Statistical Approaches To Texture Classification

Manik Varma
Jesus College

Robotics Research Group
Department of Engineering Science
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2004

Abstract

This thesis investigates the problem of classifying textures from their imaged appearance without imposing any constraints on, or requiring any *a priori* knowledge of, the viewing or illumination conditions under which the images were obtained. Classification algorithms based on the statistical distribution of texton primitives are developed to categorise single, uncalibrated images into a set of pre-learnt material classes.

The thesis starts by introducing a filter bank based approach to the problem of texture classification. We design low dimensional, rotation and scale invariant filter sets which are nevertheless capable of extracting rich features at multiple orientations and scales. Textures are modelled by the frequency distribution of exemplar filter response features. Characterising a texture by multiple models allows the classification of single images without requiring any knowledge of the imaging conditions. Using this framework, it is demonstrated that the new filter sets achieve superior performance as compared to their traditional counterparts when benchmarked on real world databases conatining many classes with significant imaging variations.

There are two major approaches to building texture classifiers based on filter responses. One approach, motivated by Psychophysics, is to first determine the texton primitives of a material, next model the texture by their frequency distribution and finally classify novel images by nearest neighbour matching. An alternative is offered by the more statistical, Bayesian paradigm which recommends learning the joint probability distribution of filter responses followed by MAP classification. We show that both approaches are essentially the same and that they can actually be made equivalent under suitable choices of PDF representation and similarity measure.

The issue of whether filter banks are necessary for material classification is addressed next. A novel texture representation is developed based on the joint probability distribution of pixel intensities in compact neighbourhoods. Using this representation within the standard classification framework leads to two astonishing results: (a) very small neighbourhoods can yield superior performance as compared to multi-scale, multi-orientation filter banks with large support and (b) the performance of filter banks is always inferior to the new representation with equivalent neighbourhood size. Theoretical arguments are presented as to why these two results might hold.

Finally, the related problem of determining the illuminant's direction from textured images is explored. A theory for estimating the illuminant's azimuthal angle from images of Lambertian, rough surfaces with spatially varying albedo is formulated. In certain cases, the theory is able to accommodate the effects of non-Lambertian factors such as shadows, specularities, inter-reflections, etc. This is evidenced by the good results achieved on numerous real world images which deviate strongly from the ideal assumptions.

Acknowledgements

I would like to begin by acknowledging the guidance, support and generosity of my supervisor Prof. Andrew Zisserman whose immense, and infectious, zest for the subject has constantly motivated me to learn more and do better. His mentorship over the course of this thesis has been invaluable and his sense of humour has always kept things in perspective.

I am also grateful to my many peers and colleagues who have contributed to my research. I would particularly like to thank Frederik Schaffalitzky and Alexey Zalesny for numerous discussions and some very valuable feedback. I would also like to thank Oana Cula, Thomas Leung and Cordelia Schmid for supplying details of their filter banks and algorithms. Discussions with David Forsyth and Bill Triggs helped shape some of the chapters while Mike Chantler, Antonio Criminisi and Alan Yuille provided the Heriot-Watt TextureLab, Microsoft Textile and San Francisco databases respectively. I am also very grateful to Mike Brady and Alan Yuille for their advice.

Equally important has been the support that I've received from all my VGG and LAV lab mates, both past and present. Amongst them, I would especially like to thank Josef and Timor for all their help and Teo, Walterio, Alyosha and Manjari for their friendship. I would also like to acknowledge David, Phil, Sach and Manjari for proof reading various chapters at such short notice.

Financial support for this research was provided by a University of Oxford Graduate Scholarship in Engineering at Jesus College, a Rhodes Scholarship, an ORS award and the EC project CogViSys. I am indebted to all the funding bodies involved for providing me the opportunity to study at Oxford and particularly to my Department and College for making the stay a pleasant one.

Finally, I would like to thank my parents for their love and support, and for the many sacrifices that they've had to make. My family away from home, Miguel and Gareth. And, of course, I must never ever forget my wife Sachi. Not only is she the most intelligent and beautiful person I know but she is also the one who writes my acknowledgements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A perusal of the texture analysis literature quickly reveals that most books and theses begin with one of two standard ways of introducing the concept of visual texture. The first is to cull many definitions of texture from dictionaries and other sources before concluding that none of them are satisfactory or amenable to precise mathematical modelling. The second is to exemplify the concept with a variety of images but then leave the reader to draw his own definition. We take a slightly different approach for the purposes of this thesis. Instead of attempting a universally applicable definition ourselves, we start by looking at some of the problems that motivate interest in visual texture analysis from a computational or algorithmic point of view.

One such important problem crops up when an automated system has to inspect a woman's mammogram and analyse the breast tissue to predict her risk of developing breast cancer (see figure 1.1a). A somewhat different, but closely related, problem is to first determine the region of the mammogram occupied by the breast tissue and segment it from the pectoral muscle and the background. A very different type of problem arises during the process of digital image restoration where damaged, missing or unwanted bits of an

(a) (b) (c) (d)

Figure 1.1: Some of the computational problems which motivate the study of visual texture: (a) Where is the breast tissue in this mammogram as opposed to the pectoral muscle and the background? Does the patient have a high or low risk of developing breast cancer in the future? (b) Removing an unwanted person from a scene, (c) Compressing an image of a fingerprint and (d) Determining the 3D shape of a person from her clothing.

image have to be seamlessly replaced by their backgrounds. For example, in figure 1.1b one might like to remove the person blocking the view and replace her by the textured scenery. Another important problem is the compression of fingerprint images (figure 1.1c) so that the FBI can store its very large database in an efficient and practical manner. The determination of a person's 3D shape from the clothing that she's wearing presents yet another challenge (figure 1.1d).

Each of these problems can be tackled successfully by exploiting the information conveyed by the textural content in the images. Keeping this in mind, Computer Vision research into texture has been divided into the canonical areas of classification, segmentation, synthesis, compression and shape from texture. It should be noted that the types of textures under consideration can range from the purely stochastic to the completely structured and everything in between. Furthermore, the term texture can have slightly different connotations in each of the areas depending on the objective. Therefore, to get a feel of visual texture and the different ways in which it is used and

defined we now briefly touch upon the five canonical areas. In addition, the psychophysics of texture perception is also discussed. We start with texture classification which is the primary concern of this thesis.

1.1 Texture classification

The canonical texture classification task is to design an algorithm for categorising previously unseen images as belonging to one of a set of known materials of which training examples have been provided. Particular instances of the problem arise depending on how much training data is available, what properties it has and how it's related to the novel images intended for classification. More recently, classification has also come to refer to the simultaneous localisation and categorisation of the textures of interest in an image

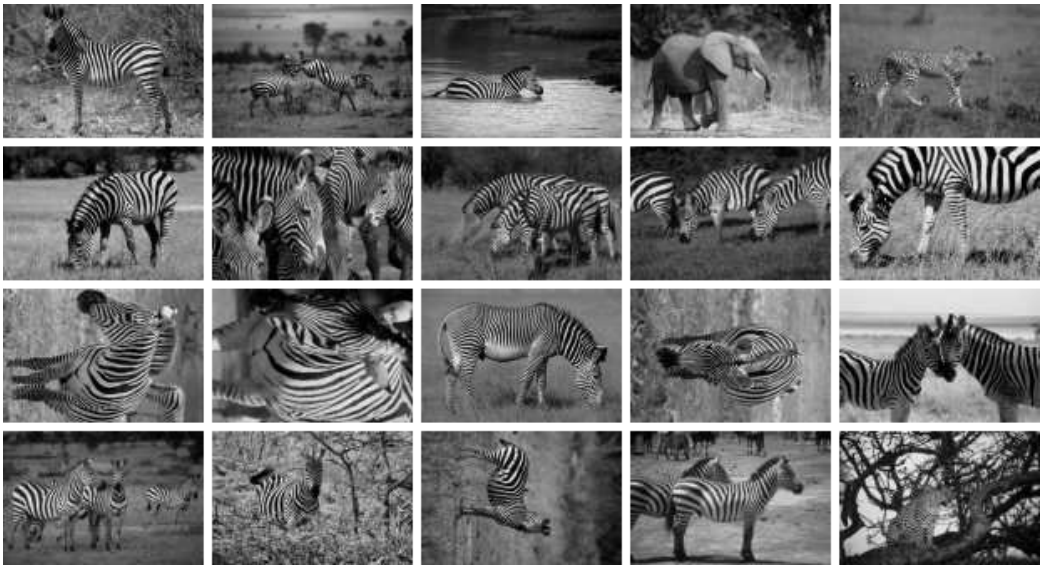


Figure 1.2: Using texture features for content based image retrieval: In the top row are some sample training images, both positive and negative, used to learn the zebra texture. The subsequent rows show the top 15 images retrieved from a subset of the COREL dataset which match the zebra model. Results from [Schmid, 2001].

(detecting the presence and location of the zebras in figure 1.2 for example).

Texture classification has applications in many areas. Within Computer Vision itself, it could be used in the problem domains of object recognition and content based image retrieval. For example, the performance of existing shape based object detection systems could be enhanced by providing them with a textural description of the scene. Similarly, as figure 1.2 illustrates, images in large databases could be automatically annotated with a list of textures present in them for archiving and retrieval purposes [Lazebnik et al., 2003b, Manjunath and Ma, 1996, Schmid, 2001, Xu et al., 2000]. An extension of this idea is to present texture classification as a tool to art historians for determining the different fabrics being worn by people in paintings and art works [Savarese and Criminsi, 2004].

Moving on a bit further, texture classification lends itself to applications in Medical Image Analysis. It has been used to screen women for early signs of breast cancer by classifying parenchymal density and detecting microcalcifications [James et al., 2001, Miller and Astley, 1992, Petrick et al., 1996, Petroudi et al., 2003]. It has also been used to diagnose pulmonary diseases [Sutton and Hall, 1972] as well as leukaemia [Harms et al., 1986].

Even fields such as Remote Sensing have relied on texture classification to automate processing [Brady, 2003, Haralick et al., 1973, Paget, 1999, Schistad and Jain, 1992, Solberg and Jain, 1997, Rellier et al., 2004]. As figure 1.3 shows, terrain types appear as statistically textured regions in remotely sensed images and can be classified as such [Lorette et al., 2000, Ruiz et al., 2004, Weszka et al., 1976]. This is very important for monitoring land use patterns and seeing how they evolve over time. For example, texture classification could be used to monitor the deforestation of tropical rainforests from SAR images [Kuntz et al., 1999, Miranda et al., 1998, Podest and Saatchi,

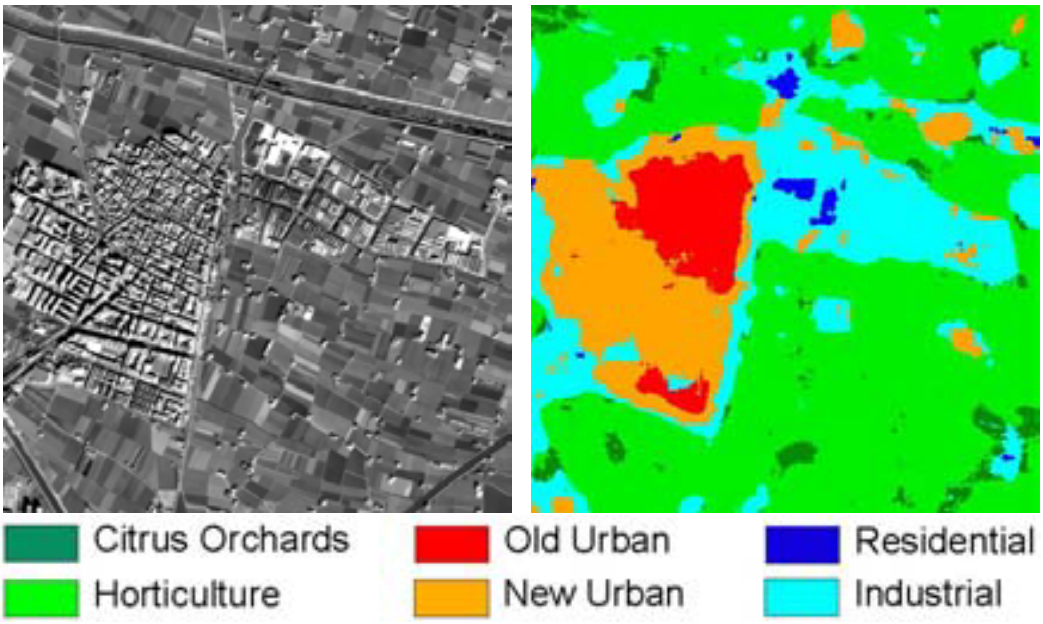


Figure 1.3: Applications in Remote Sensing: Texture classification of a detail image of an urban area. Results from [Ruiz et al., 2004].

2002, Simard et al., 2000]. It can also be used to study global warming by monitoring glaciers and ice formation [Barber and LeDrew, 1991, Deng and Clausi, 2003]. Another important application is to be able to automatically detect oil slicks and other types of pollution in the sea so as to aid in their clean up [Benelli and Garzelli, 1999].

New applications of texture classification are also being found in the areas of automated inspection, defect detection and quality control [Chetverikov and Hanbury, 2002, Cuenca and Camara, 2003, Newman and Jain, 1995, Song et al., 1992]. Typical examples include the detection of defects in images of textiles [Bodnarova et al., 2000, Mamic and Bennamoun, 2000, Ozdemir et al., 1998], wood [Conners et al., 1990] and leather and assessing the quality of carpets [Siew et al., 1988], machined surfaces, steel [Wiltschi et al., 2000] and automotive finishes [Jain et al., 1990]. Figure 1.4 shows how texture classification can be used to monitor rusting on steel girders. Such applications are

Grade A rusting Grade B rusting Grade C rusting

Figure 1.4: Applications in automated inspection: Texture classification can be used to assess the degree of rusting of steel girders. Image courtesy of [Unsalan and Ercil, 1999].

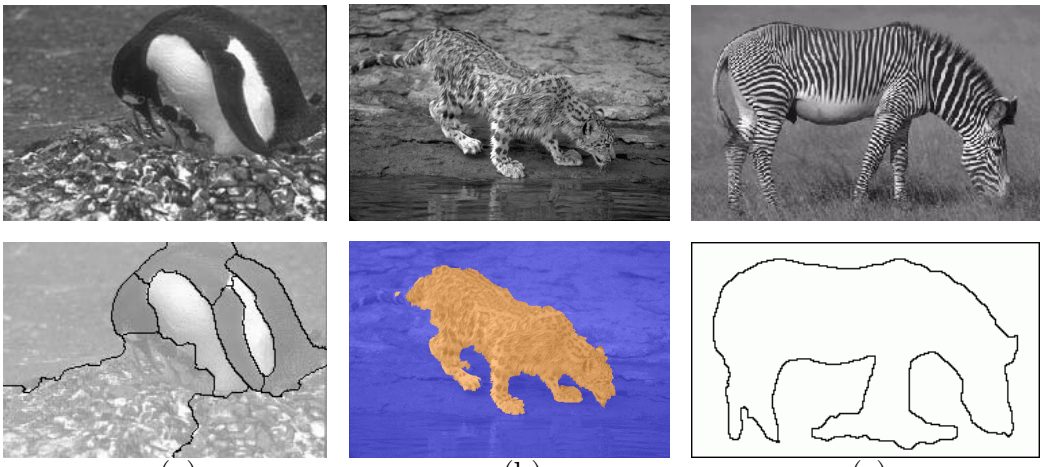
critical to the manufacturing industry due to the high cost associated with human inspection. Successful automation of such tasks can therefore have a significant impact on the industry.

Thus, texture classification finds uses in many diverse areas and the goal of this thesis is to develop algorithms which can be successfully applied to problems in them.

1.2 Texture segmentation

The goal of texture segmentation is to partition a given image into disjoint regions of coherent texture [Derin and Elliot, 1987, Galun et al., 2003, Jain and Farrokhnia, 1991, Krishnamachari and Chellappa, 1997, Lee et al., 1992, Malik et al., 2001, Paragios and Deriche, 2002, Sandberg et al., 2002, Tu and Zhu, 2002, Weldon and Higgins, 1996, Xie and Brady, 1996, Zhu and Yuille, 1996]. The task can either be supervised, in which case *a priori* information about the texture classes one expects to see in the image is available and can be made use of, or unsupervised so that pixels and regions have to be

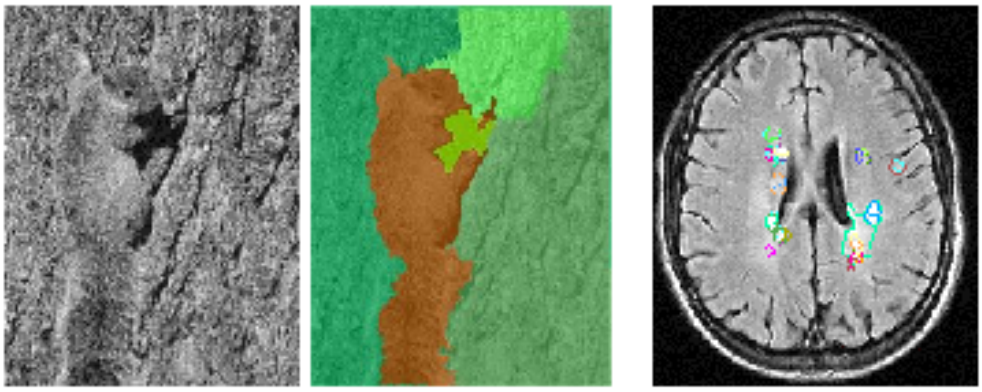
grouped together by some measure of perceptual similarity. Texture can also be combined with other cues, such as contour information, in order to segment generic images. Figure 1.5 shows some illustrative examples.



(a) (b) (c)

Figure 1.5: Segmentation examples: (a) [Ren and Malik, 2003], (b) [Galun et al., 2003] and (c) [Tu and Zhu, 2002].

As can be seen, segmentation in itself is not always a well defined task since an image might have many "correct" segmentations. Nevertheless, segmentation can be a powerful tool leading on to classification and recognition. For instance, [Mori et al., 2004] show how segmentation can be used as a pre-processing stage to recognise limbs and thereby detect human beings and recover their body pose. Similarly, many of the texture classification applications listed in the previous section first rely on segmentation methods to demarcate regions of interest before classifying them. As an example, segmentation can be used to break the camouflage of hidden objects and the detected regions can then be classified. Similarly, in Medical Image Analysis, segmentation can be used to detect abnormal tissue patterns such as multiple sclerosis lesions in the brain. Both tasks are illustrated in figure 1.6. There are many other applications of being able to segment figure from ground



(a) (b)

Figure 1.6: Applications of texture segmentation: (a) breaking the camouflage of hidden objects and (b) detecting multiple sclerosis lesions in the brain. Image courtesy of [Galun et al., 2003].

including applications in graphics, such as image editing and background substitution [Rother et al., 2004], and document processing, such as extracting printed text regions [Jain and Bhattacharjee, 1992].

1.3 Texture synthesis

The goal of synthesis is to compose an output image in the form of a specified target texture [Ashikhmin, 2001, De Bonet, 1997, Efros and Leung, 1999, Heeger and Bergen, 1995, Popat and Picard, 1993, Portilla and Simoncelli, 2000, Zhu et al., 1998]. One instance of the problem is to simply synthesise more of the target. The resultant output must be perceptually similar to the target but not identical. This is illustrated in figure 1.7. A variant of the problem is to synthesise the target from a novel view or under a different illumination [Zalesny and Van Gool, 2000] or even map the target onto a different surface [Dana et al., 1999].

Texture synthesis has many interesting applications. It can be used to

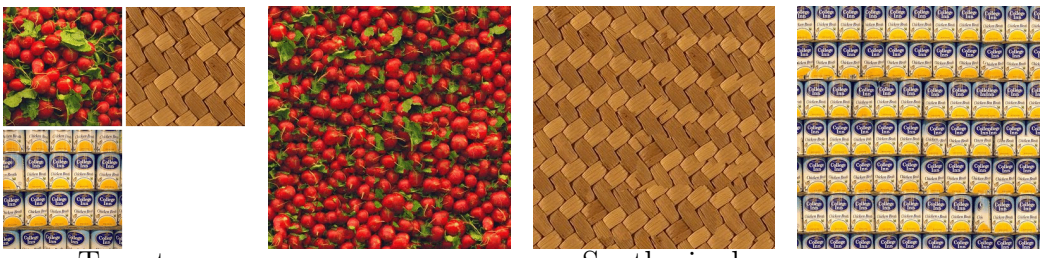


Figure 1.7: The texture synthesis problem: Given target textures on the left, the task is to synthesise similar, but not identical, output images such as the ones on the right. Results from [Efros and Freeman, 2001].

synthesize high resolution texture detail in low resolution images [Hertzmann et al., 2001, Pickup et al., 2003, Zalesny and Van Gool, 2000]. It can also be combined with inpainting to fill in holes left by deleting large unwanted objects in images [Criminisi et al., 2004]. A texture from one image can be synthesised onto another image to effect texture transfer [Efros and Freeman, 2001, Hertzmann et al., 2001] and various painting filters can be learnt so as to render new content in a different style [Drori et al., 2003, Hertzmann et al., 2001]. Some of these applications are illustrated in figure 1.8.

While this thesis is not explicitly concerned with texture synthesis, studying the problem can be instructive as it is closely intertwined with texture classification. For example, one way of classifying textures is via the analysis-by-synthesis route in which a model is first constructed for synthesizing textures and then inverted for the purposes of classification. Conversely, a texture classification algorithm which is capable of determining the probability with which an image or texture patch belongs to a particular class will find many applications in synthesis. For instance, it could be used to provide a termination criteria for synthesis algorithms which mainly rely on human intervention at the moment. Furthermore, it could be used to automatically determine the quality of a synthesized image. In a sense, the two problems

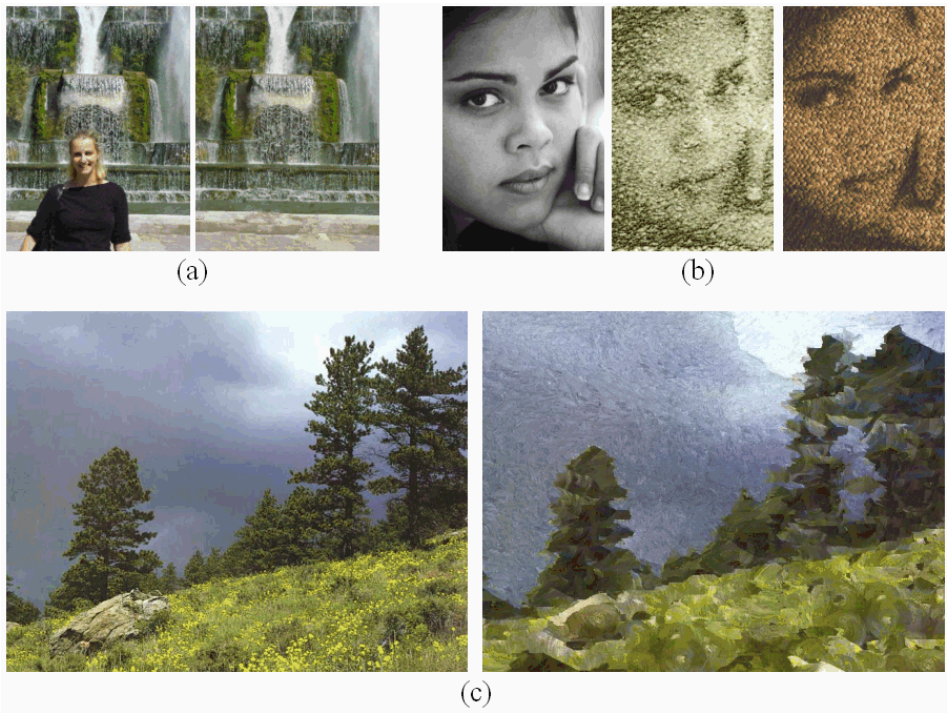
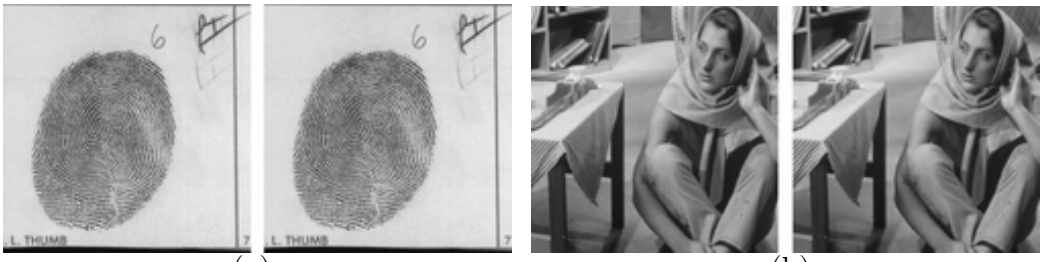


Figure 1.8: Applications of texture synthesis: (a) removing unwanted objects [Criminisi et al., 2004]. (b) texture transfer where the original textured image (left) has been resynthesised so as to look like a carpet (middle) and rug (right) [Hertzmann et al., 2001], (c) artistic filtering where an original painting is resynthesised in a different style [Hertzmann et al., 2001].

can be thought of as duals, since a solution to one will quickly lead to a way of validating the other.

1.4 Texture compression and coding

Texture compression schemes aim to minimise the amount of data required to store a textured image [Bradley et al., 1993, Chai et al., 1999, Chellappa et al., 1985, Li et al., 1995, Li et al., 2000, Meyer et al., 2000, Popat and Picard, 1993]. In general, there are two approaches to compression, lossless and lossy. Most algorithms for texture compression tend to be lossy and



(a) (b)

Figure 1.9: Texture compression: (a) the original fingerprint image is on the left while the image on the right has been compressed more than 18 times without much compromise in quality. The compression rate is 0.444 bits/pixel down from 8 bits/pixel for the original image and the PSNR value is only 35.65 dB. Similarly for (b), the original image is on the left while the image on the right has been compressed to 0.5 bits/pixel with PSNR 31.89 dB. Results from [Chai et al., 1999].

trade off fidelity for higher compression ratios (see figure 1.9). In fact, for certain applications, texture synthesis methods based on compact models can be thought of as providing excellent compression which is very "lossy" yet where the "compressed" image is perceptually equivalent to the original. It should be noted that texture compression algorithms are not the same as standard image compression algorithms such as JPEG. The former are specifically designed to exploit the statistical structure of textured images and to outperform image compression algorithms in this particular domain. For instance, the WSQ texture compression algorithm of [Bradley et al., 1993] is good at compressing fingerprints and, at the same compression ratio as JPEG, produces a much more accurate image which has all the important high frequency signals while getting rid of the typical blocking artifacts that plague JPEG. Figure 1.9a shows an example of a fingerprint image compressed to less than an eighteenth the size of the original without any significant loss of detail.

Compression algorithms also have a link to segmentation and boundary

detection [Froment and Mallat, 1992, Kocher and Kunt, 1983, Ran and Farvardin, 1992]. Typically, in a natural image, people pay more attention to foreground objects rather than the background (which can often be out of focus). Therefore, if a segmentation algorithm can extract the foreground object then more bits can be used to represent it as compared to the background. Similarly, people are sensitive to edges caused by object boundaries. Hence, different compression schemes can be applied to the object's shape or contour and the textured region in the interior.

The main applications of compression arise when large amounts of data needs to be stored or transmitted. For instance, most multimedia applications such as digital encyclopedias, videos and games as well as databases of astronomical and remotely sensed images rely heavily on compression techniques to manage the data. The FBI also uses texture compression to store its large database of fingerprint images. Similarly, applications such as video telephony and teleconferencing are only made possible by fast and efficient compression schemes.

1.5 Shape from texture

Textures provide a powerful shape cue to humans and the goal of shape from texture is to recover the 3D shape of a textured object from its image [Blake and Marinos, 1990,Clerc and Mallat, 2002,Lindeberg and Gårding, 1993,Knill, 2001,Lobay and Forsyth, 2004,Malik and Rosenholtz, 1997,Todd and Oomes, 2002, Witkin, 1981]. Most shape from texture algorithms work on the assumption that the patterns of texture on a surface are regular and any deformations visible in an image are due to surface geometry (see figure 1.10). Shape from texture algorithms come in many flavours and tackle

different aspects of the problem. There are those that assume orthographic versus perspective projection, those that assume that the underlying physical surface is planar or curved, those that focus on individual texture elements versus their statistics (such as isotropy or homogeneity) and therefore those that recover global surface models or only local estimates of the differential geometric parameters at points on the surface.

Figure 1.10: Recovering the 3D shape of clothing using a shape from texture technique. Results from [Lobay and Forsyth, 2004].

As [Forsyth, 2002] notes, shape from texture has tended to be "a core vision problem" without many immediate practical applications. However, some of the real world problems to which shape from texture has been applied include defect detection of materials on inclined planes [Plantier et al., 2002] and image based rendering of clothing [Lobay and Forsyth, 2004].

1.6 Psychophysics and Neurobiology

Results from Psychophysics and Neurobiology have had a great influence on many aspects of Computer Vision research into texture including classification and segmentation [Beck, 1983, Beck et al., 1987, Bergen and Landy, 1991, Bergen, 1991, Gurnsey and Browse, 1987, Julesz et al., 1973, Julesz, 1981, Malik and Perona, 1990], synthesis [Heeger and Bergen, 1995, Portilla and Simoncelli, 2000, Zhu et al., 1998] and shape from texture [Cumming

et al., 1993, Gibson, 1950, Knill, 2001, Li and Zaidi, 2001, Rosenholtz and Malik, 1997, Todd et al., 2004].

Julesz maintained that "in all texture perception the preattentive system is dominant" and his psychophysical experiments laid much of the foundation for the research that followed in segmentation and classification. He demonstrated that the process of texture discrimination by humans is very different from that of form recognition and hypothesized that textures can be discriminated pre-attentively (in less than 100 ms) by only looking at the first and second order statistics of pixel intensities [Julesz, 1962, Julesz et al., 1973]. Furthermore, he conjectured that any two textures with the same third order statistics, i.e. identical distributions of the co-occurrence of intensity triples, would not be effortlessly distinguishable. The conjecture was ultimately proved wrong [Caelli and Julesz, 1978] but lead to the alternative hypothesis that texture discrimination by the preattentive visual system is based on the density of texture primitives called textons, but not on their spatial or positional relationships [Julesz, 1981, Julesz and Bergen, 1983]. This view has proved to be very influential, and while the definition of textons might have evolved over the years, many modern day texture classification algorithms [Cula and Dana, 2004, Hayman et al., 2004, Leung and Malik, 2001] still follow the basic paradigm of discrimination by looking at the differences in the first order texton statistics.

Neurobiology and Psychophysics have also played a major role in identifying what features should be extracted from images for texture analysis [Beck et al., 1987, Caelli and Moraglia, 1985, Daugman, 1985, Fogel and Sagi, 1989, Malik and Perona, 1990, Turner, 1986]. Many studies have concluded that the centre-surround and simple cells found in the mammalian visual system can be effectively modelled by Gabor or Gaussian derivative

filters. As such, filter banks of Gabors and Gaussians at multiple scales, orientations and phases are frequently used to extract texture features. However, [Olshausen and Field, 2005] caution that much is still unknown about the primary visual cortex and future theories might move away from the current interpretation of V1 as being a "Gabor pyramid".

The study of human visual perception has also influenced other areas of texture research. Inspired by Julesz's conjecture, [Portilla and Simoncelli, 2000] developed a minimal statistical texture description and tried to make it consistent with human visual perception so as to use it for synthesis. In segmentation, the Gestalt cues of proximity, similarity, closure and simplicity have often been used as guiding principle while designing algorithms [Reed and Wechsler, 1990,Ren and Malik, 2003,Wertheimer, 1958]. Finally, numerous psychophysical studies have investigated how various aspects of surface texture influence an observer's perceptions of the 3D shape of objects. Using such psychophysical evidence, [Sakai and Finkel, 1994] propose a network model for recovering shape from texture based on spatially averaged peak frequencies. More recently, [Todd et al., 2004] conducted experiments to show that the constraints assumed by most algorithms are too severe and that human perception of shape is much more robust than previously assumed.

1.7 Problem statement: Texture classification

The form of the texture classification problem addressed in this thesis is the categorization of materials on the basis of their appearance in *single* images (see figure 1.11). Furthermore, no constraints are imposed on the acquisition of the training or novel images and, in particular, no *a priori* knowledge of their viewing or illumination conditions is required. Finally, in order to

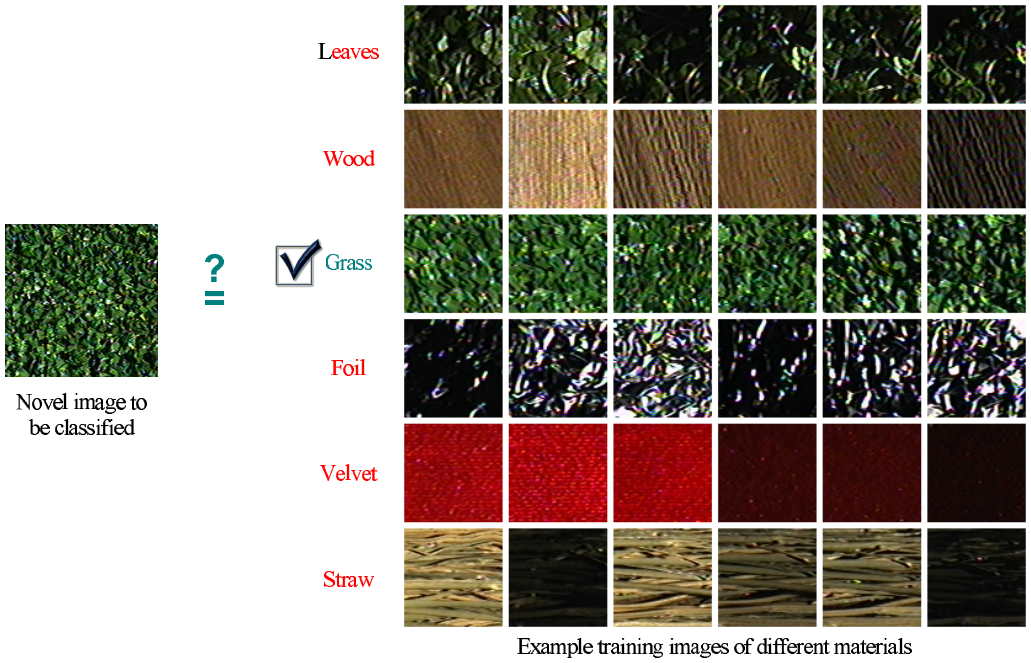


Figure 1.11: The classification problem addressed in this thesis: Given a single, uncalibrated, previously unseen image of a textured material, categorise it into one of a set of pre-learnt classes.

make the problem as general as possible, no use is made of colour information whatsoever to assist classification. A material's colour is an additional and distinct source of information as compared to its texture. If its use was felt necessary in a particular application then the algorithms developed in this thesis could be extended to incorporate colour into the classification scheme. Though, it should be noted that while colour provides a very strong cue for discrimination, it can also be misleading due to the colour constancy issue [Funt et al., 1998].

Since almost no restrictions have been placed on the problem as stated, its scope is broad and it finds many applications in diverse fields. However, this also implies that the problem is exceedingly hard. In particular, what makes the problem so difficult is that unlike other forms of classification, where the

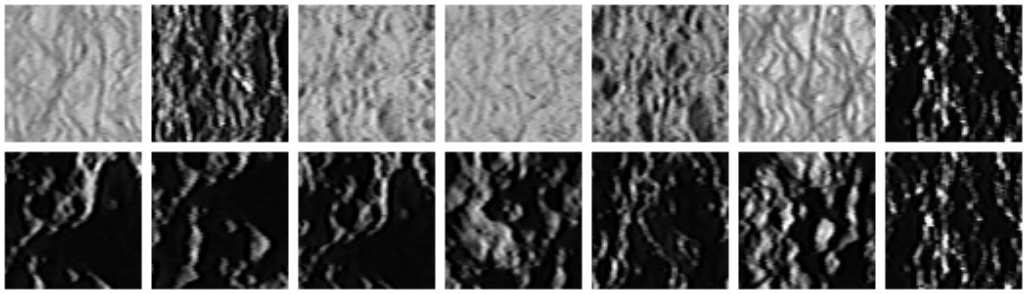
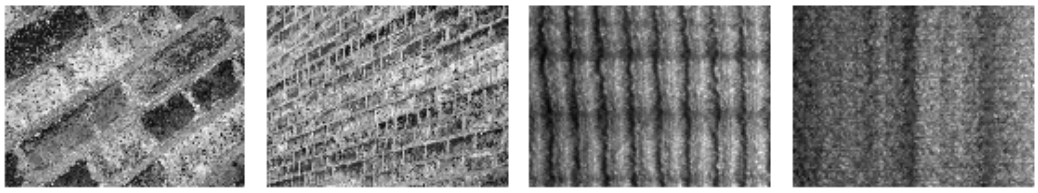


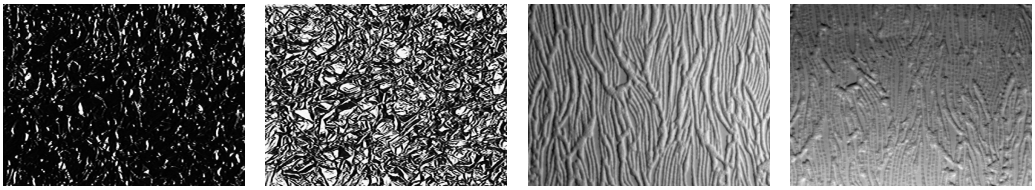
Figure 1.12: The change in imaged appearance of the same texture (Plaster B, texture # 30 from the Columbia-Utrecht database) with variation in imaging conditions. Top row: constant viewing angle and varying illumination. Bottom row: constant illumination and varying viewing angle. There is a considerable difference in the appearance across images.

objects being categorised have a definite structure which can be captured and represented, most textures have large stochastic variations which make them difficult to model. Furthermore, textured materials often undergo a sea change in their imaged appearance with variations in illumination and camera pose (see figure 1.12). For instance, keeping all the parameters fixed but just changing the scale or the rotation can result in a completely new texture with new descriptors and different statistics. This is illustrated in figure 1.13. The variation can be just as large if the illumination is changed. As figure 1.14 demonstrates, changing just the illuminant's elevation or azimuthal angle



Change in viewpoint Change in surface orientation

Figure 1.13: The effect of scale and rotation on textures. On the left, the camera was moved away from the brick wall. On the right, the surface was rotated in plane by 90° while all other imaging conditions were left unchanged.



Change in elevation Change in azimuth

Figure 1.14: The effect of change in illumination direction. All parameters are kept fixed except, on the left, the illuminant's elevation is changed while on the right the azimuth is varied.

can have a marked impact on the appearance of a texture [Chantler, 1994]. Dealing with these variations successfully is one of the main tasks of any classification algorithm.

Another factor which comes into play is that, many a time, two materials when photographed under very different imaging conditions can appear to be quite similar, as is illustrated by figure 1.15. It is a combination of all these factors which makes the texture classification problem so challenging.

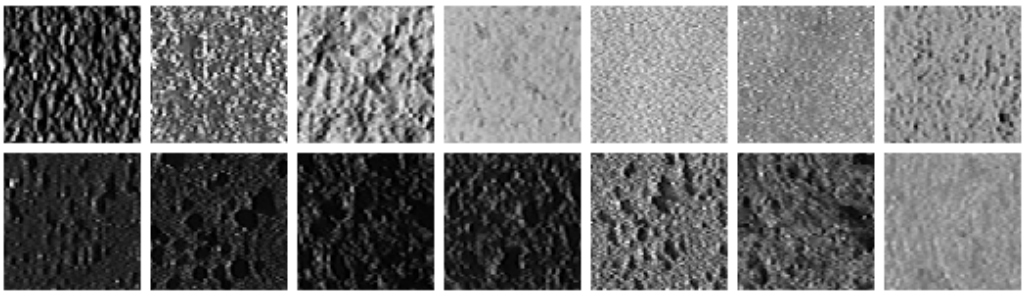


Figure 1.15: Small inter class variations between textures can make the problem harder still. In the top row, the first and the fourth image are of the same texture while all the other images, even though they look similar to at least one other image, belong to different classes. Similarly, in the bottom row, the images appear similar and yet there are three different texture classes present.

1.8 Thesis outline and novelty

The remainder of the thesis is organised as follows. Chapter 2 presents a survey of the literature and places the current versions of the classification and synthesis problems in a historical perspective. We review how Julesz had initially postulated textures to be the distribution of fundamental primitives such as line terminators, crossings, intersections, etc. and how this definition has now evolved to become the distribution of filter responses computed at multiple orientations and scales. We also review how the texture classification problem has advanced from segregation of binary patterns to the classification of grayscale images of synthetic 2D textures to the classification of real world 3D materials. The prominent role that filter banks have played during this evolution is charted and some of the more relevant literature discussed in detail. An overview is also presented of the different databases used in this thesis to measure classification performance.

The state of the art for classifying materials photographed under real world conditions is the algorithm of [Leung and Malik, 2001]. While the algorithm is a major improvement on previous work, it also has some severe limitations. In particular, the algorithm requires multiple registered images, with known viewpoint and illumination, both during training and classification. Furthermore, it is not robust to viewpoint changes (such as those due to rotation or scaling) and operates is a very high dimensional space.

Chapter 3 introduces a filter bank based solution to the problem of real world texture classification which overcomes all these limitations. The emphasis in this chapter is therefore on two points: (a) coming up with a basic framework to classify single, uncalibrated images of materials and (b) designing low dimensional, highly selective yet invariant filter banks so as to reduce the number of models needed to characterise texture classes. The per-

formance of the new filter banks is empirically compared to traditional filter sets by classifying all 61 materials present in the Columbia-Utrecht database. It is demonstrated that the new filter banks outperform their more traditional counterparts (including those of Leung and Malik), especially when having to cope with imaging variations due to rotations.

Chapter 4 then extends this basic classification framework. Two techniques for model reduction are developed and compared to state of the art methods in the field. It is shown how the number of models can be reduced to a handful without impairing classification performance. The effect of varying the parameters of the algorithm are also studied including the effects of size of texton dictionary and number and choice of training images. Finally, we validate the conjecture that the first order statistics of textons are sufficient for classification and experimentally verify on the Columbia-Utrecht database that no significant information is being lost by working within a rotationally invariant framework.

The algorithm of Leung and Malik is motivated to a large extent by Psychophysics and Julesz's conjecture. Another very successful approach, that due to Konishi and Yuille [Konishi and Yuille, 2000], favours a more statistical approach based upon Bayesian classification. At first glance, these two methodologies might seem disparate but chapter 5 draws a connection between the two and shows how they can be made equivalent under a suitable choice of representation and similarity measure. In this chapter, it is first shown that texton frequencies and binned histograms form the same semi-parametric representation of filter response distributions. Next, it is noted how *naïve* Bayesian classification is equivalent to nearest neighbour matching using KL divergence or cross entropy as a similarity measure. Coupling these two facts together permits the implementation of a Bayesian classifier using

texton frequencies and allows direct comparisons between the methodologies of [Konishi and Yuille, 2000] and [Leung and Malik, 2001].

Chapter 6 then turns to the important question of whether multi-scale, multi-orientation filter banks are necessary for texture classification. Two results are established empirically. First, it is demonstrated that if the filter responses obtained using banks with support as large as 49×49 are replaced by compact image patches with neighbourhood sizes as small as 3×3, 5×5 or 7×7 then equivalent, or even superior, performances can be achieved. Given the contentious nature of the result it is validated on three separate databases. Second, it is also shown that the performance of filter banks is inferior to that of image patches with equivalent neighbourhoods. This is verified for different neighbourhood sizes. Both results run contrary to what is commonly held true in the texture literature and theoretical arguments are presented as to why patch based classification can outperform filter banks.

Chapter 7 is concerned with estimating the illuminant's direction from textured images. The aim is to develop a robust method which can be used to infer properties of the imaging conditions and thereby aid future classification. Most traditional formulations tend to tackle the problem under assumptions of constant albedo and strict Lambertian conditions. However, these stringent conditions are rarely met in the real world. In this chapter, we take a first step towards easing these restrictions. We develop a theory for estimating the illuminant's azimuthal angle from single images while making only general assumptions about the texture's surface and albedo. Deviations from the perfect Lambertian model due to shadows, specularities, inter-reflections, etc. are allowed as long as they can be incorporated into an albedo map drawn from a log-normal distribution. This permits our theory to be applied to all available images in the Columbia-Utrecht database and

it is shown to give good results even when its basic assumptions are violated. Finally, the theory is extended to cover the case where extra information is available in the form of a reference image.

We end in chapter 8 by exploring some of the avenues for future work and discussing some of the conclusions that can be drawn from this thesis.

Chapter 2

Literature Survey

In this chapter, we review the evolution of the texture classification problem over the last decade or so and highlight some of the important achievements and innovations made in attacking the problem. Section 2.1 presents an overview of the field while section 2.2 covers some of the particularly relevant literature in greater detail. Finally, section 2.3 discusses the databases that have been used in this thesis to benchmark performance.

2.1 Overview

Julesz's work on the visual perception of texture laid the ground for a lot of the subsequent research that followed, both in analysis and in synthesis. Since then, many different approaches have been formulated, including those based on filter banks [Konishi and Yuille, 2000, Leung and Malik, 2001, Schmid, 2001] and wavelets [Chang and Kuo, 1993, Do and Vetterli, 2002, Laine and Fan, 1993], affine invariant measurements [Caenen and Van Gool, 2004, Chetverikov, 2000, Lazebnik et al., 2003b, Lazebnik et al., 2003a] photometric stereo [Chantler et al., 2002b, Penirschke et al.,

2002, Wu and Chantler, 2003], Markov random fields [Chellappa and Chatterjee, 1985, Cross and Jain, 1983, Kashyap and Khotanzad, 1986, Lorette et al., 2000], co-occurrence matrices [Gotlieb and Kreyszig, 1990, Haralick et al., 1973], Voronoi polygons [Tuceryan and Jain, 1990] and fractals [Chen et al., 1989, Keller et al., 1989, Super and Bovik, 1991].

In this section, we present an overview of some of these approaches. The goal is not to be comprehensive or exhaustive but rather to put into perspective some of the important research that has shaped our understanding of texture as it is today. We begin with a discussion on recent progress over the last decade or so and review how the classification problem has evolved from binary pattern discrimination to 2D texture classification and finally to 3D texture classification. Proposed solutions using filter banks and co-occurrence have coped by building more and more complex representations and this evolution is charted as well. In subsection 2.1.1, the emphasis is primarily on filter bank based methods as they have provided key breakthroughs and insights into the field. Since filters have played such a crucial role, work on designing optimal filter banks for texture classification is reviewed next in subsection 2.1.2. Finally, subsection 2.1.3 looks at some MRF and other methods which have had an impact on the field. While the stress will be on approaches to texture classification some synthesis techniques will also be considered as the two problems are closely related.

2.1.1 Recent progress

Much of the early work on texture classification was concerned with Julesz's conjecture [Julesz et al., 1973] that two textures were perceptually indistinguishable if they had identical second order statistics (i.e. the distribution of intensities over pairs of pixels was identical). Efforts to (dis)prove the sup-

(a) (b)

Figure 2.1: The texture pair in each image have identical second order statistics. Yet, the two textures in (a) are not pre-attentively distinguishable while the pair in (b) are. Julesz surmised that this was because the textures in (b) had different texton densities. Image courtesy of [Tuceryan and Jain, 1998].

position focused on whether binary image patterns, such as the ones shown in figure 2.1, were pre-attentively discriminable by the human eye or not. The conjecture was eventually disproved [Caelli and Julesz, 1978] but only to be replaced by a statistical theory of *textons* [Julesz, 1981]. The theory postulated that textons were fundamental texture primitives such as line terminators, corners, intersections, etc. and two textures having different texton densities were easily distinguishable.

All this while, the classification task was to separate, in a binary image, two textures formed by the repeated placement of basic micro patterns. Thus, efforts were concentrated on synthetic images with little attention being paid to real world textures (with exceptions such as [Coggins and Jain, 1985] which presented results on the Brodatz album). The theory of textons was prevalent till the late eighties by when its two major shortcomings were established. The first was about how to formalise a list of universal textons and the second

was how to generalise the theory to gray scale images.

Meanwhile, people had been experimenting with using filter banks for texture analysis throughout the decade [Coggins and Jain, 1985, Caelli and Moraglia, 1985, Faugeras, 1978, Fogel and Sagi, 1989, Laws, 1980, Turner, 1986, Unser, 1986]. In fact, Julesz had tried to explain counter examples to his second order statistics theory using filter outputs [Julesz et al., 1973] and then later posited a link between textons and filter banks [Julesz and Bergen, 1983]. And, as the limitations of the texton theory were realised, filter bank based methods started gaining popularity. Two very influential theories which drew attention away from textons and sparked an alternate

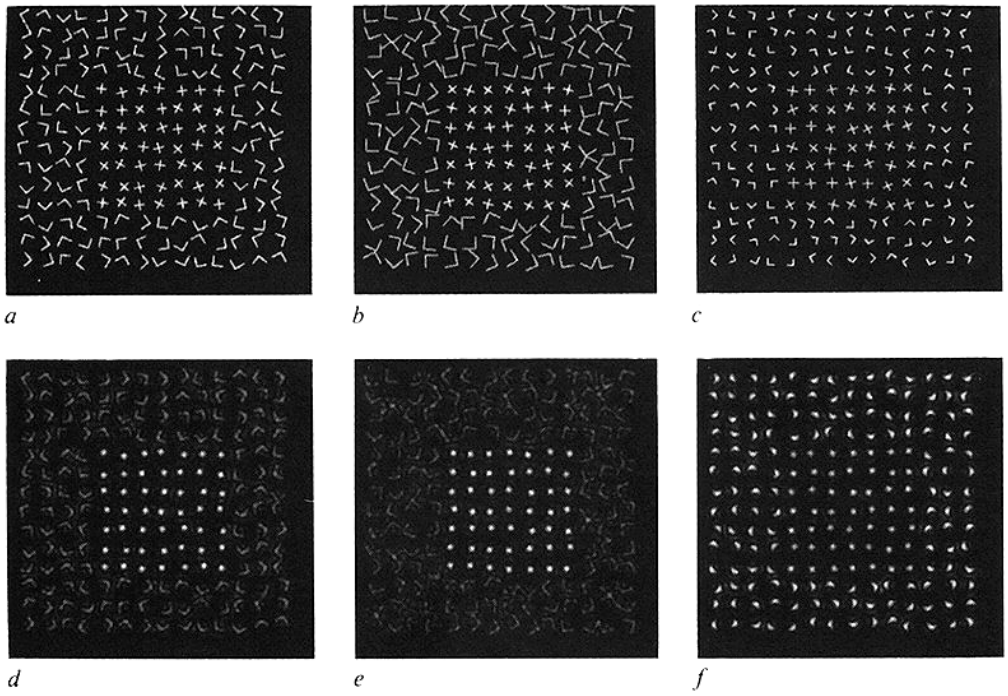


Figure 2.2: Results from [Bergen and Adelson, 1988]: The two textures shown in (a) get easier or harder to distinguish as the size of the "L" pattern increases in (b) or decreases in (c) even though the density of crossing and terminators has remained the same. However, this discriminability is predicted well by the responses of a centre surround filter shown in (d) - (f).

interest in filter banks were developed in [Bergen and Adelson, 1988, Malik and Perona, 1990]. Bergen and Adelson demonstrated that the responses of size tuned centre surround filters could be used to predict the discriminability between two textures. For instance, figure 2.2 shows that if the size of the basic micro pattern is increased (decreased) then the density of terminators and crossings remains unchanged whereas humans find it easier (harder) to distinguish between the two textures. Such behaviour is faithfully mimicked by the distribution of filter responses as the difference between the total energies of the two distributions increases or decreases according to human perception. The results of [Voorhees and Poggio, 1988] backed up this theory as they showed how basic texture elements for natural gray scale images could be computed by thresholding the outputs of centre surround filter followed by morphological operations. They also showed how texture boundaries could be computed for natural images.

Malik and Perona also drew inspiration from early visual processing in humans and developed a filter bank based model with biological plausibility as its cornerstone. Their computational results accurately predicted psychophysical data about the degree of texture discriminability and their method was able to successfully distinguish between cases previously thought to be very hard. The work of [Freeman, 1992] and [Perona, 1992, Perona, 1995] was also influential, but for more computational reasons, as they provided methods of calculating filter responses at all possible orientations and scales from a small basis set.

One of the major advantages that filter banks enjoyed over texton feature methods was that they could be used to analyse gray scale images. This resulted in the problem of 2D texture classification being brought to the fore. The emphasis shifted from distinguishing patterns in synthetic binary images

to classifying gray scale images of real world textures. Due to computational limitations, the early filter bank based methods were only able to use low order moments to characterise the distribution of filter responses. The feature extraction scheme that became standardised was to form a long vector whose components were the mean or variance of each of the individual filter response distributions. Some processing of filter responses, such as rectification, energy measurement or conversion to a rotationally invariant frame, was also done. A classifier of choice was then trained on the feature vectors and used to classify novel images. Typical examples of such frameworks are [Greenspan et al., 1994, Haley and Manjunath, 1995, Smith and Chang, 1994]. Performance was generally assessed on variations of the Brodatz album [Brodatz, 1966] and fairly good classification results were obtained.

From the mid nineties onwards, filter bank and wavelet based methods became increasingly successful at texture classification and synthesis and came to be regarded as the method of choice. Their improved performance was largely due to the fact that richer representations of the filter response distributions were being learnt. The representations were richer in primarily two respects: first, full filter response distributions were learnt as opposed to recording just the low order moments and second, the *joint* distribution, or co-occurrence, of filter responses was learnt as opposed to independently learning distributions for each filter separately. Another general trend was that the number of filters and wavelets used kept increasing so as to measure features at many scales and orientations.

A good example of the progression in complexity of filter response representations comes from texture synthesis. The synthesis task is to take a target texture and generate an output image which is similar but not identical to the target. The synthesis techniques developed during this phase of

progression heavily influenced the next generation of classification algorithms and so we take a brief detour and review them now.

Synthesis techniques by the mid nineties had graduated from using the mean and variance of distributions to model textures. Instead, two of the leading algorithms of the time [Heeger and Bergen, 1995, Zhu et al., 1996], advocated synthesis by forcing the marginal filter distributions of the output image to match the marginal filter distributions of the target texture. Heeger and Bergen's method formed a steerable pyramid using twelve filters while Zhu *et al.* relied on between four to six adaptively chosen ones. The main point of difference between the methods came from the realization that there exist many images which have the same filter response marginals. From amongst all these, [Zhu et al., 1996, Zhu et al., 1998] preferred choosing those images which were drawn from a distribution having maximal entropy – as this imposed no additional constraints on the image PDF.

Synthesis results for both algorithms are shown in figure 2.3. Drawing samples from the image PDF did result in improved results for Zhu *et al.* though the use of Gibbs sampling involved significant computational costs. However, choosing the distribution with maximal entropy left the image free to vary considerably as long as the marginals matched. This meant that the synthesised images could be noisy and did not necessarily have the same spatial structure as the target texture beyond that which was captured by the local filter support.

The next step forward was when DeBonet [De Bonet, 1997] extended Heeger and Bergen's algorithm to match the joint distribution of filter responses. There was no explicit texture model and the representation was defined non-parametrically in terms of "parent vectors" in the target image. Essentially, the probability of observing a particular filter response vector

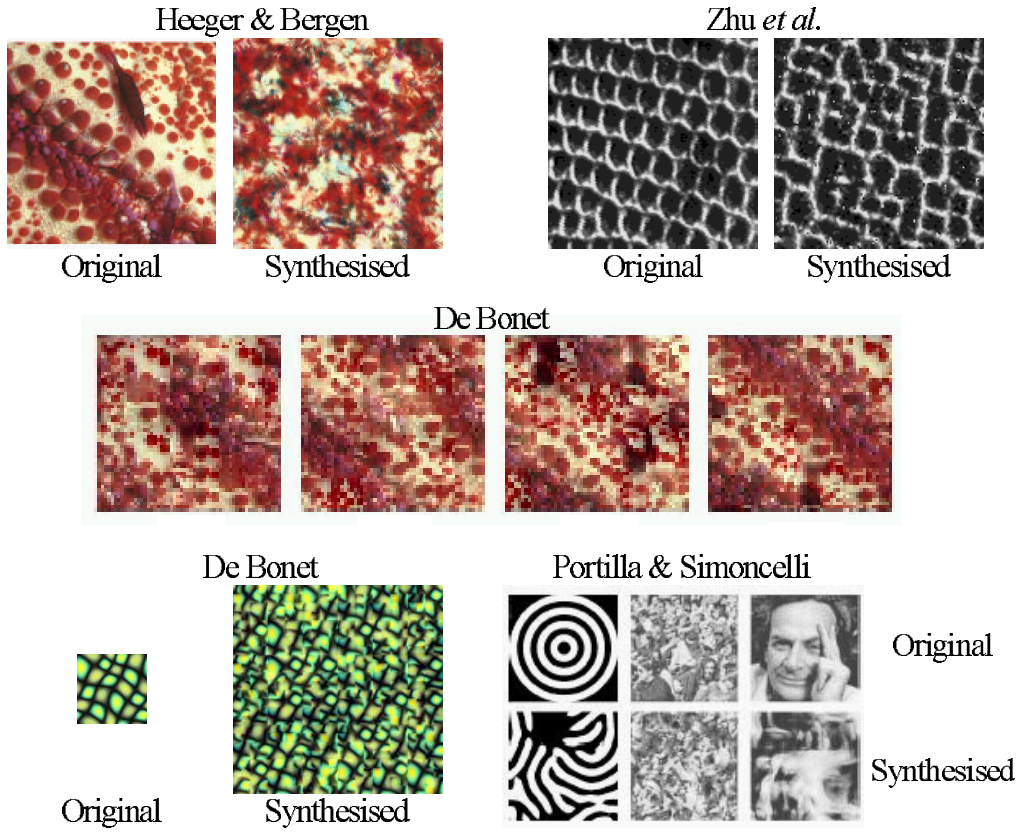


Figure 2.3: Texture synthesis using filter banks: the algorithm of Heeger and Bergen completely breaks down for structured textures. The method of Zhu *et al.* is an improvement as the overall structure of the pattern is discernible. However, the synthesis is noisy and has missing connectors. De Bonet's algorithm should be better as it models the joint structure over larger scales but it is still not successful when asked to synthesise the original image shown for Heeger and Bergen. Even the method of Portilla and Simoncelli fails for textures with global structure.

was determined by counting the number of parent vectors which lay within a certain L_2 distance of it in filter space. This probability distribution was sampled from to synthesise a filter response pyramid and thereby the output image. Modelling the joint distribution over the filter pyramid resulted in the target's spatial structure being captured over a range of scales and not just being restricted to the local filter support. However, the algorithm could

still not capture long range spatial interactions and had a tendency to tile the input image.

Finally, [Simoncelli and Portilla, 1998, Portilla and Simoncelli, 2000] imposed constraints on the joint PDF of 18 filter responses in order to validate Julesz's conjecture from psychophysics that two textures would be indistinguishable if their constraint functions had identical statistics. The statistical constraints imposed were that the low order moments of pixel intensities and filter response marginals must match between target and output as must the central regions of the auto and cross-correlation functions of filter magnitudes and phases. Using the correlations helped to impose both the target's local structure and long range interactions onto the output image. Synthesis results are shown in figure 2.3. All these methods cited biological plausibility for choosing filters to mimic the human visual system in an effort to synthesise perceptually similar textures. Yet, even though the results had improved tremendously as more filters were used and more effort put into modelling their PDF, the synthesised results were often not close to the target textures. Thus, biological plausibility and psychophysical arguments did not appear to hold the key for generating good synthesis results.

Right through this period of progression, synthesis and classification algorithms treated textures as pure albedo patterns painted on a flat surface. Under such an assumption, a single image could completely characterise all the possible variations of a texture patch. For instance, synthetic affine transformations of the image were accurate reflections of what the texture would look like if physically rotated and scaled in the real world. However, it soon became apparant that such 2D texture models were not very physically plausible as they ignored all 3D effects including surface normal variations, BRDF variations, illumination changes, scale and perspective effects, etc.

Realizing the need to study and model 3D texture effects, [Dana et al., 1997, Dana et al., 1999] compiled the Columbia-Utrecht (CUReT) database and included over 200 images of each of 61 materials taken under different viewpoints and illumination (see subsection 2.3.1). They initially used the database to demonstrate that 3D texture synthesis on a cylindrical surface gave much more realistic results as compared to traditional 2D texture mapping. However, the database later came to be an invaluable tool not just for synthesis but also for building and testing theoretical models of 3D textures.

Since 3D effects have a dramatic impact on the imaged appearance of real world textures (see figure 1.12), the next phase in texture classification was to bring 3D textures within the ambit of the problem. While it might be natural to assume that such effects could be compensated for by building physical models of 3D textures this did not turn out to be the case (see subsection 2.1.3). Instead, the lead was taken once again by filter bank based methods.

Leung and Malik [Leung and Malik, 1999, Leung and Malik, 2001] were amongst the first to seriously attempt the problem of classifying 3D textures under varying viewpoint and illumination. They made an important innovation by giving an operational definition of a texton based on filter responses and clustering. They defined a 2D texton as a cluster centre in filter response space. This not only enabled textons to be generated automatically from an image, but also opened up the possibility of a *universal* set of textons for all textures. To compensate for 3D effects, they proposed 3D textons which were cluster centres of filter responses over a stack of 20 images with representative viewpoints and lighting. The frequency distribution of these textons was shown to be sufficient for classifying a set of registered novel images taken under the same conditions. A nearest neighbour classifier based on

the χ^2 statistic was used. While the paper was very important as it set up a framework in which 3D textures could be classified successfully, it had serious limitations in that it needed multiple images during classification which had to be taken under exactly the same conditions as those during training. The paper is discussed in greater detail in subsection 2.2.3.

Meanwhile, filter banks were also posting the best results in other closely related areas. In particular, the success of Bayesian classification applied to filter responses was convincingly demonstrated by Konishi and Yuille [Konishi and Yuille, 2000]. They learnt the joint PDF of the responses of six filters for classes such as air, road, vegetation, etc. and used them to label individual pixels in the Sowerby and San Francisco outdoor datasets (described in subsection 2.3.2). Thus, they were able to do classification and segmentation relying solely on domain specific knowledge. There was also some preliminary investigation to see whether PDFs learnt from one dataset could be used to classify and segment the other. Unfortunately, special attention was not paid to the fact that 3D textures vary considerably with imaging conditions and only a single distribution of filter responses was learnt for each class. The paper is discussed in more detail in subsection 2.2.1.

Similarly, [Schmid, 2001] modelled the joint PDF of thirteen rotationally invariant filters for the purposes of image retrieval. In addition, even the co-occurrence of these thirteen dimensional filter responses was modelled to achieve impressive results. More details are given in subsection 2.2.2.

Cula and Dana [Cula and Dana, 2001, Cula and Dana, 2004] then proposed a system which addressed some of the major shortcomings of Leung and Malik's algorithm. They demonstrated that 2D textons (learnt from filter responses of single images instead of image stacks) could themselves be used for uncalibrated, single image classification without compromising on

performance. The use of 2D textons allowed a texture to be characterised by multiple probability distributions (*models*). Theoretically as many as one from each training image could be learnt to sample the effects of viewpoint and illumination variations. In practice, only a few models were needed as the rest were discarded by a manifold shape preserving technique for model reduction. A somewhat similar approach was independently suggested in [Varma and Zisserman, 2002a] and is developed in chapters 3 and 4. The overarching framework of both these approaches was also very similar to the algorithm of [De Bonet and Viola, 1998] though the details were significantly different.

The problem of reducing the number of models required to characterise a texture is a major one and, broadly speaking, two different approaches have been proposed. The first approach is Geometric and focuses on building affine invariant texture descriptors so as to reduce the number of models needed to cope with variation in camera pose. [Schaffalitzky and Zisserman, 2001] exploited the fact that a texture with sufficient directional variation can be *pose normalised* by maximising the weak isotropy of its second moment matrix (the technique is applicable in the absence of 3D texture effects). In essence, two images of the same texture which differ by an affine transformation are reduced to a canonical frame where they differ by only a similarity transformation. Full invariance can then be achieved by using a scale and rotation invariant filter bank to extract features.

One drawback of this technique is that the proposed normalisation is global rather than local. Not only would local normalisation be more robust but it would also allow the method to be extended to textures which are not globally planar but which can be approximated as being locally planar. Realizing this, [Lazebnik et al., 2003a, Lazebnik et al., 2003b] proposed an alternative method of generating local, affine invariant, texture features. In

their framework, certain interest regions were first detected using a Laplacian blob detector. The characteristic scale at each point was determined and the region pose normalised locally. Spin images were then used instead of filter banks to generate affine invariant features for each region. The system achieved good classification results on both the Brodatz and the UIUC datasets. The paper is discussed in subsection 2.2.5.

In the second approach to model reduction, concepts from Machine Learning can be used to select a subset of the models while maximising some criteria of classification and generalisation. A good example of this is [Hayman et al., 2004] where the nearest neighbour classifier used in [Varma and Zisserman, 2002a] is replaced by a Support Vector Machine. They show that this not only improves classification performance on the CUReT database but also provides a principled way of selecting the required models. The average number of support vectors used is demonstrated to be 10 - 20% lower than the number of models required by a nearest neighbour classifier. The paper also considers how far pure Learning approaches can go towards coping with imaging variations, especially those due to scale. Not surprisingly, the conclusion is that classification performance is acceptable as long as the scaled images are included in the training set but deteriorates very rapidly if they are not. A similar effect was observed for different instances of the same material, i.e. training on one instance of a material was no guarantee that another instance could then be classified correctly.

To summarise, the texture classification problem has matured considerably over the last two decades. The emphasis in the eighties was on separating patterns in synthetic binary images. This progressed in the early nineties to attempting classification of gray scale images of real world textures but with 2D variations due to synthetic rotations and scaling. Finally, in the late

nineties, the classification task embraced real world 3D textures with real variations caused due to changing viewpoint and illumination. Throughout this period the most effective solution had been provided by filter banks which had themselves progressed by building more and more complex representations. They were introduced at first due to their biological plausibility and were soon seen as operators to extract features at multiple orientations and scales. Initially, during the early nineties, feature vectors were formed from only the mean and variance of filter response distributions. This has now changed considerably and the full joint PDF of filter responses is modelled. However, it must be pointed out that some recent papers still persist in attempting the 2D problem and approach classification via the dated technique of concatenating the low order moments of distributions to form feature vectors [Pun and Lee, 2003, Sebe and Lew, 2000, Singh and Singh, 2002].

2.1.2 Optimal filtering

All the algorithms discussed so far had chosen their filter banks heuristically rather than by optimising classification rates. It is therefore expected that the performance of the algorithms will get even better if their filters were to be replaced by ones optimised specifically for the given classification task. While attempts have been made at designing optimal filter banks they have not had much of an impact on the field. This is primarily because such methods tend not to minimise the classification error itself but rather optimise other criteria, such as the separation between filter responses, in the hope that this will decrease the error. Such choices are necessary as it is often impossible to analytically express classification error as a function of the input filter bank while numerical techniques for classification error minimization are often too costly. As such, these optimisation techniques are not widely in use but are

nevertheless important as tools to reason about the filtering process. We therefore present a very brief overview of different optimisation methods in this subsection.

On one end of the spectrum are methods which determine filter banks and wavelets to be optimal because they are biologically plausible, or because they are optimally localised in space and frequency [Daugman, 1985], or because of their shift invariance and regularity properties [Mojsilovic et al., 2000]. A bit more relevant are methods which are optimal at characterising textures though these are not necessarily optimal at discrimination. Such methods include those which derive filters from the eigenvectors of the autocorrelation function [Ade, 1983] and those derived by PCA/ICA of image patches [Messer et al., 1999]. Another such method is the predictive linear filter of [Randen, 1997] which turns out to be extremely similar to a Gaussian Markov random field model [Li, 2001] where the parameters are learnt using the *pseudo-likelihood* estimate. However, such methods are geared more towards characterising textures rather than discriminating between them.

Moving along the spectrum, another class of methods is based on choosing the best subset of filters from a fixed filter bank. For example, [Zhu et al., 1998] propose starting with a large set of filters and then iteratively choosing an optimal subset which maximises the L_1 norm of the probability distributions between classes. Similarly, [Bovik et al., 1990, Dunn and Higgins, 1995, Weldon and Higgins, 1996] have proposed methods to optimise the parameters of Gabor filters for texture classification. However, these approaches are restricted by the need for the optimal filters to be already present within the initial basis set.

A more general approach is taken by methods which apply techniques from Discriminant Analysis. Most such approaches focus on the two class

problem and theorise that classification errors must decrease if the separation between the filter responses of the two classes is maximised (due to the reduced PDF overlap). Many different ways of measuring the distance between two classes have been proposed in the literature. Three of the most popular ones are due to Fisher [Duda et al., 2001], Unser [Unser, 1986] and Mahalanobis and Singh [Mahalanobis and Singh, 1994]. Each of the three optimisation criteria are derived by making different assumptions about the underlying distribution which generated the filter responses. The criteria are

$$J_F = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}; \quad J_U = \frac{(\mu_1 - \mu_2)^2}{\mu_1 \mu_2}; \quad J_{MS} = \frac{\mu_1}{\mu_2} \tag{2.1}$$

where μ_1, σ_1, μ_2 and σ_2 are the means and variances of the two classes respectively. Interestingly, it should be noted that the linear SVM also provides a measure of the maximum separation between two classes and relates it directly to classification. In fact, the normal to the separating hyperplane acts as an optimal filter (see subsection 6.4.2). As such, a linear SVM should often turn out to be the best optimizer, specially as its underlying assumptions are the least restrictive. However, a major drawback of such methods is that they do not generalise readily to many class problems. Even though an N class problem can be decomposed into $\mathcal{O}(N^2)$ two class problems, the resulting number of filters would be too large for most classification applications today.

Finally, at the other end of the spectrum, [Jain and Karu, 1996] developed one of the few optimised fully end-to-end classification systems. They noted that the first layer weights in a neural network essentially play the same role as a filter bank. Therefore, training a neural network classifier to learn the weights is equivalent to designing optimal filters for the given classification task. Furthermore, it is even possible to determine the number of optimal

filters by using an iterative pruning algorithm to reduce the number of nodes in the first layer. However the size of the filter support still has to be determined *a priori*. A similar system, the Convolutional Neural Network, has been developed by [LeCun et al., 1998] and shown to give outstanding results for document recognition. Unfortunately, these methods are embedded into the neural network framework of learning and classification and are difficult to adapt for other methodologies.

2.1.3 Physical models and MRF methods

Physical models

The process of developing physical models to explain the behaviour of 3D textures became much easier with the availability of the Columbia-Utrecht database. Dana and Nayar [Dana and Nayar, 1998] were then able to experimentally validate a model which predicted a texture's intensity distribution under varying viewpoint and illumination. The model's parameter was the roughness of the textured surface. Results were presented for five materials in the database. [Dana and Nayar, 1999] were also able to predict the change in correlation length of the textured rough surface with viewing direction. Meanwhile, Chantler *et al.* broached the subject of how changes in the illuminant's direction could adversely effect a classifier's performance unless modelled and compensated for. Their research [Chantler et al., 2002b,Penirschke et al., 2002] has provided valuable theoretical insight into how the variances of filter responses change with the illuminant's tilt. Later on, [Koenderink and Pont, 2003] showed that a statistical description of surface roughness was also sufficient to estimate the illuminant's tilt direction from single images and thereby explained some of Chantler's results. A generalisation of

Koenderink and Pont's theory is presented in chapter 7.

While these physical models of 3D textures were theoretically very appealing, they didn't translate into practical classification algorithms because of their restrictive assumptions – uniform albedo, Lambertian surfaces, inability to model shadows, occlusions, specularities, etc. However, physical BRDF models did lead to good results in synthesis [He et al., 1991,Ashikhmin et al., 2000]. For example, as figure 2.4 shows, BRDF methods appear to do a very good job of synthesising materials such as satin and velvet.

Figure 2.4: Examples of satin and velvet synthesised by [Ashikhmin et al., 2000].

MRFs and other methods

While filter bank and wavelet methods achieved great successes all through the nineties, their supremacy for texture synthesis has begun to be challenged recently by MRF and image patch methods.

The trend started with [Efros and Leung, 1999,Efros and Freeman, 2001] which had a major impact on the field of texture synthesis. Efros and Le-

ung discarded the traditional use of filter banks and demonstrated that an MRF (image patch) model could be used to directly synthesise textures with complicated spatial structure. This was done without even explicitly representing the distribution of the central pixel conditioned on its neighbourhood. A non-parametric technique was used instead. The basic premise was simple: an image already contains all the necessary information and there is no need to extract many features at different orientations and scales for synthesis. To illustrate the point, suppose all the pixels in the output image have been synthesised but one. That pixel can be synthesised by searching the input image and determining all the central pixels whose neighbourhood matches the current neighbourhood. The pixel value can then be set by sampling from this distribution of central pixels.

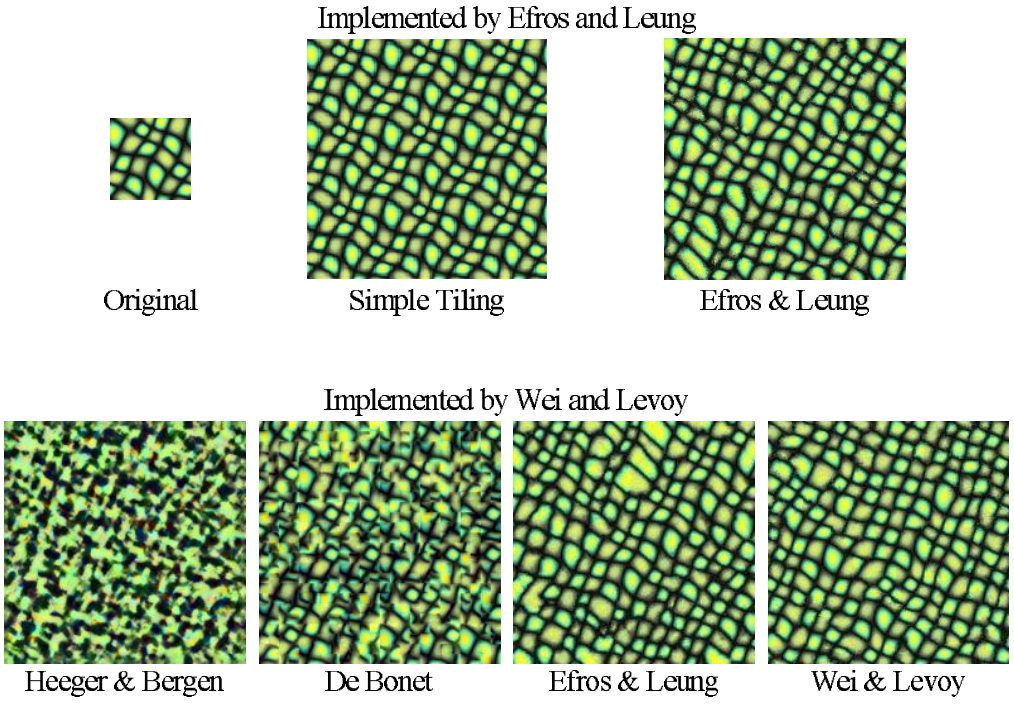


Figure 2.5: Texture synthesis using the MRF method of [Efros and Leung, 1999] and the deterministic method of [Wei and Levoy, 2000].

The idea is simple but revolutionised synthesis as the results obtained were far superior to anything achieved by filter banks (see figure 2.5). The main disadvantages of the method were that it was slow and since there wasn't an explicit texture model, the method could not be directly used for classification. There were also issues concerning the choice of neighbourhood and the convergence properties though some consistency results were proved by [Levina, 2002]. [Wei and Levoy, 2000] developed a variant of the algorithm which was fast and operated at multiple scales (somewhat similar to [De Bonet and Viola, 1998]) but which used a causal neighbourhood and was deterministic apart from the initialization.

[Zalesny and Van Gool, 2000] also showed that extremely good synthesis results could be obtained using the first and second order distributions of pixel intensities directly and without any filtering. Furthermore, they gave an iterative scheme for determining a compact MRF neighbourhood over which these distributions should be learnt. Synthesis was carried out by Gibbs sampling. The results are very appealing (see figure 2.6) and their

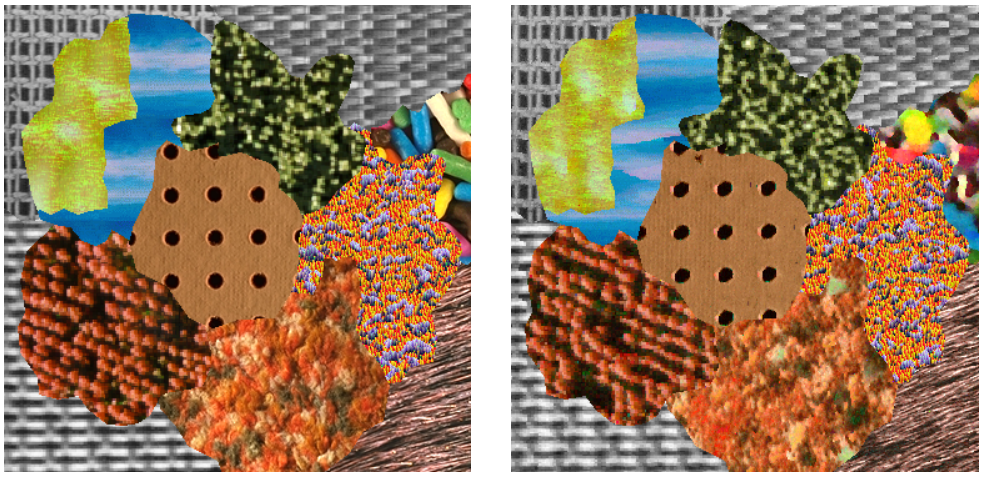


Figure 2.6: Texture synthesis using the MRF methods reported in [Gimel'farb et al., 2004]. On the left is a collage of the different images input to the algorithm and on the right is a collage of the synthesised results.

algorithm has many desirable characteristics. Firstly, it yields a compact model that can be used for analysis and classification. Secondly, the model can be used with minimum retraining to account for 3D texture effects due to viewpoint changes. Thirdly, the method can be shown to converge to a desirable solution. Recently, in [Gimel'farb et al., 2004] the authors have criticised [Zhu et al., 1998] for the use, and method of selection, of filter banks.

While such MRF methods have taken the lead from filter banks in texture synthesis, equivalent results have yet to be obtained for classification. The only non filter bank based method which has so far reported good results on the CUReT database is [Suen and Healey, 2000]. Instead of using filter banks to extract features, Suen and Healey used correlation functions across multiple colour bands to determine basis textures for each of the 61 materials in the CUReT database. They assumed that, for every texture image picked from a given class, the correlation function for that image could be represented as a linear combination of the basis texture correlation functions of that class. A nearest neighbour classifier employing the sum of squared differences metric was used. The number of basis images for a particular texture class also provided information about the *dimensionality* of that class, i.e. the number of models needed to successfully characterise the texture for classification purposes.

2.2 Papers in detail

In this section we review in greater detail some of the more recent literature that is particularly relevant for texture classification. The papers that will be covered are: [Konishi and Yuille, 2000], [Schmid, 2001], [Leung and Malik,

2001], [Cula and Dana, 2004] and [Lazebnik et al., 2003b].

2.2.1 The algorithm of Konishi and Yuille

The aim of [Konishi and Yuille, 2000] was to use colour and texture cues derived from local filter responses to label pixels as belonging to one of six classes in images of outdoor scenes. The six classes were Air, Building, Car / Edge, Road, Vegetation and Other. Information about smoothness of region boundaries or neighbouring pixel intensities was deliberately not used so as to demonstrate the classification power of the learnt filter response distributions.

During training, the joint PDF of the empirical probabilities of six filter responses was learnt for each of the categories. The filter bank was formed by combining gradient, Laplacian of Gaussian, Nitzberg (for texture) and Gaussian (for colour) filters. The distributions were then represented as histograms by quantizing the filter responses into six bins per dimension. Thus, each of the six texture classes was modelled by a single joint distribution of filter responses represented as a histogram with 6^6 bins.

Two types of experiments were conducted on the Sowerby and San Francisco databases. In the first, the entire database was used for both training and testing. In the second, PDFs were learnt from one half of the database while performance was measured on the other half. In each experiment, pixels in novel images were labelled using Bayesian classification. Both uniform and data driven priors were tried. An upper bound on the Bayes' error for region classification was also reported in terms of the Chernoff information. For the Sowerby database, the best results were obtained using colour and texture filters with data driven priors. Uniform priors were good for finding buildings and true edges. The results were broadly similar for the San Francisco database (see figure 2.7) except for the fact that uniform priors

gave much better results as there wasn't enough reliable training data from which to compute data driven priors. Some preliminary investigations were also carried out comparing the statistics of the six classes between the two datasets. Colour filters were not so useful this time as the two domains had significant differences between images of air.

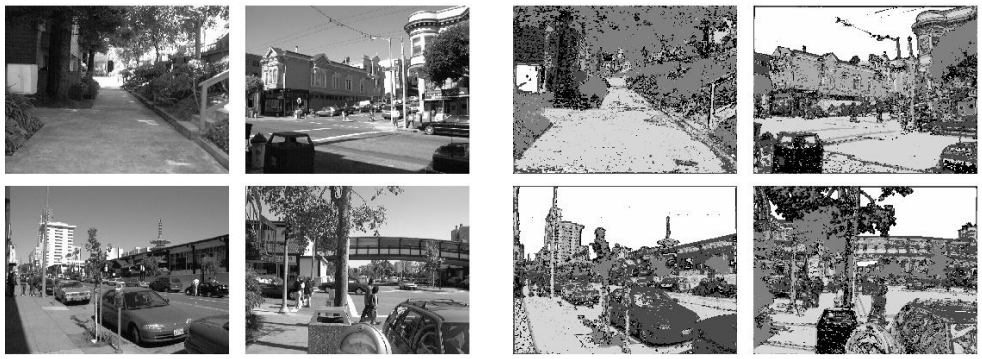


Figure 2.7: Pixel classification results on the San Francisco database from [Konishi and Yuille, 2000].

2.2.2 The algorithm of Schmid

A two layer semi-supervised approach was proposed by [Schmid, 2001] for image retrieval based on texture models. The first layer of the model was similar to Konishi and Yuille's and was formed by computing the joint PDF of thirteen rotationally invariant filters per texture class. However, instead of using binned histograms or a mixture model, the probability of obtaining a particular filter response was determined by the distance to the closest "generic descriptor" (a Gaussian with a given mean and covariance matrix). This resulted in the model not being a true PDF as the probabilities did not integrate to one. A second layer was then added to capture the frequency distribution of these generic descriptors so as to impose spatial constraints. This too was represented by a set of Gaussians called "spatial-frequency descrip-

tors". Associated with each pair of generic and spatial-frequency descriptors was a *significance* of how likely the pair was to have been generated from a given texture class (model). Each pixel could be quantized to its closest descriptor pair and thereby labelled with the probability of being generated by a particular model. Unfortunately, even though priors were assumed and a *naïve* Bayesian assumption was explicitly made, classification was done via a voting mechanism (by adding all the pixel probabilities) rather than the MAP rule.

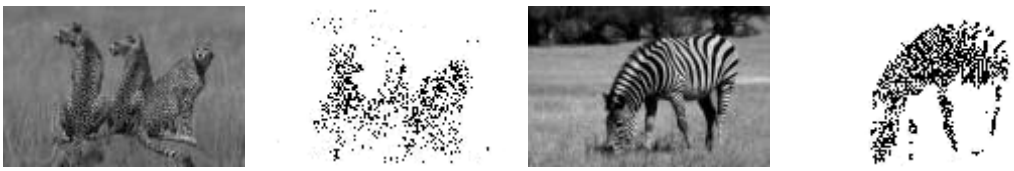


Figure 2.8: Texture localization results from [Schmid, 2001].

Results were demonstrated on the Corel database. Roughly sixty images from each of four classes (zebras, giraffes, cheetahs and faces) were included with five positive and ten negative images being used to train each class. The results for both localising significant neighbourhoods (see figure 2.8) and image retrieval were very good.

2.2.3 The algorithm of Leung and Malik

[Leung and Malik, 2001] developed a weak classifier to specifically overcome the problem of 3D texture classification. During learning, a stack of 20 registered images with known viewpoint and illumination were convolved with a 48 dimensional filter bank to generate features. The registration was necessary in order to learn how a texture varied with changing imaging conditions. The filter responses were concatenated together to form vectors in a $20 \times 48 = 960$ dimensional space. These 960-vectors were then clustered

using *K-Means* to determine exemplar filter responses called 3D textons. Just as the frequency distribution of 2D textons had been largely sufficient for 2D texture classification, the distribution of these 3D textons would suffice for 3D texture classification. Thus, to learn a model, all the concatenated filter responses of a material were labelled with the texton that lay closest to them in the 960 dimensional space. The distribution of texton frequencies then formed a single model for a given texture class.

During classification, a stack of 20 novel images belonging to one of the materials was presented. These novel images had to have been taken under the same conditions as the training set and had to have the same ordering in the stack (i.e. the viewpoint and illumination had to be known implicitly). Given the stack, filter responses could again be generated and labelled with the textons learnt during training. The distribution of texton frequencies was determined and compared to the learnt models using the χ^2 statistic. Classification was performed on the basis of nearest neighbour matching.

Results were presented for 40 textures in the CUReT database. Leung and Malik reported a remarkable accuracy rate of 95.6% when stacks of 20 images were classified from each of these 40 texture classes. They also developed an MCMC algorithm for classifying a single image under known imaging conditions. Training models were learnt from 4 images per class rather than 20. However, the classification accuracy of this algorithm was not as good as that achieved by the multiple image method. An accuracy rate of 87% was achieved when classifying 5 test images per material.

2.2.4 The algorithm of Cula and Dana

[Cula and Dana, 2001, Cula and Dana, 2004] then extended Leung and Malik's framework and showed how 2D textons could themselves be used

for uncalibrated, single image classification without compromising on performance. They demonstrated that characterising a texture by multiple models conditioned on viewpoint and illumination would permit a nearest neighbour classifier to return comparable results but without requiring knowledge of imaging conditions.

Thus the overall methodology was the same as Leung and Malik's except that all occurrences of 3D textons were replaced by 2D textons. For instance, during dictionary generation, filter responses were grouped by scale, aggregated across training images and clustered immediately without being concatenated. Filter responses from single training images were then labelled with these 2D textons and a texton frequency histogram was used to form texture models. However, there were multiple models now for each texture class sampling the viewing and illumination spheres. Nearest neighbour classification of single images was performed by matching their texton frequency histogram to the learnt models using the χ^2 statistic.

Cula and Dana also developed an algorithm for reducing the number of models. In the first step, both training and test histograms of a class were projected into a low dimensional space using PCA. A manifold was fitted to the projected points and then reduced by systematically discarding those points which least affected the shape of the manifold. The points which were left at the end corresponded to the model images that defined the texture. Since the models for a texture were chosen in isolation from the other textures, Cula and Dana's algorithm ignored the inter class variation between textures. Therefore, the selected models were not geared specifically towards classification.

Results were presented for 156 images selected from each of 20 classes in the CUReT database. If 56 models were chosen for training from each

texture then a classification rate of nearly 96% was achieved on the remaining $100 \times 20 = 2000$ images. However, if 8 models were chosen per class then a rate of only 71% was achieved while classifying the remaining $148 \times 20 = 2960$ images.

2.2.5 The algorithm of Lazebnik et al.

Two main innovations distinguished [Lazebnik et al., 2003b] from previous texture classification approaches. The first was the use of local affine invariant descriptors to reduce the number of models needed to characterise a texture. The second was to learn a local set of textons per image during training and classification as opposed to learning a universal texton dictionary in the first phase of training. The Earth Mover's Distance (EMD) was used to compare histograms in order to cope with local texton dictionaries.

The main steps of the algorithm were the following: interest points were detected in a given image using the Harris [Harris and Stephens, 1988] and normalised Laplacian of Gaussian (LOG) [Lindeberg, 1998] operators. A characteristic scale was determined for each point and the surrounding region reduced to a canonical frame via pose normalisation. Spin images of size 20×20 were then used to give a fully affine invariant descriptor at each interest point. These descriptors were then clustered agglomeratively using the L_2 distance to learn $10 - 15$ textons per image. The clustering was done separately for the two different types of interest points. The image was then labelled using the textons and its corresponding model was determined by the histogram of texton frequencies. The positions of the textons in spin image space was also recorded.

During classification, the texton frequencies and positions were determined for the novel image. These were then compared to the learnt models

using the EMD. A separate comparison was performed for the Harris interest points and the LOG interest points and the values added to form a final score for the distance between the two images. Classification was done using nearest neighbour matching.

Results were reported on a database of 10 materials each being imaged under 20 different viewpoints and illumination. The training set comprised 5 different images per material while the rest of the database was used for testing. Accuracy rates of 89% were achieved despite there being significant viewpoint and scale changes in the database. Results were also reported on the Brodatz dataset when the images were cropped into 9 non-overlapping regions to give a total of 999 images in all. Three images were chosen from each of the 111 classes for training. This time a third distance based on region shape was added to the distances computed using the two sets of interest points. The classification accuracy was determined to be 85%.

2.3 Databases

We conclude this chapter by describing the four databases that have been used in this thesis. The first three are the Columbia-Utrecht, San Francisco and Microsoft Textile databases and these have been used to benchmark the performance of classification algorithms. The fourth is the Heriot-Watt TextureLab database on which some of the illumination direction estimation experiments have been carried out in chapter 7.

2.3.1 The Columbia-Utrecht (CUReT) database

The Columbia-Utrecht (CUReT) database [Dana et al., 1999] contains images of 61 materials. The textures selected for inclusion in the database

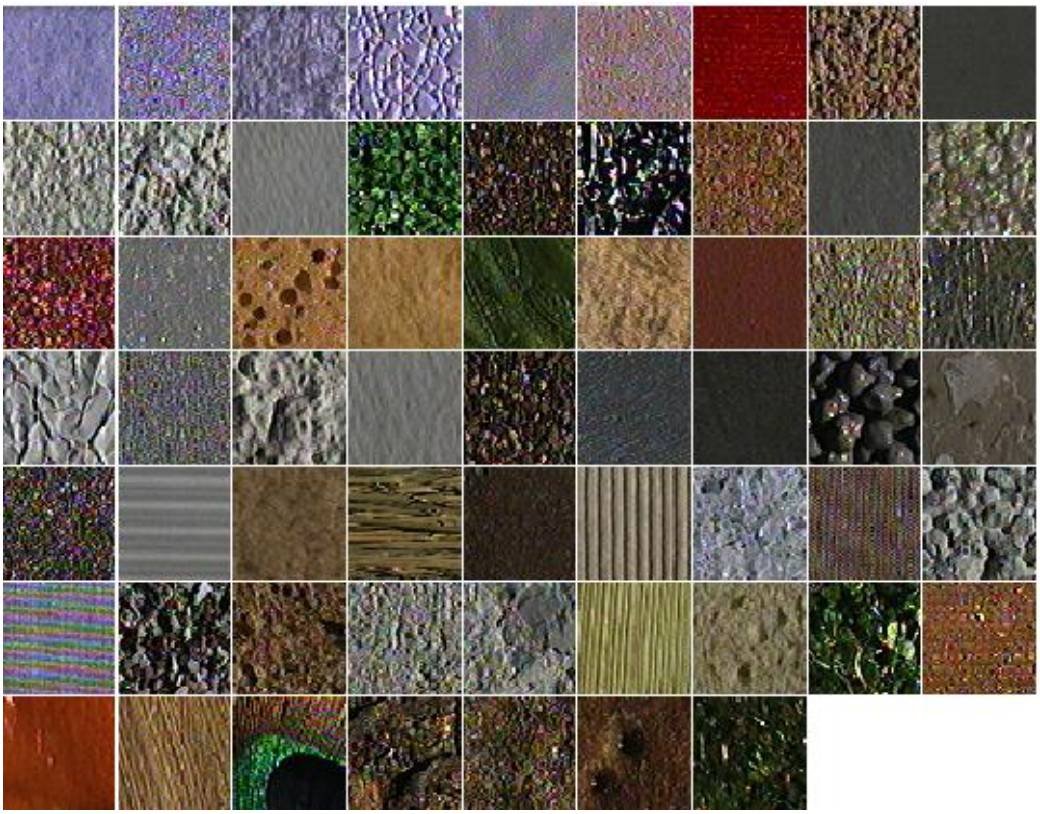


Figure 2.9: One sample of each of the materials present in the Columbia-Utrecht (CUReT) database.

attempt to span the range of different surfaces that one might commonly see in todays environment. The database has textures that are rough, those which have specularities, exhibit anisotropy, are man-made and many others. The variety of textures present in the database is shown in figure 2.9.

Each of the materials in the database has been imaged under 205 different viewing and illumination conditions. The effects of specularities, inter-reflections, shadowing and other surface normal variations are plainly evident and can be seen in figures 1.12 and 1.14 where their impact is highlighted due to varying imaging conditions. This makes the database far more challenging for a classifier than the often used Brodatz collection where all such

effects are absent. Therefore, the CUReT database will be used in most of the experiments performed in this thesis.

However, the raw image data by itself is not amenable for running classification experiments. Each of the images has extraneous background clutter and images taken from some of the more extreme viewpoints have only a very small region of texture that is visible. Thus, the following modifications are made to the database in order to make it suitable for experimentation: for each material present in the database, there are 118 images where the azimuthal viewing angle is less than 60 degrees. Out of these, 92 images are chosen for which a sufficiently large region of texture is visible across all materials. A central 200×200 region is cropped from each of these images and the remaining background discarded. The selected regions are converted to grey scale and then intensity normalised to have zero mean and unit standard deviation. Thus, no colour information is used in any of the experiments and we make ourselves invariant to affine changes in the illuminant intensity. Fig-

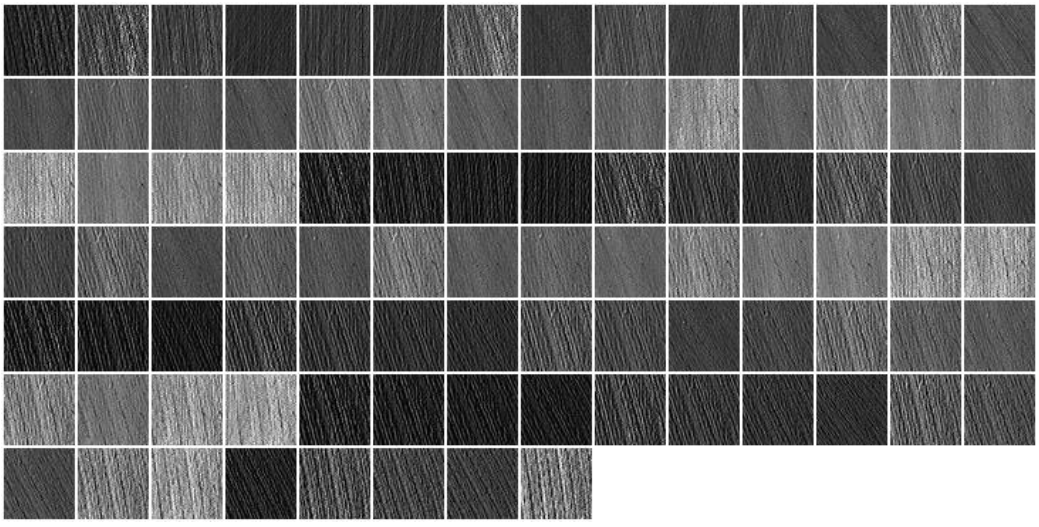


Figure 2.10: The 92 images which were selected from the texture class Wood (material number 56).

ure 2.10 shows the 92 cropped regions of the Wood (texture number 56). The cropped CUReT database has a total of $61 \times 92 = 5612$ images. These are evenly split into two disjoint sets of 2806 images each, one for training and the other for testing. Most of the results reported in the following chapters will be based on these sets.

While the CUReT database has now become a benchmark and is widely used to assess classification performance, it also has some limitations. These are mainly to do with the way the images have been photographed and the choice of textures. For the former, there is no significant scale change for most of the materials and very limited in-plane rotation. As all the images have been taken in the lab, the illumination has been controlled to a very large extent as well. For example, even though the illuminant's direction varies a lot, the illuminant's intensity has been kept relatively constant. Furthermore, the use of multiple and diffuse illuminants has not been explored. With regard to choice of texture, the most serious drawback is that multiple instances of the same texture are present for only a few of the materials, so intra-class variation cannot be thoroughly investigated. Hence, it is difficult to make generalisations. Nevertheless, it is still one of the largest and toughest databases for a texture classifier to deal with and is therefore used extensively in this research.

2.3.2 The San Francisco database

The San Francisco database has 37 images of outdoor scenes taken on the streets of San Francisco. It has been segmented by hand into 6 classes: Air, Building, Car, Road, Vegetation and Trunk. Note that this is slightly different from the description reported in [Konishi and Yuille, 2000] where only 35 images were used and the classes were: Air, Building, Car, Road,

Figure 2.11: Sample images from the San Francisco database.

Vegetation and Other. Figure 2.11 shows some sample images from the database. The images all have resolution 640×480.

As can be seen, the database is easy to classify on the basis of colour alone – the sky is always blue, the road mostly black and the vegetation green. Therefore, in this thesis, the images are converted to gray scale to make sure classification is done only on the basis of texture and not of colour. Also, when the database is used in chapter 6, each image patch is normalised by subtracting off the median value and dividing by the standard deviation. This further ensures that classification is actually carried out on the basis of textural information and not just intensity differences (i.e. a bright sky versus a dark road).

The database is challenging because individual texture regions can be

small and irregularly shaped. The images of urban scenes are also quite varied. However, the three main classes, Air, Road and Vegetation, tend not to change all that much from image to image (the database does not include any images taken at night or under artifical illumination). The other shortcoming of the database is its small size.

2.3.3 The Microsoft Textile database

The Microsoft Textile database [Savarese and Criminsi, 2004] has 16 folded textures with 20 images available of each taken under diffuse artifical lighting. This is one of the first attempts at studying non-planar textures and therefore represents an important step in the evolution of the texture analysis problem.

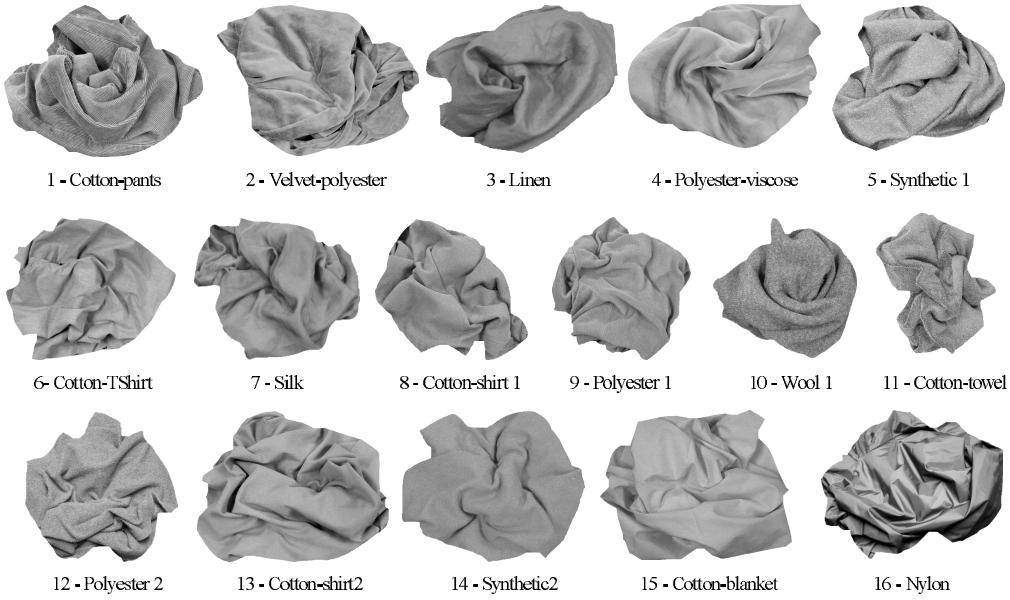


Figure 2.12: Textures present in the Microsoft Textile database.

The images all have resolution 1024×768. The foreground texture has been segmented from the background using GrabCut [Rother et al., 2004]. The impact of non-Lambertian effects is plainly visible as in the Columbia-

Figure 2.13: Images of Nylon from the Microsoft Textile database.

Utrecht database. Figure 2.13 shows some sample images of Nylon. The variation in pose and the deformations of the textured surface make it an interesting database to analyse. Furthermore, additional data is available which has been imaged under large variations in illumination conditions.

2.3.4 The Heriot-Watt TextureLab database

The Heriot-Watt TextureLab database [Wu and Chantler, 2003] was developed to study the photometric properties of nearly Lambertian materials. The goal was to investigate how such 3D textures vary with changing illumination and surface rotation.

There are 30 textures in the database and each has been imaged from a

fixed viewpoint. The illuminant's elevation is also fixed at $\nu = 45°$ but the azimuth varies between $\psi = 0°$ and $\psi = 315°$ in steps of 45°. Each texture has also been imaged for seven different values of surface rotation from 0° to 180° in steps of 30°. Thus, there are 56 images of each material all at resolution 512×512. The textures occupy the entire image so there is no need for segmentation.

Since the images are all registered this is an excellent database for conducting photometric stereo analysis of the included materials (the light source positions are known as well). While there is rotation present in the database, the samples have all been imaged at the same scale unfortunately. The

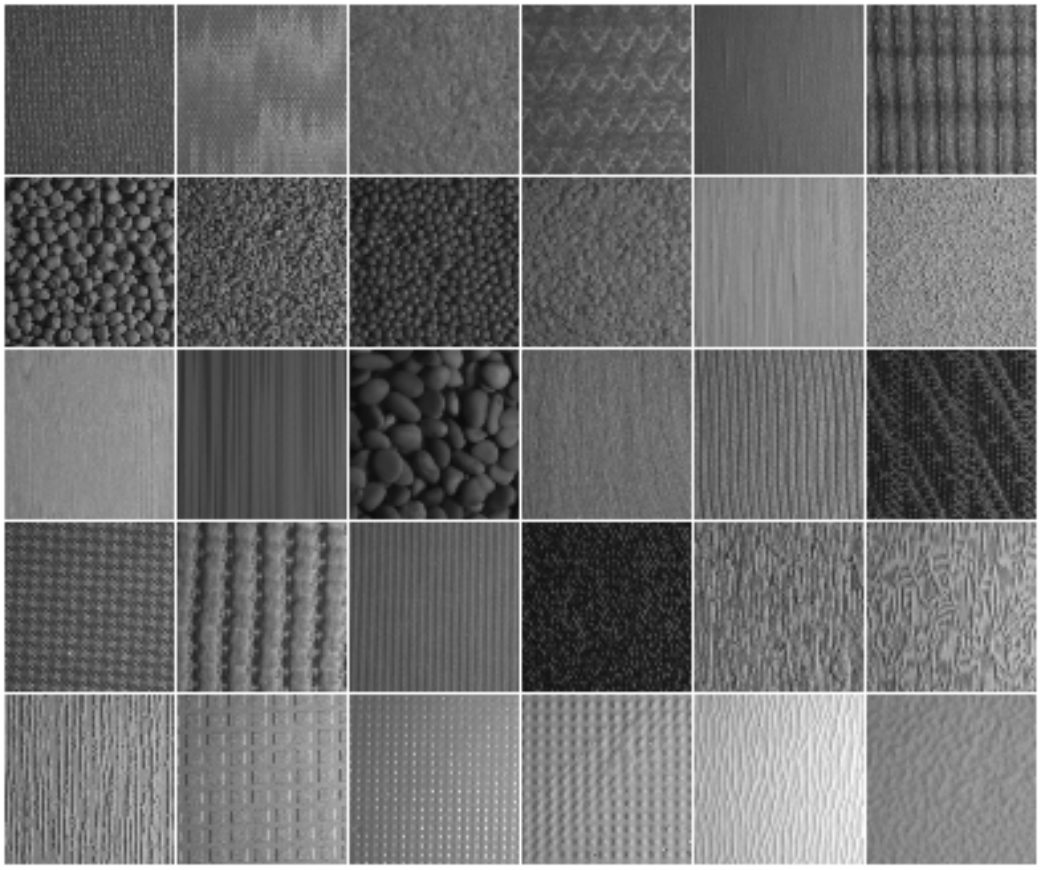


Figure 2.14: Materials present in the Heriot-Watt TextureLab database.

slant angle of the illuminant is also fixed and therefore the variability in the database is limited. Therefore, in this thesis, the database is only used for experiments on determining the illuminant's azimuthal direction.

Chapter 3

A Filter Bank Based Approach To Texture Classification

This chapter focuses on building a framework in which filter banks are used to extract features for texture classification. The use of filter banks is motivated with a discussion on the type of information extracted by a filter and its representation. The particular aspects of interest are the dimensionality and invariance of filter responses. We explore designing low dimensional, maximum response filter banks which are rotationally invariant but nevertheless able to extract rich features from textured images.

To tackle single image classification, a statistical approach is developed where textures are modelled by the joint probability distribution of filter responses. This distribution is represented by the frequency histogram of filter response cluster centres (textons). A nearest neighbour classifier is used and performance is assessed on the materials in the Columbia-Utrecht database. Empirical results are presented for the different types of filter banks considered and the advantages of using low dimensional, rotationally invariant features discussed.

3.1 Introduction

In this chapter, a statistical approach to single image texture classification is formulated based on extracting features via convolution with a filter bank followed by nearest neighbour matching. This is motivated by the remarkable successes achieved by recent filter bank based classification algorithms [Cula and Dana, 2004, Hayman et al., 2004, Konishi and Yuille, 2000, Leung and Malik, 2001, Schmid, 2001].

Two main issues crop up when designing or selecting a filter bank for classification. The first is about which filters should be included and here one runs into the well known selectivity (discriminability) versus invariance dilemma. On the one hand, it is tempting to choose filters which are able to discriminate very well between textures, i.e. the filter responses should change considerably when going from one class to another. On the other hand, it is also desirable for the filters to be invariant to imaging conditions, i.e. the filter responses should stay the same when the texture class is fixed but the surface or camera rotates or the illumination or viewpoint changes. Attaining both goals is well nigh impossible and, conventionally, filter banks have either been designed to be very discriminative (by including filters at many orientations and scales [Leung and Malik, 2001]) but have no invariance or have complete rotational symmetry [Schmid, 2001], and thereby be invariant but not effective at picking out anisotropic features.

The second issue which must be confronted is the dimensionality of the filter response space. It has traditionally been thought that many filters at different orientations and scales are necessary to extract rich features. However,a low dimensional filter bank is preferable because class distributions must be learnt from a finite, and often quite small, amount of training data. Thus, the curse of dimensionality forces a compromise between the quality

of features extracted and the number of filters used.

We propose to overcome both these problems by introducing reduced sets of low dimensional, rotationally invariant filters based on taking the maximal response. Thus, the two primary objectives of this chapter are (a) to develop a statistical framework for classifying materials on the basis of their appearance in single textured images obtained under unknown viewpoint and illumination and (b) to design a filter bank which is capable of detecting and extracting good features from a diverse set of classes but which is nevertheless low dimensional and rotationally invariant.

In our proposed framework, textures are modelled by the joint distribution of filter responses. This distribution is represented by texton (cluster centre) frequencies, and textons and texture models are learnt from training images. Classification of a novel image proceeds by mapping the image to a texton distribution and comparing this distribution to the learnt models using the chi-square statistic. This enables the classification of materials from single images while representing each texture class by a small set of models.

The chapter is organised as follows: in section 3.2, we study feature extraction using filter banks and discuss how such features can be used for classification. Then, the basic VZ classification algorithm is developed within a rotationally invariant framework in section 3.3. The clustering, learning and classification stages of the algorithm are described, and the performance of different filter sets is compared. The sets include the rotationally invariant filter bank of Schmid [Schmid, 2001], the discriminative filter bank of Leung and Malik [Leung and Malik, 2001], and the rotationally invariant sets developed here based on taking the maximal filter responses. Performance is assessed by classifying all the materials present in the Columbia-Utrecht (CUReT) database [Dana et al., 1999] Preliminary versions of these results appeared

in [Varma and Zisserman, 2002a, Varma and Zisserman, 2002b, Varma and Zisserman, 2005].

3.2 Filter response features

Convolution with a filter bank can be viewed in two ways. It can either be seen as extracting certain frequencies from the image signal or, more appropriately in our case, as detecting and matching the filter pattern in the image. In this match filter paradigm, the strength of the filter response at an image patch is an indicator of the similarity of the patch to the filter. Assuming intensity mean normalisation, the filter responds most strongly to patches which are scalar multiples of itself and responds least strongly to patches which are orthogonal to it (for which the filter response is zero). This is perhaps best seen by noting that filtering is identical to taking the dot product and therefore convolving an image with a filter is equivalent to projecting all the patches in the image onto the vector representation of the filter. Thus,

$$\text{Image} \star \text{Filter} \equiv \mathbf{F}_{1 \times N_s} \mathbf{I}_{N_s \times N_p} \tag{3.1}$$

where $\mathbf{I}$ is a matrix of all the overlapping patches in the image and has dimensions support size of filter (N_s) times number of patches (N_p), $\mathbf{F}$ is the vector representation of the filter obtained by row re-ordering and $\star$ represents convolution.

The general form of (3.1) also shows how a set of filters can be designed to select and enhance information present in an image. If it were known which features were good for discrimination then a bank of filters could be designed to match those features. Research on pre-attentive texture discrimination and psychophysics has concluded that textures can be characterised

by counting the frequency of occurrence of basic texture primitives (*textons*) such as bars, edges, spots and rings [Fogel and Sagi, 1989, Malik and Perona, 1990, Leung and Malik, 2001]. Thus, filters designed to detect such features at multiple scales and orientations can be expected to do quite well. Such match filter banks are often implemented using families of Gabors or Gaussians and their derivatives.

Figure 3.1 illustrates how such filters can be used to distinguish between two texture classes. The Laplacian of Gaussian (LOG) filter can be used to "count" the number of holes or spots in a textured image. Similarly, the first

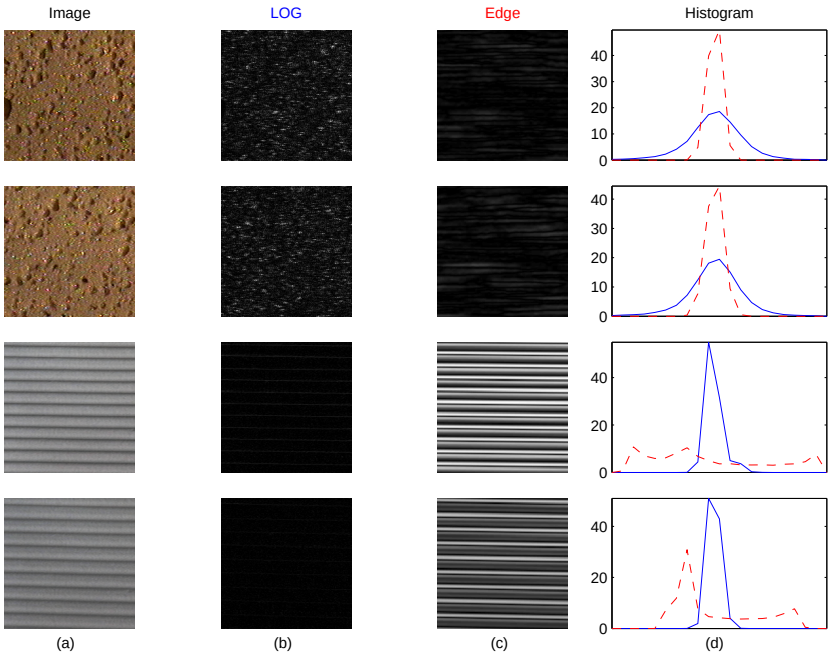


Figure 3.1: The images in (a) are convolved with a Laplacian of Gaussian (LOG) filter and the responses are shown in (b). Notice how the filter reacts strongly to the holes and spots in the sponge but almost not at all to the edges of the ribbed paper. Similarly, (c) is the response of the images to a horizontal edge filter (implemented as the first derivative of a Gaussian). Again, note that the filter has a strong preference for edges over spots. The frequency distributions of the filters (solid blue for the LOG and dashed red for the edge) are shown in (d) and can be used to distinguish between the two classes.

derivative of a Gaussian can be used to count the number of edges. As is shown by the frequency distribution, either of these statistics is sufficient for distinguishing between the two texture classes. For example, the classification rule might be that if there are more than fifty holes detected then the image must be that of sponge otherwise it must be ribbed paper. Or, that if there are more than a hundred horizontal edges detected then the image is of ribbed paper otherwise of sponge. Thus, it is possible to classify textures on the basis of just the thresholded frequency response of well selected filters.

As one moves to more complex situations with many texture classes such

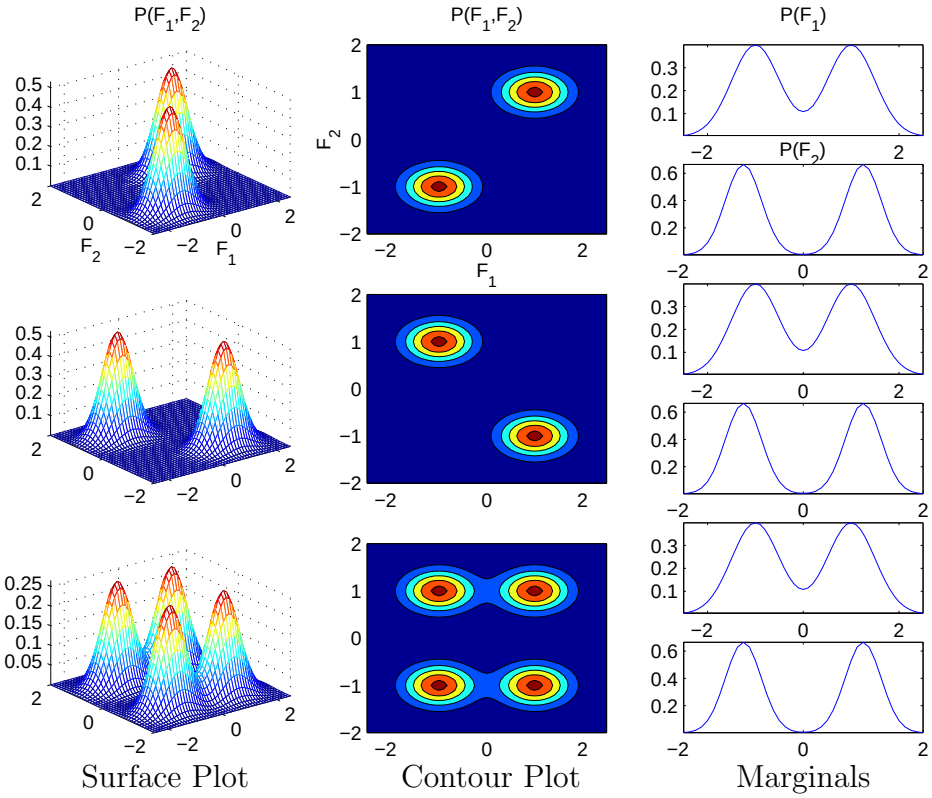


Figure 3.2: All three joint distributions have identical marginals. For classification purposes, it is therefore preferable to store the joint PDF rather than the marginals. In fact, this is often necessary when there are many texture classes and there is a significant overlap amongst the marginals.

simple classification rules no longer work. While the frequency distribution of filter responses still contains information sufficient for classification, we must now look at the joint distribution of responses rather than look at filter marginals singly. This is illustrated in figure 3.2. Obviously, care must be taken as to how this joint probability distribution function (PDF) is represented as different representations have different advantages and lead to different classification algorithms. The issue of representation is explored in detail in chapter 5. In this chapter, we will represent the distribution using *textons*.

Textons can be thought of as the basic building blocks for a given texture, i.e. the texture can be thought to have been generated by the repeated overlaying (either stochastic or periodic) of its textons. For example, the sponge texture shown in figure 3.3, can be thought of as being generated by a handful of textons, some modelling the brown background while others represent the different types of holes present. In [Leung and Malik, 2001], Leung and Malik made an important innovations and gave an operational definition of

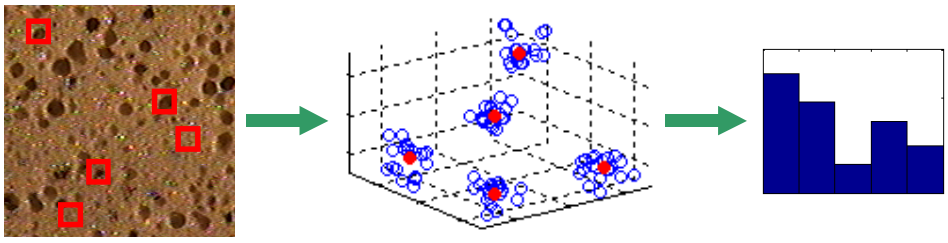


Image Filter responses Texton frequencies

Figure 3.3: A texture can be thought of as being generated by the repeated overlaying of its textons. For instance, the sponge texture can be thought of as being generated by first tiling the brown background textons and then overlaying the hole textons. The textons can be automatically determined by the cluster centres in filter response space. Texton frequencies then represent a count of how many basic texton primitives of each type are needed to characterise a texture.

a texton based on filters and clustering. They defined textons to be cluster centres in filter response space. The intuition being that most textures are generated by a finite, and small, texton vocabulary and that all other filter responses are just noisy variations of these textons (see figure 3.3). The plausibility of the hypothesis is illustrated in figure 1.7 which shows three textures artificially synthesised using a small vocabulary of textons. Note that the synthesised textures are almost indistinguishable from the originals.

3.3 The VZ algorithm

In this section, we develop the basic VZ algorithm for texture classification using features extracted by filter banks. As is customary amongst weak

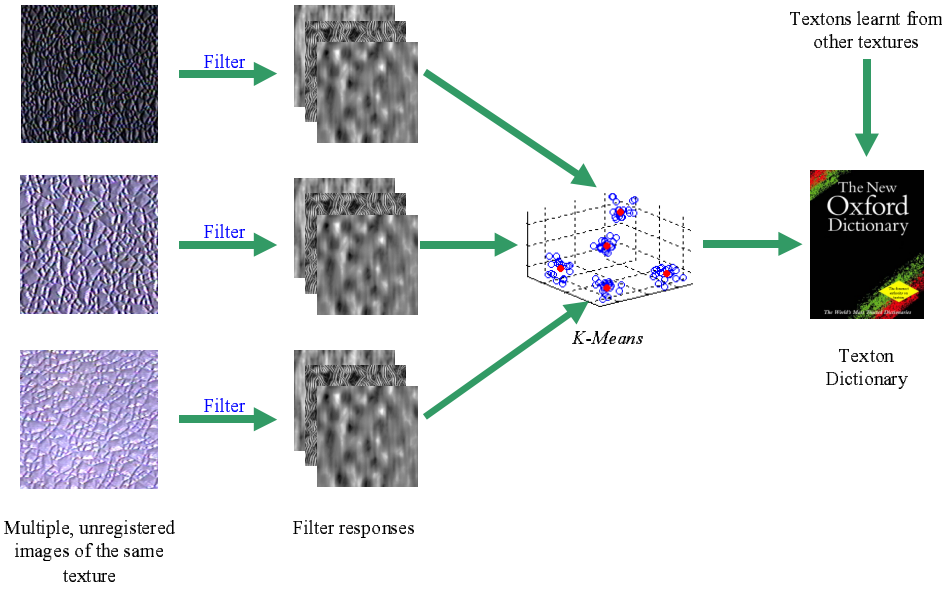


Figure 3.4: Learning stage I (generating the texton dictionary): Multiple, unregistered images from the training set of a particular texture class are convolved with a filter bank. The resultant filter responses are aggregated and clustered into textons using the *K-Means* algorithm. Textons from different texture classes are combined to form the texton dictionary.

classifiers, the algorithm is divided into a learning stage and a classification stage.

The first part of the learning stage consists of generating a texton dictionary. This is done, in turn, for every texture class by choosing certain training images and convolving each of them individually with a filter bank to generate filter responses (see figure 3.4). The responses are aggregated together and exemplars (textons) chosen via *K-Means* clustering [Duda et al., 2001]. Finally, all the textons learnt from all the different classes are brought together to form a single texton dictionary.

The choice of clustering each of the textures separately is made so that important texton primitives can be learnt from each class. If all the textures had been clustered together to learn the dictionary in one shot, only those

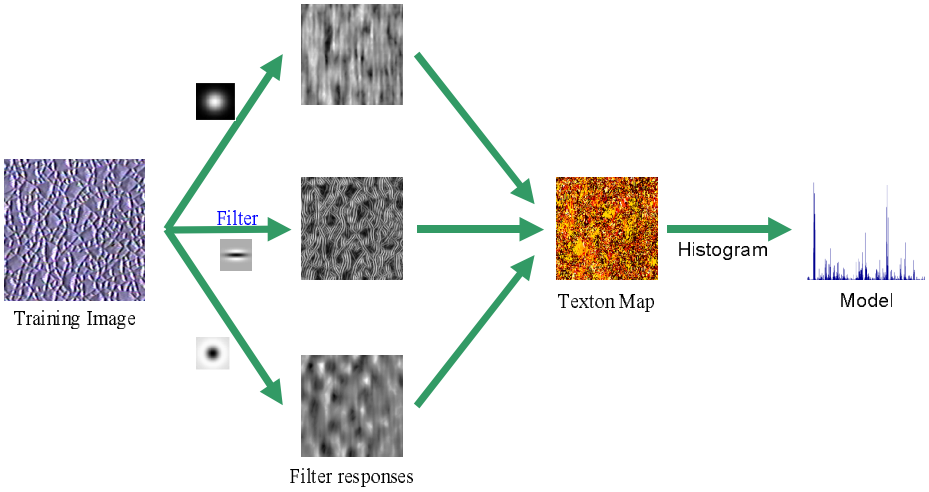


Figure 3.5: Learning stage II (model generation): Given a training image, its corresponding model is generated by first convolving it with a filter bank and then labelling each filter response with the texton which lies closest to it in filter response space. The histogram of textons, i.e. the frequency with which each texton occurs in the labelling, forms the model corresponding to the image.

textons would be selected which occur frequently across many classes rather than those textons which are unique to particular textures. It is also necessary to use multiple images from each class while generating the dictionary so as to learn the different types of textons that might be required to characterise a texture as viewpoint and illumination vary. However. it is not necessary for these images to be registered in order to learn the variation.

The next step is to learn models to characterise the various texture classes. Models are generated by first taking certain selected training images and labelling each of their filter responses with the texton that lies closest to it in filter response space. Thus each of the training images is vector quantized

```
1: function f̄=NormaliseResponses(f)
2: % f - the input filter responses generated via convolution
3: % f̄ - the output filter responses normalised according to (3.3)
4:
5:    for each x ∈ {(x, y)} do
6:       L(x) ← ‖f(x)‖_2
7:       f̄(x) ← f(x) [log (1 + L(x)/0.03)] /L(x)
8:    end
9:    return f̄
10: end
11:
12: function F̄=NormaliseFilters(F)
13: % F - the input filter bank
14: % F̄ - the output filter bank made mean zero and L_1 normalised
15:
16:    for i = 1 to N_Filters do
17:       F̄_i ← F_i − μ_{F_i}        % Make each filter mean zero
18:       F̄_i ← F̄_i / ‖F̄_i‖_1       % L_1 Normalise
19:    end
20:    return F̄
21: end
```

Algorithm 3.1: Pseudo code for normalising filter banks and filter responses during learning and classification. More details of the normalisation procedures are given in subsection 3.3.2.

into a texton map. The histogram of texton frequencies of a map is then used to form a model corresponding to the particular training images (see figure 3.5). Algorithms 3.1 and 3.2 summarises the functions for texton

function $\mathbf{T}$=LearnTextons($\mathcal{T}, \mathbf{F}, K$)
% $\mathcal{T}$ - the subset of training images from which textons are to be learnt
% $\mathbf{F}$ - the selected filter bank
% K - the number of textons to be learnt from each class
% $\mathbf{T}$ - the resultant texton dictionary

  $\mathbf{T} \leftarrow [\,]$ % Set $\mathbf{T}$ to the empty matrix
  $\mathbf{I} \leftarrow (\mathbf{I} - \mu_I)/\sigma_I \quad \forall \mathbf{I} \in \mathcal{T}$ % Normalise images
  $\mathbf{F} \leftarrow$ NormaliseFilters($\mathbf{F}$) % Normalise filter bank
  for $i = 1$ to $N_{Classes}$ **do**
    $\mathcal{F} \leftarrow \{\, \mathbf{f}(\mathbf{x}) \mid \mathbf{f} = \text{NormaliseResponses}(\mathbf{I} \star \mathbf{F}) \wedge \mathbf{I} \in \mathcal{T} \wedge \text{Class}(\mathbf{I}) = i \,\}$
    $\boldsymbol{\mu} \leftarrow$ KMeans($\mathcal{F}, K$) % Learn K textons from each class
    $\mathbf{T} \leftarrow$ Concatenate($\mathbf{T}, \boldsymbol{\mu}$) % Append textons to dictionary
  end
  return $\mathbf{T}$
end

function $\mathbf{M}$=LearnModels($\mathcal{T}, \mathbf{F}, \mathbf{T}$)
% $\mathcal{T}$ - the subset of training images from which models are to be learnt
% $\mathbf{F}$ - the selected (and normalised) filter bank
% $\mathbf{T}$ - the texton dictionary generated during the first stage of learning
% $\mathbf{M}$ - the learnt models

  $\mathbf{F} \leftarrow$ NormaliseFilters($\mathbf{F}$) % Normalise filter bank
  for each $\mathbf{I} \in \mathcal{T}$ **do**
    $\mathbf{I} \leftarrow (\mathbf{I} - \mu_I)/\sigma_I$ % Normalise image
    $\mathbf{f} \leftarrow$ Normalise($\mathbf{I} \star \mathbf{F}$) % Generate normalised filter responses
    $t(\mathbf{x}) \leftarrow argmin_i \, ||\mathbf{f}(\mathbf{x}) - \mathbf{T}_i||_2$ % Generate texton map
    $\mathbf{M}_{Ii} \leftarrow \sum_{\mathbf{x}} \delta(t(\mathbf{x}) - i) \quad \forall 1 \leq i \leq \#\mathbf{T}$
    $\mathbf{M}_{Ii} \leftarrow \mathbf{M}_{Ii} / \sum_j \mathbf{M}_{Ij}$ % Compute normalised texton histogram
  end
  return $\mathbf{M}$
end

Algorithm 3.2: An algorithmic description of the functions for texton dictionary generation and model generation that are used during learning.

dictionary generation and model generation that are used during learning.

In our framework, a texture is characterised by multiple models which account for the inter class variation due to imaging conditions. There are two important advantages in using this formulation. Firstly, it sets the ground for the classification of *single* novel images as opposed to the multiple image classification framework of [Leung and Malik, 2001]. Secondly, allowing for multiple models per texture class ensures that the PDF of filter responses is not "mixed" or "confused" by the variations in imaging conditions as would have happened if only a single model had been used per class [Konishi and Yuille, 2000, Schmid, 2001]. Though, it should be noted that as Bayes' theorem tells us, having a sinlge model would be sufficient if we were attempting only individual pixel classification (as is the case in [Konishi and Yuille, 2000]) rather than entire image classification.

In the classification stage, a similar procedure is followed to build the his-

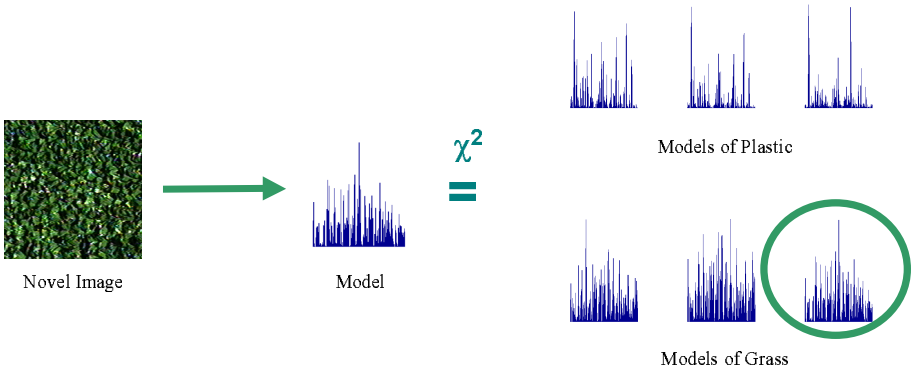


Figure 3.6: Classification stage: A novel image is classified by forming its histogram and then using a nearest neighbour classifier to pick the closest model to it (in the χ^2 sense). The novel image is declared as belonging to the texture class of the closest model.

togram corresponding to the texton map of the novel image. This histogram is then compared with the models learnt during training and is classified on the basis of the comparison (see figure 3.6). A nearest neighbour classifier is used and the χ^2 statistic [Press et al., 1992] defined as

$$\chi^2(P_{Model}, P_{Novel}) = \sum_i \frac{(P_{Model_i} - P_{Novel_i})^2}{P_{Model_i} + P_{Novel_i}} \tag{3.2}$$

employed to measure distances. Algorithm 3.3 summarises the classification function in pseudo code. More details, both of the algorithmic steps as well as the filter banks used, are given in the following subsections.

```
1: function C=NNClassify(I, F, T, M)
2: % I - the novel image to be classified
3: % F - the selected filter bank
4: % T - the texton dictionary generated during learning
5: % M - the models characterising the various texture classes
6: % C - the class allocated to the novel image
7:
8:    I ← (I − μ_I)/σ_I                          % Normalise image
9:    F ← NormaliseFilters(F)                    % Normalise filter bank
10:   f ← NormaliseResponses(I ⋆ F)              % Get normalised filter responses
11:   t(x) ← argmin_i ||f(x) − T_i||_2           % Generate texton map
12:   N_i ← Σ_x δ(t(x) − i)                      ∀ 1 ≤ i ≤ #T
13:   N_i ← N_i / Σ_j N_j                        % Compute normalised histogram
14:   M* ← argmin_{M_I} χ²(M_I, N)               % 1NN classification
15:   C ← ClassOfModel(M*)
16:   return C
17: end
```

Algorithm 3.3: Pseudo code for the classification of a novel image using nearest neighbour matching of the χ^2 statistic between model and novel histograms.

3.3.1 Rotationally invariant filters

In this subsection, we introduce the rotationally invariant filter sets that are used in the VZ algorithm. We also describe two other filter sets that will be used in classification comparisons in subsection 3.3.4. The aspects of interest are the dimension of the filter space, and whether the filter set is rotationally invariant or not.

The filter sets that will be compared are: those of Leung and Malik [Leung and Malik, 2001] which are not rotationally invariant; those of Schmid [Schmid, 2001] which are; and reduced sets of filters based on taking the maximum response (which are again rotationally invariant). Parameters of the various filter sets are listed in table 3.1. Filter sets will be assessed by their classification performance using textons clustered in their response spaces.

Filters	Dim.	Parameters
S	13	13 Gabor like filters with $(\sigma, \tau) = \{$(2,1), (4,1), (4,2), (6,1), (6,2), (6,3), (8,1), (8,2), (8,3), (10,1), (10,2), (10,3) and (10,4)$\}$
LMS	48	18 Edge and 18 Bar filters with - Scales $(\sigma_x, \sigma_y) = \{(1,3), (\sqrt{2}, 3\sqrt{2}), (2,6)\}$ - Orientations $\theta = \{0, 30, 60, 90, 120, 150\}$ 4 Gaussian filters at scales $\sigma = \{1, \sqrt{2}, 2, 2\sqrt{2}\}$ 8 LOG filters at scales σ and 3σ
LML	48	18 Edge and 18 Bar filters with - Scales $(\sigma_x, \sigma_y) = \{(\sqrt{2}, 3\sqrt{2}), (2,6), (2\sqrt{2}, 6\sqrt{2})\}$ - Orientations $\theta = \{0, 30, 60, 90, 120, 150\}$ 4 Gaussian filters at scales $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}$ 8 LOG filters at scales σ and 3σ
BFS	38	18 Edge and 18 Bar filters with - Scales $(\sigma_x, \sigma_y) = \{(1,3), (2,6), (4,12)\}$ - Orientations $\theta = \{0, 30, 60, 90, 120, 150\}$ A Gaussian filter at scale $\sigma = 10$ A LOG filter at scale $\sigma = 10$

Table 3.1: The parameters of the various filter banks.

The Leung-Malik (LM) set

The LM set is a multi scale, multi orientation filter bank with 48 filters. It consists of first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36; 8 Laplacian of Gaussian filters; and 4 Gaussians. The filters are shown in figure 3.7 and their parameters are listed in table 3.1. We consider two versions of the LM filter bank. In LM Small (LMS), the filters occur at basic scales $\sigma = \{1, \sqrt{2}, 2, 2\sqrt{2}\}$. For LM Large (LML), the filters occur at the larger scales $\sigma = \{\sqrt{2}, 2, 2\sqrt{2}, 4\}$.

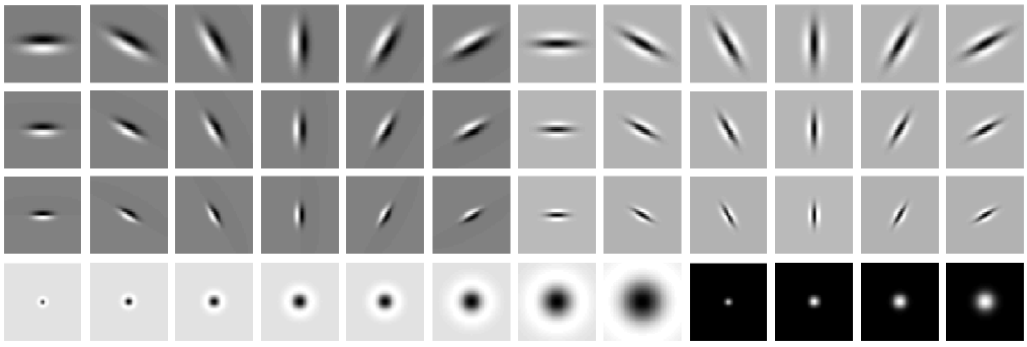


Figure 3.7: The LM filter bank has a mix of edge, bar and spot filters at multiple scales and orientations. It has a total of 48 filters - 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters.

The Schmid (S) set

The S set consists of 13 rotationally invariant filters of the form

$$F(r, \sigma, \tau) = F_0(\sigma, \tau) + cos\left(\frac{\pi \tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}}$$

where $F_0(\sigma, \tau)$ is added to obtain a zero DC component. The filters are shown in figure 3.8 and table 3.1 lists the different values that the (σ, τ) parameters can take. As can be seen all the filters have rotational symmetry.

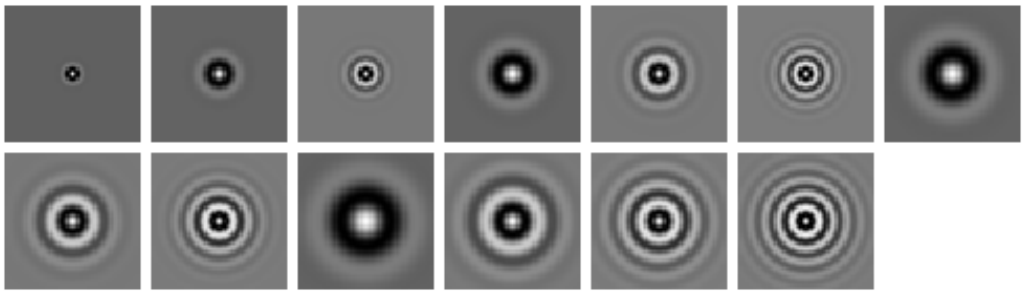


Figure 3.8: The S filter bank is rotationally invariant and has 13 isotropic, "Gabor-like" filters.

The Maximum Response (MR) sets

Each of the reduced MR sets is derived from a common Base Filter Set (BFS) which consists of 38 filters and is very similar to LM. The filters included in BFS are a Gaussian and a Laplacian of Gaussian both at a single scale (these filters have rotational symmetry), as well as anisotropic edge and bar filters at 3 scales and 6 orientations – just as in LM. The filter bank is shown in figure 3.9.

To achieve rotational invariance, we derive the Maximum Response 8 (MR8) filter bank from BFS by recording only the maximum filter response across all orientations for the two anisotropic filters. Measuring only the maximum response across orientations reduces the number of responses from 38 (6 orientations at 3 scales for 2 oriented filters, plus 2 isotropic) to 8 (3 scales for 2 filters, plus 2 isotropic). Thus, the MR8 filter bank consists of 38 filters but only 8 filter responses.

The dimensionality of the filter response space can be reduced even further by taking the maximum over both scales and orientations. This leads to the MRS4 filter bank. In it, each of the 4 different types of filters contributes only a single response. As in MR8, the responses of the two isotropic filters (Gaussian and LOG) are recorded directly. However, for each of the

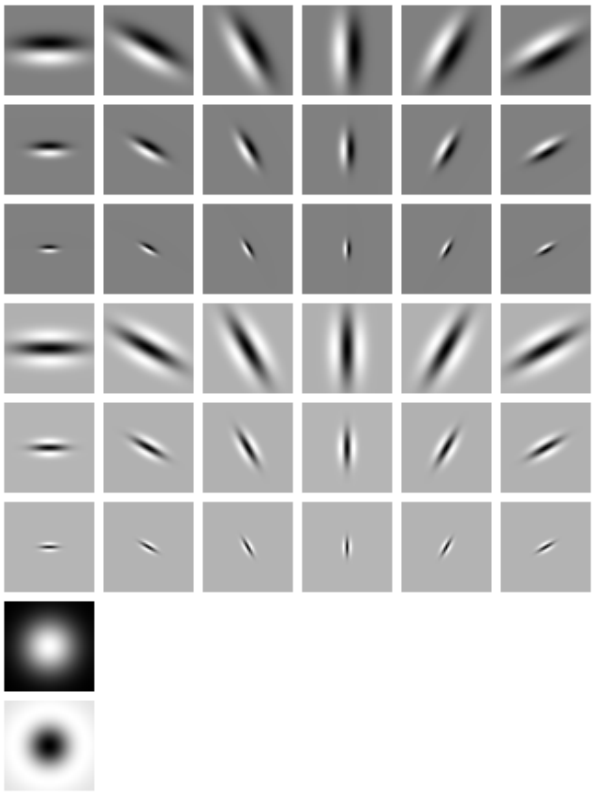


Figure 3.9: The MR filter sets: The BFS filter bank consists of 2 anisotropic filters, an edge and a bar, at 6 orientations and 3 scales and 2 rotationally symmetric ones, a Gaussian and a Laplacian of Gaussian. For the rotationally invariant MR8 filter bank, only 8 responses are recorded by taking, at each scale, the maximal response of the anisotropic filters across all orientations. Essentially, this is the maximum response of each row of the filter bank above. For the MRS4 filter bank, the maximum is taken across both scales and orientations for each type of filter. This corresponds to taking one maximum over the top three rows, another over the next three and one each for the bottom two rows.

anisotropic filters, the maximum response is taken over both orientations and scale again giving a single response per filter type. With proper normalisation, MRS4 is both rotation and scale invariant [Lindeberg, 1998].

Finally, we also consider the MR4 filter bank where we only look at filters at a single scale. Thus, the MR4 filter bank is a subset of the MR8 filter bank

where the oriented edge and bar filters occur at a single fixed scale ($\sigma_x = 4$, $\sigma_y = 12$).

The motivation for introducing these MR filters sets is twofold. The first is to overcome the limitations of traditional rotationally invariant filters which do not respond strongly to oriented image patches and thus do not provide good features for anisotropic textures. Since the MR sets contain both isotropic filters as well as anisotropic filters at multiple orientations they are expected to generate good features for all types of textures. Additionally, unlike traditional rotationally invariant filters, the MR sets are also able to record the angle of maximum response. This enables us to compute higher order co-occurrence statistics on orientation and such statistics may prove useful in discriminating textures which appear to be very similar. We return to this in the next chapter in subsection 4.2.2.

The second motivation arises out of a concern about the dimensionality of the filter response space. Quite apart from the extra processing and computational costs involved, the higher the dimensionality, the harder the clustering problem. In general, not only does the number of cluster centres needed to cover the space rise dramatically, so does the amount of training data required to reliably estimate each cluster centre. This is mitigated to some extent by the fact that texture features are sparse and can lie in lower dimensional subspaces. However, the presence of noise and the difficulty in finding and projecting onto these lower dimensional subspaces can counter these factors. Therefore, it is expected that the MR filter banks should generate more significant textons not only because of improved clustering in a lower dimensional space but also because rotated features are correctly mapped to the same texton.

3.3.2 Pre-processing

The following pre-processing steps are applied before going ahead with any learning or classification. First, before convolving with any of the filter banks, a central 200×200 texture region is cropped and retained from each of the images selected from the CUReT database and the extraneous background discarded. All processing is done on these cropped regions and they are converted to grey scale and intensity normalised to have zero mean and unit standard deviation. This normalisation gives invariance to global (i.e. across the entire region) affine transformations in the illumination intensity.

Second, within each bank, every filter is made mean zero. It is also L_1 normalised so that the responses of all filters lie roughly in the same range. In more detail, every filter F_i is divided by $\|F_i\|_1$ so that the filter has unit L_1 norm. This helps vector quantization, when using Euclidean distances, as the scaling for each of the filter response axes becomes the same [Malik et al., 2001]. Note that dividing by $\|F_i\|_1$ also scale normalises [Lindeberg, 1998] the Gaussians (and their derivatives) used in the filter banks.

Third, following [Malik et al., 2001] and motivated by Weber's law, the filter response at each pixel $\mathbf{x}$ is (contrast) normalised as

$$\mathbf{F}(\mathbf{x}) \leftarrow \mathbf{F}(\mathbf{x}) \left[\log\left(1 + L(\mathbf{x})/0.03\right)\right] / L(\mathbf{x}) \tag{3.3}$$

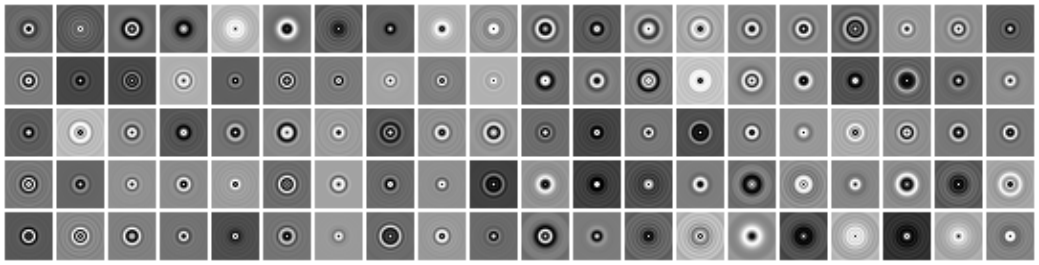
where $L(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2$ is the magnitude of the filter response vector at that pixel. This was emperically determined to lead to better classification results.

3.3.3 Textons by clustering

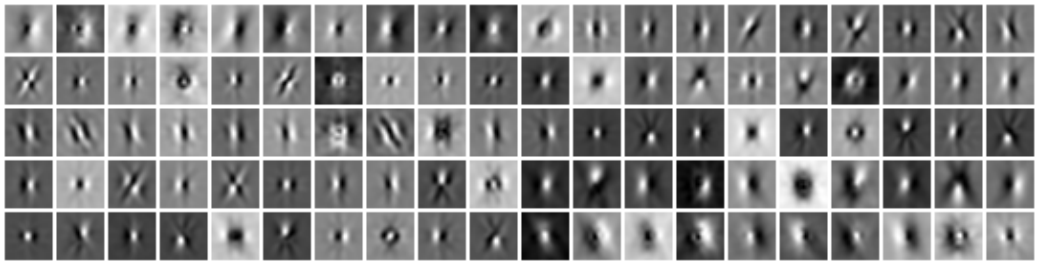
We now consider clustering the filter responses in order to generate a texton dictionary. This dictionary will subsequently be used to define texture models

based on texton frequencies learnt from training images.

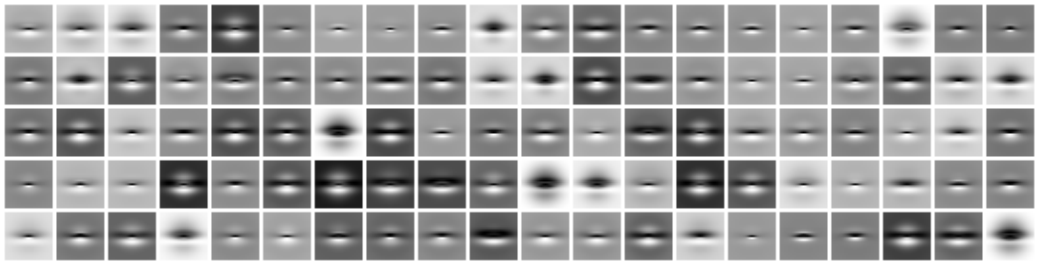
For each filter set, we adopt the following procedure for computing a texton dictionary: A selection of 13 images is chosen randomly for each texture (these images sample the variations in illumination and viewpoint), the filter responses over all these images are aggregated, and 10 texton cluster centres computed using the standard *K-Means* algorithm [Duda et al., 2001]. The learnt textons for each texture are then collected into a single dictionary. For example, if there are 5 texture classes then the dictionary will contain



(a) S Textons



(b) LM Textons



(c) MR8 Textons

Figure 3.10: The image patches (recovered by the pseudo inverse) corresponding to the first 100 textons learnt from 20 training textures using 13 images per texture: (a) S textons. (b) LM textons. (c) MR8 textons. Note that the LM textons are not rotationally symmetric.

50 textons. Examples of the textons for the S, LMS and MR8 filter banks are shown in figure 3.10.

Our clustering task is considerably simpler than that of Leung and Malik, and Cula and Dana (who use essentially the same filter bank) as we are able to cluster in low, 4 and 8, dimensional spaces. This compares to 13 dimensional for S, and 48 dimensional for LM (we are not considering 3D textons at this point where the dimensionality is 960).

Concerning the rotation properties of the LM and MR textons, consider a texture and an (in plane) rotated version of the same texture. Corresponding features in the original and the rotated texture will map to the same point in MR filter space, but to different points in LM. It is therefore expected that more significant clusters will be obtained in the rotationally invariant case. Secondly, for the LM filter set, which is not rotationally invariant, it would be expected that its textons can not classify a rotated version of a texture unless the rotated version is included in the training set (both of these points

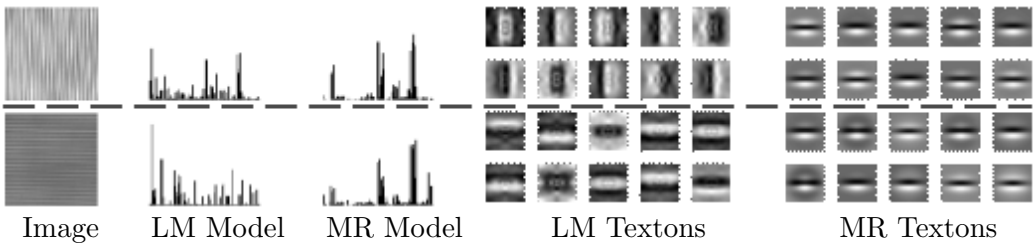


Figure 3.11: Classification of rotated textures. Two rotated images of Ribbed Paper have been taken from the CUReT database (texture numbers 38 and 38B) and their corresponding models generated using the LMS and MR4 filter banks. Note that the MR models are very similar while the LM models are not. Therefore, in the case of MR, it is expected that by having one image present in the training set the other will be classified correctly. However, this will not hold true for LM as its models are quite dissimilar. Also note, that since the LM filter bank is not rotationally invariant, the textons that are generated by the two images are rotated copies of each other while, for MR, they are essentially the same.

are demonstrated in figure 3.11).

This establishes that there is an advantage in being rotationally invariant as rotated versions of the same texture can be represented by one histogram, while several are required for the LM textons. However, there is still the possibility that rotation invariance has the disadvantage that two different textures (which are not rotationally related) have the same histogram. We address this point next, where we compare classification rates over a variety of textures.

3.3.4 Experimental setup and classification results

In this subsection, the performance of the basic VZ classification algorithm is assessed on the Columbia-Utrecht database (the database is described in detail in subsection 2.3.1 of the literature review). Three experiments are performed to compare texture classification rates over 92 images for each of 20, 40 and 61 texture classes respectively. The first experiment, where images from 20 textures are classified, corresponds to the setup employed by Cula and Dana [Cula and Dana, 2004]. The second experiment, where 40 textures are classified, is modelled on the setup of Leung and Malik [Leung and Malik, 2001]. In the third experiment, *all* 61 textures present in the Columbia-Utrecht database are classified. The 92 images are selected as follows: for each texture in the database, there are 118 images where the viewing angle θ_v is less than 60 degrees. Out of these, only those 92 are chosen for which a sufficiently large region could be cropped across all texture classes.

Each experiment consists of three stages: texton dictionary generation; model generation, where texture models are learnt from training images; and, classification of novel images. The 92 images for each texture are partitioned into two, disjoint sets. Images in the first (training) set are used for dictio-

nary and model generation, classification accuracy is only assessed on the 46 images for each texture in the second (test) set.

A model is generated from each of the 46 training images per texture by vector quantizing the image's filter responses into textons and then building the corresponding texton frequency distribution (histogram). Thus, each texture class is represented by a set of 46 histograms. An image from the test set is classified by forming its histogram and then choosing the closest model histogram learnt from the training set. The distance function used to define closest is the χ^2 statistic.

In all three experiments we follow both [Cula and Dana, 2004] and [Leung and Malik, 2001] and learn the texton dictionary from 20 textures (using the procedure outlined before in subsection 3.3.3). The particular textures used are specified in figure 7 of [Leung and Malik, 2001].

In the first experiment, 20 novel textures are chosen (see figure 19a in [Cula and Dana, 2004] for a list of the novel textures) and $20 \times 46 = 920$ novel images are classified in all. In the second experiment, the 40 textures specified in figure 7 of [Leung and Malik, 2001] are chosen and a total of $40 \times 46 = 1840$ novel images classified. Finally, in the third experiment, all 61 textures in the Columbia-Utrecht database are classified with there being a total of $61 \times 46 = 2806$ test images. The results for all three experiments are presented in table 3.2.

Discussion

Two points are notable in these results. First, the MR8 filter bank achieves similar performance to LML and BFS. It incorrectly classifies 2 and 5 more images than LML and BFS respectively when 20 textures are present and correctly classifies 9 more images when there are 61 textures (note, however,

			# of texture classes		
Filters	Dim.	Inv.	20	40	61
S	13	R	96.30%	95.27%	94.62%
LMS	48	N	96.08%	93.75%	93.44%
LML	48	N	98.04%	**96.47%**	96.08%
BFS	38	N	**98.37%**	96.36%	96.08%
MR8	08	R	97.83%	96.41%	**96.40%**
MR4	04	R	94.13%	92.07%	90.73%
MRS4	04	SR	96.41%	94.08%	93.26%

Table 3.2: Comparison of the classification rates for varying number of texture classes for each of the seven filter sets. In all cases, the dictionary used has 200 textons learnt from 20 textures and there are 46 models per texture class. Key: R - Rotational invariance, S - Scale invariance, N - No invariance.

that none of these differences is statistically significant). This indicates that a rotationally invariant descriptor is not a disadvantage and that salient information for classification is not being lost. The reason why the non invariant filters LML and BFS do so well is because there is no significant in-plane rotation within the textures of the CUReT database. This is easily seen if we repeat the first experiment (where there are 20 classes) keeping all the training images but rotating all the test images by 90°. In this case, the performance of the LML and BFS filter banks drop dramatically to 33.70% and 26.85% respectively, but the performance of MR8 remains unaffected at 97.83%. Second, the fact that MR8 does better than S is also evidence that it is detecting better features, for both isotropic and anisotropic textures, and that clustering in a lower dimensional space can be advantageous. The MR4 filter bank loses out because it only contains filters at a single scale and hence can't extract such rich features (MRS4 is a more viable alternative). What is also very encouraging with these results is that as the number of texture classes increases there is only a small decrease in the accuracy of the classifier.

3.4 Conclusions

In this chapter, filter banks have been used to tackle the problem of texture classification. The basic VZ algorithm was introduced and it was demonstrated how single images could be classified without requiring any information about their imaging conditions. This is a substantial improvement over the previous work of Leung and Malik which required multiple images obtained under known conditions.

The use of filter banks was motivated both by arguments from invariance as well as those from feature extraction which claim that a classifier must extract features at many different scales and orientations to be successful. While it is impossible to find one single filter bank which will completely achieve both goals of invariance as well as discriminability, we introduced the low dimensional MR sets which lead to an effective compromise. The sets include anisotropic filters at multiple orientations and scales, which provide discriminability, but are nevertheless invariant to image rotations and also image rescalings (for MRS4). The MR sets have a further advantage over traditional forms of invariant filter banks as they are able to record the angle (or scale) of maximum response and use that information if necessary for classification.

Empirical results of classifying the various textures in the CUReT database were presented to validate the performance of the MR sets. In each experiment, the MR8 filters proved to be superior to S, the traditional rotationally invariant filter bank included in the comparisons. MR8's performance was also as good as LML or BFS, the non rotationally invariant filters, when no significant rotations were present in the included texture classes. However, MR8's superiority over LML and BFS was clearly brought out as soon as the camera was rotated. The set's performance remained unaffected at over 95%

while the performance of the non invariant filter banks plummeted to around 30%. Unfortunately, the performance of MRS4 was not as good because the scale parameter is, by and large, fixed for the images present in the CUReT database. However, as will be shown in the next chapter, the comparative performance of MRS4 improves when classifying using only a few models which have to cope with larger variability.

Chapter 4

Model Reduction and Algorithmic Variations

Chapter 3 introduced the basic VZ algorithm for single image texture classification using filter banks. In this chapter, we study various extensions of the VZ algorithm. The objective is to see how robust the algorithm is with change in parameters and how it is affected by different modifications. The three questions that are particularly of interest are (a) how can the number of models needed during classification be reduced and how does this affect performance? (b) what is the effect of changing the texton dictionary and the training image set? and (c) how valid is the hypothesis that first order statistics are sufficient for texture classification?

The first question is tackled in section 4.1 where we investigate methods which minimise the number of models used to characterise the various texture classes. The *K-Medoid* and *Greedy* algorithms are introduced and results compared with those of [Cula and Dana, 2004, Leung and Malik, 2001]. Section 4.2 then deals with the effect of choice of texton dictionary and training images upon the classifier. A benchmark rate is defined and size of the

texton dictionary and number of training images varied to investigate the impact on classification performance. Finally, the issue of whether information is lost by using only the first order statistics of rotationally invariant filter responses is discussed. A method for reliably measuring the relative orientation co-occurrence of textons is presented in order to incorporate second order statistics into the classification scheme.

4.1 Reducing the number of models

In this section, our objective is to reduce the number of training models required to characterise each texture class. In chapter 3, the number of models was the same as the number of training images (and in effect [Leung and Malik, 2001] used 20 models/images for every texture). Here, we want to reduce the number of models to that appropriate for each class, independent of the number of training images.

One would expect that the number of different models that are needed to characterise a texture is a function of how much the texture changes in appearance with imaging conditions, i.e. it is a function of the material properties of the texture. For example, if a texture is isotropic then the effect of varying the lighting azimuthal angle will be less pronounced than for one that is anisotropic. Thus, other parameters (such as relief profile) being equal, fewer models would be required for the isotropic texture (than the anisotropic) to cover the changes due to lighting variation. This is demonstrated in figure 4.1.

However, if we are selecting models for the express purpose of classification, then another parameter, the inter class image variation, also becomes very important in determining the number of models. For example, even if a

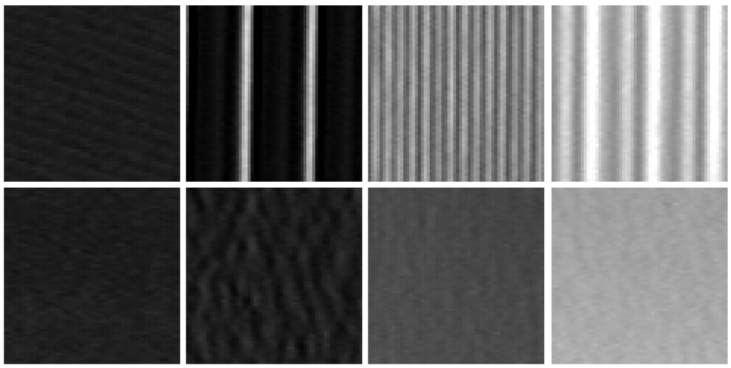


Figure 4.1: Models per texture: The top row shows four images of the same texture, Ribbed Paper, photographed under different viewing and lighting conditions. The images look very different. The bottom row shows images of Rough Paper taken under the same conditions as the images in the first row. These images don't differ so markedly because the texture doesn't exhibit surface normal effects. The consequence is that fewer models are required to represent Rough Paper over all viewpoints and lighting than Ribbed Paper.

texture varies considerably with changing imaging conditions it can be classified accurately using just a few models if all the other textures look very different from it. Conversely, if two textures look very similar then many models may be needed to distinguish between them even if they do not show much variation individually.

Broadly speaking, there are two major approaches to the problem of model reduction. In the first, various concepts from the Machine Learning literature can be used to select a subset of the models while maximising some criteria of classification and generalisation. The second approach is geometric and focuses on building descriptors invariant to imaging conditions so as to reduce the number of models needed.

4.1.1 Model selection

Many Machine Learning techniques have been developed to reduce the number of models in a classification algorithm. One of the simplest exam-

ples [Duda et al., 2001], for a nearest neighbour classifier, is to remove each model for which all the neighbouring models belong to the same class. This can be done safely as these models make no contribution in determining the classification boundaries (as can be seen from the Voronoi tessellation). However, in practice this has often been found not to lead to a substantial reduction in the number of models.

The Voronoi condensing algorithm given above is an example of a method which is *decision boundary consistent*, i.e. the reduced set of models has exactly the same decision boundaries as the original training set. Another class of nearest neighbour reduction algorithms aim to be *training set consistent* where the criteria is that the reduced set must be able to correctly classify the original training set. Examples of such algorithms are the Condensed Nearest Neighbour (CNN) method of Hart [Hart, 1968], the Reduced CNN of Gates [Gates, 1972] and the Batch CNN of Devi and Murti [Devi and Murty, 2002]. Unfortunately, most training set consistent methods can be prone to over-fitting. To counter this, *editing* methods [Wilson, 1972] have been developed to improve generalisation and should be applied before the reduction process [Dasarathy et al., 2000].

It is also possible to reduce the number of models by completely switching classifiers. For instance, Support Vector Machines [Cristianini and Shawe-Taylor, 2000, Hayman et al., 2004, Kim et al., 2002, Scholkopf and Smola, 2002], and perhaps more appropriately Relevance Vector Machines [Tipping, 2001] and Reduced Support Vector Machines [Lee and Mangasarian, 2001, Romdhani et al., 2001], are both capable of reducing the number of models while providing good generalisation. Methods have also been developed to reduce the number of models by incorporating invariance into the learning framework. In [Scholkopf et al., 1996], virtual support vectors

are generated in accordance with the desired invariance transformations to account for the effects of translation and rotation. Other approaches are to pre-process the data to make the kernels themselves invariant to input transformations [Burges, 1999] or to modify the cost function so as to obtain a decision hyperplane whose normal is orthogonal to the invariance "tangent" [Chapelle and Scholkopf, 2002].

In this subsection, we investigate two schemes for model reduction in a nearest neighbour classifier framework. Both these schemes take into account the inter and intra class image variation. Two types of experiments are performed for either method. In the first, models are selected only from the training set and classification results reported only on the test set. In the second type, the classification experiments are modified slightly so as to maximise the total number of images classified. Following [Cula and Dana, 2004], if only M models per texture are used for training, then the rest of the $46 - M$ training images are added to the 46 test images so that a total of $92 - M$ images are classified per material. For example, when classifying 61 textures, if only $M = 10$ models are used on average then a total of 82 images per texture are classified giving a total of $82 \times 61 = 5002$ test images. This is done so as to be able to make accurate comparisons with [Cula and Dana, 2004]. The texton dictionary used in all experiments is the same as the one in the previous chapter and has 200 textons.

K-Medoid algorithm

Each histogram may be thought of as a point in $\mathbb{R}^N$, where N is the number of bins in the histogram, so that the models for a particular texture class simply consist of a set of points in $\mathbb{R}^N$ space. Given a distance function between two points, in our case χ^2, the set of points corresponding to a

texture's models may be *clustered* into a few representative prototypes, and the set of points then replaced by the prototypes. There are three different choices of how the clustering can be implemented: (a) clustering can be done within each class individually without looking at the information present in other classes, (b) clustering can be done within each class but while making use of other class labels in the form of a validation set (in a fashion similar to the hierarchical algorithms of [Chang, 1974, Mollineda et al., 2002]) or (c) all the models, across all classes, can be clustered together and no special use is made of the class information. We opt for (c) due to historical reasons. This has the disadvantage of "mixing" classes within a cluster but is also potentially less prone to over-fitting.

The clustering is implemented using the *K-Medoid* algorithm. This is a standard clustering algorithm [Kaufman and Rousseeuw, 1990] where the update rule always moves the cluster centre to the nearest data point in the

	Average # of models / texture			Average # of models / texture		
filters	3	6	9	3	6	9
S	77.47%	86.05%	91.08%	75.87%	85.76%	90.65%
LMS	75.28%	85.06%	89.52%	74.89%	85.22%	89.35%
LML	77.47%	89.30%	93.43%	78.15%	88.59%	92.50%
BFS	**78.65%**	89.59%	**94.10%**	78.80%	88.59%	92.50%
MR8	77.08%	**89.88%**	93.55%	**79.35%**	**89.57%**	**93.59%**
MR4	71.07%	80.93%	86.39%	71.09%	81.85%	84.57%
MRS4	77.92%	86.63%	91.08%	77.39%	86.74%	91.09%
		(a)			(b)	

Table 4.1: Classification results for each of the filter sets when the models are automatically selected by the *K-Medoid* algorithm. In (a), the training and test sets are kept distinct while in (b) the images from the training set which are not selected as models are added to the test set and classified. Both types of experiments give very similar results, even though many more images have to be classified correctly in (b) to achieve the same performance as in (a). In all cases a dictionary of 200 textons is used and there are 20 textures being classified.

cluster, but does not merge the points as in the case of the more popular *K-Means*. The *K-Means* algorithm can only be applied to points within a texture class. It can not be applied across classes as it merges data points and thus the resultant cluster centres can not be identified uniquely with individual textures. Another problem with *K-Means* is that using the χ^2 statistic does not lead to a closed form solution for the update equations. Neither of these points is a problem for the *K-Medoid* algorithm as the cluster centres are always data points themselves. Table 4.1 lists the results of classifying 20 textures using the different filter banks with $K = 60, 120$ and 180, resulting in an average of $3, 6$ and 9 models per texture.

Using, on average, 9 models per texture class, MR8 achieves an accuracy of 93.55% in the first type of experiment (table 4.1a) and in the second type (table 4.1b) achieves 93.59% while classifying many more test images. Considering that the number of models per texture has been reduced from 46 to 9, this compares very well to the 97.83% obtained by the basic VZ algorithm (see column 1 in table 3.2). Another interesting fact is that now, the 4 dimensional filter bank MRS4 is doing better than, or at least as well as, the 13 dimensional S and the 48 dimensional LMS filter banks.

However, *K-Medoid* clustering does have the disadvantage that very similar models are aggregated into a single cluster even if they come from different texture classes. Similarly, many clusters centres, rather than just one, might be used to represent models which are spread apart even if they belong to the same texture class. Both these shortcomings can be overcome by using a greedy algorithm which prunes the list of models on the basis of classification boundaries.

Greedy algorithm

An alternative to the *K-Medoid* clustering algorithm is a greedy algorithm, based on the post-processing step of the reduced nearest neighbour rule [Gates, 1972,Toussaint, 2002], designed to maximise the classification accuracy while minimising the number of models used. The algorithm is initialised by setting the number of models equal to the number of training images available. Then, at each iteration step, one model is discarded. This model is chosen to be the one for which the classification accuracy decreases the least when it is discarded. The iterations are repeated until no more models are left. Note that while the algorithm is constrained to select models only from the training set, classification performance is being assessed on the test set. This emulates the setup of [Cula and Dana, 2004] where the model reduction algorithm has access to both training and test images for each texture class and should therefore facilitate a faithful comparison with their work. However, it must be emphasised that in real world classification, the test set is not avail-

	Average # of models / texture			Average # of models / texture		
filters	3	6	9	3	6	9
S	88.80%	96.30%	96.30%	88.37%	97.21%	98.01%
LMS	87.28%	96.09%	96.20%	86.69%	95.99%	97.83%
LML	90.65%	98.04%	98.04%	90.06%	98.49%	98.92%
BFS	92.83%	**98.37%**	**98.37%**	90.17%	**98.55%**	**99.10%**
MR8	**93.70%**	97.83%	97.83%	**90.28%**	98.14%	98.80%
MR4	85.22%	94.02%	94.24%	85.00%	93.66%	96.39%
MRS4	89.89%	96.74%	96.74%	88.20%	96.69%	98.19%
		(a)			(b)	

Table 4.2: Classification results for each of the filter sets when the models are automatically selected by the *Greedy* algorithm. In (a), the test set is kept distinct by not adding discarded models to it while in (b) the discarded models are added to the test set and classified. A dictionary of 200 textons is used in all cases and there are 20 textures being classified.

able for inspection to the training set and in such situations it is preferable to subdivide the training set further into model learning and validation sets.

Table 4.2 lists the results of classifying 20 textures using the different filter banks. It is very interesting to note that the classification accuracy obtained using 9 models can actually be better than that obtained using all 46 models (see column 1 in table 3.2). In table 4.2a, this implies that using a fewer number of models can improve performance and that the *Greedy* algorithm is good at rejecting noisy or outlier models. In table 4.2b, this also indicates that most of the training images being added to the test set are

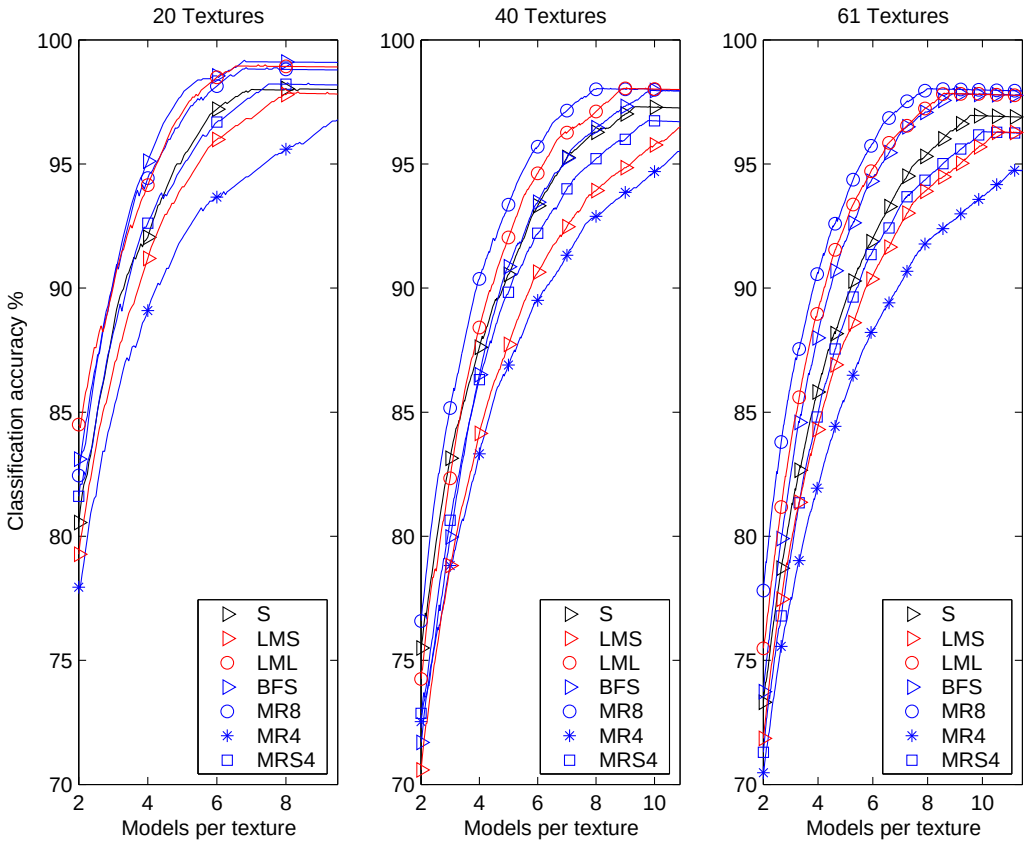


Figure 4.2: Classification rates for models selected by the *Greedy* algorithm for 20, 40 and 61 textures. In these experiments, the images from the training set which were not selected as models were added to the test set, as in table 4.2b.

being classified correctly.

Table 4.2 also shows that, once again, the 4 dimensional filter bank MRS4 is outperforming the 13 dimensional S and the 48 dimensional LMS filter banks (apart from the case of 3 and 6 models in table 4.2b when S does better). However, the reason that the non rotationally invariant filters LML and BFS are doing so well is because there are hardly any strongly oriented textures with significant rotation in the 20 classes selected. As the number of classes increases and more oriented textures are included, the performance of the rotationally invariant filter bank MR8 surpasses that of LML and BFS. This is shown in figure 4.2 which plots the classification accuracy versus number of models for each of the filter banks when classifying 20, 40 and 61 textures. As can be seen, in the case of 40 and 61 textures, the MR8 curve lies on top followed by LML, BFS. For MR8, a very respectable classification rate of over 97% correct is achieved using on an average only 9 models per texture, even when all 61 classes are included. Figure 4.3 shows the 9 textures that

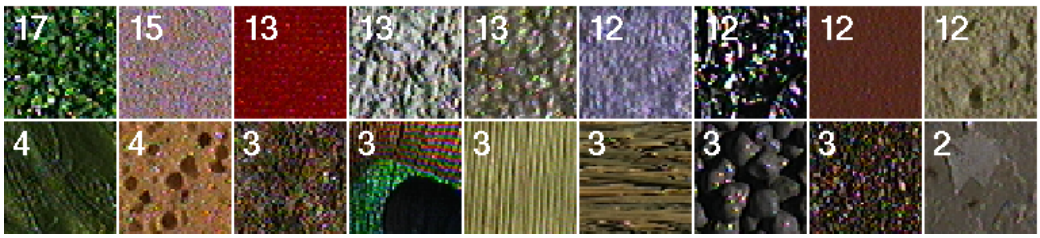


Figure 4.3: Models selected by the *Greedy* algorithm while classifying all 61 textures: The top row shows the 9 texture classes, and the corresponding number of models, that were assigned the most number of models by the *Greedy* algorithm while the bottom row shows the 9 classes that were assigned the least number of models. Moving from left to right, the textures and the number of models assigned to it are: Artificial grass (17), Sandpaper (15), Velvet (13), Plaster B (13), Rug A (13), Terrycloth (12), Aluminium Foil (12), Quarry Tile (12), White Bread (12), Lettuce Leaf (4), Sponge (4), Cracker A (3), Peacock Feather (3), Corn Husk (3), Straw (3), Painted Spheres (3), Roof Shingle (3) and Limestone (2).

were assigned the most models as well as the 9 textures that were assigned the least models while classifying all 61 textures.

Discussion

The results for both the *K-Medoid* and the *Greedy* algorithms, while using the MR8 filter bank, compare very favourably with those reported in [Cula and Dana, 2004] and [Leung and Malik, 2001]. In the case where there are 20 textures to be classified, the *K-Medoid* algorithm has a classification accuracy of 93.59% while using, on average, 9 models per texture class while the *Greedy* algorithm achieves an accuracy of 98.80%. In contrast, for the same 20 textures, Cula and Dana obtain a classification rate of 71% while using 8 models per texture class (by taking the most *significant* image from each texture and using a *manifold merging procedure*). This increases marginally to 72% if 11 models are used per texture (see figure 19b and table 4 in [Cula and Dana, 2004]). Note that the comparison is not exact since we classify only $92 - 9 = 83$ images per texture class as compared to the $156 - \{8, 11\}$ classified by Cula and Dana. Hence, [Cula and Dana, 2004] classify many more images, some of which might be hard to categorise correctly because of the oblique viewing angle.

Nevertheless, there is a significant level of difference between the performance of the *K-Medoid* and the *Greedy* algorithms on one hand and the manifold method of [Cula and Dana, 2004] on the other. This is primarily due to the fact that the methods developed here take into account both the *inter* class variation, as well as intra class variation. The models that Cula and Dana learn are general models and not geared specifically towards classification. They ignore the inter class variability between textures and concentrate only on the intra class variability. The models for a texture are

selected by first projecting all the training and test images into a low dimensional space using PCA. A manifold is fitted to these projected points, and then reduced by systematically discarding those points which least affect the "shape" of the manifold. The points which are left in the end correspond to the model images that define the texture. Since the models for a texture are chosen in isolation from the other textures, their algorithm ignores the inter class variation between textures.

For 40 textures, Leung and Malik report an accuracy rate of 95.6% for classifying multiple (20) images using, in effect, 20 models per texture class. For single image classification under *known* imaging conditions, using 4 models per texture class results in a drop in the accuracy rate to 87% (as computed for 5 test images per texture). The MR8 filter bank achieves 95.6% accuracy on the same textures using only 5.9 models per texture, and furthermore achieves 98.06% accuracy using, on average, 8.25 models per texture.

4.1.2 Pose normalisation

In this subsection we discuss some geometric approaches to model reduction. In theory, these approaches are valid only in the absence of 3D effects, i.e. for planar textures where illumination does not play a major role, and where a 3D rotation and translation of the texture is equivalent to an affine transformation of its image. However, in practice, these methods are quite robust.

The fundamental idea is to incorporate some level of geometric invariance into a model. This will ultimately allow us to be invariant to changes in the camera viewpoint and thereby reduce the number of models required to characterise a texture. The use of rotationally invariant filters is already a first step in this direction but the problem of scale still needs to be resolved

(we are ignoring perspective effects for the moment). One approach could be to extend the MR sets to take the maximum response over all affine transformations of the basic filter [Caenen and Van Gool, 2004], but that is not investigated here. Instead we focus on the method of pose normalisation.

In [Schaffalitzky and Zisserman, 2001] it was demonstrated that, provided a texture has sufficient directional variation, it can be pose normalised by maximising the isotropy of its gradient second moment matrix (a method originally suggested in [Lindeberg and Gårding, 1994]). The method is applicable in the absence of 3D texture effects. Here we investigate if this normalisation can be used to at least reduce the effects of changing viewpoint, and hence provide tighter clusters of the filter responses, or better still reduce the number of models needed to account for viewpoint change.

In detail, if the normalisation is successful, then for moderate changes in the viewing angle, two such "pose normalised" images of the same texture should differ from each other by only a similarity transformation. If there are no major scale effects, the responses of a rotationally invariant filter bank (MR or S) to these images should be much the same. A preliminary investigation shows that this is indeed the case for suitable textures.

Figure 4.4 shows results for two textures - Plaster A and Rough Plastic. Twelve images of each texture are selected to have similar photometric appearance (i.e. constant illumination conditions), but monotonically varying viewing angle. The graph shows the χ^2 distance between the texton histogram of one of the images (selected as the model image) and the rest, before and after pose normalisation. As can be seen, the χ^2 distance is reduced for the pose normalised images. This in turn translates to better classification as well. On experiments on 4 textures, using the same 12 image set and one model per texture, the classification rate increased from 81.81% before pose

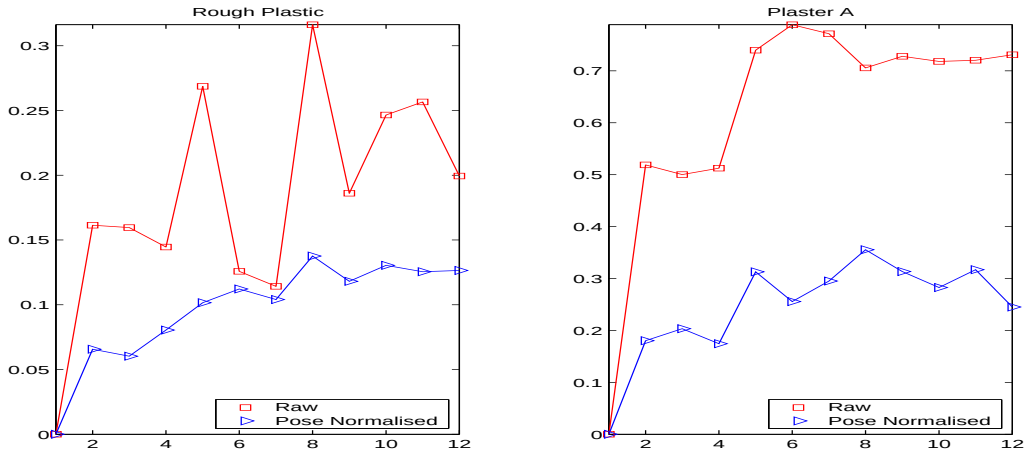


Figure 4.4: The effect of pose normalisation on a set of 12 images for two textures: Rough Plastic and Plaster A. The 12 images have been sorted according to increasing viewing angle and this is represented on the X axis. The Y axis is the χ^2 distance between the model image and the given image. The pose normalised images consistently have a reduced χ^2 distance which translates into better classification.

normalisation to 93.18% afterwards.

One drawback of this method is that the proposed normalisation is global rather than local. Not only would local normalisation be more robust but it would also allow the method to be extended to textures which are not globally planar but which can be approximated as being locally planar. Realizing this, [Lazebnik et al., 2003a, Lazebnik et al., 2003b] proposed alternative methods of generating local, affine invariant, texture features. In their framework, certain interest regions are first detected in texture images using blob and corner detectors. A characteristic scale is found and the interest regions are pose normalised locally rather than globally. Spin images are then used instead of filter banks to generate rotationally invariant features for each region. Their results are very encouraging though no direct comparison is possible as their experiments are not carried out on the CUReT database. One point of concern however, is the reliance on the detection of

blob and corner like interest regions as there exist many textures which do not exhibit such markings. Recently, Ravela [Ravela, 2004] has proposed an alternative affine invariant operator which overcomes this difficulty and can be applied at almost every pixel in the image. This is achieved by looking at not just the second order Laplacian but also at the first derivative of a Gaussian (which matches edges) as well as higher order derivatives (which match corners, etc).

4.2 Algorithmic variations

In this section, the various generalisations and modifications that can be made to the basic VZ classification algorithm are investigated. In subsection 4.2.1, we study the effect of some of the more important parameters on our classifier. In particular, the effect of the choice of texton dictionary and training images is studied. We also look at how scaling the images impacts performance. Finally, the issue of whether information is lost by using just the first order statistics of rotationally invariant filter responses is discussed in section 4.2.2. A method for reliably measuring relative orientation texton co-occurrence is presented in order to incorporate second order statistics into the classification scheme.

4.2.1 Varying the texton dictionary and training images

In this subsection, various parameters of the VZ algorithm are varied and the effect on the classification performance determined. We first calculate a benchmark classification rate and then vary the images in the training set and also the size of the texton dictionary to see how performance is affected.

For the benchmark case, referred to from now on as VZ Benchmark, the texton dictionary is built by learning 10 textons from each of the 61 textures (using the procedure described in subsection 3.3.3) to have a total of 610 textons. The 46 training images per texture from which the models will be generated are chosen by selecting every alternate image from the set of 92 available. Under these conditions, the MR8 filter bank achieves a classification accuracy of 96.93% using 46 models per texture for all 61 textures. On running the *Greedy* algorithm the classification accuracy increases to 98.3% using, on average, only 8 models per texture. This defines the benchmark rate.

We now investigate the effect of choice of textons on the classification performance. First the number of textons is reduced by learning 10 textons each from 31 randomly chosen textures to get a dictionary of only 310 textons. The VZ classifier was retrained and it was found that the accuracy decreased only slightly from the benchmark to 98.19%.

The number of textons in the dictionary can be further reduced by merging textons which lie very close to each other in filter response space. The texton dictionary can be pruned down from 310 to 100 by selecting 80 of the most distinct textons (i.e. those textons that didn't have any other textons lying close by) and then running *K-Means*, with $K = 20$, on the rest. This procedure entailed another slight decrease in the classification accuracy to 97.38%. These results indicate that the pruned dictionaries are still universal [Leung and Malik, 2001], i.e. texton primitives learnt from some randomly chosen texture classes can be used to successfully characterise other classes as well.

The size of the texton dictionary is now increased to see if classification improves accordingly. Table 4.3 gives a summary of the results. The best

performance is obtained with a dictionary of 2440 textons when the classification accuracy is 97.43% using 46 models per texture. Once again, these 46 models were generated by selecting every alternate image from the set of 92 available. On running the *Greedy* algorithm, the number of models used is reduced to, on average, 7.14 per texture. If the unused training images are added to the test set, the classification rate improves to 98.61%. These results will be referred to as VZ Best.

Number of	Before Greedy		After Greedy	
Textons	Classification	Models	Classification	Models
1220	97.11%	46	98.43%	7.56
1830	97.18%	46	98.49%	7.26
2440	97.43%	46	98.61%	7.14
3050	97.32%	46	98.57%	7.41

Table 4.3: The effect of increasing the size of the texton dictionary while classifying all 61 textures from the CUReT database using the MR8 filter bank.

In these experiments, we have essentially been comparing different representations of the joint probability distribution of filter responses in terms of their classification performance. A set of textons can be thought of as adaptively partitioning the space of filter responses into bins (determined by the Voronoi diagram) and a histogram of texton frequencies can be equated to a probability distribution over filter responses (this is explored in detail in chapter 5). In such a situation, the number of bins should not be too few otherwise the approximation to the true PDF will be poor nor should there be too many bins so as to prevent over-fitting.

As can be seen in table 4.3 there is a point beyond which increasing the number of textons actually decreases performance as the data is now being over fitted. This can be used to automatically select the appropriate number of textons for a given problem by partitioning the data into a training

and validation set and then choosing the texton dictionary which maximises classification on the validation set.

As regards the choice of training images, it could be argued that the results presented here are biased since the training set has been chosen by including every alternate image from the set of 92 available per texture. This issue is addressed by repeating the classification experiment but with the training images chosen randomly. The dictionary of 2440 textons from VZ Best is used and the experiment repeated 50,000 times. Figure 4.5 shows the distribution of classification results when 46 images were chosen randomly from every texture class to form the training set while table 4.4 provides a summary of the results for varying sizes of the training set. The mean classification accuracy when the 46 models were chosen randomly was 97.28% which is very similar to the 97.43% attained by VZ Best when the 46 models were chosen by including every alternate image. The standard deviation was 0.316% and the maximum accuracy attained was 98.40%. This shows that the VZ Best experimental setup is not biased.

In summary, the best classification rate achieved, while classifying all 61 textures, was 98.61% obtained when 2440 textons were used and the worst

Training images	Classification Statistics			
per texture	Mean	STD	Min	Max
46	97.28%	0.316%	95.72%	98.40%
23	94.22%	0.456%	91.97%	95.82%
12	89.02%	0.679%	85.92%	91.84%
6	80.67%	0.986%	76.46%	84.50%
3	69.70%	1.373%	63.90%	75.52%

Table 4.4: Classification statistics when the training images were chosen randomly. A dictionary of 2440 textons was used and all 61 textures were classified. In each case, the statistics were gathered over 50,000 runs of the classification experiment.

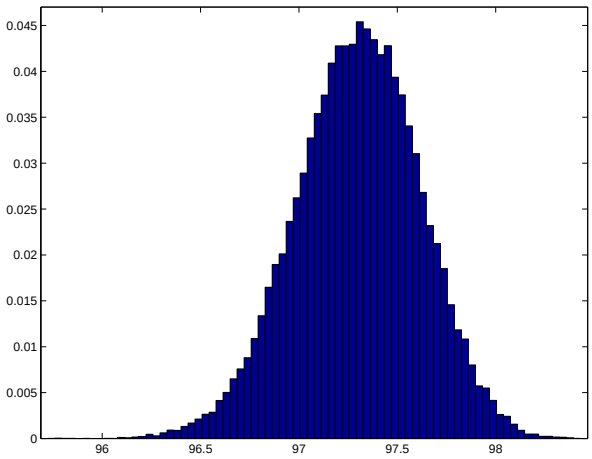


Figure 4.5: The distribution of classification percentages when 46 training images are chosen randomly per texture from the set of 92 available. The experiment was run 50,000 times with a dictionary of 2440 textons and all 61 materials in the CUReT database were classified. The mean classification accuracy was 97.28% with a standard deviation of 0.316%. The maximum was 98.4% and the minimum was 95.72%.

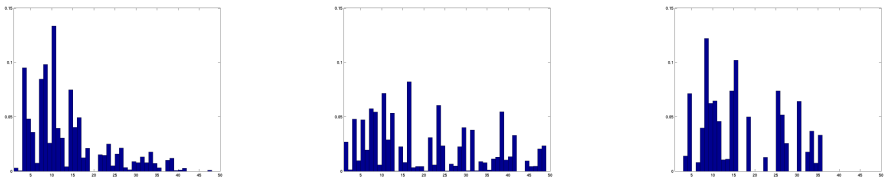
rate was 97.38% when only 100 textons were used. These results are listed in table 4.5. We can therefore conclude that our algorithm is robust and relatively insensitive to the choice of training image set and texton vocabulary with the classification rate not being affected much by changes in these parameters.

	Number of	Before Greedy		After Greedy	
	Textons	Accuracy	Models	Accuracy	Models
VZ Worst	100	95.32%	46	97.38%	9.83
VZ Benchmark	610	96.93%	46	98.30%	8.00
VZ Best	2440	97.43%	46	98.61%	7.14

Table 4.5: Benchmark, worst and best case results for varying parameters of the VZ algorithm.

Finally, a word about scale. It may be of concern that the MR4 filter bank does not have filters at multiple scales and hence will be unable to handle scale changes successfully. To test this, 25 images from 14 texture classes

were artificially scaled, both up and down, by a factor of 3. The classification experiment was repeated using the original, normal sized, filter banks and texton dictionaries. It was found that as long as models from the scaled images were included as part of the texture class definition, classification accuracy was virtually unaffected and classification rates of over 97% were achieved. However, if the choice of models was restricted to those drawn from the original sized images, then the classification rate dropped to 17%. It is evident from this that filter bank and texton vocabulary are sufficient, and it is the model that must be extended (see figure 4.6).



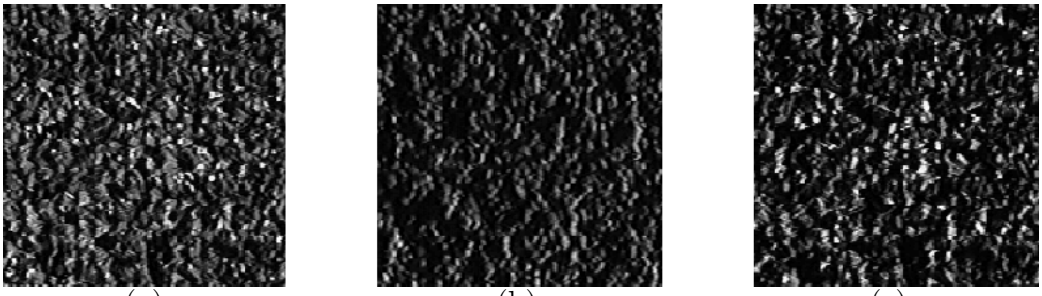
(a) (b) (c)

Figure 4.6: Scaling the data results in new models: The histogram of texton labellings of (a) the original image (b) the image scaled up by a factor of 3 and (c) the image scaled down by a factor of 3. All three models are substantially different indicating that the model must be extended.

4.2.2 Orientation co-occurrence

The classification scheme, up to this stage, has only used information about first order texton statistics (i.e. their frequency and not a measure of their co-occurrence). However, recent research into texture driven content-based image retrieval [Schmid, 2001] has shown that a hierarchical system which uses co-occurrence of textons over a spatial neighbourhood can lead to good results. Therefore, in this subsection, we investigate whether incorporating such second order statistics can improve classification performance on the CUReT database.

As was seen in the previous subsection, classification on the basis of texton frequency information alone is already very good and rates of over 97% can be achieved. What is also interesting is that, of the images that were misclassified, the correct texture class was ranked within the top 5 most of the times. Figure 4.7 shows how similar one of the misclassified novel images is to both the top ranked, but incorrect, texture model and the second ranked, but correct, model. Since the MR8 filter bank is rotationally invariant, there is the possibility that some of these misclassifications are due to two different texture classes, which are not rotationally related, being mapped to the same texton frequency distribution. Therefore, we focus on the question of whether incorporating second order texton statistics, in the form of co-occurrence of angles, can improve classification (though the method developed here is general and can also be applied to spatial co-occurrence).



(a) (b) (c)

Figure 4.7: Misclassifications: (a) is an image of Artificial Grass taken from the test set which was misclassified as (b) Pebbles. The next closest model image to (a) is (c) which belongs to the correct texture class - Artificial Grass. The misclassified novel image is perceptually quite similar to both the correct and the incorrect model images.

Reliably measuring a relative orientation co-occurrence statistic

Given a texton in an image labelling, the objective is to measure the relative angle of occurrence of surrounding textons, that lie within a circular

neighbourhood, with respect to the given texton. Certain difficulties have to be overcome in order to reliably measure this relative angle co-occurrence. Firstly, the angles of occurrence of the textons have to be measured robustly. Conventionally, working in a match filter paradigm, the orientation of a feature (such as an edge or a bar) is determined to be the angle of maximum response of a filter designed to match that feature. However, features can

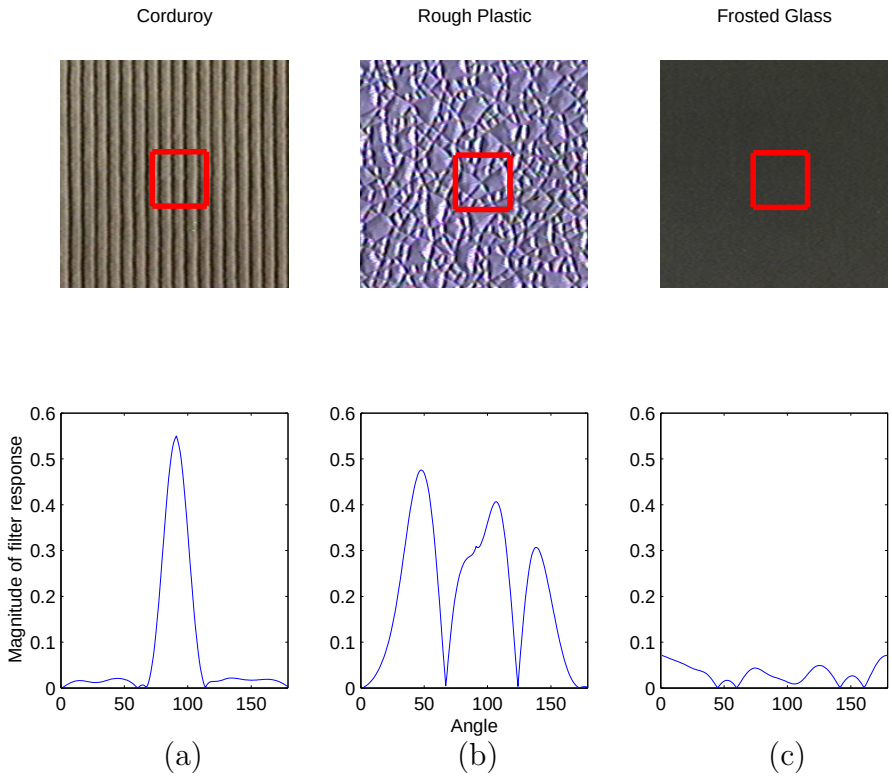


Figure 4.8: Determining the orientation of image features: The top row shows images of 3 textures (a) Corduroy, (b) Rough Plastic and (c) Frosted Glass, with a highlighted central image patch which is matched with an edge filter at all orientations. The magnitude of the filter response versus the orientation is plotted in the bottom row. As can be seen: (a) is a strongly oriented texture having a single direction and therefore its filter response is uni-modal; (b) the texture contains edges along several directions and this is reflected in its filter response; (c) the texture is isotropic and the features have no specific orientation. Plots (b) and (c) show that defining the orientation of a feature to be the angle at which the maximal filter response occurs can be unstable.

occur at multiple angles at the same point and, as such, it is difficult to assign them a particular orientation (see figure 4.8). For instance, an edge filter will have a maximal response at two orientations when matching a corner and choosing one edge orientation over the other will lead to instabilities. Note that these instabilities do not affect the MR representation because only the value of the response (not its angle) is significant – if the same value occurs at two orientations the orientation corresponding to the maximum response is unstable, but the maximum response is not. Here we use the oriented filter (of MR8) that has the maximum response to determine the orientation.

Returning to relative orientation, a robust representation can be obtained if the magnitude of the filter response at each angle (normalised so that the sum of magnitudes squared over all angles is unity) is treated as a confidence measure in the feature occurring at that orientation. Thus, in our case, this *normalised magnitude vector* will be a 6 vector representing the confidence that the given feature occurs at the 6 angles corresponding to the orientations present in the MR8 filter bank (though a richer representation can be obtained using approximated steerable kernels and interpolation [Perona, 1992]). The relative angles between two features, which is invariant to rotation, can now be calculated by computing the cross-correlation between their normalised magnitude vectors. Given a central texton, we can compute the frequency with which other textons occur at various relative angles to it by forming the sum of the cross-correlations between the normalised magnitude vectors of the central texton and the surrounding textons. Essentially, this is computing (via soft binning) the count of how many times a neighbouring texton occurs at a given angle relative to the central texton. To maintain rotational invariance, the surrounding textons come from a circular neighbourhood with a predefined radius, centred around the given texton.

Extending the VZ algorithm

Now that a co-occurrence 6-vector can be associated with every texton in an image labelling, the VZ algorithm can be extended to use the joint distribution of filter responses and co-occurrence vectors. Just as filter responses were clustered into filter response textons in subsection 3.3.3, co-occurrence vectors can be clustered to find exemplars as well, and a dictionary of co-occurrence vector textons can be formed. Textons from this dictionary can be used to label the co-occurrence vectors for a given image. The model for a training image then becomes the joint histogram of the frequency of occurrence of filter response textons and co-occurrence vector textons. Thus, a model is an $K_{fr} \times K_{cv}$ matrix M where K_{fr} is the number of filter response textons and K_{cv} is the number of co-occurrence vector textons. Each entry M_{ij} in this matrix represents the probability of filter response texton K_{fr_i} and orientation co-occurrence texton K_{cv_j} occurring together in the training image. This is somewhat similar to the co-occurrence representation of [Schmid, 2001]. To classify a novel image, its joint histogram is built and is then compared to all the models using χ^2 over all elements of the M matrix. Thus, the essence of the classifier remains the same, the only extension is that joint distribution of filter response and co-occurrence textons are used rather than just the histogram of filter response textons. Hence, we get to add extra information and yet retain all the benefits of the existing classification scheme.

Experimental setup and classification results

The orientation co-occurrence texton dictionary is created by clustering the co-occurrence vectors (calculated for a particular radius of the circular neighbourhood) from the same set of 13 training images per texture that were used

to generate the filter response texton dictionary. The filter responses and co-occurrence vectors of the training images are then labelled using the two texton dictionaries. Finally, the models are built by forming the frequencies, in the $K_{fr} \times K_{cv}$ texton space, of the joint occurrence of the filter response textons and the orientation co-occurrence textons.

Obviously, the choice of K_{fr} and K_{cv} is important as $K_{fr} \times K_{cv}$ equals the total number of textons used and therefore determines how accurately the joint PDF is approximated. However, K_{fr} cannot be chosen to equal 610 as had been done for VZ Benchmark, because the total number of textons becomes too large and we start over-fitting the data (see table 4.6 (a)-(c)). A lower value, such as $K_{fr} = 30$, was found to be more appropriate. Table 4.6 (d)-(f) lists the classification results obtained for various values of the radius when K_{cv} is also set to 30. The performance, using the joint representation,

Radius	610 FR Textons	610 CV Textons	610×610 Joint	30 FR Textons	30 CV Textons	30×30 Joint
01	96.86%	74.51%	88.02%	92.94%	63.93%	95.22%
02	96.75%	68.13%	85.28%	92.62%	60.08%	94.72%
05	96.86%	65.39%	85.88%	92.87%	54.84%	94.15%
10	96.65%	61.26%	85.13%	92.23%	48.68%	93.33%
	(a)	(b)	(c)	(d)	(e)	(f)

Table 4.6: Classification results for all 61 textures using 46 models per texture when orientation co-occurrence information is incorporated into the classification scheme. (a) classification accuracy if only 610 filter response (FR) textons are used to label images and build models. There are minor variations in the classification rate as the number of points available for labelling changes with the radius. (b) classification accuracy if only 610 co-occurrence vector (CV) textons are used. (c) classification rate if the joint distribution is used. The results are poor as there are too many textons and the data is being over fitted. The next three columns have the same format except now both the texton dictionaries have been pruned to 30 textons each. The joint classification rate improves and is better than either of the marginals, though it is still not as good as that obtained by just using 900 FR textons.

is better than using just 30 filter response textons or just 30 co-occurrence vector textons. Though it is worse than if 900 filter response textons were used without any co-occurrence. If the radius is kept fixed and K_{cv} varied then the performance of the joint representation, predictably, first increases, reaches a maximum and then falls (though in no case is it ever able to surpass the performance achieved using an equivalent number of filter response textons alone).

These results indicate, that at least for this dataset, the density of filter response textons is the best measure of discrimination and that orientation co-occurrence does not help much in classification (similar results were found for spatial co-occurrence as well). They also confirm that rotational invariance is advantageous and that no significant information is being lost in this case by using a rotationally invariant filter bank.

4.3 Conclusions

In this chapter, we have studied variations and extensions of the VZ classification algorithm. In particular, two novel methods for reducing the number of models needed to characterise textures were introduced and their superiority over existing algorithms demonstrated. While the MR8 filters tended to do the best in general, the extremely low dimensional MRS4 filter bank also did very well when the number of models was reduced. It always outperformed the 48 dimensional LMS filters and often did better than the 13 dimensional S filter bank. One can therefore hope that MRS4's performance will be comparatively even better when there are large variations in scale and rotation and only a few models are available for classification.

As regards choice of training images and texton dictionary, it was shown

that the VZ classifier is fairly robust. For VZ Benchmark, a classification accuracy of 96.93% was obtained using a dictionary of 610 textons and 46 models per class. This went up to 97.43% for VZ Best using 2440 textons. The number of models was reduced to between 7 and 8 per texture class using the *Greedy* algorithm.

Finally, we concluded that even though the basic VZ classification algorithm can be extended by incorporating second order statistics this does not lead to an improvement in the overall classification. This implies that using only the frequency distribution of textons is sufficient and that no significant information is being lost by employing rotationally invariant filter banks in this case.

Chapter 5

Unifying Classification Frameworks

In this chapter, we examine two of the most successful frameworks in which the problem of texture classification has been attempted – Leung and Malik's texton frequency comparison framework described in chapter 3 and the Bayesian framework as exemplified by the classifier of [Konishi and Yuille, 2000]. While the frameworks appear seemingly unrelated, we draw out the similarities between them, and show that the two can be made equivalent under certain choices of representation and distance measure.

The equivalence is made possible as there is a close correspondence between the two common representations of filter outputs – textons and binned histograms. Furthermore, nearest neighbour matching and Bayesian classification can be shown to give identical results for particular choices of the distance measure. These facts allow a direct comparison to be made between the nearest neighbour texton frequency comparison framework and the Bayesian framework.

5.1 Introduction

Chapter 3 introduced the concept of textons as being the fundamental building blocks of a texture, i.e. a texture could be thought of as being generated by the overlaying of its textons. The textons, in turn, could be determined automatically by clustering the texture's filter responses. This, quite naturally, led to a framework where textures were modelled by keeping a count of how frequently a particular texton occurred in a given texture image. Such models were sufficient for classification because, as was shown in section 3.2, if a texture was found to have many more textons representing holes rather than textons representing horizontal edges, it could safely be concluded that the texture must be sponge rather than ribbed paper. This was validated by the classification results of the VZ algorithm where nearest neighbour matching of texton frequencies using the χ^2 statistic achieved very good performance on the CUReT database. As such, this modelling and classification framework has come to be widely used (although other distance measures could also be employed such as the Bhattacharya metric [Thacker et al., 1997], Earth Mover's distance [Rubner et al., 2000], Mutual Information, KL divergence and Cross Entropy [Kullback, 1968]).

Konishi and Yuille [Konishi and Yuille, 2000] argued in favour of a Bayesian alternative. Filter banks were still used to extract texture features, but now the goal was to learn the class conditional distribution of filter responses (stored as binned histograms) so that Bayes' decision rule could be used to classify each pixel. While Konishi and Yuille did not classify image regions, their algorithm could easily be extended to do so [Schmid, 2001] by making the *naïve* Bayes assumption that a region is a collection of statistically independent pixels, and whose probability is therefore the product of the individual pixel probabilities.

On the surface, it seems that these are two different and competing frameworks. Leung and Malik's methodology appears to be motivated by the concept of a texton and psychophysical research suggesting that their first order statistics are sufficient for pre-attentive texture discrimination [Malik and Perona, 1990]. On the other hand, Konishi and Yuille's methodology appears to be statistically grounded and applies the more general principle that Bayesian classification, whenever possible, is optimal and therefore to be preferred.

However, in this chapter, we show that these two schools of thought are actually very similar. This is illustrated in figure 5.1 which shows that although textons had initially been introduced as physical entities representing texture

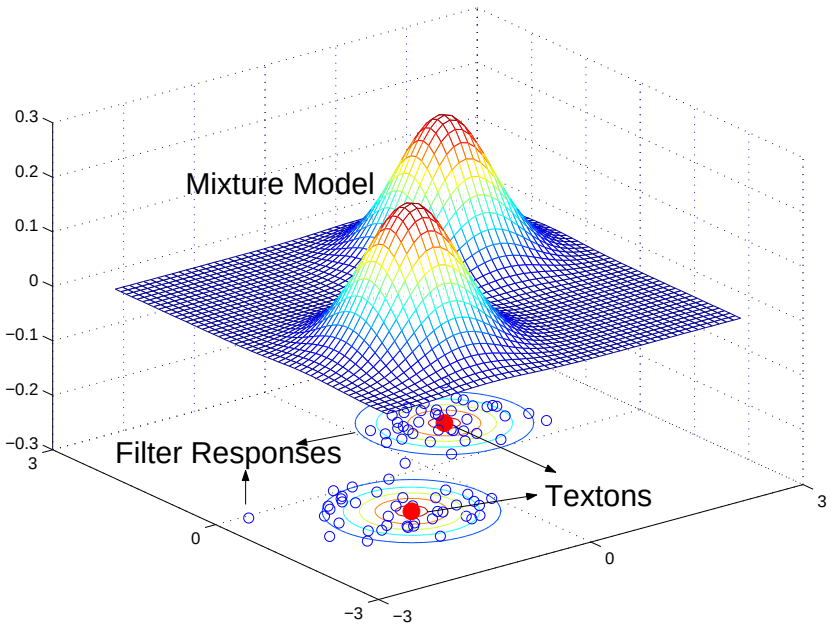


Figure 5.1: Although textons were introduced as exemplar filter responses generated by physical texture primitives, they also have a statistical interpretation wherein they represent the joint PDF of filter responses. For instance, they could be viewed as representing the means in a Gaussian Mixture Model. Alternatively they could form a semi-parametric representation using binned histograms.

primitives, they also have a statistical interpretation wherein they represent the joint PDF of filter responses (as a mixture model or binned histogram for instance). Thus, there is a close correspondence between texton frequencies and binned histograms and section 5.2 shows how one representation can be converted into the other. Experiments on the CUReT dataset, described in section 5.3, confirm that very similar results are achieved by the VZ algorithm using either representation. To complete the equivalence between the two frameworks, section 5.4 shows how nearest neighbour matching using suitable distance measures gives identical results to *naïve* Bayesian classification with uniform priors. Coupled with the texton-bin correspondence, this lets us implement a Bayesian classifier using the texton representation and thereby make direct comparisons between the two frameworks. These results have previously appeared in [Varma and Zisserman, 2004, Varma and Zisserman, 2002c].

5.2 Filter response representation

The texton representation of filter responses was discussed in chapter 3 and details can be found in section 3.3. In this section, we first introduce the binned histogram representation and associated statistical model and then show how to convert one representation to the other.

5.2.1 Histogram representation by binning

In this representation, the model corresponding to a given image is the joint probability distribution of the image's filter responses – obtained by quantizing the responses into bins and normalising so that the sum over all bins is unity. It should be noted that the number of bins and their placement can

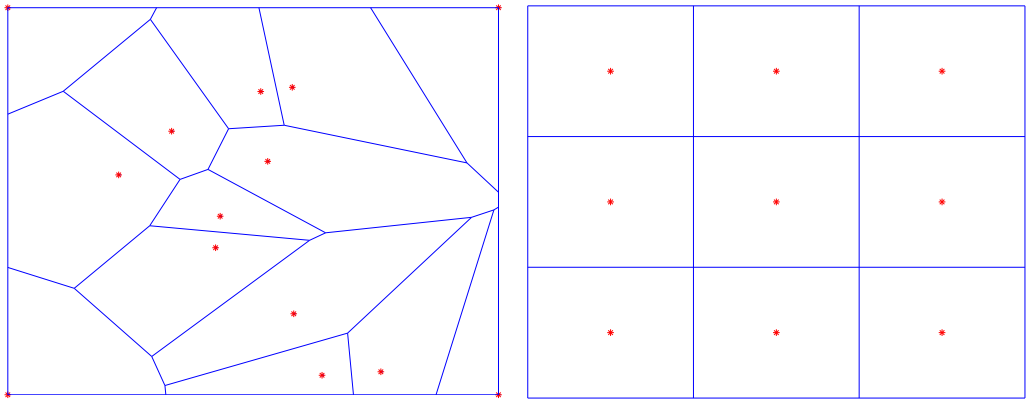
be important parameters as they determine how crudely, or how well, the underlying probability distribution is approximated and whether the data is over-fitted or not. Just as in the VZ algorithm there were multiple models characterising each texture class for the texton representation, there are multiple models for the bin representation as well. This is a necessary departure from the standard Leung & Malik and Konishi & Yuille frameworks, where each texture class has a single model, so as to be able to accurately account for the variation in viewing and illumination conditions.

As an implementation detail, the histogram is stored as a sparse matrix and the space it occupies is given by: number of non-empty bins $\times$ number of bytes required to store a bin value and its corresponding index. This is bounded above by the number of data points and compares favourably to a naïve implementation which stores the full matrix in $\mathcal{O}$(total number of bins) bytes, but where most of the bins are empty. For example, using this implementation for the MR8 filter bank with 20 bins per dimension, we were able to store the PDF of all the training images in less than a hundred megabytes whereas the naïve implementation would have taken over five hundred terabytes. Furthermore, it is efficient to store the histogram as a sparse matrix as the χ^2 statistic can be evaluated in $\mathcal{O}$(number of non-empty bins) flops.

5.2.2 Moving between representations

The two representations of filter responses can be made identical by a suitable choice of bins or textons. For example, an equally spaced bin representation can be converted into an identical texton representation by placing a texton at the centre of every bin (see figure 5.2). It is possible to go the other way round as well. Every texton representation can be converted into an identical

bin representation. In this case, the bins will be irregularly shaped and will correspond to the Voronoi polytopes obtained by forming the Voronoi diagram of the texton sites. Thus, clustering to get textons can be thought of as an adaptive binning method and a histogram of texton frequencies can be equated to a bin count of filter responses. In essence, the comparisons made next in section 5.3 can be thought of as a comparison between two different texton dictionaries.



(a) (b)

Figure 5.2: Texton and bin correspondence in two dimensions: (a) Every texton representation can be converted into an equivalent bin representation where the bins are the Voronoi polygons. (b) Conversely, an equally spaced bin representation can be converted into an identical texton representation by placing a texton at the centre of each bin. A similar equivalence holds in $\mathbb{R}^N$.

However, it should be noted that in general, not every bin representation can be converted to an equivalent texton representation in which there is a bijective mapping between textons and bins. Though it might be possible to find a similar representation if there are more textons than bins with certain textons being grouped together to form a particular bin.

5.3 Classification by distribution comparison

In this section, we investigate the effect of representation on classification carried out via nearest neighbour matching of filter response distributions. Given a set of models characterising the 61 material classes present in the CUReT database, the task is to classify a novel (test) image as one of these textures. This follows the standard procedure developed for the VZ algorithm: the filter response distribution is computed for the test image, and both types of representation (texton and bin) are then determined. In either case, the closest model image, in terms of the χ^2 statistic, is found and the novel image declared to belong to the model's texture class.

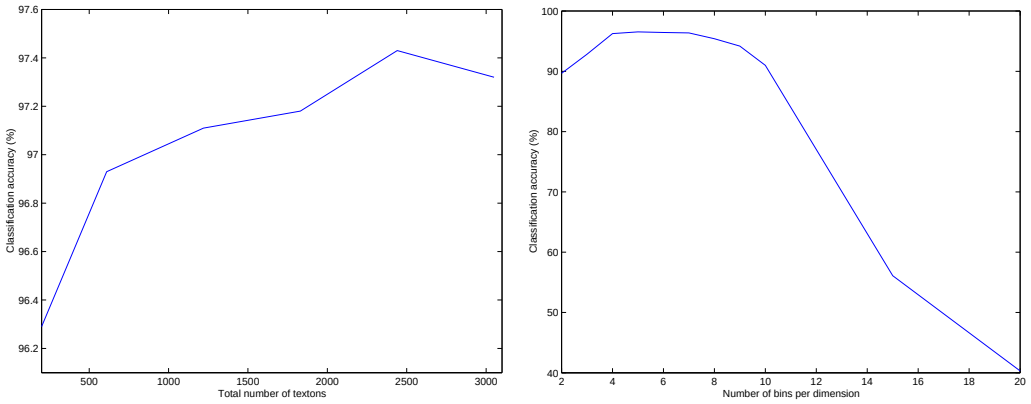
5.3.1 Experimental setup and classification results

The experimental setup is kept unchanged from the one used in subsection 4.2.1 where VZ Benchmark was defined. Thus, classification is carried out on all 61 texture classes for both the representations. The texton dictionary is learnt from all 61 classes as well. In addition, there are 46 models per texture class chosen by selecting every alternate image from the set of 92 available. Classification performance is measured by the proportion of the 2806 test images which are correctly classified as the right texture.

Figure 5.3a plots the classification results obtained using the texton based representation as the size of the dictionary is varied. The best result (VZ Best) was 97.43% when $K = 40$ textons were learnt per texture class resulting in a dictionary of size $S = 61 \times K = 2440$ textons (please refer to subsection 4.2.1 for a discussion of the results).

For the bin representation, the number and location of the bins are, in general, important parameters. However, it turns out that in this case ex-

cellent results are obtained using equally spaced bins. Figure 5.3b plots the classification accuracy for the test set versus the number of bins used in the quantization process. The classifier achieves a maximum accuracy of 96.54% when the filter responses are quantized into 5 bins per dimension. Increasing the number of bins decreases the performance, indicating that the distribution is being over-fitted and that noise is being learnt as well. The classification accuracy also decreases with a decrease in the number of bins as the binning is now coarse.



(a) (b)

Figure 5.3: The variation in classification performance with the number of (a) textons and (b) bins for a nearest neighbour classifier using the χ^2 statistic to match distributions. The best classification results obtained are (a) 97.43% using a dictionary of size $S = 2440$ textons and (b) 96.54% using 5 equally spaced bins per dimension.

Both the representations give very similar classification results. Of course, this is not surprising in light of the fact that the two can be made identical. In this particular instance, however, the texton representation slightly outperforms the bin representation as the bins are always equally sized while the textons are learnt adaptively from the given data.

5.4 Bayesian classification

Given that texton frequencies and histogram binning are equivalent ways of representing the PDF of filter responses, it is now possible to calculate the class conditional probability of obtaining a particular filter response using textons. This setting of a texton representation in a Bayesian paradigm effectively lets us compare, in this section, the Bayesian framework of Konishi and Yuille with the texton based distribution comparison framework developed so far.

The Bayesian classifier of Konishi and Yuille is also divided into a learning stage and a classification stage. In the learning stage, class priors and empirical filter response probabilities are learnt from the training data. Once again, we emphasise that to take into account the variation due to changing viewpoint and illumination a number of models will be used to characterise each texture class, rather than just learning a single model per texture class. In the classification stage, Bayes' theorem is invoked to calculate the posterior probability of a given filter response from a novel image belonging to a particular class.

5.4.1 A Bayesian classifier using the texton representation

The class conditional joint PDF of filter responses is obtained directly from the histogram of texton frequencies for the various images in the training set (of section 5.2). It is straight forward to implement the Bayesian classifier given this information. For a particular model we want to estimate the posterior

$$P(M_{ij}|I) = P(M_{ij}|\{\mathbf{F}(\mathbf{x})\}) \tag{5.1}$$

where $\{\mathbf{F}(\mathbf{x})\}$ is the collection of filter responses generated from the novel image I to be classified, M_{ij} is a particular model corresponding to training image number i taken from texture class number j, and the equality sign arises from the assumption that all the available information in the image has been extracted by the filtering process. The image I is classified as the texture j for which $P(M_{ij}|\{\mathbf{F}(\mathbf{x})\})$ is maximised over all models (i.e. all ij). Using Bayes' rule,

$$P(M_{ij}|\{\mathbf{F}(\mathbf{x})\}) \propto P(\{\mathbf{F}(\mathbf{x})\}|M_{ij})P(M_{ij}) \tag{5.2}$$

where $P(\{\mathbf{F}(\mathbf{x})\}|M_{ij})$ is the likelihood of the model M_{ij}, and $P(M_{ij})$ the prior on model M_{ij}. Since all models are equally likely in our case, the MAP class selection reduces to maximising the likelihood, i.e.

$$\hat{M} = \underset{M_{ij}}{argmax}\, P(\{\mathbf{F}(\mathbf{x})\}|M_{ij}) \tag{5.3}$$

If the filter responses are assumed spatially independent, then the probability of all the filter responses from the novel image belonging to the model M_{ij} is obtained by taking the product of the probabilities of the individual filter responses, i.e.

$$P(\{\mathbf{F}(\mathbf{x})\}|M_{ij}) = \prod_{\mathbf{x}} P(\mathbf{F}(\mathbf{x})|M_{ij}) \tag{5.4}$$

At this point, we take logs and focus on the log-likelihood to clarify the subsequent discussion,

$$\hat{M} = \underset{M_{ij}}{argmax} \prod_{\mathbf{x}} P(\mathbf{F}(\mathbf{x})|M_{ij}) = \underset{M_{ij}}{argmax} \sum_{\mathbf{x}} \log P(\mathbf{F}(\mathbf{x})|M_{ij}) \tag{5.5}$$

$$= \underset{M_{ij}}{argmax} \sum_{\mathbf{x}} \log P(T(\mathbf{x})|M_{ij}) \tag{5.6}$$

$$= \underset{M_{ij}}{argmax} \sum_{k=1}^{S} N_k \log P(T_k|M_{ij}) \tag{5.7}$$

where N_k is the number of times the k^{th} texton occurs in the novel image labelling and $P(T_k|M_{ij})$ is the probability of occurrence of the k^{th} texton in the model M_{ij}. This equality follows because the log likelihood essentially amounts to counting the number of times each filter response falls in a particular bin – but this is exactly what is recorded in the texton frequency histogram.

5.4.2 Equivalence with minimum Cross Entropy and KL divergence

We now show that Bayesian classification in this form can be viewed as nearest neighbour matching of distributions where the distance between two distributions is measured using the Cross Entropy or the KL divergence. Cross Entropy is an information theoretic measure of the average number of bits required to encode symbols from a given alphabet using another alphabet. It is minimised if the same alphabet is used throughout. Based on this observation, Cross Entropy can be used to determine how similar a given distribution is to another distribution. The Cross Entropy between two discrete distributions, p and q, is given by $H(p,q) = -\sum p_k \log q_k$ and the smaller this value the better the match between the two distributions. The KL divergence is a related measure of the similarity between two distributions and is defined as $D(p\|q) = \sum p_k \log (p_k/q_k)$.

Nearest neighbour matching using Cross Entropy or KL divergence can be shown to be equivalent to the Bayesian formulation [Bach and Jordan,

2002, Vasconcelos and Lippman, 2000, Viola, 1995] by noting that from (5.7),

$$\begin{aligned}
\hat{M} &= argmax_{M_{ij}} \quad \sum_{k=1}^{S} N_k \log P(T_k|M_{ij}) \\
&= argmax_{M_{ij}} \quad \sum_{k=1}^{S} \frac{N_k}{\sum_l N_l} \log P(T_k|M_{ij}) \qquad (5.8) \\
&= argmin_{M_{ij}} \quad -\sum_{k=1}^{S} P(T_k|M_I) \log P(T_k|M_{ij}) \qquad (5.9) \\
&= argmin_{M_{ij}} \quad H(p,q)
\end{aligned}$$

where $p_k = P(T_k|M_I) = N_k / \sum_l N_l$ and $q_k = P(T_k|M_{ij})$ give the probabilities of the occurrence of the k^{th} texton in the novel (M_I) and model (M_{ij}) image labellings respectively.

Therefore, a Bayesian classifier which assumes uniform priors and the spatial independence of filter responses will give equivalent results to a nearest neighbour classifier based on the Cross Entropy between the texton distributions of the novel and model images.

The result can be straight forwardly extended to KL divergence by adding the constant $\sum_{k=1}^{S} p_k \log p_k$ to (5.9) and taking it inside the *argmin* operation as it does not depend on M_{ij}. Thus,

$$\begin{aligned}
\hat{M} &= argmin_{M_{ij}} \quad \sum_{k=1}^{S} p_k \log p_k - \sum_{k=1}^{S} p_k \log q_k \qquad (5.10) \\
&= argmin_{M_{ij}} \quad \sum_{k=1}^{S} p_k \log \frac{p_k}{q_k} \qquad (5.11) \\
&= argmin_{M_{ij}} \quad D(p\|q)
\end{aligned}$$

Most of these results are by now standard. The significance for us is that

by combining these equivalences with the texton-bin correspondence shown in section 5.2, it becomes immediately clear that Leung and Malik's texton distribution comparison framework can be made equivalent to the Bayesian framework of Konishi and Yuille.

5.4.3 Relationship with χ^2

The equivalence can be taken further by noting that the capacitory discriminant, a commonly used extension of the KL divergence, is bounded very tightly by the χ^2 statistic and that the bounds are attained when the two probability distributions being compared are similar. In fact, Topsoe [Topsoe, 2000] has shown that

$$\tfrac{1}{2}\chi^2(p,q) \leq D(p\|q) - 2D(\tfrac{1}{2}(p+q)\|q) \leq \ln 2 \cdot \chi^2(p,q) \tag{5.12}$$

where the quantity $D(p\|q) - 2D(\frac{1}{2}(p+q)\|q)$ is denoted by $C(p,q)$ and known as the capacitory discriminant (it is also referred to as Jeffreys' divergence sometimes [Rubner et al., 2000] even though C differs slightly from the original definition [Kullback, 1968]). The capacitory discriminant is often preferred over the KL divergence as it is symmetric, more robust and $\sqrt{C(p,q)}$ is a metric [Endres and Schindelin, 2003]. Furthermore, a Taylor series expansion of C gives

$$\lim_{p \to q} C(p,q) = \frac{1}{2}\chi^2(p,q) \tag{5.13}$$

and a similar relation holds for D and the asymmetrical form of χ^2 (both relations are derived in Appendix A).

5.4.4 Bayesian classification experiments

The equivalence results from the previous subsections allow us to cast a Bayesian classifier as a nearest neighbour classifier with KL divergence as the distance measure. Thus, the experimental setup remains exactly the same as in section 5.3 except now the distance measure being used is KL divergence rather than the χ^2 statistic. Figure 5.4 plots the classification accuracy versus the size of the texton dictionary for the Bayesian classifier. The best results are 97.46% for a dictionary of size $S = 1830$ textons (i.e. $K = 30$ textons learnt from each texture class). This is slightly better than the 97.18% achieved by the VZ algorithm using the same dictionary of 1830 textons as well as the 97.43% achieved by VZ Best using 2440 textons.

A technical point about implementing probability products is that if the

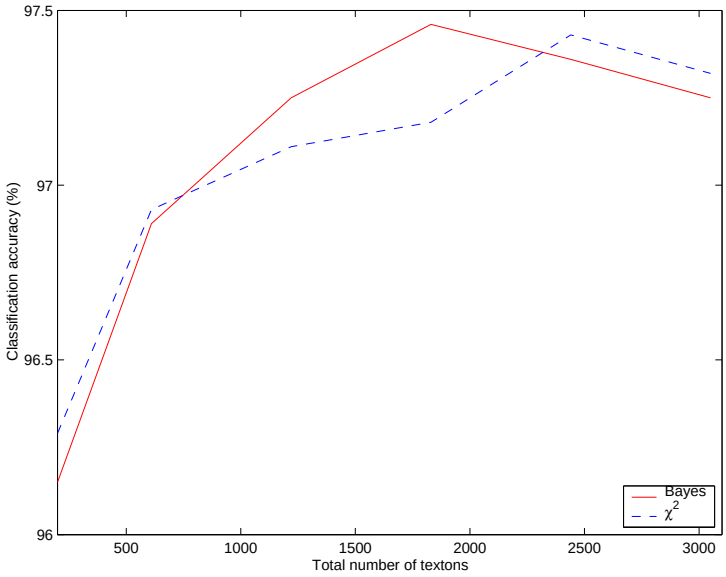


Figure 5.4: The variation in classification performance of a Bayesian classifier with the size of the texton dictionary. In each case, there are 2806 models and 2806 test images. The best classification result obtained is 97.46% using a dictionary of 1830 textons. The results of the texton based χ^2 classifier (i.e. figure 5.3a) are also plotted for the sake of comparison.

model histograms are determined directly from the textons frequencies of the training images then the classification accuracy of the Bayesian classifier is an astonishingly low 1.06%, i.e. almost all the test images are classified incorrectly. This is because most novel images contain a certain percentage of pixels (filter responses) which do not occur in the correct class models in the training set. This may be a result of an inadequate amount of training data, or due to outliers or noise. As a consequence, the posterior probability of these pixels is zero and hence when all the pixel probabilities are multiplied together the image posterior probability also turns out to be zero.

This is a standard pitfall in histogram based density estimation and three solutions are generally proposed: (a) smoothing the histogram, (b) assigning small nonzero values to each of the empty bins, and (c) discarding a certain percentage of the least occurring filter responses in the belief that they are primarily noise and outliers. A combination of (b) and (c) is used here: instead of starting the bin occupancy count from 0, it is started from 1 to ensure that no bin is ever empty. Some of the least frequently occurring bins are also discarded. These modifications lead to the classification performance plotted in figure 5.4.

5.4.5 Comparisons

On the basis of experimental results, there is very little to choose between the Bayesian and distribution comparison classifiers using the texton representation. This is to be expected as, in essence, the test being performed is a comparison of χ^2 and KL divergence as distance measures. Nevertheless, while both classifiers attain rates of over 97%, there are different theoretical pros and cons associated with the two approaches.

There can be no doubt that theoretically, when the underlying distribu-

tions are known perfectly, the Bayesian classifier minimises the classification error. However, when we don't have enough data to accurately determine the true distribution or only have noisy approximations which suffer from the inherent quantization effects of either clustering or histogram binning then the superiority of the Bayesian classifier is much less clear. This is evident from the capability of χ^2 to practically cope with empty bins (even though it is theoretically incapable of doing so) and noisy measurements while the Bayesian classifier completely collapses unless the probability distribution is modified. Furthermore, as can be seen from figure 5.4, the Bayesian classifier is often marginally surpassed by the nearest neighbour χ^2 classifier.

There is also the question about the *naïve* Bayesian assumption that the observed data is independent. However, this can not be considered a major drawback of the Bayesian classifier as compared to χ^2 because (a) χ^2 also makes the very same assumption in its derivation, (b) the experimental results indicate that extremely good classification results are obtained even when the assumption is violated (Schmid [Schmid, 2001] notes that this holds true even for other texture datasets) and (c) if violating the assumption was leading to large errors then this could be tackled by randomly sampling filter responses from disjoint regions of the novel image in a bid to decrease their dependence.

Yet, despite their theoretical limitations, both classifiers appear to work extremely well in practice as is evidenced by the classification results.

5.5 Conclusions

In conclusion, we have shown that the texton representation of the PDF of filter responses is equivalent to an adaptive bin representation and, con-

versely, that every regularly partitioned bin representation can be converted into an equivalent texton representation. This has enabled the use of texton densities for texture classification in the Bayesian framework which itself, under certain circumstances, can be viewed as another measure of distance in a distribution comparison classification scheme. Doing so has brought together two seemingly unrelated schools of thought in texture classification – one based on the Bayesian paradigm and the other on textons and their first order statistics.

Chapter 6

Are Filter Banks Necessary?

In this chapter, we take a fresh look at the problem of texture classification and question the dominant role that filter banks have played so far. An alternative image patch texture representation is developed based on the joint distribution of pixel intensities in a neighbourhood. Using this new representation in the VZ algorithm leads to two startling results, namely that (a) very good classification performance can be achieved using extremely compact neighbourhoods (starting from as small as 3×3 pixels square) and that (b) for any fixed size of the neighbourhood, image patches lead to superior classification as compared to filter banks with the same support. We discuss theoretical reasons as to why this might be the case.

6.1 Introduction

Texture research is generally divided into five canonical problem areas: (1) synthesis; (2) classification; (3) segmentation; (4) compression; and (5) shape from texture. The first four areas have come to be heavily influenced by the use of filter banks and wavelets. This is particularly true of synthesis and

classification where some of the best results have been achieved by filter bank based methods.

However, even though there has been ample empirical evidence to suggest that filter banks and wavelets can lead to good performance, scant theoretical justification has been provided as to their optimality or, even for that matter, their necessity for texture classification or synthesis. In fact, the supremacy of filter banks for texture synthesis was brought into question by the approach of Efros and Leung [Efros and Leung, 1999]. They demonstrated that superior synthesis results could be obtained using local pixel neighbourhoods directly, without resorting to large scale filter banks. In a related development, Zalesny and Van Gool [Zalesny and Van Gool, 2000] also eschewed filter banks in favour of a Markov random field (MRF) model.

Both these works put MRFs firmly back on the map as far as texture synthesis was concerned. Efros and Leung gave a computational method for generating a texture with similar MRF statistics to the original sample, but without explicitly learning or even representing these distributions. Zalesny and Van Gool, using a subset of all available cliques present in a neighbourhood, showed that it was possible to learn and sample from a parametric MRF model given enough computational power.

In this chapter, it is demonstrated that the second of the canonical problems, texture classification, can also be tackled effectively by employing only local neighbourhood distributions, and without the use of large filter banks. A previous version of this work has appeared in [Varma and Zisserman, 2003].

6.2 The image patch based classifiers

The use of filter banks in texture classification algorithms has largely been motivated by arguments from feature extraction and dimensionality reduction. In keeping with these philosophies, filter banks have traditionally tended to include filters with large support occurring at multiple scales and orientations.

In this section, we investigate the effect of replacing filter responses with the source image patches from which they were derived. The rationale for doing so comes from (3.1) which shows that, essentially, a filter response is a lower dimensional projection of an image patch onto a linear subspace spanned by the vector representation of the individual filters (obtained by row reordering each filter mask). This immediately provides a way of validating the hypotheses put forward in favour of filtering – for, if the hypotheses are valid, then lower dimensional filter responses must lead to superior performance as compared to their higher dimensional source patches and, furthermore, small patches which cannot capture the low frequency signal component should not do well.

In order to test these hypotheses, the VZ algorithm of chapter 3 is modified so that filter responses are replaced by their source image patches. Thus, the new classifier is identical to the VZ algorithm except that, at the filtering stage, instead of using a filter bank to generate filter responses at a point, the raw pixel intensities of an $N \times N$ square neighbourhood around that point are taken and row reordered to form a vector in an N^2 dimensional feature space. All pre and post processing steps are retained and no other changes are made to the classifier. Hence, in the first stage of learning, all the image patches from the selected training images in a texture class are aggregated and clustered. The cluster centres from the various classes are

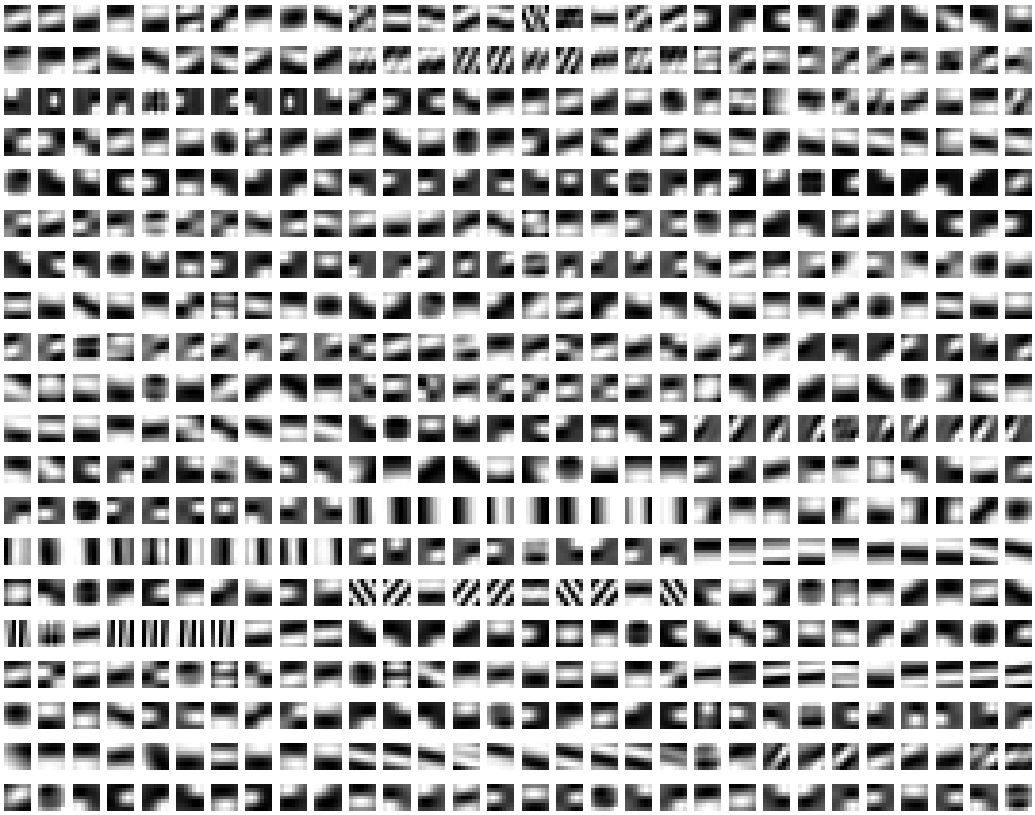


Figure 6.1: Image patch textons learnt from the CUReT database using neighbourhoods of size 7×7.

grouped together to form the texton dictionary. The textons now represent exemplar image patches rather than exemplar filter responses (see figure 6.1). However, the model corresponding to a training image continues to be the histogram of texton frequencies and novel image classification is still achieved by nearest neighbour matching using the χ^2 statistic. This classifier will be referred to as the Joint classifier. Figure 6.2 highlights the main difference in approach between the Joint classifier and the VZ classifier using the MR8 filter bank.

We also design two variants of the Joint classifier – the Neighbourhood classifier and the MRF classifier. Both of these are motivated by the recog-

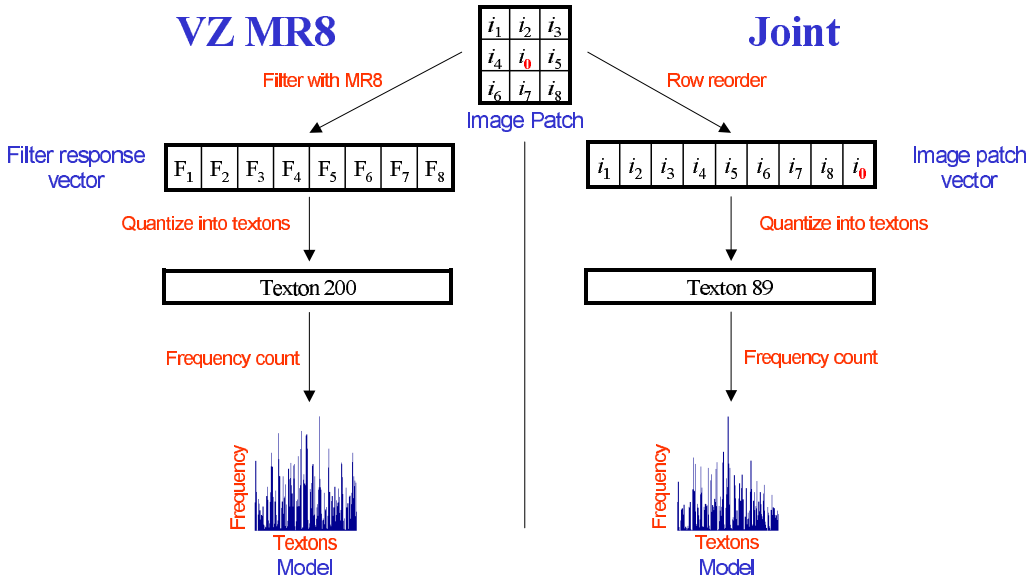


Figure 6.2: The only difference between the Joint and the VZ MR8 representations is that the source image patches are used directly in the Joint representation as opposed to the derived filter responses in VZ MR8.

nition that textures can often be considered realizations of a Markov random field. In an MRF framework [Geman and Geman, 1984, Li, 2001], the probability of the central pixel depends only on its neighbourhood. Formally,

$$p(I(\mathbf{x}_c)|I(\mathbf{x}), \forall \mathbf{x} \neq \mathbf{x}_c) = p(I(\mathbf{x}_c)|I(\mathbf{x}), \forall \mathbf{x} \in \mathcal{N}(\mathbf{x}_c)) \tag{6.1}$$

where $\mathbf{x}_c$ is a site in the 2D integer lattice on which the image I has been defined and $\mathcal{N}(\mathbf{x}_c)$ is the neighbourhood of that site. In our case, $\mathcal{N}$ is defined to be the $N \times N$ square neighbourhood (excluding the central pixel). Thus, although the value of the central pixel is significant, its distribution is conditioned on its neighbours alone. The Neighbourhood and MRF classifiers are designed to test how significant this conditional probability distribution is for classification.

For the Neighbourhood classifier, the central pixel is discarded and only

the neighbourhood is used for classification. Thus, the Neighbourhood classifier is essentially the Joint classifier retrained on feature vectors drawn only from the set of $\mathcal{N}$: i.e. the set of $N \times N$ image patches with the central pixel left out. For example, in the case of a 3×3 image patch, only the 8 neighbours of every central pixel are used to form feature vectors and textons.

We next go to the other extreme and, instead of ignoring the central pixel, explicitly model $p(I(\mathbf{x}_c) \wedge I(\mathcal{N}(\mathbf{x}_c)))$, i.e. the joint distribution of the central pixels and its neighbours. Up to now, textons have been used to implicitly represent this joint PDF. The representation is implicit because, once the texton frequency histogram has been formed, neither the probability of the central pixel nor the probability of the neighbourhood can be recovered straightforwardly by summing (marginalizing) over the appropriate textons. Thus, the texton representation is modified slightly so as to make explicit the central pixels PDF within the joint and to represent it at a finer resolution than its neighbours (just as in the Neighbourhood classifier, the central pixel PDF was discarded by representing it at a much coarser resolution using a single bin).

To learn the PDF representing the MRF model for a given training image, the neighbours' PDF is first represented by textons as was done for the Neighbourhood classifier – i.e. all pixels but the central are used to form feature vectors in an $N^2 - 1$ dimensional space which are then labelled using the same dictionary of 610 textons. Then, for each of the S_N textons in turn ($S_N = 610$ is the size of the neighbourhood texton dictionary), a one dimensional distribution of the central pixels' intensity is learnt and represented by an S_C bin histogram. Thus the representation of the joint PDF is now an $S_N \times S_C$ matrix. Each row is the PDF of the central pixel for a given neighbourhood intensity configuration as represented by a specific texton.

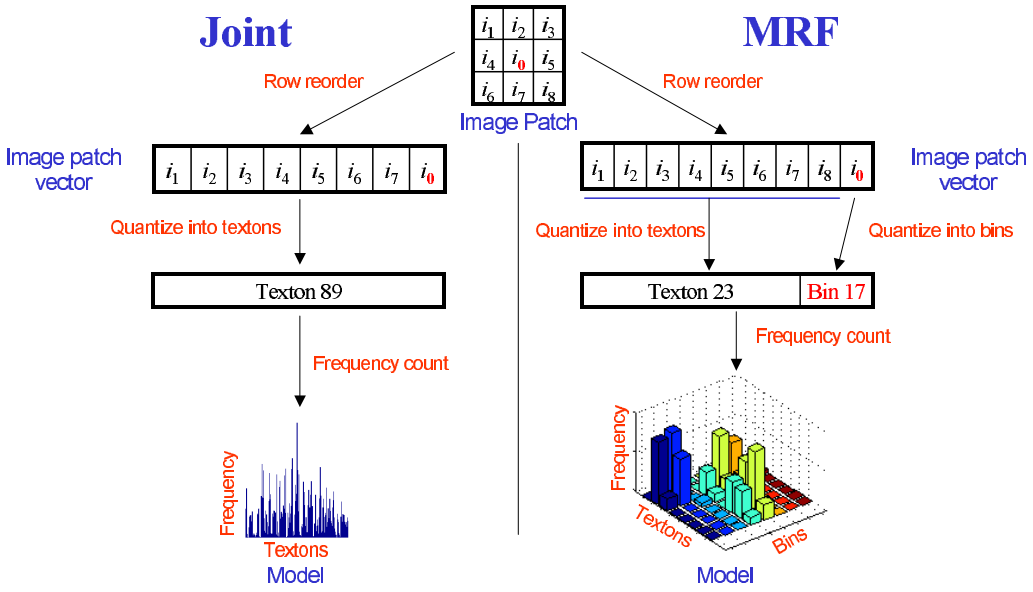


Figure 6.3: MRF texture models as compared to those learnt using the Joint representation. The only point of difference is that the central pixel PDF is made explicit and stored at a higher resolution. The Neighbourhood representation can be obtained from the MRF representation by marginalizing out the central pixel.

Figure 6.3 highlights the differences between MRF models and models learnt using the Joint representation. Using this matrix, a novel image is classified by comparing its MRF distribution to the model MRF distributions (learnt from training images) by computing the χ^2 statistic over all elements of the $S_N \times S_C$ matrix. This will be referred to as the MRF classifier.

Table 6.1 presents a comparison of the performance of the Joint, Neighbourhood and MRF classifiers when classifying all 61 textures in the CUReT database. Image patches of size $3 \times 3, 5 \times 5$ and 7×7 are tried while using a dictionary of 610 textons. For the Joint classifier, it is remarkable to note that classification results of over 95% are achieved using patches as small as 3×3. In fact, the classification result for the 3×3 neighbourhood is actually better than the results obtained by using the MR4 (91.70%), MRS4

N	Joint Classifier	Neighbourhood Classifier	MRF with 90 bins
3	95.33% (9.6)	94.90% (9.3)	95.87% (9.4)
5	95.62% (8.4)	95.97% (8.6)	97.22% (8.1)
7	96.19% (8.4)	96.08% (8.2)	97.47% (7.9)
	(a)	(b)	(c)

Table 6.1: Comparison of classification results of all 61 textures in the CUReT database for different $N \times N$ neighbourhood (patch) sizes: (a) all the pixels in an image patch are used to form vectors in an N^2 feature space; (b) all but the central pixel are used (i.e. an $N^2 - 1$ space); (c) the MRF classifier where 90 bins are used to represent the joint neighbourhood and central pixel PDF. The bracketed values report the number of models per texture class as determined by the *Greedy* algorithm. A dictionary of 610 textons learnt from all 61 textures is used throughout. Notice that the performance using these small patches is as good as that achieved by the multi orientation, multi scale, large support MR8 filter bank (VZ Benchmark gets 96.93% using 610 textons while VZ Best achieves 97.43% using 2440 textons).

(94.23%), LMS (94.65%) or S (95.22%) filter banks with 610 textons learnt from all 61 classes. This is strong evidence that there is sufficient information in the joint distribution of the nine intensity values (the central pixel and its eight neighbours) to discriminate between the texture classes. For the Neighbourhood classifier, as shown in column (b), there is almost no significant variation in classification performance as compared to using all the pixels in an image patch. Classification rates for $N = 5$ are slightly better when the central pixel is left out and marginally poorer for the cases of $N = 3$ and $N = 7$. Thus, the joint distribution of the neighbours is largely sufficient for classification. Column (c) presents a comparison of the performance of the Joint and Neighbourhood classifiers to the MRF classifier when a resolution of 90 bins is used to store the central pixels' PDF. As can be seen, the MRF classifier does better than both the Joint and Neighbourhood classifiers. What is also very interesting is the fact that using 7×7 patches, the

performance of the MRF classifier (97.47%) is at least as good as the best performance achieved by the multi-orientation, multi-scale MR8 filter bank with support 49×49 (97.43% using 2440 textons).

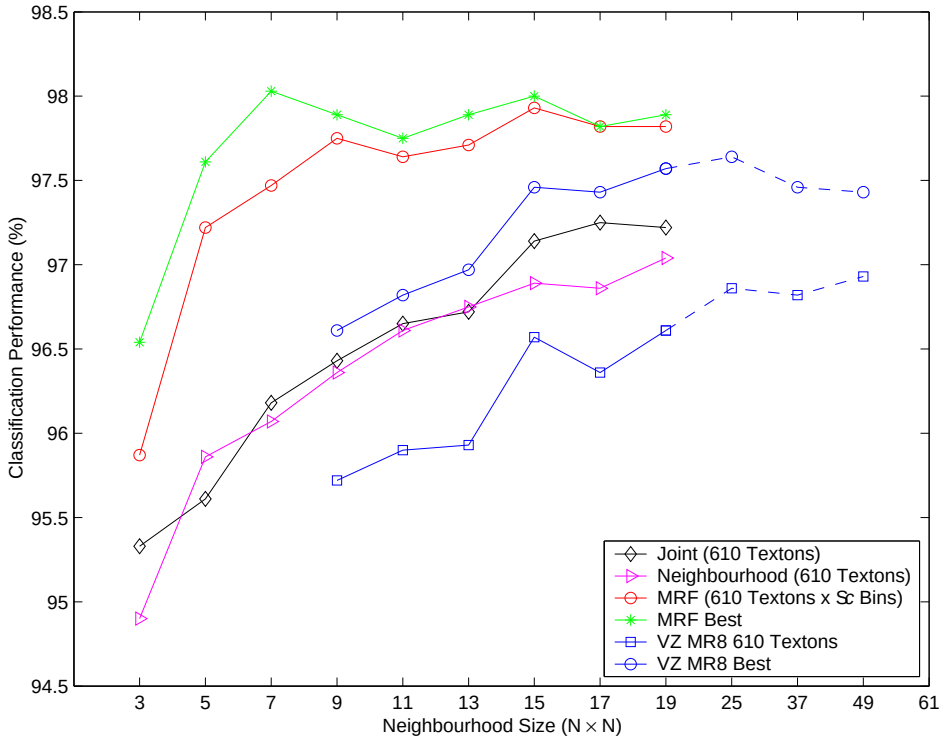


Figure 6.4: Classification results as a function of neighbourhood size. The VZ MR8 Best curve shows the best results obtained by varying the size of the texton dictionary up to 3050 textons. Similarly, the MRF Best curve shows results obtained for the best combination of texton dictionary and number of bins as the neighbourhood size is varied. For neighbourhoods up to 11×11, dictionaries of up to 3050 textons and up to 200 bins are tried. For 13×13 and larger neighbourhoods, the maximum size of the texton dictionary is restricted to 1220 because of computational expense. The best result achieved by the MRF classifiers is 98.03% using a 7×7 neighbourhood with 2440 textons and 90 bins. The best result for MR8 is 97.64% for a 25×25 neighbourhood and 2440 textons. The performance of the VZ algorithm using the MR8 filter bank (VZ MR8) is always worse than any other comparable classifier at the same neighbourhood size. VZ MR8 Best is inferior to the MRF curves, while VZ MR8 with 610 textons is inferior to the Joint and Neighbourhood classifiers also with 610 textons.

This result showing that image patches can outperform filters raises the important question of whether filter banks are providing beneficial information for classification, for example perhaps by increasing the signal to noise ratio, or by extracting useful features. To answer this question, the performance of the VZ classifier using the MR8 filter bank (VZ MR8) is compared to that of the Joint, Neighbourhood and MRF classifiers as the size of the neighbourhood is varied. In each experiment, the MR8 filter bank is scaled down so that the support of the largest filters is the same as the neighbourhood size. Figure 6.4 plots the classification results. It is apparent that for any given size of the neighbourhood, the performance of VZ MR8 is worse than that of the Joint or even the Neighbourhood classifiers. This would suggest that using all the information present in an image patch is more beneficial for classification than relying on lower dimensional responses of a pre-selected filter bank. A classifier which is able to learn from all the pixel values is superior.

However, before proceeding further, it should be established that the comparisons between the image patch based Joint, Neighbourhood and MRF classifiers on the one hand and VZ MR8 on the other, are indeed fair and that comparisons are being made between classifiers of equal complexity. Unfortunately, there does not exist a single definitive measure of the complexity of a classification algorithm. We therefore focus on two measures of information and classification complexity but do not measure the time complexity (though it can be critical for certain applications).

Given a fixed set of input images for training and a disjoint test set on which to measure performance, one measure of complexity is the number of internal parameters of the classifier. In our case, this is the size S of the texton dictionary which controls the dimensionality of the space in which the

classifier operates (except for the MRF classifier where the dimensionality is $S_N \times S_C$). It is also a measure of the complexity with which the joint PDF of image patches or filter responses has been approximated. The second parameter is the neighbourhood dimension N (equivalently filter bank support size). In a sense, this is a measure of how much input information is visible to each of the classifiers and is also an indicator of the number of parameters of the Markov random field. For instance, a classifier which can only see 3×3 patches has access to much less information about the true MRF distribution than a 49×49 classifier. Analogously, a filter bank with support 3×3 can "see" far less interesting features than a filter bank with support 49×49 which can model large scale interactions.

A third parameter could be the dimensionality of the feature space. This is the length of the filter response vector in the case of filter banks (for example, 48 for LM and 4 for MR4S) while for patches it equals N^2, i.e. the size of the patch. However, even though this parameter is important for measuring the time complexity of the algorithms it is not very meaningful as a measure of classification complexity unless the input (dimensionality) is fixed (even in which case it's a parameter which we've chosen to ignore so far – for instance, when comparing the 4 dimensional MRS4 filter bank to the 48 dimensional LM filters).

Given these two parameters, S and N, measuring the texton dictionary and neighbourhood sizes respectively, there are four possibilities as to how the comparisons between image patch and filter bank based classifiers can be performed. These are shown in table 6.2 and correspond to whether a parameter is held fixed or allowed to vary during the comparison.

For constant S and N, the size of the texton dictionary is fixed to $S = 610$ while the MR8 filter bank is scaled down to have support $N \times N$. In this

	Fixed N			**Variable N**		
Fixed S (610)	Joint	$>$	VZ MR8	Joint 15×15	$>$	VZ MR8 49×49
Variable S	MRF Best	$>$	VZ MR8 Best	MRF Best 7×7	$>$	VZ MR8 Best 25×25

Table 6.2: Different ways in which the image patch based Joint, Neighbourhood and MRF classifiers can be compared to the VZ algorithm using the MR8 filter bank. In each case, image patches lead to superior classification as compared to filter banks.

case, figure 6.4 shows that the Joint classifier always does better than VZ MR8 for each choice of N. Note that the MRF classifier can not be brought into this comparison as its effective dictionary size is always greater than 610. For fixed S but variable N, the texton dictionary size is held constant at 610 textons but classifiers are compared for the best neighbourhood size. For VZ MR8 with 610 textons the best results are 96.93% using 49×49 support (VZ Benchmark) but again, figure 6.4 shows that this is inferior to the performance of the Joint classifier for any neighbourhood size greater than 15×15. The results with N held fixed but variable S follow the same pattern with MRF Best always being superior to VZ MR8 Best for every neighbourhood size. Finally, allowing both S and N to vary compares the best performance of the classifiers irrespective of neighbourhood size and texton dictionary. Again, image patches are superior to filter banks in this comparison as the best overall image patch result is 98.03% for the MRF Classifier while for MR8 it is 97.64%.

These results have demonstrated that a classification scheme based on MRF local neighbourhood distributions can achieve very high classification rates and can outperform methods which adopt large scale filter banks to extract features and reduce dimensionality. Before turning to discuss theoretical reasons as to why this might be the case, we first explore how issues

such as rotation and scale impact the image patch classifiers.

6.3 Scale, rotation, synthesis & other datasets

Three main criticisms can be levelled at the classifiers developed in the previous section. Firstly, it could be argued that the lack of significant scale change in the CUReT textures might be the reason why image patch based classification outperforms the multi scale MR8 filter bank. Secondly, the image patch representation has a major disadvantage in that it is not rotationally invariant. And thirdly, the reason why small image patches do so well could be because of some quirk of the CUReT dataset and that classification using small patches will not generalise to other databases. In this section, each of these three issues is addressed experimentally and it is shown that the image patch representation is as robust to scale changes as MR8, can be made rotationally invariant and generalises well to other datasets. We also briefly illustrate how the representation can be used to synthesise textures.

6.3.1 The effect of scale changes

To test the hypothesis that the image patch representation will not do as well as the filter bank representation in the presence of scale changes, four texture classes were selected from the CUReT database (material numbers 2, 11, 12 and 14) for which additional scaled data is available (as material numbers 29, 30, 31 and 32). Two experiments were performed. In the first, models were learnt only from the training images of the original textures while the test images of both the original and scaled textures were classified. In the second experiment, both test sets were classified once more but this time models were

learnt from the original as well as the scaled textures. Table 6.3 shows the results of the experiments. It also tabulates the results when the experiments are repeated but this time with the images being scaled synthetically by a factor of two.

	Naturally Scaled		Synthetically Scaled ×2	
	Original	Original + Scaled	Original	Original + Scaled
MRF	93.48%	100%	65.22%	99.73%
MR8	81.25%	99.46%	62.77%	99.73%

Table 6.3: Comparison of classification results of the MRF and VZ MR8 classifiers for scaled data. Models are learnt either from the original textures only or the original + scaled textures while classifying both texture types. In each case, the performance of the MRF classifier is at least as good as that using the multi scale MR8 filter bank.

In the naturally scaled case, when classifying both texture types using models learnt only from the original textures, the MRF classifier achieves 93.48% while VZ MR8.gets only 81.25%. This shows that the MRF classifier is not being adversely affected by the scale variations. When images from the scaled textures are included in the training set as well, the accuracy rates go up to 100% and 99.46% respectively. A similar trend is seen in the case when the scaled textures are generated synthetically. Both these results show that image patches cope as well with scale changes as the MR8 filter bank, and that features do not have to be extracted across a large range of scales for successful classification.

6.3.2 Incorporating rotational invariance

The fact that the image patch representation developed so far is not rotationally invariant can be a serious limitation. However, it is straight forward to incorporate invariance into the representation and this is done as follows:

instead of using an $N \times N$ square patch, the neighbourhood is redefined to be circular with a given radius. Then, before forming feature vectors, each circular neighbourhood is reduced to a canonical frame by determining its local orientation and rotating the neighbourhood by the determined angle. This achieves rotational invariance. Table 6.4 lists the results for the Neighbourhood and MRF classifiers when classifying all 61 textures using circular neighbourhoods with radius 3 pixels (corresponding to a 7×7 patch) and 4 pixels (9×9 patch).

	Neighbourhood Classifier		MRF Classifier	
	Rot. Invariant	Not Invariant	Rot. Invariant	Not Invariant
7×7	96.36%	96.08%	97.07%	97.47%
9×9	96.47%	96.36%	97.25%	97.75%

Table 6.4: Comparison of classification results of the Neighbourhood and MRF classifiers using the standard and the rotationally invariant patches.

Using the rotationally invariant representation, the Neighbourhood classifier with a dictionary of 610 textons achieves 96.36% for a radius of 3 pixels and 96.47% for a radius of 4 pixels. This is slightly better than what the same classifier achieves using the standard (not invariant) representation with corresponding 7×7 and 9×9 patches. The rates for the rotationally invariant MRF classifier are 97.07% and 97.25% using 610 textons and 45 bins. These results are slightly worse than those obtained using the standard representation. However, the fact that such high classification percentages were obtained strongly indicates that rotation invariance can be successfully incorporated into the image patch representation.

6.3.3 Synthesis

Before testing the image patch classifiers on other datasets, we briefly demonstrate that our MRF representation may also be used for texture synthesis.

The algorithm is very similar to [Efros and Leung, 1999, Efros and Freeman, 2001]. First, the MRF statistics of the input texture block are learnt using the matrix representation of the PDF of image patches. The parameters that can be varied are N, the size of the neighbourhood, and K the number of textons used to represent the neighbourhood distribution. The central pixel PDF is stored in 256 bins in this case. Next, to synthesise the texture, the input block is initially tiled to the required dimensions. A new image is synthesised from this tiled image by taking every pixel, determining its neighbourhood (i.e. closest texton) and setting the value of the pixel to a value sampled from the learnt MRF distribution. This iteration is repeated until a desired synthesis is obtained. Results are shown in figure 6.5.

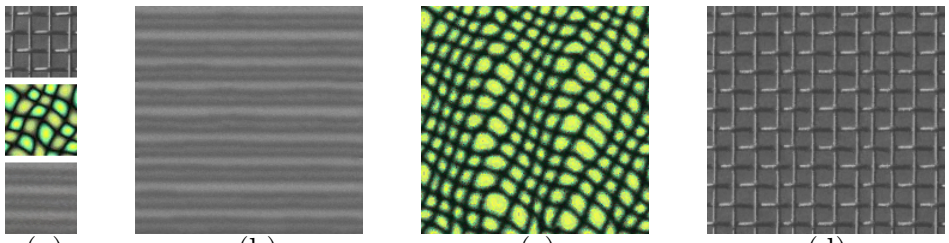


Figure 6.5: Synthesis Results: (a) Input texture blocks, (b) Ribbed Paper (CUReT) synthesised using a 7×7 neighbourhood and 100 textons (c) Efros and Leung [Efros and Leung, 1999] - 15×15, 800 textons and (d) D6 (Brodatz) - 11×11, 300 textons.

6.3.4 Results on other datasets

We now show that small image patches can also be used to successfully classify textures other than those present in the CUReT database. It is demonstrated that using the Joint classifier with patches of size $3 \times 3, 5 \times 5$ and 7×7 is sufficient for classifying the Microsoft Textile and San Francisco databases. Both databases are described in section 2.3 of the literature re-

view. While the MRF classifier leads to the best results in general, we show that on these databases the Joint classifier already achieves very high performances (99.21% on the Microsoft Textile database and 97.9% on the San Francisco database using only a single training image).

For the Microsoft Textile database, the experimental setup is kept identical to the one used by [Savarese and Criminsi, 2004]. Fifteen images were selected from each of the sixteen texture classes to form the training set. While all the training images were used to form models, textons were learnt from only 3 images per texture class. Various sizes of the texton dictionary $S = 16 \times K$ were tried when $K = 10, \ldots, 40$ textons were learnt per textile. The test set comprised a total of 80 images. Table 6.5 shows the variation in performance of the Joint classifier with neighbourhood size N and texton dictionary size S.

	Size of Texton Dictionary S			
$N \times N$	160	320	480	640
3×3	96.82%	96.82%	96.82%	96.82%
5×5	99.21%	99.21%	99.21%	99.21%
7×7	96.03%	97.62%	96.82%	97.62%

Table 6.5: The Joint classifier performs excellently on the Microsoft Textile database – only a single image is misclassified using 5×5 patches. These results reinforce the fact that very small patches can be used to classify textures with global structure far larger than the neighbourhoods used (the image resolutions are 1024×768).

As can be seen, excellent results are obtained using very small neighbourhoods. In fact, only a single image is misclassified using 5×5 patches (see figure 6.6). These results should reinforce the fact that very small patches can indeed be used to classify textures with global structure far larger than the neighbourhoods used (the image resolutions are 1024×768).

The results are just as good for the San Francisco database. The database

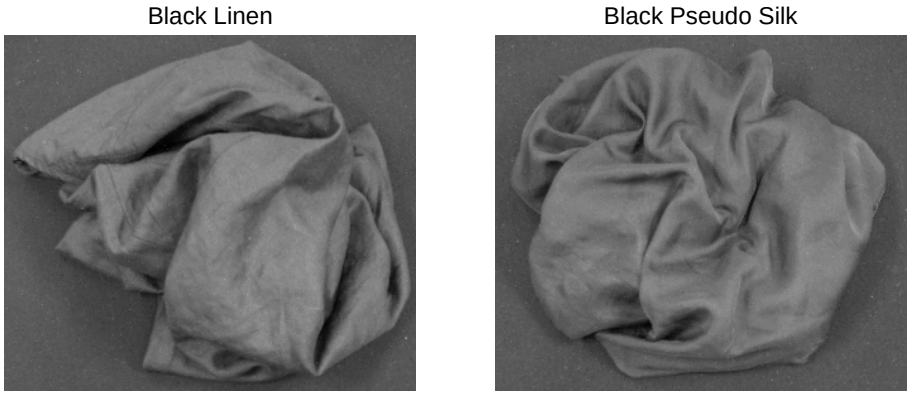


Figure 6.6: Only a single image in the Microsoft Textile database is misclassified by the Joint classifier using 5×5 patches: (a) is an example of Black Linen but is incorrectly classified as Black Pseudo Silk (b).

has 37 images of outdoor scenes taken on the streets of San Francisco. The images have been segmented by hand [Konishi and Yuille, 2000] into 6 classes: Air, Building, Car, Road, Vegetation and Trunk. From the database, a single image is selected for training the Joint classifier (figure 6.7 shows the selected training image and its associated hand segmented regions). All the rest of the 36 images are kept as the test set. Performance is measured by the proportion of pixels that are labelled correctly during classification of the

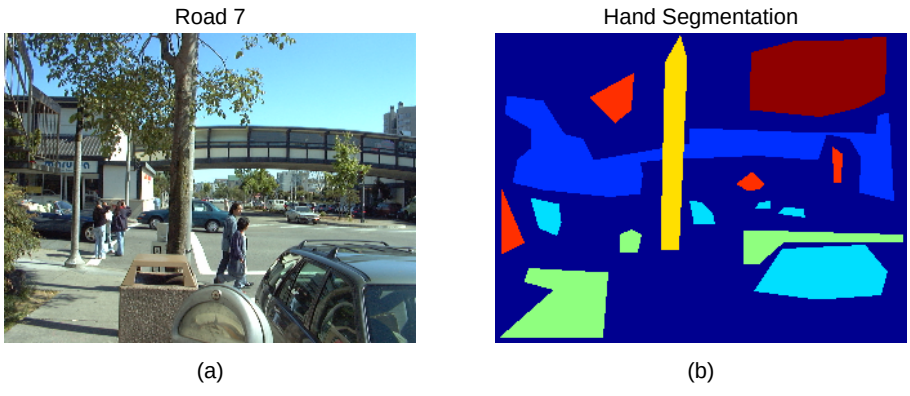


Figure 6.7: The single image used for training on the San Francisco database and the associated hand segmented regions.

hand segmented regions. Using this setup, the Joint classifier achieves an accuracy rate of 97.9%, i.e. almost all the pixels are labelled correctly in the 36 test images. Figure 6.8 shows an example of a test image and the regions that were classified in it. This result again validates the fact that small image patches can be used to successfully classify textured images. In fact, using small patches is particularly appealing for databases such as the San Francisco set because large scale filter banks will have problems near region boundaries and will also not be able to produce many measurements for small, or irregularly shaped, regions.

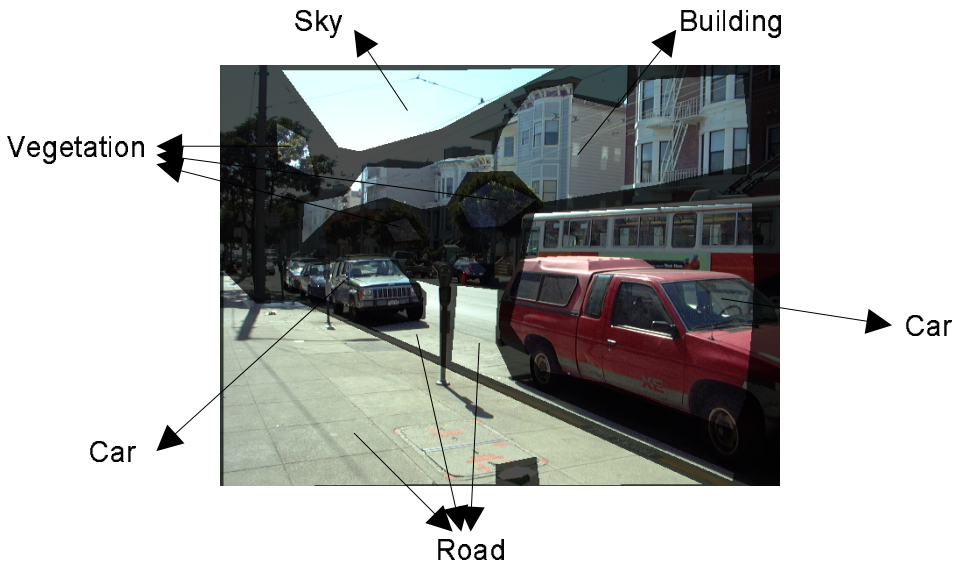


Figure 6.8: Region classification results using the Joint classifier with 7×7 patches for a sample test image from the San Francisco database.

6.4 Why does patch based classification work?

The results of the previous sections have demonstrated two things. Firstly, neighbourhoods as small as 3×3 can lead to very good classification results

even for textures whose global structure is far larger. Secondly, classification using image patches is superior to that using filter banks with equivalent support. In this section, we discuss some of the theoretical reasons as to why these results might hold. Even though a formal analysis could be carried out in terms of Markov random fields, to clarify the discussion we'll focus on the texture descriptors as being source image patches from which filter responses are derived.

6.4.1 Classification using small patches

One of the driving arguments for the use of large scale filter banks in texture classification has been that of feature extraction, i.e. that features at

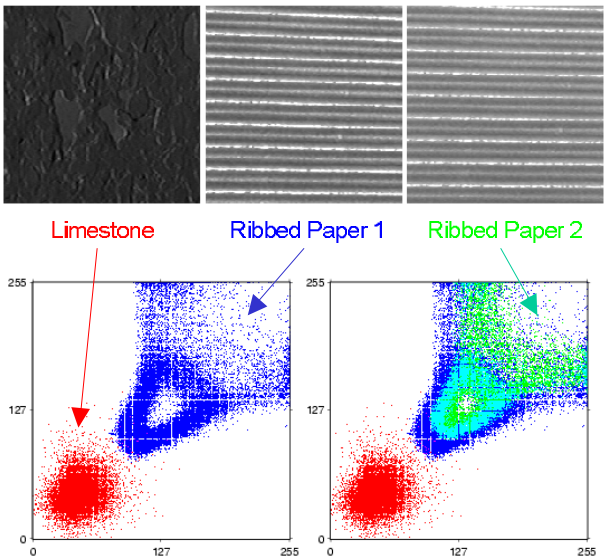


Figure 6.9: Information present in 3×3 neighbourhoods is sufficient to distinguish materials. The top row shows three images drawn from two texture classes, Limestone and Ribbed Paper. The bottom row shows scatter plots of $I(\mathbf{x})$ against $I(\mathbf{x}+(2,2))$. On the left are the distributions for Limestone and Ribbed Paper 1 while on the right are the distributions for all three images. The Limestone and Ribbed Paper distributions can easily be distinguished and hence the textures can be discriminated from this information alone.

many orientations and scales need to be extracted for successful classification. However, the results on the CUReT, San Francisco and Microsoft Textile databases show that this hypothesis is evidently not true and that small image patches contain enough information to discriminate between different textures. The explanation for this is illustrated in figure 6.9. Three images are selected from the Limestone and Ribbed Paper classes of the CUReT dataset, and scatter plots of their grey level co-occurrence matrix shown for the displacement vector $(2, 2)$ (i.e. the joint distribution of the top left and bottom right pixel in every 3×3 patch). Notice how the distributions of the two images of Ribbed Paper can easily be associated with each other and distinguished from the distribution of the Limestone image. Thus, 3×3 neighbourhood distributions can contain sufficient information for successful discrimination.

To take a more analytic example, consider two functions $f(x) = A\sin(\omega_f t +$

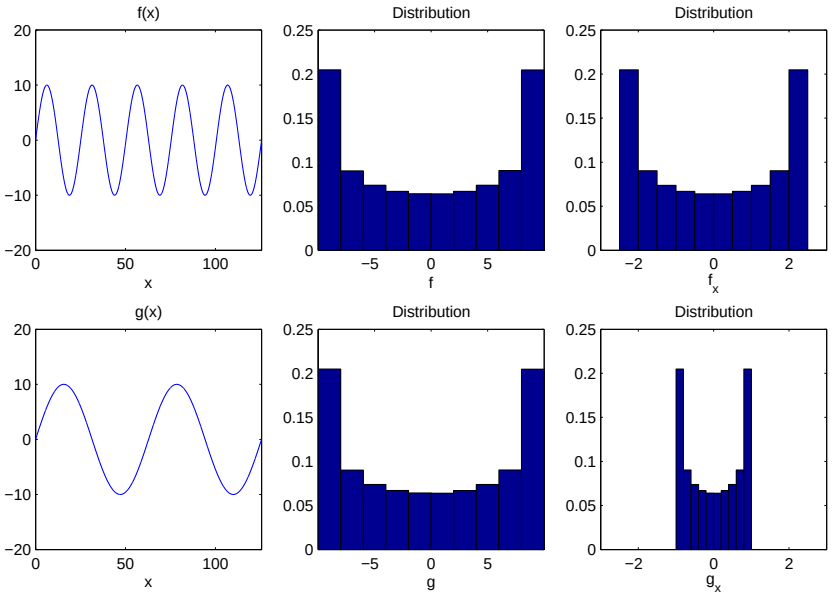


Figure 6.10: Similar large scale periodic functions can be classified using the distribution of their derivatives computed from two point neighbourhoods.

$\delta)$ and $g(x) = A\sin(\omega_g t + \delta)$, where ω_f and ω_g are small so that f and g have large structure. Even though f and g are very similar (they are essentially the same function at different scales) it can be shown that they are easily distinguished by the Joint classifier using only two point neighbourhoods. Figure 6.10 illustrates that while the intensity distributions of f and g are identical, the distributions of their derivatives, f_x and g_x, are not. Since derivatives can be computed using just two points, these two functions can be distinguished by looking at two point neighbourhoods alone.

In a similar fashion, other complicated functions such as triangular and saw tooth waves can be distinguished using compact neighbourhoods. Not only that, the Taylor series expansion of a polynomial of degree $2N-1$ immediately shows that a $[-N, +N]$ neighbourhood contains enough information to *determine* the value of the central pixel. Thus, any function which can be locally approximated by a cubic polynomial can actually be synthesised using a $[-2, 2]$ neighbourhood (see figure 6.11 for example). Since, in general, synthesis requires much more information than classification it is therefore expected that more complicated functions can still be distinguished just by looking at small neighbourhoods. This illustrates why it is possible to classify very large scale textures using small patches.

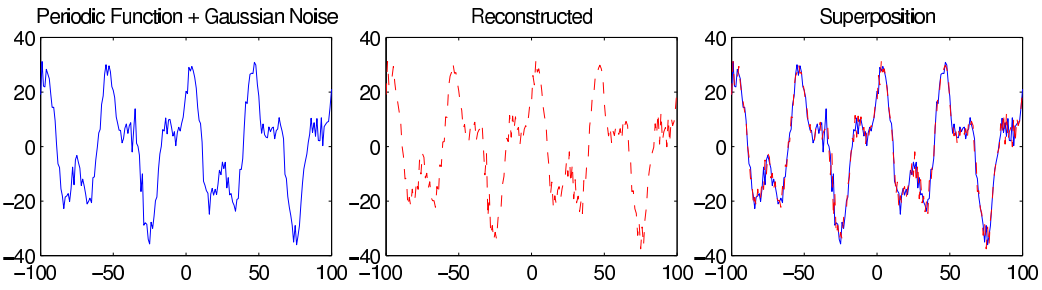


Figure 6.11: Small neighbourhoods can be used to not just discriminate but even synthesise large scale functions which can locally be approximated by cubic polynomials.

There also exist entire classes of textures which can not be distinguished on the basis of local information alone. One such class comprises of textures made up of the same textons and with identical first order texton statistics but which differ in their higher order statistics. To take a simple example, consider texture classes generated by the repeated tiling of two textons (a circle and a square for instance) with sufficient spacing in between so that there is no overlap between textons in any given neighbourhood. Then, any two texture classes which differ in their tiling pattern but have identical frequencies of occurrence of the textons will not be distinguished on the basis of local information alone. However, the fact that classification rates of nearly 98% have been achieved using extremely compact neighbourhoods on three separate data sets indicates that such textures do not occur frequently in the real world.

6.4.2 Filter banks are not superior to image patches

We now turn to the question of why filter banks do not provide superior classification as compared to their source image patches. To fix the notation, $\mathbf{f}_+$ and $\mathbf{f}_-$ will be used to denote filter response vectors generated by projecting $N \times N$ image patches $\mathbf{i}_+$ and $\mathbf{i}_-$, of dimension $d = N^2$, onto a lower dimension N_f using the filter bank $\mathbf{F}$. Thus,

$$\mathbf{f}_{\pm_{N_f \times 1}} = \mathbf{F}_{N_f \times d}\ \mathbf{i}_{\pm_{d \times 1}} \tag{6.2}$$

In the discussion, we'll focus on the properties of linear (including complex) filter banks. This is not a severe limitation as most popular filters and wavelets tend to be linear. Non linear filters can also generally be decomposed into a linear filtering step followed by non linear post-processing.

Furthermore, since one of the main arguments in favour of filtering comes from dimensionality reduction, it will be assumed that $N_f < d$, i.e. the number of filters must be less than the dimensionality of the source image patch. Finally, it should be clarified that throughout the discussion performance will be measured by classification accuracy rather than the speed with which classification is carried out. While the time complexity of an algorithm is certainly an important factor and can be critical for certain applications, our focus is on achieving the best possible classification results.

The main motivations which have underpinned filtering are: dimensionality reduction, feature extraction and biological plausibility, noise reduction and invariance. Arguments from each of these areas are now examined to see whether filter banks can lead to better performance than image patches.

Dimensionality reduction

Two arguments have been used from dimensionality reduction. The first, which comes from optimal filtering, is that an optimal filter can increase the separability between key filter responses from different classes and is therefore beneficial for classification. The second argument, from statistical machine learning, is that reducing the dimensionality is desirable because of better parameter estimation (by improved clustering or maximised independence) and also due to regularization effects which smooth out noisy filter responses and prevent over-fitting. We examine both arguments in turn to see whether such factors can compensate for the inherent loss of information associated with dimensionality reduction.

Increasing separability Since convolution with a linear filter is equivalent to linearly projecting onto a lower dimensional space, the choice of projection

direction determines the distance between the filter responses. Suppose we have two image patches $\mathbf{i}_\pm$, with filter responses $\mathbf{f}_\pm$ computed by orthogonal projection as $\mathbf{f}_\pm = \mathbf{F}\mathbf{i}_\pm$ (where the rows of $\mathbf{F}$ span the hyperplane orthogonal to the projection direction). Then the distance between $\mathbf{f}_+$ and $\mathbf{f}_-$ is clearly less than the distance between $\mathbf{i}_+$ and $\mathbf{i}_-$. The choice of $\mathbf{F}$ affects the separation between $\mathbf{f}_+$ and $\mathbf{f}_-$, and the optimum filter maximises it, in the manner of a Fisher Linear Discriminant, but the scaled distance between the projected points cannot exceed the original. This result holds true for many popular distance measures including the Euclidean, Mahalanobis and the signed perpendicular distance used by linear SVMs and related classifiers (analogous results hold when $\mathbf{F}$ is not orthogonal). It is also well known [Kohavi and John, 1997] that under Bayesian classification, the Bayes error either increases or remains at least as great when the dimensionality of a problem is reduced by linear projection. However, the fact that the Bayes error has increased for the low dimensional filter responses does not mean the classification is necessarily worse. This is because of issues related to noise and over-fitting which brings us to the second argument from dimensionality reduction for the superiority of filter banks.

Improved parameter estimation The most compelling argument for the use of filters comes from statistical machine learning where it has often been noted that dimensionality reduction can lead to fewer training samples being needed for improved parameter estimation (better clustering) and can also regularise noisy data and thereby prevent over-fitting. The assumptions underlying these claims are that textures occupy a low dimensional subspace of image patch space and if the patches could be projected onto this true subspace (using a filter bank) then the dimensionality of the problem would

be reduced without resulting in any information loss.

While these are undoubtedly sound claims there are three reasons why they might not lead to the best possible classification results. The first is due to the great difficulty associated with identifying a texture's true subspace (in a sense, this itself is one of the holy grails of texture analysis). More often than not, only approximations to this true subspace can be made and these result in a frequent loss of information when projecting downwards.

The second counter argument comes from the recent successes of Kernel methods. Dimensionality reduction is necessary if one wants to accurately model the true texture PDF. However, Kernel methods have demonstrated that for classification purposes a better solution is to actually project the data non-linearly into an even higher (possibly infinite) dimensional space where the seperability between classes is increased. Thus the emphasis is on maximising the distance between the classes and the decision boundary rather than trying to accurately model the true texture PDF (which, though ideal, is impractical). Implemented properly, the kernel trick can lead to both improved classification and generalisation without much associated overhead and with none of the associated losses of downward projection. The reason this argument is applicable in our case is because it can be shown that χ^2, with some minor modifications, can be thought of as a Mercer kernel [Wallraven et al., 2003]. Thus, the patch based classifiers take the distribution of image patches and project it into the much higher χ^2 space where classification is carried out. The filter bank based VZ algorithm does the same but it first projects the patches onto a lower dimensional space which results in a loss of information. This is the reason why the performance of filter banks studied here is consistenly inferior to their source patches.

The third argument is an engineering one. While it is true that clus-

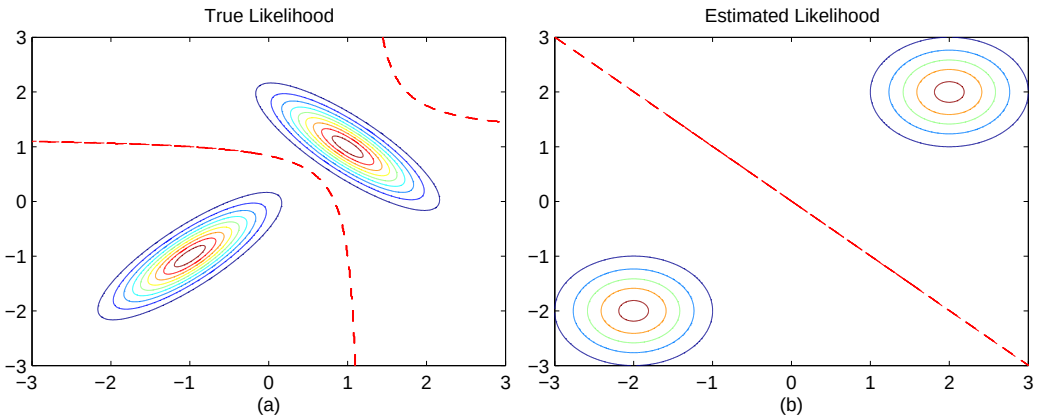


Figure 6.12: Incorrect parameter estimation can still lead to good classification results: the true likelihoods of two classes are shown in (a) along with the MAP decision boundary obtained using equal priors (dashed red curves). In (b) the estimated likelihoods have gross errors. The estimated means have relative errors of 100% and the covariances are estimated as being diagonal leading to a very different decision boundary. Nevertheless the expected classification error is just 1.4% and thus 98.6% of the data will be classified correctly despite the poor parameter estimation.

tering is better and that parameters are estimated more accurately in lower dimensional spaces, [Domingos and Pazzani, 1997] have shown that even gross errors in parameter estimation can have very little effect on classification. This is illustrated in figure 6.12 which shows that even though the means and covariance matrices of the true likelihood are estimated inccrrectly, 98.6% of the data is still correctly classified. This is supported by the results plotted in figure 6.4 which show that classification obtained by clustering 441 dimensional 21×21 patches gives better results than those obtained by clustering the 8 dimensional MR8 filter responses with equivalent support (in fact, 21×21 patches are also better than the 9 dimensional 3×3 patches even though fewer measurements are available for accurate parameter estimation in the higher dimensional space). Another interesting result, which supports the view that accurate parameter estimation is

not necessary for accurate classification, is obtained by selecting the texton dictionary at random (rather than via *K-Means* clustering) from amongst the filter response vectors. In this case, the classification result for VZ MR8 drops by only 5% and is still well above 90%. A similar phenomenon was observed by [Georgescu et al., 2003] when *Mean-Shift* clustering was used to approximate the filter response PDF. Thus the loss due to inaccurate parameter estimation in high dimensions might still be less than the loss associated with projecting into a lower dimensional subspace even though clustering may be improved.

Feature extraction

The main argument from feature extraction is that many features at multiple orientations and scales must be detected accurately for successful classification. Furthermore, studies of early vision mechanisms and pre-attentive texture discrimination have suggested that the detected features should look like edges, bars, spots and rings. These have most commonly come to be implemented using Gabor or Gaussian filters and their derivatives. However, results from the previous sections have shown that the multi-scale, multi-orientation large support filter bank argument is not valid. Small image patches can also lead to successful classification. Furthermore, while an optimally designed bank might be maximising some measure of seperability in filter space, it is hard to argue that "off the shelf" filters such as BFS, LM or S (whether biologically motivated or not) are the best for any given classification task. In fact, as has been demonstrated, a classifier which learns from all the input data present in an image patch should do better than one which depends on these pre-defined features bases.

Noise reduction and invariance

Most filters have the desireable property that, because of their large smoothing kernels (such as Gaussians with large sigma), they are fairly robust to noise. This property is not shared by image patches. However, pre-processing the data can solve this problem. For example, the classifiers developed in this chapter rely on vector quantisation of the patches into textons to help cope with noise. This can actually provide a superior alternative to filtering, because even though filters reduce noise, they also smooth over all the high frequency information present in the signal. Yet, as has been demonstrated in the 3×3 patch case, this information can be beneficial for classification. Therefore, if image patches can be denoised by pre-processing or quantization without the loss of high frequency information then they should provide a superior representation for classification as compared to filter banks.

Virtually the same argument can be used to build invariance into the patch representation without losing information by projecting onto lower dimensions. For example, patches are pre-processed and made to have zero mean and unit standard deviation to achieve invariance to affine transformations in the illuminant's intensity. Similarly, to achieve rotational invariance, the dominant orientation can be determined and then corrected for by reduction to a canonical frame. However, this does have the drawback of being potentially unstable if the dominant direction cannot be determined accurately. For instance, corners have two dominant orientations and, in the presence of noise, can be transformed incorrectly upon reduction to the canonical frame. One solution to the problem could be to discard such ambiguous patches altogether. Alternatively, many transformed copies of the patch (for instance, all rotated versions) can be included in the training set to overcome this problem.

6.5 Conclusions

Filter banks and wavelets have become ubiquitous in the texture classification literature over the last decade or so. Though there are many reasons for their popularity their use in particular classification problems has not always been justified. The work in this chapter, following that of Efros and Leung, demonstrates that for tasks such as synthesis and classification, filter banks are sufficient but *not* necessary and that their performance while tackling either task is inferior.

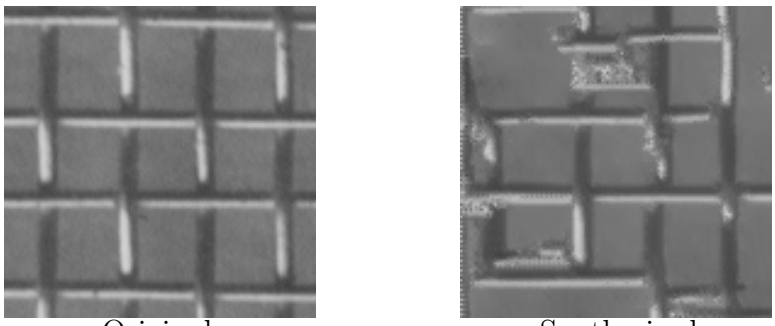
Indeed, filter banks have a number of disadvantages compared to smaller image patches: first, the large support they require means that far fewer samples of a texture can be learnt from training images (there are many more 3×3 neighbourhoods than 50×50 in an 100×100 image). Second, the large support is also detrimental in texture segmentation, where boundaries are localised less precisely due to filter support straddling region boundaries; A third disadvantage is that the blurring (e.g. Gaussian smoothing) in many filters means that fine local detail can be lost. This is another reason why the image patch based classifier achieves superior results as compared to the VZ algorithm using the large scale MR8 filter bank.

The disadvantage of the patch representation is the quadratic increase in the dimension of the feature space with the size of the neighbourhood. This problem may be tackled by using a multi-scale representation. For instance, an image pyramid could be constructed and patches taken from several layers of the pyramid if necessary. An alternative would be to use large neighbourhoods but store the pixel information away from the center at a coarser resolution. Finally, a scheme such as Zalesny and Van Gool's [Zalesny and Van Gool, 2000] could be implemented to determine which long range interactions were important and use only those cliques.

Before concluding, it is worth while to reflect on how the image patch algorithms and their results relate to what others have observed in the field. In particular, [Fowlkes et al., 2003, Levina, 2002, Randen and Husoy, 1999] have all noted that in their respective texture analysis tasks, filters with small support have outperformed the same filters at larger scales. Thus, there appears to be an emerging consensus that small support is not necessarily detrimental to performance.

Another interesting fact is that the "new" image patch algorithms, such as the synthesis method of Efros and Leung and the Joint classifier developed in this chapter, have actually been around for quite a long time. For instance, Efros and Leung note a strong resemblance between their algorithm and that of [Garber, 1981]. Furthermore, both the Joint classifier and Efros and Leung's algorithm are near identical in spirit to [Popat and Picard, 1993]. The relationship between the Joint classifier and Popat and Picard's algorithm is particularly close as both use clustering to learn a distribution over image patches which then forms a model for novel texture classification. Apart from the choice of neighbourhoods, the only minor differences between the two methods are in the representation of the PDF and the distance measure used during classification. Popat and Picard use a Gaussian mixture model with diagonal covariances to represent their PDF while the texton representation used in this thesis can be thought of as fitting a spherical Gaussian mixture model via *K-Means*. During classification, Popat and Picard use a *naïve* Bayesian method which, for the Joint classifier, would equate to using nearest neighbour matching with KL divergence instead of the χ^2 statistic (as shown in chapter 5).

Certain similarities also exist between the Joint classifier and the MRF model of [Cross and Jain, 1983]. In particular, Cross and Jain were the



Original Synthesised

Figure 6.13: Synthesis results from [Popat and Picard, 1993].

first to recommend that χ^2 over the distribution of central pixels and their neighbours could be used to determine the best fit between a sample texture and a model. Had they actually used this for classification rather than just model validation of synthesised textures, the two algorithms would have been very similar apart from the functional form of the PDFs learnt (Cross and Jain treat the conditional PDF of the central pixel given the neighbourhood as a unimodal binomial distribution).

Thus, alternative approaches to filter banks have been around for quite some time. Perhaps the reason that they didn't become popular then was due to the great computational costs they required to achieve good results. For instance, the synthesis results of [Popat and Picard, 1993] are of a poor quality which is perhaps why their theory didn't attract the attention it deserved. Figure 6.13 shows Popat and Picard's result obtained using a dictionary of 2048 textons and with a 14 pixel, causal image patch applied across consecutive scales. The same texture has been synthesised in figure 6.5 using our image patch representation with 300 textons and 11×11 neighbourhoods [Varma and Zisserman, 2003]. The improved results are due to the significantly larger neighbourhoods used – an option perhaps not available to [Popat and Picard, 1993] in their day. However, with computational power being readily accessible today, MRF and image patch methods are

outperforming filter bank based methods. Though it remains to be seen whether the trend will be reversed if the practical classification problems being attempted become much more complex without a matching increase in processor speeds and availability – for example, when moving to real time or embedded systems.

To conclude, in this chapter we have introduced a texton based image patch representation for textures and demonstrated that superior classification results can be obtained by using compact, local neighbourhoods and without the use of large scale filter banks.

Chapter 7

Estimating Illumination Direction

This chapter studies the the problem of estimating the illuminant's direction from images of textured surfaces. The goal is to overcome our lack of prior knowledge and develop a robust method which can be used to infer the illuminant's azimuthal angle and thereby aid future classification.

The problem is considered in a statistical framework unlike the more traditional geometric formulations, Given an isotropic, Gaussian random surface with constant albedo, Koenderink and Pont [Koenderink and Pont, 2003] developed a theory for recovering the illuminant's azimuthal angle from a single image of the texture formed under a Lambertian model. In this chapter, the theory is extended to deal with cases of spatially varying albedo. This extension also allows the theory to better handle real world phenomenon such as shadows, specularities, inter-reflections, etc.

7.1 Introduction

This thesis has not explicitly considered how to overcome the lack of prior knowledge about viewpoint and illumination conditions while attempting classification so far. Using rotation and scale invariant descriptors has enabled the system to implicitly cope with viewpoint changes to a certain extent. However, as figure 1.14 shows, illumination changes can also cause significant variations in the appearance of a material. Thus an algorithm capable of detecting the illuminant's direction can be used to help model such changes and provide explicit information about the illumination properties of the scene. In this chapter, we take a first step towards developing such an algorithm and address the problem of estimating the illuminant's azimuthal angle ψ from images of textured rough surfaces (see figure 7.1).

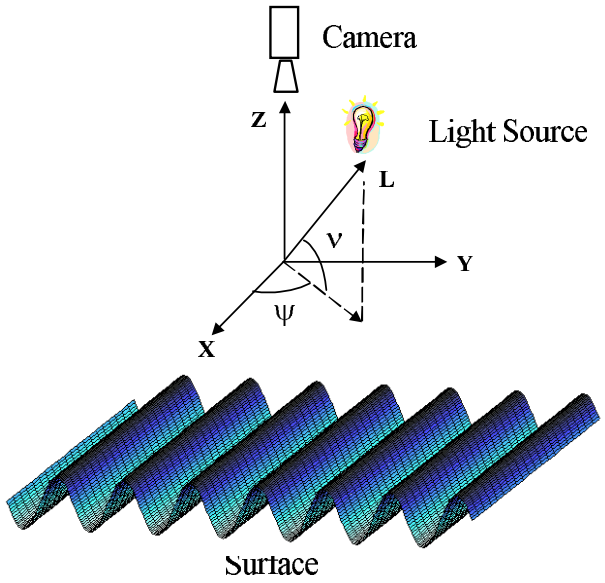


Figure 7.1: The goal is to estimate the illuminant's azimuth ψ from images of textured rough surfaces.

It should be stressed that the aim is not to build an illumination estimation algorithm in itself. If this were the case, one could have taken any

of the successful classification systems of the previous chapters, used it to determine the closest model in the training set and returned the model's illumination direction as an approximation of the illumination direction of the given test image. On the contrary, our goal is to ultimately use the illumination estimation algorithm to improve classification. And, in particular, use it in situations where only a few training models are available in which case the direction approximation returned by the classification method would be very crude and often inaccurate (as turns out to be the case on tests on the CUReT database).

Traditionally, techniques from Shape from Shading have been used for estimating the illuminant's direction [Brooks and Horn, 1985, Lee and Rosenfeld, 1985, Nillius and Eklundh, 2001, Pentland, 1982, Vega and Yang, 1994, Yang and Yuille, 1991, Zheng and Chellappa, 1991]. Most of these techniques assume a Lambertian [Foley et al., 1990] image formation model and try to simultaneously recover both shape, i.e. the surface height map or the surface normals, and the direction of the light source. However, this is an ill posed problem and many constraints have to be imposed in order to find a reasonable solution. Some of the most common constraints are that the albedo must be constant and that the surface be smooth or the normals integrable. Alternatively, other methods focus on local estimates or the occluding contour but, once again, have to impose very similar constraints to determine the illuminant's direction.

Recently, methods have been developed which specifically exploit the statistical nature of rough textures. [Chantler et al., 2002b, Chantler et al., 2002a] have shown that the variance of filter responses obtained from a textured image lie on Lissajous' ellipses as a function of the illuminant's azimuthal angle. Given three reference images of the texture, taken under

fixed viewpoint and illuminant elevation, it is possible to determine the ellipse. This ellipse can then be used to read off the illuminant's azimuthal angle for any novel image of that texture. Similarly, [Koenderink and Pont, 2003] develop a statistical theory based on second order moments of the surface gradients to recover the illuminant's azimuth from a single view taken under orthographic projection.

However, none of these methods has translated into a practical tool to assist in classification. The reason is primarily due to the imposition of strong constraints which are violated in most real world situations. In particular, the assumption that the texture must have constant albedo is severely restrictive and limits the applicability of such methods. In section 7.2, we take a first step towards removing this restriction by generalizing Koenderink and Pont's theory to the case where the albedo can be thought of as a spatially varying random variable drawn from a log-normal distribution [Evans et al., 2000]. The extension also accommodates, in certain cases, the effects of factors such as attached shadows, specularities and other deviations from the perfect Lambertian model. This permits the application of the theory to all 5612 images chosen from the CUReT database and it is demonstrated in section 7.3 that extremely good results are achieved on real world, uncalibrated images taken under a variety of conditions. Next, section 7.4 explores how the theory can be further generalized to take into account arbitrarily varying albedo if extra information is present in the form of an additional reference image. The theory is then tested in section 7.5 on the Heriot-Watt TextureLab database, where additional reference images are available, and it is demonstrated that superior results are achieved with the new formulation. In section 7.6 the advantages of using local regions to form estimates of the azimuthal angle are investigated. Finally, section 7.7 concludes with a

discussion on the implications of automatically determining the illuminant's azimuth for resolving the Generalized Bas-Relief ambiguity.

7.2 Estimating the light source azimuth

This section develops the basic theory for recovering the illuminant's azimuth from a single texture image. We consider the case where the underlying texture can be modelled as a Gaussian, random, rough surface. None of the parameters of the surface, the mean, variance or even the autocorrelation function, need actually be known. Instead by making general assumptions about the surface height distribution, the second order statistics of the surface derivatives can be used to robustly recover the light source azimuth. In particular, it will be shown that the structure tensor $\mathbf{S} = < (\nabla \log I)(\nabla \log I)^T >$ has its larger eigenvector $\mathbf{v}_1$ pointing in the direction of the illuminant's azimuth, i.e. $\mathbf{v}_1 = \left[\begin{smallmatrix} \cos\psi \\ \sin\psi \end{smallmatrix} \right]$ from which ψ can be recovered. The derivation will follow principally along the lines of [Koenderink and Pont, 2003].

7.2.1 Theoretical assumptions and their validity

Under the basic assumptions that the underlying model which produced the textured image has (a) an isotropic, Gaussian random rough surface with shallow relief viewed orthographically, (b) an albedo which is also isotropic but distributed log-normally, (c) an illuminant whose elevation ν is high as compared to the surface tangent plane, and (d) a perfect Lambertian image formation model without shadowing, specularities or inter-reflections, it will be shown that the illuminant's azimuthal angle ψ can be recovered from the largest eigenvector of the structure tensor $\mathbf{S}$.

As these assumptions might appear to be overly restrictive, it will be demonstrated that the theory holds even for cases when the textures deviate strongly from this model. For example, the results are empirically valid for elevations as small as $\nu = 5^\circ$ and when there are significant shadows. We will explain why this might be the case by considering the situation where the effects of shadowing, specularities, inter-reflections etc. can be incorporated into the albedo map. It should also be noted that requiring the surface have a Gaussian height distribution or the albedo a log-normal distribution are not major restrictions. [Thomas, 1998] indicates that many naturally occurring rough surfaces can be treated as random variables drawn from a Gaussian distribution. Figure 7.2 shows some synthesised examples of homogenous, anisotropic and rotationally symmetric rough surfaces with Gaussian height

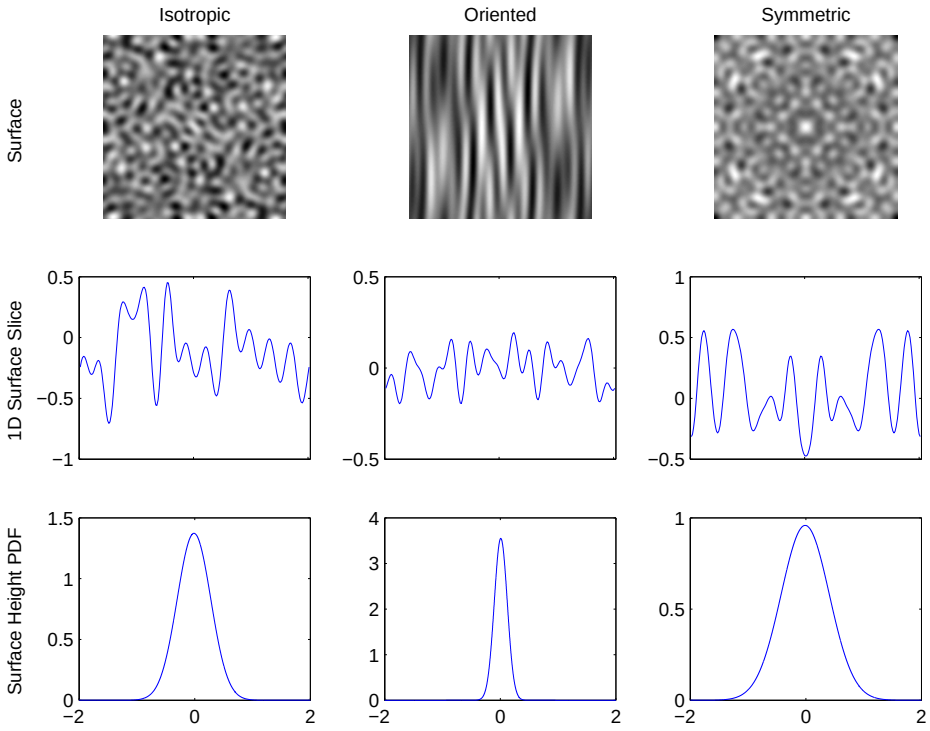


Figure 7.2: Synthetic Gaussian random rough surfaces generated using (7.5) with a corresponding 1D horizontal slice and surface height distribution.

profiles. Similarly, requiring that the albedo be log-normal distributed is a plausible assumption easily satisfied by many different albedo maps (see figure 7.3).

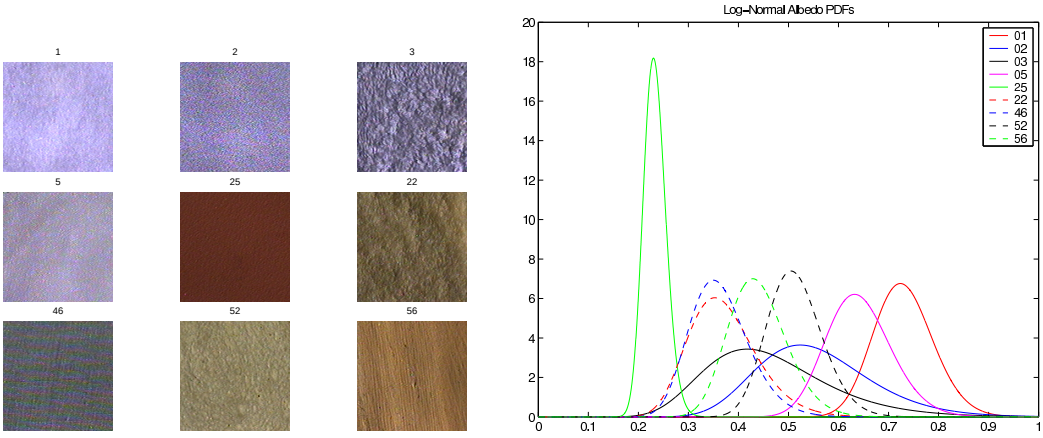


Figure 7.3: On the left are images of nine different materials from the CUReT database. Treating each of the images as an albedo map, the graph on the right plots its distribution of pixel intensities. The distributions can be modelled very accurately by log-normal PDFs.

7.2.2 Derivation of the basic theory

If a textured surface is imaged under the Lambertian model [Foley et al., 1990], then the image intensities are independent of the viewing direction and depend on only the angle between the surface normal at each point and the light source direction. When there is a single, collimated, parallel light source, relatively high enough from the surface tangent plane so that shadows can be neglected, the image intensities are given by

$$I(x,y) = \frac{\rho(x,y)L_\lambda \sin\nu}{\sqrt{1+h_x^2+h_y^2}}[1-\cot\nu(h_x\cos\psi + h_y\sin\psi)] \qquad (7.1)$$

where $\mathbf{L} = L_\lambda[\cos\nu\cos\psi, \cos\nu\sin\psi, \sin\nu]$ is the light source vector with elevation ν and azimuthal angle ψ, $h(x,y)$ is the Monge patch parameter-

ization of the surface height with partial derivatives $h_x(x,y)$ and $h_y(x,y)$, and $\rho(x,y)$ is the spatially varying surface albedo. Thus only a very simple image formation model is being considered and effects due to specularities, inter-reflections and shadows are neglected for the moment. Yet, as will be demonstrated, even this simple analysis can give very good results on real world datasets.

If the surface has shallow relief then the factor in the denominator can be ignored as $h_x, h_y \ll 1$. Following [Koenderink and van Doorn, 2002, Koenderink and Pont, 2003], we work with the log intensity distribution given by

$$\log I(x,y) = \log(\rho L_\lambda \sin\nu) - \cot\nu(h_x \cos\psi + h_y \sin\psi) \tag{7.2}$$

where the fact that $\cot\nu$ is small has been used to form the truncated Taylor series expansion $\log(1-x) = -x$. Denoting $LI = \log I, s = \sin\psi, c = \cos\psi$ and taking partial derivatives gives

$$LI_x(x,y) = \frac{\rho_x}{\rho} - \cot\nu(ch_{xx} + sh_{xy}) \tag{7.3}$$

$$LI_y(x,y) = \frac{\rho_y}{\rho} - \cot\nu(ch_{xy} + sh_{yy}) \tag{7.4}$$

Generic information about the surface height and albedo distributions is now needed in order to proceed further with the analysis. In [Longuet-Higgins, 1957, Berry and Hannay, 1977], it is shown that a Gaussian random rough surface can be generated by the interaction of a number of waves at different frequencies and orientations. Thus,

$$h(x,y) = \sum_n \sum_m h_{nm} \cos(nx + my) \tag{7.5}$$

where $n, m \in \mathbb{Z}$ and h_{nm} are random variables which determine the auto-correlation of the rough surface. Figure 7.2 shows some sample Gaussian random rough surfaces which can be expressed as (7.5). Since a Gaussian surface must have an equal number of protrusions and indentations, the first order statistics such as the mean will not reveal any information about the illuminant's azimuth (because the bright image regions will cancel out the dark image regions). Mathematically, $< LI_x >$ and $< LI_y >$ should vanish as the expected values of all partial derivatives of h must be equal to zero. Hence we turn to the square terms $< LI_x^2 >$, $< LI_y^2 >$ and $< LI_x LI_y >$ which become

$$\begin{aligned} < LI_x^2 > \;\; = \;\; & < (\rho_x/\rho)^2 > \\ & + \cot^2 \nu < (ch_{xx} + sh_{xy})^2 > \\ & -2 \cot \nu (c < \rho_x h_{xx}/\rho > + s < \rho_x h_{xy}/\rho >) \end{aligned} \tag{7.6}$$

To account for the albedo, a similar kind of assumption is made about its distribution. If the albedo can be modelled as a random variable with a log-normal distribution [Evans et al., 2000], then $\log \rho$ should also be of the form (7.5) and therefore the third term in (7.6) must vanish as the product of any odd and even numbered derivatives has zero expected value. Denoting, $A_x =< (\rho_x/\rho)^2 >$ we then have,

$$< LI_x^2 > \;\; = \;\; A_x + \cot^2 \nu < (ch_{xx} + sh_{xy})^2 > \tag{7.7}$$

$$< LI_y^2 > \;\; = \;\; A_y + \cot^2 \nu < (ch_{xy} + sh_{yy})^2 > \tag{7.8}$$

where $< LI_y^2 >$ has been obtained by a similar treatment. The expression

for $< LI_x LI_y >$ is also very similar

$$< LI_x LI_y >= A_{xy} + \cot^2 \nu < (ch_{xx} + sh_{xy})(ch_{xy} + sh_{yy}) >$$

where $A_{xy} =< \rho_x \rho_y / \rho^2 >$.

The expectations of the height derivatives now need to be evaluated. Some straight forward trigonometry and integration yields

$$\begin{aligned}
< h_{xx}^2 > &= (1/2) \sum_n \sum_m n^4 h_{nm}^2 \\
< h_{yy}^2 > &= (1/2) \sum_n \sum_m m^4 h_{nm}^2 \\
< h_{xy}^2 > &= (1/2) \sum_n \sum_m n^2 m^2 h_{nm}^2 =< h_{xx} h_{yy} > \\
< h_{xx} h_{xy} > &= (1/2) \sum_n \sum_m n^3 m h_{nm}^2 \\
< h_{yy} h_{xy} > &= (1/2) \sum_n \sum_m n m^3 h_{nm}^2
\end{aligned} \tag{7.9}$$

At this point, there are more unknowns than equations and therefore the system must be constrained further for the light source azimuth to be recovered. One way of reducing the number of free variables is by constraining the underlying surface and albedo. In the case that both are isotropic, the expectations in (7.9) can be greatly simplified to

$$\begin{pmatrix} < h_{xx}^2 > & < h_{xx} h_{yy} > & < h_{xx} h_{xy} > \\ < h_{yy} h_{xx} > & < h_{yy}^2 > & < h_{yy} h_{xy} > \\ < h_{xy} h_{xx} > & < h_{xy} h_{yy} > & < h_{xy}^2 > \end{pmatrix} = H \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7.10}$$

while the albedo expectations simplify to

$$\begin{pmatrix} A_x & A_{xy} \\ A_{yx} & A_y \end{pmatrix} = A \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{7.11}$$

where H and A are constants which depend on the surface height and albedo of the textured material (for instance, $A = 0$ for constant albedo textures). Substituting these values back into the expressions for $< LI_x^2 >$, $< LI_x^2 >$ and $< LI_x LI_y >$ gives

$$\begin{aligned} < LI_x^2 > &= A + H \cot^2 \nu (3 \cos^2 \psi + \sin^2 \psi) \\ < LI_y^2 > &= A + H \cot^2 \nu (\cos^2 \psi + 3 \sin^2 \psi) \\ < LI_x LI_y > &= H \cot^2 \nu (1 + 1) \sin \psi \cos \psi \end{aligned}$$

There are now exactly three equations in three unknowns and therefore it is possible to recover the illuminant azimuth ψ from the eigenvectors of the structure tensor [Koenderink and Pont, 2003] defined as

$$\mathbf{S} = < \nabla LI \ \nabla LI^T > = \begin{pmatrix} < LI_x^2 > & < LI_x LI_y > \\ < LI_x LI_y > & < LI_y^2 > \end{pmatrix} \tag{7.12}$$

In the present case, the structure tensor turns out to have a very simple form

$$\mathbf{S} = A\mathbf{I} + H \cot^2 \nu \begin{pmatrix} 2 + \cos 2\psi & \sin 2\psi \\ \sin 2\psi & 2 - \cos 2\psi \end{pmatrix} \tag{7.13}$$

where $\mathbf{I}$ is the 2×2 identity matrix. The larger eigenvalue and corresponding

eigenvector of the structure tensor are given by

$$\lambda_1 = A + 3H\cot^2\nu \Rightarrow \mathbf{v}_1 = \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix} \tag{7.14}$$

while the smaller eigenvalue and eigenvector are given by

$$\lambda_2 = A + H\cot^2\nu \Rightarrow \mathbf{v}_2 = \begin{bmatrix} \cos(\psi+\pi/2) \\ \sin(\psi+\pi/2) \end{bmatrix} \tag{7.15}$$

Thus, $\mathbf{v}_1$ points in the direction of the illuminant's azimuthal component and represents the desired solution. However, note that there is an ambiguity of 180° in the recovered angle as $\mathbf{S}$ depends on 2ψ rather than ψ.

The *coherence* of the structure tensor $\mathbf{S}$ is defined to be

$$\text{coh} = \frac{\lambda_1^2 - \lambda_2^2}{\lambda_1^2 + \lambda_2^2} \tag{7.16}$$

$$= \frac{H\cot^2\nu(2A + 4H\cot^2\nu)}{A^2 + H\cot^2\nu(4A + 5H\cot^2\nu)} \tag{7.17}$$

and it gives a measure of the stability of the solution. From (7.17) it can be seen that the coherence depends upon both ν and A and it must always be less than or equal to 0.8. For example, when A^2 is negligible as compared to the second term in the denominator, the expression for the coherence simplifies to

$$\text{coh} = \frac{2A + 4H\cot^2\nu}{4A + 5H\cot^2\nu} \tag{7.18}$$

which varies between 0.5 and 0.8 depending on the elevation ν. Of course, if A^2 is not negligible then the coherence can be lower still.

7.2.3 Deviations from the perfect Lambertian model

The model up till now has been derived under the assumption of perfect Lambertian reflectance without any shadowing (see figure 7.4), specularities, inter-reflections etc. However, in general, it is not possible to distinguish these effects from albedo variations given just a single image (unless there is prior information available) [Forsyth and Zisserman, 1991, Koenderink and Van Doorn, 1983]. For example, it is not possible to tell apart dark regions due to shadows from dark regions due to low albedo from only one image. Therefore, it might be possible to model these effects as albedo variations as long as the distribution remains roughly log-normal (the log-normal distribution can accommodate a large number of low intensity shadow regions in the bulk of the distribution with the specularities fitting into the tail). In such a situation, (7.14), (7.15) and (7.17) will still hold and the largest eigenvector will point in the direction of the azimuth. However, A will now become a function of both ν and ψ as well as the camera position and therefore the coherence is no longer expected to be a monotonic function of the elevation.

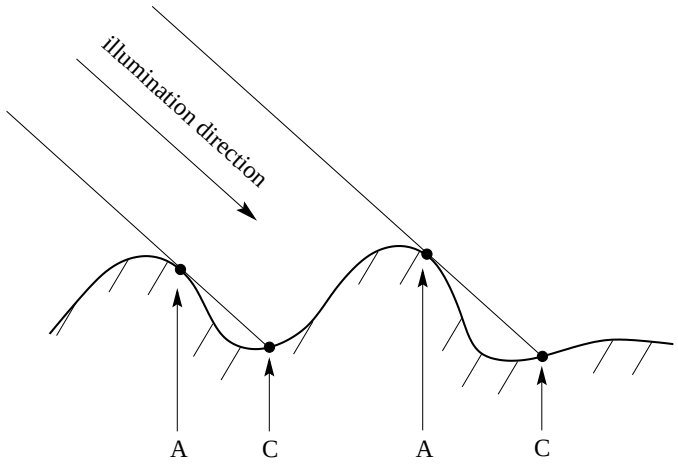


Figure 7.4: A indicates an attached (self) shadow boundary, and C a cast shadow boundary. Significant cast shadows can suddenly appear below a certain elevation for rough surfaces.

If we were to focus on shadowing as the major source of deviation from the the model, then depending on how quickly A increases with decreasing ν, as compared to $H\cot^2\nu$, the coherence curve can either increase or decrease. It can also do both if the shadowing pattern changes after a certain elevation and one can expect kinks in the graph. Figure 7.5 plots some sample scenarios.

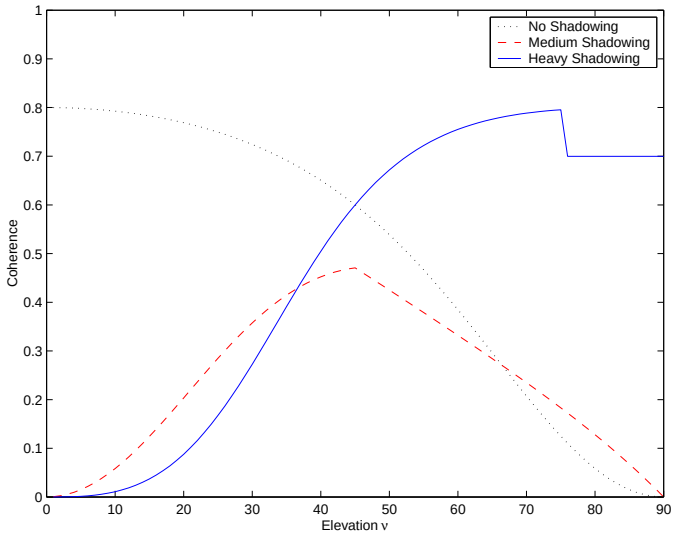


Figure 7.5: The variation in coherence with elevation in the presence of shadows. The curve can change dramatically if the shadowing pattern changes after a certain elevation. This can also cause a jump in the curve, for example, when significant cast shadows suddenly appear below a certain elevation. It should be noted that these are just a few sample curves from the set of all such possible. Each can vary considerably depending on how shadowing influences the albedo parameter A.

7.3 Single image experiments and comparisons

It is interesting to note that the eigenvectors recovered in (7.14) and (7.15) are identical to those found by Koenderink and Pont. Thus, even though

their theory was derived under the assumption of a constant albedo map, their method should hold for a much wider range of textures. However, since a constant albedo map implies $A = 0$, their model expects that the measured coherence should always equal 0.8 and should not change with varying elevation, azimuth or texture sample. The theory derived here predicts otherwise. For almost constant albedos, i.e. small A, (7.18) expects the coherence to lie between 0.5 and 0.8 with lower values being expected for larger variations in albedo. These predictions match very well with the "deviations" from the ideal as measured by Koenderink and Pont.

In the case of a true Gaussian random rough surface with painted white albedo, Koenderink and Pont report that the azimuth is estimated correctly within a few degrees but "the coherences are significantly lower" and vary between 0.4 and 0.7 with changing elevation. Similarly, on a sample texture from the CUReT database, the illuminants azimuth is detected to within a degree of the ground truth ($\psi = 0$) but the coherences are again found to be slightly lower with the 25 to 75 percentiles being 0.53 to 0.78. Both these results are regarded as anomalies by Koenderink and Pont while they are predicted almost exactly by (7.17) and (7.18).

Even though the current model has been derived by assuming Gaussian and log-normal distributions, it may also hold to some degree for other distributions for which the appropriate expected values cancel out. To determine how well the model copes with various materials, albedos and height distributions, it is applied to all the textures present in the CUReT database. We use the same set of 5612 cropped images that have been used in the classification experiments so far. In order to verify the robustness of the model, the images are not photometrically or geometrically calibrated but instead just converted to grey scale after which their raw pixel intensities are used.

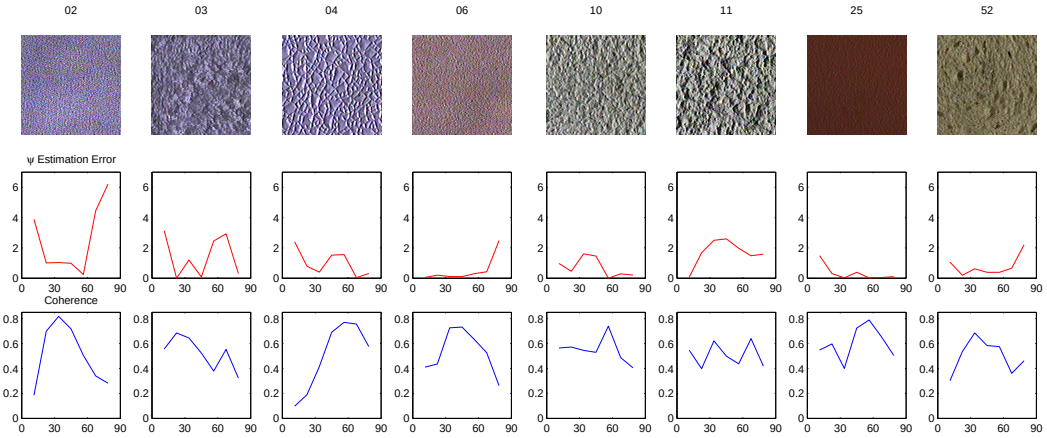


Figure 7.6: The top row shows some sample CUReT textures from a fronto-parallel view. For each material the middle row plots the error in estimating ψ in degrees as the illuminant's elevation varies from 11.25° to 78.75° (the viewing angle also varies but is always within 15° of the surface plane normal). The associated coherence values are plotted in the bottom row. The samples are: Polyester (texture number 02), Terrycloth (03), Rough Plastic (04), Sandpaper (06), Plaster A (10), Plaster B (11), Quarry Tile (25), and White Bread (52). Note that for each sample, derivatives are computed at various scales and the best result reported. No photometric or geometric calibration has been done and all images are converted to grey scale.

Figure 7.6 shows the results of the algorithm on some CUReT textures. For each texture, 7 images are chosen for which the viewing angle is within 15° of the surface plane normal. The value of the illuminant's azimuth is estimated using (7.14) and the estimation error in degrees is plotted as a function of ν in the middle row. The error is less than a few degrees even though the view is not perfectly normal, the albedo not constant and the surface not necessarily isotropic Gaussian. The results are valid even in the presence of shadows for the smaller values of elevation. In the bottom row, the associated coherences have also been plotted as a function of ν. As can be seen, they are not always equal to the constant value 0.8 determined by Koenderink and Pont but vary with ν and albedo as predicted by the theory developed here. The jumps in the curves are most probably due to shadows

introduced by the change in elevation but could also be due to other effects caused by changes in viewpoint.

Next, the method is applied to all 5612 images selected from the database and the light source azimuth estimated. As can be expected some results will not be very good due to the oblique viewpoint and the strong deviation of the textures from the assumptions. Nevertheless, the azimuth is recovered to within a few degrees for a majority of the cases. Figure 7.7 is a plot of the estimation error versus the number of images having that error. For 1475 images (more than 25%) the azimuth is estimated to within an accuracy of 1° while 3255 images (roughly 58% of those selected) have an error less than 5°.

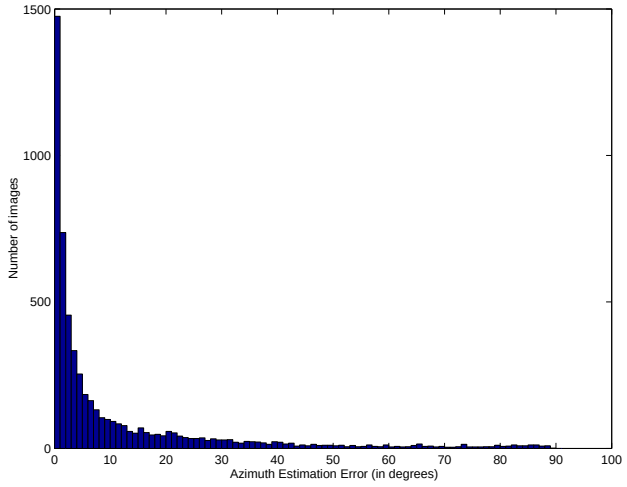


Figure 7.7: A count of the azimuth estimation errors (in degrees) for all 5612 images in the CUReT database. Results are given for the best scale for computing derivatives.

However, the algorithm does have a source of error which could be biasing these results. When a texture is strongly anisotropic, the perpendicular partial derivative dominates the structure tensor and forces the estimated illuminant to lie in its direction irrespective of the true azimuthal angle. For example, the iso-illumination contours for a texture with translational

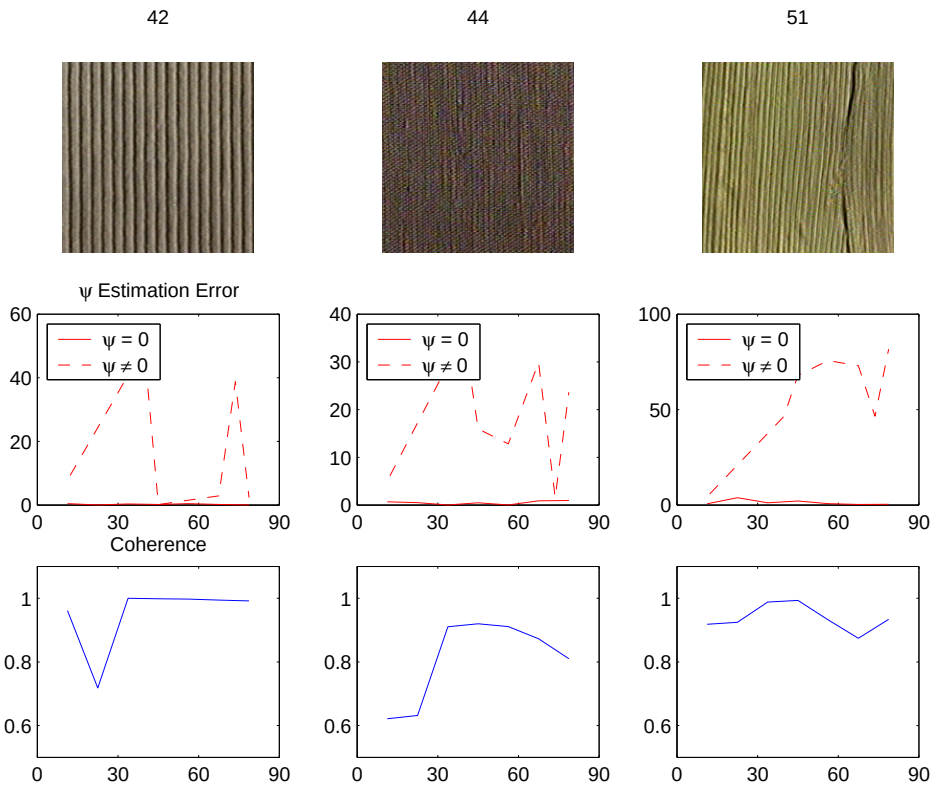


Figure 7.8: The model can appear to be working well even though it is being fooled by orientation effects. As long as the illuminant's true azimuth is around 0° the algorithm returns good results (solid red curve in the graphs in the middle row) for Corduroy (42), Linen (44) and Corn Husk (51). However, the estimates for all other azimuthal angles are very poor as can be seen by the dashed curve in the same graphs. The fact that the coherence is greater than 0.8 can be used to flag this error.

symmetry are straight lines parallel to the translation direction. Hence the derivatives in this direction will be negligible as compared to the perpendicular derivatives. So, for images which are vertically oriented (see figure 7.8), the x derivative becomes very large and forces the structure tensor to assume the form

$$\mathbf{S} = H \begin{pmatrix} 1 & \epsilon \\ \epsilon & \epsilon \end{pmatrix} \Rightarrow \lambda_1 = 1, \lambda_2 = 0, \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and thus the estimated azimuth is 0° irrespective of the actual direction of

the illuminant. A similar problem exists for horizontal textures and $\psi = 90°$. And since most illuminant directions in the CUReT database are either $\psi = 0°$, $\psi = 90°$ or $\psi = 180°$ it is difficult to tell whether the algorithm is working properly or giving erroneous results because of the dominance of oriented edges. However, in these cases the coherence will be *greater* than 0.8 and can therefore be used to flag errors. Figure 7.8 illustrates this effect. The algorithm seems to be working well as the estimated azimuth appears to lie very close to ground truth for $\psi = 0°$. However, in reality, it is the orientation effects which are causing this and once the true illuminant direction moves away from 0° the errors become very large. The fact that the coherences are greater than 0.8 can be used to flag this occurrence.

Nevertheless, the model appears to be quite robust when its basic conditions are met. For example, for Plaster A (texture number 10) which appears to be isotropic, the azimuth was estimated to within 5° nearly 90% of the times, irrespective of viewpoint and shadowing. Thus, even though there is room for improvement, the simple model derived without taking into account many physical phenomenon still appears to work quite well.

7.4 Estimation from two images

There are often cases when multiple images are available of a texture taken from the same viewpoint but with varying illumination. Photometric Stereo techniques rely on such data for example. In these cases, it is possible to use the extra information available to lift some of the restrictions imposed on the model in section 7.2. In particular, it is possible to have freely varying albedo and, in this section, a theory for estimating the illuminant's azimuth under such circumstances is developed.

Suppose there are available two registered images I_1 and I_2 imaged by varying the illuminant's azimuth. Then, under the Lambertian model, the image intensities are given by

$$I_i(x,y) = \frac{\rho(x,y)L_\lambda \sin\nu}{\sqrt{1+h_x^2+h_y^2}}[1-\cot\nu(h_x\cos\psi_i + h_y\sin\psi_i)]$$

Note that by taking the ratio of the two images, it is possible to immediately get rid of both the albedo variation as well as the normalising constant in the denominator. Thus, it is no longer necessary to make the explicit assumption that the surface has shallow relief in order to remove the $\sqrt{1+h_x^2+h_y^2}$ factor. Furthermore, the albedo can be allowed to vary arbitrarily as it has no influence on the ratio. Taking logarithms and again making use of the truncated Taylor series expansion gives

$$LR = \log(\frac{I_1}{I_2}) \tag{7.19}$$

$$= \cot\nu[h_x(\cos\psi_2 - \cos\psi_1) + h_y(\sin\psi_2 - \sin\psi_1)] \tag{7.20}$$

Denote $C = \cos\psi_2 - \cos\psi_1$ and $S = \sin\psi_2 - \sin\psi_1$. Then

$$\begin{aligned} LR &= \cot\nu(Ch_x + Sh_y) \\ \Rightarrow\ LR_x &= \cot\nu(Ch_{xx} + Sh_{xy}) \\ \Rightarrow\ LR_y &= \cot\nu(Ch_{xy} + Sh_{yy}) \end{aligned} \tag{7.21}$$

Again, $< LR_x >$ and $< LR_y >$ are not expected to contain any information and their values equal zero. Instead, one must look at the second order terms $< LR_x^2 >$, $< LR_y^2 >$ and $< LR_xLR_y >$. If the surface is isotropic and Gaussian, then $< h_{xx}^2 >=< h_{yy}^2 >= 3H, < h_{xy}^2 >=< h_{xx}h_{yy} >= H$ while all

other expectations are zero. Therefore,

$$\begin{aligned} < LR_x^2 > &= H \cot^2 \nu (3C^2 + S^2) \\ < LR_y^2 > &= H \cot^2 \nu (C^2 + 3S^2) \\ < LR_x LI_y > &= H \cot^2 \nu 2CS \end{aligned} \tag{7.22}$$

and the structure tensor is given by

$$\mathbf{S} = H \cot^2 \nu \begin{pmatrix} 3C^2 + S^2 & 2CS \\ 2CS & C^2 + 3S^2 \end{pmatrix} \tag{7.23}$$

Making use of the trigonometric identities $\cos(\psi_2 \pm \psi_1) = \cos\psi_2 \cos\psi_1 \mp \sin\psi_2 \sin\psi_1$, $\sin(\psi_2 \pm \psi_1) = \sin\psi_2 \cos\psi_1 \pm \cos\psi_2 \sin\psi_1$ and performing some careful, but straight forward, algebra yields

$$\mathbf{S} = \alpha \begin{pmatrix} 2 - \cos(\psi_1 + \psi_2) & -\sin(\psi_1 + \psi_2) \\ -\sin(\psi_1 + \psi_2) & 2 + \cos(\psi_1 + \psi_2) \end{pmatrix} \tag{7.24}$$

where

$$\alpha = 4H \cot^2 \nu \sin^2 \left(\frac{\psi_2 - \psi_1}{2} \right) \tag{7.25}$$

The eigenvalues of the structure tensor are now $\lambda_1 = 3\alpha$ and $\lambda_2 = \alpha$ while the larger eigenvector is

$$\mathbf{v}_1 = \begin{bmatrix} -\sin(\psi_1 + \psi_2) \\ 1 + \cos(\psi_1 + \psi_2) \end{bmatrix} \tag{7.26}$$

from which it is possible to recover the joint angle $\psi_1 + \psi_2$.

The coherence of the structure tensor now becomes

$$\text{coh} = \frac{\lambda_1^2 - \lambda_2^2}{\lambda_1^2 + \lambda_2^2} = 0.8 \tag{7.27}$$

7.5 Experimental results for two images

The validity of the theory developed in the previous section is now assessed on sample textures from the Heriot-Watt TextureLab database [Wu and Chantler, 2003]. The database is described in subsection 2.3.4 of the litera-

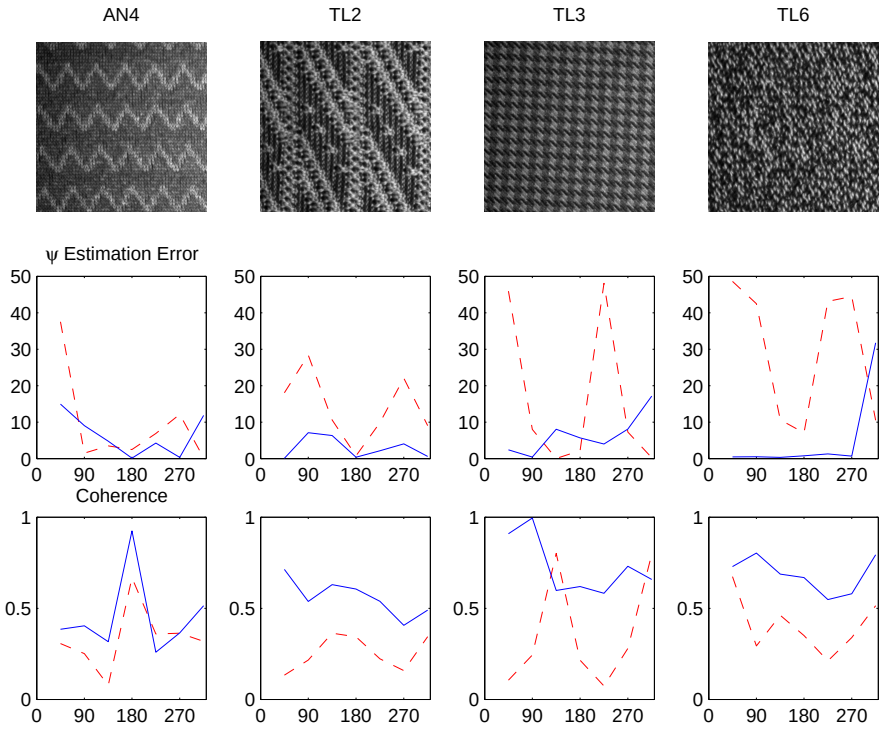


Figure 7.9: Estimating the illuminant's azimuth for samples in the Heriot-Watt TextureLab database. For each material, the image at $\psi = 0°$ is chosen as the reference image. The solid blue curves (middle row) then represent the error in estimating ψ in degrees for all the remaining images using (7.26). The dashed red curves represent the estimation error as measured using (7.14). The bottom row is a plot of the associated coherences. Derivatives are computed at various scales and the best results reported for both methods.

ture survey. It has textures representing various kinds of materials: isotropic, oriented (in both surface and albedo), rough, etc. Each material has been imaged under a fixed viewpoint. The illuminant's elevation is also fixed at $\nu = 45°$ but the azimuth varies between $\psi = 0°$ and $\psi = 315°$.

To test the theory, samples from the database are taken whose surface might be modelled as isotropic and Gaussian but for which the albedo varies considerably. For each sample, the image taken at $\psi = 0°$ is retained as the reference image while (7.26) is then used to recover the azimuthal angle for all the rest. Figure 7.9 is a plot of the estimation error for four samples, AN4, TL2, TL3 and TL6, each of which has signification variation in its albedo. The middle row shows plots of the estimation error versus ψ for the remaining images. The solid blue curves represent the errors in the angle estimated using (7.26) and generally tend to be much lower than the dashed red curve representing the error in estimation due to (7.14). The bottom row is a plot of the associated coherences. Even though (7.27) predicts that the coherences should now equal 0.8 this is clearly not the case. The variation is most probably due to deviations from the model in terms of shadowing.

7.6 Local estimation

Even though the methods developed in sections 7.2 and 7.4 appear to cope fairly well with deviations in the model, there are often cases where a few bad measurements can adversely affect the recovery of the azimuthal angle. It is therefore desirable to estimate the illuminant's direction using local regions rather than the entire image.

As has been noted in section 7.3, the presence of strong edges can bias the structure tensor and therefore these regions should be excluded while

computing the expectations. Similarly, regions of constant intensity where the signal variation is very low should also be excluded.

There exist many operators [Förstner and Gülch, 1987,Harris and Stephens, 1988, Lindeberg and Gårding, 1994] to discard exactly such regions. Most of them are based around computing the second moment matrix which is extremely similar to the structure tensor $\mathbf{S}$. We use the Harris corner detector operator [Harris and Stephens, 1988] to reject edge and constant intensity regions which might deviate from the assumed model and therefore return bad estimates. To determine the statistics locally, the most interesting Harris points are computed and then the regions around those points are used to calculate the expectations $< LI_x^2 >$, $< LI_y^2 >$ and $< LI_{xy} >$. Thus at each chosen Harris point the structure tensor is computed locally to return a local estimate of the illuminant's direction. This can then be used to return the probability distribution of the azimuthal angle from which the mode can be chosen as the most likely estimate.

Preliminary experiments indicate favourable results. As discussed in section 7.3 the azimuth can be estimated to within an accuracy of a few degrees for most images of Plaster A in the CUReT database. This indicates that the texture satisfies the basic model. However, for a few images the estimation error is as high as 15° indicating that viewpoint and shadowing effects are causing deviations from the model and thereby contributing incorrect measurements. It is hoped that if these measurements can be excluded from the estimation process then the azimuthal angle should be recovered more accurately. This is found to be exactly the case when the top 300 Harris points are used to choose the regions for estimating the statistics. Figure 7.10 plots the probability distribution of the angles estimated using the Harris regions. The mode of the distribution is at 65° which is within 0.15° of the ground

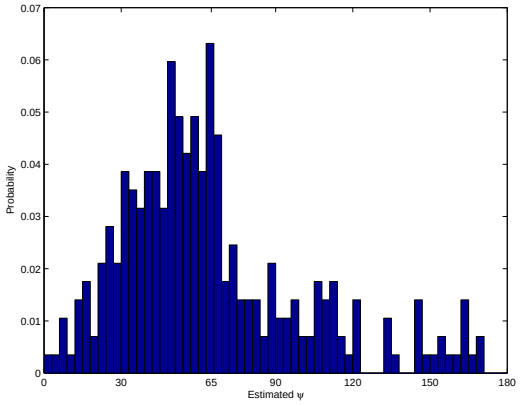


Figure 7.10: Recovering the illuminant's azimuth using local estimates for an image of Plaster A from the CUReT database. The ground truth is $\psi = 65.10°$. The angle recovered using (7.14) which computes statistics over the entire image is $\psi = 49.61°$. By estimating the angle locally using Harris regions and rejecting others it is possible to improve the accuracy of the estimate as the mode of the distribution is 65°.

truth. Had the entire image been used to compute the structure tensor the recovered angle would have been $\psi = 49.61°$ with an error of 15.49°.

7.7 Conclusions

In this chapter, we have developed a theory for estimating the illuminant's azimuth for isotropic, Gaussian random textures with spatially variable albedo. Even though the theory was derived under very strong assumptions it was empirically demonstrated that good results were achieved for over 5000 CUReT images taken under various conditions which deviate from the ideal. In certain cases, the reason for the model's insensitivity to these deviations can be explained by incorporating non Lambertian effects into the albedo map. Thus, the theory appears to robustly handle the effects of shadows, specularities, inter-reflections, etc.

When the albedo itself is isotropic and randomly distributed log-normally,

then the solution for the illuminant's azimuth is identical to the one found by Koenderink and Pont. However, the coherence of the structure tensor is no longer a constant but varies with both the elevation and the azimuth and is dependent on the texture's albedo and shadowing pattern. In the case that extra information is available in the form of a registered image with the same elevation, then it is possible to extend the theory to arbitrarily varying albedo as long as the surface itself is roughly isotropic Gaussian.

Being able to recover the illuminant's azimuth raises the interesting possibility of resolving parts of the Generalized Bas-Relief ambiguity (GBR) [Belhumeur et al., 1999, Yuille et al., 1999]. Unfortunately, it turns out that once integrability has been enforced, the GBR does not affect the azimuthal angle of the light source but only its elevation and strength. However, the fact that a Gaussian distribution has been imposed on the height function does restrict the ambiguity. If the transformed surface is given by

$$\bar{h}(x, y) = \lambda h(x, y) + \mu x + \nu y + d \tag{7.28}$$

then, in theory, both μ and ν must be zero and the ambiguity reduces to λ which affects the variance of the Gaussian, and the constant of integration in the surface reconstruction d which affects the mean. However, in practice, due to numerical reasons and because the Gaussian distribution is approximated by a finite number of surface height points, it may well be the case that the ambiguity is not resolved to just λ and d but may also involve spurious values of μ and ν.

Chapter 8

Conclusions

We began this thesis by listing some of the applications of texture classification followed by a literature survey. Perhaps, it is therefore appropriate to end by looking at some of the spplications that our work has been put to and reviewing extensions of it in the literature. Possible directions of future research are also explored. Finally, we conclude the thesis with an attempt at an operational definition of texture.

8.1 Applications and extensions

The single image classification framework developed in this thesis imposes very few restrictions on the training or novel images and is therefore applicable in many domains. It is also easy to implement. Consequently, researchers have been able to transfer our algorithm to other problems and have also extended and modified our basic framework. Most of this research builds upon the work done in chapter 3.

8.1.1 Breast parenchymal density classification

Breast parenchymal density is thought to be an important indicator of the possibility of developing breast cancer. A patient can be classified as low or high risk depending on the presence of duct patterns in her mammogram. As figure 8.1 shows, these patterns manifest themselves as textural regions rather than as clearly delineated objects. Based on this observation, [Petroudi et al., 2003] used the VZ classifier in conjunction with the MR8 filter bank to first learn the filter response distributions for both types of patients and then use the learnt models to predict the risk to a new patient. Some important domain specific pre-processing steps had to be performed (such as the removal of the pectoral muscle regions from the mammograms) but otherwise the classification framework was left unchanged.

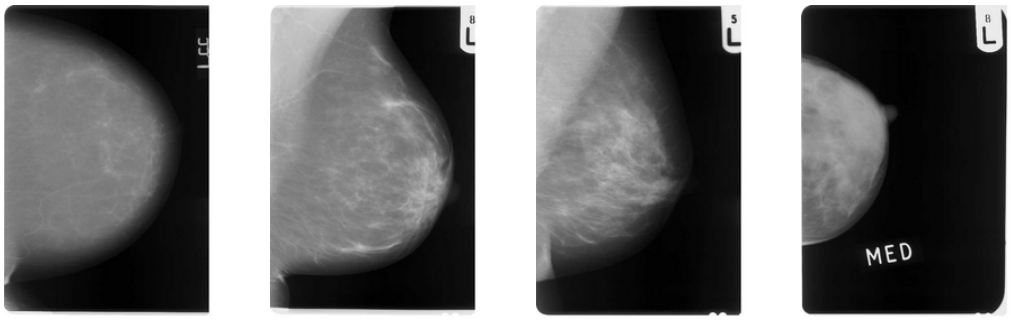


Figure 8.1: Mammograms can be classified as low or high risk on the basis of their parenchymal density [Petroudi et al., 2003]. On the left are mammograms of two low risk patients whose breasts are mainly composed of fatty tissue. On the right are two high risk images with dense duct patterns.

Results were presented for a database of 132 hand labelled mammograms. Using a very small training set, classification rates of over 90% were achieved. This was a substantial improvement over previous work and is a hopeful indicator that early breast cancer diagnosis tools might soon be fully automated in the near future.

8.1.2 Learning better textons

In [Georgescu et al., 2003], the authors investigate the impact of choice of clustering method on the VZ classifier. Throughout this thesis, *K-Means* clustering has been used to learn the texton dictionary. However, the method has some potential drawbacks. In particular, the underlying density is approximated using spherical Gaussians and the number of Gaussians has to be known in advance. Furthermore, the method is not robust as each cluster centre can be heavily influenced by outliers. Georgescu *et al.* claim that almost all these problems are taken care of if the *Mean-Shift* algorithm is used for clustering instead of *K-Means*.

In *Mean-Shift* the underlying density is estimated non-parametrically and the number of textons in the dictionary is determined automatically by selecting the modes of the distribution. Selecting the modes rather than the means also makes the method robust to outliers. This may translate to learning better textons as compared to *K-Means* since the learnt textons are no longer corrupted by noisy or outlier measurements (see figure 8.2).

(a) (b) (c)

Figure 8.2: Results from [Georgescu et al., 2003]. Image (a) shows the D040 texture from the Brodatz album. Images (b) and (c) show the pixels labelled using the top mode and mean textons respectively. The mode textons learnt using *Mean-Shift* appear to capture the local structure much better than the mode textons learnt using *K-Means*.

Results are presented on the Brodatz album using the LM, S, MR4 and MR8 filter banks. The VZ algorithm is used and no changes are made to the classification scheme except that textons are learnt using *Mean-Shift* rather than *K-Means*. The results obtained are very interesting. The performance of the *Mean-Shift* classifier (as compared to *K-Means*) was slightly worse for the MR4 and MR8 filter banks, had no effect on S, and was slightly better for LM. Another interesting result was that if the texton dictionary was chosen randomly, i.e. by selecting points at random in filter response space, then performance dropped only by 1 – 6%. These results match our finding that the performance of the MR8 filter bank dropped by roughly 5% on the CUReT database if the textons were chosen randomly. All these results imply that clustering is not crucial in our framework. While improved clustering might result in slightly better classification, the overall benefits might not be worthwhile – particularly if they have to be obtained at the cost of discarding information by projecting to low dimensions using filters.

One final result which must be mentioned is that the performance of the LM filter bank is found to be superior to MR8. Unfortunately, no analysis of this result is presented. However, it should be noted that the database used contains absolutely no rotation. Images in the Brodatz album are taken and subdivided into non-overlapping regions. Half the regions are retained for training while test results are reported on the other half. As such, there are no rotation variations, which might be the reason, apart from implementation issues, why the LM filters perform better than the MR and S sets.

8.1.3 Improving classification via SVMs

The VZ MR8 framework was extended in [Hayman et al., 2004] by replacing nearest neighbour classification with Support Vector Machines. The paper

has already been mentioned in the literature survey so we just summarise its main points now. A tree structured SVM setup was used to convert the 61 class problem for the CUReT texture to 1830 pairwise classification problems. This improved the classification performance from 97.66% to 98.46%. However, the main attraction of SVMs are that they provide a principled learning approach to the problem of model reduction. Hayman *et al.* found that SVMs reduced the number of models needed by 10 – 20% of the original training set. The paper also presented two other significant results about real world material classification. It experimentally verified that pure learning techniques were incapable of successfully coping with variations caused by scale changes or by the presence of different instances of the same material.

8.1.4 Maximum response over affine transformations

The MR filter sets introduced in chapter 3 operate by taking the maximum response over orientation and scale. The idea was extended in [Caenen and Van Gool, 2004] by taking the maximum response over all affine transformations of a basic filter. The maximum filter response at each pixel was computed using gradient ascent over the parameter space of affine transformations. Note that this is different from the technique used in this thesis of discretizing the parameter space into 6 orientations and 3 scales and then using brute force search to determine the maximum response. The use of steerable kernels would provide a halfway meeting point between the two techniques.

For the classification experiments, a filter bank is chosen by generating 9 filters randomly. Each filter in the bank is chosen to have initial support 3×3 (though the final support at a given pixel can be much larger depending on the scale selected during affine transformation estimation). The standard VZ

framework is then used for classification except that distances are measured using the Bhattacharya metric rather than the χ^2 statistic. Results were given for the first 27 materials present in the CUReT database. Under this setup, it was found that the maximum response classifier had significantly better performance than the Joint classifier (of chapter 6) using 3×3 patches. Some supervised segmentation results are also shown when texture models have been learnt from minimal training data and using a window classification algorithm with no boundary adaptation (see figure 8.3).

Figure 8.3: Supervised segmentation results obtained by [Caenen and Van Gool, 2004] using maximal filter responses. There are five classes: brick, ivy, sky, stone and window. Three small patches of each class are chosen from the left image for training. 20×20 windows in the middle image are then classified into one of these five classes and the results shown on the right.

The idea of computing local affine invariant descriptors using maximal filter responses provides an alternative to the framework of [Lazebnik et al., 2003b]. One of the main advantages of filter responses is that they can be computed at every pixel in the image. This is in contrast to the descriptors of Lazebnik *et al.* which are computed at only certain interest points.

8.2 Future work

Our emphasis in this thesis has been on moving texture classification algorithms out of the lab and into the real world. Developing a framework for

classifying single, uncalibrated images has been a first step in this direction. Another important step has been the use of compact image patches as opposed to filter banks with large support. However, many hurdles still need to be overcome before a full fledged texture classifier can be successfully deployed in the real world. We now discuss two of the most important issues which need to be addressed.

8.2.1 Automatic segmentation and classification

One of the main challenges posed by real world composite images is how to determine which regions of the image should be submitted to a texture classification algorithm for categorization. These regions are typically pre-segmented by hand both during training and classification. For instance, even though the Joint classifier could correctly label nearly 98% of the pixels in the San Francisco images, each region had to be marked by hand before it could be classified. This reliance on human judgement to pre-segment the regions limits the applicability of classification algorithms in the real world.

One way of overcoming this problem is to use an automatic segmentation algorithm to determine texture regions. However, even state of the art unsupervised segmentation algorithms [Galun et al., 2003, Kadir and Brady, 2003, Malik et al., 2001, Tu and Zhu, 2002] do not generate regions good enough for texture classification across a broad spectrum of images.

An alternative is to make use of the training data and learn a model for simultaneously segmenting and classifying regions in composite texture scenes. Two questions need to be resolved in order to successfully implement such an algorithm. The first is about how to deal with background data, i.e. data which does not fall into any of the class categories but which nevertheless occurs frequently in the novel images. Possible solutions are to either learn an

explicit model for the background and clutter or, in case that is not feasible, to implicitly model these nuisance regions as anything that doesn't fit the class models with a given probability or threshold.

The second question is concerned with the design of the segmentation algorithm. One class of segmentation algorithms start with small seed regions which are then grown and merged to come up with a final segmentation of the full image. An attractive way of doing this is to start by over-segmenting the image into superpixels [Ren and Malik, 2003]. The final segmentation is then obtained using a series of moves such as migration of superpixels from one region to another as well as the merging and splitting of two regions. The moves are chosen so as to optimise some cost function – generally related to the probability of the overall segmentation as determined by the fit of the individual regions to the models learnt during training. Note that in a supervised segmentation scheme it is possible to verify that a cost function has been chosen appropriately. In particular, the ground truth segmentations should have the highest probability or globally optimum cost. If any other segmentation of the image results in a higher probability then it is immediately clear that the framework in which the solution is being attempted is incorrect.

While it is possible to get some good results using such "growth" algorithms (see figure 8.4), they have a serious drawback which can lead to poor results for many real world images. It is frequently the case that very small regions do not contain enough data for representative statistics (such as texton distributions) to be gathered and classified accurately. Hence, these regions invariably get mapped to the wrong models. Since growth algorithms start with lots of very small regions they often end up in a local optimum which is of a noticeably poorer quality than the desired ground truth segmentation.

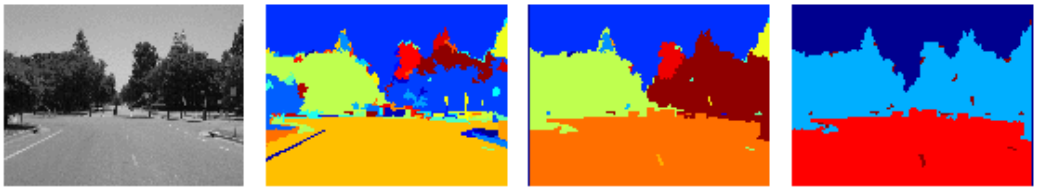


Image Initial Seg. Final Seg. Classification

Figure 8.4: Automatic segmentation and classification results on an image from the San Francisco database. Texture models were first learnt for all the six classes using 5×5 neighbourhoods (larger neighbourhoods resulted in boundary artifacts). Next, an initial segmentations was obtained using [Felzenszwalb and Huttenlocher, 2004]. The initial segments were then evolved using region merge and segment transition moves. A greedy algorithm was used to pick the move which resulted in the largest probability of the subsequent segmentation as determined by the χ^2 fit to the learnt models. The segmentation was stopped when no move resulted in an increased probability. The final segmented regions were then classified. Note that in this case good results are achieved because most of the initial segments are fairly large themselves.

Consequently, the final classification results can also be poor.

The problem of small regions can be tackled using a segmentation algorithm which splits under-segmented regions rather than merging over-segmented ones. First, a generative model is trained to predict how many classes are present in an under-segmented region. This is achieved by matching the region's texton distribution against linear combinations of the learnt texture models. The coefficients of the combination which best matches the target region determine the percentages of each class present. Given this generative model, the automatic classification process starts by building a hierarchical segmentation of the novel image. The top level of the segmentation has the entire image as a single region while at the bottom level each pixel is a separate region by itself. An iterative scheme is applied, where at each iteration, a region is selected to see if it has more than one class present. If it doesn't, then it is retained for the final segmentation otherwise it is split

into its two subregions. The iterations continue until a final segmentation is arrived at where each region contains pixels from only a single class. The final segments are then classified.

An initial implementation of this algorithm has shown promising results. For the generative model, we chose to use the L_2 norm to measure the similarity between two distribution rather than the χ^2 statistic. The L_2 norm was chosen as it leads to an analytical solution for the optimum combination of models which best match a region's texton distribution. To obtain a hierarchical segmentation, the image is subdivided into progressively finer grids. Using this setup, it was determined that for the larger regions, the generative model was very good at correctly picking out not just which classes comprised a region but also their relative percentages. However, the use of a grid based segmentation meant that towards the bottom levels the regions still comprised more that one class but were now too small to be classified correctly. We propose to address this problem by extending the scheme of [Felzenszwalb and Huttenlocher, 2004] to obtain a hierarchical segmentation based on region boundaries. Going down the segmentation hierarchy would then ensure that we do not have regions which are mixed but too small to be classified correctly.

8.2.2 Incorporating physical information for model reduction

In most real world situations, only a small amount of training data is ever available from which to learn texture models. Therefore, the problem of how to characterise textures using only a few models is key to moving out of the lab. It has already been determined that pure learning techniques are incapable of dealing with the model reduction problem.

Instead, we propose to explore solutions which incorporate physical information about textures. In particular, we propose to learn a universal dictionary of *surfons* – basic, small scale surface primitives such as bumps, ridges, grooves, etc. whose repeated placement constitutes any surface. The goal is to model a texture by the distribution of its surfons as this is invariant to changes in imaging conditions. Hence, a single surfon distribution learnt for a particular texture could be adequate for coping with imaging variations. The main challenge lies in estimating the surfon distribution from images. If multiple, registered novel images are present with known viewpoint and illumination then the problem is relatively easy. Either photometric stereo techniques can be used to recover surface detail and thereby compute the surfon distribution or the surfons can be rendered for the specific imaging conditions to determine which surfons match which image patches in the texture.

Some investigations into such methods have recently been carried out by [Wang and Dana, 2004]. However, the need for registered images makes the algorithm of Wang and Dana less, rather than more, applicable in the real world. Therefore, an interesting area of future research would be to study whether the problem can still be solved using unregistered novel images or, ideally, using single images alone. Developing a theory of *statistical photometric stereo* could lead to a possible solution. Working with unregistered images, the goal would be recover statistical, rather than exact, descriptions of the surface and albedo which are nevertheless sufficient for classification.

8.3 Conclusions

In this thesis, we have studied the problem of texture classification of single images without requiring any *a priori* information about their imaging

conditions. First, a basic framework was set up in which the problem of single image classification could be attempted and we introduced low dimensional, rotation and scale invariant filter banks which were neverthless capable of accurate discrimination. Then, parallels were drawn between the psychophyscially motivated classification scheme of Leung and Malik and the statistical approach of Konishi and Yuille and it was shown how the two could be made equivalent. Next, we questioned the predominant use of filter responses as texture features and offered an alternative representation based on image patches. This was demonstrated to be superior for tasks such as classification and synthesis. Finally, a theory for recovering the illuminant's azimuthal angle was developed so as to overcome the lack of prior knowledge and aid in future classification.

It might be worthwhile to reflect on two interesting points before finishing. First, do textons really form isolated clusters in the spirit of Leung and Malik and Julesz's definitions or do they just provide a way of vector quantising the feature space? Second, are we any closer to providing a definition of texture than we were at the start of the thesis?

There are two ways of answering the first question – and both lead to the conclusion that texture features are not uniformly distributed throughout the space. The most direct way of determining this is to actually look at all possible 2D projections of image patches (or filter responses). In each case, it was found that the distributions were not uniform. For example, the scatter plots in figure 6.9 clearly show that the Limestone distribution forms a tight cluster. Furthermore, if we were to inspect the PDF on top of the scatter plot, it turns out that even the Ribbed Paper distribution forms tight clusters. Another way of determining that the texton representation adequately captures the true texture distribution is by inspecting the synthesis and clas-

sification results. As figure 6.5 illustrates, a few hundred textons can be used to synthesise high quality textures which are perceptually indistinguishable from the originals. Therefore, it must be the case that the true PDF is not distributed throughout the space as otherwise important information would be lost and the synthesis would be poor. This view is backed up by the classification results of subsection 5.3.1 where it was determined that quantising the PDF into uniformly spaced bins did not improve classification and where most of the bins turned out to be empty.

To answer the second question, we focus on an operational definition of texture from the perspective of classification and synthesis rather than attempt a definition which might be universal but non-functional. In the recent past, textures have come to be characterised by the distribution of filter responses gathered across the entire image. Since most filter designs were biologically motivated, a texture was effectively defined by its global distribution of edges, bars, spots and rings. While this is a valid definition and has worked well for a long time, this thesis has demonstrated that it is not the best possible one. Instead, we prefer to revive the older definitions of texture based on image patches.

We define the texture of an image *patch* as the co-occurrence of pixel intensities in that patch. Note that the emphasis has shifted from texture being a global statistical property of the image to it being a local property of every image patch (though global context is still very important while measuring the similarity between two patches). Thus, even a uniform planar patch with no albedo markings has some texture.

It is possible to draw an analogy between pixel intensities and texture patches using this definition. The intensity at a particular pixel becomes a limiting case of its texture as its neighbourhood is made vanishingly small.

Thus, operators on pixel intensities can have analogous counterparts for texture patches. For instance, using the given definition, texture gradients can be computed just as intensity gradients. Similarly, the degree of texture homogeneity can be measured analogously to intensity homogeneity. Most importantly, the two questions "What is the global texture / colour of this image?" have the same general answer. For certain applications, such as content based image retrieval and classification, a valid response might be the colour distribution or the patch distribution. However, in a more general context both questions may well be nonsensical.

It should be noted that the relationship between texture and intensity cannot be taken too far. In particular, similar image patches do not need to have similar central pixel intensities and similar central pixel intensities do not imply similar texture patches.

While a local patch based definition has proved to be very useful for both classification and synthesis, we have so far brushed under the carpet the crucial question of how to determine the size and shape of a patch for either application. The algorithm of [Zalesny and Van Gool, 2000] presents one way of learning the neighbourhood structure using the synthesis-via-analysis route. However, they assume that textures are stationary (i.e. the same structure is applicable at every pixel in the image). Clearly, the shape and size of the neighbourhood structure must vary with the local geometry and photometry of the texture. As such, affine region adaptation methods such as [Caenen and Van Gool, 2004, Lazebnik et al., 2003b, Ravela, 2004] might provide a better way of determining the shape and scale of patches at different locations in an image.

In conclusion, we put in perspective our efforts at defining texture with a quote from Mark Twain – "The researches of many commentators have

already thrown much darkness on this subject, and it is probable that, if they continue, we shall soon know nothing at all about it".

Appendix A

Relating χ^2 to the Capacitory Discriminant & KL Divergence

The capacitory discriminant [Topsoe, 2000] between two discrete probability distributions p and q is defined as

$$\begin{aligned} C(p,q) &= \sum_k p_k \log\left(\frac{2p_k}{p_k+q_k}\right) + q_k \log\left(\frac{2q_k}{p_k+q_k}\right) && \text{(A.1)} \\ &= D\left(p\|\tfrac{1}{2}(p+q)\right) + D\left(q\|\tfrac{1}{2}(p+q)\right) && \text{(A.2)} \\ &= D(p\|q) - 2D\left(\tfrac{1}{2}(p+q)\|q\right) && \text{(A.3)} \end{aligned}$$

where $D = \sum p_k \log(p_k/q_k)$ is the KL divergence. The capacitory discriminant is sometimes also referred to [Rubner et al., 2000] as Jeffreys' divergence (even though C differs slightly from the original definition [Kullback, 1968]) and is generally preferred to the KL divergence as it is symmetric, more robust and $\sqrt{C(p,q)}$ is a metric [Endres and Schindelin, 2003].

To show that $\lim_{p\to q} C = \frac{1}{2}\chi^2$ we rewrite C as

$$C(p,q) = \sum_k p_k \log\left(1 + \frac{p_k - q_k}{p_k + q_k}\right) + q_k \log\left(1 - \frac{p_k - q_k}{p_k + q_k}\right) \tag{A.4}$$

and take the Taylor series expansion $\log(1+x) = x - \frac{1}{2}x^2 + \mathcal{O}(x^3)$ to get

$$\begin{aligned} C(p,q) &= \sum_k (p_k - q_k)\frac{p_k - q_k}{p_k + q_k} - \tfrac{1}{2}(p_k + q_k)\left(\frac{p_k - q_k}{p_k + q_k}\right)^2 + \epsilon \\ &= \tfrac{1}{2}\sum_k \frac{(p_k - q_k)^2}{p_k + q_k} + \epsilon \qquad (A.5) \\ &= \tfrac{1}{2}\chi^2(p,q) + \epsilon \qquad (A.6) \\ \Rightarrow \lim_{p\to q} C(p,q) &= \tfrac{1}{2}\chi^2(p,q) \qquad (A.7) \end{aligned}$$

Thus, the χ^2 statistic is a limiting case of the capacitory discriminant and the two should give the same results when the distributions being compared are similar. Furthermore, since $\epsilon \geq 0$ we must have $\frac{1}{2}\chi^2(p,q) \leq C(p,q)$. In fact, Topsoe has shown

$$\tfrac{1}{2}\chi^2(p,q) \leq C(p,q) \leq \ln 2 \cdot \chi^2(p,q) \tag{A.8}$$

which provides very tight bounds on C in terms of the χ^2 statistic.

A somewhat analogous result can also be shown for the KL divergence. By noting that $D \geq 0$ and combining (A.3) and (A.6) we get $\frac{1}{2}\chi^2(p,q) \leq D(p\|q)$. Thus χ^2 also provides a lower bound for the KL divergence. The upper bound for D can be derived in terms of the asymmetric chi-squared statistic

$\chi^2_a(p,q) = \sum_k (p_k - q_k)^2/q_k$ by noting that

$$\begin{aligned} D(p\|q) &= \sum_k p_k \log\left(\frac{p_k}{q_k}\right) = \sum_k p_k \log\left(1 + \frac{p_k - q_k}{q_k}\right) && \text{(A.9)} \\ &= \sum_k p_k \frac{p_k - q_k}{q_k} - \epsilon + \sum_k q_k - \sum_k p_k && \text{(A.10)} \\ &= \sum_k \frac{(p_k - q_k)^2}{q_k} - \epsilon && \text{(A.11)} \\ \Rightarrow \lim_{p\to q} D(p\|q) &= \chi^2_a(p,q) && \text{(A.12)} \end{aligned}$$

where we have used the truncated Taylor series $\log(1+x) = x - \mathcal{O}(x^2)$ and the fact that $\sum q_k - p_k = 0$. Noting that $\epsilon \geq 0$ in (A.11) gives

$$\tfrac{1}{2}\chi^2(p,q) \leq D(p\|q) \leq \chi^2_a(p,q) \qquad \text{(A.13)}$$

which are the required bounds on the KL divergence in terms of the chi-squared statistics.

Bibliography

[Ade, 1983] Ade, F. (1983). Characterization of texture by eigenfilters. *Signal Processing*, 5:451–457.

[Ashikhmin, 2001] Ashikhmin, M. (2001). Synthesizing natural textures. In *ACM Symposium on Interactive 3D Graphics*, pages 217–226.

[Ashikhmin et al., 2000] Ashikhmin, M., Premoze, S., and Shirley, P. (2000). A microfacet based brdf generator. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 65–74, New Orleans, Louisiana.

[Bach and Jordan, 2002] Bach, F. R. and Jordan, M. I. (2002). Tree-dependent component analysis. In *Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 36–44, Edmonton, Canada.

[Barber and LeDrew, 1991] Barber, D. G. and LeDrew, E. F. (1991). SAR sea ice discrimination using texture statistics: A multivariate approach. *Photogrammetric Engineering and Remote Sensing*, 57(4):385–395.

[Beck, 1983] Beck, J. (1983). Textural segmentation, second-order statistics, and textural elements. *Biological Cybernetics*, 48(2):125–130.

[Beck et al., 1987] Beck, J., Sutter, A., and Ivry, R. (1987). Spatial frequency channels and perceptual grouping in texture segregation. *Computer Vision, Graphics and Image Processing*, 37(2):299–325.

[Belhumeur et al., 1999] Belhumeur, P., Kriegman, D., and Yuille, A. L. (1999). The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44.

[Benelli and Garzelli, 1999] Benelli, G. and Garzelli, A. (1999). Oil-spills detection in SAR images by fractal dimension estimation. In *IEEE Geoscience and Remote Sensing Symposium*, volume 1, pages 218–220, Hamburg, Germany.

[Bergen, 1991] Bergen, J. R. (1991). Theories of visual texture perception. In Regan, D., editor, *Spatial Vision*, volume 10, pages 114–134. Macmillan, New York.

[Bergen and Adelson, 1988] Bergen, J. R. and Adelson, E. H. (1988). Early vision and texture perception. *Nature*, 333:363–364.

[Bergen and Landy, 1991] Bergen, J. R. and Landy, M. S. (1991). Computational modeling of visual texture segregation. In Landy, M. S. and Movshon, J. A., editors, *Computational Models of Visual Perception*, pages 253–271. MIT Press, Cambridge MA.

[Berry and Hannay, 1977] Berry, M. V. and Hannay, J. H. (1977). Umbilic points on gaussian random surfaces. *Journal of Physics A: Mathematical and General*, 10(11):1809–1821.

[Blake and Marinos, 1990] Blake, A. and Marinos, C. (1990). Shape from texture: Estimation, isotropy and moments. *Artifical Intelligence*, 45(3):323–380.

[Bodnarova et al., 2000] Bodnarova, A., Bennamoun, M., and Latham, S. J. (2000). Textile flaw detection using optimal gabor filters. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 799–802, Barcelona, Spain.

[Bovik et al., 1990] Bovik, A. C., Clark, M., and Geisler, W. S. (1990). Multichannel texture analysis using localised spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:55–73.

[Bradley et al., 1993] Bradley, J. N., Brislawn, C. M., and Hopper, T. (1993). The FBI wavelet/scalar quantization standard for grayscale fingerprint image compression. In *Proceedings of the SPIE Conference on Visual Information Processing II*, pages 293–304, Orlando, Florida.

[Brady, 2003] Brady, K. (2003). *A Probabilistic Framework for Adaptive Texture Description*. PhD thesis, University of Nice-Sophia Antipolis.

[Brodatz, 1966] Brodatz, P. (1966). *Textures: A Photographic Album for Artists & Designers*. Dover, New York.

[Brooks and Horn, 1985] Brooks, M. J. and Horn, B. K. P. (1985). Shape and source from shading. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 932–936.

[Burges, 1999] Burges, C. J. C. (1999). Geometry and invariance in kernel based methods. In Scholkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods - Support Vector Learning*, pages 89–116. MIT Press, Cambridge, Massachusetts.

[Caelli and Julesz, 1978] Caelli, T. and Julesz, B. (1978). On perceptual analyzers underlying visual texture discrimination. part i. *Biological Cybernetics*, 25:167–175.

[Caelli and Moraglia, 1985] Caelli, T. and Moraglia, G. (1985). On the detection of gabor signals and discrimination of gabor textures. *Vision Research*, 25(5):671–684.

[Caenen and Van Gool, 2004] Caenen, G. and Van Gool, L. (2004). Maximum response filters for texture analysis. In *Proceedings of the IEEE Workshop on Perceptual Organization in Computer Vision*, Washington, DC.

[Chai et al., 1999] Chai, B. B., Vass, J., and Zhuang, X. (1999). Significance-linked connected component analysis for wavelet image coding. *IEEE Transactions on Image Processing*, 8(6):774–784.

[Chang, 1974] Chang, C. L. (1974). Finding prototypes for nearest neighbour classifiers. *IEEE Transactions on Computers*, 23(11):1179–1184.

[Chang and Kuo, 1993] Chang, T. and Kuo, C. C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2(4):429–441.

[Chantler, 1994] Chantler, M. J. (1994). *The effect of variation in illuminant direction on texture classification*. PhD thesis, Heriot-Watt University.

[Chantler et al., 2002a] Chantler, M. J., McGunnigle, G., Penirschke, A., and Petrou, M. (2002a). Estimating lighting direction and classifying textures. In *Proceedings of the British Machine Vision Conference*, pages 737–746, Cardiff, UK.

[Chantler et al., 2002b] Chantler, M. J., Schmidt, M., Petrou, M., and McGunnigle, G. (2002b). The effect of illuminant rotation on texture filters: Lissajous's ellipses. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 289–303, Copenhagen, Denmark.

[Chapelle and Scholkopf, 2002] Chapelle, O. and Scholkopf, B. (2002). Incorporating invariances in nonlinear Support Vector Machines. In *Advances in Neural Information Processing Systems*, pages 609–616.

[Chellappa and Chatterjee, 1985] Chellappa, R. and Chatterjee, S. (1985). Classification of textures using gaussian markov random fields. *IEEE Transactions on Acoustics. Speech and Signal Processing*, 33(4):959–963.

[Chellappa et al., 1985] Chellappa, R., Chatterjee, S., and Bagdazian, R. (1985). Texture synthesis and compression using gaussian-markov random field models. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(2):298–303.

[Chen et al., 1989] Chen, C. C., Daponte, J., and Fox, M. (1989). Fractal feature analysis and classification in medical imaging. *IEEE Transactions on Medical Imaging*, 8:133–142.

[Chetverikov, 2000] Chetverikov, D. (2000). Pattern regularity as a visual key. *Image and Vision Computing*, 18:975–985.

[Chetverikov and Hanbury, 2002] Chetverikov, D. and Hanbury, A. (2002). Finding defects in texture using regularity and local orientation. *Pattern Recognition*, 35:203–218.

[Clerc and Mallat, 2002] Clerc, M. and Mallat, S. (2002). The texture gradient equation for recovering shape from texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):536–549.

[Coggins and Jain, 1985] Coggins, J. M. and Jain, A. K. (1985). A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, 3:195–203.

[Conners et al., 1990] Conners, R. W., McMillin, C. W., Lin, K., and Vasquez-Espinosa, R. E. (1990). Identifying and location surface defects in wood: Part of an automated lumber processing system. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 726–728.

[Criminisi et al., 2004] Criminisi, A., Perez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212.

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.

[Cross and Jain, 1983] Cross, G. K. and Jain, A. K. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39.

[Cuenca and Camara, 2003] Cuenca, S. A. and Camara, A. (2003). New texture descriptor for high-speed web inspection applications. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 537–540.

[Cula and Dana, 2001] Cula, O. G. and Dana, K. J. (2001). Compact representation of bidirectional texture functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1041–1047, Kauai, Hawaii.

[Cula and Dana, 2004] Cula, O. G. and Dana, K. J. (2004). 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60.

[Cumming et al., 1993] Cumming, B. G., Johnston, E. B., and Parker, A. J. (1993). Effects of different texture cues on curved surfaces viewed stereoscopically. *Vision Research*, 33:827–838.

[Dana and Nayar, 1998] Dana, K. J. and Nayar, S. (1998). Histogram model for 3d textures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 618–624, Santa Barbara, California.

[Dana and Nayar, 1999] Dana, K. J. and Nayar, S. (1999). Correlation model for 3D texture. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1061–1067, Corfu, Greece.

[Dana et al., 1997] Dana, K. J., van Ginneken, B., Nayar, S. K., and Koenderink, J. J. (1997). Reflectance and texture of real world surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–157, San Juan, Puerto Rico.

[Dana et al., 1999] Dana, K. J., van Ginneken, B., Nayar, S. K., and Koenderink, J. J. (1999). Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34.

[Dasarathy et al., 2000] Dasarathy, B. V., Sanchez, J. S., and Townsend, S. (2000). Nearest neighbour editing and condensing tools - synergy exploitation. *Pattern Analysis and Applications*, 3(1):19–30.

[Daugman, 1985] Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(3):1160–1169.

[De Bonet, 1997] De Bonet, J. S. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 361–368, Los Angeles, California.

[De Bonet and Viola, 1998] De Bonet, J. S. and Viola, P. (1998). Texture recognition using a non-parametric multi-scale statistical model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 641–647, Santa Barbara, California.

[Deng and Clausi, 2003] Deng, H. and Clausi, D. A. (2003). Advanced gaussian mrf rotation-invariant texture features for classification of remote sensing imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 685–690, Madison, Wisconsin.

[Derin and Elliot, 1987] Derin, H. and Elliot, H. (1987). Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55.

[Devi and Murty, 2002] Devi, V. S. and Murty, M. N. (2002). An incremental prototype set building technique. *Pattern Recognition*, 35(2):505–513.

[Do and Vetterli, 2002] Do, M. N. and Vetterli, M. (2002). Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden markov models. *IEEE Transactions on Multimedia*, 4(4):517–527.

[Domingos and Pazzani, 1997] Domingos, P. and Pazzani, M. J. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

[Drori et al., 2003] Drori, I., Cohen-Or, D., and Yeshurun, H. (2003). Example-based style synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 143–150, Madison, Wisconsin.

[Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley and Sons, second edition.

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

[Efros and Freeman, 2001] Efros, A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 341–346, Los Angeles, California.

[Efros and Leung, 1999] Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1039–1046, Corfu, Greece.

[Endres and Schindelin, 2003] Endres, D. M. and Schindelin, J. E. (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1857–1860.

[Evans et al., 2000] Evans, M., Hastings, N., and Peacock, B. (2000). *Statistical Distributions*. Wiley-Interscience, third edition.

[Faugeras, 1978] Faugeras, O. D. (1978). Texture analysis and classification using a human visual model. In *Proceedings of the International Conference on Pattern Recognition*, pages 549–552, Kyoto, Japan.

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[Fogel and Sagi, 1989] Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, 61:102–113.

[Foley et al., 1990] Foley, J. D., Van Dam, A., Feiner, S. K., and Hughes, J. F. (1990). *Computer Graphics: Principles and Practice*. Addison-Wesley.

[Förstner and Gülch, 1987] Förstner, W. and Gülch, E. (1987). A fast operator for detection and precise location of distinct points, corners and center of circular features. In *Proceedings of the ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland.

[Forsyth, 2002] Forsyth, D. A. (2002). Shape from texture without boundaries. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 225–239, Copenhagen, Denmark.

[Forsyth and Zisserman, 1991] Forsyth, D. A. and Zisserman, A. (1991). Reflections on shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):671–679.

[Fowlkes et al., 2003] Fowlkes, C., Martin, D., and Malik, J. (2003). Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 54–61, Madison, Wisconsin.

[Freeman, 1992] Freeman, W. T. (1992). *Steerable Filters and The Local Analysis of Image Structure*. PhD thesis, MIT.

[Froment and Mallat, 1992] Froment, J. and Mallat, S. G. (1992). *Second Generation Compact Image Coding With Wavelets*. Froment and Mallat.

[Funt et al., 1998] Funt, B., Barnard, K., and Martin, L. (1998). Is machine colour constancy good enough? In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 445–459, Freiburg, Germany.

[Galun et al., 2003] Galun, M., Sharon, E., Basri, R., and Brandt, A. (2003). Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 716–723, Nice, France.

[Garber, 1981] Garber, D. D. (1981). *Computational Models for Texture Analysis and Texture Synthesis*. PhD thesis, University of Southern California.

[Gates, 1972] Gates, G. W. (1972). The reduced nearest neighbour rule. *IEEE Transactions on Information Theory*, 18(3):431–433.

[Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

[Georgescu et al., 2003] Georgescu, B., Shimshoni, I., and Meer, P. (2003). Mean shift based clustering in high dimensions: A texture classification example. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 456–463, Nice, France.

[Gibson, 1950] Gibson, J. J. (1950). The perception of visual surfaces. *American Journal of Psychology*, 63:367–384.

[Gimel'farb et al., 2004] Gimel'farb, G., Van Gool, L., and Zalesny, A. (2004). To FRAME or not to FRAME in probabilistic texture modelling? In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 707–711, Cambridge, UK.

[Gotlieb and Kreyszig, 1990] Gotlieb, C. C. and Kreyszig, H. E. (1990). Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics and Image Processing*, 51(1):76–80.

[Greenspan et al., 1994] Greenspan, H., Belongie, S., Perona, P., and Goodman, R. (1994). Rotation invariant texture recognition using a steerable pyramid. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 162–167, Jerusalem, Israel.

[Gurnsey and Browse, 1987] Gurnsey, R. and Browse, R. A. (1987). Micropattern properties and presentation conditions influencing texture discrimination. *Perception and Psychophysics*, 41(3):239–252.

[Haley and Manjunath, 1995] Haley, G. M. and Manjunath, B. S. (1995). Rotation-invariant texture classification using modified gabor filters. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 262–265, Washington, DC.

[Haralick et al., 1973] Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621.

[Harms et al., 1986] Harms, H., Gunzer, U., and Aus, H. M. (1986). Combined local color and texture analysis of stained cells. *Computer Vision, Graphics and Image Processing*, 33:364–376.

[Harris and Stephens, 1988] Harris, C. J. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147–151, Manchester, UK.

[Hart, 1968] Hart, P. E. (1968). The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*, 14(3):515–516.

[Hayman et al., 2004] Hayman, E., Caputo, B., Fritz, M., and Eklundh, J.-O. (2004). On the significance of real-world conditions for material classification. In *Proceedings of the European Conference on Computer Vision*, volume 4, pages 253–266, Prague, Czech Republic.

[He et al., 1991] He, X. D., Torrance, K. E., Sillion, F. X., and Greenberg, D. P. (1991). A comprehensive physical model for light reflection. *Computer Graphics*, 25(4):175–186.

[Heeger and Bergen, 1995] Heeger, D. J. and Bergen, J. R. (1995). Pyramid-Based texture analysis/synthesis. In *Proceedings of the ACM SIGGRAPH*

Conference on Computer Graphics, pages 229–238, Los Angeles, California.

[Hertzmann et al., 2001] Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., and Salesin, D. (2001). Image analogies. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 327–340.

[Jain and Bhattacharjee, 1992] Jain, A. K. and Bhattacharjee, S. (1992). Text segmentation using gabor filters for automatic document processing. *Machine Vision and Applications*, 5(3):169–184.

[Jain and Farrokhnia, 1991] Jain, A. K. and Farrokhnia, F. (1991). Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186.

[Jain et al., 1990] Jain, A. K., Farrokhnia, F., and Alman, D. H. (1990). Texture analysis of automotive finishes. In *SME Machine Vision Applications Conference*, pages 1–16.

[Jain and Karu, 1996] Jain, A. K. and Karu, K. (1996). Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):195–205.

[James et al., 2001] James, D., Clymer, B. D., and Schmalbrock, P. (2001). Texture detection of simulated microcalcification susceptibility effects in magnetic resonance imaging of breasts. *Journal of Magnetic Resonance Imaging*, 13(6):876–881.

[Julesz, 1962] Julesz, B. (1962). Visual pattern discrimination. *IRE Transactions on Information Theory*, 8:84–92.

[Julesz, 1981] Julesz, B. (1981). Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97.

[Julesz and Bergen, 1983] Julesz, B. and Bergen, J. R. (1983). Textons, the fundamental elements in preattentive vision and perception of textures. *Bell Systems Technical Journal*, 62(6):1619–1645.

[Julesz et al., 1973] Julesz, B., Gilbert, E. N., Shepp, L. A., and Frisch, H. L. (1973). Inability of humans to discriminate between visual textures that agree in second-order statistics – revisited. *Perception*, 2(4):391–405.

[Kadir and Brady, 2003] Kadir, T. and Brady, M. (2003). Unsupervised non-parametric region segmentation using level sets. In *Proceedings of the*

International Conference on Computer Vision, volume 2, pages 1267–1274, Nice, France.

[Kashyap and Khotanzad, 1986] Kashyap, R. L. and Khotanzad, A. (1986). A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):472–481.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons.

[Keller et al., 1989] Keller, J. M., Chen, S., and Crownover, R. M. (1989). Texture description and segmentation through fractal geometry. *Computer Vision, Graphics and Image Processing*, 45:150–166.

[Kim et al., 2002] Kim, K. I., Jung, K., Park, S. H., and Kim, H. J. (2002). Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1542–1550.

[Knill, 2001] Knill, D. C. (2001). Contour into texture: Information content of surface contours and texture flow. *Journal of the Optical Society of America*, 18:12–35.

[Kocher and Kunt, 1983] Kocher, M. and Kunt, M. (1983). Image data compression by contour texture modeling. In *Proceedings of the SPIE Conference on Applications of Digital Image Processing*, pages 131–139, Geneva, Switzerland.

[Koenderink and Pont, 2003] Koenderink, J. J. and Pont, S. C. (2003). Irradiation direction from texture. *Journal of the Optical Society of America*, 20(10):1875–1882.

[Koenderink and Van Doorn, 1983] Koenderink, J. J. and Van Doorn, A. J. (1983). Geometrical modes as a general method to treat diffuse interreflections in radiometry. *Journal of the Optical Society of America*, 73:843–850.

[Koenderink and van Doorn, 2002] Koenderink, J. J. and van Doorn, A. J. (2002). Image processing done right. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 158–172, Copenhagen, Denmark.

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.

[Konishi and Yuille, 2000] Konishi, S. and Yuille, A. L. (2000). Statistical cues for domain specific image segmentation with performance analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 125–132, Hilton Head, South Carolina.

[Krishnamachari and Chellappa, 1997] Krishnamachari, S. and Chellappa, R. (1997). Multiresolution gauss-markov random field models for texture segmentation. *IEEE Transactions on Image Processing*, 6(2):251–267.

[Kullback, 1968] Kullback, S. (1968). *Information Theory and Statistics*. Dover, New York.

[Kuntz et al., 1999] Kuntz, S., Siegert, F., and Rucker, G. (1999). ERS SAR images for tropical rainforest and land use monitoring: change detection over five years and comparison with RADARSAT and JERS SAR images. In *IEEE Geoscience and Remote Sensing Symposium*, volume 2, pages 910–912, Hamburg, Germany.

[Laine and Fan, 1993] Laine, A. and Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1186–1191.

[Laws, 1980] Laws, K. I. (1980). *Textured image segmentation*. PhD thesis, University of Southern California.

[Lazebnik et al., 2003a] Lazebnik, S., Schmid, C., and Ponce, J. (2003a). Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 649–655, Nice, France.

[Lazebnik et al., 2003b] Lazebnik, S., Schmid, C., and Ponce, J. (2003b). A sparse texture representation using affine-invariant regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 319–324, Madison, Wisconsin.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lee and Rosenfeld, 1985] Lee, C. H. and Rosenfeld, A. (1985). Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26:125–143.

[Lee et al., 1992] Lee, T. S., Mumford, D., and Yuille, A. L. (1992). Texture segmentation by minimizing vector-valued energy functionals: The coupled-membrane model. In *Proceedings of the European Conference on Computer Vision*, pages 165–173.

[Lee and Mangasarian, 2001] Lee, Y.-J. and Mangasarian, O. L. (2001). RSVM: Reduced support vector machines. In *First SIAM International Conference on Data Mining*, Chicago, Illinois.

[Leung and Malik, 1999] Leung, T. and Malik, J. (1999). Recognizing surfaces using three-dimensional textons. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1010–1017, Kerkyra, Greece.

[Leung and Malik, 2001] Leung, T. and Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44.

[Levina, 2002] Levina, E. (2002). *Statistical Issues in Texture Analysis*. PhD thesis, University of California at Berkeley.

[Li and Zaidi, 2001] Li, A. and Zaidi, Q. (2001). Information limitations in perception of shape from texture. *Vision Research*, 41(22):2927–2942.

[Li et al., 1995] Li, J., Cheng, P., and Kuo, C. C. J. (1995). An embedded wavelet packet transform technique for texture compression. In *Proceedings of the SPIE Conference on Wavelet Applications in Signal and Image Processing II*, pages 602–613, San Diego, California.

[Li, 2001] Li, S. Z. (2001). *Markov Random Field Modeling in Image Analysis*. Springer-Verlag.

[Li et al., 2000] Li, W., Zhang, Y. Q., Sodagar, I., Liang, J., and Li, S. (2000). MPEG-4 texture coding. In Puri, A. and Chen, T., editors, *Multimedia Systems, Standards, and Networks*. Marcel Dekker, New York, third edition.

[Lindeberg, 1998] Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116.

[Lindeberg and Gårding, 1993] Lindeberg, T. and Gårding, J. (1993). Shape from texture from a multi-scale perspective. In *Proceedings of the International Conference on Computer Vision*, pages 683–691.

[Lindeberg and Gårding, 1994] Lindeberg, T. and Gårding, J. (1994). Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *Proceedings of the European Conference on Computer Vision*, pages 389–400, Stockholm, Sweden.

[Lobay and Forsyth, 2004] Lobay, A. and Forsyth, D. A. (2004). Recovering shape and irradiance maps from rich dense texton fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 400–406, Washington, DC.

[Longuet-Higgins, 1957] Longuet-Higgins, M. S. (1957). The statistical analysis of a random, moving surface. *Philosophical Transactions of the Royal Society of London, Series A*, 249(966):321–387.

[Lorette et al., 2000] Lorette, A., Descombes, X., and Zerubia, J. (2000). Texture analysis through a markovian modelling and fuzzy classification: Application to urban area extraction from satellite images. *International Journal of Computer Vision*, 36(3):221–236.

[Mahalanobis and Singh, 1994] Mahalanobis, A. and Singh, H. (1994). Application of correlation filters for texture recognition. *Applied Optics*, 33(11):2173–2179.

[Malik et al., 2001] Malik, J., Belongie, S., Leung, T., and Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27.

[Malik and Perona, 1990] Malik, J. and Perona, P. (1990). Preattentive texture discrimination with early vision mechanism. *Journal of the Optical Society of America*, 7(5):923–932.

[Malik and Rosenholtz, 1997] Malik, J. and Rosenholtz, R. (1997). Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23(2):149–168.

[Mamic and Bennamoun, 2000] Mamic, G. and Bennamoun, M. (2000). Automatic flaw detection in textiles using a neyman-pearson detector. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 767–770, Barcelona, Spain.

[Manjunath and Ma, 1996] Manjunath, B. S. and Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.

[Messer et al., 1999] Messer, K., de Ridder, D., and Kittler, J. (1999). Adaptive texture representation methods for automatic target recognition. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 443–452, Nottingham, UK.

[Meyer et al., 2000] Meyer, F. G., Averbuch, A. Z., and Strömberg, J. O. (2000). Fast adaptive wavelet packet image compression. *IEEE Transactions on Image Processing*, 9(5):792–800.

[Miller and Astley, 1992] Miller, P. and Astley, S. (1992). Classification of breast tissue by texture analysis. *Image and Vision Computing*, 10(5):277–282.

[Miranda et al., 1998] Miranda, F. P., Fonseca, L. E. N., and Carr, J. R. (1998). Semivariogram textural classification of JERS-1 (fuyo-1) sar data obtained over a flooded area of the amazon rainforest. *International Journal of Remote Sensing*, 19(3):549–556.

[Mojsilovic et al., 2000] Mojsilovic, A., Popovic, M. V., and Rackov, D. M. (2000). On the selection of an optimal wavelet basis for texture characterization. *IEEE Transactions on Image Processing*, 9(12):2043–2050.

[Mollineda et al., 2002] Mollineda, R. A., Ferri, F. J., and Vidal, E. (2002). A merge-based condensing strategy for multiple prototype classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32(5):662–668.

[Mori et al., 2004] Mori, G., Ren, X., Efros, A., and Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 326–333, Washington, DC.

[Newman and Jain, 1995] Newman, T. and Jain, A. K. (1995). A survey of automated visual inspection. *Computer Vision and Image Understanding*, 61(2):231–262.

[Nillius and Eklundh, 2001] Nillius, P. and Eklundh, J.-O. (2001). Automatic estimation of the projected light source direction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1076–1083, Kauai, Hawaii.

[Olshausen and Field, 2005] Olshausen, B. A. and Field, D. J. (2005). How close are we to understanding v1? *Neural Computation*. to appear.

[Ozdemir et al., 1998] Ozdemir, S., Baykut, A., Meylani, R., Ercil, A., and Ertuzun, A. (1998). Comparative evaluation of texture analysis algorithms for defect inspection of textile products. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 1738–1740, Brisbane, Australia.

[Paget, 1999] Paget, R. (1999). *Nonparametric Markov random field models for natural texture images*. PhD thesis, University of Queensland.

[Paragios and Deriche, 2002] Paragios, N. and Deriche, R. (2002). Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247.

[Penirschke et al., 2002] Penirschke, A., Chantler, M. J., and Petrou, M. (2002). Illuminant rotation invariant classification of 3D surface textures using lissajous's ellipses. In *Proceedings of the Second International Workshop on Texture Analysis and Synthesis*, pages 103–108, Copenhagen, Denmark.

[Pentland, 1982] Pentland, A. P. (1982). Finding the illuminant direction. *Journal of the Optical Society of America*, 72:448–455.

[Perona, 1992] Perona, P. (1992). Steerable-scalable kernels for edge detection and junction analysis. In *Proceedings of the European Conference on Computer Vision*, pages 3–18, Ligure, Italy.

[Perona, 1995] Perona, P. (1995). Deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):488–499.

[Petrick et al., 1996] Petrick, N., Chan, H. P., Wei, D., Sahiner, B., and Helvie, M. A. (1996). Automated detection of breast masses on mammograms using adaptive contrast enhancement and texture classification. *Medical Physics*, 23(10):1685–1696.

[Petroudi et al., 2003] Petroudi, S., Kadir, T., and Brady, M. (2003). Automatic classification of mammographic parenchymal patterns: A statistical approach. In *Proceedings of the IEEE Conference of the Engineering in Medicine and Biology Society*, Beach Cancun, Mexico.

[Pickup et al., 2003] Pickup, L. C., Roberts, S. J., and Zisserman, A. (2003). A sampled texture prior for image super-resolution. In *Advances in Neural Information Processing Systems*.

[Plantier et al., 2002] Plantier, J., Boutte, L., and Lelandais, S. (2002). Defect detection on inclined textured planes using the shape from texture method and the delaunay triangulation. *EURASIP Journal of Applied Signal Processing*, 7:659–666.

[Podest and Saatchi, 2002] Podest, E. and Saatchi, S. (2002). Application of multiscale texture in classifying JERS-1 radar data over tropical vegetation. *International Journal of Remote Sensing*, 23(7):1487–1506.

[Popat and Picard, 1993] Popat, K. and Picard, R. W. (1993). Novel cluster-based probability model for texture synthesis, classification, and compression. In *Proceedings of the SPIE Conference on Visual Communication and Image Processing*, pages 756–768, Boston, Massachusetts.

[Portilla and Simoncelli, 2000] Portilla, J. and Simoncelli, E. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70.

[Press et al., 1992] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C*. Cambridge University Press, second edition.

[Pun and Lee, 2003] Pun, C. M. and Lee, M. C. (2003). Log-polar wavelet energy signatures for rotation and scale invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):590–603.

[Ran and Farvardin, 1992] Ran, X. and Farvardin, N. (1992). Adaptive dct image coding on a three-component image model. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 201–204, San Francisco, California.

[Randen, 1997] Randen, T. (1997). *Filter and Filter Bank Design for Image Texture Recognition*. PhD thesis, Norwegian University of Sciene and Technology.

[Randen and Husoy, 1999] Randen, T. and Husoy, J. H. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310.

[Ravela, 2004] Ravela, S. (2004). Shaping receptive fields for affine invariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 725–730, Washington, DC.

[Reed and Wechsler, 1990] Reed, T. R. and Wechsler, H. (1990). Segmentation of textured images and gestalt organization using spatial/spatial-frequency representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):1–12.

[Rellier et al., 2004] Rellier, G., Descombes, X., Falzon, F., and Zerubia, J. (2004). Texture feature analysis using a gauss-markov model in hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1543–1551.

[Ren and Malik, 2003] Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 10–17, Nice, France.

[Romdhani et al., 2001] Romdhani, S., Torr, P., Scholkopf, B., and Blake, A. (2001). Computationally efficient face detection. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 695–700, Vancouver, Canada.

[Rosenholtz and Malik, 1997] Rosenholtz, R. and Malik, J. (1997). Surface orientation from texture: Isotropy or homogeneity (or both)? *Vision Research*, 37:2283–2293.

[Rother et al., 2004] Rother, C., Kolmogorov, V., and Blake, A. (2004). GrabCut - interactive foreground extraction using iterated graph cuts. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, Los Angeles, California.

[Rubner et al., 2000] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

[Ruiz et al., 2004] Ruiz, L. A., Fdez-Sarria, A., and Recio, J. A. (2004). Texture feature extraction for classification of remote sensing data using wavelet decomposition: A comparative study. In *XXth ISPRS Congress*, pages 1109–1114, Istanbul, Turkey.

[Sakai and Finkel, 1994] Sakai, K. and Finkel, L. H. (1994). A shape-from-texture algorithm based on human visual psychophysics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 527–532, Seattle, Washington.

[Sandberg et al., 2002] Sandberg, B., Chan, T. F., and Vese, L. A. (2002). 'a level-set and gabor-based active contour algorithm for segmenting textured images. Technical Report CAM Report 02-39, UCLA.

[Savarese and Criminsi, 2004] Savarese, S. and Criminsi, A. (2004). Classification of folded textiles. Personal communications.

[Schaffalitzky and Zisserman, 2001] Schaffalitzky, F. and Zisserman, A. (2001). Viewpoint invariant texture matching and wide baseline stereo. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 636–643, Vancouver, Canada.

[Schistad and Jain, 1992] Schistad, A. H. and Jain, A. K. (1992). Texture analysis in the presence of speckle noise. In *IEEE Geoscience and Remote Sensing Symposium*, Houston, Texas.

[Schmid, 2001] Schmid, C. (2001). Constructing models for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45, Kauai, Hawaii.

[Scholkopf et al., 1996] Scholkopf, B., Burges, C. J. C., and Vapnik, V. (1996). Incorporating invariances in support vector learning machines. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 1112, pages 47–52, Berlin, Germany.

[Scholkopf and Smola, 2002] Scholkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press.

[Sebe and Lew, 2000] Sebe, N. and Lew, M. S. (2000). Wavelet-based texture classification. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 959–962, Barcelona, Spain.

[Siew et al., 1988] Siew, L. H., Hodgson, R. M., and Wood, E. J. (1988). Texture measures for carpet wear assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(1):92–105.

[Simard et al., 2000] Simard, M., Saatchi, S. S., and de Grandi, G. (2000). The use of decision tree and multiscale texture for classification of JERS-1 SAR data over tropical forest. *IEEE Transactions on Geoscience and Remote Sensing*, 38(5):2310–2321.

[Simoncelli and Portilla, 1998] Simoncelli, E. and Portilla, J. (1998). Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 62–66, Chicago, Illinois.

[Singh and Singh, 2002] Singh, M. and Singh, S. (2002). Spatial texture analysis: A comparative study. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 676–679, Quebec, Canada.

[Smith and Chang, 1994] Smith, J. R. and Chang, S. F. (1994). Transform features for texture classification and discrimination in large image databases. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 407–411, Austin, Texas.

[Solberg and Jain, 1997] Solberg, A. H. S. and Jain, A. K. (1997). Texture fusion and feature-selection applied to SAR imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 35(2):475–479.

[Song et al., 1992] Song, K. Y., Petrou, M., and Kittler, J. (1992). Texture defect detection: A review. In *SPIE Conferences*, volume 1708, pages 99–106.

[Suen and Healey, 2000] Suen, P. and Healey, G. (2000). The analysis and reconstruction of real-world textures in three dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):491–503.

[Super and Bovik, 1991] Super, B. J. and Bovik, A. C. (1991). Localized measurement of image fractal dimension using gabor filters. *Journal of Visual Communication and Image Representation*, 2(2):114–128.

[Sutton and Hall, 1972] Sutton, R. and Hall, E. L. (1972). Texture measures for automatic classification of pulmonary disease. *IEEE Transactions on Computers*, C-21:667–676.

[Thacker et al., 1997] Thacker, N. A., Ahearne, F., and Rockett, P. I. (1997). The bhattacharya metric as an absolute similarity measure for frequency coded data. *Kybernetica*, 34(4):363–368.

[Thomas, 1998] Thomas, T. R. (1998). *Rough Surfaces*. Imperial College Press, London, second edition.

[Tipping, 2001] Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.

[Todd and Oomes, 2002] Todd, J. T. and Oomes, A. H. (2002). Generic and non-generic conditions for the perception of the surface shape from texture. *Vision Research*, 42(7):837–850.

[Todd et al., 2004] Todd, J. T., Oomes, A. H., Koenderink, J. J., and Kappers, A. M. L. (2004). The perception of doubly curved surfaces from anisotropic textures. *Psychological Science*, 15(1):40–46.

[Topsoe, 2000] Topsoe, F. (2000). Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 46(4):1602–1609.

[Toussaint, 2002] Toussaint, G. (2002). Proximity graphs for nearest neighbor decision rules: recent progress. In *Interface 2002, 34th Symposium on Computing and Statistics*.

[Tu and Zhu, 2002] Tu, Z. W. and Zhu, S. C. (2002). Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):657–673.

[Tuceryan and Jain, 1990] Tuceryan, M. and Jain, A. K. (1990). Texture segmentation using voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):211–216.

[Tuceryan and Jain, 1998] Tuceryan, M. and Jain, A. K. (1998). Texture analysis. In Chen, C. H., Pau, L. F., and Wang, P. S. P., editors, *Handbook of Pattern Recognition and Computer Vision*, pages 207–248. World Scientific, second edition.

[Turner, 1986] Turner, M. R. (1986). Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82.

[Unsalan and Ercil, 1999] Unsalan, C. and Ercil, A. (1999). Automated inspection of steel structures. In Kaynak, O., Tosunoglu, S., and Ang, M., editors, *Recent Advances in Mechatronics*, pages 468–480. Springer-Verlag.

[Unser, 1986] Unser, M. (1986). Local linear transforms for texture measurements. *Signal Processing*, 11(1):61–79.

[Varma and Zisserman, 2002a] Varma, M. and Zisserman, A. (2002a). Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 255–271, Copenhagen, Denmark.

[Varma and Zisserman, 2002b] Varma, M. and Zisserman, A. (2002b). Classifying materials from images: to cluster or not to cluster? In *Proceedings of the Second International Workshop on Texture Analysis and Synthesis*, pages 139–144, Copenhagen, Denmark.

[Varma and Zisserman, 2002c] Varma, M. and Zisserman, A. (2002c). Statistical approaches to material classification. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, pages 167–172, Ahmedabad, India.

[Varma and Zisserman, 2003] Varma, M. and Zisserman, A. (2003). Texture classification: Are filter banks necessary? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 691–698, Madison, Wisconsin.

[Varma and Zisserman, 2004] Varma, M. and Zisserman, A. (2004). Unifying statistical texture classification frameworks. *Image and Vision Computing*, 22(14):1175–1183.

[Varma and Zisserman, 2005] Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, 62(1–2):61–81.

[Vasconcelos and Lippman, 2000] Vasconcelos, N. and Lippman, A. (2000). A unifying view of image similarity. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 1038–1041, Barcelona, Spain.

[Vega and Yang, 1994] Vega, E. V. and Yang, Y.-H. (1994). Default shape theory: With the application to the computation of the direction of the light source. *Journal of the Optical Society of America*, 60:285–299.

[Viola, 1995] Viola, P. (1995). *Alignment by Maximization of Mutual Information*. PhD thesis, MIT, AI Lab.

[Voorhees and Poggio, 1988] Voorhees, H. and Poggio, T. (1988). Computing texture boundaries from images. *Nature*, 333:364–367.

[Wallraven et al., 2003] Wallraven, C., Caputo, B., and Graf, A. (2003). Recognition with local features: the kernel recipe. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 257–264, Nice, France.

[Wang and Dana, 2004] Wang, J. and Dana, K. J. (2004). Hybrid textons: Modeling surfaces with reflectance and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 372–378, Washington, DC.

[Wei and Levoy, 2000] Wei, L.-Y. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 479–488, New Orleans, Louisiana.

[Weldon and Higgins, 1996] Weldon, T. P. and Higgins, W. E. (1996). Integrated approach to texture segmentation using multiple gabor filters. In *Proceedings of the IEEE International Conference on Image Processing*, pages 955–958, Lausanne, Switzerland.

[Wertheimer, 1958] Wertheimer, M. (1958). Principles of perceptual organization. In Beardslee, D. C. and Wertheimer, M., editors, *Readings in Perception*, pages 115–135. Van Nostrand, Princeton NJ.

[Weszka et al., 1976] Weszka, J. S., Dyer, C. R., and Rosenfeld, A. (1976). Comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(4):269–285.

[Wilson, 1972] Wilson, D. L. (1972). Asymptotic properties of nearest neighbour rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–420.

[Wiltschi et al., 2000] Wiltschi, K., Pinz, A., and Lindeberg, T. (2000). An automatic assessment scheme for steel quality inspection. *Machine Vision and Applications*, 12:113–128.

[Witkin, 1981] Witkin, A. P. (1981). Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45.

[Wu and Chantler, 2003] Wu, J. and Chantler, M. J. (2003). Combining gradient and albedo data for rotation invariant classification of 3D surface texture. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 848–855, Nice, France.

[Xie and Brady, 1996] Xie, Z. Y. and Brady, M. (1996). Texture segmentation using local energy in wavelet scale space. In *Proceedings of the European Conference on Computer Vision*, pages 304–315, Cambridge, UK.

[Xu et al., 2000] Xu, K., Georgescu, B., Comaniciu, D., and Meer, P. (2000). Performance analysis in content-based retrieval with textures. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 275–278, Barcelona, Spain.

[Yang and Yuille, 1991] Yang, Y. and Yuille, A. L. (1991). Sources from shading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 534–539, Maui, Hawaii.

[Yuille et al., 1999] Yuille, A. L., Snow, D., Epstein, R., and Belhumeur, P. (1999). Determining generative models for objects under varying illumination: Shape and albedo from multiple images using SVD and integrability. *International Journal of Computer Vision*, 35(3):203–222.

[Zalesny and Van Gool, 2000] Zalesny, A. and Van Gool, L. (2000). A compact model for viewpoint dependent texture synthesis. In *Proceedings of the European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 124–143, Dublin, Ireland.

[Zheng and Chellappa, 1991] Zheng, Q. and Chellappa, R. (1991). Estimation of illuminant direction, albedo and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):680–702.

[Zhu et al., 1996] Zhu, S. C., Wu, Y. N., and Mumford, D. B. (1996). FRAME: Filters, random fields and maximum entropy – to a unified theory for texture modeling. Technical report, Harvard Robotics Lab.

[Zhu et al., 1998] Zhu, S. C., Wu, Y. N., and Mumford, D. B. (1998). Filters, random-fields and maximum-entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126.

[Zhu and Yuille, 1996] Zhu, S. C. and Yuille, A. L. (1996). Region competition: Unifying snake/balloon, region growing and bayes/mdl/energy for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900.