Multi-scale texture segmentation and recognition

Heri Gunawan Widjojo
University of Wollongong

Follow this and additional works at: https://ro.uow.edu.au/theses

University of Wollongong
Copyright Warning
You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation
https://ro.uow.edu.au/theses/2461

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
MULTI-SCALE TEXTURE SEGMENTATION AND RECOGNITION

A thesis submitted in fulfilment of the requirements for
the award of the degree

MASTER OF ENGINEERING
(HONOURS)

from

The University of Wollongong

by

Heri Gunawan Widjojo, BE.

Department of Electrical and Computer Engineering
July 1993
ABSTRACT

Texture is a surface property of an object which humans routinely use in recognition and classification tasks. In image processing and pattern recognition texture is recognised as one of the main features useful for analysis in a variety of applications ranging from medical imaging to remote sensing. This thesis addresses the problem of texture-based identification of objects by a guided mobile robot. In the situation envisaged the acquisition of the texture image is allowed to take place at any distance. Since the viewing distance of the scene varies, the scale of the acquired image will also vary.

The possibility of incorporating scale change arising from varying viewing distance into a multi-scale texture recognition scheme is explored in this thesis. A model akin to the human visual system model in which there is a pre-attentive stage of segmenting the region of interest, followed by an attentive stage of recognition is proposed. Gabor filter, which is known to possess similar characteristic as the visual cortex, is employed to perform the pre-attentive task, i.e. to detect the region of interest. In order to reduce the computation time of the pre-attentive stage, a composite Gabor filter is introduced.

In the attentive stage, a multi-stage, hierarchical classification technique using Gabor filter is used. The performance of this method in terms of the classification accuracy is then compared with other statistical texture classification techniques. Two most widely used texture classification techniques are chosen, namely the gray-level co-occurrence and the frequency domain based methods.
I would like to express my deepest gratitude and appreciation to my supervisors Dr. Golshah A. Naghdy and Dr. Philip O. Ogunbona for their guidance, support and encouragement and also for shaping my research skills and abilities as an independent researcher throughout the course of this research.

My sincere thanks are also extended to the Australian Government for awarding me the Equity and Merit Scholarship Scheme without which I wouldn't be able to pursue higher education in Australia.

I wish to thank Ms. Maree Fryer and all members of staff in the Department of Electrical and Computer Engineering for their assistance throughout my research work. I am also indebted to all my colleagues who in one way or another contributed to the successful completion of this research work. Their help throughout the course is gratefully acknowledged.

Lastly, I could never forget my parents, who have provided vast amounts of moral support and encouragement which have contributed immensely to this work.
Table of Contents

Abstract i
Acknowledgment ii
Table of Contents iii
List of Figures v
List of Tables vii

1. INTRODUCTION 1
  1.1 Introduction 1
  1.2 Overview of the Project 3
    1.2.1 Image Acquisition and Preprocessing 7
    1.2.2 Region of Interest Detection and Segmentation 9
    1.2.3 Feature Extraction and Classification 10
    1.2.4 System Setup 11
  1.3 Organisation of the Thesis 12

2. LITERATURE SURVEY 13
  2.1 Introduction 13
  2.2 Texture Segmentation 14
  2.3 Texture Classification 17
  2.4 Contribution of the Thesis 19

3. PRE-ATTENTIVE AND ATTENTIVE TEXTURE RECOGNITION USING GABOR FILTERS 21
  3.1 Introduction 21
  3.2 Distance and Sampling Rate 22
  3.3 Maximum Viewing Distance 25
  3.4 Gabor Filter 30
  3.5 Pre-attentive Texture Segmentation 33
## List of Figures

1.1 The eight textures used in the experiment \( (d=100 \text{ cm}) \)  
1.2 Texture 3 at different scales  
1.3 Block diagram of object recognition process  
1.4 Block diagram of image acquisition process  

3.1 The relationship between distance and sampling rate  
3.2 The effect of varying distances on the power spectrum  
3.3 The effect of downsampling on the power spectrum  
3.4 The maximum viewing distance, \( d_{\text{max}} \) (in cm) for various values of \( p \)  
3.5 Gabor filter and its power spectrum  
3.6 The real and imaginary part of composite Gabor filter and its power spectrum for \( d=50 \text{ cm} \)  
3.7 The real and imaginary part of composite Gabor filter and its power spectrum for \( d=150 \text{ cm} \)  
3.8 The result of segmentation using composite Gabor filter, \( d=100 \text{ cm} \)  
3.9 The result of segmentation using composite Gabor filter, \( d=75 \text{ cm} \)  
3.10 The result of segmentation using composite Gabor filter, \( d=150 \text{ cm} \)  
3.11 The Gaussian filter used in smoothing
4.1 Power spectra of $x(t) = \cos(2\pi10.5t)$ sampled at 128 Hz and 256 Hz

4.2 Power spectra of $x(t) = \cos(2\pi10.5t)$ sampled at 64 Hz and 32 Hz

4.3 Intra-subclass spread measure (method 1), Frequency domain feature measures

4.4 Intra-subclass spread measure (method 2), Frequency domain feature measures

5.1 Intra-subclass spread measure (method 1), Gray level Co-occurrence feature measures

5.2 Intra-subclass spread measure (method 2), Gray level Co-occurrence feature measures
List of Tables

3.1 The maximum viewing distance, $d_{max}$ value (in cm) for each texture for various values of $p$ 29
3.2 Parameters of the filters as a function of viewing distance 37
3.3 The location of the major frequency component of the eight textures at $d = 50$ cm 37
3.4 The location of the major frequency component of the eight textures at $d = 50$ cm 43
3.5 Classification result of the first stage 44
3.6 Classification result of the second stage 45
3.7 The final result using a two-stages, hierarchical classification 45

4.1 The size of the subimage related to the distance 61
4.2 Intra-subclass spread measure (%) for method 1 and 2, Frequency domain feature measures 62
4.3 Intra-class spread measure (%) for method 1 and 2, Frequency domain feature measures 66
4.4 Classification results using nearest-neighbour classifier for method 1 and 2, Frequency domain feature measures 67
4.5 Classification results using Multi-layer Perceptron for method 1 and 2, Frequency domain feature measures 68

5.1 The size of the subimage and the displacements related to the distance 75
5.2 Intra-subclass spread measure (%) for method 1 and 2,
Gray level co-occurrence feature measures

5.3 Intra-class spread measure (%) for method 1 and 2,
Gray level co-occurrence feature measures

5.4 Classification results using nearest-neighbour classifier for method 1 and 2, Gray level co-occurrence feature measures

5.5 Classification results using Multi-layer Perceptron for method 1 and 2, Gray level co-occurrence feature measures
CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The application of computer vision keeps expanding. One application area of computer vision is in manufacturing industry. The applications range from fully automated assembly to inspection and verification tasks. Vision is used to automate the process, taking over high risk and dangerous tasks, thus eliminating the need for human interference which tends to be slow and inaccurate.

Autonomous vehicles are another possible use. Such vehicles are intended to operate with little or no human guidance. In performing their functions, they must be able to successfully cope with difficult tasks, such as navigating through their environment. Multiple sensory signals are often used, including vision and audition. Robot navigation is concerned with the guiding of robots along a path and detecting any obstacles in their passageways. Most of the mobile robots in present use, however, are not autonomous (self guided). They rely on tracks which have been laid out to guide them and on ultrasonic sensors to detect obstacles. Current research in mobile robots has been directed towards the utilisation of vision sensors as a substitute for both track and ultrasonic sensors and also as a means of environment perception.
A robot needs to extract important information from the environment in order to perceive it. One such important information is the knowledge of the objects in the surrounding area. The knowledge about the object can be obtained from the main features of the object. The main distinguishing features of an object for visual perception are: the outline or the shape of the object, the skeleton, and the textures of the materials that fill or cover it (Pickett, 1970). Most objects in the real world have different textured surfaces. This makes texture an important characteristic for object recognition.

In computer vision, texture analysis plays an important role in segmentation and classification techniques. It provides cues for distinguishing objects in natural scenes and is useful for recognising type of surfaces. Object recognition methods can be viewed as having two steps. In the first step, appropriate measurements are extracted from an image using low-level image processing techniques. These measurements are then utilised to derive a preliminary segmentation of the image into regions. In the second step, higher-level descriptors of the segmented regions are compared with the object properties stored in the knowledge base to recognise the object. These methods are modelled after the human visual system in which visual perception operates in two modes, a pre-attentive stage of inspection followed by an attentive stage of recognition (Treisman, 1985).

In the pre-attentive stage, the eye takes in the surroundings without focusing on any particular object. It scans the entire field of view at a low resolution to take in the general state of being in its surroundings. It is in the attentive stage that the visual system concentrates on a particular region within the field of view to register the details of the object and only in this mode is recognition possible.
In this thesis texture is considered as a characteristic for recognising objects. Therefore, the problem of object recognition can be narrowed down to texture classification problem. Most work in texture classification in the past, however, has been limited to images acquired from a fixed distance. They have also been restricted to the classification of an entire image of homogeneous texture. This thesis addresses the problem of texture-based identification of objects by a guided mobile robot. In the situation envisaged the acquisition of the texture image is allowed to take place at any distance and the image acquired may not be a homogenous image. A method of texture recognition which is not restricted to one particular image scale and homogeneous images is proposed.

1.2 OVERVIEW OF THE PROJECT

Figures 1.1 and 1.2 show the images that are used in the experiments. Figure 1.1 shows eight images containing different textures at the same scale (viewing distance, \(d=100\) cm) while figure 1.2 shows one texture (texture 3) at different viewing distances. The textures are synthetic and deterministic textures. The reason why natural textures were not used for the experiments is due to the random property of the natural textures. Using deterministic textures, one can concentrate on the scale variance problem since the main source of the variance, that is the random property of natural textures, has been excluded. Although the analysis is confined to deterministic textures, the result can be readily extended to natural and stochastic textures.

These multi-scale textures arise as a result of a motion toward an object. The number of basic pattern element (texture primitive) inside a given area is inversely proportional
to the distance between the camera and the textural surface. As the robot moves
toward an object, the image area corresponding to the texture cells increases and as
the robot moves farther away the area decreases. This viewing distance is, however,
limited. The texture can still be recognised as the original texture only within a certain
range.

Figure 1.1 The eight textures used in the experiment \((d=100 \, \text{cm})\)
(a) texture 1  (b) texture 2  (c) texture 3  (d) texture 4
Figure 1.1 (contd.)
(e) texture 5  (f) texture 6  (g) texture 7  (h) texture 8
Figure 1.2 Texture 3 at different scales. 
(a) $d=50$ cm (b) $d=75$ cm (c) $d=100$ cm (d) $d=125$ cm
Figure 1.3 depicts the process of object recognition. The first two processes constitute the pre-attentive stage of the human visual system while the last two processes are accomplished in the attentive stage. Although this thesis mainly addresses the multi-scale texture recognition problem in the application of object recognition by a mobile robot, the method adopted here is generally applicable to other multi-scale texture recognition problem which might arise in practice.

1.2.1 IMAGE ACQUISITION AND PREPROCESSING

The image acquisition process is shown in figure 1.4. The distance between the robot/camera and the object is measured using the ultrasonic sensor. An autofocus camera is used to grab the image. The camera is fixed with zero degree of freedom, so
it can only capture frontal view in the direction of robot movement. The image is then digitised into 256x256 pixels and the intensities are quantised into 64 gray levels.

![Block diagram of object recognition process](image)

**Figure 1.3** Block diagram of object recognition process

One might assume that the robot motion can be partitioned into discrete "move-stop" motions. During the "stop" period, image is captured and the distance is measured. Hence the analysis of images can be confined to static and monocular images. Both the image and the distance are then processed to recognise what is in the front of the
robot. This process should be performed in real time so that the robot motion looks as a continuous motion.

![Image acquisition process diagram](image)

**Figure 1.4** Image acquisition process.

### 1.2.2 REGION OF INTEREST DETECTION AND SEGMENTATION

Segmentation is the process of dividing the image into meaningful regions. The principal objective is to segment a given image, in order to be able to process only the region of interest (ROI) for optimum information, thus reducing the volume of data that needs to be processed for cognition. This has the effect of localising the ROI to a small part, or the aggregate of small parts of the entire image.
There are a number of techniques for image segmentation. The method that is adopted in this thesis is based on spatial filtering. There is a class of spatial filters which is appropriate for textural segmentation: these filters are known as the 2D Gabor filters (Clark et al., 1987). There is also a physiological evidence that receptive field profiles of the "simple cells" existing in the visual cortex have a Gabor filter characteristic (Daugman 1985; Berges and Pribram, 1992). Apart from this relationship, Gabor filters also have some desirable properties: tunable orientation and radial frequency bandwidth, tunable centre frequencies, localisation in the spatial and frequency domains and optimally achieving joint resolution in space and spatial frequency (Clark et al, 1987). By tuning the parameters of the Gabor filter to one of the dominant frequencies, a particular texture can be segmented from the image.

1.2.3 FEATURE EXTRACTION AND CLASSIFICATION

Once the ROI has been found, the next step is to represent the region with a set of numerical features whose dimension is much lower than the original image data. This is an important step in any pattern classification techniques since it reduces redundancy in the data and drastically cuts the computational cost. Nevertheless, choosing a set of features which retains much of the useful discriminatory information present in the original data is the most difficult part in pattern recognition. The possibility of selecting good features will in general simplify the classification process. Similar textures will have numerically close values while textures from different classes will have numerically different values. Therefore, a simple classifier will be good enough to discriminate distinctly clustered patterns. Intra-class and inter-class variance are usually used to evaluate the discriminating performance of the features.
Many feature extraction methods for texture recognition have been proposed in the literature (Haralick and Shapiro, 1992), but most of the them address the classification problem under the assumption that the given textures possess the same scale and orientation and the ROI is known. Most of these methods perform poorly when the scale or orientation of the texture varies. This thesis is mainly concerned with detection of ROI and the scale variation problem. Several popular texture feature extraction method will be studied and possibly extended to incorporate scale normalisation.

After the ROI is found, a window is extracted from the region. Since the size of texture primitive is a function of distance, a different window size should be employed so that each window contains the same number of texture elements. The size should be large enough to contain a number of repetition of texture primitives required to identify a particular texture. Thompson (1976) referred to this minimal image region as texture resolution. Before features are extracted from the window, histogram equalisation is applied to remove the variability caused by differences in illumination. A set of features is calculated from this window and then, used for classification.

1.2.4 SYSTEM SETUP

All the experiments in this project have been carried out on IBM PC compatible with 80386DX/33 MHz processor and 80387 math coprocessor. Images of different resolutions were obtained for each texture by varying the distance between the CCD camera and the object. The frame grabber then digitised the analog signal from the camera. The output is a 256 x 256 pixel image with 256 gray levels which were then linearly reduced to 64 gray levels for memory and display reasons. The images were obtained from 50 cm, 60 cm, 75 cm, 90 cm, 100 cm, 115 cm, 125 cm and 150 cm, so
there were 64 images from eight different textures. These images form the image database which will be used in the experiments.

1.3 ORGANISATION OF THE THESIS

A literature survey of current research in the area of texture recognition is presented in chapter 2. The material described in chapter 2 will provide background information for the work described in later chapters. A summary of contribution of this thesis is provided at the end of chapter 2. Chapter 3 discusses the use of Gabor filter to perform both pre-attentive and attentive task of human visual system. It begins with the discussion of the relationship between the change of scale in spatial and frequency domains, followed by the determination of maximum viewing distance from the sampling theorem. The composite Gabor filter for the purpose of texture segmentation is introduced and experimental results of the application of these filters for pre-attentive and attentive tasks are given next. Chapters 4 and 5 discuss two statistical texture classification methods to perform higher level interpretation in the attentive stage. The experimental studies show the effect of the normalisation and the merit of each method. Chapter 4 discusses the frequency domain feature extraction method while the gray level co-occurrence feature extraction method is studied in chapter 5. Finally some discussions and concluding remarks on the work are presented in chapter 6. Appendices A and B provide the theoretical background of bilinear interpolation algorithm and statistical pattern classification, respectively.
CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

Texture has long been known as an important characteristic for the analysis of many types of images, from multispectral scanner images obtained from aircraft or satellite platforms, which the remote-sensing community analyses, and microscopic images of cell and tissue samples, which the biomedical community analyses, to the outdoor scenes and object surfaces which the machine vision community analyses (Haralick and Shapiro, 1992). A number of applications in various areas can be found in the literature, for example: classification of land-use category and types of rocks (Haralick, 1973), classification of pulmonary disease (Sutton et al., 1972), carpet wear assessment (Siew et al., 1988), inferring three-dimensional shape and surface orientation on a perspective projection image (Moerdler, 1988; Ohta et al., 1981), etc.

There is no universally accepted definition of texture. The key point of texture definition is that texture refers to the arrangement or spatial distribution of intensity variations in an image. It is commonly attributed to images containing repetitive pattern in which "elements" or "primitives" are arranged according to certain "placement rules" or as a result of some random process. Image texture can be qualitatively evaluated as having one or more properties of fineness, coarseness,
smoothness, granulation, randomness, lineation or as being mottled, irregular, or hummocky. Each of these qualities translates into some property of the gray level primitives and the spatial interaction between them (Haralick and Shapiro, 1992).

There are three main issues in texture analysis (Ehrich and Foith, 1978):
1. Texture segmentation, in which the boundaries between different textured regions are to be determined.
2. Texture classification, in which, for a given textured region, the class, from a finite set, to which it belongs is to be determined.
3. Texture synthesis and modelling, in which a description or model for a texture is to be determined. Given a model and the values of its parameters one can synthesise homogeneous image texture.

These three divisions, however, deal with two-dimensional images only. Recent research efforts in texture analysis have been directed toward three-dimensional images, such as inferring three-dimensional shape from texture and surface orientation on a perspectively projected image of a textured planar surface. Therefore, one can regard this as the fourth issue in texture analysis.

### 2.2 TEXTURE SEGMENTATION

Segmentation is the partitioning of an image into regions that are homogeneous with respect to one or more characteristics (Gonzales and Woods, 1993). Homogeneous regions can be characterised by a uniform gray level or by a uniform texture. Image segmentation methods based only on gray level information are not suitable for images
in which regions exhibit almost the same average gray level and differ only for local variation or texture (Alparone et al., 1989).

Segmentation is essential for the functioning of a machine vision system, since segmentation leads to picking out the ROI for optimum information, thus reducing the volume of data that needs to be processed for cognition. Two conventional methods of segmentation are edge or boundary detection and region growing. The edge detection approach is based on the principles of recognising dissimilarities in an image, whereas region growing utilises the inherent similarities.

More recently, attention has been focused on human-visual-system-based perception scheme. Two general classes of models describing texture segmentation in the human visual system (HVS) have been proposed. The first model suggests that texture discrimination is accomplished in the HVS via the computation of the densities of certain texture primitives called *textons* (Julesz and Bergen, 1983). Textons are elongated blobs defined in terms of their width, length, color, orientation and terminators. This model also suggests the existence of two separate, functionally distinct levels of visual processing, one pre-attentive and the other attentive (Julesz, 1986; Treisman 1985).

In the pre-attentive mode (which is parallel), differences in a few local structural features are detected over the entire visual field. The HVS is capable of detecting the ROI if it is different from other areas in frequency, size, color, brightness, or orientation (Beck et al., 1983). Julesz (1984) found evidence that the pre-attentive system performs immediate texture discrimination only on the basis of first order statistics of textons. In the attentive mode (which is sequential), a serial search over a
narrow aperture of attention is performed in fast steps. Only in this mode is recognition possible.

The second class of texture segmentation models is based on spatial filtering. This approach, which is more consistent with the early processing that occurs in the HVS, makes use of a large set of spatial filters, each tuned to specific texture properties. This suggests that texture segmentation may be accomplished using narrowband spatial filters tuned to different portions of the 2D Fourier spectrum characterizing varying degrees of texton density and orientation, rather than by first locating textons and then computing these quantities.

Several authors have advanced algorithms for processing texture using frequency measures, principally by partitioning the spectrum into bins (Ikonomopolidoulos and Unser, 1984) or by locating spectral peaks (D'Astous and Jernigan, 1984). However, the methods used by these authors did not model specific receptive field characteristics found in the HVS and were not defined with any specific optimality criteria.

In recent years, research in vision biology suggested that biological processing based on spatial filtering can be approximated by 2D Gabor operators (Daugman, 1980; Porat and Zeevi, 1988). Physiological experiments have shown that there is a close relationship between the architecture of the neural receptive field in the striate visual cortex and the Gabor elementary functions (Daugman, 1980; Bovik et al., 1990; Berges and Pribram, 1992). Apart from this relationship Gabor filters also have some desirable properties: tunable orientation and radial frequency bandwidth, tunable centre frequencies, localisation in the spatial and frequency domains and optimally
achieving joint resolution in space and spatial frequency (Daugman, 1980; Clark et al., 1987).

Texture segmentation using Gabor filters can be accomplished in supervised and unsupervised mode. In supervised mode, one has a priori knowledge about the texture. The frequency and orientation characteristics of these textures can be determined so that Gabor filters can be tuned to these orientations and frequencies (Bovik et al., 1990; Clark, et.al., 1987). In unsupervised mode, a set of Gabor filters of different scale and orientation is used (Turner, 1986; Jain and Farrokhnia, 1991). In this method it is important that a large set of channel filters is used to sample the frequency plane densely to ensure that a filter exists that will respond strongly to any dominant texture frequency component.

2.3 TEXTURE CLASSIFICATION

The first step in the solution to any classification problem is to reduce the dimension of data to a computationally reasonable amount while preserving much of the salient information present in the actual data. This can be accomplished by choosing a set of numerical features. Over the past few years, a number of texture classification techniques has been proposed and has achieved considerable success, although these techniques are for the most part ad hoc. Haralick and Shapiro (1992) provide a comprehensive survey of the existing feature extraction methods.

Generally, these techniques can be characterised as either structural or statistical in their approach. In the latter, statistical properties of spatial distribution of the image gray levels are utilised as textural features. On the other hand, the structural methods
lay their foundation upon characterising texture in terms of its primitives and the placement rule governing their arrangement within the field of view. This approach typically suits regular repetitive patterns but runs into difficulty when dealing with nonregular, random textures.

Statistical features may be extracted, for example, from the power spectra (D'Astous and Jernigan, 1984; Liu and Jernigan, 1990, Roan et al., 1987), sum and difference gray level statistics (Unser, 1986), co-occurrence matrices (Haralick et al., 1973; Roan et al., 1987), gray level run length (Galloway, 1974; Weszka et al., 1976). Another method is to model texture as a stochastic process and to look at textured images as realizations or samples from parametric probability distributions on the image space (Kashyap and Khotanzad, 1986). This approach has the clear advantage that it is capable of producing textures that match the observed textures as well as parameters of various random field models.

Spatial frequency is related to texture because fine textures are rich in high spatial frequencies while coarse textures are rich in low spatial frequencies. The gray level co-occurrence approach characterises texture by the distribution of the co-occurrence of its gray level values. Coarse textures are those for which the distribution changes only slightly with displacement and fine textures are those for which the distribution changes rapidly with displacement. The gray level run length approach characterises coarse textures as having many pixels in a constant gray level run and fine textures as having few pixels in a constant gray level run. The random field model characterises texture by using linear estimates of a pixel's gray level, conditioned on the pixels in a neighbourhood containing it. For coarse textures, the coefficients of the model will all be similar while for fine textures, the coefficients will have wide variation because the
gray level values in the neighbourhood are similar for coarse textures. (Haralick, 1979).

A flexible recognition system should be insensitive to parameters such as orientation, scale and location of the object in the field of view. If the features extracted from the image are invariant to such parameters, the classifier is relieved from the difficult task of handling these variations. Roan et al. (1987) showed that gray level co-occurrence and power spectra methods are not scale invariant. They applied the methods to multi-resolution textures which were obtained by varying the viewing distance to the textural surface. Kashyap and Khotanzad (1986) introduced a model-based approach, called circular symmetric autoregressive model, for texture classification which is rotation invariant. Cohen and Fan (1988) pointed out the limitation of this model, arguing that the model imposes isotropic structure on possibly nonisotropic texture. They, instead, proposed a model-based rotation and scale invariant classifier and reported 100% accuracy rate using their method.

In the classification stage, a proper class is assigned to the feature measures according to a decision rule. There are well defined decision rules in pattern classification theory. Appendix B gives a summary of the theory of statistical pattern recognition.

### 2.4 CONTRIBUTION OF THE THESIS

Most works in image texture analysis have been limited to one of the three issues mentioned in section 2.1. This thesis addresses the real world problem, that is the problem of texture recognition from various distances, in a unified approach which encompasses the most important issues in texture analysis problem, i.e. segmentation,
and classification. Furthermore, the segmentation and classification is not performed on single-scale textures but on multi-scale textures.

The method adopted here is based on a combination of the human visual system model and statistical method. There are a large number of textural analysis techniques already. A new technique of multi-scale texture segmentation and classification using Gabor filter, based on the pre-attentive and attentive stage of human visual system, is proposed. A comparison study of this technique and other existing texture classification methods is presented. Some modification and extension for the existing methods is proposed to speed up the computation and to normalise against various scales. This thesis also gives a comparative result of the performance of two different classifiers: statistical and neural network classifiers, in terms of the classification accuracy.
CHAPTER 3
PRE-ATTENTIVE AND ATTENTIVE
TEXTURE RECOGNITION
USING GABOR FILTERS

3.1 INTRODUCTION

The framework for textural analysis developed in this chapter has its origin in recent
physiological evidence of the architecture of the neural receptive field in the striate
visual cortex (Daugman, 1985; Bovik et al., 1990; Berges and Pribram, 1992). It has
been suggested that visual perception operates in two modes, one pre-attentive and the
other attentive (Julesz, 1986; Treisman 1985).

In the pre-attentive phase of human vision, the eyes scan the entire field of view at a
low resolution to take in the general state of being in its surroundings and to detect
approximate location of the region of interest (ROI), without focusing on any object.
This allows fast identification of the areas of the image where relevant information lies.
Further analysis of the region and the interpretation of the information are performed
in the next phase of human visual process, that is, in the attentive phase. In this stage,
the area of field of view comprising the ROI is scanned at a higher resolution which is
the optimum required to register the details of that object (Catanzariti et al., 1990).

The problem of recognising multi-scale textures necessitates the use of a classification
scheme that is, in principle, scale invariant or in which one can at least normalise for
the scale changes. The possibility of incorporating scale change arising from varying viewing distance into a multi-scale texture classification scheme will be explored further in this chapter.

This chapter starts with the discussion about the relationship between distance and sampling rate and then employs this relationship to find the maximum viewing distance. The composite Gabor filter will be introduced and employed to detect the ROI in the image. Finally, the performance of the Gabor filter in the attentive texture recognition is explored.

### 3.2 DISTANCE AND SAMPLING RATE

If the visual angle subtended by the camera is constant, as the robot moves toward a surface, the image area of the texture primitive increases. At some distance the surface may have image characteristic that differ from those of the original texture. A similar effect occurs if the distance is made larger. As the robot moves farther away, texture details of the original pattern elements may no longer be discernible and the repetitive placement of those elements may yield new texture cells (Roan et al., 1987). Texture cannot be analysed without a frame of reference. For any textural surface there exists a scale at which, when the surface is examined, it appears smooth and textureless. As resolution increases, the surface appears as a fine texture and then a coarse one (Haralick and Shapiro, 1992).

The process of viewing an image from various distances can be considered as the process of sampling a two-dimensional signal at different sampling rates. Increasing the distance is equivalent to reducing the sampling rate. Figure 3.1 describes the relationship between distance and sampling rate. The original configuration is shown in figure 3.1a with the camera centre at distance $d$ from texture surface and a total visual
angle of $\alpha$. Changing the sampling angle of $\beta$ to $\delta$, where $\delta = 0.5\beta$, will double the resolution. The same resolution can also be achieved when the distance between the camera and the texture surface is reduced to $0.5d$ while holding the total visual angle at $\alpha$ and the fixed sampling angle at $\beta$ as shown in figure 3.1b. Thus, increasing the sampling rate (angle) has the same effect as reducing the distance proportionally while the sampling rate is fixed.

![Diagram of camera and texture surface angles](image)

**Figure 3.1** (a) Original configuration, $\alpha =$ total visual angle; $\beta =$ sampling angle, $\delta = 0.5\beta$. (b) Halving the distance (Roan et al., 1987)
If a signal is sampled at different sampling rates, the resulting power spectra will still retain the spectral shape of the original signal as long as the sampling theorem is not violated. Only the frequency axis is relabelled. In order to show that varying the viewing distance has the same effect on the power spectrum as changing the sampling rate, three images were obtained from three different viewing distances. The power spectra of these images were then calculated and plotted. Figure 3.2 shows a texture viewed from $d = 50$ cm, 100 cm and 150 cm and its power spectra. These figures demonstrate that the power spectra of a single texture viewed from different distances are the same as the power spectra of one signal sampled at different sampling rates. This confirms the hypothesis that varying the viewing distance can be considered as changing the sampling rate and thus, further analysis can be based on the signal processing theory of multi-rate sampling.
3.3 MAXIMUM VIEWING DISTANCE

The sampling theory states that a bandlimited low pass signal can be recovered without aliasing error if the sampling rate \( f_s \) is at least twice of the highest frequency \( f_0 \) in the signal.

\[
f_s > 2f_0
\]

The sampling rate \( f_s \) can be reduced by a factor of \( M \) without aliasing if the original sampling rate is at least \( M \) times the Nyquist rate or if the bandwidth of the sequence is first reduced by a factor of \( M \) and every \( M \)th sample is retained by discrete filtering (Oppenheim and Schafer, 1989)

\[
f_s > Mf'_N
\]

where \( f_s \) is the original sampling rate and \( f'_N \) is the Nyquist rate.
Suppose that an image, taken from a distance $d_0$, is bandlimited such that the highest frequency along the $x$-axis, $u_0$, or along the $y$-axis, $v_0$, is $1/2M$ of the sampling frequency $f_s$ ($f_x = f_y = f_s$). Based on the relationship between distance and sampling rate and on the sampling theorem, the maximum distance $d_{\text{max}}$, from which the image can be viewed without aliasing can be determined as:

$$d_{\text{max}} = M d_0$$

And this maximum distance $d_{\text{max}}$ corresponds to the minimum sampling rate $f_{\text{min}}$:

$$f_{\text{min}} = \frac{f_x}{M}$$

Figure 3.3 shows the power spectra of a texture viewed from 50 cm, 100 cm and 150 cm respectively. At 100 cm the power spectrum still retains most of the dominant frequencies of the power spectrum at 50 cm, but at 150 cm some of the dominant peaks have been shifted out as shown in figure 3.3(c). It is assumed that these frequency components have been filtered out by the camera before the signal is digitised so no aliasing error has occurred. Even though there is no aliasing error, some information is lost due to the downsampling process. If too much information is lost, the downsampled signal cannot be identified as the original signal.
Based on the percentage of the lost energy, a quantitative analysis for determining the maximum viewing distance $d_{\text{max}}$ can be derived. Suppose that the power spectrum of a $N \times N$ image, sampled at the rate $f_s$ (or viewed from distance $d_0$) has been calculated. Let $P$ denote the total power of the $N \times N$ power spectrum and $P_n$ denote the total
power in $n \times n$ region centred at the origin (low-frequency area) of the power spectrum. Then the power ratio $p$ can be defined as:

$$p = \frac{P_n}{P} \cdot 100\%$$

The power ratio $p$ can then be employed to determine how far the image can be downscaled so that, at least $p\%$ of the total energy is preserved. The minimum sampling rate for which $p\%$ of the total power is retained is given by:

$$f_{\text{min}} = \frac{n}{N} \cdot f_s$$

and correspondingly, the maximum distance ($d_{\text{max}}$) is:

$$d_{\text{max}} = \frac{N}{n} \cdot d_o$$

Table 3.1 shows the average value of $d_{\text{max}}$ for various values of $p$ for the eight textures used in the experiments and figure 3.4 plots the $d_{\text{max}}$ as a function of $p$ ($d_o = 50$ cm). The figure shows the rate of information loss as the viewing distance is varied. Since the degree of information loss varies for each texture, so does the maximum viewing distance. In this study, however, the maximum viewing distance is fixed at 150 cm which results in varying degree of information loss for each texture. This may have an effect on the classification performance of each texture. The higher the frequencies present in the texture the more energy is lost and, as a consequence, the smaller the maximum viewing distance.

It can also be seen that texture 3 contains mostly high frequencies. As a result, the maximum distance for this texture is the lowest among other textures. Only 75% of the energy is preserved when the distance is changed from 50 cm to 115 cm. On the other
hand, the dominant frequencies of texture 1 lie in the low frequency area and most of the energy (more than 90%) is still preserved when the distance is increased, from 50 cm to 150 cm.

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>70%</td>
<td>300.00</td>
<td>187.00</td>
<td>115.00</td>
<td>166.00</td>
<td>187.00</td>
<td>166.00</td>
<td>166.00</td>
<td>166.00</td>
</tr>
<tr>
<td>75%</td>
<td>300.00</td>
<td>187.00</td>
<td>115.00</td>
<td>166.00</td>
<td>166.00</td>
<td>166.00</td>
<td>125.00</td>
<td>166.00</td>
</tr>
<tr>
<td>80%</td>
<td>300.00</td>
<td>180.00</td>
<td>107.00</td>
<td>140.67</td>
<td>166.00</td>
<td>166.00</td>
<td>107.00</td>
<td>150.00</td>
</tr>
<tr>
<td>85%</td>
<td>250.00</td>
<td>160.67</td>
<td>107.00</td>
<td>125.00</td>
<td>166.00</td>
<td>130.00</td>
<td>93.00</td>
<td>136.00</td>
</tr>
<tr>
<td>90%</td>
<td>187.00</td>
<td>125.00</td>
<td>88.00</td>
<td>107.00</td>
<td>150.00</td>
<td>93.00</td>
<td>88.00</td>
<td>118.33</td>
</tr>
<tr>
<td>95%</td>
<td>140.67</td>
<td>95.33</td>
<td>76.00</td>
<td>88.00</td>
<td>93.00</td>
<td>88.00</td>
<td>83.00</td>
<td>84.67</td>
</tr>
</tbody>
</table>

Table 3.1 $d_{max}$ value (in cm) for each texture for various values of $p$.

![Figure 3.4a](image)

Figure 3.4a $d_{max}$ (in cm) value for various values of $p$ (texture 1-4)
3.4 GABOR FILTER

Here textures are modelled as patterns which are distinguished by a high concentration of localised spatial frequencies. This model of texture allows the use of a channel filter for segmenting the image. There is a class of spatial filters which are appropriate for textural segmentation. These filters are known as the 2D Gabor filters. Recent physiological experiment has shown that there is a relationship between the architecture of the neural receptive field in the striate cortex and the Gabor elementary functions (Daugman, 1980; Berges and Pribram, 1992). Apart from this relationship Gabor filters also have some desirable properties: tunable orientation and radial frequency bandwidth, tunable centre frequencies, localisation in the space and
frequency domains and optimally achieving joint resolution in space and spatial frequency (Daugman, 1985; Clark et al., 1987; Bovik et al., 1990).

Gabor filter can be described in terms of a sinusoidal plane wave of some spatial frequency and orientation within a two-dimensional Gaussian envelope in the spatial domain or a shifted Gaussian function in the spatial frequency domain. The class of Gabor functions was described by Gabor (1946) and was extended to two dimensions by Daugman (1980). Daugman (1985) showed that Gabor filters are optimal in the sense that they minimise the product of effective areas occupied in the 2D space and 2D frequency domains.

The 2D Gabor function and its associated 2D Fourier transform have the general form:

\[ h(x,y) = \exp\left(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right) \exp\left(2\pi j (u_0 x + v_0 y)\right) \]

\[ H(u,v) = \exp\left(-2\pi^2 \left[\sigma_x^2 (u-u_0)^2 + \sigma_y^2 (v-v_0)^2\right]\right) \]

where \(\sigma_x\) and \(\sigma_y\) are the standard deviations of the Gaussian envelope along the \(x\) and \(y\) directions respectively and \((u_0, v_0)\) is the coordinate of the centre of the 2D Gaussian in the spatial frequency domain, which can also be expressed in the polar form \((f_0, \theta)\):

\[ f_0 = \sqrt{u_0^2 + v_0^2} \]

\[ \theta = \tan^{-1}\left(\frac{v_0}{u_0}\right) \]

where \(f_0\) is the radial frequency and \(\theta\) is the orientation.
Figure 3.5 shows one example of Gabor filter and its power spectrum. By varying the parameters $f_0$, $\theta$, $\sigma_x$ and $\sigma_y$, a Gabor filter of arbitrary orientation, centre frequency and bandwidth characteristics can be generated spanning any elliptical portions of the spatial frequency domain.

![Figure 3.5 A sample of Gabor filter. (a) Real part (b) Imaginary part (c) Power spectrum](image-url)
3.5 PRE-ATTENTIVE TEXTURE SEGMENTATION

The principal objective of this stage is to approximate location of the ROI in order to be able to process only the area of interest for optimum information, thus reducing the volume of data that needs to be processed for cognition. This has the effect of localising the ROI to a small part of the entire image. The exact boundary between different regions is not of major concern since the overall objective of this pre-attentive stage is to find the rough location of the ROI.

A simple texture can be modelled as modulated sinusoidal functions of the form (Bovik et al., 1990):

\[ t(x, y) = a(x,y) \cos 2\pi f_0 (x \cos \theta - y \sin \theta) + \phi(x, y) \]

This is an ideal model for a texture since real-world textures generally contain multiple frequency components distributed over the spatial frequency planes. Nevertheless, using the a priori knowledge that the textures used in this study are highly oriented and periodic and are quite distinct from the background, it might suffice to use one dominant frequency to discriminate these textures from the background.

The image \( t(x, y) \) is then filtered using a Gabor filter which is tuned to the radial frequency \( f_0 \) and orientation \( \theta \). Filtering an image with a 2D Gabor filter results in an image containing only a limited range of frequencies and orientations. The filtering process is often implemented as a convolution of the image data with a two-dimensional mask suitably defined to carry out the filtering operation in the spatial domain.

Thus,

\[ k_e(x, y) = \text{Re}(h(x,y)) * t(x, y) \]
\[ k_s(x,y) = \text{Im}(h(x,y)) \ast f(x,y) \]

where \( \ast \) represents convolution and \( k_c \) and \( k_s \) denote the filtered outputs.

The amplitude of the response can be obtained via:

\[ m(x,y) = k_c^2(x,y) + k_s^2(x,y) \]

Segmentation is usually carried out by comparing the values of channel amplitude response, \( m(x,y) \), over different regions of the image. The magnitude of the filtered image will be maximised over those regions of the original image characterised by the particular spatial frequency attributes to which the filter is tuned (Bovik et al., 1990). This segmentation method has been used successfully in a number of pre-attentive texture segmentation experiments (Clark et al., 1989; Fogel and Sagi, 1989; Bovik et al., 1990; Jain and Farrokhnia, 1991). This method, however, has some limitation, especially when there are many textures to be discriminated.

Suppose that an image is to be segmented and it may contain one of \( n \) available textures, then at least \( n \) filters are required for the segmentation, with one filter for each texture. Since the texture in the image is unknown, segmentation is carried out by convolving the image with all of the filters. As the number of the textures to be segmented increases, this method becomes computationally impractical. Nevertheless, this method has an advantage that both pre-attentive and attentive stages are carried out in a single pass, i.e. recognition is implicit in the segmentation process.

To reduce the computation time, a composite Gabor filter was employed in this study. The composite Gabor filter \( g(x,y) \) is defined as a sum of Gabor filters \( h_l(x,y) \), each tuned to a particular channel frequency:
\[ g(x, y) = \sum_{i=1}^{n} h_i(x, y) \]

and

\[ h_i(x, y) = \exp \left\{ -\frac{1}{2} \left[ \frac{x^2}{\sigma_{xi}^2} + \frac{y^2}{\sigma_{yi}^2} \right] \right\} \exp \left\{ 2\pi i [u_{oi}x + v_{oi}y] \right\} \]

where \( n \) is the number of Gabor filters, \( h_i(x, y) \) is \( i \)-th Gabor filter, tuned to \((u_{oi}, v_{oi})\) and \( \sigma_{xi}, \sigma_{yi} \) are the standard deviations of the Gaussian envelope of the \( i \)-th Gabor filter along the \( x \) and \( y \) directions respectively.

This method has the advantage that the convolution needs to be performed only once for each image. Therefore, only one composite Gabor filter is required for all the images at one particular scale. The number of filters is independent of the number of textures to be discriminated. In this method, however, the recognition has to be done in a separate step.

Once the ROI has been segmented, higher-level interpretation of the texture types can be carried out over that area only. Assuming that the region is uniform, recognition can then be performed on samples taken from the region. Many texture classification algorithms that can be applied to these samples exist in the literature. In this work, however, Gabor filter is used in both pre-attentive and attentive stage to perform low-level segmentation and higher-level interpretation of the texture types.

As part of the pre-processing, a spectral analysis of the textures was carried out to determine the location of the dominant peaks. This information was then used in designing the Gabor filters. Spectrum analysis was performed on each texture at one particular scale only. Using the relationship between distance and sampling rate as
discussed in the previous section, the location of the peaks for the rest of the textures can be derived. In this study \( d_0 = 50 \) cm was chosen to be the reference.

Notwithstanding its merit, the composite Gabor filter has some limitation. For a particular texture, the composite Gabor filter contains a frequency channel, which is tuned to the most dominant frequency of the texture, as well as other channels which are not. These channels can pass other frequency components of the texture. Although the responses to those channels may be assumed to be small, these, however, can render the channel amplitude response \( m(x, y) \) not smooth enough for the segmentation purpose. Nevertheless, this leakage effect can be reduced by post-filtering the channel amplitude response with a low-pass filter.

The segmentation was carried out by convolving the filter with the image in the spatial domain. Different filter sizes were used for different scales. By choosing filter size which was in proportion with the scale (distance), the number of cycles covered by the filters is kept constant. It can be assumed here that one cycle represents one textural element. Therefore, the number of textural elements which was convolved with the filter is kept constant regardless of the scale of the elements. Table 3.2 gives the size of the filter and the spread of the Gaussian function \((\sigma_x, \sigma_y)\) as a function of viewing distance. In this study \( \sigma_x = \sigma_y \) is chosen for simplicity and \( \sigma_x = \sigma_y = \text{filter size}/3 \) is used so that the image is covered within, at least, 95% of the spread of the Gaussian function.

Table 3.3 shows the location of the major peak: the radial frequency \( f_0 \) (in cycles/pixel) and the orientation \( \theta \) (in radian) for the eight textures at \( d = 50 \) cm. These parameters were then used to design the composite Gabor filter for \( d = 50 \) cm. Figures 3.6 and 3.7 show the perspective plot of composite Gabor filters and its power spectra for \( d = 50 \) cm and 100 cm. The parameters for other distances can be derived from table 3.2 and
table 3.3. The orientation parameter $\theta$ does not change for different distances. The radial frequency $f_0$ is inversely proportional to the distance.

<table>
<thead>
<tr>
<th>Viewing distance</th>
<th>Filter size</th>
<th>$\sigma_x$ ($\sigma_y$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>60</td>
<td>20.0</td>
</tr>
<tr>
<td>75</td>
<td>40</td>
<td>13.3</td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>10.0</td>
</tr>
<tr>
<td>125</td>
<td>24</td>
<td>8.0</td>
</tr>
<tr>
<td>150</td>
<td>20</td>
<td>6.67</td>
</tr>
</tbody>
</table>

**Table 3.2** Parameters of the filters as a function of viewing distance

<table>
<thead>
<tr>
<th>Texture</th>
<th>$f_0$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.083333</td>
<td>0.643501</td>
</tr>
<tr>
<td>2</td>
<td>0.094281</td>
<td>2.356194</td>
</tr>
<tr>
<td>3</td>
<td>0.116667</td>
<td>1.570796</td>
</tr>
<tr>
<td>4</td>
<td>0.066667</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.083333</td>
<td>0.643501</td>
</tr>
<tr>
<td>6</td>
<td>0.133333</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.116667</td>
<td>1.570796</td>
</tr>
<tr>
<td>8</td>
<td>0.134371</td>
<td>1.051650</td>
</tr>
</tbody>
</table>

**Table 3.3** The location of the major frequency component of the eight textures at $d = 50$ cm.
Figure 3.6 The composite Gabor filter, $d=50$ cm
(a) real part  (b) imaginary part  (c) power spectrum

Figure 3.7 The composite Gabor filter, $d=150$ cm
(a) real part  (b) imaginary part  (c) power spectrum
Figures 3.8, 3.9 and 3.10 depict the original images, the images after being filtered with a composite Gabor filter and the final segmentation results after being smoothed and thresholded. As expected, the responses were not smooth. By post-filtering the envelope, a more consistent segmentation was obtained. A Gaussian filter of size 30 x 30 and standard deviation = 1/3 of its size was employed to smooth the filtered image. Figure 3.11 shows the Gaussian filter which was used for smoothing.

Figure 3.8 The result of segmentation using composite Gabor filter
(a) The original image: texture 3, $d=100$ cm  (b) The image after being filtered.
(c) The filtered image after smoothing and thresholding.
Figure 3.9 The result of segmentation using composite Gabor filter
(a) The original image: texture 5, $d=75$ cm   (b) The image after being filtered.
(c) The filtered image after smoothing and thresholding.
Since the objective of this pre-attentive stage is to approximate the location of the ROI, and not to find an accurate segmentation or complete description of the textures, the computation time can be further reduced by sacrificing the resolution. Normally convolution is performed at every position \((x, y)\) but in order to save the number of computation, the convolution can be performed only at every \((\gamma x, \gamma y)\), where \(\gamma\) is an
integer constant. This method is similar to convolving the filter with a low-resolution version of the original image.

### 3.6 ATTENTIVE TEXTURE RECOGNITION

Once the ROI has been found, the next step is to classify its textural content. In this section Gabor filter is employed for the purpose of texture classification. Assuming that the characteristics of the $N$ different textures to be discriminated are known, then a set of filters needed for each texture type can be designed. The radial frequency and orientation of the dominant frequencies of each texture are the characteristics that will be used to discriminate different textures. Similar to the approach employed in the pre-attentive stage, classification of textures is also based on the amplitude response of the filtered subimages. First, each of the $N$ Gabor filters is convolved with the image over $b \times b$ grids within the ROI. This produces $b \times b$ output values for each filter.

$$m_n(x, y) = |t(x, y) \ast h_n(x, y)|, \quad n = 1, \ldots, N; \quad 0 \leq x, y \leq b - 1$$

where $h_n(x, y)$ is the Gabor filter tuned to texture $n$; $m_n(x, y)$ denotes the filter amplitude response; $t(x, y)$ is the image; and $| \ldots |$ is the norm function.

The average amplitude response of the filter, $\overline{m}_n$, is calculated from these $b \times b$ values and the region is assigned to the class corresponding to the filter which has the maximum average amplitude response $\overline{m}_n$. 
The location of the major peak frequency of the eight textures at \( d = 50 \) cm is shown again in table 3.4 for convenience. For each texture, at least one Gabor filter, particularly tuned to the texture, is required. The Gabor filter was designed based on the peak frequency location of a particular texture. As shown in table 3.4, some textures, used in this study, have the same peak locations, for example texture 1 and 5, texture 3 and 7.

<table>
<thead>
<tr>
<th>Texture</th>
<th>( f_a )</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.083333</td>
<td>0.643501</td>
</tr>
<tr>
<td>2</td>
<td>0.094281</td>
<td>2.356194</td>
</tr>
<tr>
<td>3</td>
<td>0.116667</td>
<td>1.570796</td>
</tr>
<tr>
<td>4</td>
<td>0.066667</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.083333</td>
<td>0.643501</td>
</tr>
<tr>
<td>6</td>
<td>0.133333</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.116667</td>
<td>1.570796</td>
</tr>
<tr>
<td>8</td>
<td>0.134371</td>
<td>1.051650</td>
</tr>
</tbody>
</table>

Table 3.4 The location of the major frequency component of the eight textures at \( d = 50 \) cm.

To solve this problem, a multi-stage, hierarchical classification method was employed. In the first stage, classification was performed using the location of the major peak frequencies. Gabor filters with channel tuned to the location of the major peak frequencies were applied. Textures which have the same location of the major peak frequencies were resolved in the subsequent stages using Gabor filters with the channel tuned to the location of the second peaks, third peaks, etc.
The results of classification testing are usually cross-tabulated in the form of a contingency table (confusion matrix). The values along the diagonal represent the percentage of correctly classified patterns for each class; values along a given row indicate how misclassified patterns are distributed among the classes. The sum of the values in each row is 100 percent to account for all test patterns.

Eight textures from five different viewing distances (50, 75, 100, 125 and 150 cm) were used in the experiment. The composite Gabor filter was applied to extract the ROI from these images. Then, five sample points were randomly picked from within the ROI of each texture and scale. These points were chosen so that they didn't lie near the edge of the ROI in order to avoid convolution problem. From the parameters given in table 3.4, five Gabor filters were designed and employed to perform the first stage classification. The filters were convolved within 4 x 4 area around the selected points and the average amplitude response of each filter was calculated. Table 3.5 gives the result of classification based on the amplitude response of each filter.

<table>
<thead>
<tr>
<th>Texture</th>
<th>1 &amp; 5</th>
<th>2</th>
<th>3 &amp; 7</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 &amp; 5</td>
<td>42</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3 &amp; 7</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>21</td>
</tr>
</tbody>
</table>

Classification accuracy = 84%

Table 3.5 Classification result of the first stage.

Since some textures have the same location of major peak frequencies, further classification stages are required. In this experiment only two stages were required, since the location for the second or third major peaks are not the same. Texture 1 and
5 could be discriminated using the second major peaks, while texture 3 and 7 were resolved using the third major peaks, since the first and second peaks for these textures are on the same location. Table 3.6 shows the result of the second stage classification and table 3.7 gives the final classification result after the results from both stages are combined. As one can see, the maximum classification accuracy using this multi-stage, hierarchical classification approach, is determined from the first stage. Further stages only help in resolving textures which are indiscriminable in prior stages but they cannot improve the accuracy rate obtained from the first stage.

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Texture</th>
<th>3</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 3.6 Classification result of the second stage.

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>

Classification accuracy = 82.5%

Table 3.7 The final result using a two-stages, hierarchical classification.
The classification accuracy, on the other hand, is determined from how the textures are grouped in the first stage. Most of the samples from texture 3 were misclassified as texture 6 due to the fact that the location of the secondary peak frequency of texture 3 is actually the location of the major peak frequency of texture 6, as one can see from table 3.8. Suppose that texture 3 and 6 were put in one group, then the classification result should be different. Texture 3 would then be mostly misclassified as texture 7.

Other possible source of misclassification is that the resolution is not high enough to distinguish closely-spaced frequencies. The dominant frequencies of the textures may not actually be on the same spot. They are just close in position and the resolution is not high enough to resolve them. To increase the resolution, the size of the filter and observation window need to be increased. Increasing the size of the filter and observation window, however, might not be feasible if the ROI in the image is not large enough. Other method to increase the resolution is to use a model-based spectral estimation instead of periodogram in finding the peak frequency locations.

3.7 CONCLUSION

In this chapter it was shown that the pre-attentive stage of human vision can be simulated using Gabor filter banks. The relationship between distance and sampling rate was presented and the maximum viewing distance for each texture was derived from this relationship.

To reduce the number of computation, composite Gabor filter was introduced and was used to perform the pre-attentive task of detecting the ROI. Further post-processings
such as smoothing and thresholding were then performed on the filtered image to improve the result of the composite Gabor filter. Although the composite filter is prone to leakage, the experiment showed that this method produced quite satisfactory results.

The experiments using the Gabor filter to perform the attentive task, has also been tried. The problem arose when some textures have common peak locations. Increasing the filter size and the observation window size might help in resolving the common peak location problem, but this method might not be feasible if the ROI is not large enough. Other method which can be employed to resolve closely-spaced frequencies is to use model-based spectrum analysis instead of periodogram in finding the location of the peaks.

The experiment shows that this problem can also be solved by using a multi-stage, hierarchical classification technique. In the first stage of this method, discrimination is based on the location of the major peak frequencies. If some textures cannot be distinguished, further stages are then carried out using second, third peaks etc. A single-channel Gabor filter can only capture the basic structure of a texture. This might not suffice to discriminate different textures since the detail of the textures cannot be captured by utilising one frequency component only. Therefore, a multi-stage, hierarchical classification technique is usually required. The upper bound of the classification accuracy of this method is determined from the accuracy rate obtained from the first stage. Further stages can only help in discriminating textures which cannot be resolved in prior stages. It cannot improve the accuracy rate.
CHAPTER 4
FREQUENCY DOMAIN FEATURE MEASURES

4.1 INTRODUCTION

It has been shown in the previous experiments that the Gabor filter method is not suitable for the attentive task. It has several limitations and is a computationally-intensive process. In this chapter another method for the attentive task of the visual system, i.e. recognition, is discussed. Recognition is performed using features extracted from the power spectra of the textures. This method does not follow any human visual system model as in the approach adopted in the previous chapter. The possibility of incorporating scale-normalisation in this method will also be explored.

Texture refers to the arrangement or spatial distribution of intensity variations in an image. Texture features should be measures which most completely embody information about the spatial distribution. Since specific components in the spatial frequency domain representation of an image contain explicit information about the spatial distribution, one expects to obtain useful texture features in the frequency domain (D' Astous and Jernigan, 1984).
Frequency domain technique constitutes one of the most widely used feature extraction methods in texture analysis. Discrete Fourier Transform has shift invariance property and it has also been shown to be insensitive to additive noise (Liu and Jernigan, 1990). Ideally, if there is no information loss due to the downsampling process, the feature values calculated from the intra-class images, that is images obtained from one source and viewed from different distances, should be similar and have small variance. This will reduce the classification problem to that of distinguishing between different source images (textures). This, however, will never happen in a practical situation. The variance for intra-class images always exists because of the inaccuracy in measurement, noise, different lighting condition and many other factors. Nevertheless, the variance is negligible compared to the variance produced by the sampling process.

4.2 TEXTURE FEATURE MEASURES

In this section feature measures which will be used in the classification are presented. Liu and Jernigan (1990) gave an exhaustive list of feature measures that can be extracted from the power spectrum. Only six out of twenty-eight features in this list were used in the experiments since the rest are obviously sensitive to scale variation.

Dealing with digital images, one can only approximate Fourier transform using the Discrete Fourier Transform. The power spectrum calculated using DFT, or periodogram, gives an estimate of the real power spectrum. There are several limitations in spectral estimation using DFT. For example the location and the amplitude of the peak in DFT are not necessarily the location and amplitude of the real peaks in the Fourier transform. These limitations, which will be discussed in the
next section, increase the variance of the feature measures, since all of the features are derived from the magnitude and/or the peak location.

The limitation of the periodogram method has never been addressed on texture analysis, discrimination or classification since the experiments were performed on single-scale images. All images were taken from the same distance or, in other words, sampled at the same rate. It will be shown from the experiments in the following sections that this limitation of DFT does not have much effect on single-scale images. The problem arises when recognition is to be carried out on multi-scale images. The variance of the power spectra of one signal of one particular scale is negligible compared to the variance of the power spectrum of one signal at different scales. It will be shown later that one signal can produce different power spectra when it is sampled at different rates due to the inherent limitation of DFT. But before going any further into this problem, the features used in the classification experiments will be introduced.

Let \( P(u,v) \) be the power spectrum obtained from the image and the normalised power spectrum \( p(u,v) \) is defined as

\[
p(u,v) = \frac{P(u,v)}{\sum_{u,v} P(u,v)}
\]

Features derived from the normalised power spectrum \( p(u,v) \) are:
1. Energy in major peak

\[ f_1 = p(u_1, v_1) \]

where \( u_1, v_1 \) are the frequency coordinates of the maximum peak of the power spectrum.

2. Percentage energy, quadrant I,

\[ f_2 = \sum_{u>0} \sum_{v>0} p(u, v) \]

3. Moment of Inertia, quadrant I,

\[ f_3 = \sum_{u>0} \sum_{v>0} (u^2 + v^2)^{\frac{3}{2}} p(u, v) \]

4. Moment of Inertia, quadrant II,

\[ f_4 = \sum_{u<0} \sum_{v>0} (u^2 + v^2)^{\frac{3}{2}} p(u, v) \]

5. Relative entropy of power spectrum, R1,

\[ f_5 = \left[ - \sum_{u, v \in R_1} p_1(u, v) \log p_1(u, v) \right] / \log K_1 \]
where

\[
p_i(u, v) = \frac{P(u, v)}{\sum_{u,v \in R_i} P(u, v)}
\]

and

\[
K_i = \text{number of distinct frequencies in } R_i
\]

\[
R_i = \left\{ u, v \text{ such that } \frac{i-1}{2} u_m \leq |u| < \frac{i}{2} u_m \text{ and } \frac{i-1}{2} v_m \leq |v| < \frac{i}{2} v_m \right\}
\]

\(u_m, v_m\) are the maximum frequency components for the local spectrum

6. Relative entropy \(R_2\)

\[
f_6 = \left[ - \sum_{u,v \in \mathbb{Z}_2} p_2(u,v) \log p_2(u,v) \right] / \log K_2,
\]

### 4.3 DFT LIMITATION

The DFT of a windowed sequence are the samples of the Fourier transform at the equally spaced discrete-time frequencies. Spectral sampling as imposed by the DFT can produce misleading results. Some problems that result in errors between the computed and the desired transform are leakage and Picket-fence effect.
4.3.1 Leakage

This problem arises because of the practical requirement that observation of the signal must be limited to a finite interval. The process of terminating the signal after a finite number of terms is equivalent to multiplying the signal by a window function. If the window function is a rectangular function, the series is abruptly terminated without modifying any coefficients within the window.

Windowing smears the impulses in the theoretical Fourier representation. This smearing effect is known as leakage. The result of leakage is an undesirable modification of the total spectrum. Limiting observation to a finite number of samples also reduces the ability to resolve sinusoidal signals that are closely spaced in frequency. In order to resolve these closely-spaced frequency components, a large number of data samples must be used.

Reduced resolution and leakage are the two primary effects on the spectrum as a result of truncating the signal to a finite number of samples. The resolution is primarily influenced by the width of the mainlobe of the window function's spectrum, while the degree of leakage depends on the relative amplitude of the mainlobe and the sidelobes of the spectrum of the truncating window.

The leakage effect cannot always be isolated from the aliasing effect because leakage may also lead to aliasing. Since leakage results in a spreading of the spectrum, the upper frequency of the spectrum may move beyond the folding frequency and aliasing may then result. This possibility is particularly significant in the case of a rectangular window function, since the tail of the window spectrum does not converge rapidly.
4.3.2 Picket-Fence Effect

The spectral sampling inherent in the DFT has the effect of potentially giving a misleading or inaccurate picture of the true spectrum of the sinusoidal signal. This effect is produced by the inability of DFT to observe the spectrum as a continuous function, since computation of spectrum is limited to integer multiple of the fundamental frequency.

It is possible that the major peak lies between two of the discrete transform lines. The location of peaks in the DFT sequence does not necessarily coincide with the exact frequency location of the peaks in the continuous Fourier transform. Correspondingly, the relative amplitudes of the peaks in the DFT will not necessarily reflect the relative amplitudes of the true spectral peaks. This presents a major problem since the feature measures are derived from the magnitude and the location of the peaks. This problem can be easily demonstrated using a one-dimensional sinusoidal signal sampled at different rates.

Figure 4.1 shows the power spectrum of a one-dimensional signal sampled at two different frequencies. The signal, containing only one frequency component, \( f = 10.5 \) Hz, was sampled at two different frequencies: \( f_s = 128 \) Hz and \( f_s = 256 \) Hz. The length of the observation windows \( (T_o = 1 \text{ s and } T_o = 0.5 \text{ s}) \) is made proportional to the sampling rate so that the number of sample points is kept constant. It can be shown that \( f \) is not an integer multiple of fundamental frequency for either \( f_s = 128 \) Hz or \( f_s = 256 \) Hz.

For \( f_s = 128 \) Hz:

\[
k = \frac{f}{f_s} \cdot N = \frac{10.5}{128} \cdot 128 = 10.5
\]
and for $f_s = 256$ Hz:

$$k = \frac{f}{f_s} \cdot N = \frac{10.5}{256} \cdot 128 = 5.25$$

where $N$ is the number of samples, $f$ is the frequency component of the signal, $f_s$ is the sampling rate and $k$ is the DFT frequency index (horizontal axis in the power spectrum plot).

The magnitude of the true peak will be very difficult to find if the true peak lies in between two integer indices such as in the example above since the frequency index ($k$) is limited to an integer number in the DFT. As a result, the peaks in the power spectrum plot are not necessarily the real peaks. The true peaks can only be found using the continuous Fourier transform. Therefore, the peaks in figure 4.1 are not the true peaks since the frequency index $k$, obtained from the calculation above, is not integer. This also explains the reason why the magnitude of the peak in figure 4.1 (a) is not the same as in figure 4.1 (b).

If in the example above, $f_s = 64$ Hz or $f_s = 32$ Hz is used, one will get a picture of the true spectrum of the sinusoidal signal as it will be obtained using the continuous Fourier transform. This is illustrated in figures 4.2(a) and (b). The magnitude of the peaks in these two figures are consistent as opposed to the ones in figures 4.1 (a) and (b). This is possible because at $f_s = 64$ Hz or $f_s = 32$ Hz the peak lies at integer frequency index $k$.

For $f_s = 64$ Hz:

$$k = \frac{f}{f_s} \cdot N = \frac{10.5}{64} \cdot 128 = 21$$
and for $f_s=32$ Hz:

$$k = \frac{f}{f_s} \cdot N = \frac{10.5}{32} \cdot 128 = 42$$

### 4.4 PERFORMANCE MEASURE

In order to assess the texture discrimination performance of the features, a quantitative measure is required. But before further detail about the measures is discussed, some definitions of terms that will be used throughout this chapter and the subsequent chapters are given, i.e. intra-subclass, intra-class and inter-class. Intra-subclass is used to refer to the same texture viewed from the same distance while intra-class refers to the same texture viewed from different distances and inter-class refers to different textures.

Figure 4.1a Power spectrum of $x(t) = \cos(2\pi 10.5t)$ sampled at 128 Hz.
Figure 4.1b Power spectrum of $x(t)=\cos(2\pi 10.5t)$ sampled at 256 Hz.

Figure 4.2a Power spectrum of $x(t)=\cos(2\pi 10.5t)$ sampled at 64 Hz.
Figure 4.2b  Power spectrum of \( x(t) = \cos(2\pi 10.5t) \) sampled at 32 Hz.

The spread and the clustering properties of the intra-subclass and intra-class samples are the factors that determine the performance of the feature measures. Since the variance of a cluster domain about its mean can be used to infer the relative distribution of the samples in the domain, the sample variance will be employed as intra-subclass spread measure and intra-class spread measure. The intra-subclass and intra-class spread measure show the relative distribution of the samples within the subclass and the class. Small value in the measures indicates that the samples are closely clustered about their centroid.

The intra-subclass spread of class \( i \) and subclass \( j \), \( \eta_{ij} \), is defined as:

\[
\eta_{ij} = \frac{1}{n} \sum_{k=1}^{n} \eta_{ij}, \quad i = 1, 2, \ldots, M \text{ and } j = 1, 2, \ldots, R
\]
where \( n \) is the number of feature measures or the dimensionality of the pattern vector, \( M \) is the number of pattern class, \( R \) is the number of subclass, \( \eta_{ijk} \) is the \( k \)-th component of vector \( \eta_{ij} \), and

\[
\eta_{ij} = \frac{\sigma_y}{\bar{x}_y}
\]

\[
\sigma_y = \sqrt{\frac{1}{N_y} \sum_{x \in \omega_y} (x - \bar{x}_y)^2}
\]

\[
\bar{x}_y = \frac{1}{N_y} \sum_{x \in \omega_y} x
\]

where \( x \) refers to the pattern vector, \( \omega_y \) is \( j \)-th subclass of the \( i \)-th pattern class, \( \bar{x}_y \) and \( \sigma_y \) denote the sample mean and standard deviation of \( j \)-th subclass of \( i \)-th class, respectively and \( N_y \) is the number of sample patterns in \( \omega_y \).

The intra-class spread of class \( i \) (\( \eta_i \)) is defined as:

\[
\eta_i = \frac{1}{n} \sum_{k=1}^{n} \eta_{ik}, \quad i = 1, 2, \ldots, M
\]

where \( n \) is the number of feature measures or the dimensionality of the pattern vector, \( M \) is the number of pattern classes, \( \eta_k \) is the \( k \)-th component of vector \( \eta_i \), and

\[
\eta_i = \frac{\sigma_i}{\bar{x}_i}
\]
\[ \sigma_i = \sqrt{\frac{1}{N_i} \sum_{x \in \omega_i} (x - \bar{x}_i)^2} \]

\[ \bar{x}_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \]

\[ N_i = \sum_{j=1}^{R} N_{ij} \]

where \( x \) refers to the pattern vector, \( \omega_i \) is the \( i \)-th pattern class, \( \bar{x}_i \) and \( \sigma_i \) denote the sample mean and standard deviation of the \( i \)-th class, respectively and \( N_i \) is the number of sample patterns in \( \omega_i \).

### 4.5 EXPERIMENTAL STUDY

In this experiment the maximum viewing distance, \( d_{\text{max}} \) was chosen to be 150 cm for all textures. As is shown in table 3.1 in the previous chapter, setting \( d_{\text{max}} \) constant on different textures will result in various degree of information loss for each texture.

The ROI was segmented during the pre-attentive stage, therefore in this attentive stage, the analysis was carried out on the ROI only. Sample subimages were randomly extracted from the ROI to form training and test databases. The training database consists of ten sample subimages for each texture and distance. Only images from 50 cm, 75 cm, 100 cm, 125 cm, and 150 cm were used for the training database. Thus there were four-hundred sample subimages in the training database from eight textures and five distances. The test database contains five sample subimages for all textures and distances (50 cm, 60 cm, 75 cm, 90 cm, 100 cm, 115 cm, 125 cm, and 150 cm). Since there were eight different textures and eight different viewing distances, the test
database comprises 320 samples. Note that the test database contains samples which are not available in the training database, i.e. samples of images taken from 60 cm, 90 cm and 115 cm.

The size of the subimage (the length of observation window) is inversely proportional to the distance from which the image is taken (sampling rate). Therefore, each subimage will contain the same number of textural elements regardless of their size. Using subimage size of 60x60 pixels for the image taken from 50 cm as a reference, the window size for other distances can be determined using the proportionality condition as shown in table 4.1.

Experiments involving two methods dealing with subimages were carried out. In the first method, features were extracted from the subimage without any normalisation while in the second method, the size of the subimage was normalised and the features were extracted from the new subimages. Therefore, in the second method, all subimages have the same size. They were all normalised to a size of 30 x 30 pixels. Bilinear interpolation algorithm, explained in Appendix A, was employed to normalise the size of the subimages. For both methods, before feature extraction was carried out, each subimage was first histogram equalised to remove the variability caused by differences in illumination.

<table>
<thead>
<tr>
<th>Distance (cm)</th>
<th>Subimage size</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>60 x 60</td>
</tr>
<tr>
<td>60</td>
<td>50 x 50</td>
</tr>
<tr>
<td>75</td>
<td>40 x 40</td>
</tr>
<tr>
<td>90</td>
<td>34 x 34</td>
</tr>
<tr>
<td>100</td>
<td>30 x 30</td>
</tr>
<tr>
<td>115</td>
<td>26 x 26</td>
</tr>
<tr>
<td>125</td>
<td>24 x 24</td>
</tr>
<tr>
<td>150</td>
<td>20 x 20</td>
</tr>
</tbody>
</table>

Table 4.1 The size of the subimage related to the distance
The features defined in the previous section were extracted using the Discrete Fourier Transform (DFT) instead of the Fast Fourier Transform (FFT) since the subimages can be of any size which need not be integer power of two. In this case the DFT is still more flexible than the mixed-radix FFT.

Table 4.2 shows the intra-subclass spread measures for both methods. On the average the value of the spread measure is less than ten percent which means that patterns from the same subclass are well clustered in the feature space. Since the main source of variation for intra-subclass images are noise, measurement error, lighting condition, etc., the result shows that the variance caused by these factors is not significant. Comparing the spread measures from both tables, one will find that the size-normalisation increases the intra-subclass measures. This might be explained from the fact that the bilinear interpolation employed here is just an approximation to the real up/down sampling process that occurs as a result of changing the viewing distance. Bilinear interpolation performs a linear approximation of a pixel value from its nearest neighbours while the real up/down sampling process is not necessarily linear and depends on larger neighbourhood.

<table>
<thead>
<tr>
<th>Txtr</th>
<th>50cm</th>
<th>75cm</th>
<th>100cm</th>
<th>125cm</th>
<th>150cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.8863</td>
<td>2.4029</td>
<td>3.5598</td>
<td>3.0528</td>
<td>3.8257</td>
</tr>
<tr>
<td>2</td>
<td>2.6278</td>
<td>2.9710</td>
<td>3.7881</td>
<td>2.9880</td>
<td>4.4988</td>
</tr>
<tr>
<td>3</td>
<td>2.3092</td>
<td>3.1450</td>
<td>3.4886</td>
<td>2.0910</td>
<td>3.5154</td>
</tr>
<tr>
<td>4</td>
<td>4.9638</td>
<td>4.6706</td>
<td>5.9707</td>
<td>5.3110</td>
<td>5.1760</td>
</tr>
<tr>
<td>5</td>
<td>4.2461</td>
<td>4.7520</td>
<td>4.6658</td>
<td>5.1593</td>
<td>6.3902</td>
</tr>
<tr>
<td>6</td>
<td>5.7147</td>
<td>5.4304</td>
<td>6.5240</td>
<td>6.6037</td>
<td>7.0918</td>
</tr>
<tr>
<td>7</td>
<td>2.3123</td>
<td>3.4944</td>
<td>2.9515</td>
<td>5.1793</td>
<td>4.8365</td>
</tr>
<tr>
<td>8</td>
<td>2.6386</td>
<td>3.5693</td>
<td>3.6581</td>
<td>5.7419</td>
<td>9.8988</td>
</tr>
</tbody>
</table>

(a)
Table 4.2 Intra-subclass spread measure (%)  
(a) First method  (b) Second method

Figures 4.3 and 4.4 show $\eta_{ij}$ as a function of distance. As can be observed from these figures, $\eta_{ij}$ does not always increase with the increasing distance. This means that reducing the sampling rate does not always increase the variance of the intra-subclass images.
Figure 4.3 Intra-subclass spread measure (method 1)
(a) Texture 1-4  (b) Texture 5-8

(a) Texture 1 + Texture 2  ○ Texture 3  △ Texture 4

(b) Texture 5 + Texture 6  ○ Texture 7  △ Texture 8
Table 4.3 shows the intra-class spread measures. The value of intra-class spread measure is much larger compared to that of intra-subclass spread measure given in table 4.2. This confirms the hypothesis that the main source of data variability is the scale variance. The variance caused by noise etc. is negligible compared to the variance caused by the scale variation. Table 4.3 also shows that the size-normalisation approach in the second method reduces the intra-class spread although it increases the intra-subclass spread.

The classification was performed employing nearest neighbour and artificial neural network classifier. The details of the nearest neighbour and artificial neural network classifiers are given in Appendix B. The neural network used in the experiment is the
Multi-layer Perceptron (MLP) with three layers. The input layer consists of six nodes which is equal to the number of features. The hidden layer has twelve nodes and the output layer has eight nodes in which each node represent one texture class. Among these eight output nodes, the node with the maximum value is compared to a threshold, and if it is above the threshold value, the texture class corresponding to that node is chosen to be the output. However, if none of the nodes is above the threshold value, the input is grouped as 'unclassified class'.

<table>
<thead>
<tr>
<th>Texture</th>
<th>$\eta_i(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.0849</td>
</tr>
<tr>
<td>2</td>
<td>15.1341</td>
</tr>
<tr>
<td>3</td>
<td>24.5999</td>
</tr>
<tr>
<td>4</td>
<td>15.1047</td>
</tr>
<tr>
<td>5</td>
<td>18.2629</td>
</tr>
<tr>
<td>6</td>
<td>19.3446</td>
</tr>
<tr>
<td>7</td>
<td>15.0896</td>
</tr>
<tr>
<td>8</td>
<td>20.6083</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Texture</th>
<th>$\eta_i(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.0942</td>
</tr>
<tr>
<td>2</td>
<td>6.1204</td>
</tr>
<tr>
<td>3</td>
<td>12.7668</td>
</tr>
<tr>
<td>4</td>
<td>7.2781</td>
</tr>
<tr>
<td>5</td>
<td>9.8308</td>
</tr>
<tr>
<td>6</td>
<td>8.7663</td>
</tr>
<tr>
<td>7</td>
<td>8.1175</td>
</tr>
<tr>
<td>8</td>
<td>8.9519</td>
</tr>
</tbody>
</table>

(b)

Table 4.3 Intra-class spread measure (\%)
(a) Method 1  (b) Method 2
The classification results using nearest neighbour classifier are given in table 4.4 while the results using MLP are given in table 4.5. The accuracy rate for the first method is quite low that is 76.6% for the nearest neighbour and 77.2% for the MLP. The second method, however, achieved a good classification accuracy, i.e. 97.2 % for nearest neighbour and 96.9 for MLP.

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>24</td>
<td>3</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>19</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
</tr>
</tbody>
</table>

Classification accuracy: 76.6%

(a)

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>24</td>
<td>3</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>19</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
</tr>
</tbody>
</table>

Classification accuracy: 97.2%

(b)

Table 4.4 Classification results using nearest-neighbour classifier
(a) Method 1 (b) Method 2
The results show that the features are sensitive to size variation. The frequency domain features, defined in section 4.2, depend mainly on the relative power of each frequency component. The relative power, on the other hand, depends on the size of the subimage. The longer the length of the observation window (i.e. the size of the subimage), the higher the relative power of the signal. However, there is no simple way to relate the relative power to the size of the subimage due to the limitation of the DFT as discussed in the previous section. Therefore, even though the relative power
spectrum has been normalised in the calculation of the feature, the features are still sensitive to size variation as is indicated in the result of the classification.

Two methods of comparing the performance of the nearest neighbour and MLP classifiers are the classification accuracy and the computational complexity. The classification accuracy of both nearest neighbour and MLP classifiers in these experiments were about the same. The main computational requirement of a nearest neighbour classifier is during the testing phase while for MLP it is during the training phase. Both nearest neighbour and MLP have their own merits and demerits. The nearest neighbour classifier is perhaps the simplest of all classifiers to implement on a computer. In addition, it has the advantage of being guaranteed to give an error rate within a factor of two of the ideal error rate obtainable with a Bayesian classifier (Duda and Hart, 1973). The nearest neighbour, however, has the disadvantage that it requires enormous storage to record enough training set pattern vectors and correspondingly large amounts of computation to search through them to find an optimal match for each test pattern. The computational complexity is, therefore, in proportion to the number of patterns in the training set.

The testing phase of the MLP, on the other hand, does not depend on the number of patterns in the training set. The main difficulty with the MLP is the absence of a systematic method for selection of the appropriate number of hidden nodes. The other problem is the slow rate of convergence of the back-propagation learning rule and the fact that it does not guarantee a global minimum when searching for the best weights. However, once the MLP has been trained, the classification can be accomplished almost instantaneously.
4.6 CONCLUSION

The experiments show that the feature measures used are sensitive to size variation. The main source for the large variance is the size variation while the effects of noise, measurement error and other factors on the variance are negligible. This can be seen from the fact that the variance of intra-subclass patterns is small compared to the variance of intra-class patterns. It has also been shown that a simple size normalisation and interpolation technique can reduce the intra-class spread and consequently, improve the classification accuracy. The classification accuracy for the nearest neighbour classifier and MLP classifier are at about the same level, but for practical reason, MLP is better than the nearest neighbour classifier, since the classification can be accomplished almost instantaneously and it doesn't need a vast amount of storage.
CHAPTER 5
GRAY LEVEL CO-OCCURRENCE FEATURE MEASURES

5.1 INTRODUCTION

The fact that the gray level co-occurrence (GLC) method has been used successfully on a number of texture analysis problems (Kruger et al., 1974; Haralick et al., 1973) makes it a good choice for further study in the multi-scale texture discrimination problem in this project. Furthermore, both theoretical and empirical comparison studies (Weszka et al., 1976; Conners and Harlow, 1980) have shown it to be superior to the other texture analysis algorithms. And perceptual psychology studies (Julesz, 1962; Julesz et al., 1973) have shown it to match a level of human perception. Visa (1989) also showed that GLC method is quite insensitive to moderate variations in spatial and in gray scale resolution.

In this chapter the GLC method is applied to the multi-scale texture. The GLC method has been known to be invariant to translation but it's not invariant to scaling. To obtain scale-invariant property, some kind of normalisation should be performed. This chapter discusses the normalisation process and its effect on the performance of the GLC method in the multi-scale texture recognition.
5.2 GRAY LEVEL CO-OCCURRENCE MATRICES

GLC matrices are based on second-order statistics, that is the estimation of the joint probability functions of two pixels in some given relative position. GLC matrices count the number of pairs of gray levels of pixels that are separated by a certain distance along a certain direction.

The GLC matrix $M_d$ is a matrix whose $(i,j)$th element is the number of times that gray levels $i$ and $j$ occur in the relative position $d=(\Delta x, \Delta y)$. Given an $M \times N$ neighbourhood of the input image containing gray levels ranging from 0 to $(L-1)$, let $f(m,n)$ be the intensity level of pixel at location $(m,n)$ of the neighbourhood. Then the normalised GLC matrix $P_d(i,j)$ is given by (Peckinpaugh, 1991):

$$P_d(i,j) = W \cdot M_d(i,j)$$

where:

$W$ is normalising factor.

$$W = \frac{1}{(M - \Delta x)(N - \Delta y)},$$

and

$$M_d(i,j) = \#\{(x_1,y_1),(x_2,y_2)) \mid f(x_1,y_1) = i, f(x_2,y_2) = j \text{ and } x_2 = x_1 + \Delta x, y_2 = y_1 + \Delta y\}$$

where $\#$ denotes the number of elements in the set (the cardinality of the set).
The normalised GLC matrix, $P_{ij}(i,j)$ is an $L \times L$ matrix of second-order probability estimates of a transition from a pixel with gray level $i$ to a pixel with gray level $j$ given that the two pixels are $(\Delta x, \Delta y)$ units apart.

Co-occurrence matrix is not used directly as a feature for classification, but it is employed as an intermediate matrix, from which features are computed. Haralick et al. (1973) suggest 14 features to extract useful textural information from the GLC matrices. These features capture some characteristic of textures, such as homogeneity, coarseness, periodicity and others. Two features were used in this experiment, i.e. contrast ($f_1$) and cluster shade ($f_2$). These features are chosen among the most widely used features which have been used successfully in the literature (Weszka et al., 1976; Conners and Harlow, 1980; Peckinpaugh, 1991).

$$f_1 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i - j)P(i,j|\Delta x, \Delta y)$$

$$f_2 = \sum_{i=0}^{L-1} \sum_{j=1}^{L-1} (i + j - \mu_x - \mu_y)^2 P(i,j|\Delta x, \Delta y)$$

### 5.3 EXPERIMENTAL STUDY

Similar classification experiments were carried out as in chapter four. The experiments were carried out using the same training and test database set and two different methods were employed. In the first method, the feature extraction was
applied to the subimage of various sizes without size normalisation on these
subimages. Different displacements $d=(\Delta x, \Delta y)$ were used for different image scale
during the calculation of GLC matrices in order to compensate the size variation.
Recall that in order to generate GLC matrices, $P_{i,j}(i,j)$, the displacements $\Delta x$ and $\Delta y$
need to be specified. The displacements $\Delta x$ and $\Delta y$ were made proportional to the
distance in order to perform normalisation. If $\Delta x_0$ and $\Delta y_0$ were the displacements
used to calculate the GLC matrices from the reference image (i.e. the image taken
from the reference distance, $d_0$), then the displacements for the image taken from the
distance $d$ can then be obtained from:

$$\Delta x = \frac{\Delta x_0}{\alpha}$$
$$\Delta y = \frac{\Delta y_0}{\alpha}$$

where

$$\alpha = \frac{d}{d_0}$$

In case that the values of $\Delta x$ and $\Delta y$ are not integer, the pixel value at that point were
interpolated from the neighbouring pixels using bilinear interpolation technique, which
is discussed in appendix A. Using subimage size of 30x30 pixels for the image taken
from 100 cm as a reference and $\Delta x_0=3$ and $\Delta y_0=3$, the window size and the
displacements for other distances can be determined using the proportionality
condition as shown in table 5.1.

In the second method, the size of the subimage was normalised using bilinear
interpolation. Since the size of all subimages was the same after normalisation, the
displacements $\Delta x$ and $\Delta y$ were also fixed. All subimages were scaled to a size of 30 x
30 pixels with the displacements $\Delta x$ and $\Delta y = 3$. 
Table 5.1 The size of the subimage and the displacements related to the distance

<table>
<thead>
<tr>
<th>Distance (cm)</th>
<th>Subimage size</th>
<th>$\Delta x, \Delta y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>60 x 60</td>
<td>6.0</td>
</tr>
<tr>
<td>60</td>
<td>50 x 50</td>
<td>5.0</td>
</tr>
<tr>
<td>75</td>
<td>40 x 40</td>
<td>4.0</td>
</tr>
<tr>
<td>90</td>
<td>34 x 34</td>
<td>3.4</td>
</tr>
<tr>
<td>100</td>
<td>30 x 30</td>
<td>3.0</td>
</tr>
<tr>
<td>115</td>
<td>26 x 26</td>
<td>2.6</td>
</tr>
<tr>
<td>125</td>
<td>24 x 24</td>
<td>2.4</td>
</tr>
<tr>
<td>150</td>
<td>20 x 20</td>
<td>2.0</td>
</tr>
</tbody>
</table>

For both methods, four GLC matrices were generated for each subimage. Each matrix was calculated using one of four displacements: $(\Delta x, 0)$, $(\Delta x, \Delta y)$, $(0, \Delta y)$, $(-\Delta x, \Delta y)$. The two features were then extracted from each matrix, thus each pattern was represented by an eight-dimensional feature vector. Table 5.2 shows the intra-subclass spread measures for both method. As in the previous chapter the value of the intra-subclass spread measure is less than ten percent on the average. Figures 5.1 and 5.2 give the plot of this spread measure as a function of distance.
Table 5.2 Intra-subclass spread measure (%)  
(a) Method 1 (b) Method 2

<table>
<thead>
<tr>
<th>Txt.</th>
<th>50cm</th>
<th>75cm</th>
<th>100cm</th>
<th>125cm</th>
<th>150cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6942</td>
<td>3.0310</td>
<td>1.9770</td>
<td>2.1088</td>
<td>1.9037</td>
</tr>
<tr>
<td>2</td>
<td>3.0231</td>
<td>3.7571</td>
<td>3.5516</td>
<td>2.5809</td>
<td>3.5625</td>
</tr>
<tr>
<td>3</td>
<td>3.9016</td>
<td>4.4218</td>
<td>3.9650</td>
<td>5.9807</td>
<td>8.0307</td>
</tr>
<tr>
<td>4</td>
<td>4.1775</td>
<td>3.6598</td>
<td>4.4133</td>
<td>2.8480</td>
<td>4.6879</td>
</tr>
<tr>
<td>5</td>
<td>2.7397</td>
<td>3.8023</td>
<td>2.0187</td>
<td>2.8807</td>
<td>2.8559</td>
</tr>
<tr>
<td>6</td>
<td>3.4704</td>
<td>4.9666</td>
<td>2.5397</td>
<td>3.4005</td>
<td>3.7782</td>
</tr>
<tr>
<td>7</td>
<td>2.9040</td>
<td>3.5136</td>
<td>2.5155</td>
<td>4.0822</td>
<td>4.1811</td>
</tr>
<tr>
<td>8</td>
<td>2.7400</td>
<td>2.5575</td>
<td>2.1288</td>
<td>2.8845</td>
<td>6.4983</td>
</tr>
</tbody>
</table>

(a)

(b)

\[ \eta_j \]

Distance (cm)

- □ Texture 1
- + Texture 2
- ◦ Texture 3
- △ Texture 4
Figure 5.1 Intra-subclass spread measure (method 1)
(a) Texture 1-4 (b) Texture 5-8
Both intra-subclass and intra-class spread measures indicate that GLC method are superior to frequency domain method since the spread measures of GLC method are smaller than those of frequency domain method. Table 5.4 shows the classification result using nearest neighbour classifier while table 5.5 gives the result of the MLP classifier. The ninth column ('unclassified' column) of table 5.5 represents the input patterns which cannot be classified as one of the eight classes when none of the output nodes has value above the threshold.

These experiments also demonstrate that the normalisation in GLC method can also be carried out without normalising the size of the subimage. Unlike the frequency domain method, the result of this method is better than the result of normalising the
size of the subimage. The classification result of the nearest neighbour classifier, as in
the frequency domain technique, is comparable to that of the MLP classifier.

<table>
<thead>
<tr>
<th>Texture</th>
<th>$\eta_t(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.0984</td>
</tr>
<tr>
<td>2</td>
<td>9.1152</td>
</tr>
<tr>
<td>3</td>
<td>13.2123</td>
</tr>
<tr>
<td>4</td>
<td>9.5861</td>
</tr>
<tr>
<td>5</td>
<td>9.7293</td>
</tr>
<tr>
<td>6</td>
<td>12.2282</td>
</tr>
<tr>
<td>7</td>
<td>10.7484</td>
</tr>
<tr>
<td>8</td>
<td>13.1130</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Texture</th>
<th>$\eta_t(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.6510</td>
</tr>
<tr>
<td>2</td>
<td>5.4485</td>
</tr>
<tr>
<td>3</td>
<td>8.5818</td>
</tr>
<tr>
<td>4</td>
<td>6.0723</td>
</tr>
<tr>
<td>5</td>
<td>5.6995</td>
</tr>
<tr>
<td>6</td>
<td>6.5912</td>
</tr>
<tr>
<td>7</td>
<td>5.2280</td>
</tr>
<tr>
<td>8</td>
<td>4.6730</td>
</tr>
</tbody>
</table>

(b)

Table 5.3 Intra-class spread measure (\%)
(a) Method 1 (b) Method 2

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>35</td>
</tr>
</tbody>
</table>

Classification accuracy: 96.6\%
<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Unclassified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td></td>
</tr>
</tbody>
</table>

Classification accuracy: 95.6%

(b)

**Table 5.4** Classification results using nearest-neighbour classifier

(a) Method 1  (b) Method 2

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Unclassified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Classification accuracy: 97.8%

(a)

<table>
<thead>
<tr>
<th>Texture</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Unclassified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>32</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Classification accuracy: 96.6%

(b)

**Table 5.5** Classification results using Multi-layer Perceptron

(a) Method 1  (b) Method 2
5.4 CONCLUSION

The experiments showed that the GLC based feature measures are superior to the Fourier-based methods. This agrees with the result of the comparison study reported in the literature (Weszka et al., 1976; Conners and Harlow, 1980). Furthermore, GLC based feature measures are insensitive to slight variations in spatial and in gray scale resolution which is caused by normalisation.

Another advantage of the GLC method over the Fourier method is that normalisation can be performed implicitly during the generation of GLC matrices. Normalisation is carried out automatically by using different displacement \( d=(\Delta x, \Delta y) \) for different image scale. Therefore, a separate size normalisation stage is not required as in the other methods. This has the advantage of reducing the overall computation time.

As in the Fourier method, the spread of intra-class patterns are larger than the spread of intra-subclass patterns. This again confirms that the variation of the extracted feature values are mainly caused by the scale variation. The nearest neighbour classifier also has about the same degree of the classification accuracy as the MLP classifier.
CHAPTER 6
CONCLUSION

The problem of texture-based identification of objects by a guided mobile robot has been studied in this thesis. Since the acquisition of the texture image is allowed to take place at any distance, the scale of the acquired image varies. The possibility of incorporating scale change arising from varying viewing distance into a multi-scale texture recognition scheme has been explored.

A model akin to the human visual system model in which there is a pre-attentive stage of segmenting the region of interest, followed by an attentive stage of recognition was proposed in this thesis. Gabor filter, which is known to possess similar characteristic to the visual cortex, was employed to perform the pre-attentive task, i.e. to detect the Region of Interest (ROI).

In order to reduce the computation time of the pre-attentive stage, a composite Gabor filter was introduced. Although it is prone to leakage, the experiments have shown that the composite Gabor filter performed reasonably well to discriminate the texture against the background. Further post-processings, i.e. smoothing and thresholding were required to reduce the effect of leakage.

The relationship between distance and sampling rate from which the scale change can be incorporated into the Gabor filter was derived. Increasing the viewing distance has the same effect as reducing the sampling rate which can result in aliasing. In order to prevent aliasing, the signal should be low-pass filtered so that it is sufficiently bandlimited. This loss of information, due to the low-pass filtering or spectral truncation process affects the classification performance. The maximum viewing distance for each texture can be determined based on the degree of truncation that can
be tolerated. Therefore, there is always a trade-off between the maximum viewing distance and the classification performance. The viewing distance cannot be increased without sacrificing the classification accuracy.

For the attentive task of recognition, three methods were examined, i.e. the multi-stage, hierarchical classification using Gabor filter, the Gray level co-occurrence (GLC) method and the frequency domain based method. It was found that the Gray level co-occurrence method outperformed the other two in terms of classification accuracy. Comparison in terms of computational complexity is more involved. There are many variables that affect the complexity, such as the number of features used, the number of stages, etc. Furthermore, extra pre-processing is required in the Gabor classification technique, i.e. spectral analysis has to be performed to determine the location of the peaks while the other two methods do not require any pre-processing. The complexity of the Gabor classification method and the lower classification accuracy make it less attractive for practical purposes. The experiments indicated that the GLC based feature measures are superior to the Fourier-based methods. This agrees with the result of the comparison study reported in the literature (Weszka et al., 1976; Conners and Harlow, 1980). Furthermore, GLC based feature measures are insensitive to slight variation in spatial and in gray scale resolution which is due to normalisation. Another advantage of the GLC method over the Fourier method is that normalisation can be performed implicitly during the generation of GLC matrices by using different displacement for different image scale. Thus, no scale normalisation stage is required.

The experiments showed that the GLC and frequency domain based feature measures are sensitive to scale variation. The fact that the variance of intra-subclass patterns is small compared to the variance of intra-class patterns indicated that the main source of the large variance is the scale variation while the effect of noise, measurement error
and other factors are negligible. A simple size normalisation and interpolation technique can reduce the intra-class variance and consequently, improve the classification accuracy.

The classification accuracy for the nearest neighbour and Multi-layer Perceptron classifiers are at about the same level, but for practical reasons, Multi-layer Perceptron is better than the nearest neighbour classifier, since the classification can be accomplished almost instantaneously and it does not need a vast amount of storage.

The classification techniques developed in this thesis can be utilized in the following applications:

**Object Recognition**

Texture based object recognition can be conducted using the segmentation method developed in this thesis. This method however, has a number of limitations in that only objects with distinct textures can be classified. Many objects have similar textures, therefore, another method of recognition such as shape-based could complement this method.

**Road Tracking**

Road tracking for mobile robots can be accomplished based on the segmentation of the road and non-road sections. The segmentation can be performed based on the uniformity of the regions. A Gabor filter tuned to unique texture of the road can segment the road from the rest (trees, houses, etc.).

**Distance and Speed Prediction**

In this thesis the distance is known (measured) and is used for spectrum analysis. On the contrary, if spectrum of the surface texture of an object seen through the camera lens is known, then the distance can be predicted. This is because the spectrum changes in proportion to the distance. If time is measured during the movement of the camera as well, then the speed of the movement can be predicted.


A.1 INTRODUCTION

All the texture algorithms used in this thesis are translation invariant but not scale invariant. To obtained scale-invariant property, some kind of normalisation should be performed before features are derived. Since the viewing distance is known, it can be used to perform normalisation. The kind of normalisation adopted here is accomplished by scaling the subimage to the reference size. The scaling factor, $\alpha$, can be calculated as follows:

$$\alpha = \frac{d}{d_0}$$

where $d$ is the viewing distance of the image and $d_0$ is the reference distance.

A.2 LINEAR INTERPOLATION

The pixel value $g(m', n')$ of the transformed image is obtained from the original image $f(m,n)$ by using
\[ g(m', n') = f(\alpha m', \alpha n'), \quad \text{if } \alpha m' \text{ and } \alpha n' \text{ integer} \]

Interpolation need to be performed for non-integer \( \alpha m' \) or \( \alpha n' \). For two-dimensional linear interpolation or bilinear interpolation, \( g(m', n') \) is given by:

\[ g(m', n') = (1 - a)(1 - b)f(k, l) + (1 - a)bf(k, l + 1) + a(1 - b)f(k + 1, l) + abf(k + 1, l + 1) \]

where

\[
\begin{align*}
    k &= \lfloor \alpha m' \rfloor \\
    l &= \lfloor \alpha n' \rfloor \\
    a &= \alpha m' - k \\
    b &= \alpha n' - l
\end{align*}
\]

and \( \lfloor \cdot \rfloor \) represents the floor function.

In one of the GLC methods, the subimage is not scaled, only different displacements are used for different image scale. Suppose that \( \Delta x_0 \) and \( \Delta y_0 \) are the displacements used to calculate the GLC matrices from the reference image, the displacements for the other images can then be obtained from:

\[
\begin{align*}
    \Delta x &= \frac{\Delta x_0}{\alpha} \\
    \Delta y &= \frac{\Delta y_0}{\alpha}
\end{align*}
\]

where \( \alpha \) is the scaling factor defined in previous section.

If \( \Delta x \) or \( \Delta y \) is non-integer, interpolation is required. For bilinear interpolation, the gray level value \( f(m + \Delta x, n + \Delta y) \) is given by:
\[ f(m + \Delta x, n + \Delta y) = (1 - a)(1 - b)f(k, l) + (1 - a)bf(k, l + 1) + a(1 - b)f(k + 1, l) + abf(k + 1, l + 1) \]

where

\[ k = m + \lfloor \Delta x \rfloor \quad \text{and} \quad l = n + \lfloor \Delta y \rfloor \]

\[ a = \Delta x - \lfloor \Delta x \rfloor \quad \text{and} \quad b = \Delta y - \lfloor \Delta y \rfloor \]

and \( \lfloor \cdot \rfloor \) represents the floor function.
APPENDIX B
PATTERN CLASSIFICATION

B.1 INTRODUCTION

Pattern classification or pattern recognition is an essential part of any high level computer vision system. The process of pattern classification is mainly composed of two stages: a feature extraction stage, and a classification stage. In the feature extraction stage, the large dimensional pattern is transformed into a set of measurements which is called features. These features are usually expressed in the form of a vector \( X = [x_1, x_2, \ldots, x_n] \) where each vector component is a measured quantity that describes a characteristic of the pattern or object.

It is often useful to think of a pattern vector as a point in an \( n \)-dimensional Euclidean space \( E^n \). Figure B.1 shows two pattern classes, denoted by \( \omega_1 \) and \( \omega_2 \) which form two distinct clusters in two-dimensional feature space. Patterns belonging to the same class are often referred to as intra-class or within-class patterns and those which are not in the same class are called inter-class or between-class patterns. Intuitively, feature vectors computed from intra-class patterns should be close together in \( E^n \) while feature vectors computed from inter-class patterns should lie farther apart. This, however, is not usually the case in practical situations.
Representing an image with a set of numerical features (a feature vector) whose dimension is much lower than the original image data is a key step in the solution of any classification problem. This removes redundancy from the data and drastically cuts the computational cost of the classification stage. The most important criterion for the extracted features is that they must retain much of useful discriminability information present in the original data.

![Figure B.1 Two disjoint pattern classes](image)

Good features should satisfy two requirements: small intra-class variance and large inter-class separation. Small intra-class variance means that patterns with similar general characteristics should have features with numerically close values while large inter-class separation suggests that features from different classes should be quite different numerically. If a complete set of discriminatory features for each pattern class can be determined from the measured data, then the recognition and classification in the next stage would be easy.
In the classification stage, a proper class is assigned to the feature vector according to a decision rule. Assume that there are \( M \) different pattern classes, denoted by \( \omega_1, \omega_2, \ldots, \omega_M \), then the pattern space can be considered as consisting of \( M \) regions, each of which encloses the pattern points of a class. The recognition problem can now be viewed as that of generating the decision boundaries which separate the \( M \) pattern classes on the basis of the observed feature vectors.

There are well defined decision rules in pattern classification theory. If we know the a priori probabilities and the class conditional densities, we can design an optimal classifier using the Bayes decision theory. If the a priori probabilities and the class conditional density functions are unknown, but the forms for the underlying density functions are known, we can use two well known procedures, maximum likelihood estimation and Bayesian estimation to estimate the parameters of the density functions and carry out the classification. However, in most pattern recognition applications, the underlying statistical distributions of data or their functional forms are unknown. The common parametric forms rarely fit the densities actually encountered in practice. In these circumstances, non-parametric techniques that operate directly on the training data is better than parametric techniques. Some of these non-parametric techniques which will be discussed here are the nearest neighbour classifier and neural network classifier.

During the design phase of a pattern recognition system, the classifier has to be trained to recognise patterns. There are two different training methods: supervised and unsupervised. In a supervised learning environment, the system is "taught" to recognise patterns using representative patterns from each class. The essentials of this approach are a set of training patterns of known classification and implementation of an learning procedure. If only a set of training patterns of unknown classification is available, unsupervised pattern recognition techniques can then be applied.
**B.2 NEAREST NEIGHBOUR ALGORITHM**

The principle of the nearest neighbour algorithm is that of comparing input patterns $X = [x_1, x_2, \ldots, x_n]$, $x_i$ being the $i$-th feature, against a number of training patterns and then classifying the input pattern according to the class of the training samples that gives the closest match. Here a number of patterns are presented to the computer in the training phase of the algorithm then the test patterns are presented one at a time and compared against each of the training patterns.

The distance between $X$ and a training sample can be measured using Euclidean distance. This is a mapping from the $n$-dimensional feature space to a one-dimensional distance space. However, since the feature vector components have different dynamic ranges, it is possible for one or a subgroup of them to dominate the distance measure. To prevent this from happening, the distances need to be normalised. The normalisation consists of dividing by the standard deviations of the corresponding component of the class. Let $t_k = t_{k1}, t_{k2}, \ldots, t_{kn}$ denote the $k$-th element of the $n$-dimensional training feature vector, $k=1, \ldots, M$. The unknown test sample $X$ is classified to class $i^*$, where

$$i^* = \left\{ \text{class of } t_j \mid d(X, t_j) \leq d(X, t_k), \ k = 1, \ldots, M \right\}$$

and

$$d^2(X, t_k) = \sum_{m=1}^{n} \left\{ \frac{(x_m - t_{km})^2}{\sigma_{km}} \right\}$$

with $\sigma_{km}$ representing the sample standard deviation of the $m$-th element of the training feature vectors of class $s$, where $s$ is the class of $t_k$.
The nearest neighbour classifier is perhaps the simplest of all classifiers to implement on a computer. In addition, it has the advantage of being guaranteed to give an error rate within a factor of two of the ideal error rate obtainable with a Bayesian classifier (Duda and Hart, 1973). Another variation on the nearest neighbour technique is to assign a pattern vector to the class of the majority of its $k$ nearest neighbouring patterns from the training set. This method is known as $k$-NN method.

The nearest neighbour rule is superior to the $k$-NN rule if all the distances between patterns of a class are smaller than any distance between patterns of different classes (Tou and Gonzales, 1974). Both the nearest neighbour and the $k$-NN, however, have the disadvantage that they often require enormous storage to record enough training set pattern vectors and correspondingly large amounts of computation to search through them to find an optimal match for each test pattern. The computational complexity is in proportion to the number of patterns in the training set. The training set can be edited to remove pattern vectors that have little or no influence on the class assignment of an arbitrary pattern vector $x$ in order to reduce the number of calculations. These patterns are the ones located in the middle rather than the boundaries of a class region.

**B.3 NEURAL NETWORK CLASSIFIER** (Khotanzad and Lu, 1991)

Recent developments in the field of artificial neural networks (ANN) have provided potential alternatives to the traditional techniques of pattern recognition. An ANN is composed of many simple non linear computational elements operating in
parallel and arranged in pattern reminiscent of biological nervous systems. These non linear elements which are the building blocks of the network play the same role as the neurons in the brain and thus are usually called neurons or nodes. The nodes are interconnected via weights that can be adapted and changed according to a given situation, analogous to synaptic connections to neurons in the brain. It should be noted that while ANNs are inspired from studies of biological systems, they are far from mimicking functions of the human brain. It is only inspired by the tremendous potential, highly parallel operation, and fault tolerant nature of the brain and is not constrained by the exact details.

ANNs are capable of many functions, among them optimisation, clustering, mapping, and classification. In this study, the net is utilised in the context of a supervised classifier. This is a decision making process which requires the net to identify the class or category which best represents an input pattern. The net is adapted to the classes it is expected to recognise through a learning process using labelled training prototypes from each category.

The neural network classifier used in this study is Multi-Layer Perceptron (MLP). A MLP is a feed-forward net with one or more layers of nodes between the input and output nodes. These in-between layers are called hidden layers. A MLP with one hidden layer is shown in figure B.2. Each node in a layer is connected to all the nodes in the layer above it. The outputs of nodes in one layer are transmitted to nodes in another layer through links that amplify or attenuate or inhibit such outputs through weighting factors. Except for the input nodes, the net input to each node is the sum of the weighted outputs of the nodes in the prior layer. Each node is activated in accordance with the input to the node, the activation function of node, and the bias of the node.
Training is equivalent to finding proper weights for all the connections such that a desired output is generated for a corresponding input. Using MLP in the context of a classifier requires all output nodes to be set to zero except for the node that is marked to correspond to the class the input is from. That desired output is one. The input to this net consists of the features extracted from the image and the output is the class label of the input.

![Multi-Layer Perceptron](image)

**Figure B.2** Multi-Layer Perceptron with one hidden layer

The training algorithm used is Back-Propagation Algorithm which is also called Generalised Delta Rule. According to this algorithm, for each pattern in the training set, learning proper weights is conducted by computing the discrepancy between the desired and actual outputs and feeding back this error signal level-by-level to the inputs, changing the connection weights in such a way as to modify them in proportion to their responsibility for the output error. The output of this training process is a set of weights and biases that will satisfy all (input, output)
pairs presented to it. The major steps of the algorithm are as follows: (Khotanzad and Lu, 1991)

1. Initialise all $w_{ij}$s to small random values. $w_{ij}$ is the value of the connection weight between node $j$ and node $i$ in the layer below.
2. Present an input from class $m$ and specify the desired output. The desired output is 0 for all the output nodes except the $m$-th node which is 1.
3. Calculate actual outputs of all the nodes using the present value of $w_{ij}$s. The output of node $j$, denoted by $y_j$, is a non-linear functions of its total inputs:

$$y_j = \frac{1}{1 + \exp(-\sum_{i} y_i w_{ij})}$$

This particular non-linear function is called a "Sigmoid" function.

4. Find an error term, $\delta_j$, for all the nodes. If $d_j$ and $y_j$ stand for desired and actual values of a node respectively, for an output node

$$\delta_j = (d_j - y_j)y_j(1-y_j)$$

and for a hidden layer node

$$\delta_j = y_j(1-y_j)\sum_k \delta_k w_{jk}$$

where $k$ is over all nodes in the layer above node $j$.

5. Adjust weights by

$$w_{ij}(n+1) = w_{ij}(n) + \alpha \delta_j y_i + \gamma (w_{ij}(n) - w_{ij}(n-1))$$
where \((n+1)\), \((n)\), and \((n-1)\) index next, present, and previous respectively. \(\alpha\) is a learning rate similar to step size in gradient search algorithms. \(\gamma\) is a constant between 0 and 1 which determines the effect of past weight changes on the current direction of movement in weight space. This provides a kind of momentum that effectively filters out high frequency variations of the error surface.

6. Present another input and go back to step 2. All the training inputs are presented cyclically until weights converge.

The training features are normalised to have zero mean and unit variance before being input to the MLP. This is necessary in order to ensure that a subgroup of the features do not dominate the weight adjustment process during training. The \(m\)-th feature is normalised by

\[
\tilde{t}_m = \frac{t_m - \bar{t}_m}{\sigma_m}
\]

where \(\tilde{t}_m\) and \(\sigma_m\) are the sample mean and standard deviation of the \(m\)-th features of all training samples. During the testing phase each test sample is also normalised in a similar manner.

The main computational requirement of an ANN is during its training phase which can usually be performed off-line. The main difficulty with the MLP is the absence of a systematic method for selection of the appropriate number of hidden nodes. The other problem is the slow rate of convergence of the back-propagation learning rule and the fact that it does not guarantee a global minimum when searching for the best weights.
B.4 CLASSIFICATION SCHEME

There are two classification schemes that can be employed. If the number of samples in the database set is small, then the *leave-one-out* classification method can be applied. In this method, all samples but one in the database are used as the training set and the sample left out is classified. Next, the test sample, i.e., the one left-out is reinserted into the database with another sample selected from the database for classification. This process is repeated until all the samples in the database have been treated as the unknown and classified.

If the number of samples in the database is large enough, then the second method should be employed. In this method, the database is divided into two parts: training database and test database. Training is performed using the samples in the training database and classification is performed using the test database. Thus, in both cases, classification is performed on samples which have never been used in the training.