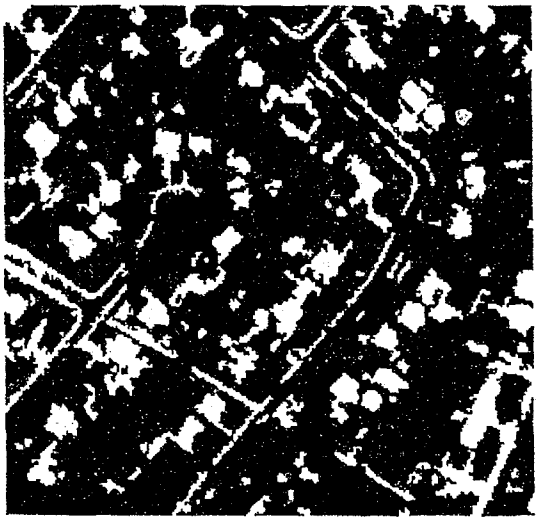
Figure 6.3: (a) Sub-image of IKONOS satellite imagery (b) Segmentation results by CPMLP-GFHF (c) Segmentation results using DCT (d) Segmentation results obtained by Laws filters (e) Segmentation results using CPMLP-GFHF after refinement



(a) (b)



(c) (d)



(e)

Figure 6.4: (a) Sub-image of IKONOS satellite imagery (b) Segmentation results obtained by CPMLP-GFHF (c) Segmentation results using DCT (d) Segmentation results by applying Laws filters (e) Segmentation results by CPMLP-GFHF after refinement

Chapter 7

Conclusion and Future Work

Discoveries about Human Visual System (HVS) have proved that electrical impulses are passed from receptor cells that are located on the back inner surface of the eyes to the visual cortex placed in the brain. To reach the visual cortex they are passed through various nervous pathways and the optic nerve. There is a considerable compression in the number of nerve fibers when this visual information is transferred from the eye to the optic nerves. Data compression and redundant information removal lead to this fact that HVS performs some kind of coding before data be received by the visual cortex.

These discoveries about the mysteries of biological vision show that HVS is a complex system composed of different kinds of visual cells and very sophisticated various nervous pathways that transfer information to optic nerves and visual cortex in the brain. This was the motivation toward implementing a very abbreviate imitation of HVS. To mimic this system Gabor filter bank is used as the first stage for feature extraction.

Extracted features by simple cells in visual cortex are condensed, redundant information are removed and compressed information are sent by human nervous system to human brain for segmentation and recognition. Neural Networks are used to remove redundant information and segment the final extracted features because of their attributes which bear a strong resemblance with general functions that an abbreviated simple simulation of human visual system and human nervous system must performs.

7.1 Conclusion

In this thesis, a method for segmenting textured images using Gabor filters has been presented. Considering the attributes of HVS, a hybrid algorithm was implemented using multi-channel decomposition by Gabor filter bank for feature extraction in conjunction with Artificial Neural Networks for both feature reduction and texture segmentation. Three approaches were implemented to optimize Gabor filter bank for image segmentation as follows:

- First, A bank of 42 narrow band Gabor filters composed of 40 Gabor filters, low frequency and high frequency components. Gabor filters have 22.5° angular bandwidth and $k \times 22.5°$ orientations when k changes from 0 to 7 and five radial frequencies having one octave bandwidth. The low frequency component is obtained by a Gaussian low pass filter and the high frequency component is computed by using the Gabor filter bank and Gaussian low pass filter. The feature vectors have 42 elements for each pixel of the image. This approach is named CPMLP-GFHF as: Competitive layer and Multi-Layer Perceptron (CMLP) in conjunction with Gabor Filter bank and High Frequency component (GFHF)

- Second, A bank of 28 filters composed of 20 Gabor filters with 45° bandwidth, $k \times 45°$ orientations when k changes from 0 to 3 and five radial frequencies, and 8 DCT filters. Both Gabor filter bank and DCT are applied to the textured image to generate a total number of 28 filtered images including 20 images generated by Gabor filter bank and the 8 others provided by DCT. The resultant feature vectors have 28 elements for each pixel of the image. This approach is named GDCT as: a joint Gabor filter bank and DCT

- Third, Adaptive Gabor filter bank which composed of adaptively selected filters from the Gabor filter bank with 22.5° angular bandwidth. In this approach the best Gabor filters are selected so that the energy ratio is maximum for each pair of distinct textures in a multiple textured image. The number of elements in feature vectors differs

according to the number of textures in the image. This algorithm is referred AGFB as: Adaptive Gabor Filter Bank

PCA was used to reduce the feature dimension of the first approach. The important practical application of PCA is providing an effective method to reduce the dimension of input data space. By eliminating those linear combinations that have small variances the number of necessary features for effective data representation could be reduced. To imitate HVS, a competitive network was implemented for performing PCA to reduce the dimension of extracted features. Competitive layer is trained by applying the training samples that are selected randomly from different regions of the image.

Eventually MLP will accomplish the segmentation by learning the nonlinear relationship between quantized vectors and desired outputs. After MLP is trained, quantized feature vectors corresponds to each pixel of the image are segmented to proper regions. The learning method is backpropagation and mean square error is employed as the performance of the network.

The segmentation problem is formulated as a combinational optimum problem to obtain a better filter bank, to reduce feature dimension and to segment with a better classifier. The main advantages of the method include:

- It enables the use of larger filter banks consisting of a higher number of channels.
- It is capable to select better discriminative features.
- Capability of extract and combine the information from multiple banks of filters.
- The feature quantization and classification component of this algorithm is a generic approach with the ability to learn and process different kinds of input feature vectors to quantize and segment.
- This algorithm is capable of segmenting complex textured images.

By three mentioned approaches Gabor filter bank is optimized either by achieving more accurate results and reducing the classification error or by reducing the filter bank size and

as a result feature dimension reduction. CPMLP-GFHF and GDCT produces better results for both synthetic and satellite images than DCT and Laws filters and outperforms them. On the other hand AGFB, despite noticeable feature dimension reduction provides comparable results. Although for synthetic images the segmented results of DCT are acceptable, in the case of satellite images the classification error is very high. On the other hand Laws filter bank is more consistent than DCT to segment both synthetic and satellite images and it better performs to segment images.

7.2 Future Work

Since there are several unknowns regarding Gabor filter implementation which must be determined accurately to extract the features precisely for texture discrimination, designing a better filter bank, more accurate selection from a wide range of values for each parameter and combination of these values have a crucial importance and may improve the accuracy of this algorithm.

Another active area for future research is minimizing the number of filters in the Gabor filter bank which is a very important and difficult challenging goal.

Using a better learning method for neural networks may also improve the accuracy of this algorithm. The algorithm could also be extended to problems of segmenting the biomedical images.

It should be pointed out that although using proposed method for segmentation of satellite images good results are obtained, there are several overlapped classes in IKONOS satellite images of urban areas such as the roof of the houses and the roads which cause negative effect in the accuracy level of the classification process. The overlapping areas have the same textual signatures and are one of the main problems in segmentation. Hence, considering some excessive features (such as colour information) may improve the segmentation results of these overlapped areas.

Future research also could be conducted on the application of the proposed technique for segmentation of the other varieties of satellite images and SAR texture images.

On the other hand, this area is a broad research area and is still developing rapidly. By improving our understanding about the HVS, more accurate artificial vision systems could be implemented to mimic HVS and be used as comparison basis to measure the performance of machine vision models.

List of Acronyms

MLP Multi-Layer Perceptron

CN Competitive Network

CMLP Competitive layer in conjunction with Multi-Layer Perceptron

GFHF Gabor Filter bank in conjunction with High Frequency component

DCT Discrete Cosine Transform

GDCT Gabor filter bank in conjunction with Discrete Cosine Transform

AGFB Adaptive Gabor Filter Bank

MSE Mean Square Error

PCA Principal Component Analysis

Bibliography

[1] T. M. Lillesand, R. W. Kiefer, and J. W. Chipman, *Remote Sensing and Image Interpretation*, John Wiley and Sons, New York, 2003.

[2] B. G. Savitsky and T. E. L. Jr., *GIS methodologies for developing conservation strategies*, Columbia University Press, New York, 1998.

[3] D. F. Maune, *Digital elevation model technologies and applications*, Springer-Verlag, Berlin, 2001.

[4] B. Julesz, "Visual pattern discrimination," *IRE Trans. on Information Theory* **8**, pp. 84–92, 1962.

[5] A. C. Bovik, M. Clark, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(1), pp. 55–73, 1990.

[6] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition* **91**, pp. 1167–1186, 1991.

[7] D. Dunn and W. Higgins, "Optimal gabor filters for texture segmentation," *IEEE Transactions on Image Processing* **4**, pp. 947–964, 1995.

[8] C. Soanes, *Oxford Dictionary of English*, Hardcover, New York, 2003.

[9] Merriam-Webster, *Merriam-Webster's Collegiate Dictionary*, Hardcover, London, 2003.

[10] J. M. Coggins, *A Framework for Texture Analysis on Spatial Filtering*, Ph.D. Thesis, Computer Science Department, Michigan State University, East Lansing, Michigan, 1982.

[11] T. Randen and H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**, pp. 291–310, 1999.

[12] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, 1989.

[13] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE* **67**, pp. 786–804, 1979.

[14] B. Petzold, P. Reiss, and W. Stssel, "Derivation of digital terrain models," *Journal of Photogrammetry and Remote Sensing (PE&RS)* **54**, pp. 95–104, 1999.

[15] S. K. S. J. Lee, R. C. Weger and R. M. Welch, "A neural network approach to cloud classification," *IEEE Trans. Geoscience and Remote Sensing* **28**, pp. 846 – 855, 1990.

[16] M. K. Hazarike, L. Samarakoon, and H. Kiyoshi, "Application of remote sensing and g.i.s for road (asian highway) network database creation and risk assessment," *Asian Center for Remote Sensing (ACRoRS)* , 1998.

[17] B. Nicolin and D. Gabler, "A knowledge-based system for the analysis of aerial images," *IEEE Trans. on Geoscience Remote Sensing* **25**, pp. 317–328, 1987.

[18] D. S. Robinson, F. Manns, J. M. Parel, P. Milne, N. Salas, A. Minhaj, and D. B. David, "A clinical approach to the development of interstitial therapy of primary breast cancer," *Lasers and Electro-Optics Society 2000 Annual Meeting* **1**, pp. 210 –211, 2000.

[19] J. K. Kim and H. W. Park, "Statistical textural features for detection of microcalcifications in digitized mammograms," *IEEE Transactions on Medical Imaging* **18**, pp. 231 –238, 1999.

[20] B. V. [illegible]ken, B. M. T. Haar-Romeny, and M. A. Viergever, "Computer-aided diagnosis [illegible]hest radiography: a survey," *IEEE Transactions on Medical Imaging* **20**, pp. 1228 –1241, 2001.

[21] A. R. Rao and B. G. Schunck, "Computing oriented texture fields," *CVGIP: Graphical Models and Image Processing* **53**, pp. 157–185, 1991.

[22] C. Fernandez, S. Fernandez, P. Campoy, and R. Aracil, "On-line texture analysis for flat products inspection," *20th International Conference on Industrial Electronics, Control and Instrumentation* **2**, pp. 867–872, 1994.

[23] F. P. Lovergine, A. Branca, G. Attolico, and A. Distante, "Leather inspection by oriented texture analysis with a morphological approach," *International Conference on Image Processing* **2**, pp. 669–671, 1997.

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

[25] F. W. Campbell and J. J. Kulikowski, "Orientational selectivity of the human visual system," *Journal of Physiology* **187**, pp. 437–445, 1966.

[26] D. A. Pollen and S. F. Ronner, "Visual cortical neurons as localized spatial frequency filters," *IEEE Transactions on Systems, Man, Cybernetics* **13**, pp. 907–916, 1983.

[27] S. G. Mallat, "Multifrequency channel decomposition of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing* **37**, pp. 2091–2110, 1989.

[28] F. W. Campbell and J. G. Robson, "Application of fourier analysis to the visibility of gratings.," *Journal of Physiology* **197**, pp. 551–566, 1968.

[29] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortex filters," *J. Opt. Society* **2**, pp. 1160–1169, 1985.

[30] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Transactions on Signal Processing* **39**(9), pp. 2020–2043, 1991.

[31] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Gabor filter design for multiple texture segmentation," *SPIE Journal on Optical Engineering* **35**, pp. 1–9, 1996.

[32] M. Unser and M. Eden, "Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering," *IEEE Transactions on Systems, Man, Cybernetics* **20**, pp. 804–815, 1990.

[33] D. A. Clausi and M. Jernigan, "Designing gabor filters for optimal texture separability," *The Journal of Pattern Recognition Society* **33**, pp. 1835–1849, 2000.

[34] A. K. Jain and K. Karu, "Learning texture discrimination masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**, pp. 195–205, 1996.

[35] K. I. Laws, "Rapid texture identification," *Proc. SPIE Conf. on Image Processing for Missile Guidance* , pp. 376–380, 1980.

[36] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Education, India, 2002.

[37] I. Ng, T. Tan, and J. Kittler, "On local linear transform and gabor filter representation of texture," *Proc. Int'l Conf. on Pattern Recognition* , pp. 627–631, 1992.

[38] J. Strand and T. Taxt, "Local frequency features for texture classification," *Pattern Recognition* **27**, pp. 1397–1406, 1994.

[39] J. Beck, "Textural segmentation, second order statistics and textural elements," *Biological Cybernetics* **48**, pp. 125–130, 1983.

[40] J. A. Freeman and D. Skapura, *Neural Networks Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Massachusetts, 1992.

[41] S. Haykin, *Neural Networks a comprehensive foundation*, Prentice Hall, London, 1999.

[42] J. Dayhoff, *Neural Networks Architectures*, Van Nostrand Reinhold, Canada, 1990.

[43] T. Kohonen, T. S. Huang, and M. R. Schroder, *Self Organizing Maps*, Springer-Verlag, Berlin, 1992.

[44] T. Hsu, H. Liu, S. Tzeng, J. Yang, and C. Lee, "A mlp/bp-based equalizer for nrz signal recovery in band-limited channels," *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems* **3**, pp. 8–11, 2000.

[45] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, Chapman and Hall, London, 1989.

[46] D. L. Chester, "Why two hidden layers are better than one," *Int. Joint Conf. on Neural Networks* **1**, pp. 265–268, 1990.

[47] P. Brodatz, *Photographic Album*, Textures: A Photographic Album for Artists and Designers, 1966.

[48] M. vision and modeling group, *Texture Photography Album*, URL: http://www.media.mit.edu/vismod/, 1998.

[49] MeasTex, *MeasTex Image Texture Database and Test Suite*, URL: http://www.cssip.uq.edu.au/staff/meastex/meastex.html, 1997.

[50] M. of Finance Ontario, "Census: Population growth in ontario's cmas and the gta, url: http://www.gov.on.ca/fin/english/demographics/dtr0204e.htm," 2003.

[51] X. Li and A. G. O. Yeh, "Principal component analysis of stacked multitemporal images for the monitoring of rapid urban expansion in the pearl river delta," *International Journal of Remote Sensing* **19**, pp. 1501–1518, 1998.

[52] B. B. Madhaven, S. Kubo, N. Kurisaki, and T. Sivakunmar, "Appraising the anatomy and spatial growth of the bangkok metropolitan area using a vegetation-impervious-soil

model through remote sensing," *International Journal of Remote Sensing* **22**, pp. 789–806, 2001.

[53] Y. Zhang, "Detection of urban housing development by fusing multisensor satellite data and performing spatial feature post-classification," *International Journal of Remote Sensing* **22**, pp. 3339–3355, 2001.

[54] C. Y. Ji, Q. Liu, D. Sun, S. Wang, P. Lin, and X. Li, "Monitoring urban expansion with remote sensing in china," *International Journal of Remote Sensing* **22**, pp. 1441–1455, 2001.

[55] Q. Weng, "A remote sensing-gis evaluation of urban expansion and its impact on surface temperature in the zhujiang delta, china," *International Journal of Remote Sensing* **22**, pp. 1999–2014, 2001.

[56] Y. Zha, J. Gao, and S. Ni, "Use of normalized difference built-up index in automatically mapping urban areas from tm imagery," *International Journal of Remote Sensing* **24**, pp. 583–595, 2003.

[57] A. K. Shackelford and C. H. Davis, "A hierarchical fuzzy classification approach for high-resolution multispectral data over urban areas," *IEEE Transactions on Geoscience and Remote Sensing* **41**, pp. 1920–1932, 2003.

[58] R. Gluch, "Urban growth detection using texture analysis on merged landsat tm and spot-p data," *Photogrammetric Engineering and Remote Sensing* **68**, pp. 1283–1288, 2002.

[59] M. Herold, X. Liu, and K. C. Clarke, "Spatial metrics and image texture for mapping urban land use," *Photogrammetric Engineering and Remote Sensing* **69**, pp. 991–1001, 2003.

[60] C. H. Davis and X. Wang, "Urban land cover classification from high-resolution multi-spectral ikonos imagery," *Proc. IGARSS* **2**, pp. 1204–1206, 2002.

[61] A. K. Shackelford and C. H. Davis, "A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas," *IEEE Transactions on Geoscience and Remote Sensing* **41**, pp. 2354–2363, 2003.

Publications

Journal Papers

1- A HYBRID ALGORITHM FOR TEXTURE SEGMENTATION OF IKONOS IMAGERY OF AN URBAN AREA
Nezamoddin N. Kachouie, Javad Alirezaie and Jonathan Li
GEOMATICA, Special edition, Remote sensing of urban areas
(Accepted, In press for April 2004 Issue)

Conference Papers

1- Genetically derived fuzzy c-means algorithm for segmentation
Nezamoddin N. Kachouie, Javad Alirezaie and Kaamran Raahemifar
IEEE CCECE 2003, Montreal (Published)
Electrical and Computer Engineering, 2003, Volume: 2, May 4-7, 2003 Pages: 1119 -1122

2- Segmentation of IKONOS Images by Texture Information and Genetically Derived Hard and fuzzy c-means Algorithm
Nezamoddin N. Kachouie and Javad Alirezaie
IASTED-SIP, Hawaii, USA, Aug 13-15, (Published)
Signal and Image Processing Proceeding of the 5th IASTED International Conference Pages: 461-466

3- Texture Segmentation by Gabor filters and Multi Layer Perceptron
Nezamoddin N. Kachouie and Javad Alirezaie
IEEE-SMC, Washington, USA, Oct 2003 (Published)
SMC 2003 Conference Proceedings Pages: 2897-2902

4- Optimized multi-channel decomposition for texture segmentation using Gabor filter bank
Nezamoddin N. Kachouie and Javad Alirezaie
Image Processing: Algorithms and Systems III, IS & T/SPIE Symposium on Electronic Imaging 2004, 18-22 January 2004 in San Jose, CA USA (Accepted, Would be published)

5- Texture segmentation of satellite images using multi-channel filter bank and neural networks
Nezamoddin N. Kachouie and Javad Alirezaie
Image Processing: Algorithms and Systems III, IS & T/SPIE Symposium on Electronic Imaging 2004, 18-22 January 2004 in San Jose, CA USA (Accepted, Would be published)

6- A hybrid algorithm using Discrete Cosine Transform and Gabor Filter Bank for Texture Segmentation

Nezamoddin N. Kachouie and Javad Alirezaie
IEEE-CCECE, Canada, 2004 (Accepted)

7- Multilevel segmentation by Multi Layer Perceptron for segmentation of IKONOS images of urban areas Nezamoddin N.
Kachouie, Javad Alirezaie and Jonathan Li
ICG (11th International Conference on Geoinformatics), Toronto, 2003 (Accepted)