

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR Sciences

Ecole doctorale : STIC

THESE

pour obtenir le titre de

Docteur en Sciences

de l'UNIVERSITE de Nice-Sophia Antipolis

Spécialité : Traitement des Images

présentée et soutenue par

Karen BRADY

A PROBABILISTIC FRAMEWORK FOR ADAPTIVE TEXTURE DESCRIPTION

Thèse dirigée par Ian JERMYN et Josiane ZERUBIA

soutenue le 17 décembre 2003

JURY:

M. Michel BARLAUD	Président
M. Mohamed NAJIM	Rapporteur
M. Ioannis PITAS	Rapporteur
M. Simon WILSON	Examineur
M. Ian JERMYN	Examineur
M. Josiane ZERUBIA	Examineur

to my parents, Nuala and Noel

Acknowledgments

I would like to thank the members of my thesis committee: the president Michel Barlaud; the reporters Mohamed Najim and Ioannis Pitas; and Simon Wilson. Their comments and questions were very insightful and prompted some interesting discussions.

A little over three years ago, Josiane Zerubia welcomed me to the Ariana research group at INRIA Sophia Antipolis. I would like to thank Josiane for her ongoing support throughout my stay with the group.

Starting a thesis in image processing without any previous training in the subject area was a big challenge for me. I cannot imagine having got this far without the help and guidance of Ian Jermyn. His enthusiasm for research together with his broad scientific knowledge made for lively discussions and surprising tangents. His patience and motivation were invaluable to me during my thesis.

I would like to thank the members of the Ariana project. Office B121 would not have been the same without the presence of Andre and Guillaume who both were a wonderful source of fun and, of course, ongoing advice on the nuances of the French language!

My thesis would not have been possible without the financial support of the European Union network MOUMIR. In particular, I would like to express my gratitude to Anil Kokaram for his help and encouragement. Being part of the network has been a very enjoyable experience and has led to many interesting scientific and non-scientific exchanges during the workshops and visits.

I would like to thank Georgy Gimel'farb and Abhir Bhalerao for their interesting discussions on texture analysis methods and for the images they provided for comparison purposes. I would also like to thank IGN, the French Mapping Institute, for the donation of some of the remote sensing images used in this thesis and Space Imaging for permission to use others.

Finishing off on a more personal note, there are a handful of people who deserve a special mention: Yuling and Wasi for their long distance support since our days together in the States; Charith, Clare, and Oscar for making my stay in France a more sane one; my sister and brothers for putting my life in perspective by always managing to have a bigger crisis than me when I thought things could not get any worse; my parents for their constant love and encouragement from the very beginning; and last, but by no means least, Julien for his incredible support.

Contents

1	Introduction	1
1.1	Texture: An Important Visual Feature	1
1.2	Texture Detection	2
1.3	Previous Approaches to Texture Analysis	2
1.4	Objectives of this Thesis	3
1.5	Outline of the Thesis	3
2	Texture Analysis	5
2.1	Important Texture Properties	6
2.1.1	Regionality	6
2.1.2	Resolution	7
2.2	A Brief Overview of Texture Analysis Methods	7
2.2.1	Statistical Approaches	8
2.2.2	Model-Based Methods	10
2.2.3	Structural Methods	11
2.2.4	Spatial and Spatial-Frequency Filtering Methods	11
2.3	A Multiresolution Approach to Texture Analysis	12
2.3.1	Evidence Supporting Frequency Tuning	12
2.3.2	Gabor Filters	14
2.3.3	Wavelets	15
2.3.4	Wavelet Frames	19
2.3.5	Wavelet-Domain Tree Models	19
2.3.6	Wavelet Packets	20
2.3.7	Adaptive Wavelet Packets	21
2.4	Our Model	24
3	Theoretical Framework	27
3.1	A Gaussian Distribution for Texture	30
3.1.1	Expressing the Energy in Terms of ϕ_R and $\phi_{\bar{R}}$	32
3.1.2	Marginalising over Elements in $\phi_{\bar{R}}$	33
3.1.3	Diagonalisation of the Operator G_R	35
3.2	Translation Invariance	36
3.2.1	Translation Operator	37

3.2.2	Requiring Translation Invariance	37
3.3	Using Wavelet Packets	39
3.3.1	Dyadic Shaped Regions	43
3.3.2	Arbitrarily Shaped Regions	44
4	Training Texture Models	45
4.1	The Likelihood Function	46
4.1.1	Multiple Training Patches	46
4.2	The Prior Probability	47
4.3	Estimating the Texture Model Parameters	48
4.3.1	Algorithm	50
4.4	Training Results	52
4.5	Discussion: An Analysis of the Subband Statistics	56
5	Segmenting Textured Scenes	67
5.1	A Probabilistic Statement of the Problem	68
5.1.1	Choosing a Prior for the Class Map	70
5.1.2	Computing the Likelihood	70
5.2	Application: Brodatz Mosaics	72
5.2.1	Results	74
5.2.2	Comparison With a Gibbs Texture Model	77
5.2.3	Comparison With a Discriminative Feature Selection Approach	78
5.3	Application: Remote Sensing Images	80
5.3.1	Results	81
5.4	Application: Wildlife Scenes	86
5.5	Discussion: The Effect of the Mother Wavelet	90
6	Conclusion and Future Work	95
6.1	Summary of the Work	95
6.2	Understanding What the Adaptive Models Achieve	96
6.3	Future Work	97
6.3.1	Selecting the Mother Wavelet	97
6.3.2	The Need for a More Complex Distribution	98
A	Wavelets	99
A.1	The Wavelet Transform	99
A.2	A Multiresolution Analysis Framework	101
B	Completing the Square in $\phi_{\bar{R}}$	105
C	Implementation	109
C.1	The Java Programming Language	109
C.2	The Use of Class Inheritance	111
C.3	Multithreaded Programming	111

Contents

iii

D Publications of the Author

113

List of Tables

2.1	Features extracted from a co-occurrence matrix	9
5.1	Classification error rates for Brodatz mosaic set A	74
5.2	Classification error rates for Brodatz mosaic set B	74
5.3	Classification error rates for Brodatz mosaic set B using a discriminative feature selection approach	80
5.4	Classes defined for the remote sensing images	83
5.5	An example of an error matrix	83
5.6	Kappa values for the segmentation results on the remote sensing images	84
5.7	Classification error rates for the Brodatz mosaics segmented using differ- ent mother wavelets	92

List of Figures

2.1	Texture categories	6
2.2	Texture and regionality	7
2.3	Texture and resolution	7
2.4	Image coordinates and directions used in a co-occurrence matrix	8
2.5	Examples of co-occurrence matrices	9
2.6	A first-order neighbourhood and its corresponding cliques	10
2.7	Sinusoidal gratings and their corresponding fundamental waveforms	12
2.8	Contrast sensitivity function	13
2.9	1-D wavelet transform filter bank representation	16
2.10	A 1-D wavelet transform down to level 3	16
2.11	2-D wavelet transform filter bank representation	17
2.12	A 2-D wavelet transform down to level 3	17
2.13	Frequency content of a selection of textures	20
2.14	A 1-D wavelet packet transform down to level 3	21
2.15	Frequency content of the Bark texture from the Brodatz album	22
2.16	An adaptive wavelet packet analysis of a 1-D signal	22
2.17	An adaptive wavelet packet analysis of an image	23
3.1	Textured regions in high resolution aerial images	28
3.2	Maps defined on the space of infinite images	29
3.3	The components of the operator F	31
3.4	Translated patches of the D102 texture from the Brodatz album	37
3.5	Possible partitions of the frequency domain	39
3.6	Partitioning the frequency domain with horizontal splits.	40
3.7	Partitioning the frequency domain using a mix of all three types of splits	40
3.8	A partition of the frequency domain corresponding to a standard wavelet decomposition	41
3.9	Frequency support of several wavelet and wavelet packet basis functions	42
4.1	The patch selection procedure used in the training phase	47
4.2	The redundancy factor of a node in a decomposition tree	48
4.3	The effect of the prior probability on the decomposition T	49

4.4	Pseudocode describing the depth-first search algorithm	51
4.5	An illustration of the decision process involved in selecting the optimal wavelet packet decomposition for a texture patch	53
4.6	The frequency content of several Brodatz textures	54
4.7	The frequency content of several images cropped from high resolution aerial images	55
4.8	Adaptive decompositions for the Herring texture	57
4.9	Adaptive decompositions for the Raffia texture	58
4.10	Adaptive decompositions for the Wood texture	59
4.11	Adaptive decompositions for the Plough texture	60
4.12	Adaptive decompositions for the Forest texture	61
4.13	Adaptive decompositions for the Warehouse texture	62
4.14	An analysis of the adaptive wavelet packet subband statistics	63
4.15	Testing the suitability of a quartic distribution to model the bimodal form of certain subband histograms	65
5.1	A toy example: a textured scene to be segmented	67
5.2	Examples of possible class maps	68
5.3	The application of the wavelet filter to an arbitrarily-shaped region	72
5.4	Examples of neighbourhoods used in the segmentation scheme	74
5.5	Segmentation results on Brodatz mosaic set A	75
5.6	Segmentation results on Brodatz mosaic set B	76
5.7	Segmentation results on Brodatz mosaic set A using a Gibbs texture model	77
5.8	Training mosaics used in the Gibbs texture model approach	78
5.9	Segmentation results on Brodatz mosaic set B using a discriminative feature selection approach	79
5.10	Segmentation results for a collection of remote sensing images	82
5.11	Entropy images for the segmentation results on the remote sensing images	86
5.12	Confidence in labelling the pixels in the entropy images	86
5.13	Entropy-weighted misclassification images for the segmentation results on the remote sensing images	86
5.14	Confidence in labelling the pixels in the entropy-weighted misclassification images	87
5.15	Scenes of wildlife in their natural habitat	87
5.16	Ground truth for the wildlife scenes	88
5.17	Patches selected from the Background texture	88
5.18	Patches selected from the Zebra texture	89
5.19	Patches selected from the Leaves texture	89
5.20	Patches selected from the Jaguar texture	89
5.21	Patches selected from the Trunk texture	90
5.22	Segmentation results for the wildlife images	91
5.23	The effect of the mother wavelet on the adaptive decomposition	93

5.24	Segmentation results on some Brodatz mosaics using different mother wavelets	94
A.1	Scaled and translated versions of the Daubechies2 [22] mother wavelet. .	100
A.2	Space-frequency localisation of wavelet basis functions.	101
C.1	Comparing Java and C for matrix multiplication operations	110

Chapter 1

Introduction

Humans use computers for a variety of reasons. One of the most obvious applications of computers are tasks which involve multiplying and dividing numbers or remembering huge amounts of data very precisely. Computers are very good at performing such tasks, which is quite fortunate, as humans are usually very bad at doing these things. Historically, these tasks were the reasons why computers were used. This is obvious when one thinks of the origins of the word *computer*. Interestingly enough though, there are other tasks which pose significant problems to computers, yet humans perform them in a routine manner. A large part of the research in computer science is now devoted to making computers labouriously solve problems that humans continually solve, usually without even thinking about the task. These include, for example, understanding natural language, recognising faces in a crowd, and playing the game of Go. Detecting texture in scenes is another example of such a task, which to humans is a natural daily procedure but which challenges computers. This last problem is the topic of this thesis. This chapter will present the problem in more detail, introduce previous approaches to solving it, present our approach, and give a brief outline of the rest of the thesis.

1.1 Texture: An Important Visual Feature

Texture is as much a part of our visual environment as other features such as shape and colour. It is something which we process on a daily basis without ever giving much thought to what it actually means. If pressed for a definition, most of us would probably utter something about patterns that repeat themselves. The vagueness of such a comment highlights the difficulty involved in providing a precise definition of texture.

Although it is tough to describe, texture is present in almost every scene we encounter. It can be a characteristic of a large expanse, such as a forest canopy seen from the seat of an airplane, or it can be the minute detail painted around the rim of a cup. Texture can also be highly complicated, such as in the patterns found in Islamic and Celtic art, or it can be as simple as a checkerboard. With so much variety in its appearance, it is impressive, and indeed amazing at times, that humans have the ability to

detect texture in a multitude of different scenes. However it is this variety that makes the task of building a reliable texture analysis system so difficult.

The ability to describe and analyse texture is an integral part of many image processing applications. In the medical domain, for example, texture based features extracted from radiographic images provide specialists with an automatic aid for the diagnosis of malignant tissue [40]. Texture is also used in industry for the purposes of quality control [26] and can help in the automatic detection of defects, such as those found in lumber wood [19]. In many remote sensing images, texture is a salient cue for applications including cartography and damage assessment following natural disasters such as fires or floods.

1.2 Texture Detection

Before attempting to analyse a textured scene, however, one must first decide which approach to adopt: unsupervised or supervised. In the former, no knowledge is known about the textures present in the scene. The scene is divided into textured regions and no comment is made about the real world entities that these resulting regions represent. This approach is frequently used in remote sensing methods [69, 83], but can also be used in general segmentation tasks [73]. The advantage of such methods is that they reduce the amount of user interaction to a minimum and the segmentation can take place without any need for a learning phase. In the latter approach, a knowledge of the textures is required before segmentation can take place. Training is necessary in order to learn the characteristic features for each texture in the scene. The resulting segmentation assigns labels to each region corresponding to the real world entity it represents. Although these methods usually involve an intensive learning phase, they have the added advantage of producing a classification of the scene which can be evaluated via a comparison with the ground truth for the scene. Therefore they provide a means by which the texture analysis method can be evaluated. It is for this reason that we choose to adopt an supervised approach in this thesis.

1.3 Previous Approaches to Texture Analysis

Over the years numerous approaches have been proposed to tackle the problem of texture description. For a full overview, see [42, 48, 65, 68, 77]. One research area which has been extremely active in recent years is the application of wavelets to texture analysis [57, 60, 79]. Many textures, however, possess intermediate and high frequency content. By repetitively decomposing only the low frequencies, the standard wavelet transform does not further analyse these other areas of the frequency domain. To address this problem, one can perform a wavelet packet analysis of the texture. In [52], packets were used in a classification experiment on natural textures. The problem with this is that performing a full wavelet packet decomposition results in a large number of subbands to be analysed

and some of these will not contain much information. This was the motivation behind many of the efforts made to adapt the wavelet packet decomposition to the underlying structure of the texture. The pioneering work of [16] selected the best wavelet packet basis for an image by using an entropy criterion to decide if a subband should be further decomposed or not. Although this work was originally motivated by the need for an efficient image compression technique, it has since then been applied specifically for the purposes of texture classification [11]. Interesting though these methods are, none of them were developed within a coherent probabilistic framework; they involve *ad hoc* choices that fail to reveal the assumptions underlying the models.

1.4 Objectives of this Thesis

As will be discussed later in chapter 2, a multiresolution approach to image analysis such as the one provided by wavelets is ideal for the analysis of textures which, by their very nature, contain information at a variety of scales. We retain the advantages of such an approach by using wavelet packets as our texture analysis tool.

However, unlike the other wavelet-based approaches described above, in this thesis we will build our models within a probabilistic framework. Starting from a probability measure on the infinite texture, which we assume here to be Gaussian, we derive the measure for the texture on a finite region. This leads naturally to a class of adaptive wavelet packet models that capture the structure of a given texture, for example its principle periodicities, in a manner analogous to the 2-D Wold decomposition [32, 64]. The major strength of our approach is the coherent decision making process by which the models are trained.

1.5 Outline of the Thesis

The thesis is structured as follows.

In chapter 2, we discuss the three major categories of textures: deterministic, stochastic and observable. We then identify the two main properties of texture, namely regionality and resolution, that should be taken into consideration when attempting to develop a texture analysis system. An overview of some of the existing texture analysis methods is provided. This includes such techniques as co-occurrence matrices and Markov random fields. The processing of patterns is then discussed from a psychophysical viewpoint and observations are made regarding the suitability of wavelets as a model for the human visual system. We then present a summary of some of the wavelet-based approaches which have been proposed for the purpose of texture analysis. We end with a brief comment on the method developed in this thesis and how it compares to other existing wavelet-based techniques.

Chapter 3 details the theoretical framework proposed in this thesis. We start by making the observation that texture is naturally described via a distribution on the

space of infinite images. We choose this distribution to be Gaussian. As texture can appear anywhere translated in a scene, we require the distribution to be translation invariant. This causes the Gaussian distribution to be diagonal in the Fourier basis. To process an image however, one needs a distribution on a finite region. We detail the marginalisation steps involved in arriving at this distribution from the original distribution on the space of infinite images. The form of the operator of the resulting distribution on the finite region makes the calculation of probabilities a computationally intensive task. So we require that the operator be diagonalised. One way to achieve this is to place a restriction on the form of the inverse covariance of the distribution, namely that it is piecewise constant on a dyadic partition of the frequency domain. Wavelet packets, whose support is local in both the spatial and frequency domains, provide a basis in which this diagonalisation procedure is possible. This results in wavelet packet texture models.

The issue of training these models is addressed in chapter 4. A selection of dyadic shaped patches are used as the training data set. We detail the choice of distributions for the estimation of the parameter set for each model. A depth first search algorithm is implemented which finds an adaptive wavelet packet decomposition for each texture. Visualisations of these adaptive decompositions are presented. We carry out a statistical analysis of the subbands of the resulting decompositions to examine the histograms. An interesting discovery of a bimodal form is made.

Chapter 5 treats the topic of texture segmentation and classification. A collection of different images are used, including mosaics created from the Brodatz album, high resolution remote sensing images, and scenes of wildlife in their natural habitat. Two simple classification rules are described, both of which enable pixelwise classification of the image while retaining the advantages of more complicated prior models. We consider supervised segmentation because it is methodologically well-defined. Results are presented. A brief discussion on the choice of the mother wavelet follows.

In chapter 6 we conclude with a summary of the work and highlight the benefits of using the texture models developed in this thesis. We discuss some of the drawbacks of our method and suggest possible solutions which could be addressed in future work.

Chapter 2

Texture Analysis

Texture is all around us in our visual world. It glares up at us from the cobblestone paved streets and from the brick wall of the school. It is also there, but less obviously so, in the floral print on a summer dress and in the ripples that form on the surface of a lake. So what do these images, which are so different in appearance, have in common that they can all be referred to as texture? It is in trying to answer this question that one becomes aware of the complexity of texture. Many people have attempted to describe it [14, 77]. Despite this effort, a single definition has not been agreed upon and so texture remains an easily identifiable yet ultimately undefinable visual feature.

Although texture has proven to be an elusive concept, a given textured image can be roughly categorised as belonging to one of three different groups [37]:

- **Deterministic:** In this case, the texture is made up of primitives which are identical in nature and which always appear in the same order. Hence it can be fully described via the characteristics of one single primitive along with the set of rules which govern the spatial distribution of the primitives and hence determine where they should be placed relative to one another. An example of this type of texture is shown in figure 2.1(a).
- **Stochastic:** Texture of this kind obeys certain statistical laws. White noise is an example of such a texture and can be seen in figure 2.1(b).
- **Observable:** This describes a texture which is not ideal. It does not adhere to a fixed set of rules or laws. It is neither exactly deterministic nor stochastic, but instead falls somewhere in between the two [67]. An example of such a texture is given in figure 2.1(c). A texture of this type may at first seem to have a more or less repetitive nature, but this may not be strict. Also if it consists of primitives, they may seem to strongly resemble each other, but they might not be identical. This type of texture has also been referred to elsewhere in the literature as *visual texture* [77].

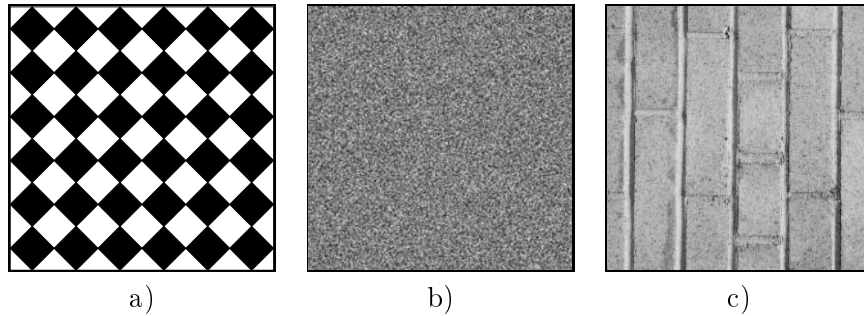


Figure 2.1: Three texture categories: a) deterministic; b) stochastic; c) observable.

The majority of textures we encounter in our visual environment fall into this third category of observable texture. This is due to the fact that they are visual representations of real world objects which contain natural variations in their physical structure. These changes in the physical world entities appear as intensity variations in the textured image and it is this that we must capture if we are to model texture accurately.

2.1 Important Texture Properties

Understanding the underlying characteristics of the texture which give rise to these variations in the image is a first step toward modelling the texture. In [37], the coarseness and directionality of the pattern are considered to be the two major defining characteristics of a texture. There have been many other terms used in the literature to describe the characteristics of texture, including contrast, granulation, randomness, lineation, and hummocky [42, 55].

Whatever adjectives one may use, there are a couple of properties which are common to almost all textures and should therefore be taken into consideration in the modelling process. These two properties are described below.

2.1.1 Regionality

Texture, by its very nature, is a property of a region or area. If one looks at a single pixel such as that on the left hand side of figure 2.2, the notion of texture does not exist. However as soon as neighbouring pixels are brought into the picture, texture becomes a possible feature for that area. Depending on the size of the area, information on the structure of the texture can change, as can be seen from the next two images in the figure. This property makes texture a particularly difficult thing to analyse because it creates problems at the boundary of textured regions. Pixels which lie on or around the boundary are difficult to classify because, depending on the size of the neighbourhood chosen to analyse the pixel, there may be equal amounts of information being contributed from several different textures.

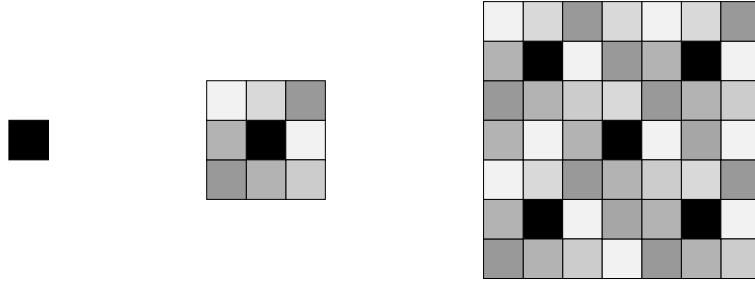


Figure 2.2: Regionality: increasing the size of the neighbourhood of a pixel allows us to detect the presence of texture.

2.1.2 Resolution

Inherent to most textures is the notion of scale. Examining a texture at different resolutions can help in understanding its underlying structure. The sequence of resolutions in figure 2.3 demonstrate this idea. As we zoom out from the image we lose the information on the small microtexture present inside each block and instead find ourselves interpreting the larger checkerboard macrotexture. Viewing a texture at only one single scale can therefore be potentially misleading and can give rise to the situation of not being able to see the forest for the trees.

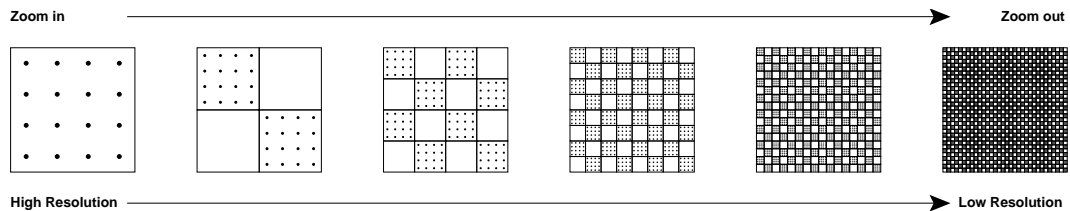


Figure 2.3: Resolution: viewing the same texture at different scales.

2.2 A Brief Overview of Texture Analysis Methods

Over the years many different approaches have been proposed to extract textural information from images. Review papers on this topic [37, 42, 65, 68, 74] easily lead one to the conclusion that there is no one single all encompassing analysis method which works for all types of texture and for every application [77]. This is an intuitive statement and should be something to bear in mind when setting out to describe texture. In this chapter we will give a brief outline of the main groups of methods used for texture analysis. Further details will be given on the methods which are most relevant to this thesis, namely those which adopt a multiresolution approach to texture analysis.

2.2.1 Statistical Approaches

Traditional statistical approaches to texture analysis try to capture the spatial distribution of pixel gray level values in the textured image. The simplest description of this sort can be derived from the image histogram which plots, for each intensity value, the number of pixels in the image which have that value. Once the histogram has been computed one can calculate various first-order statistics including, but not restricted to, the mean, variance, energy and entropy of the image. The histogram approach has the advantage of being easy to compute as it is based on statistical information from single pixels. However it suffers from the drawback that it assumes independence between pixels in the image. It is therefore not a very appropriate method for extracting textural information from the image as it ignores inter-pixel relationships, which is one of the most important characteristics of texture, as discussed earlier in section 2.1.1.

Improving on this single pixel approach involves calculating second-order statistics of the image by considering interactions between pixels. Such an approach was proposed several decades ago by Haralick [41] and since then it has become probably the most popular statistical method for texture analysis. It is based on the idea of gray level co-occurrence over a local area in the image. In this method pixel pairs are defined according to their relative displacement and angular spatial relationship as shown in figure 2.4. Some examples of pixel pairs from this figure are: $(0, 0)$ and $(0, 2)$ which have a displacement of $d = 2$ and direction $\theta = 0^\circ$; $(0, 3)$ and $(1, 2)$ which have a displacement of $d = 1$ and direction $\theta = 225^\circ$; $(3, 3)$ and $(0, 3)$ which have a displacement $d = 3$ and direction $\theta = 90^\circ$. Relative frequencies of pixel pairs which are a distance d apart

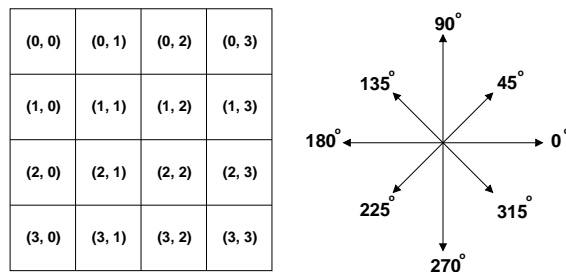


Figure 2.4: Coordinates used to reference pixels in the image and eight directions commonly used in co-occurrence matrices.

and rest along direction θ and have the same gray level values (i, j) , are denoted by $F_{d,\theta}(i, j)$ and are recorded in what is known as a co-occurrence matrix. This notion can be formalised in the following expression

$$F_{d,\theta}(i, j) = |\{ (a, b): I(a, b) = i, I(a + \delta x, b + \delta y) = j \}| \quad (2.1)$$

where (a, b) are the coordinates of a pixel in the image I , the cardinality of the set is

denoted by $|\cdot|$, and the overall displacement is given by $\delta x = d \cos \theta$ and $\delta y = d \sin \theta$. There is freedom to choose the displacement so as to obtain a symmetrical or a non-symmetrical co-occurrence matrix. Examples of these kind of matrices can be found in figure 2.5.

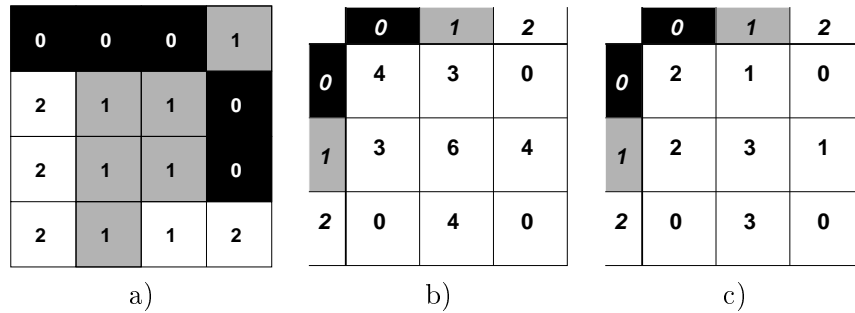


Figure 2.5: a) An example image with three gray levels; b) The symmetrical co-occurrence matrix for $d = 1$ and $\theta = 0^\circ$; c) The corresponding non-symmetrical co-occurrence matrix.

A number of useful texture features can be extracted from a co-occurrence matrix, some of which are described in table 2.1.

Uniformity of energy	$\sum_{i,j} F_{d,\theta}(i, j)$
Entropy	$-\sum_{i,j} F_{d,\theta}(i, j) \log F_{d,\theta}(i, j)$
Maximum Probability	$\max_{ij} F_{d,\theta}(i, j)$

Table 2.1: Features extracted from the co-occurrence matrix.

In a seminal texture classification study [43], Haralick *et al.* tested the usefulness of several of these features on three image data sets: photomicrographs of sandstones, aerial photographs, and satellite images. In the latter case, the addition of texture features produced a significant increase (up to 10 percent) in terrain identification rate as compared to the results obtained using just spectral features. Other classification studies have proven the strength of these texture features for texture analysis in a wide range of applications, including materials inspection [81].

Despite the success and popularity of this method, there is still a major issue surrounding its use, namely the choice of neighbourhood over which the analysis is performed. For a single (d, θ) pair, the corresponding co-occurrence matrix contains G^2 elements, where G denotes the number of gray levels present in the image. From a computational point of view, it is clearly not feasible to calculate the matrix for each

possible (d, θ) pair. Hence one needs to be able to select the correct range of spatial interactions over which to analyse the texture. No such selection process has been formally defined and the success of a particular (d, θ) pair really depends on the texture in question. If the range is small, which is usually the case, then variations in pixel intensities across a relatively small neighbourhood will be captured. However macrotexture structure in the image will not be detected by such a range.

2.2.2 Model-Based Methods

One model-based approach which is particularly popular for the task of texture analysis is the Markov Random Field (MRF) model [12, 20, 44, 45, 54, 84]. It provides us with a probabilistic model for the pixels in the image wherein the value of a given pixel is dependent only on its predefined neighbourhood [4]. The range of this dependency is captured by the order of the model which dictates the size and structure of the neighbourhood being analysed. An example of a first-order neighbourhood is shown in figure 2.6.

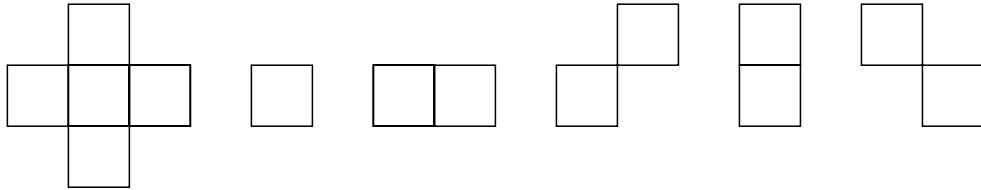


Figure 2.6: The first-order neighbourhood on the left hand side consists of the four nearest-neighbours as defined along the x and y axes. The five possible cliques types associated with this neighbourhood are shown to its right.

Also shown in this figure are the so-called cliques associated with this neighbourhood structure. Potential functions, denoted by $V_c(\cdot)$, can be computed for each clique c and used to generate an energy function

$$U(\mathbf{x}) = \sum_{\forall c} V_c(\mathbf{x}) \quad (2.2)$$

where \mathbf{x} is the lattice of sites which represents the entire image. This energy function forms the exponent of the Gibbs distribution

$$p(\mathbf{x}) = \frac{1}{Z} \exp \left\{ -\frac{1}{T} U(\mathbf{x}) \right\} \quad (2.3)$$

where T is a constant analogous to temperature and Z is the normalising constant or partition function of the system. As the energy function has the form given in

equation 2.2, this distribution provides a description of the whole image via conditional probabilities which model the local interactions of neighbouring pixels.

In practice, computational constraints mean that the size of the neighbourhood must be kept fairly small in order to avoid a combinatorial explosion of the number of states in the process. This means that MRFs are most useful for analysing microtextures whose characteristic dimension in a given direction is about the same size as that of the corresponding clique. However, certain MRF based texture models have been developed which are capable of capturing macrotexture [33, 35]. In particular, the MRF model introduced in [85] shows potential for modelling a range of different types of texture. In this method a set of filters are applied to samples of a given texture. The histograms of the resulting filtered images are extracted as texture features as they provide estimates of the marginal distributions of the underlying probability distribution which describes the texture being analysed. By invoking a maximum entropy principle, a new distribution is generated which has the same marginal distributions as those extracted via the filtering procedure. This maximum entropy distribution is therefore an estimate of the underlying texture distribution. This results in an MRF model which has a much more diverse descriptive ability. Texture synthesis experiments have been carried out by using a Gibbs sampler to draw samples from the derived distribution. The results prove this model to be very successful for a wide variety of texture types.

Other methods of this type include fractal models, which are particularly suitable for the description of textures which exhibit a degree of self-similarity at different scales [61].

2.2.3 Structural Methods

Techniques of this type try to learn the hierarchical structure present in a texture. In order to describe texture using this approach, one needs to be able to identify the primitives present in the texture and understand their geometrical structure. Placement rules which govern the position of the primitives within the texture must be learnt. It has been noted that for the case of observable or visual textures, it is usually quite difficult to extract primitives and their placement information from the image [75]. Therefore these methods are most suited to highly regular deterministic textures.

2.2.4 Spatial and Spatial-Frequency Filtering Methods

This class of texture analysis methods is very large. As a general summary one can say that they involve taking the original textured image, applying a filter to it, selecting features from the resulting filtered output, and then using these features in texture classification and/or segmentation schemes. These methods can be grouped into two categories: those which detect spatial domain features and others which aim to derive information on the frequency content of the image. Belonging to the former category are edge detection techniques which use masks such as those derived from the Roberts and Laplacian operators [77]. The second category involves techniques such as the windowed

Fourier transform. These methods also consist in the use of filter banks such as those generated by Gabor and wavelet functions.

2.3 A Multiresolution Approach to Texture Analysis

Some of the transform methods described above create a multiresolution representation of the image being analysed. This approach is of particular interest to us as it addresses the issue of scale in the image which, as was shown earlier in section 2.1.2, is central to the notion of texture. In what follows, we discuss evidence which supports such a modelling approach to images in general and to texture in particular.

2.3.1 Evidence Supporting Frequency Tuning

Understanding the human visual system's sensitivity to spatial variations in luminance has been a major focal point of psychophysical research over the last several decades [71, 8, 9, 10]. From Fourier theory we know that any signal or image can be decomposed into a number of fundamental frequencies, examples of which are shown in figure 2.7

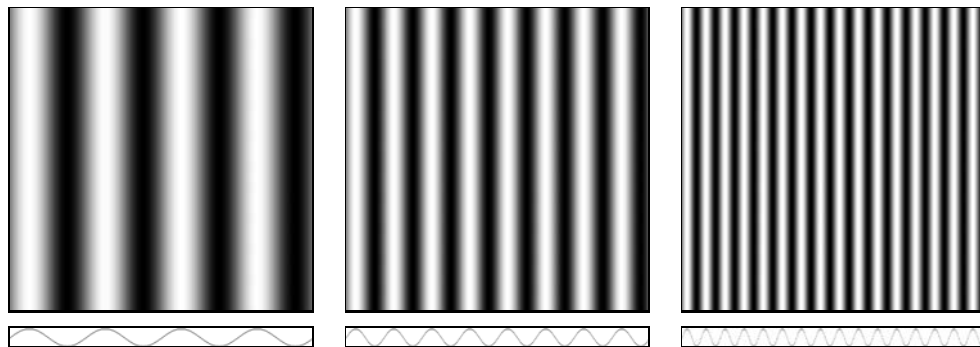


Figure 2.7: Sinusoidal gratings and their corresponding fundamental waveforms.

Visual sensitivity is measured via experiments on these basic waveforms. The notion of sensitivity to the gratings associated with these waveforms can be thought of in terms of contrast thresholds where a high level of sensitivity to a particular grating means that not much contrast is required in order to see the pattern. Covering the entire range of spatial frequencies, the minimum contrast threshold can be computed for each individual grating and its reciprocal plotted against the corresponding target frequency. This results in what is known as a *contrast sensitivity function* (CSF). In order to demonstrate the essential idea behind making such measurements we have shown in figure 2.8 an image that increases in spatial frequency in the horizontal direction and decreases in contrast along the vertical direction. By looking at this image, one can get an idea of one's own CSF. The inverted 'U' shape apparent in the image shows how the

human visual system is less sensitive to very low and very high frequencies (i.e. a very high level of contrast is needed in order for such spatial frequencies to be visible).

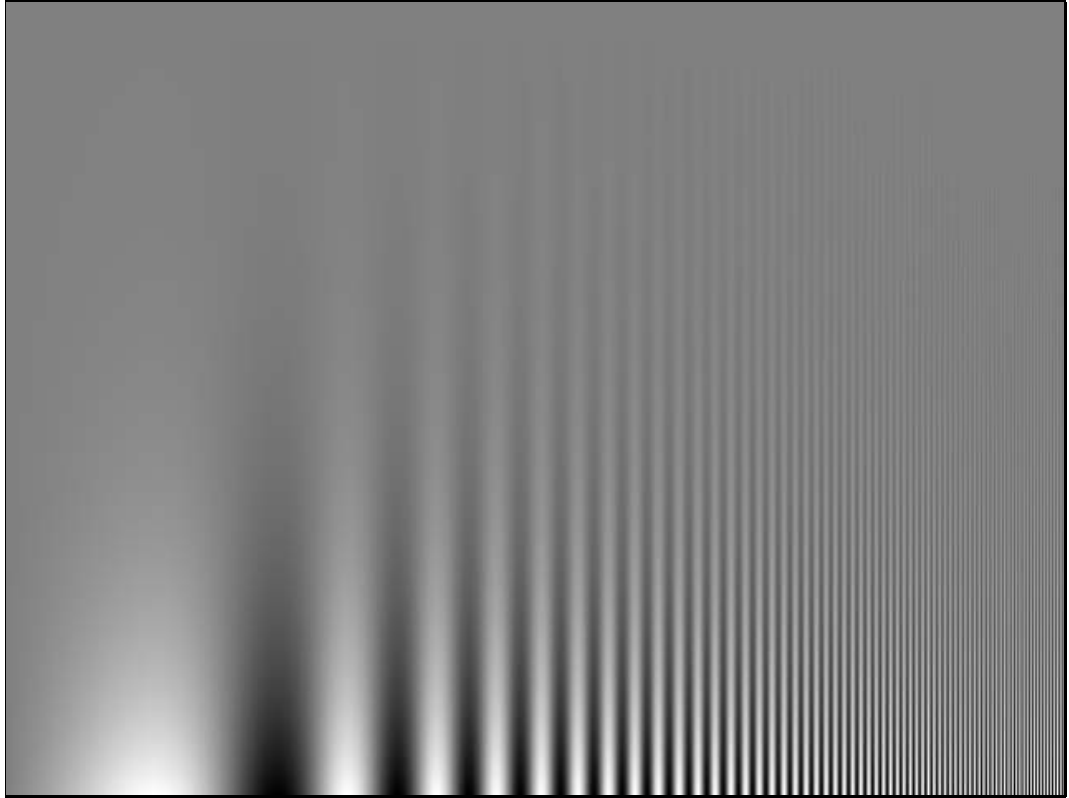


Figure 2.8: The contrast sensitivity function [63].

The earliest measurements of this kind were carried out by Schade in 1956 [71] and over the next decade or so others worked on the problem of determining the CSF of the human visual system [1, 8, 9]. The general theory at this time was that the CSF graph reflected the behaviour of a single detection mechanism in the visual system which is broadly sensitive to the entire range of spatial frequencies but which responds better to some frequencies than to others. A global linear filter was used to model this process.

A major change in this theory arose in 1968 when Campbell and Robson [10] carried out similar psychophysical experiments wherein they tested the visual sensitivity to the individual harmonic components of several different waveforms. Based on their findings they proposed the idea that early stages of visual processing do not adhere to the single detector model previously suggested, but instead decompose the retinal image into its component spatial frequencies using multiple parallel independent detection mechanisms each of which is *tuned* to a different spatial frequency range. This decomposition process is modelled using independent band-pass linear filters. Since this discovery, there have

been many attempts to measure the width of these bands and the general conclusion is that each filter has a spatial frequency range of about one octave [34, 58].

Another main result from the psychophysical community is on the sensitivity of the human visual system to the orientation of gratings. Experiments have shown a maximum sensitivity to gratings which have a directionality of 0° and 90° [7].

These psychophysical results are supported by evidence resulting from physiological experiments carried out on animals who are considered to have visual systems which are close to that of humans. The brains of both cats and monkeys have been studied in order to understand how they process gratings of different frequencies and orientations. The results of these experiments indicate that the simple cells in the visual cortex of these animals are tuned to both spatial frequency and orientation [25, 46, 76].

The experiments described above were carried out on either images of single gratings or well known waveforms which are made up of several fundamental gratings. However, it is interesting to note that for the particular case of textured images, additional tests point toward the same conclusion. Through a series of psychophysical studies on textured images Beck *et al.* [3] showed that texture segmentation, as carried out by the human visual system, was mainly a function of spatial frequency analysis rather than grouping processes which was originally considered to be the case [50, 51].

As one would expect, these studies on image interpretation have motivated many people in the computer vision community to develop systems which try to mimic this tuning behaviour of the human visual system.

2.3.2 Gabor Filters

One of the first such attempts was made by Daugman in 1980 when he modelled the selective behaviour of the cells in the visual cortex using Gabor filters [23]. In essence these are Gaussians modulated by complex sinusoids. Each filter has a set of parameters which determines the location and orientation of the area of the frequency domain which it analyses. In texture applications, banks of these filters are chosen to analyse the particular spatial frequencies and orientations present in the texture [6, 47, 80].

Although Gabor filters have been used extensively for the purpose of texture analysis, several comments have been made regarding their drawbacks. The filter parameters, for example, need to be selected by the analyst. This is an ability which undoubtedly requires a combination of intuition about the textures being examined and a lot of practice in order to master. Another issue is their computational inefficiency, especially when analysing low frequency components [52, 78]. These difficulties created a gap in the research for a more efficient and coherent multiscale approach, which was ultimately to be filled by the application of wavelets to image analysis.

2.3.3 Wavelets

The idea of wavelets has been around for a long time. These analysing functions, which have the attractive property of local support in both space and frequency, were used by Morlet in the late seventies to process geophysical data, the frequency content of which had proved too variable to be captured by analysis tools such as the windowed Fourier transform. Later, in the early eighties, the mathematics behind the wavelet transform was established [39] and since then wavelets have become a research field in their own right.

One of the most significant developments in wavelet theory was the introduction of a multiresolution analysis framework based on the wavelet transform [59]. This approach is particularly suited to the task of texture analysis as it allows the examination of texture at a variety of scales and orientations in an efficient manner which avoids the computational complexity often associated with the aforementioned Gabor filters. It also provides a sound mathematical framework by which to characterise texture. In this section we will intuitively describe the original wavelet transform and introduce some of its extensions which have been used for texture classification and segmentation over the last couple of decades. A theoretical overview of wavelet theory, complete with further references, is provided in appendix A.

2.3.3.1 The Standard Wavelet Transform

The essential idea behind a wavelet analysis is most easily communicated using a 1-dimensional example. Let us denote by f the signal to be analysed. The first level of the wavelet analysis is realised by decomposing f into its high and low frequency components through convolutions with the 1-dimensional conjugate mirror filters g and h , respectively. In the standard wavelet transform, the outputs of these two filtering operations are decimated, or downsampled, by a factor of two. This involves selecting only alternative wavelet coefficients from the outputs and hence the stored results are half the size of the original signal. The decimated output of the low-pass operation, denoted by A_1 , is a low resolution version of the original signal and so is considered to be an approximation of it. The details which are missing from this approximation are captured in the decimated output of the high-pass operation, which we denote here by D_1 . The two outputs together form a complete representation of the signal and therefore can be used to reconstruct it. The filter bank representation of this procedure is shown in figure 2.9.

To perform the next level in the analysis, the low-pass output is further decomposed using the filters h and g and the results are decimated as before. This recursive behaviour has a natural termination point: when the maximum level has been reached, its output will consist of a single pixel. However, as is often the case, an earlier cut-off level, denoted here by J , can be chosen. Figure 2.10 shows a standard wavelet decomposition down to level 3 of a signal of size 16. At each level in the decomposition the resolution of the signal approximation is coarser than that of the level before, with the difference

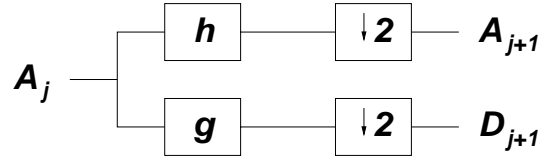


Figure 2.9: A filter bank representation of the operations carried out at level j of a 1-dimensional wavelet transform.

in information between the two being captured by the decimated high-pass operation at that level. In this manner, wavelets perform a multiresolution analysis of the signal.

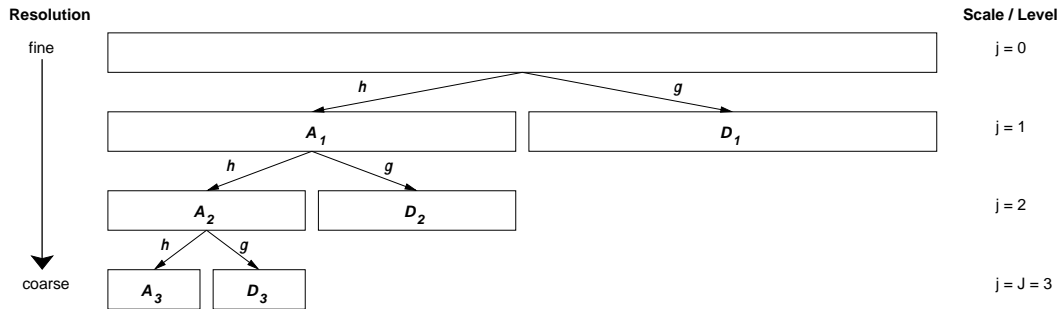


Figure 2.10: A level 3 wavelet decomposition of a 1-dimensional signal.

For the case of an image, the 2-dimensional wavelet analysis is usually carried out in a separable manner by convolving the 1-dimensional conjugate mirror filters first with the rows and then with the columns of the image, with each convolution being followed by a downsampling step. Figure 2.11 gives the filter bank representation of this sequence of convolutions.

For the first level of the decomposition, this results in four sub-images, or *subbands*: one low resolution approximation image, denoted by A_1 , and three detail images, denoted by D_1^1 , D_1^2 , and D_1^3 . These subbands are obtained via the following combinations of filters: a low-pass on rows, followed by a low-pass on columns; a low-pass on rows, followed by a high-pass on columns; a high-pass on rows, followed by a low-pass on columns; a high-pass on rows, followed by a high-pass on columns. At each level in the decomposition, the analysis is carried out on the approximation image of the previous level. The traditional layout of these subbands is shown in figure 2.12. The three detail images at each level contain the vertical, horizontal, and diagonal details of the image at that level. From this it is clear that the wavelet transform performs an analysis of the image not only at different scales, but also at different orientations.

As the approximation subband, A_j , is derived via a recursive low-pass filtering

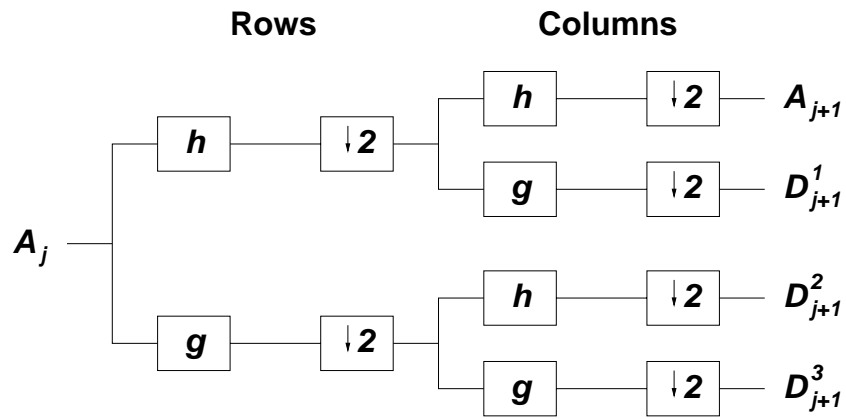


Figure 2.11: A filter bank representation of the operations carried out at level j of a 2-dimensional wavelet transform.

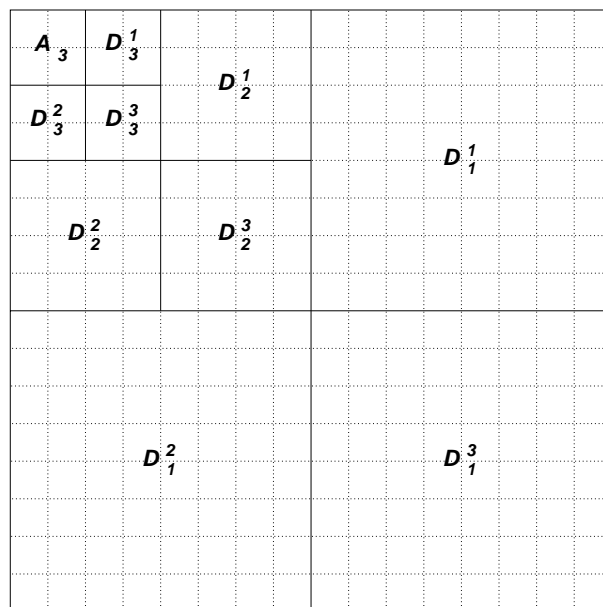


Figure 2.12: A level 3 standard wavelet analysis of an image.

technique, many regard it as devoid of any important textural information and so it is often not considered in the texture analysis process. For this reason, it is quite common to see only the detail subbands used to generate texture features.

One of the most frequently used measures for wavelet texture analysis is the mean energy which, for a fixed subband α , is given by

$$E_\alpha = \frac{1}{N_\alpha} \sum_{i=0}^{N_\alpha} \omega_{\alpha,i}^2 \quad (2.4)$$

where $\omega_{\alpha,i}$ is the i^{th} wavelet coefficient in subband α and N_α is the total number of wavelet coefficients in subband α .

As a detail subband is derived via a sequence of convolutions which involve the high-pass filter g , we can say that on average, if one is to consider this sequence applied to all translated versions of the input signal, the mean wavelet coefficient of the detail subband will be zero. In this case, the mean energy of the subband is an estimate of the variance of the individual wavelet coefficients in the subband. From a modelling viewpoint, the use of mean energy measures can therefore be compared with modelling the histogram of the wavelet coefficients in each detail subband by a Gaussian distribution with zero mean.

Although the mean energy can be a useful feature for texture classification [72], it has been suggested that in certain cases a different approach to modelling the wavelet coefficients may be more appropriate. Having carried out experiments on natural textures, Mallat [59] suggested that the wavelet coefficients in the detail subbands are best modelled by a generalised Gaussian distribution (GGD):

$$p(x; \alpha, \beta) = \frac{\beta}{2 \alpha \Gamma\left(\frac{1}{\beta}\right)} \exp \left\{ - \left(\frac{|x|}{\alpha} \right)^\beta \right\} \quad (2.5)$$

where $\Gamma(\cdot)$ is the gamma function. Here α describes the width of the distribution peak and β controls its fall-off rate. This approach has been adopted in [24] wherein the two parameters are calculated for each detail subband. These parameters are then used as texture features in a classification experiment wherein each feature vector is assigned to a class according to the k -nearest neighbour rule [28] with a variance-normalised Euclidean distance. The results of this experiment show how these features outperform the traditional mean energy features described above. Since then, others have tested this GGD modelling approach. In a recent application to retrieval from databases containing textured images this approach showed an improvement over the traditional energy-based measures [27]. In this same work, the Kullback-Leibler distance proved to be a more accurate similarity measurement than the normalised Euclidean distance.

In addition to using the parameters of a GGD as texture features, the idea of calculating the second order statistics of the detail subbands was also proposed in [24]. This

was done by computing the co-occurrence matrix, previously described in section 2.2.1, for each detail subband and extracting several features from it including those shown in table 2.1. Although these second-order statistics performed better than the parameters of the GGD model for the classification of certain types of textures, the final conclusion of their analysis was that the lowest error rates can be achieved using a combination of the two feature types.

2.3.4 Wavelet Frames

An important point to note about the type of analysis described above is its lack of translation invariance. As each level in the decomposition consists of a downsampling step, it is clear that even a slight shift in the original input signal can result in a modified representation. This is particularly problematic for applications involving the segmentation of images which contain multiple textures. As a texture can appear anywhere in the scene, features which describe spatial frequencies and orientations generated from a single standard wavelet decomposition of a training patch may or may not correlate with the features derived from that texture in the scene. Such uncertainty in a pattern recognition system is highly undesirable.

One way of trying to avoid this drawback of the standard wavelet transform is to eliminate the downsampling step altogether and retain the full output of the filtering operations at each level. This is the essential idea behind the technique known as wavelet frames. It results in an undecimated version of the wavelet transform and so provides an over-complete representation of the original image. This technique was first tested in [78] and the results showed an improvement over the use of the regular decimated wavelet transform coefficients as texture features. Similar observations have been made by other researchers [65, 66].

Although wavelet frames provide a translation invariant description of texture, it should be noted that they provide a highly redundant representation of the original signal and hence a significant amount of information is duplicated. They are therefore perhaps not the most ideal analysis tool for all applications, especially those in which computational complexity is an issue.

2.3.5 Wavelet-Domain Tree Models

The texture analysis methods discussed above in section 2.3.3.1 can be considered as independent wavelet models. They do not capture any form of interaction between the wavelet coefficients. One possible way to introduce a dependency between coefficients is through the use of Hidden Markov Tree (HMT) models [21]. HMTs model the dependencies between wavelet coefficients at different levels in the standard wavelet decomposition. These models have been used for the purpose of texture segmentation within both supervised [13] and unsupervised frameworks [73]. Although these methods have produced good results in texture analysis applications, one must ask the question

as to whether interscale dependency is really the correct thing to model in the case of textures. The general nature of texture seems to suggest that capturing the intrascale dependencies between coefficients would be a more appropriate thing to do. In the next section we introduce wavelet packet models which are capable of describing such intrascale relationships.

2.3.6 Wavelet Packets

The wavelet-based approaches discussed so far are all founded on the same recursive technique: further decomposition of the low-pass output at each level. Looking at the Fourier representation of many textures however indicates that this specific type of recursion may not be appropriate. Figure 2.13 shows several textures along with a log display of the magnitude of their Fourier coefficients. Some of these textures come from the Brodatz album, while others have been cropped from high resolution satellite images. It is clear that not all of these textures have their information concentrated in the low frequencies. On the contrary, many of them possess significant contributions from the intermediate and high frequency ranges. Performing a standard wavelet type analysis of such textures does not focus on these areas of activity in the frequency domain. In a response to this issue people have adopted the use of wavelet packets as a means of describing texture more accurately.

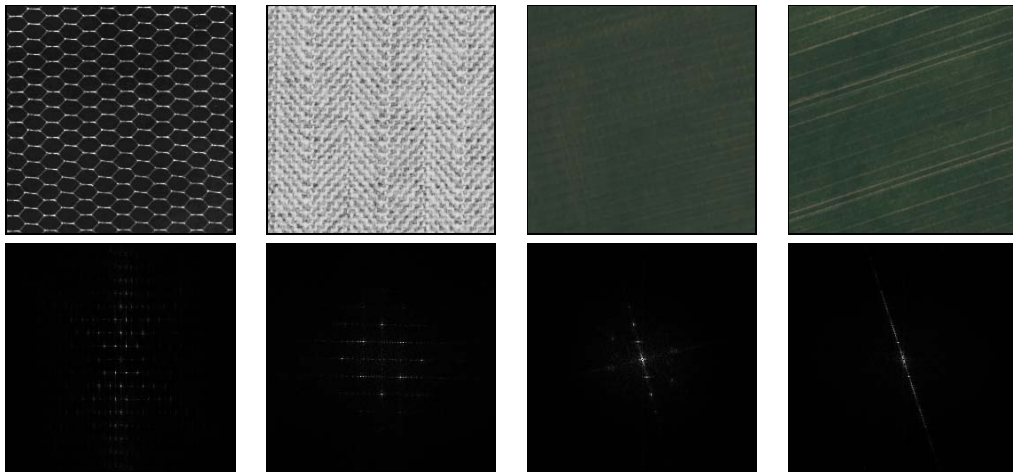


Figure 2.13: Top row: Two natural textures (D34 and Herring) taken from the Brodatz album and two texture patches cropped from a 1 metre resolution IKONOS satellite image. Bottom row: The corresponding log display of the magnitude of their Fourier coefficients.

Wavelet packets are a natural extension to the standard wavelet technique. They permit a more in-depth analysis of the frequencies present in the image [82]. At each level in a wavelet packet decomposition, the decimated output of both the high-pass and

the low-pass operations are further decomposed. Representations of this decomposition down to level 3 are shown in figure 2.14 for the case of a 1-dimensional signal.

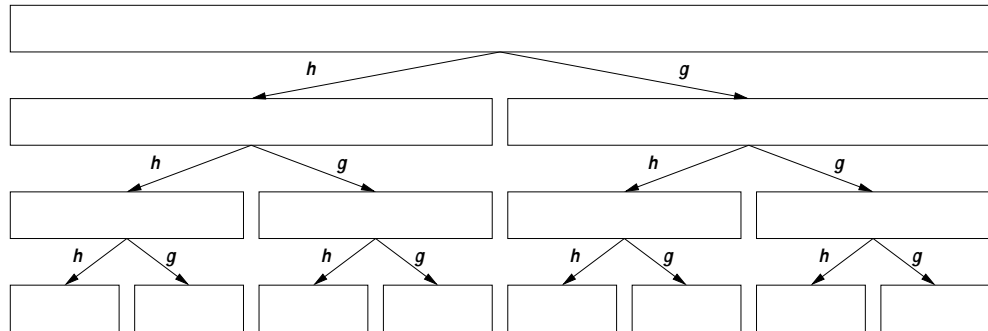


Figure 2.14: A level 3 wavelet packet decomposition of a 1-dimensional signal.

In [52], wavelet packets were tested in a texture classification experiment on 25 natural textures. Measures of entropy and energy were computed for each subband in a fixed level wavelet packet decomposition. Classification results were compared to those obtained using the same features extracted from a standard wavelet decomposition. A slight improvement in classification accuracy was reported.

An extension to a wavelet packet frame technique, analogous to that described in section 2.3.4, was introduced in [53]. Features extracted from both wavelet frame and wavelet packet frame decompositions were tested in a segmentation experiment on two-texture mosaics using a simple K-means clustering algorithm. The packet version of the decomposition was shown to outperform the regular wavelet frame technique.

2.3.7 Adaptive Wavelet Packets

Although wavelet packets provide a much more detailed analysis of the frequency content of a texture, it is often the case that areas which contain little or no frequency information are recursively decomposed. Take for example the Bark texture from the Brodatz album which is shown in figure 2.15. Due to the vertical directionality present in this texture, its Fourier representation shows a high concentration of intensity along the horizontal axis. Ideally we would like to be able to analyse this texture in and around these high intensity areas and in doing so only decompose the frequency area which is active. In order to be able to do this we need to, in some way, select which subbands need further decomposition and which should be left untouched. This is where the notion of adaptivity emerges. The first such adaptive selection technique was proposed in [16]. The motivation behind this development was not that of accurate texture description, but instead the efficient compression of music and images. In this method an entropy criterion is used to determine the *best basis* to represent the original signal from a collection of possible wavelet packet bases. For a given subband α in the

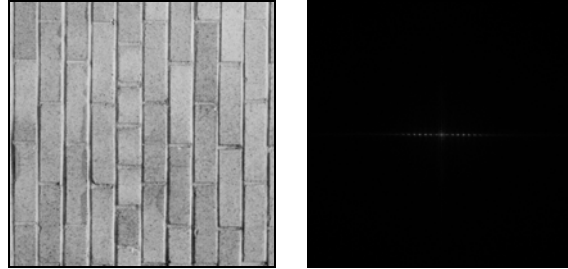


Figure 2.15: The Bark texture from the Brodatz album, along with a log display of the magnitude of its Fourier coefficients.

decomposition, the entropy is given by

$$H_\alpha = - \sum_{i=0}^{N_\alpha} \bar{e}_i \log \bar{e}_i \quad (2.6)$$

where $\bar{e}_{\alpha,i}$ is the normalised energy of the i^{th} wavelet coefficient in subband α and N_α is the total number of wavelet coefficients in subband α .

A full wavelet packet analysis is performed down to a specified level and the entropy of each subband in the decomposition is calculated. The sum of the entropies of the subbands at the bottom level, *i.e.* the leaf nodes of the tree shown in figure 2.14, is the quantity which the method attempts to minimise. This minimisation is carried out as follows: starting at the bottom level, the entropy of the parent of each pair of subbands is compared to the sum of the entropy of its children. If the parent subband has a lower entropy value then its children are removed and the parent subband remains as the best representation of the signal in that frequency range. Applying this to the whole tree yields an adaptive packet decomposition such as the one shown in figure 2.16.

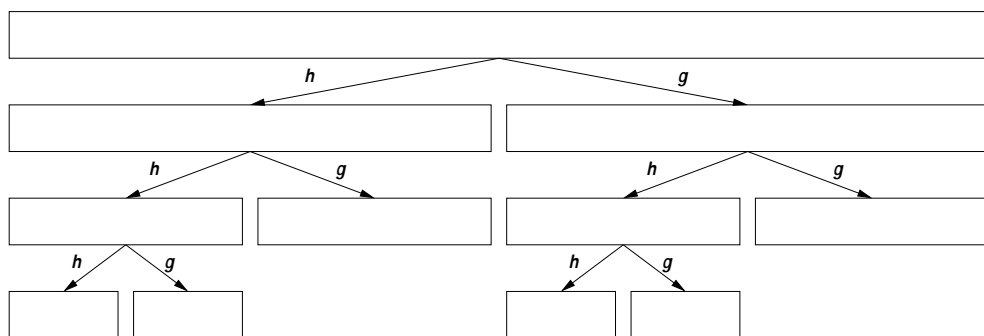


Figure 2.16: A 1D adaptive wavelet packet analysis of a signal.

The 2-dimensional version of this best basis representation is given in figure 2.17.

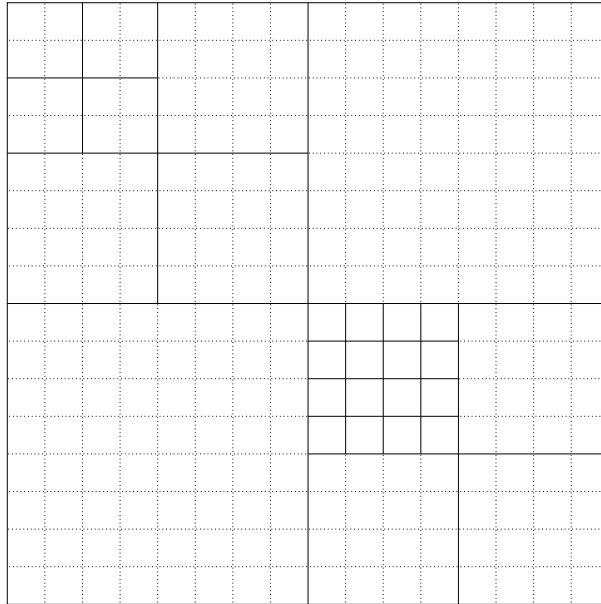


Figure 2.17: A 2D adaptive wavelet packet analysis of an image.

This idea of adapting the wavelet packet basis according to a certain criterion was readily adopted by the texture analysis community. The first results on texture classification using an adaptive wavelet packet decomposition were reported in [11]. In this paper, the averaged l_1 -norm

$$e_\alpha = \frac{1}{N_\alpha} \sum_{i=0}^{N_\alpha} |\omega_{\alpha,i}| \quad (2.7)$$

was examined for its use as a criterion to decide whether or not a subband α should be further decomposed. As before, $\omega_{\alpha,i}$ is the i^{th} wavelet coefficient in subband α and N_α is the total number of wavelet coefficients in subband α . Unlike the bottom-up approach presented in [16], in [11] the decomposition decision rule was carried out in a top-down manner. Starting with the first level of the decomposition, the above measure is computed for each subband on this level. If a subband has a relatively low value compared to the others, then this subband is not decomposed any further. Making this decision involves comparing the value of each subband to the maximum subband value at that level, which is denoted by e_{\max} . The rule is given by:

if $e_\alpha < C e_{\max}$
then do not decompose subband α .

The constant $C < 1$ must be chosen by the user. In [11], a value of $C = 0.3$ was used. If the value of the subband is greater than $C e_{\max}$, then it is decomposed and

the same procedure is carried out on its child nodes. The decomposition is stopped when the smallest subbands are of size 16×16 . For each texture in the training set, an average adaptive wavelet packet representation is generated in this manner using sample patches of the texture. For a new texture patch, its best wavelet packet representation is computed and the first J dominant subbands (*i.e.* those with the largest e_α values) are used as a feature set for that texture. For each texture in the database, this feature set is compared to the same subbands in the trained representations. The new texture patch is assigned to the class which has the shortest distance between values of the dominant subbands. A variety of distance measures were tested, including the Euclidean and Mahalanobis distance. This method was compared to several other filtering methods including Gabor filters and the standard wavelet transform and it showed improvements on these methods in many cases. This technique has been used for other texture applications, including texture classification of SAR images [31].

The general concept of adapting the wavelet packet transform to the underlying structure in the texture is an attractive one. A popular subtopic of this general research area is the idea of choosing the subbands from a fixed level full wavelet packet decomposition which provide maximum class separability between textures. Several techniques have been proposed to do this [5, 29, 30, 70].

2.4 Our Model

In the previous section, we outlined the development of multiresolution wavelet-based approaches to texture analysis. Extensions to the standard wavelet transform, which are motivated by the need for accurate and efficient texture descriptors, have followed a natural and obvious direction. These adaptive methods have ultimately evolved into searches for the minimum number of subbands which provide a maximally descriptive representation of the texture in question.

However intuitively appealing these methods may be, they appear to have an underlying weakness. This is highlighted by the lack of consistency in the choice of measure extracted from subbands as a means to either represent texture or to decide if further decomposition is needed. Numerous criteria have been proposed for these tasks: entropy, energy, mean deviation, number of coefficients in a subband whose absolute value exceeds a certain threshold, cooccurrence matrices, etc. The question of which features are best for the purposes of texture description has already been raised [56], yet it remains an open issue. There does not seem to be any real substantial evidence supporting the use of one particular measure over another.

In an attempt to avoid this seemingly ad-hoc decision making process, we propose to tackle the problem of texture analysis from a probabilistic viewpoint. Once a distribution has been fixed for a given type of texture, dynamically deriving an adaptive wavelet packet basis which best describes this texture becomes a relatively straightforward procedure. There is no issue surrounding the most appropriate measure to use,

because decisions which are made at each point in the decomposition are based on the underlying texture distribution. Therefore the adaptive decomposition which results is best in a probabilistic sense. The theory behind this approach is presented in the next chapter.

Chapter 3

Theoretical Framework

Before trying to model texture, it is necessary to first understand how it appears in the images we process. An obvious starting point is to note that texture always appears in a finite region of the image being analysed. Let us denote this finite textured region by R . It can happen that the entire image being analysed consists of only one texture and so R will be the size of the image and will, more than likely, have a very regular boundary corresponding to that of the image itself. This is usually the case when texture patches are selected for training and testing within a texture classification scheme. The other possibility is that R only makes up part of the image being analysed and it may have either a regular or irregular shaped boundary. This is the case in applications which require a scene to be segmented into textured regions. Examples of textured regions are shown in figure 3.1.

From this it is clear that modelling texture in a probabilistic manner means deriving a probability measure for the texture on a finite region R . So for a given texture, which we denote here by l , one needs to specify the following distribution

$$\Pr(\phi_R \mid \lambda_{R_l}, \mathcal{K}_l) \tag{3.1}$$

where ϕ_R denotes the image ϕ restricted to the finite region R ; the class map λ_{R_l} represents the action of assigning to each pixel in ϕ_R a label which identifies the pixel as belonging to texture l ; and the set \mathcal{K}_l contains information regarding the texture l , *i.e.* the parameters of the probability distribution. The usefulness of this quantity for image segmentation, rather than texture description, depends on a number of assumptions which we will discuss further in chapter 5.

Attempting to derive such a distribution immediately raises an important question: should the description of a texture explicitly depend on the region of the image in which it appears? As discussed above, the kind of textures we wish to analyse generally exhibit great variability in both size and shape. Let us consider, for example, remote sensing images of agricultural areas. The texture corresponding to a ploughed field can cover the majority of the image and have an irregular border or it might be confined to a small square shaped section of the image. In modelling such textures, we therefore do



Figure 3.1: Examples of images cropped from high resolution aerial images provided by the French Mapping Institute (IGN). For each image, the textured region of interest, R , is outlined in red.

not have any *a priori* information regarding how much of the texture will be present in the image being analysed and what form it will have. The essential notion of what makes a region textured is therefore not explicitly related to the size and shape of the region.

Therefore, although equation 3.1 defines a probability measure for texture on a finite region R , it is important to realise that this expression is not an explicit function of the shape and size of R . By this we mean that the probability on the region is derived by marginalising the probability distribution on an infinite image. We therefore start our modelling procedure by treating images of these textures as functions on an infinite (or at least very large) domain, which we denote by D_∞ . We use a probability distribution on the space of such images which will describe the variations possible for images of the texture. We denote such a distribution by

$$\Pr(\phi \mid \lambda \equiv l, \mathcal{K}_l). \quad (3.2)$$

Here $\phi: D_\infty \rightarrow C$ represents the infinite image, which can be thought of as a map from the infinite image domain D_∞ to the image co-domain $C = \mathbb{R}$ or \mathbb{Z} . The set of all possible textures is denoted by \mathcal{L} . The class map, $\lambda: D_\infty \rightarrow \mathcal{L}$, in the above distribution takes every pixel in the infinite image to texture l . The set \mathcal{K}_l , as before, represents the knowledge we have on the texture l .

Deriving a distribution on a finite region $R \subset D_\infty$ in a coherent manner requires that we marginalise equation (3.2) over the values of the pixels outside this region. An outline of the abstract details of this operation is given below.

Let Φ be the space of infinite images. As introduced above, R defines a finite region. The complement of this region is denoted by \bar{R} . These two regions give rise to Φ_R and $\Phi_{\bar{R}}$, the space of images defined on R and \bar{R} , respectively. Figure 3.2 provides a visual representation of the maps that exist between Φ and these two subspaces. The map π_R

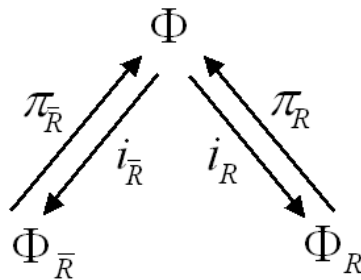


Figure 3.2: Four maps defined on the space of infinite images, Φ .

restricts an infinite image in Φ to the region R . This map is surjective (or onto) because two infinite images which have the same value inside the region, but have different values

outside the region will both be mapped to the same finite image defined on R . Similarly, $\pi_{\bar{R}}$ is a surjective map which restricts an infinite image in Φ to \bar{R} . The map i_R takes an image defined on the region and maps it back to Φ , setting pixels in the infinite image which lie outside the region to zero. This map is injective (or one-to-one) because for each image defined on the finite region R there is one and only one corresponding infinite image. Similarly, $i_{\bar{R}}$ is an injective map which takes an image defined on \bar{R} and maps it back to Φ , setting the values of all the pixels inside the region to zero. These four maps together allow us to define two projections P_R and $P_{\bar{R}}$ as follows:

$$\begin{aligned} P_R &= i_R \pi_R \\ P_{\bar{R}} &= i_{\bar{R}} \pi_{\bar{R}}. \end{aligned} \quad (3.3)$$

Together these projections form an orthogonal decomposition of Φ such that an infinite image may be expressed as

$$\phi = P_R(\phi) + P_{\bar{R}}(\phi). \quad (3.4)$$

Marginalising the distribution in equation (3.2) over all pixels outside the region (*i.e.* over $\phi_{\bar{R}} \in \Phi_{\bar{R}}$), will result in a probability measure for an image on the finite region R :

$$\Pr(\phi_R | \cdot) = \int_{\phi_{\bar{R}}} \Pr(\phi | \cdot) \quad (3.5)$$

which, in principle at least, solves the boundary problem for texture.

3.1 A Gaussian Distribution for Texture

The next step in our analysis is to choose a specific distribution to model texture. As the main objective of this work is to develop and test a probabilistic framework for adaptive texture description, we restrict ourselves, for the sake of simplicity, to the choice of a Gaussian distribution on the space of images. In this section we will work through the above procedure using a Gaussian as the specific distribution on Φ . In abstract notation, a Gaussian distribution can be expressed as

$$\Pr(\phi | \cdot) = \left| \frac{F}{\pi} \right|^{\frac{1}{2}} \exp \{ -\langle \phi - \mu | F | \phi - \mu \rangle \} D\phi \quad (3.6)$$

where $\langle I | J \rangle$ is the inner product of the functions $|I\rangle$ and $|J\rangle$ in the space of images; $F|\phi\rangle$ denotes the operator F acting on ϕ ; μ is the mean of the distribution; F is the inverse variance-covariance operator of the distribution (with a factor of 1/2 implicit in its representation); $|F|$ denotes the determinant of this operator; $D\phi$ is the underlying measure on the space Φ and represents the infinite product $\prod d\phi(x)$ over all points $x \in D_\infty$, the infinite image domain.

This general expression of the distribution is independent of the coordinate system. It can be rewritten in terms of a particular basis depending on the analysis being carried out. In the position basis, for example, the distribution takes the form

$$\Pr(\phi | \cdot) = \left| \frac{F}{\pi} \right|^{\frac{1}{2}} \exp \left\{ - \sum_{(x,x') \in D_\infty} (\phi(x) - \mu(x)) F(x, x') (\phi(x') - \mu(x')) \right\} D\phi \quad (3.7)$$

where $F(x, x')$ expresses the correlation between the values of ϕ evaluated at the point x and x' . The operator F is diagonal, *i.e.* $F(x, x') = f(x) \delta(x, x')$, in the position basis only if the values of the pixels in the image are independent of each other.

Once a finite region of interest R is specified for analysis, the operator F can be split up into four parts according to the orthogonal decomposition of Φ by the projections in equation (3.3). The operators are defined as follows:

$$\begin{pmatrix} F_{RR} = \pi_R F i_R & F_{R\bar{R}} = \pi_R F i_{\bar{R}} \\ F_{\bar{R}R} = \pi_{\bar{R}} F i_R & F_{\bar{R}\bar{R}} = \pi_{\bar{R}} F i_{\bar{R}} \end{pmatrix} \quad (3.8)$$

The intuitive meaning behind these operators can be seen in figure 3.3: F_{RR} relates pixels inside the region to each other, $F_{R\bar{R}}$ relates pixels outside the region to pixels inside the region, $F_{\bar{R}R}$ relates pixels inside the region to pixels outside the region, and $F_{\bar{R}\bar{R}}$ relates pixels outside the region to each other.

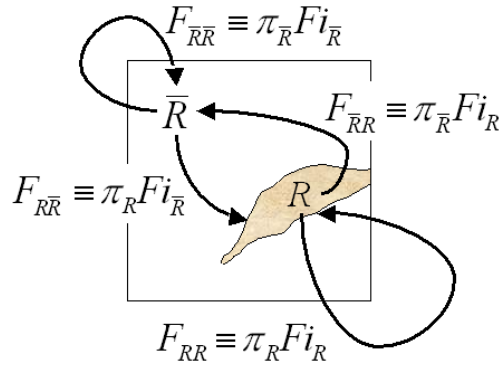


Figure 3.3: Splitting the operator F up according to the projections P_R and $P_{\bar{R}}$.

In performing the calculations needed to arrive at a probability measure for a texture on a finite region R , we assume here, without loss of generality¹, that the Gaussian

¹The assumption of a zero mean is made for the sake of clarity in the computations which follow. If it were to be included, the mean μ of the infinite image would split up into a mean μ_R on the region and a mean $\mu_{\bar{R}}$ on the complement of the region.

distribution on Φ has zero mean and so the probability measure takes on the following form:

$$\Pr(\phi | \cdot) = \left| \frac{F}{\pi} \right|^{\frac{1}{2}} \exp \{ -\langle \phi | F | \phi \rangle \} D\phi. \quad (3.9)$$

3.1.1 Expressing the Energy in Terms of ϕ_R and $\phi_{\bar{R}}$

We define the energy of the distribution on the space Φ to be

$$E(\phi) = \langle \phi | F | \phi \rangle. \quad (3.10)$$

At the beginning of this chapter, we introduced two projections, P_R and $P_{\bar{R}}$, which form an orthogonal decomposition of Φ . With these in mind we can rewrite the energy as

$$E(\phi) = \langle P_R\phi + P_{\bar{R}}\phi | F | P_R\phi + P_{\bar{R}}\phi \rangle \quad (3.11)$$

which when expanded yields

$$\begin{aligned} E(\phi) &= \langle P_R\phi | F | P_R\phi \rangle + \langle P_R\phi | F | P_{\bar{R}}\phi \rangle + \\ &\quad \langle P_{\bar{R}}\phi | F | P_R\phi \rangle + \langle P_{\bar{R}}\phi | F | P_{\bar{R}}\phi \rangle. \end{aligned} \quad (3.12)$$

Expressing these two projections in terms of their defining injective and surjective maps on Φ , as in equation (3.3), allows us to write the energy as

$$\begin{aligned} E(\phi) &= \langle i_R\pi_R\phi | F | i_R\pi_R\phi \rangle + \langle i_R\pi_R\phi | F | i_{\bar{R}}\pi_{\bar{R}}\phi \rangle + \\ &\quad \langle i_{\bar{R}}\pi_{\bar{R}}\phi | F | i_R\pi_R\phi \rangle + \langle i_{\bar{R}}\pi_{\bar{R}}\phi | F | i_{\bar{R}}\pi_{\bar{R}}\phi \rangle \\ &= \langle i_R\pi_R\phi | F i_R\pi_R\phi \rangle + \langle i_R\pi_R\phi | F i_{\bar{R}}\pi_{\bar{R}}\phi \rangle + \\ &\quad \langle i_{\bar{R}}\pi_{\bar{R}}\phi | F i_R\pi_R\phi \rangle + \langle i_{\bar{R}}\pi_{\bar{R}}\phi | F i_{\bar{R}}\pi_{\bar{R}}\phi \rangle \\ &= \langle \pi_R\phi | i_R^\dagger F i_R\pi_R\phi \rangle + \langle \pi_R\phi | i_R^\dagger F i_{\bar{R}}\pi_{\bar{R}}\phi \rangle + \\ &\quad \langle \pi_{\bar{R}}\phi | i_{\bar{R}}^\dagger F i_R\pi_R\phi \rangle + \langle \pi_{\bar{R}}\phi | i_{\bar{R}}^\dagger F i_{\bar{R}}\pi_{\bar{R}}\phi \rangle \end{aligned} \quad (3.13)$$

where $i_R^\dagger = \pi_R$ and $i_{\bar{R}}^\dagger = \pi_{\bar{R}}$ are the adjoints of i_R and $i_{\bar{R}}$, respectively. The energy thus becomes

$$\begin{aligned} E(\phi) &= \langle \pi_R\phi | \pi_R F i_R\pi_R\phi \rangle + \langle \pi_R\phi | \pi_R F i_{\bar{R}}\pi_{\bar{R}}\phi \rangle + \\ &\quad \langle \pi_{\bar{R}}\phi | \pi_{\bar{R}} F i_R\pi_R\phi \rangle + \langle \pi_{\bar{R}}\phi | \pi_{\bar{R}} F i_{\bar{R}}\pi_{\bar{R}}\phi \rangle \\ &= \langle \pi_R\phi | \pi_R F i_R | \pi_R\phi \rangle + \langle \pi_R\phi | \pi_R F i_{\bar{R}} | \pi_{\bar{R}}\phi \rangle + \\ &\quad \langle \pi_{\bar{R}}\phi | \pi_{\bar{R}} F i_R | \pi_R\phi \rangle + \langle \pi_{\bar{R}}\phi | \pi_{\bar{R}} F i_{\bar{R}} | \pi_{\bar{R}}\phi \rangle. \end{aligned} \quad (3.14)$$

Note here that $\pi_R Fi_R$, $\pi_R Fi_{\bar{R}}$, $\pi_{\bar{R}} Fi_R$ and $\pi_{\bar{R}} Fi_{\bar{R}}$ in equation (3.14), are the four operators of figure 3.3 obtained by splitting F up according to the orthogonal decomposition formed by the projections P_R and $P_{\bar{R}}$. These operators therefore enter into the energy as follows

$$E(\phi) = \langle \pi_R \phi | F_{RR} | \pi_R \phi \rangle + \langle \pi_R \phi | F_{R\bar{R}} | \pi_{\bar{R}} \phi \rangle + \langle \pi_{\bar{R}} \phi | F_{\bar{R}R} | \pi_R \phi \rangle + \langle \pi_{\bar{R}} \phi | F_{\bar{R}\bar{R}} | \pi_{\bar{R}} \phi \rangle. \quad (3.15)$$

Using the definition of the surjective maps, π_R and $\pi_{\bar{R}}$, we can further reduce the expression for the energy to

$$E(\phi) = \langle \phi_R | F_{RR} | \phi_R \rangle + \langle \phi_R | F_{R\bar{R}} | \phi_{\bar{R}} \rangle + \langle \phi_{\bar{R}} | F_{\bar{R}R} | \phi_R \rangle + \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle. \quad (3.16)$$

As our goal is to obtain an expression for the energy in terms of ϕ_R and $\phi_{\bar{R}}$, we must manipulate equation (3.16) to separate out completely the action on R from the action on \bar{R} . To do this we must complete the square in $\phi_{\bar{R}}$ which yields the following expression for the energy

$$E(\phi) = \langle \phi_{\bar{R}} + F_{\bar{R}\bar{R}}^{-1} F_{\bar{R}R} \phi_R | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} \phi_R \rangle - \langle \phi_R | F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} | \phi_R \rangle + \langle \phi_R | F_{RR} | \phi_R \rangle. \quad (3.17)$$

Details of this operation are provided in appendix B. Setting

$$(F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} \phi_R = \zeta_{\bar{R}}$$

allows us to finally write our energy as desired, in terms of ϕ_R and $\phi_{\bar{R}}$ separately:

$$E(\phi) = \langle \phi_{\bar{R}} + \zeta_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + \zeta_{\bar{R}} \rangle + \langle \phi_R | F_{RR} - F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} | \phi_R \rangle. \quad (3.18)$$

3.1.2 Marginalising over Elements in $\phi_{\bar{R}}$

Now that we have an expression for the energy of our distribution in terms of ϕ_R and $\phi_{\bar{R}}$ separately, we can write our distribution on the space of infinite images as:

$$\Pr(\phi | \cdot) = \left| \frac{F}{\pi} \right|^{\frac{1}{2}} e^{-\langle \phi_{\bar{R}} + \zeta_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + \zeta_{\bar{R}} \rangle - \langle \phi_R | F_{RR} - F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} | \phi_R \rangle} D\phi_R D\phi_{\bar{R}}. \quad (3.19)$$

To obtain a measure on the region R one must marginalise over all points in $\phi_{\bar{R}}$ as follows:

$$\Pr(\phi_R | \cdot) = \int_{-\infty}^{+\infty} \exp \{ -\langle \phi_{\bar{R}} + \zeta_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + \zeta_{\bar{R}} \rangle \} D\phi_{\bar{R}} * \left| \frac{F}{\pi} \right|^{\frac{1}{2}} \exp \{ -\langle \phi_R | F_{RR} - F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} | \phi_R \rangle \} D\phi_R. \quad (3.20)$$

Making the following change of variable $\phi'_{\bar{R}} = \phi_{\bar{R}} + \zeta_{\bar{R}}$ we get that

$$\begin{aligned} \Pr(\phi_R | \cdot) &= \int_{-\infty}^{+\infty} \exp \{ -\langle \phi'_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi'_{\bar{R}} \rangle \} D\phi'_{\bar{R}} * \\ &\quad \left| \frac{F}{\pi} \right|^{\frac{1}{2}} \exp \left\{ -\langle \phi_R | F_{RR} - F_{R\bar{R}}(F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R} | \phi_R \rangle \right\} D\phi_R. \end{aligned} \quad (3.21)$$

Here we note that

$$\int_{-\infty}^{+\infty} \exp \{ -\langle \phi'_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi'_{\bar{R}} \rangle \} D\phi'_{\bar{R}} \quad (3.22)$$

is a non-normalized Gaussian distribution, the correct normalization constant of which is given by

$$\left| \frac{F_{\bar{R}\bar{R}}}{\pi} \right|^{-\frac{1}{2}}.$$

So the probability measure on the region becomes:

$$\Pr(\phi_R | \cdot) = \frac{|F|^{\frac{1}{2}}}{|F_{\bar{R}\bar{R}}|^{\frac{1}{2}}} \exp \left\{ -\langle \phi_R | F_{RR} - F_{R\bar{R}}(F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R} | \phi_R \rangle \right\} D\phi_R. \quad (3.23)$$

For notational simplicity we shall denote the operator on the region R by G_R :

$$G_R = F_{RR} - F_{R\bar{R}}(F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R} \quad (3.24)$$

and so the probability distribution for images defined on the finite region R is given by:

$$\Pr(\phi_R | \cdot) = Z^{-1} \exp \{ -\langle \phi_R | G_R | \phi_R \rangle \} D\phi_R$$

Reintroducing the mean of the region gives us the following probability measure for the image ϕ restricted to the finite region R :

$$\Pr(\phi_R | \cdot) = Z^{-1} \exp \{ -\langle \phi_R - \mu_R | G_R | \phi_R - \mu_R \rangle \} \quad (3.25)$$

where $\mu_R = P_R \mu$ represents the mean of the infinite image ϕ restricted to the region, and the underlying measure on the region is assumed implicit in this expression.

3.1.3 Diagonalisation of the Operator G_R

In principle, we can evaluate the operator G_R and hence the exponent and probability in equation (3.25). However, in practice, the computational complexity of such a calculation requires that we diagonalise the operator G_R , *i.e.* find a set of functions $|a\rangle$ on the region R such that

$$\langle a' | G_R | a \rangle = g_R(a) \delta(a, a').$$

One possible way to diagonalise the operator G_R is to find a set of functions on the region

$$\mathcal{B} = \{ |a\rangle : a \in A \text{ an index set} \} \quad (3.26)$$

that satisfies the following two conditions:

Condition 1: The set of functions $\{i_R |a\rangle : a \in A\}$ on the infinite domain are eigenfunctions of the operator F with eigenvalues $\{f_a : a \in A\}$;

Condition 2: The set \mathcal{B} forms an orthonormal basis for functions on the region R ;

To outline the diagonalisation procedure we must start by expressing the distribution on Φ_R in terms of the elements of the set \mathcal{B} as follows:

$$\Pr(\phi_R | \cdot) = Z^{-1} \exp \left\{ - \sum_{a, a' \in A} \langle \phi_R - \mu_R | a' \rangle \langle a' | G_R | a \rangle \langle a | \phi_R - \mu_R \rangle \right\}. \quad (3.27)$$

Using equation (3.24), we can write the middle operation out in full:

$$\langle a' | G_R | a \rangle = \langle a' | F_{RR} - F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} | a \rangle. \quad (3.28)$$

In applying the second term, $F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R}$, to the function $|a\rangle$, the operator $F_{\bar{R}R}$ is first applied to $|a\rangle$. Expanding this operator in terms of the maps defined in figure 3.2, we can see that this operation becomes

$$F_{\bar{R}R} |a\rangle = \pi_{\bar{R}} F i_R |a\rangle.$$

Condition 1 states that the function $i_R |a\rangle$ is an eigenfunction of the operator F . Hence we have that

$$\pi_{\bar{R}} F i_R |a\rangle = \pi_{\bar{R}} f_a i_R |a\rangle.$$

By definition of the injective map i_R , the function $i_R |a\rangle$ is equal to $|a\rangle$ inside the region R and to zero in \bar{R} . From this it follows that the support of $f_a i_R |a\rangle$ lies in the region R . Applying the surjective map $\pi_{\bar{R}}$ to this therefore yields zero

$$\pi_{\bar{R}} f_a i_R |a\rangle = 0$$

which in turn annihilates the second term in the operator G_R completely.

We are left with the application of the operator F_{RR} to the function $|a\rangle$. When expanded, this becomes

$$F_{RR}|a\rangle = \pi_R F i_R |a\rangle.$$

Using the same arguments as above, the function $F i_R |a\rangle = f_a i_R |a\rangle$ has its support within the region R . As $\pi_R i_R = id$, applying the surjective map π_R to $f_a i_R$ results in the first term of the operator G_R being reduced to $f_a |a\rangle$.

So we have shown that condition 1 means that the set \mathcal{B} are eigenfunctions of the operator on the region, G_R , with eigenvalues f_a , *i.e.*

$$\langle a' | G_R | a \rangle = \langle a' | f_a | a \rangle = f_a \langle a' | a \rangle. \quad (3.29)$$

Condition 2, which states that the set \mathcal{B} should form an orthonormal basis for functions on the region R , *i.e.* $\langle a' | a \rangle = \delta(a, a')$, means that G_R is diagonalised by B :

$$\sum_{a, a' \in A} f_a \delta(a, a') \langle \phi_R - \mu_R | a' \rangle \langle a | \phi_R - \mu_R \rangle = \sum_{a \in A} f_a \langle \phi_R - \mu_R | a \rangle \langle a | \phi_R - \mu_R \rangle$$

which allows us to write our measure on the finite region R as

$$\Pr(\phi_R | \cdot) = \prod_{a \in A} \left(\frac{f_a}{\pi} \right)^{\frac{1}{2}} \exp \left\{ - \sum_{a \in A} f_a \langle \phi_R - \mu_R | a \rangle \langle a | \phi_R - \mu_R \rangle \right\}. \quad (3.30)$$

To summarise, if we can find such a set \mathcal{B} that satisfies the above two conditions then we can diagonalise our operator G_R and hence obtain an expression for the probability of an image defined on a region R which is relatively quick to compute. The question that remains is: how do we find such a set \mathcal{B} ?

3.2 Translation Invariance

Before investigating the existence of a set of functions which satisfy the above conditions, there is an important property of the distribution which needs to be discussed. This property arises from the fact that texture can appear in an image arbitrarily translated and so, as part of our model development, we require that the distribution on Φ for a given texture be translation invariant. This means that images such as those in figure 3.4 will have equal probability of belonging to the same texture class. At this point we do not impose rotation or scale invariance. Therefore the models describe texture with a particular orientation at a particular resolution. Later, in chapter 6, we discuss a possible extension of the model to include rotation invariance.

The requirement of translation invariance and our choice of a Gaussian distribution on Φ simplifies the form of the operator F of the distribution. This can be seen in what follows.

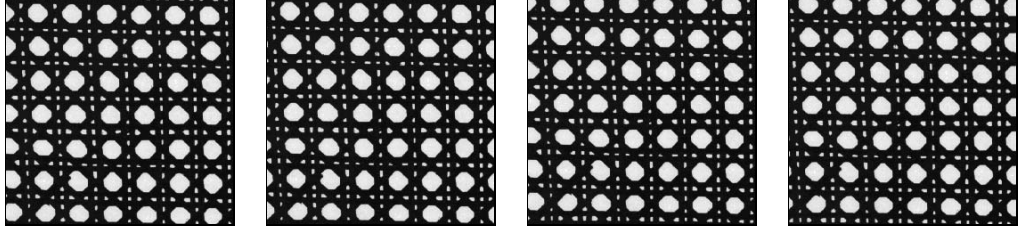


Figure 3.4: Four shifted versions of the D102 texture from the Brodatz album.

3.2.1 Translation Operator

We define a translation operator T_δ on the spatial domain as:

$$[T_\delta \phi](x) = \phi(x + \delta) \quad (3.31)$$

for all points $x \in D_\infty$. The Fourier transform of this operator applied to the image ϕ is given by

$$\begin{aligned} \left[\widehat{T_\delta \phi} \right](k) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-ikx} [T_\delta \phi](x) dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-ikx} \phi(x + \delta) dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-ik(x'-\delta)} \phi(x') dx' \\ &= e^{ik\delta} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-ikx'} \phi(x') dx' \\ &= e^{ik\delta} \hat{\phi}(k). \end{aligned} \quad (3.32)$$

3.2.2 Requiring Translation Invariance

Saying that we require the distribution to be translation invariant means that we need

$$\Pr(\phi | \cdot) = \Pr(T_\delta \phi | \cdot) \quad (3.33)$$

for all possible shifts δ and for all possible realisations of the image ϕ . Expressing these two probabilities in the Fourier basis yields the requirement that, for all δ and ϕ :

$$Z^{-1} e^{-\int_k \int_{k'} \hat{\phi}^*(k) F(k, k') \hat{\phi}(k') dk dk'} = Z^{-1} e^{-\int_k \int_{k'} [\hat{T}_\delta \phi]^*(k) F(k, k') [\hat{T}_\delta \phi](k') dk dk'} \quad (3.34)$$

where $*$ denotes the complex conjugate. Using equation (3.32) we can further develop the right hand side of this expression to give the requirement that for all δ and ϕ :

$$\begin{aligned} Z^{-1} e^{-\int_k \int_{k'} \hat{\phi}^*(k) F(k, k') \hat{\phi}(k') dk dk'} &= Z^{-1} e^{-\int_k \int_{k'} e^{-ik\delta} \hat{\phi}^*(k) F(k, k') e^{ik'\delta} \hat{\phi}(k') dk dk'} \\ &= Z^{-1} e^{-\int_k \int_{k'} e^{i(k'-k)\delta} \hat{\phi}^*(k) F(k, k') \hat{\phi}(k') dk dk'} . \end{aligned}$$

The only way this can be true is if the exponents are equal, *i.e* if for all δ and ϕ :

$$\int_k \int_{k'} \hat{\phi}^*(k) F(k, k') \hat{\phi}(k') dk dk' = \int_k \int_{k'} e^{i(k'-k)\delta} \hat{\phi}^*(k) F(k, k') \hat{\phi}(k') dk dk' .$$

This in turn implies that

$$F(k, k') = e^{i(k'-k)\delta} F(k, k')$$

for all δ . In order for this to hold, either $F(k, k') = 0$ or $e^{i(k'-k)\delta} = 1$ for all δ . We examine the cases for the diagonal and the off-diagonal elements separately:

Case $k' \neq k$: For the off-diagonal elements, if $e^{i(k'-k)\delta} = 1$ is to hold for all shifts δ , then it must be that for each δ there exists a constant m which satisfies the following equality: $(k' - k) \delta = 2 \pi m + \frac{\pi}{2}$. This is clearly impossible for all shifts δ . So, for the off-diagonal elements it must be that $F(k, k') = 0$.

Case $k' = k$: When dealing with the diagonal elements of the operator F , the condition that $e^{i(k-k)\delta} = 1$ must hold for all shifts δ , reduces to the condition that $e^0 = 1$ must hold for all shifts δ , which is always true. Hence for diagonal elements $F(k, k') = f(k)$.

From this it can be seen that the condition of translation invariance on the Gaussian distribution causes the distribution to be diagonal in the Fourier basis. Another consequence of translation invariance is that the mean of the distribution, μ , must be constant, so that its Fourier transform is a delta function: $\hat{\mu}(k) \propto \delta(k)$.

We can therefore express the probability of the infinite image ϕ as:

$$\begin{aligned} \text{Pr}(\phi \mid \cdot) &= Z^{-1} e^{-\int_k \int_{k'} [\hat{\phi}^*(k) - \hat{\mu}^*(k)] F(k, k') [\hat{\phi}(k') - \hat{\mu}(k')] dk dk'} \\ &= Z^{-1} e^{-\int_k \int_{k'} [\hat{\phi}^*(k) - \hat{\mu}^*(k)] f(k) \delta(k, k') [\hat{\phi}(k') - \hat{\mu}(k')] dk dk'} \\ &= Z^{-1} e^{-\int_k [\hat{\phi}^*(k) - \hat{\mu}^*(k)] f(k) [\hat{\phi}(k) - \hat{\mu}(k)] dk} . \end{aligned}$$

So the requirement of translation invariance means that the Gaussian distribution is characterised by a function, f , on the Fourier domain:

$$\Pr(\phi | \cdot) = \left| \frac{F}{\pi} \right|^{\frac{1}{2}} \exp \left\{ - \int_k f(k) [\hat{\phi}^*(k) - \hat{\mu}^*(k)] [\hat{\phi}(k) - \hat{\mu}(k)] dk \right\}. \quad (3.35)$$

3.3 Using Wavelet Packets

Requiring the Gaussian distribution on texture to be translation invariant has simplified the form of its variance-covariance matrix. As shown above, if we express the distribution in the Fourier basis, then the operator F is reduced to a diagonal matrix on the Fourier basis elements. This has, without a doubt, made the task of finding the set \mathcal{B} much easier because we no longer have to consider correlations between these fundamental frequencies. However, the function f , which now characterises our distribution, may still be very complicated. For a completely arbitrary function f on the set of Fourier frequencies, it is very difficult to find a set of functions, \mathcal{B} , on the finite region that satisfies the two conditions of section 3.1.3. Therefore the form of the function f must be restricted in such a way that it is varied enough to capture the structure present in the texture, but also limited enough that it will satisfy the conditions.

We propose the following restriction on the function f : Consider the set \mathcal{T} of dyadic partitions of a quarter of the Fourier domain. We allow a choice of three types of partitions: a quad split, a binary split in the horizontal direction, and a binary split in the vertical direction. These three different partitions can be seen in figure 3.5.

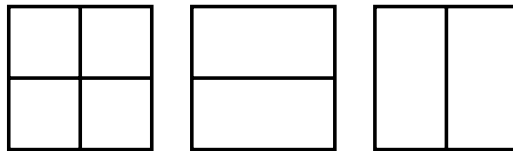


Figure 3.5: Possible partitions of the frequency domain: a quad split, a horizontal binary split, and a vertical binary split.

By recursively applying these partitions we can focus on the area of the frequency domain which has a significant amount of activity. The level of the partition refers to the maximum number of recursive splits carried out in any one area of the partition. Usually, in analyses which involve partitioning the Fourier domain, only quad splits are considered. We have introduced the other two binary splits in order to add an extra element of adaptivity to the partitioning operation. Having three possible splits allows the partition to adapt slightly better to the structure present in the texture. Inherent to this partitioning process is the notion of a decomposition tree where the root of the tree corresponds to the quarter of the frequency domain being analysed. A quad split

will therefore generate four leaves at level one of this tree and the binary splits will each generate two leaves. The tree is defined recursively on these splits. Examples of partitions and their corresponding trees are shown in figures 3.6, 3.7 and 3.8. From this point onwards, we will use the terms partition and tree interchangeably.

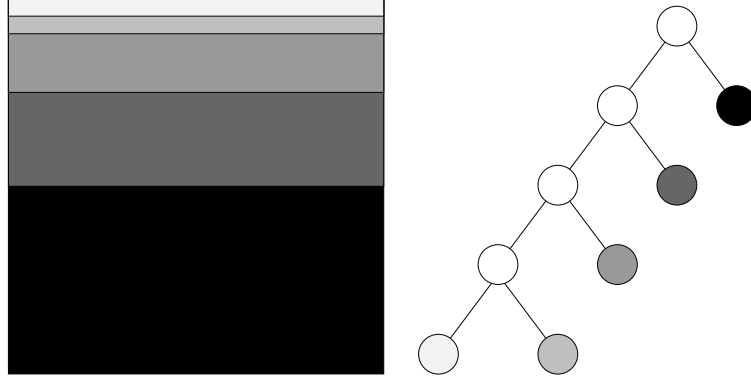


Figure 3.6: A level-4 partition of the Fourier domain using only horizontal splits.

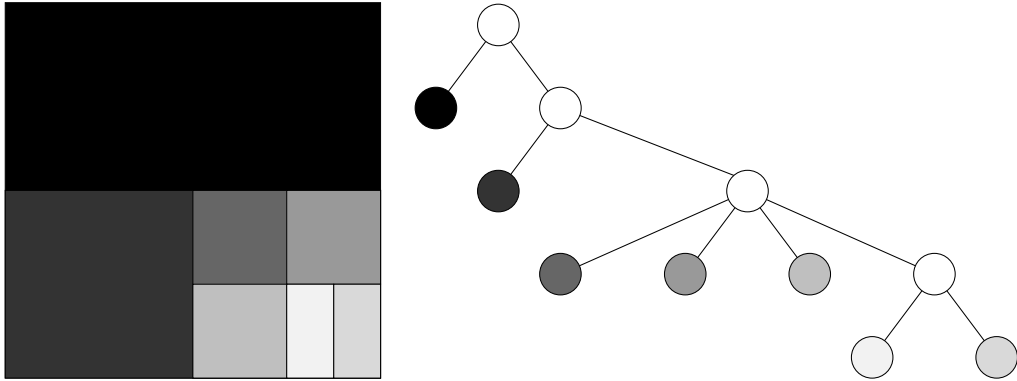


Figure 3.7: A level-4 partition of the Fourier domain using all three types of splits.

Based on the set \mathcal{T} of all possible partitions, we limit our choice of f to functions which are piecewise constant on a partition in this set. More formally, this is expressed as follows. Define a set of functions \mathcal{F} by:

$$\mathcal{F} = \bigcup_{T \in \mathcal{T}} \mathcal{F}_T \quad (3.36)$$

where the set $\mathcal{F}_T = \{f : \forall \alpha \in T, f \text{ is constant on } \alpha\}$. Here α indexes the elements in the partition.

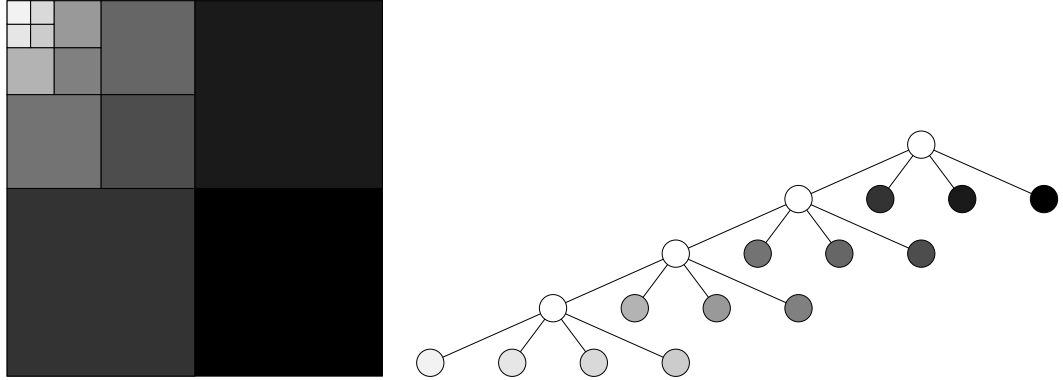


Figure 3.8: A level-4 partition of the Fourier domain using only quad splits and decomposing only the low frequency section of the Fourier domain. This partition corresponds to a standard wavelet decomposition of the frequency domain.

Given a partition $T \in \mathcal{T}$, and a mother wavelet, we can define a wavelet packet basis W_T , with each element of the partition corresponding to a subband in the final wavelet packet decomposition.

Due to the piecewise constancy of the functions in \mathcal{F}_T , every Fourier basis function k , which lies within the α^{th} element of the dyadic partition T , is an eigenfunction of the operator f with the same eigenvalue f_α . Every linear combination of these Fourier basis functions which lie within the α^{th} element of the partition is also an eigenfunction of the operator f and has eigenvalue f_α .

As discussed in appendix A each function in a wavelet (or wavelet packet) basis has a local support in both space and frequency. Examples of the frequency support of several of these basis functions are shown in figure 3.9. A wavelet basis function with frequency support in the α^{th} element of the dyadic partition is, by definition, a linear combination of Fourier basis elements in α and so it is an eigenfunction of the operator f with eigenvalue f_α .

More formally, this can be demonstrated by investigating the effect of applying the

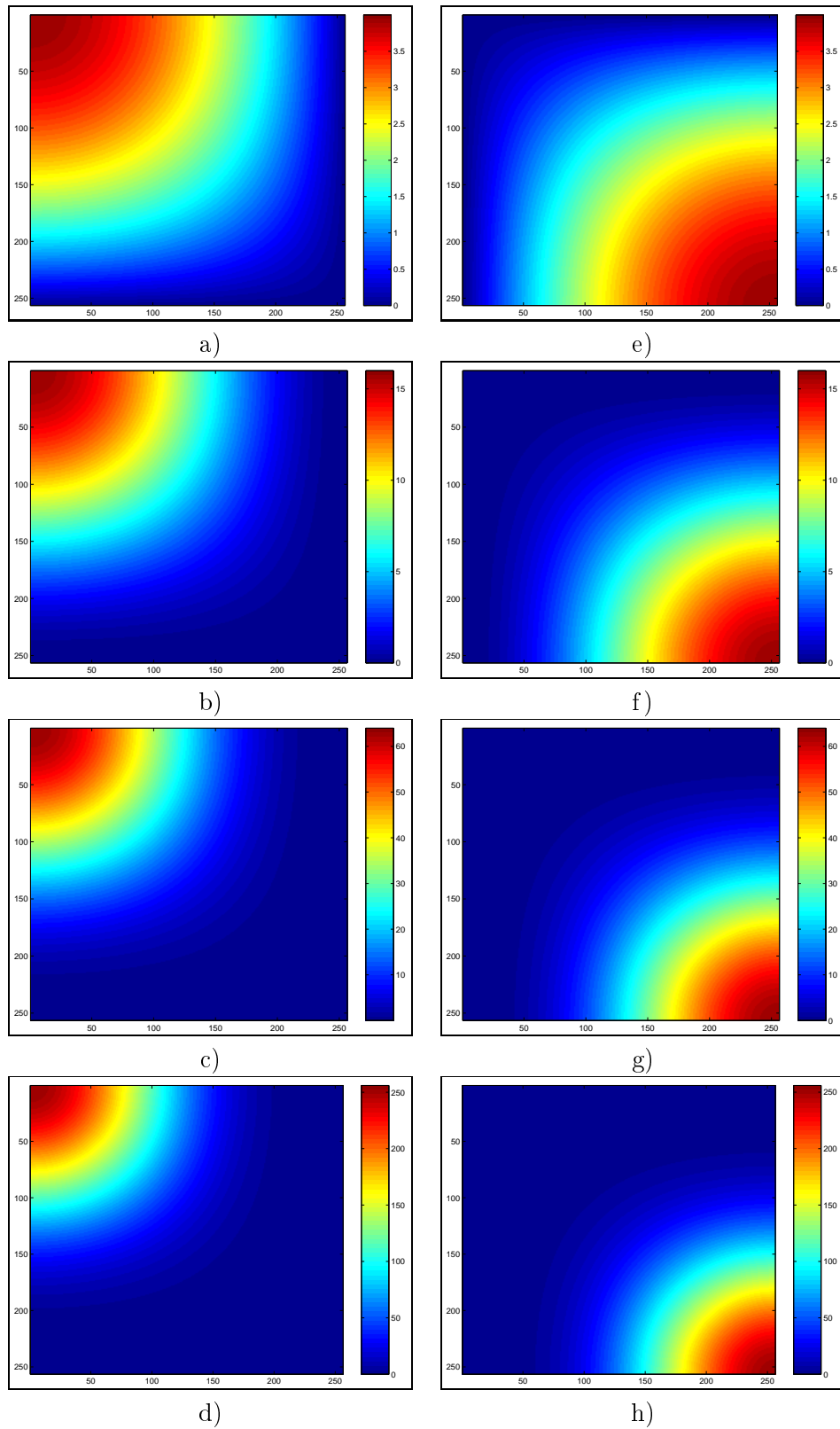


Figure 3.9: Left column: The decreasing frequency support of the lowpass wavelet basis functions at levels 1–4 (a–d) in a standard wavelet decomposition. Right column: The decreasing frequency support of the highpass wavelet packet basis functions at levels 1–4 (e–f) in a wavelet packet decomposition.

operator F to the α^{th} wavelet packet basis function ω_α as follows:

$$\begin{aligned}
F|\omega_\alpha\rangle &= \sum_k \sum_{k'} |k\rangle\langle k|F|k'\rangle\langle k'|\omega_\alpha\rangle \quad ; \text{ where } |k\rangle\langle k| = id \\
&= \sum_k \sum_{k'} |k\rangle f(k) \delta(k, k') \langle k'|\omega_\alpha\rangle \quad ; \text{ translation invariance} \\
&= \sum_k |k\rangle f(k) \langle k|\omega_\alpha\rangle \\
&= f_\alpha \sum_{k \in \alpha} |k\rangle\langle k|\omega_\alpha\rangle \quad ; \langle k|\omega_\alpha\rangle = 0 \text{ unless } k \in \alpha \in T \\
&= f_\alpha \sum_k |k\rangle\langle k|\omega_\alpha\rangle \\
&= f_\alpha |\omega_\alpha\rangle.
\end{aligned}$$

Similarly, each element of the basis W_T has a frequency support that lies approximately in the element of the partition T corresponding to its subband. Due to the piecewise constancy of the functions in \mathcal{F}_T , the basis elements are thus approximate eigenfunctions of the operators defined by those functions. Those basis elements whose support lies in the region R thereby satisfy condition 1 to a certain approximation. Our next task is to complete the set of wavelets inside the region R in order to make a basis for the region and in doing so satisfy condition 2. How we do this depends on the shape of R . We consider two possibilities: dyadic and arbitrarily shaped regions.

3.3.1 Dyadic Shaped Regions

A region which is dyadic in shape has dimensions which are a power of two. If we have such a region R , then we can use a decimated wavelet packet decomposition to obtain a basis for R . Given a partition T and a function $f \in \mathcal{F}_T$, the measure on the region presented earlier in equation (3.30) takes on the form:

$$\Pr(\phi_R | \mathbf{f}, \boldsymbol{\mu}, T) = \prod_\alpha \left[\left(\frac{f_\alpha}{\pi} \right)^{\frac{N_\alpha}{2}} \exp \left\{ -f_\alpha \sum_{i \in \alpha} (\omega_{\alpha,i} - \mu_\alpha)^2 \right\} \right]. \quad (3.37)$$

Here α indexes the subbands (leaf nodes) of the decomposition T . The number of coefficients in subband α is given by N_α . The i^{th} wavelet coefficient of the finite dyadic shaped image ϕ_R is given by $\omega_{\alpha,i}$. The mean of subband α is denoted by μ_α . It should be noted here that μ_α is zero unless $\alpha = 0$, which corresponds to the scaling coefficient subband. f_α is the (constant) value of f on subband α . On the left hand side of equation (3.37), the boldface symbols $\boldsymbol{\mu}$ and \mathbf{f} denote the sets containing the means and inverse variances, respectively, for all values of α .

3.3.2 Arbitrarily Shaped Regions

For arbitrarily shaped regions, dyadic wavelet packets no longer form a basis. There are two problems. First, the basis elements may not be aligned with the boundary and so may include a significant information from outside the region R . Second, a shifting of the region with respect to the basis elements will produce a different representation of the same texture.

The problem of dealing with such regions comes into play in applications such as the segmentation of textured scenes, where we do not have control over the shape of the region being analysed. Later, in section 5, we detail two methods for treating this situation.

Chapter 4

Training Texture Models

The probabilistic framework developed so far provides us with the means to describe texture via a model which adapts to its underlying structure. Training such a model can be expressed as the task of finding

- the wavelet packet basis, or equivalently the dyadic partition T , for the texture of interest and
- the values of the means, μ_α , and the inverse variances, f_α , associated with each leaf node (subband) $\alpha \in T$.

In order to train a texture model, we need representative samples of the texture which can be used to estimate the parameters of the model. We will use Maximum A Posteriori (MAP) estimation to find the optimal parameter set for the texture model. The first step in this estimation process is the construction of a posterior probability of the parameters given the training data available for the texture. Using Bayes' theorem, this probability is expressed as

$$\Pr(\mathbf{f}, \boldsymbol{\mu}, T \mid \mathbf{d}) = \frac{\Pr(\mathbf{d} \mid \mathbf{f}, \boldsymbol{\mu}, T) \Pr(\mathbf{f}, \boldsymbol{\mu}, T)}{\Pr(\mathbf{d})}. \quad (4.1)$$

The probability of the training data given the model parameters is determined via the likelihood function, $\Pr(\mathbf{d} \mid \mathbf{f}, \boldsymbol{\mu}, T)$. The *a priori* knowledge about the model parameters is captured by the prior probability, $\Pr(\mathbf{f}, \boldsymbol{\mu}, T)$.

As a MAP approach to parameter estimation requires us to maximise the expression in equation (4.1) over all possible parameter values, the denominator on the right hand side can be disregarded as it is a function only of the data and so will not affect the optimisation step. The reduced form of the posterior probability considered is therefore given by

$$\Pr(\mathbf{f}, \boldsymbol{\mu}, T \mid \mathbf{d}) \propto \Pr(\mathbf{d} \mid \mathbf{f}, \boldsymbol{\mu}, T) \Pr(\mathbf{f}, \boldsymbol{\mu}, T). \quad (4.2)$$

In what follows, we discuss each term of this expression in turn.

4.1 The Likelihood Function

As previously shown in section 3.3.1, it is easy to construct wavelet packet bases for dyadic shaped regions in an image. Sample patches used to train the texture model are therefore chosen to be of this shape. If only a single texture patch, which we will denote by ϕ_R , is used to estimate the parameters of the model, then the form of the likelihood $\Pr(\phi_R \mid \mathbf{f}, \boldsymbol{\mu}, T)$ is given by equation (3.37).

Choosing only one data sample, however, will cause problems later on if we try to use the resulting texture model to analyse other sample patches from the same texture. The reason for this is clear and is related to the notion of translation invariance. As we discussed in chapter 2, the decimated wavelet transform of a texture patch involves a downsampling step which produces a representation of the texture which will differ significantly from the representation obtained by performing the same transform on a patch taken from another part of the same texture. Such a difference in representations will prove fatal for any classification or segmentation scheme which relies on the model to recognise samples of the texture which appear anywhere translated in a scene. It is for this reason that modelling the underlying structure present in the texture, while ignoring the effect of different translations, requires us to use a number of samples. In this way we may obtain what can be considered as an *average* representation of the texture.

4.1.1 Multiple Training Patches

As we would like our distribution to be translation invariant, we select M patches, denoted by $\phi_R^{(1)}, \dots, \phi_R^{(M)}$, which represent shifted versions of the texture. The size of the patches are manually chosen so that the largest periodicity in the texture is present in the patch. The number of patches, M , is chosen to cover a few periods of the largest periodicity in the texture. Multiple patches are selected from the original texture using an evenly spaced process which starts from the top left corner and proceeds in a left to right, top to bottom manner. Figure 4.1 shows an example of this patch selection procedure.

In this case, the form of the likelihood becomes

$$\Pr(\phi_R^{(1)}, \dots, \phi_R^{(M)} \mid \mathbf{f}, \boldsymbol{\mu}, T) = \prod_{\alpha} \left[\left(\frac{f_{\alpha}}{\pi} \right)^{\frac{MN_{\alpha}}{2}} \exp \left\{ -f_{\alpha} \sum_m \sum_i (\omega_{\alpha,i}^m - \mu_{\alpha})^2 \right\} \right] \quad (4.3)$$

where $\omega_{\alpha,i}^m$ is the i^{th} wavelet coefficient in subband α of the decomposition on the m^{th} training patch, and the mean of subband α is given by μ_{α} .

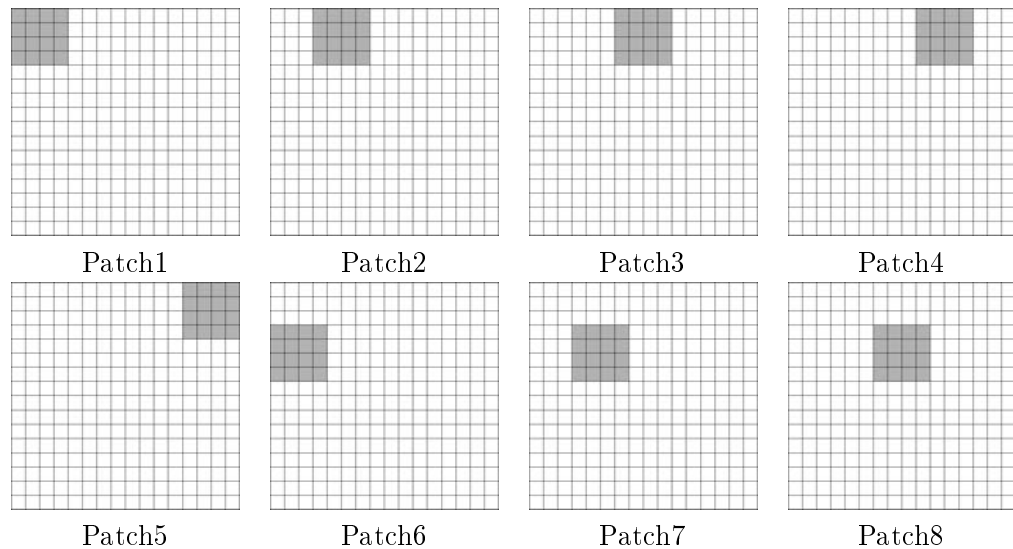


Figure 4.1: An example of the patch selection procedure wherein eight translated patches are chosen from the original textured image for use in training.

4.2 The Prior Probability

Now that we have a suitable expression for the likelihood, we turn our attention to the choice of prior probability for the model parameters. This probability can be rewritten as

$$\Pr(\mathbf{f}, \boldsymbol{\mu}, T) = \Pr(\mathbf{f}, \boldsymbol{\mu} \mid T) \Pr(T). \quad (4.4)$$

Let us first consider the prior distribution on the wavelet packet decomposition tree T . It is important to realise that a small number of subbands will ease the computational burden of any classification or segmentation scheme which makes use of these texture models. Ideally, therefore, the decomposition should adapt to the underlying structure of the texture using as few subbands as possible. Incorporating this requirement into the prior distribution on T will promote *smooth* decompositions, *i.e.* those which are constant on large regions. An obvious choice for this kind of prior is

$$\Pr(T) = Z^{-1}(\beta) e^{-\beta |T|} \quad (4.5)$$

where $|T|$ is the total number of nodes in the entire wavelet packet decomposition tree, and β determines the severity of the penalisation. Although this prior penalises large trees, it does so by assigning an equal weight of β to each node in the tree. As we are performing a decimated wavelet packet decomposition, the size of the nodes in the tree decreases as their level in the tree increases. Therefore, the prior in equation (4.5) does not penalise the tree size in an accurate manner.

In order to take into account the difference in node sizes throughout the tree, we do not consider each new node as a single unit, but instead as the number of pixels it contains. Therefore, each node α in the tree should be penalised less than its parent node, but more than its child nodes. We have adopted this approach by assigning a weight of $\frac{\beta}{M_\alpha}$ to each node in the tree, where M_α is the factor which represents the difference in size of the root node and the current node α . We refer to this amount as the *redundancy factor*. The notion of a redundancy factor for a node comes from the redundancy present if the wavelet transform is not decimated. Figure 4.2 shows the meaning of M_α for each level in an undecimated wavelet transform.

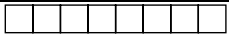
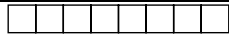
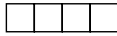
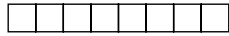

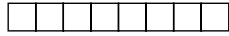

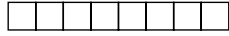
Decimated	Level	Undecimated	Redundancy Factor
	0		$2^0 = 1$
	1		$2^1 = 2$
	2		$2^2 = 4$
	3		$2^3 = 8$

Figure 4.2: The difference between the outputs of a decimated and an undecimated wavelet transform which shows the redundancy associated with each level in the latter.

With this in mind, we can write the prior distribution on T as

$$\Pr(T) = Z^{-1}(\beta) \prod_{\alpha \in T} \exp \left\{ -\frac{\beta}{M_\alpha} \right\}. \quad (4.6)$$

This prior penalises decompositions of uniform depth relative to decompositions of the same size but with varying depth. An example of this is shown in figure 4.3 wherein both decompositions have the same number of nodes.

The decomposition on the left is a full wavelet packet decomposition down to level 2 and so consists of 16 partitions of the same size. The decomposition on the right hand side is an adaptive wavelet packet decomposition, down to level 4. It is clear from the diagram that this decomposition has focused on a particular area of the frequency domain and has left the other areas intact. The prior in equation 4.6 will favour the adaptive decomposition over the full standard wavelet packet decomposition.

Once the decomposition T is known, we express our ignorance about the values of \mathbf{f} and $\boldsymbol{\mu}$ by assuming a uniform distribution on both parameters.

4.3 Estimating the Texture Model Parameters

With these distribution choices in mind, we can perform a MAP estimation for \mathbf{f} and $\boldsymbol{\mu}$. As shown in section 3.2, the Gaussian assumption combined with the requirement

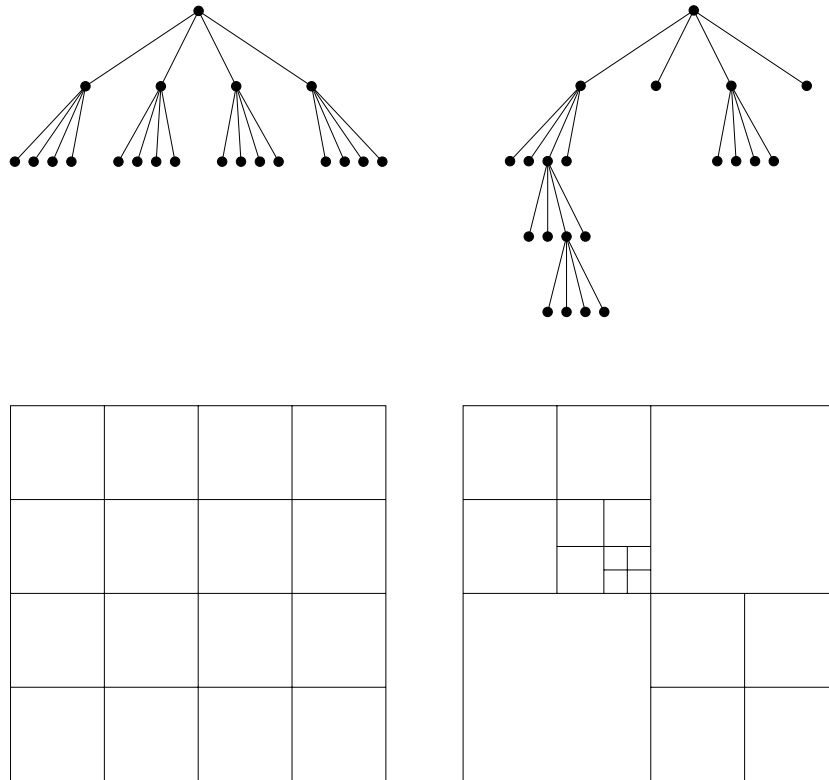


Figure 4.3: An example of how the prior probability on the decomposition T prefers varying depth decompositions over uniform depth ones.

of translation invariance produced a distribution which is characterised by a function f on the Fourier domain. Our restriction on the form of this function, along with the frequency support of the wavelet basis functions gives rise to the fact that the subbands in the resulting decomposition are independent to the degree that the wavelets frequency support lies in a partition element. This approximate independence between subbands means that we can perform the MAP estimation on a per subband basis which, for a fixed tree structure T , yields

$$\hat{f}_\alpha = \frac{1}{2\bar{e}_\alpha} = \frac{MN_\alpha}{2 \sum_m \sum_i (\omega_{\alpha,i}^m - \mu_\alpha)^2} \quad (4.7)$$

and

$$\hat{\mu}_\alpha = \frac{1}{MN_\alpha} \sum_m \sum_i \omega_{\alpha,i}^m \quad (4.8)$$

where \bar{e}_α is the average energy per wavelet coefficient in subband α .

In order to find the optimal decomposition and its corresponding f_α and μ_α values we use an algorithm which performs a depth-first search through the space \mathcal{T} to find the exact MAP estimates for T , \mathbf{f} , and $\boldsymbol{\mu}$.

4.3.1 Algorithm

An outline of this search algorithm for a single texture patch is provided by the pseudocode in figure 4.4. In this section, all references to line numbers are relative to this pseudocode.

The algorithm starts by calling a recursive depth-first search function FOD on the root node of the decomposition tree. For each node α generated by this recursive call, its associated parameter values, μ_α and f_α , are calculated (line 06). In the pseudocode presented in figure 4.4 the root node of the decomposition tree is set to contain a single texture patch (line 02). However, in practice, we use M texture patches which means that the parameter values are actually calculated from equations (4.7) and (4.8), which use information from all M available data sources. A brief insight into the code structure for multiple training texture patches is given in appendix C.

At this point, the probability of node α is calculated using equation (4.3). It should be noted that this equation represents the product over all leaf nodes in the decomposition and so here only the contribution of node α to this product is computed. The penalty incurred by the addition of node α to the existing decomposition tree is calculated as outlined in section 4.2. This penalty is then combined with the probability of node α which results in what we refer to as the penalised probability associated with the node (line 07).

When the parameters and the penalised probability for a given node have been calculated, the algorithm performs a test to determine the depth at which the node is

```
01| Create the root node R of the decomposition tree.
02| Set it to contain the original training patch.
03| Call the following recursive function FOD on R.
04|
05| FOD (N) // Find Optimal Decomposition for node N
06|   Compute parameter values for node N
07|   Compute the penalised probability of node N
08|   IF node N is at the maximum decomposition level
09|     Set subtree to be empty
10|   ELSE
11|     Generate the three possible splits of node N
12|     FOR EACH split
13|       FOR EACH node C in the split
14|         FOD (C)
15|       END
16|     END
17|     Make decision about subtree structure
18|   RETURN
```

Figure 4.4: Pseudocode for the depth-first search algorithm which finds the optimal wavelet packet decomposition for a given texture.

located in the tree (line 08). If the node is already at the maximum level, then the function FOD sets the subtree of this node to be empty (line 09) and returns, causing the depth-first search algorithm to backtrack one level (line 18).

In the case where the maximum level has not yet been reached, the algorithm creates three possible splits of the current node (line 11) which correspond to the quad split, binary split in the vertical direction, and binary split in the horizontal direction that were introduced in section 3.3. The function FOD is called on each child node of each split (lines 13 and 14). This is where the depth-first search of the decomposition tree actually takes place.

Once all the calls to the function FOD on the child nodes have returned, the optimal subtree for each child node has been computed (line 16). The algorithm must now make the decision to either keep the current node as the best representation of this section of the frequency domain (this amounts to setting its subtree to be empty, *i.e.* no further splits), or select one of the three subtrees which represent the three possible splits of this section of the frequency domain (line 17).

In order to make this decision, a penalised probability must be calculated for each of the three possible splits. This is done by summing the penalised probabilities of the child nodes in the split and then adding a penalty corresponding to the root node of the subtree this split represents (*i.e.* the current node). Once computed, these penalised split probabilities are compared to the penalised probability of the current node. If the current node has a higher penalised probability, then its subtree is set to be empty. Otherwise the subtree of the current node is set to be the subtree corresponding to the split which has the highest penalised probability.

When the function FOD returns from its initial call on the root node, the tree which remains is the optimal wavelet packet decomposition tree for the training patch contained in the root node for the particular choice of mother wavelet and penalty. Figure 4.5 shows a simplified example of how the depth-first recursion is carried out.

4.4 Training Results

In this section we present a visualisation of the optimal wavelet packet decompositions obtained with our training scheme. Models were trained for several textures from both the Brodatz texture album and from high resolution remote sensing images. The original textures are shown in figures 4.6 and 4.7, respectively. Below the textures are the magnitude images of their Fourier transforms which shows the type of structure present in each texture. Our algorithm attempts to capture this underlying structure in as few a subbands as possible.

To start our training process, we selected 64 sample patches of size 128×128 for each texture, according to the procedure outlined in section 4.1.1. The presence of the mean in an image usually greatly simplifies the task of describing and classifying regions of the image. Therefore, it may prevent us from accurately evaluating our framework for

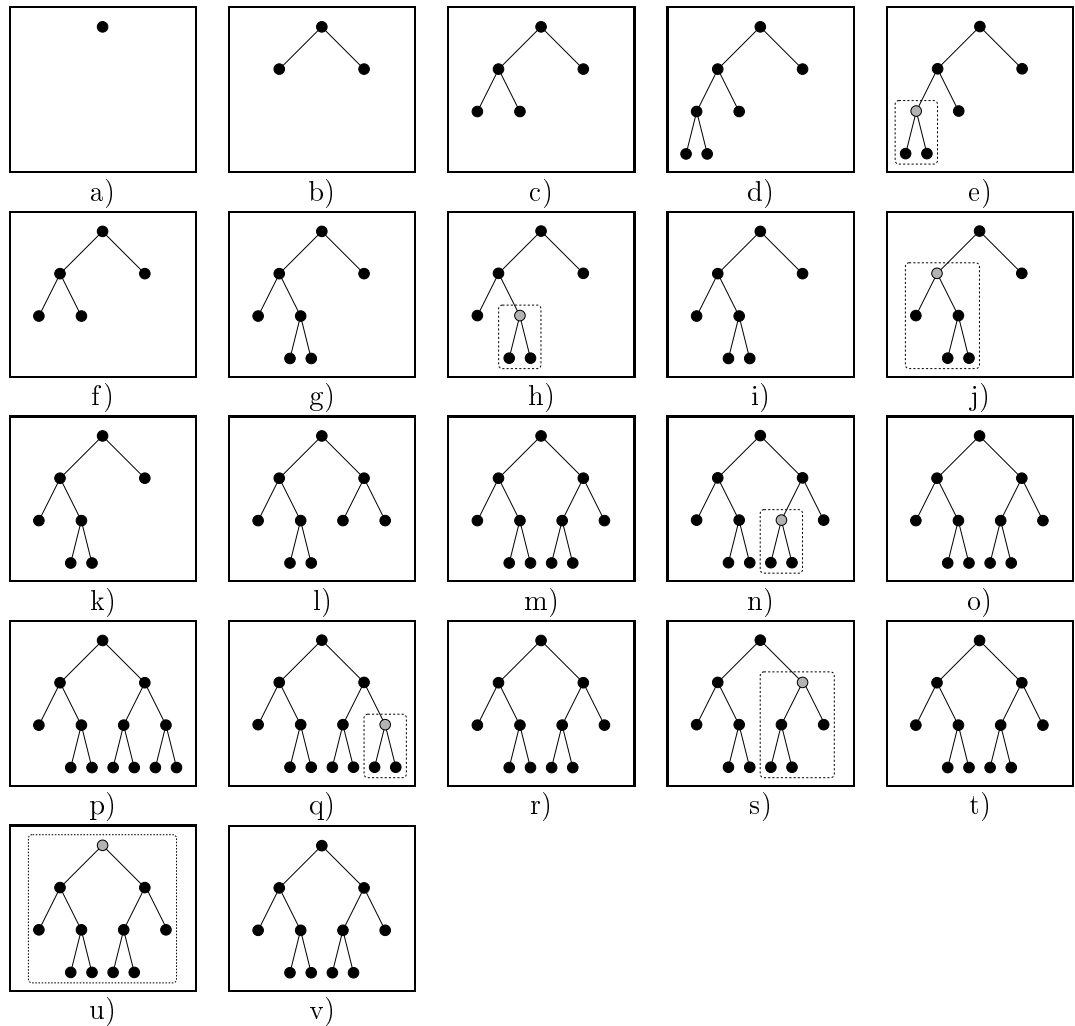


Figure 4.5: Choosing the optimal wavelet packet decomposition for a texture patch which is placed at the root of the decomposition tree (a). By considering only horizontal binary splits, each node will have either an empty subtree or a subtree consisting of two nodes. The depth-first search algorithm expands the decomposition tree down the left hand side until the maximum level (chosen here to be level 3) is reached (b)–(d). At this point, the algorithm backtracks to the node which forms the root of the subtree enclosed in the dashed box (e). Calculations are carried out to compare the penalised probability of the current node (coloured in gray) to that of the two way split that makes up its subtree. A decision is made to prune back the decomposition and set the subtree of the current node at level two to be empty (f). The other existing node at level two is expanded (g) and a similar process is carried out (h) resulting in the decision to keep the expansion (i). This process continues until every node in the tree has been visited and its optimal subtree fixed. At this point the option of retaining only the root node versus the option of keeping the adaptive subtree which has been constructed below it is considered (u). The decision is to leave the subtree untouched (v). The final diagram therefore shows the optimal wavelet packet decomposition tree structure for the single training patch contained in the root node for a given mother wavelet and penalty.

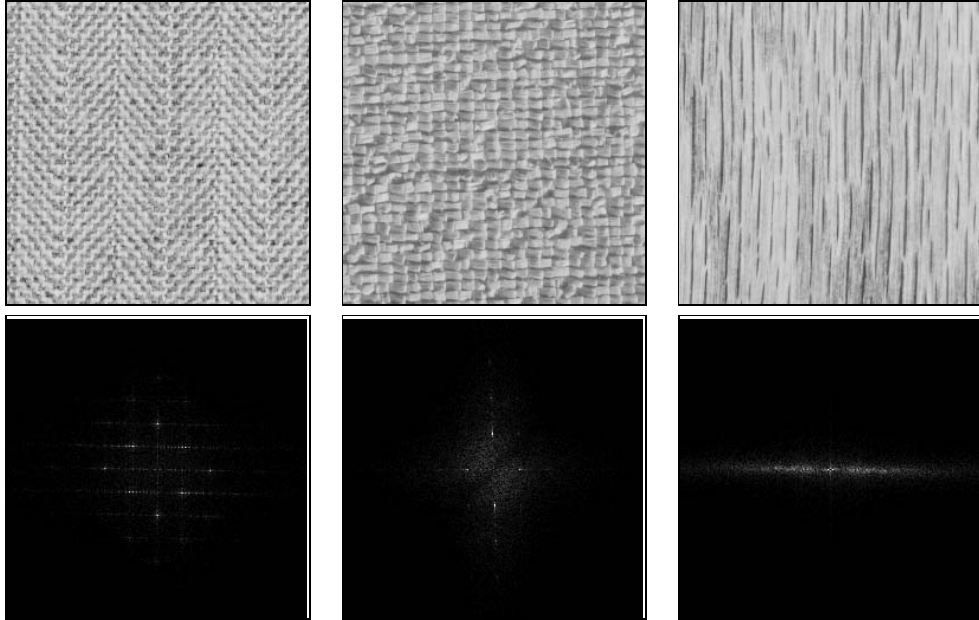


Figure 4.6: A selection of textures from the Brodatz album: Herring (left), Raffia (centre), Wood (right), and their corresponding Fourier transform magnitudes below.

texture description. It is for this reason that we decided to omit it from the calculations. We thus subtracted the mean intensity from each of the patches before using them to train the texture model. These sample texture patches were then used in the depth first search algorithm presented above in section 4.3.1.

A maximum level of 6 was chosen for the decomposition. This level allows a very fine division of the Fourier domain to take place if necessary, yet it avoids the computational explosion associated with higher levels.

We experimented with several kinds of mother wavelets. Results presented in this section show the different optimal wavelet packet decompositions obtained using the Haar, Daubechies4, and Coiflet1 mother wavelets.

The penalty β was chosen so as to produce a wavelet packet decomposition of the texture which strikes an even balance between the final number of leaf nodes and the amount of texture structure captured. This value was tuned using visual comparisons with the Fourier transform of the texture. Instead of comparing the resulting decompositions directly to the magnitude images of the Fourier transform as shown in figures 4.6 and 4.7, we generated folded versions of the magnitude images which represented the quarter of the frequency plane which is analysed by a wavelet decomposition. These folded magnitude images are shown in the first row of the figures 4.8– 4.13.

Changes in the choice of mother wavelet and the value of β yield very interesting and quite different decompositions. The images of these adaptive wavelet packet de-

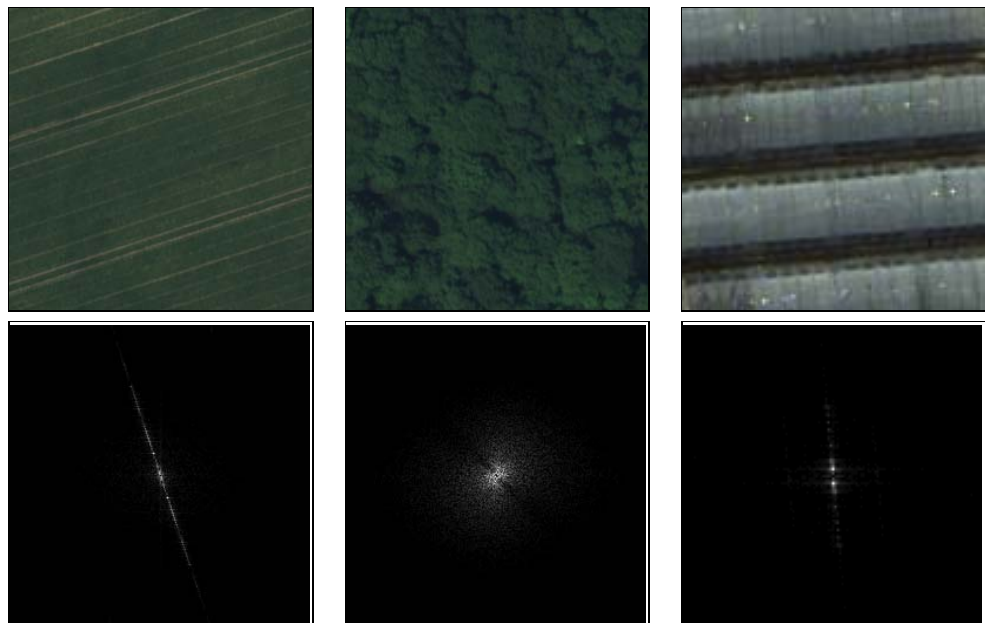


Figure 4.7: A selection of textures cropped from 50cm resolution aerial images: Ploughed (left), Forest (centre), Warehouse (right), and their corresponding Fourier transform magnitudes below. Images were provided by the French Mapping Institute (IGN).

compositions are shown in this section. Each image displays the logarithm of \bar{e}_α , shifted and rescaled to the range $[0, 255]$. The reason for not displaying the raw \bar{e}_α values is because most of these values are zero when scaled and quantised to eight bits.

From these results it can be seen that on average the resulting decomposition adapts well to the structure present in the texture. What remains to be seen is how these models perform in applications such as texture segmentation where an accurate description of texture is essential. This issue will be addressed in the following chapter.

4.5 Discussion: An Analysis of the Subband Statistics

An analysis of the subband coefficients of the adaptive wavelet packet decompositions suggests the need for a distribution which is more complicated than the Gaussian used in this thesis. Examples of subband histograms are shown in figure 4.14. It is clear that there are three distinct types of statistics present.

As mentioned in chapter 2, previous work in the area of texture analysis proposed the use of a generalised Gaussian distribution to model marginal wavelet statistics [59]. The first row in figure 4.14 shows coefficients which seem to follow this rule. These coefficients belong to high frequency subbands and the form of the distribution can easily be attributed to the fact the coefficients are a response to the edge content in the texture. Those coefficients which lie in or around zero represent the homogeneous area in the texture and some noise, while the other coefficients represent the edges in the textures. The second row shows histograms from subbands in broad intermediate to low frequency ranges which follow more closely a Gaussian distribution. Finally, in the third row, we can see a new bimodal form. Histograms of this shape arise from the subbands which adapted to the periodic content of the texture, and represent much narrower intermediate to low frequency ranges.

These three histogram forms were noted previously in the literature when the subband histograms from a standard wavelet decomposition were analysed as part of a texture classification study [24]. However, the histograms which exhibited a bimodal form were dismissed as occurring only in low frequency ranges which do not contain much textural information. The fact that they did not possess a unimodal form was ignored and all subbands were modelled by a generalised Gaussian distribution.

Our subband analysis suggests just the opposite: that the bimodal form occurs in subbands which correspond to the intensity peaks in the frequency domain that characterise the structure of the texture. This points toward an exciting new avenue of research which may permit a more accurate modelling of marginal wavelet statistics.

We tested the following quartic model:

$$\Pr(\phi \mid \cdot) = \frac{1}{Z(\mu_\alpha, \sigma_\alpha)} \exp \left\{ - \sum_{\alpha} \sum_{i \in \alpha} \frac{(e_{\alpha,i} - \mu_\alpha)^2}{2\sigma_\alpha^2} \right\} \quad (4.9)$$

where $e_{\alpha,i}$ denotes the energy of the i^{th} wavelet coefficient in subband α of the wavelet

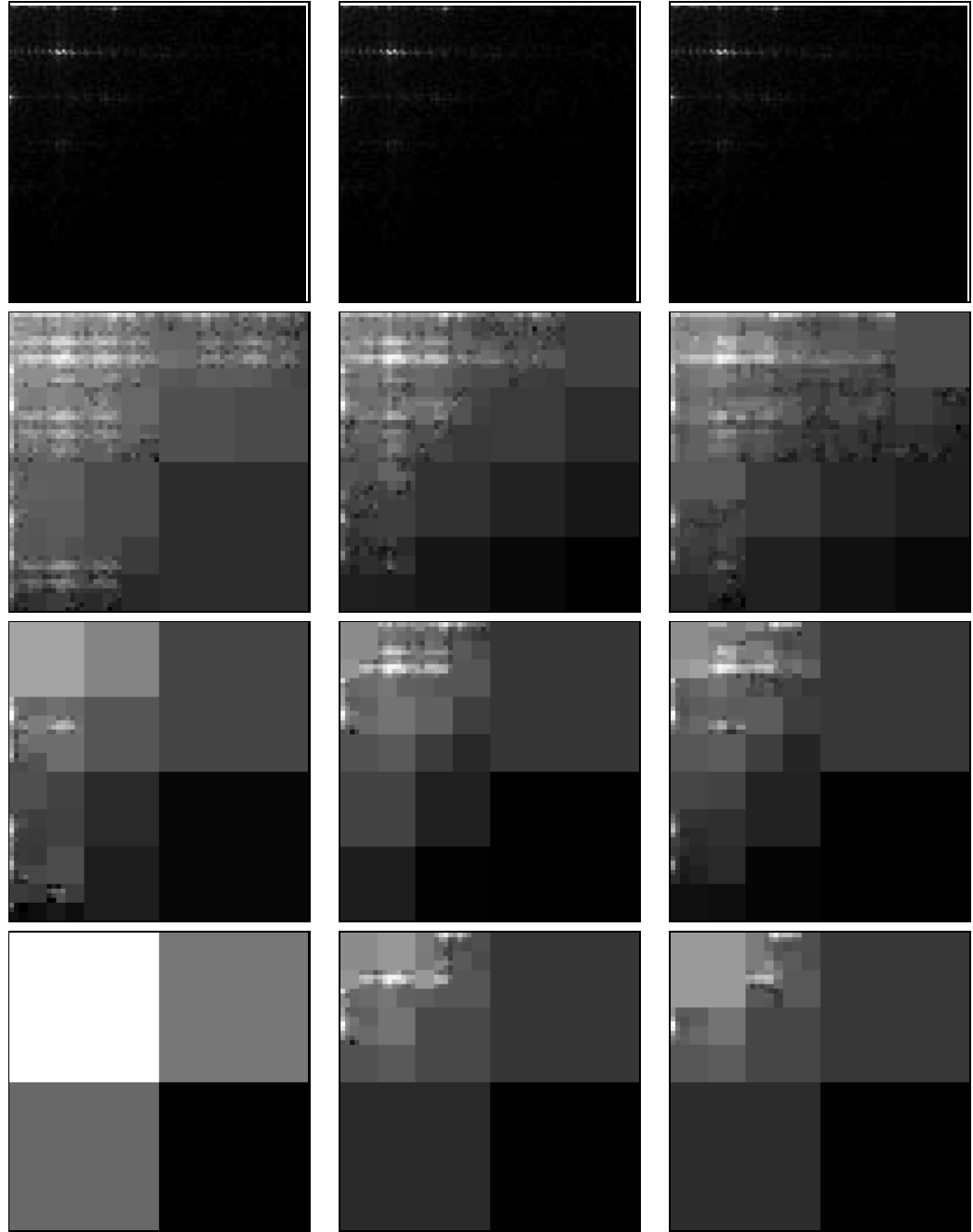


Figure 4.8: Top row: three identical copies of the folded magnitude Fourier transform of the Herring texture. The nine images below show the optimal decompositions obtained using the Haar, Coiflet1, and D4 wavelets (columns 1, 2, and 3 respectively), and β penalty values of 2000, 5000, and 8000 (rows 2, 3, and 4 respectively).

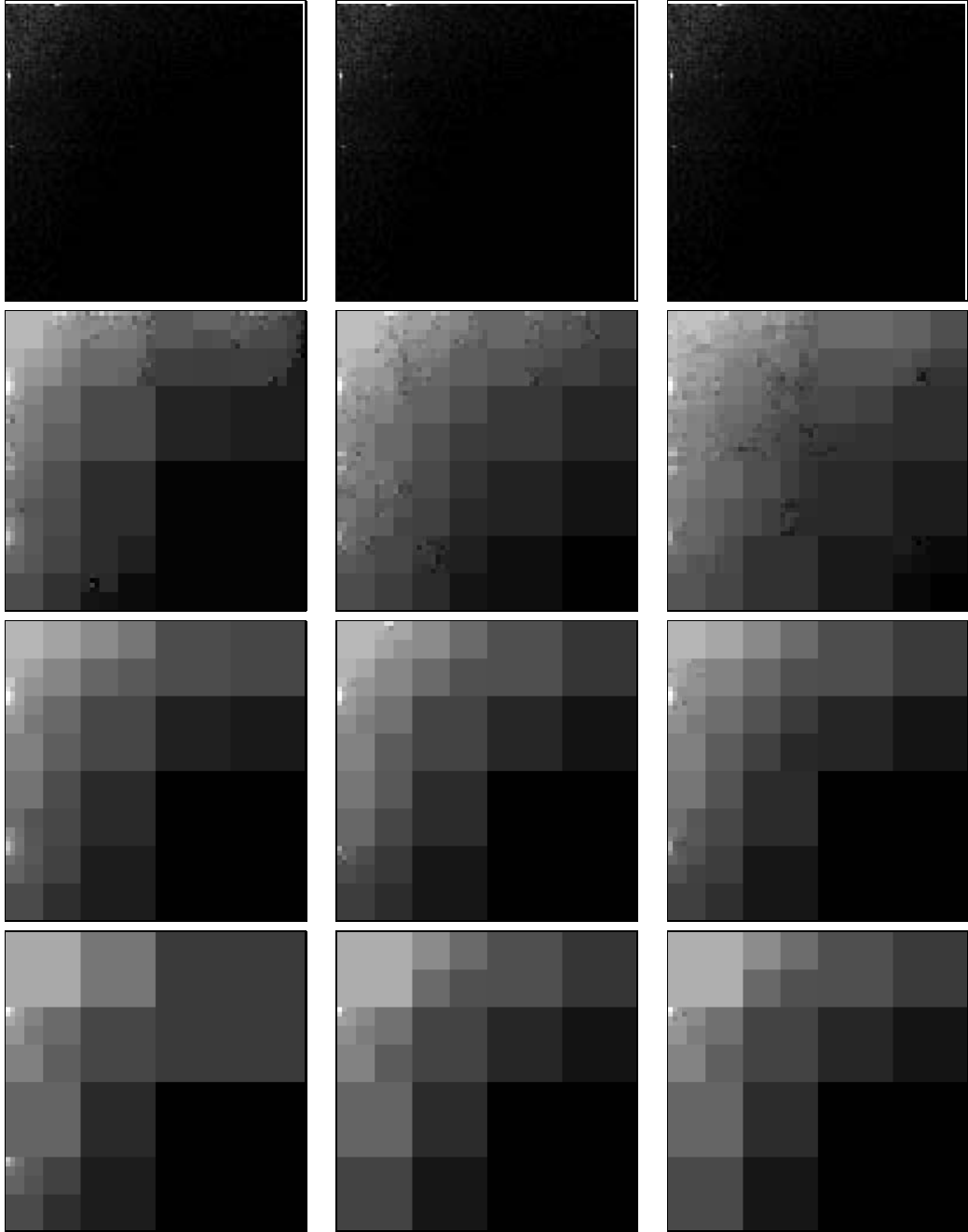


Figure 4.9: Top row: three identical copies of the folded magnitude Fourier transform of the Raffia texture. The nine images below show the optimal decompositions obtained using the Haar, Coiflet1, and D4 wavelets (columns 1, 2, and 3 respectively), and β penalty values of 2000, 5000, and 8000 (rows 2, 3, and 4 respectively).

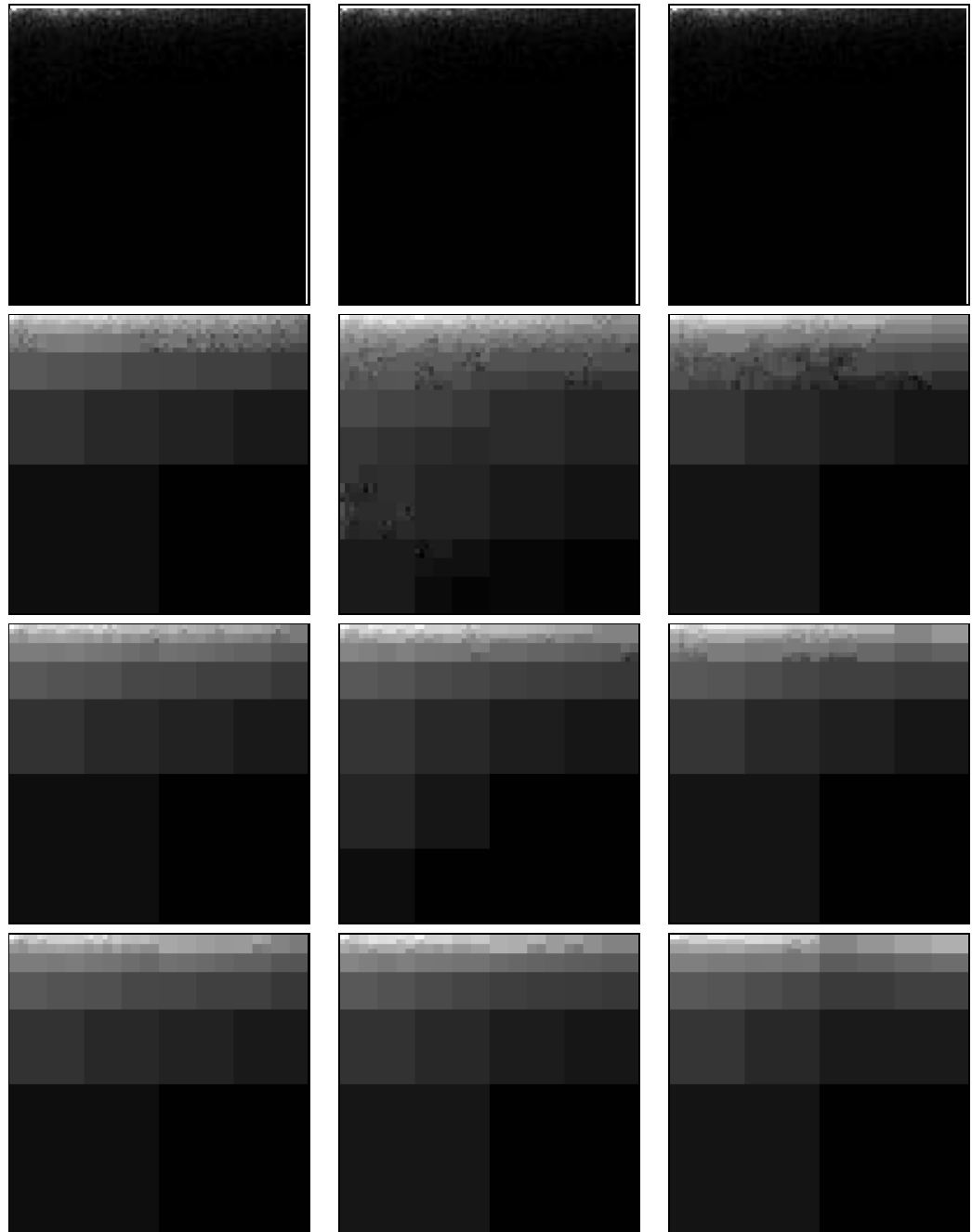


Figure 4.10: Top row: three identical copies of the folded magnitude Fourier transform of the Wood texture. The nine images below show the optimal decompositions obtained using the Haar, Coiflet1, and D4 wavelets (columns 1, 2, and 3 respectively), and β penalty values of 2000, 5000, and 8000 (rows 2, 3, and 4 respectively).

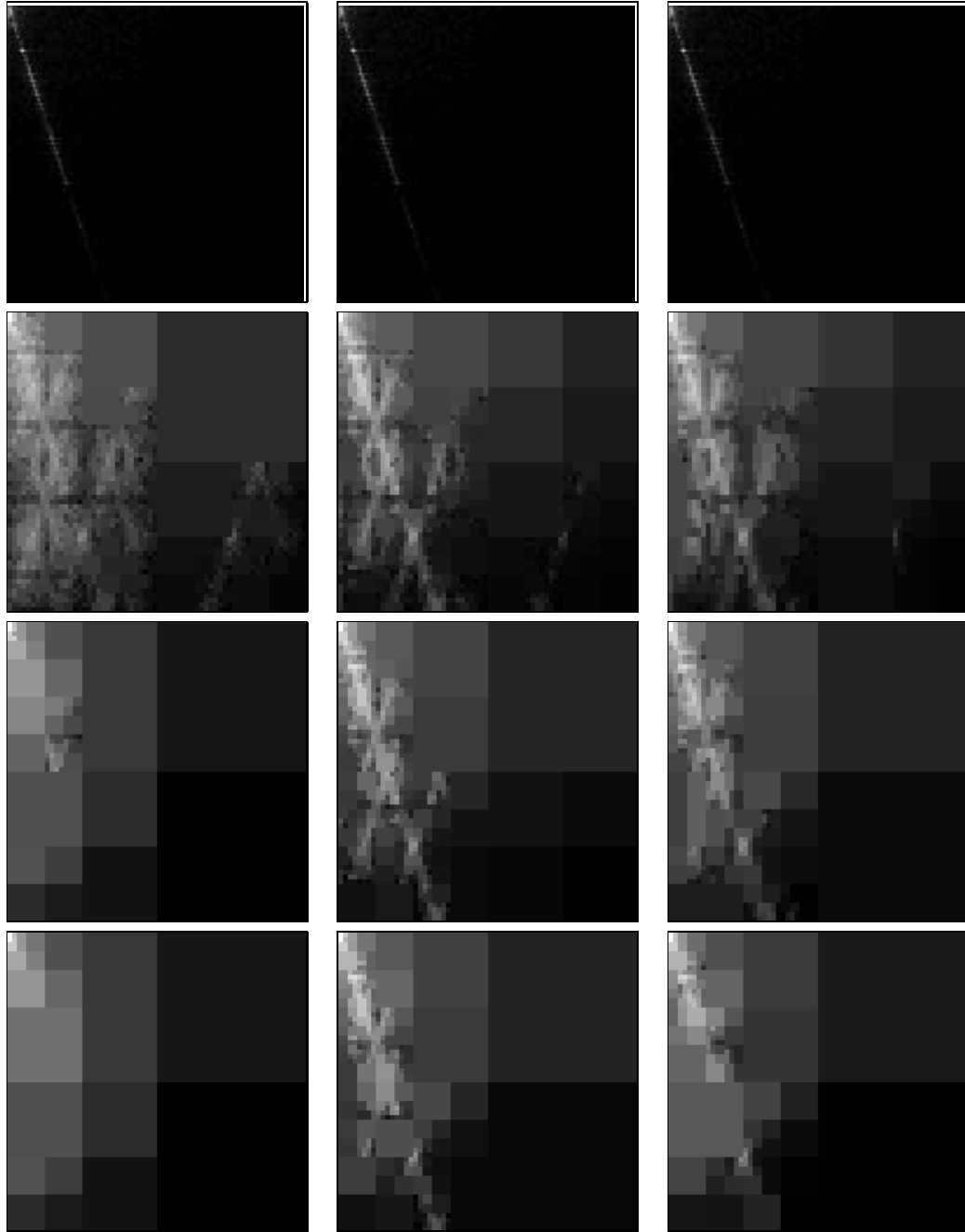


Figure 4.11: Top row: three identical copies of the folded magnitude Fourier transform of the Ploughed texture. The nine images below show the optimal decompositions obtained using the Haar, Coiflet1, and D4 wavelets (columns 1, 2, and 3 respectively), and β penalty values of 2000, 5000, and 8000 (rows 2, 3, and 4 respectively).

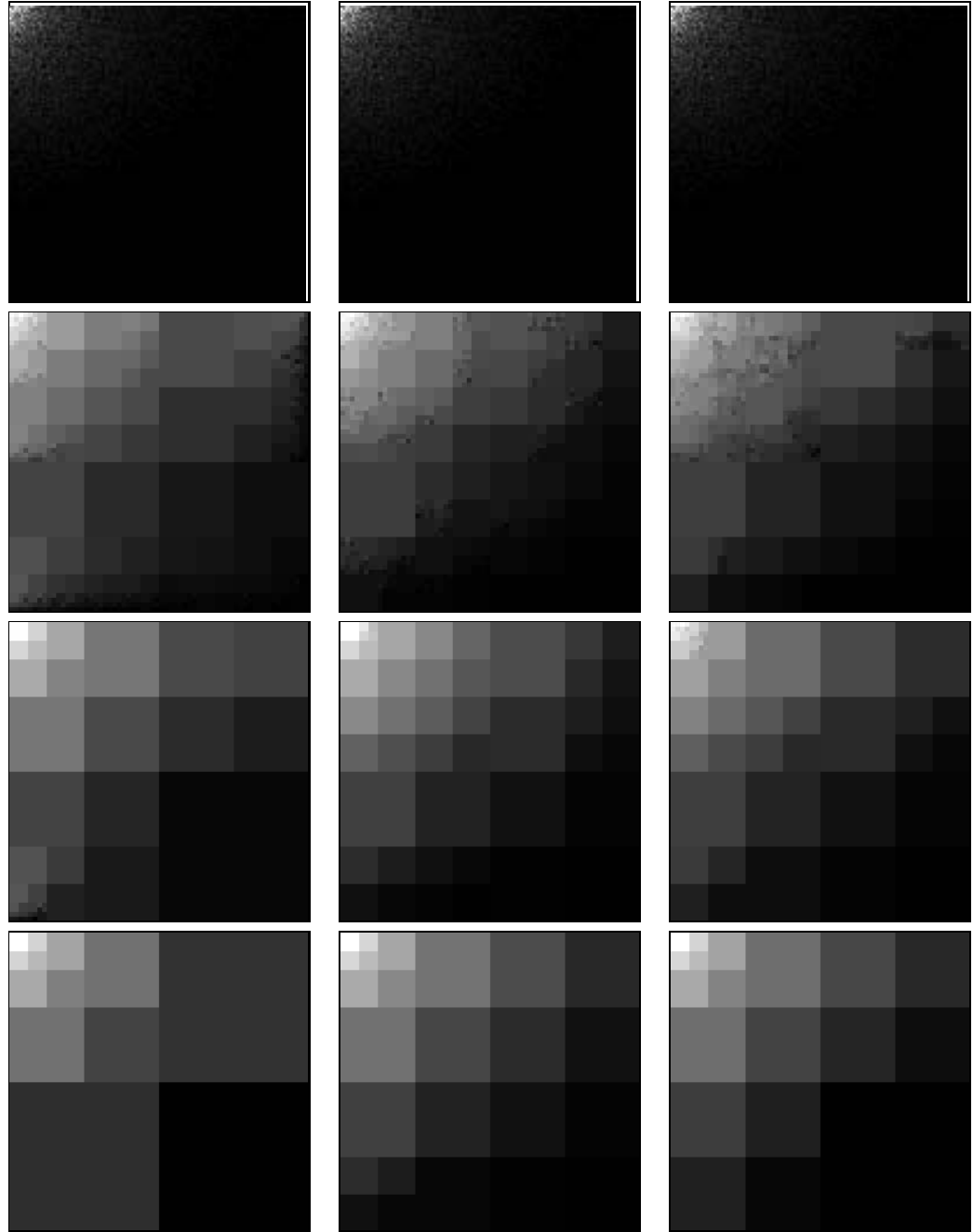


Figure 4.12: Top row: three identical copies of the folded magnitude Fourier transform of the Forest texture. The nine images below show the optimal decompositions obtained using the Haar, Coiflet1, and D4 wavelets (columns 1, 2, and 3 respectively), and β penalty values of 2000, 5000, and 8000 (rows 2, 3, and 4 respectively).

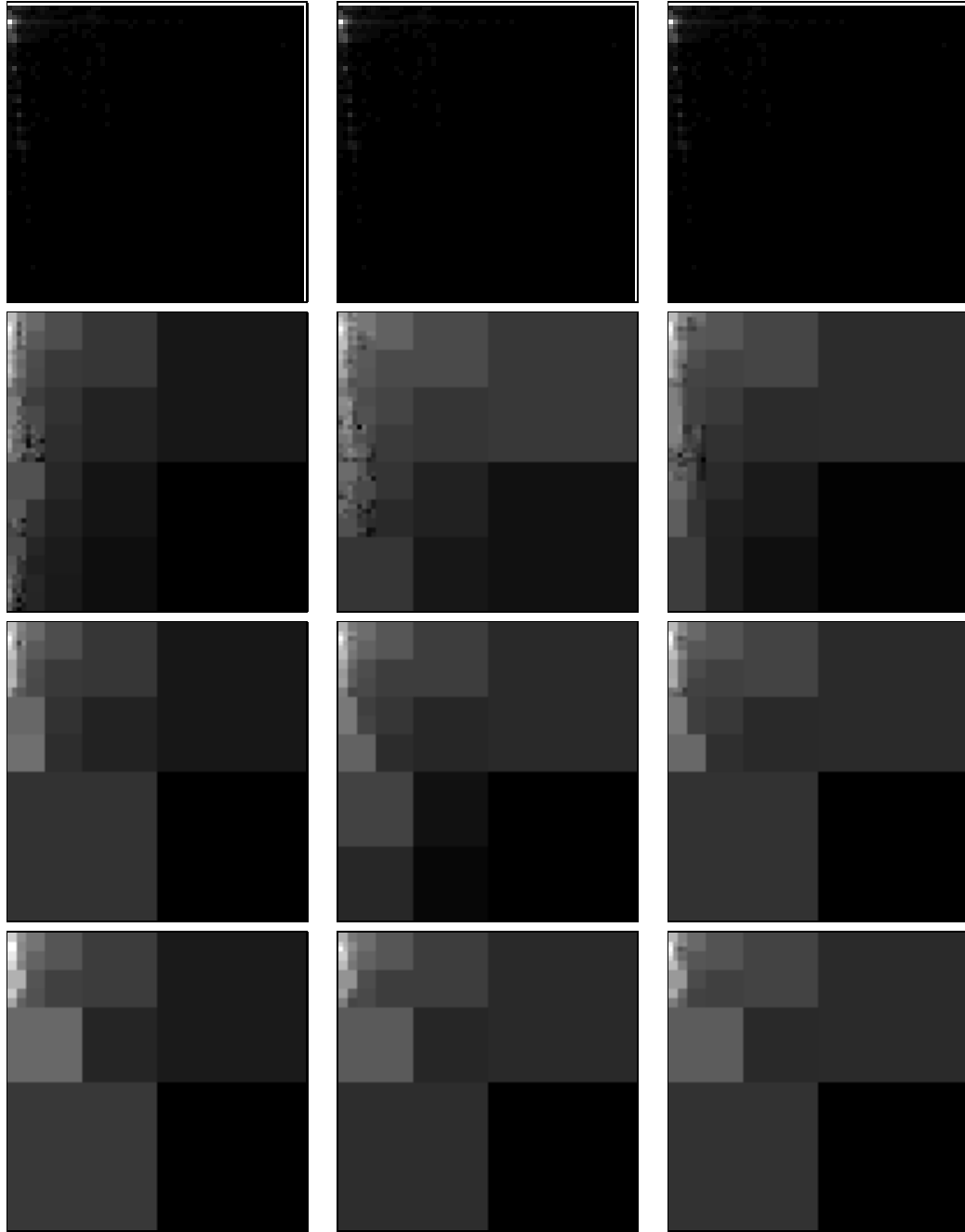


Figure 4.13: Top row: three identical copies of the folded magnitude Fourier transform of the Warehouse texture. The nine images below show the optimal decompositions obtained using the Haar, Coiflet1, and D4 wavelets (columns 1, 2, and 3 respectively), and β penalty values of 2000, 5000, and 8000 (rows 2, 3, and 4 respectively).

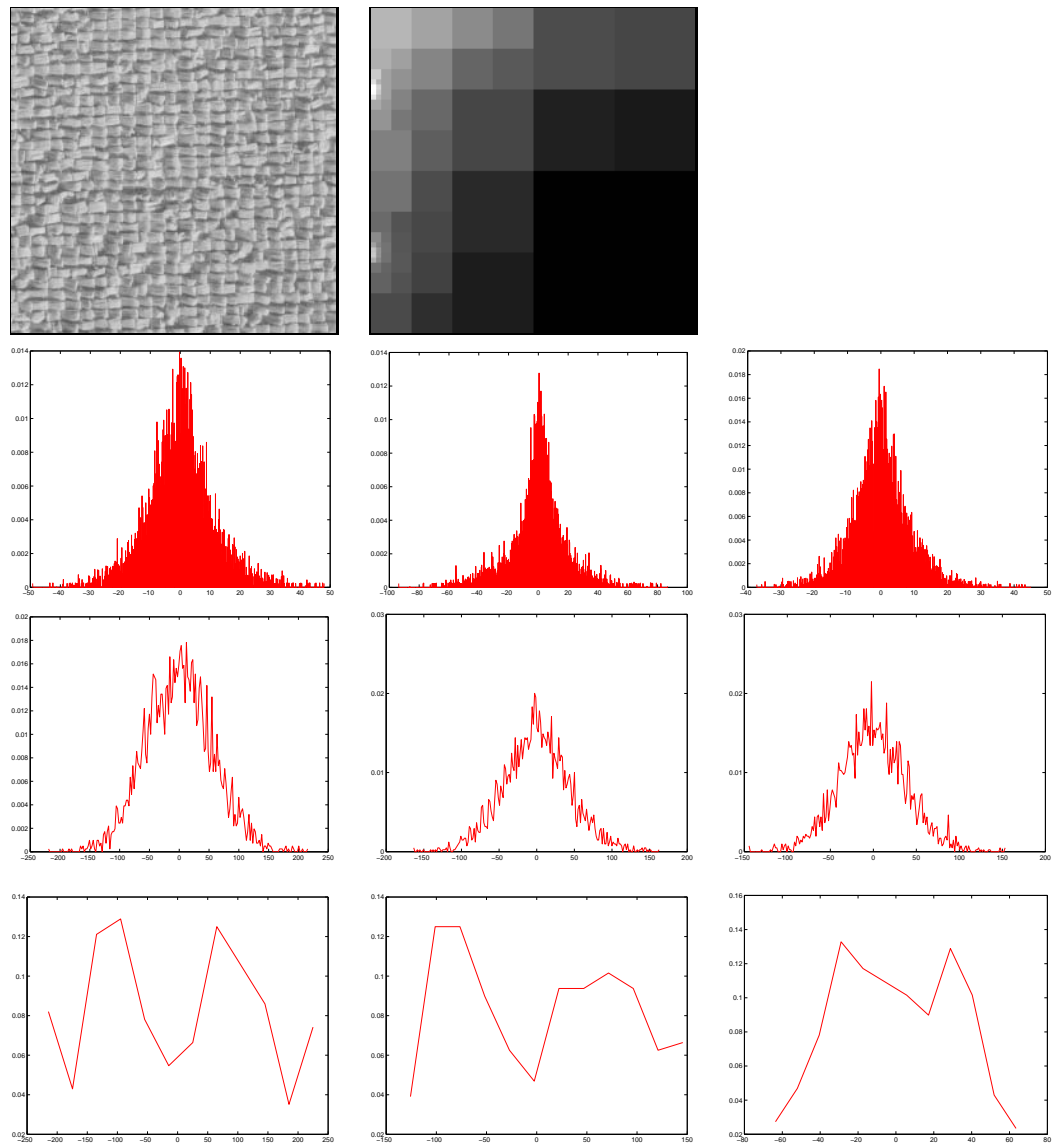


Figure 4.14: Histograms of subbands in an adaptive wavelet packet decomposition. Row1: The original Raffia texture patch and its corresponding adaptive wavelet packet decomposition. Row2: Subband histograms which have a unimodal form corresponding to that of a generalised Gaussian distribution. Row3: Subband histograms which have a unimodal form corresponding to that of a Gaussian distribution. Row4: Subband histograms which have a bimodal form.

packet decomposition, *i.e.* $e_{\alpha,i} = \omega_{\alpha,i}^2$. The normalising constant is given by

$$Z(\mu_\alpha, \sigma_\alpha) = \prod_{\alpha} \prod_{i \in \alpha} \int_0^{+\infty} \frac{1}{\sqrt{e_{\alpha,i}}} \exp \left\{ -\frac{(e_{\alpha,i} - \mu_\alpha)^2}{2\sigma_\alpha^2} \right\} de_{\alpha,i} \quad (4.10)$$

By making the following change of variable

$$z_{\alpha,i} = e_{\alpha,i} - \mu_\alpha$$

the normalising constant becomes

$$Z(\mu_\alpha, \sigma_\alpha) = \prod_{\alpha} \prod_{i \in \alpha} \int_{-\mu_\alpha}^{+\infty} \frac{1}{\sqrt{z_{\alpha,i} + \mu_\alpha}} \exp \left\{ -\frac{1}{2} \left(\frac{z_{\alpha,i}}{\sigma_\alpha} \right)^2 \right\} dz_{\alpha,i} \quad (4.11)$$

By approximating the expression in equation (4.11), it is possible to analytically compute estimates of the distribution parameters, μ_α and σ_α . Under the assumption that μ_α is large relative to σ_α , we make the following approximation [38]:

$$\int_{-\mu_\alpha}^{+\infty} \frac{1}{\sqrt{z_{\alpha,i} + \mu_\alpha}} \exp \left\{ -\frac{1}{2} \left(\frac{z_{\alpha,i}}{\sigma_\alpha} \right)^2 \right\} dz_{\alpha,i} = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\mu_\alpha}} \exp \left\{ -\frac{1}{2} \left(\frac{z_{\alpha,i}}{\sigma_\alpha} \right)^2 \right\} dz_{\alpha,i} \quad (4.12)$$

With this approximation, equation (4.13) reduces to

$$\Pr(\phi | \cdot) = \prod_{\alpha} \prod_{i \in \alpha} \frac{\sqrt{\mu_\alpha}}{\sqrt{2\pi} \sigma_\alpha} \exp \left\{ -\frac{(e_{\alpha,i} - \mu_\alpha)^2}{2\sigma_\alpha^2} \right\} \quad (4.13)$$

The resulting Maximum Likelihood estimates for the distribution parameters are given by

$$\hat{\mu}_\alpha = \sqrt{\frac{\sum_{i \in \alpha} e_{\alpha,i}^2}{N_\alpha}} \quad (4.14)$$

and

$$\hat{\sigma}_\alpha = \sqrt{\frac{2}{N_\alpha} \left(\sum_{i \in \alpha} \omega_{\alpha,i}^2 - \sum_{i \in \alpha} \omega_{\alpha,i} \hat{\mu}_\alpha \right)} \quad (4.15)$$

where N_α denotes the number of pixels in the subband α .

Figure 4.15 shows some of the double peaked histograms plotted against this quartic model. Several of these fit the distribution quite well. We have also fitted a Gaussian distribution to this data to highlight how unsuitable it is for these particular subbands.

From these graphs it is clear that such a quartic model will better represent the subband statistics of the adaptive wavelet packet decompositions.

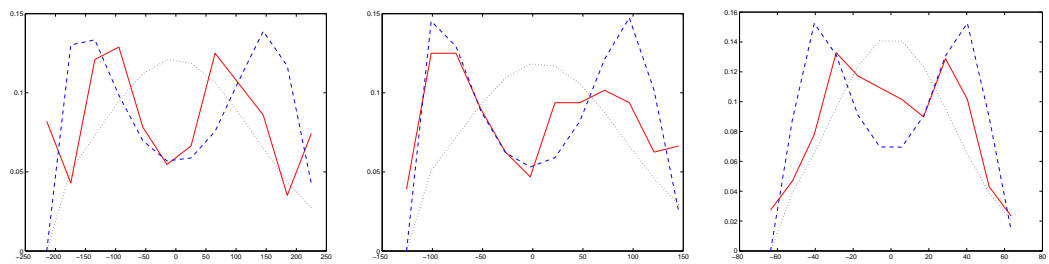


Figure 4.15: Subband histograms from the adaptive wavelet packet decompositions. Gaussian - black (dotted line); Quartic - blue (dashed line); Data histogram - red (solid line);

Chapter 5

Segmenting Textured Scenes

The probabilistic framework developed in the previous chapters allows us to adaptively describe texture in a coherent manner. Models which result from this procedure capture the underlying structure of the textures they represent and have the potential to be used within the context of a supervised texture segmentation scheme. In this chapter, we focus on this problem of partitioning scenes into textured regions and labelling each of the regions according to the real world entity it represents.

By way of introducing the notation which will be used in this chapter, let us consider a toy example of the general segmentation task. Figure 5.1 shows a finite composite image ϕ which we would like to segment.¹ We define a *label set*, denoted by \mathcal{L} , to be



Figure 5.1: An image, ϕ , to be segmented into regions each of which corresponds to the texture of an entity represented in the image.

the finite collection of labels representing the entities that might be present in an image. The three real world entities that gave rise to this particular scene are forest, sand and water. We can denote the label set for this image as:

$$\mathcal{L} = \{F, S, W\} \tag{5.1}$$

The process of classifying an image is carried out via *class map*, $\lambda: D \rightarrow \mathcal{L}$, which

¹Previously, ϕ represented the image defined on the infinite domain D_∞ . For notational simplicity, we redefine the unsubscripted ϕ here to denote the image defined on the finite domain D .

assigns a label to each pixel in the finite image domain D .² Three possible class maps of the image ϕ are presented in figure 5.2. Clearly, some class maps represent more accurate results than others. The goal of the analysis is therefore to find the correct class map, λ^* .

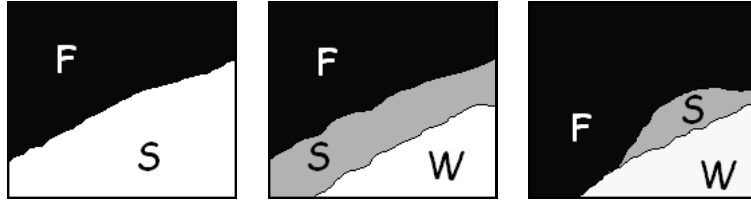


Figure 5.2: Examples of class maps for the image ϕ .

5.1 A Probabilistic Statement of the Problem

In terms of probabilities, the segmentation problem can be stated as follows.

Segmentation Task: Find the class map λ which yields the highest probability for $\Pr(\lambda \mid \phi, B)$. Here $\phi: D \rightarrow C$ denotes the observed image with the image co-domain $C = \mathbb{R}$ or \mathbb{Z} . Information on the textures in the scene is contained in the set B which includes, for each texture class l , the optimal wavelet packet decomposition for describing that texture and the corresponding parameter values.

Maximum A Posteriori (MAP) estimation requires us to find:

$$\lambda^* = \arg \max_{\lambda} \Pr(\lambda \mid \phi, B) \quad (5.2)$$

Using Bayes' Theorem, this posterior probability can be written as

$$\Pr(\lambda \mid \phi, B) = \frac{\Pr(\phi \mid \lambda, B) \Pr(\lambda \mid B)}{\Pr(\phi \mid B)} \quad (5.3)$$

In order to find λ^* , we need to maximise the expression on the right hand side of equation (5.3) over all possible class maps. As the denominator of this expression does not depend on λ , it will not influence the optimisation procedure and so can be ignored in the computation of λ^* . Therefore, we work with the following reduced form of the probability:

$$\Pr(\lambda \mid \phi, B) \propto \Pr(\phi \mid \lambda, B) \Pr(\lambda \mid B) \quad (5.4)$$

²In chapter 3, we introduced λ as the map which assigns labels to pixels in the infinite image. Here we redefine λ to denote such an assignment carried out on the finite image domain D .

The quantity $\Pr(\phi \mid \lambda, B)$ is the likelihood function which relates the observed image ϕ to the values of the conditioning parameters. Any *a priori* knowledge on the class map λ is expressed via the prior probability, $\Pr(\lambda \mid B)$.

Before proceeding with the analysis, a couple of assumptions are made about the relationship between textured regions in the scene.

The first assumption states that, given a classification, pixel values inside a region with a fixed label do not depend on pixel values outside that region. This allows us to express the likelihood function as a product over elements in the label set as follows:

$$\Pr(\phi \mid \lambda, B) = \prod_{l \in \mathcal{L}} \Pr(\phi_{R_l} \mid \lambda, B) \quad (5.5)$$

where $R_l \subset D$ is the set of points in the image domain whose label is l . An assumption of independence at this level is certainly questionable. There are many examples where the pixel values of neighbouring textured regions are correlated. The most obvious of these is the case when the regions are subject to illumination from the same source, be that from the sun, or from an artificial light. Ideally, the effect of this common light source should be built into the description of the textured scene. In order to do this, however, other factors need to be taken into consideration such as the possible presence of clouds and hence shadows in the image. Such issues make it extremely difficult to model the dependencies between pixels in the scene and so the assumption of independence is frequently adhered to in segmentation schemes.

The second assumption states that the probability of the pixel values inside a region with a fixed label does not depend on the class map outside this region. This simplifies the conditioning in the likelihood function to give

$$\Pr(\phi \mid \lambda, B) = \prod_{l \in \mathcal{L}} \Pr(\phi_{R_l} \mid \lambda_{R_l}, B_l) \quad (5.6)$$

where λ_{R_l} is the label map restricted to R_l which, by definition, is constant and equal to l . Here the set B_l contains prior information on the texture l only, *i.e.*

$$B_l = \{ \hat{T}_l, \{ \hat{\mu}_\alpha \}_{\alpha=1}^{S_l}, \{ \hat{f}_\alpha \}_{\alpha=1}^{S_l} \}$$

where S_l denotes the number of leaf-nodes in the optimal wavelet packet decomposition \hat{T}_l . The main issue surrounding this assumption is that it is quite plausible to have a situation wherein knowing the class of one textured region in the scene provides information about a neighbouring region. For example, suppose that roads which are created in forested areas are different from roads which are constructed elsewhere. Knowing that the neighbouring region is forest tells you something about the type of road you have detected. This knowledge is, more often than not, highly scene specific and so is hard to build into a general model. Hence this assumption, although it sometimes ignores information about the scene, is acceptable.

These simplifying assumptions yield the following expression for the posterior probability:

$$\Pr(\lambda \mid \phi, B) \propto \left[\prod_{l \in \mathcal{L}} \Pr(\phi_{R_l} \mid \lambda_{R_l}, B_l) \right] \Pr(\lambda \mid B) \quad (5.7)$$

From this it is clear that in order to carry out the segmentation task we need the following two things

- A **prior probability** for the class map λ and,
- For each texture represented in the label set, a **probability distribution on the finite region**, ϕ_{R_l} .

These two requirements will be discussed in turn in the following sections.

5.1.1 Choosing a Prior for the Class Map

We choose a trivial prior on the class map which assumes independence between pixels in a region and, in addition to this, assigns equal probability to each texture class. This prior is expressed in terms of the pixels inside each region as follows:

$$\Pr(\lambda \mid B) = \prod_{l \in \mathcal{L}} \left(\frac{1}{|\mathcal{L}|} \right)^{|R_l|} \quad (5.8)$$

where $|R_l|$ denotes the number of pixels in the finite region labelled l , and $|\mathcal{L}|$ denotes the number of entities represented in the label set \mathcal{L} . As each pixel in the image must belong to one and only one texture class, the prior probability on the class map for the entire image reduces to a function of the prior probabilities for the class map of the individual pixels in the image

$$\Pr(\lambda \mid B) = \prod_{x \in \phi} \frac{1}{|\mathcal{L}|} \quad (5.9)$$

where x represents a single pixel in the finite image ϕ .

Although this is an unrealistic assumption, the heuristic segmentation scheme we will introduce in section 5.2 will perform a type of average which behaves somewhat like a Potts prior.

5.1.2 Computing the Likelihood

Before calculating the likelihood function for a given region, ϕ_{R_l} , we must first consider the form of the region. Imagine that it is dyadic in shape. In this case, equation (3.37) can be used and calculating the likelihood is a relatively straightforward procedure which is outlined below:

- From the training phase, we have the set B_l containing the optimal wavelet packet decomposition for texture l , as well as the corresponding estimated parameter sets.
- Apply the optimal decomposition to the dyadic-shaped finite region ϕ_{R_l} . This will result in a set of decimated subbands containing the wavelet coefficients which describe the region. For each subband α , the coefficients are denoted by $\omega_{\alpha,i}$, where $i = 1, \dots, N_\alpha$, the number of pixels in the subband.
- Using these coefficients, along with the estimated parameter sets for texture l , we can compute the likelihood for the region as follows

$$\Pr(\phi_{R_l} \mid \lambda_{R_l}, B_l) = \prod_{\alpha=1}^{S_l} \prod_{i=1}^{N_\alpha} \left[\left(\frac{\hat{f}_\alpha}{\pi} \right)^{\frac{1}{2}} \exp \left\{ -\hat{f}_\alpha (\omega_{\alpha,i} - \hat{\mu}_\alpha)^2 \right\} \right] \quad (5.10)$$

This approach is fine if we are sure that we will analyse a dyadic-shaped region. However, in practice, images are usually made up of textured regions which have irregular boundaries. As we already mentioned in section 3.3.2, using a method such as that described above to calculate the likelihood for such regions may lead to errors due to the fact that the wavelets do not form a basis for the region.

We originally considered investigating the possibility of developing an iterative optimisation method which would attempt to satisfy, for an arbitrarily-shaped region, condition 2 of section 3.1.3 exactly. The underlying concept was to generate an initial segmentation of the scene using some heuristic classification step. This rough segmentation could then be used as a starting point for a more precise analysis. For each arbitrarily-shaped region resulting from this segmentation, wavelet basis functions which adapt to the irregular boundary could be created and in conjunction with regular wavelet functions which fit inside the interior of the region, they would provide a basis for the region. Classification would then be performed using this wavelet basis and its result would act as the starting point for the next basis creation step, followed by a new classification. This process would continue until a stopping criterion is reached. There are two major drawbacks with this proposition. Firstly, the new adaptive basis functions at each step would have to be chosen so as to have the same frequency support as the function used to generate the trained wavelet packet decomposition. This is not a trivial condition to satisfy. The second problem is that this iterative scheme would be extremely computationally intensive. For these reasons we have decided not to consider this as a possible solution to the difficulties which arise when dealing with arbitrarily-shaped regions.

So the question which remains is: how do we calculate, in a reasonable manner, $\Pr(\phi_{R_l} \mid \lambda_{R_l}, B_l)$ for a region which is arbitrarily-shaped? We will consider the segmentation of two distinct types of scenes: natural Brodatz texture mosaics and remote sensing images. Due to the difference in component textures in these images, we

treat the two cases separately. The segmentation methods proposed will be described in turn in sections 5.2 and 5.3, respectively.

5.2 Application: Brodatz Mosaics

As discussed before, applying a decimated wavelet transform to the types of textured regions present in these mosaics will lead to errors. We attempt to ameliorate the situation by completing the basis on the region using the following approximate scheme.

Step 1: For each basis function in the optimal wavelet packet decomposition, \hat{T}_l , calculate all possible shifts of it within the region. For example, assume that a basis function is given by a 2×2 filter. This filter should be applied to every single pixel in the region, as is shown in figure 5.3.

Step 2: Take a geometric mean of the probabilities computed from these shifts. With this procedure a displacement of the region with respect to the basis functions will not cause as significant a change in the response of the wavelet transform as would be the case if a decimated wavelet transform is used. Hence it will result in a more translation invariant measure of the energy.

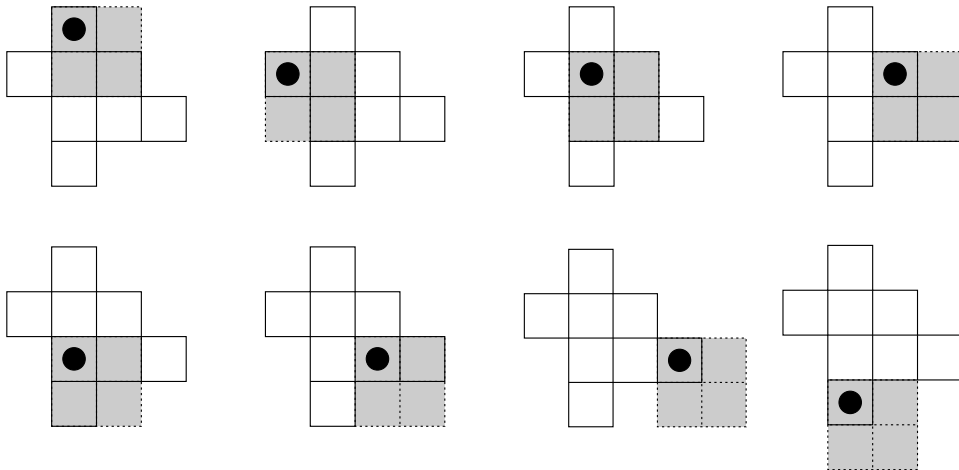


Figure 5.3: Applying the wavelet filter at all possible locations in the arbitrarily-shaped texture patch in order to approximate a basis for the region.

This procedure is equivalent to creating an undecimated wavelet packet decomposition of the region and then averaging the response of each wavelet filter according to the amount of redundancy it contains relative to a decimated wavelet transform. This can

then be used to compute the following approximation to the probability for the region:

$$\Pr(\phi_{R_l} \mid \lambda_{R_l}, B_l) \simeq \prod_{\alpha=1}^{S_l} \prod_{x=1}^{|R_l|} \left[\left(\frac{\hat{f}_\alpha}{\pi} \right)^{\frac{1}{2M_\alpha}} \exp \left\{ -\frac{\hat{f}_\alpha}{M_\alpha} (\omega_{\alpha,x} - \hat{\mu}_\alpha)^2 \right\} \right] \quad (5.11)$$

where M_α , the *redundancy factor* for subband α , captures the redundancy present in the undecimated representation of the region as demonstrated previously in figure 4.2 of section 4.2. Note that this distribution is not the same as that found by pretending that the coefficients in the undecimated wavelet decomposition are independently distributed.

Although this technique was motivated by a need to approximate a basis for the region, it also has the great advantage of creating a pixelwise likelihood expression. This results from the fact that in an undecimated transform each pixel in the region has a corresponding wavelet coefficient in each subband. Hence we can swap the order of the products in equation (5.11) to get

$$\Pr(\phi_{R_l} \mid \lambda_{R_l}, B_l) \simeq \prod_{x=1}^{|R_l|} \prod_{\alpha=1}^{S_l} \left[\left(\frac{\hat{f}_\alpha}{\pi} \right)^{\frac{1}{2M_\alpha}} \exp \left\{ -\frac{\hat{f}_\alpha}{M_\alpha} (\omega_{\alpha,x} - \hat{\mu}_\alpha)^2 \right\} \right] \quad (5.12)$$

For notational simplicity, we summarise the product over subbands by

$$\prod_{\alpha=1}^{S_l} \left[\left(\frac{\hat{f}_\alpha}{\pi} \right)^{\frac{1}{2M_\alpha}} \exp \left\{ -\frac{\hat{f}_\alpha}{M_\alpha} (\omega_{\alpha,x} - \hat{\mu}_\alpha)^2 \right\} \right] = H_l \left[\{\omega_{\alpha,x}\}_{\alpha=1}^{S_l} \right]$$

Hence this procedure results in a likelihood and prior term which can be written in terms of individual pixels in the image. This means that we can express our posterior probability as

$$\Pr(\lambda \mid \phi, B) \propto \prod_{x \in \phi} \frac{1}{L} H_l \left[\{\omega_{\alpha,x}\}_{\alpha=1}^{S_l} \right] \quad (5.13)$$

and so it is possible to perform a pixelwise Maximum Likelihood (ML) classification of the image ϕ .

In practice, we know that λ is likely to be somewhat regular. One way to introduce this, of course, is to define a Potts prior and use simulated annealing to make a Maximum A Posteriori (MAP) estimate of λ . However, as is well known, simulated annealing is very slow, and it turns out that another approach produces results that are as good, if not better, while being considerably more computationally efficient. We keep the trivial prior as defined in section 5.1.1 and use the following classification rule:

$$\lambda^*(x) = \arg \max_{l \in \mathcal{L}} \prod_{x' \in V(x)} H_l \left[\{\omega_{\alpha,x'}\}_{\alpha \in T} \right] \quad (5.14)$$

where $V(x)$ is the set of neighbours of pixel x , which includes x itself. Examples of neighbourhoods can be seen in figure 5.4. This rule has a similar effect to the Potts prior, but it still allows a pixelwise classification because it uses the data at the neighbours of a pixel but not their unknown classes. In consequence, one can use larger neighbourhoods with little extra penalty.

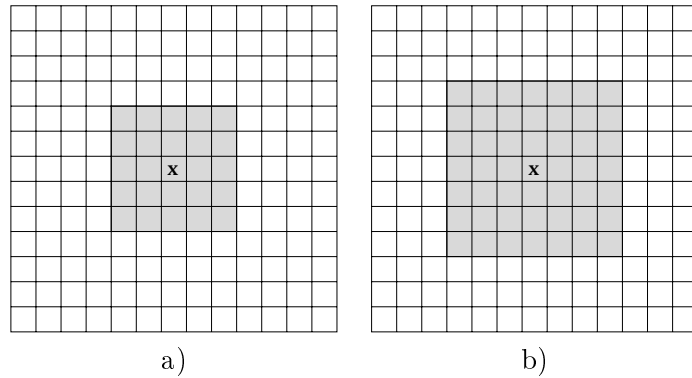


Figure 5.4: a) Neighbourhood-5 and b) Neighbourhood-7 schemes for pixel x

5.2.1 Results

This segmentation method was tested on several 512×512 Brodatz texture mosaics. The results can be seen in figures 5.5 and 5.6. Misclassification percentages are given in tables 5.1 and 5.2, respectively.

	Figure 5.5(i)	Figure 5.5(j)	Figure 5.5(k)	Figure 5.5(l)
Error	2.6%	1.7%	2.5%	2.4%

Table 5.1: Number of misclassified pixels given as a percentage of the total number of pixels in the mosaics of figure 5.5.

	Figure 5.6(i)	Figure 5.6(j)	Figure 5.6(k)	Figure 5.6(l)
Error	1.8%	1.9%	1.5%	3.1%

Table 5.2: Number of misclassified pixels given as a percentage of the total number of pixels in the mosaics of figure 5.6.

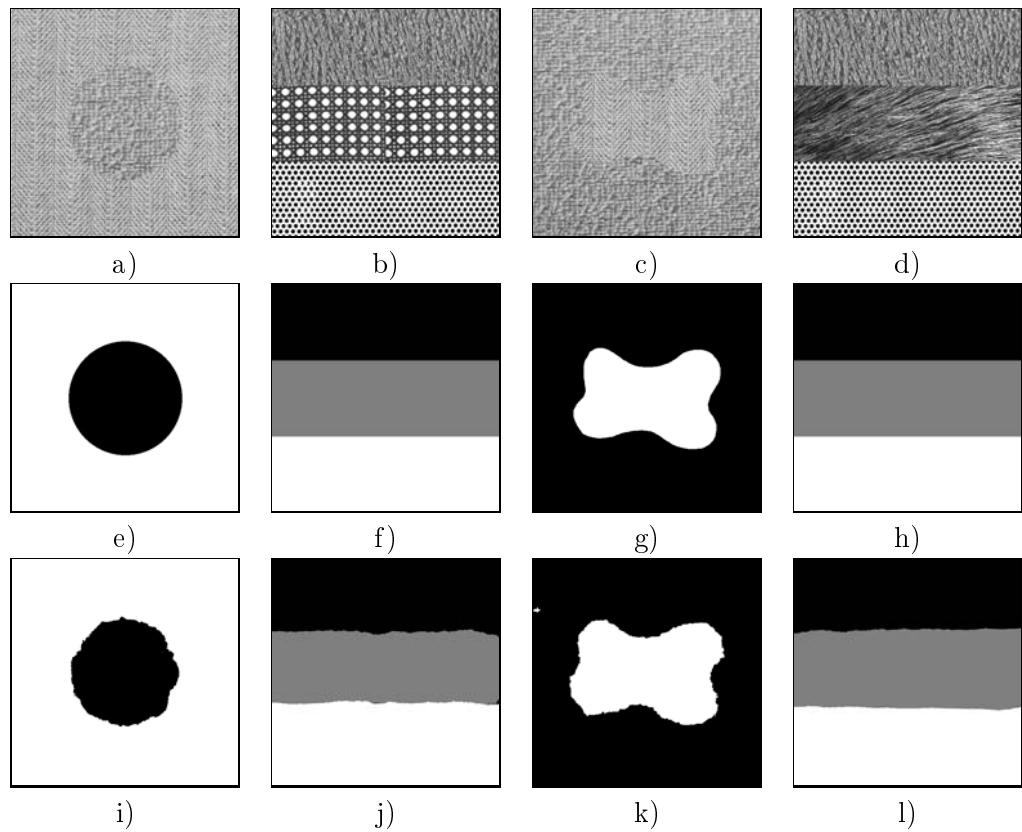


Figure 5.5: Row 1: Original mosaics: a) Circular mosaic of Herring and Raffia; b) Rectangular mosaic of Calf, D102, and Hexholes152; c) Freehand mosaic of Herring and Raffia; d) Rectangular mosaic of Calf, Fabric0004, and Hexholes152. Row 2: Their ground truth images. Row 3: Segmentation results.

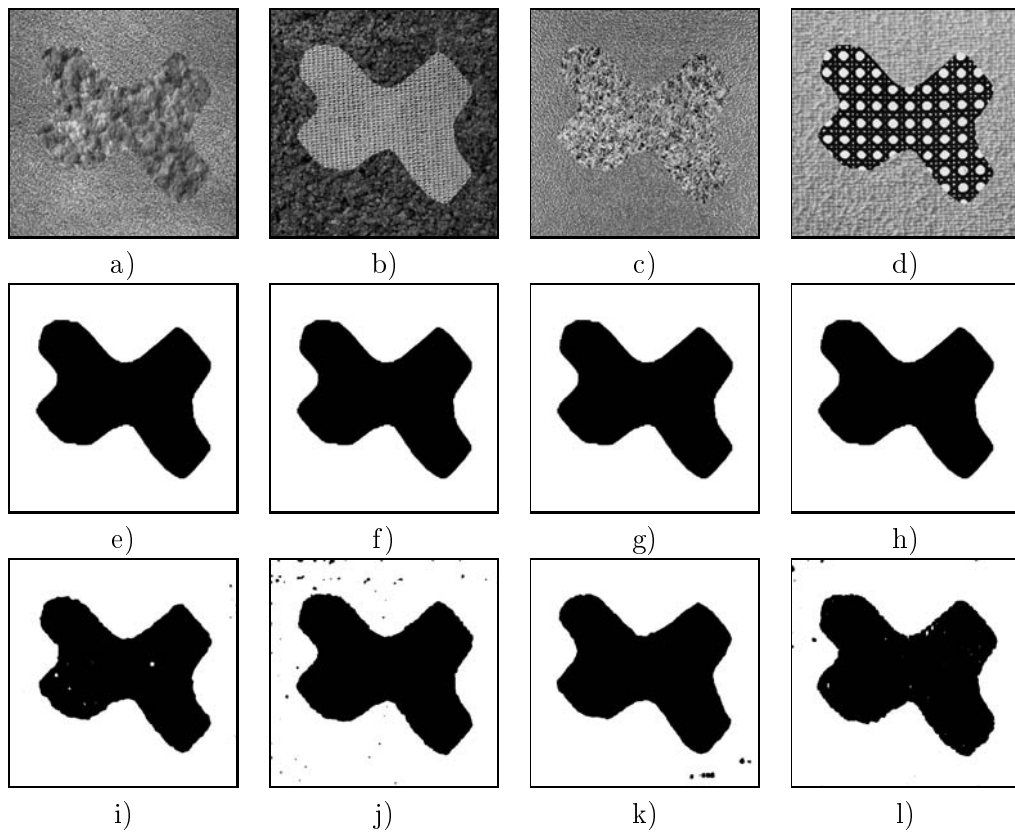


Figure 5.6: Row 1: Original freehand mosaics of textures: a) Fabric0015 and Fabric0018; b) Fabric0017 and Image38; c) Grass and Metal0000; d) D102 and Raffia. Row 2: Their ground truth images. Row 3: Segmentation results.

5.2.2 Comparison With a Gibbs Texture Model

We have tested the performance of our method for synthetic data against another supervised texture segmentation scheme developed by Gimel'farb [35]. This approach, based on a conditional Gibbs model, describes the spatial structure of a grayscale texture mosaic via the pixelwise and pairwise pixel interactions in a fixed neighbourhood. These interactions appear in the form of Gibbs potential functions in the exponent of the conditional probability distribution of the region map, given its corresponding grayscale mosaic. The model captures not only the texture properties of a single homogeneous textured region, but also the inter-region structure present in the mosaic.

When presented with a new test mosaic to segment, a likelihood maximization algorithm (Controllable Simulated Annealing) is used to minimise the probabilistic distance between the first-order and second-order statistics for the training data (mosaic and region map) and the mosaic to be segmented. Some comparison results for the original mosaics of figure 5.5 are presented in figure 5.7.

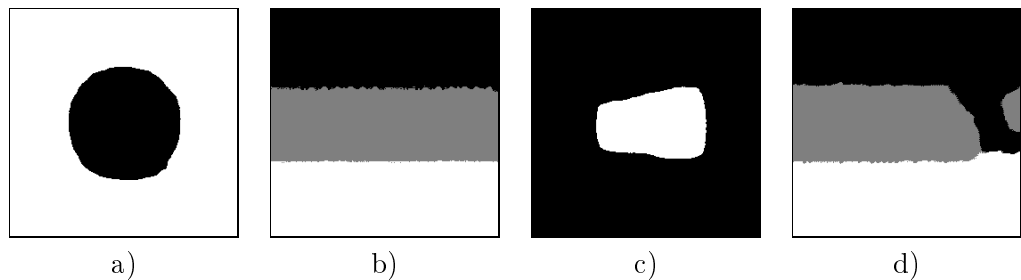


Figure 5.7: Comparison results from the method described in [35]. Images provided by Dr. Georgy Gimel'farb of the University of Auckland, New Zealand.

It should be noted that the method in [35] was tested on scaled down (by a factor of two) versions of the textures used in our experiments. The training mosaics used can be seen in figure 5.8. Results for this method could probably be improved if full scale images are used.

However, even with the results on the scaled down test images, one can see where the two methods differ. Firstly, the presence of a texture which varies in brightness and/or directionality poses a problem for the method in [35]. The fourth mosaic of figure 5.5 has a centre strip containing the texture Fabric0004. We can see where this texture changes toward the right hand edge of the mosaic. It is apparent from the segmentation results of figure 5.7 that the algorithm had difficulty with this particular area of the mosaic. The training procedure of our method, which uses a collection of sample patches from each texture, allows us to avoid such problems at the segmentation phase.

Secondly, the method in [35] relies heavily on the geometric structure of the training

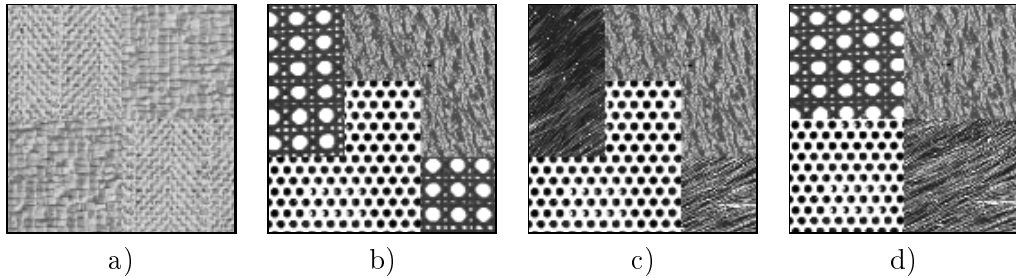


Figure 5.8: Training mosaics used in the method presented in [35]. Images provided by Dr. Georgy Gimel'farb of the University of Auckland, New Zealand.

mosaics which consist of rectangular regions. This structure created particular problems for the third test mosaic. However, we can also see how the inclusion of structure in the training process has perhaps enhanced the segmentation of the rectangular shaped test mosaics. As our method trains each texture separately, there is no inter-region structure present in our models. This might explain the lack of precision at the boundaries in our results on the rectangular mosaics, but it also allows us the freedom to segment mosaics with all kinds of inter-region structure present, such as the freehand and circular test mosaics presented above.

5.2.3 Comparison With a Discriminative Feature Selection Approach

We have shown above that our method is highly competitive against a texture classification method which belongs to the Markov Random Field class of texture models. It is also useful to compare our method against a technique which takes a spatial frequency filtering approach to texture analysis. In this section we discuss such a method and present some results obtained by applying it to our mosaics in figure 5.6.

One of the filtering methods presented in [5] uses a Gabor-like filter bank to extract information from the texture. An image, $f(\mathbf{x})$, is decomposed into a series of subbands, $s_j^l(\mathbf{u})$ where j indexes the subband and l denotes the level at which the subband lies in the decomposition of the image. These subbands are computed via the following expression

$$s_j^l(\mathbf{u}) = \Lambda_j^l \left[g \left(\mathbf{u}; \mathbf{m}_j^l, \Sigma_j^l \right) F(\mathbf{u}) \right] \quad (5.15)$$

where the isotropic Gaussian function $g(\mathbf{u})$ has a centre frequency of \mathbf{m}_j^l and a bandwidth given by Σ_j^l , the discrete Fourier transform of the image is denoted by $F(\mathbf{u})$, and Λ_j^l represents a set of band-limiting operators.

Having performed such a decomposition on a textured mosaic, the underlying goal of the method proposed in [5] is to find the subset of subbands in this decomposition which

maximally discriminates between the textures present in the mosaic. This subband selection procedure is motivated by the need to reduce the number of features used to classify the mosaic. It is also considered to help in avoiding the selection of subbands which may potentially confuse the classification procedure.

Choosing this subset of features is carried out as follows: A prototype of each texture present in the mosaic is selected. Each prototype is subjected to the same decomposition as that performed on the mosaic. A distance measure is then used to judge the similarity of matching subbands in the decomposition of each prototype. This allows a decision to be made regarding the relevance of a particular subband for the purposes of classification. The subbands selected are ranked according to their ability to discriminate between the textures present in the mosaic and they are then applied within a simple k-means classification scheme using the mean features from the prototypes as the class centres.

This technique was tested on the two-texture Brodatz mosaics shown previously on the first row of figure 5.6. For a given mosaic, a level 3 decomposition was carried out which resulted in 19 subbands. A single 64×64 prototype was selected for each texture in the mosaic. The ten best subbands were chosen using a Bhattacharyya distance measure. Several classification experiments were carried out using an increasing number of features each time. The results with the lowest error rate are shown in figure 5.9. The error rates are recorded in table 5.3 along with the corresponding number of features used to perform the classification.

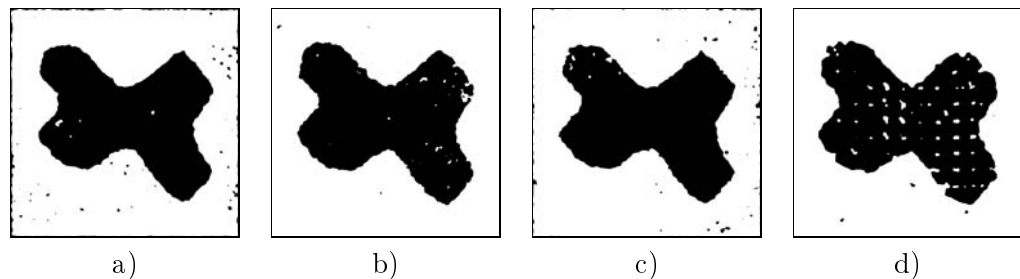


Figure 5.9: Comparison results obtained from the method described in [5]. These images (a)–(d) correspond to the mosaics (a)–(d), respectively, in figure 5.6. Images provided by Dr. Abhir Bhalerao of the University of Warwick, U.K.

This method produces similar results to those obtained with our method. The main advantage with this particular approach is that it uses only a relatively small number of features to carry out the classification. However, it is important to realise that with this method one cannot have a description of a single texture as such. Instead features are also chosen relative to another texture. Adding a new texture into a scene might therefore pose problems as the most discriminating features would need to be recalculated as opposed to just adding in one more texture model as is the case with

	Figure 5.9(a)	Figure 5.9(b)	Figure 5.9(c)	Figure 5.9(d)
Error	2.6%	1.7%	1.7%	4.3%
# Features	8	5	6	10

Table 5.3: Row 1: Number of misclassified pixels given as a percentage of the total number of pixels in the mosaics of figure 5.9. Row 2: Number of features (out of a total of 19) used to obtain these error rates.

the texture segmentation scheme developed in this thesis.

The results presented in this application section highlight the success of our adaptive texture models for the segmentation of handmade mosaics which consist of natural textures taken from the Brodatz album. Although this collection is made up of a wide variety of patterns, including many forms of micro and macro texture, it represents only a tiny subset of the observable textures present in our visual world. In order to truly test the capability of our models as texture descriptors, we need to use them as an analysis tool for other types of textures. In the next two sections we test our models on a selection of remote sensing images and wildlife scenes.

5.3 Application: Remote Sensing Images

The automatic detection of regions in remote sensing images is often helped by the inclusion of textural information from the scene. A steadfast improvement in the quality of imaging equipment has recently lead to images which boast a high resolution of the captured scene. In these types of images, texture is an easily accessible feature and so they present us with an excellent opportunity to evaluate our models.

To this end, we have cropped several areas from large remote sensing images. They are shown in the first row of figure 5.10. This selection includes a 1-metre resolution aerial image, two 1-metre resolution satellite images captured by the IKONOS satellite, and one lower resolution synthetic aperture radar (SAR) image. For each image in turn, we decided on a number of classes corresponding to entities in the images. This information is recorded in table 5.4. Based on these classes, we manually created ground truth images which can be seen in the second row of figure 5.10. A separate model was trained for each texture class according to the procedure described in chapter 4. It is clear from these images that the individual textured regions are far from dyadic in shape and so we are left in the same position as before, *i.e.* needing to analyse arbitrarily-shaped regions. The approximate probabilistic segmentation scheme developed in section 5.2 was tested on these images. It did not produce very satisfactory classifications. As a result of this we decided to investigate the use of a simple and heuristic classification method which is described below.

We start by noting that each texture model has a largest effective filter size, cor-

responding to the smallest elements in the Fourier domain partition. This filter size is dyadic. For example, optimal decompositions which involve a level three analysis of the frequency domain, have a largest effective filter size of eight. Given a set of texture models then, there is a largest effective filter size among them all. To classify a single pixel x in the image domain, we examine an image patch centred on x of size equal to this largest filter size among all models. We can calculate the probability of the image restricted to this dyadic patch for each trained texture model using equation (3.37), as was previously outlined at the beginning of section 5.1.2. Assuming that the prior on the class map λ is uniform on \mathcal{L} , these data probabilities can be normalized to give the posterior probabilities for the class of the patch. The MAP estimate of the class of the patch is then given, tautologically, by the class with maximum posterior probability. Rather than assign this class to every pixel in the patch however, we heuristically assign this class only to the pixel x at the centre of the patch. We then repeat this procedure for every pixel in the image.

The danger of such a pointwise procedure is that the class map λ can be irregular. However, as in section 5.2, we use the heuristic procedure that has a similar effect to a regularizing prior: it smooths the class map and shortens boundaries. This is done by combining the posterior probability distributions on \mathcal{L} at each pixel, as in equation (5.14).

5.3.1 Results

The results of this heuristic classification procedure are shown in the third and fourth rows of figure 5.10. From these images we can see that the segmentation method performs quite well. The major classification errors occur near the boundary between textures, as would be expected. However it is worth noting that in some cases the classification is, in a sense, more accurate than the ground truth provided for the scene. The obvious examples of this are the segmentation results of the SAR image in figure 5.10(c) and the satellite image in figure 5.10(d). In the case of the former, it is clear from the original image that there is a small mountainous region in the middle of the plain. This is ignored by the ground truth. However it is picked up by the segmentation method and correctly classified as belonging to the mountainous texture as can be seen from the black inside the white region. A similar thing has happened in the satellite image where a tree in the ploughed field is detected as not belonging to the ploughed texture. Here though it is incorrectly classified as belonging to the border texture class. In the following sections, we present two measures which can be used to evaluate and better understand the results of the segmentation task.

5.3.1.1 The Kappa Value

It is standard practice among the remote sensing community to assess the accuracy of a segmentation procedure using a statistic known as the *kappa value* [15, 17, 18]. The

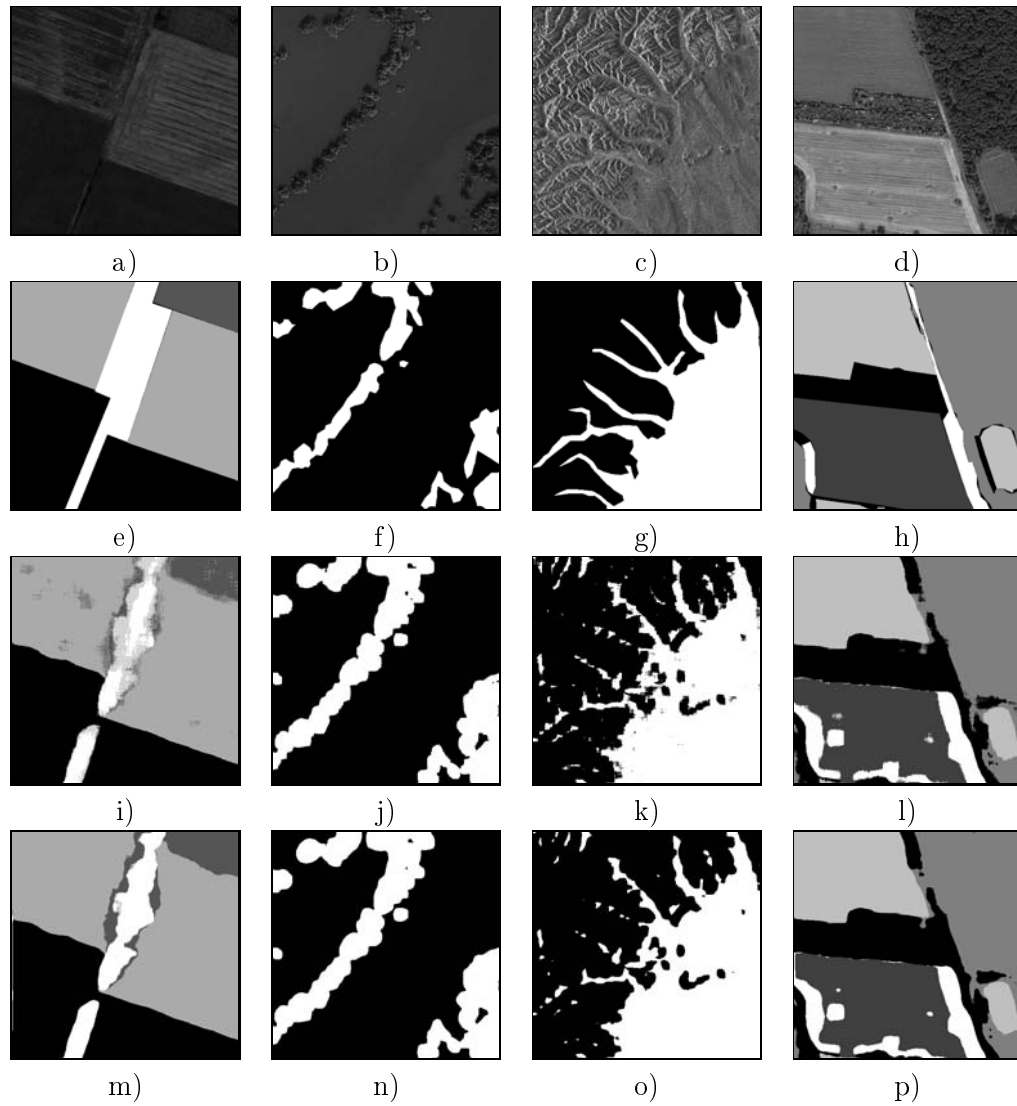


Figure 5.10: Original images: a) Aerial, 1m resolution ©IGN, b) IKONOS satellite, 1m resolution ©Space Imaging, c) SAR, d) IKONOS satellite, 1m resolution ©Space Imaging; Manually created ground truth images: e) – h); Neighbourhood-1 classifications: i) – l); Neighbourhood-7 classifications: m) – p).

	Classes (in order of increasing grey level)
Figure 5.10(a)	Field, Scrub, Ploughed, Border
Figure 5.10(b)	Flood, Trees
Figure 5.10(c)	Mountains, Plain
Figure 5.10(d)	Scrub, Ploughed, Woodland, Field, Border

Table 5.4: Classes defined for the ground truth images in figure 5.10.

concept behind this approach is to measure the degree to which two “judges” agree in their respective sorting of the pixels in an image into the L mutually exclusive groups available. For this particular test dataset, one judge is the person who created the ground truth image and the other judge is the supervised texture segmentation routine we have developed. In order to calculate the kappa value for a given segmentation result, we must first construct its *error matrix*. For each class in the corresponding ground truth image, this matrix records the number of pixels from this class which were labelled as belonging to each of the L possible classes. The error matrix for the leftmost neighbourhood-1 segmentation result presented in figure 5.10 is shown in table 5.5.

CLASS	Field	Scrub	Ploughed	Border	Row Sum
Field	71320	439	8779	1849	82387
Scrub	0	10879	2156	428	13463
Ploughed	0	1276	101365	44	102676
Border	632	7173	5728	18341	31874
<i>Column Sum</i>	71952	19758	118028	20662	230400

Table 5.5: Error matrix for the segmentation result shown in figure 5.10(i).

The kappa value is given by the following expression:

$$\kappa = \frac{N \sum_{i=1}^r x_{ii} - \sum_{i=1}^r x_i \cdot x_{\cdot i}}{N^2 - \sum_{i=1}^r x_i \cdot x_{\cdot i}} \quad (5.16)$$

where x_{ii} is the i^{th} diagonal element in the error matrix; x_i is the sum of the entries in the i^{th} row; $x_{\cdot i}$ is the sum of the entries in the i^{th} column; r denotes the total number of rows in the error matrix; and N is the total number of pixels in the image under consideration. Values range from -1 (no agreement between the two judges at all) to $+1$ (complete agreement between the judges), with a value of 0 being interpreted as no agreement above that expected from a random classification. Kappa values for the eight segmentation results shown in figure 5.10 are presented in table 5.6.

	Figure 5.10(a)	Figure 5.10(b)	Figure 5.10(c)	Figure 5.10(d)
Nbd-1	0.81	0.70	0.74	0.78
Nbd-7	0.81	0.68	0.75	0.78

Table 5.6: Kappa values for the segmentation results presented in figure 5.10.

An important point to comment on here is the fact that the ground truth images in this test dataset were generated by a non-expert individual.³ This automatically introduces an element of uncertainty into the process of evaluation by the kappa value. One might wonder what true meaning lies behind a kappa value which compares the results of our segmentation method to such a subjective representation of the ground truth. In response to this, it is worth mentioning that the methods usually employed to create ground truth images for such applications are also highly subjective in nature. A panel of experts may know more about the classes present in the image, but due to the inherent complexity of adjacent classes in the scene, their ability to precisely define the borders of these classes can be quite restricted. Comparisons between on-site measurements and expert ground truth produced by hand from the corresponding images resulted in error rates of up to 10%. This seems to suggest that perhaps the kappa value can never really be a reliable measure of the accuracy of a segmentation method. Whatever one’s opinion happens to be on this issue, it is always wise to bear in mind that the kappa value compares the ability of a method to something which is ultimately subjective and so the confidence one places in its reliability as an accuracy measure is directly related to one’s confidence in the ground truth.

5.3.1.2 Entropy Measurements

Due to this issue surrounding the kappa value, we attempt to provide a more intuitive understanding of how our segmentation method performs on such remote sensing data. We propose to use entropy information extracted from our segmentation procedure as a measure of the strength of each pixel class assignment. The details of this technique are given below.

As was discussed in section 5.2, in order to carry out a pixelwise classification of the image we need to calculate, for each pixel in the image, the probability that it comes from each possible texture in turn. These pixelwise posterior probabilities are computed using equation (5.14) with a 1x1 neighbourhood, and are denoted here by q . Hence

$$q_l(x) = H_l[\{\omega_{\alpha,x}\}_{\alpha \in T}] \quad (5.17)$$

represents the posterior probability that pixel x in the image ϕ comes from texture class l . As a first step towards calculating entropy measures, we need to normalise

³The Linux based image software GIMP was used to outline the different classes present in the remote sensing images.

these probabilities over classes in order to obtain a normalised distribution on the class of textures. The normalised probabilities are given by:

$$p_l(x) = \frac{q_l(x)}{\sum_{k \in \mathcal{L}} q_k(x)} \quad (5.18)$$

We may now calculate the entropy associated with pixel x as follows:

$$S(x) = - \sum_{l \in \mathcal{L}} p_l(x) \log p_l(x) \quad (5.19)$$

In understanding what this measure tells us, we need to examine its limiting cases. The first such case arises when the segmentation method is absolutely sure that pixel x belongs to a particular class, say $l = 2$. Hence we have that $p_2(x) = 1$ and $p_l(x) = 0$ for all $l \neq 2$. Setting $\log 0 = 0$, it is clear that we end up with an entropy measure of 0. The other extreme occurs when the segmentation method considers all texture classes as having equal probability. In this situation we end up with an entropy of $\log L$ as demonstrated below:

$$S(x) = - \sum_{l \in \mathcal{L}} \frac{1}{L} \log \frac{1}{L} = \log L \quad (5.20)$$

For each pixel in the neighbourhood-1 classification results, we have calculated its entropy and rescaled the resulting values to lie within the range $[0, 255]$. This produces an image with black pixels indicating a high degree of confidence in the label choice, white pixels indicating a high level of uncertainty in the label choice, and a range of gray-level values in between representing pixels which were assigned labels with a varying degree of certainty. A colour-bar representation of this confidence in label choice is shown in figure 5.12. Entropy images for each of the neighbourhood-1 classifications in figure 5.10 can be found in figure 5.11. Here we can see that the interior of most of the entity regions is black in colour, suggesting that the major source of uncertainty in our method lies primarily at the boundaries between entity regions, as would be expected. The entropy measure provides us with a better indication of the effect of the advice given by the model to a human expert, because that advice will be ignored if the model is uncertain about its decision.

We have also generated an entropy-weighted misclassification image for each of the neighbourhood-1 classifications. These images are presented in figure 5.13. The value of a pixel x is given by

$$e^{-S(\pi_x)} [2\delta(\lambda(x), \lambda^*(x)) - 1] \quad (5.21)$$

where λ is the algorithm output, and λ^* is the manually created ground truth (note that e^S is analogous to the number of classes that were confused by the algorithm). Thus, after rescaling the resulting values to lie within the range $[0, 255]$, those values greater than 128 represent ‘‘correct’’, with certainty increasing with increasing intensity,

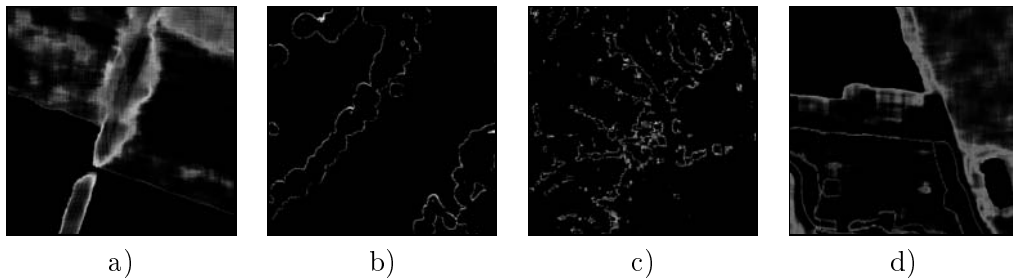


Figure 5.11: Entropy images, (a)–(d), corresponding to the neighbourhood-1 segmentation results in figure 5.10 (i)–(l), respectively.



Figure 5.12: The colour-bar indicating the confidence in labelling decision associated with each colour in the entropy images of figure 5.11.

while those values less than 128 represent “incorrect”, with certainty increasing with decreasing intensity. The entire colour-bar and its ranges are shown in figure 5.14.

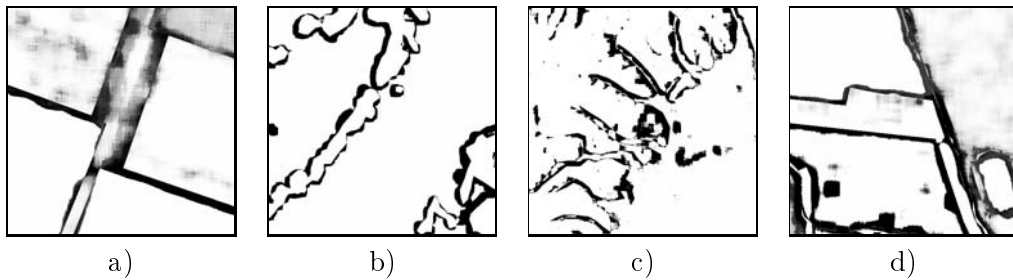


Figure 5.13: Entropy-weighted misclassification images, (a)–(d), corresponding to the segmentation results in figure 5.10 (i)–(l), respectively.

5.4 Application: Wildlife Scenes

For the final test dataset we have chosen a couple of images of wild animals in their natural habitat. The original images are shown in figure 5.15. The textures present in these images are quite different from the ones encountered in the previous sections. This is mainly due to the fact that in each scene there is a primary texture which is

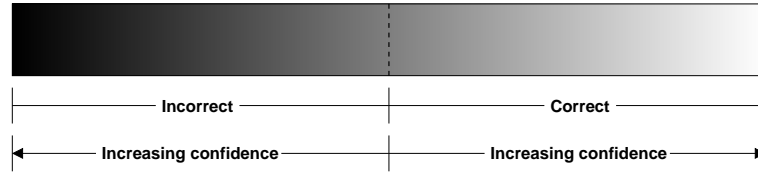


Figure 5.14: The colour-bar indicating the confidence in labelling decision associated with each colour in the entropy-weighted misclassification images of figure 5.13.

inherently linked to a distinct predefined object. This makes it relatively easy to agree upon the type of textures present in the image and their general location.

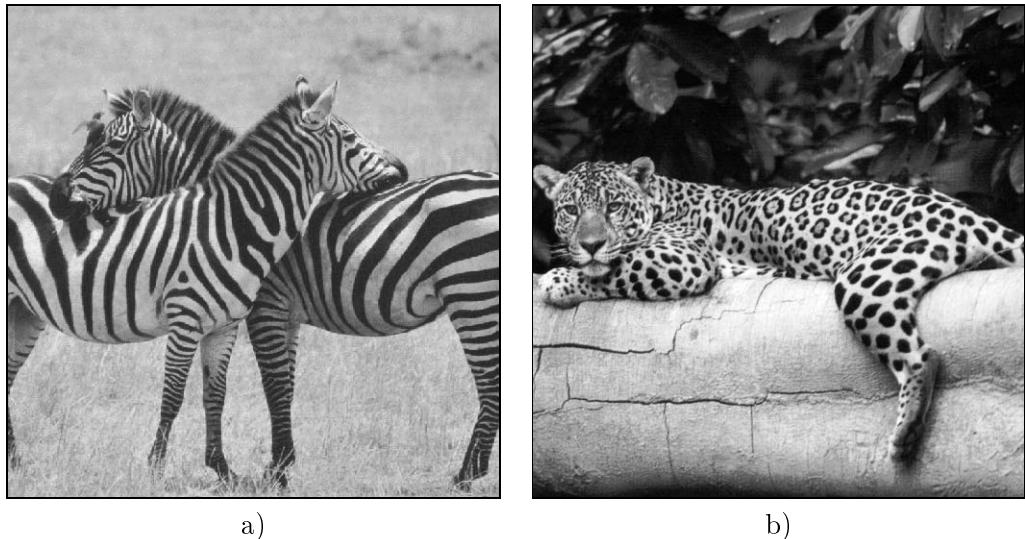


Figure 5.15: Two wildlife images of size 512×512 consisting of textures a) Background and Zebra and b) Leaves, Jaguar, and Trunk. Images provided by Dr. Abhir Bhalerao of the University of Warwick, U.K.

The image in figure 5.15(a) can be split into two regions, one corresponding to the striped texture of the zebra fur and the other representing the background which in this case is the grass in which the animals are standing. A manually generated ground truth for this image is shown in figure 5.16(a). In the right hand image we have three textures to model: the spotted texture of the jaguar fur, the bark of the tree trunk, and the leaves above the head of the animal which form the background of this particular scene. The manually created ground truth for this image is shown in figure 5.16(b). Most of the background texture of figure 5.15(a) is behind the animals and so is out of focus. However in the lower half of the image a sharper focus on this texture introduces

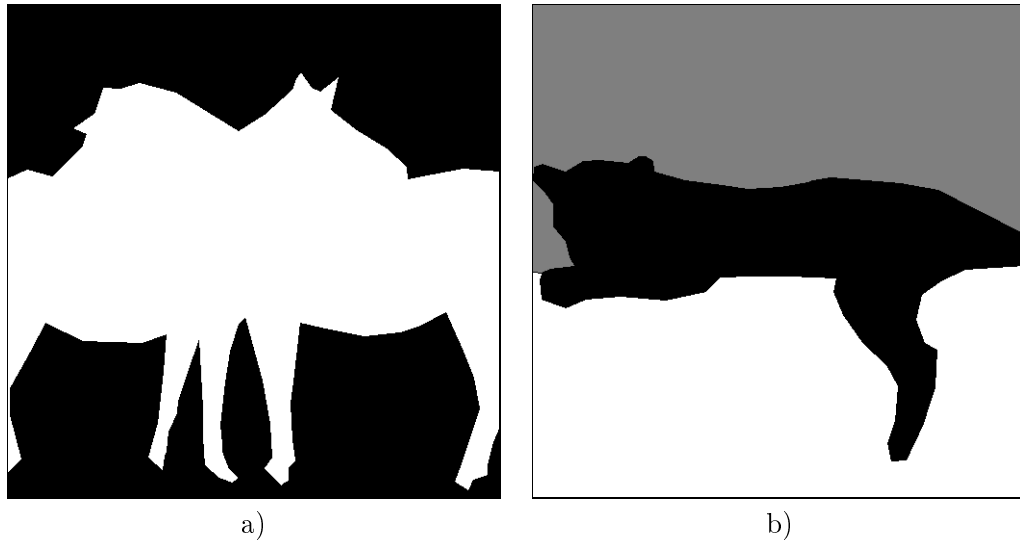


Figure 5.16: Manually created ground truth for the images shown in figure 5.15.

an element of structure. Patches chosen to model this texture were selected so as to have an even amount of both in focus and out of focus background samples. Several of these patches are shown in figure 5.17.



Figure 5.17: A selection of the patches of size 64×64 which were used to train the model for the Background texture.

Most people will happily identify all parts of the zebra as belonging to the same texture. That is why it was so easy to draw an approximate ground truth map for this particular image. More than likely, this is due to the fact that we know, *a priori*, the shape of the animal and hence we use this additional information to determine what we should consider as zebra texture. As easy as this task may seem to the human eye, a closer examination of the texture reveals some potential pitfalls for an automatic texture classification scheme. Although the texture can loosely be described as being comprised of white and black stripes, these stripes often possess a degree of curvature and they change direction depending on their location on the body of the animal. The distance between the stripes also varies considerably. We have attempted to train a single model for this Zebra texture using a combination of patches which include all these variants.

Examples are shown in figure 5.18.



Figure 5.18: A selection of the patches of size 64×64 which were used to train the model for the Zebra texture.

Similar reasoning lies behind the selection of the patches for the textures in figure 5.15(b), examples of which are presented in figures 5.19, 5.20, and 5.21. The background of this image is particularly difficult to model. It is severely out of focus which results in a possible loss of structure from the Leaves texture. It is clear from the segmentation results that this blurring affects the modelling ability of our method.

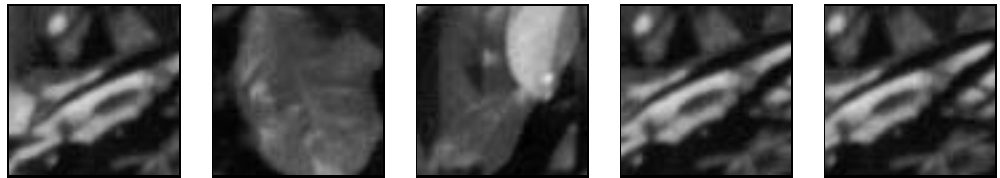


Figure 5.19: A selection of the patches of size 64×64 which were used to train the model for the Leaves texture.

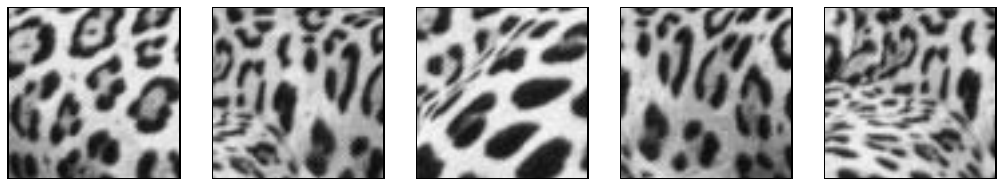


Figure 5.20: A selection of the patches of size 64×64 which were used to train the model for the Jaguar texture.

The segmentation method described in section 5.2 was used to carry out classifications on these images. Results are shown in figure 5.22. The first row presents the pixelwise classifications obtained. As the neighbourhood size increases, the zebra image starts to approximate the ground truth. However as the pixelwise result for the jaguar image was so poor, increasing the neighbourhood size does not improve matters.

As can be seen from the results, the classification of the texture corresponding to



Figure 5.21: A selection of the patches of size 64×64 which were used to train the model for the Trunk texture.

the animal in each scene is good. This is quite interesting as the directionality and scale in these textures change a lot depending on what part of the animal is being analysed.

5.5 Discussion: The Effect of the Mother Wavelet

During the training phase described in chapter 4, we noticed that the resulting adaptive decompositions were different depending on which mother wavelet was used. This suggests that the choice of mother wavelet is an important step in the modelling process.

This notion of one wavelet being more suitable than another for the purposes of signal analysis is not new. In [36] several measures have been proposed to select the best mother wavelet. One example tested a number of different mother wavelets from the Daubechies family. An artificial dataset was constructed so as to have a structure similar to that of the Daubechies2 mother wavelet. Random noise was then added to this test signal to make the job of finding the best mother wavelet harder. An entropy measure, based on normalised wavelet coefficients, was computed for the wavelet transform resulting from each mother wavelet applied to the signal. This measure identified the Daubechies2 mother wavelet as being the best wavelet for analysing the data in the signal.

Extending this selection procedure to more complicated signals and images is not obvious. The structure of the data no longer mimics that of one of the analysing mother wavelets. This is clearly an issue for using such wavelet selection techniques for complicated textures such as those analysed in this thesis.

Early on in the literature not much consideration was given to the issue of choosing the correct mother wavelet for the purposes of texture analysis. The general opinion seemed to be that the particular form of the mother wavelet used in the analysis of textures did not have any real significant effect on the classification results of images containing these textures [11, 30]. However, in recent years, this attitude has changed and there is now a growing realisation that the quality of the classification can often be strongly related to the mother wavelet used to analyse the texture [62].

In order to test the influence of the mother wavelet, we trained models for a selection of textures from the Brodatz album using the following three mother wavelets: Haar, Daubechies4, and Coiflet1. For each texture, 64 patches of size 128×128 were selected

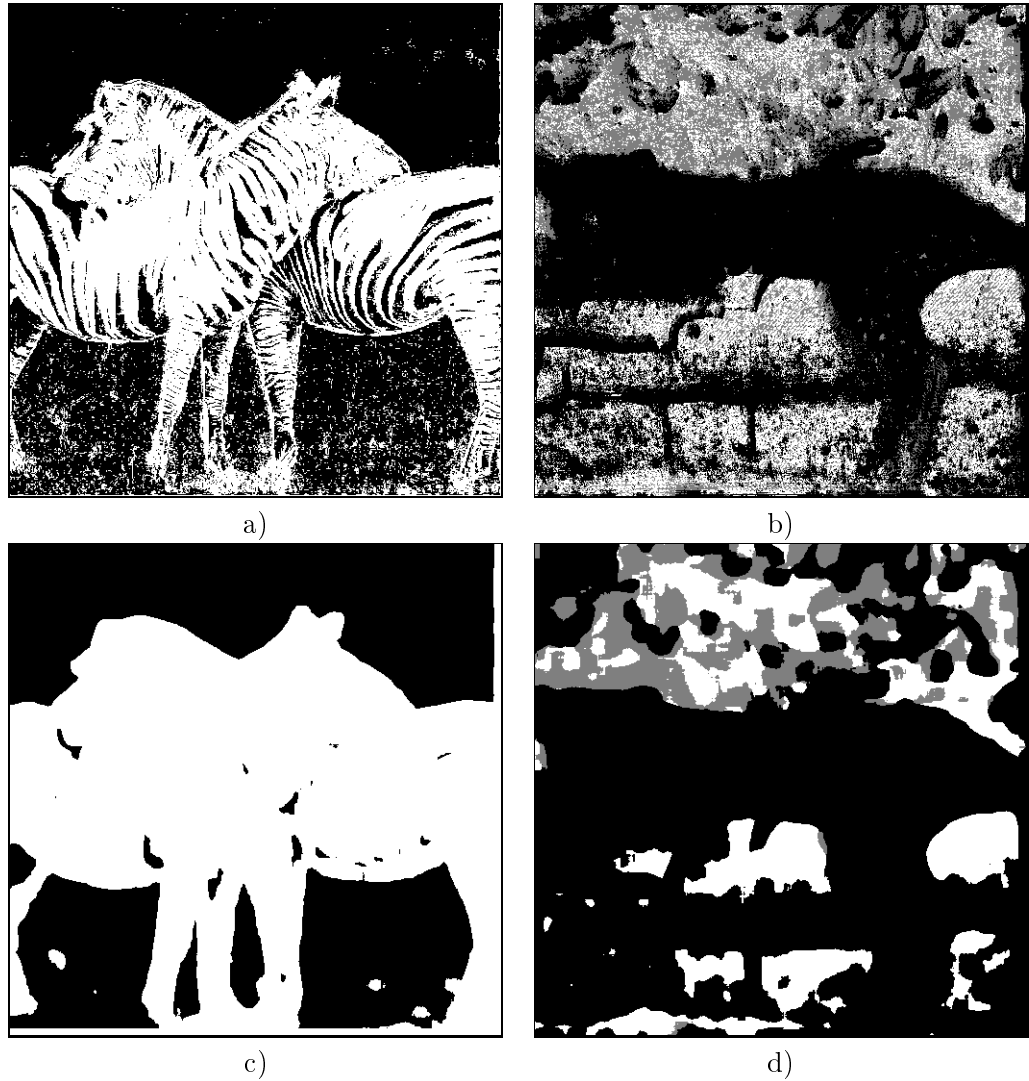


Figure 5.22: Classification results for the two images in figure 5.15: a) and b) neighbourhood-1; c) and d) neighbourhood-15

and trained, as detailed in chapter 4, with the maximum decomposition level set to 6 and a penalty of $\beta = 64000$. Examples of some of the resulting adaptive decompositions are shown in figure 5.23.

Mosaics containing these textures were segmented as in section 5.2. A size 1 neighbourhood was chosen so as to remove the smoothing effect of the neighbourhood scheme which might prevent us from evaluating the texture capabilities of our models.

Some classification results obtained using the different mother wavelet models are shown in figure 5.24. The error rates for these classifications are provided in table 5.7.

	Daubechies4	Haar	Coiflet1
D102Raffia	12.7%	14.7%	11.1%
Fabric0000Fabric0004	18.1%	18.9%	11.6%
HerringLeaves0012	11.9%	14.6%	9.8%

Table 5.7: Errors in the classification of the mosaics presented in figure 5.24 using a selection of different mother wavelets. The number of misclassified pixels is given as a percentage of the original mosaic.

It appears that the Coiflet1 mother wavelet performs best in situations where macro-textures are being analysed. This has undoubtedly something to do with the fact that it has the largest filter length among the mother wavelets being tested. However, it is also clearly better at picking up the directionality in the Fabric0004 texture in the centre mosaic. These differences in classification results warrant further investigation into the effects that different mother wavelets have on texture analysis. This issue will be discussed more in the following chapter.

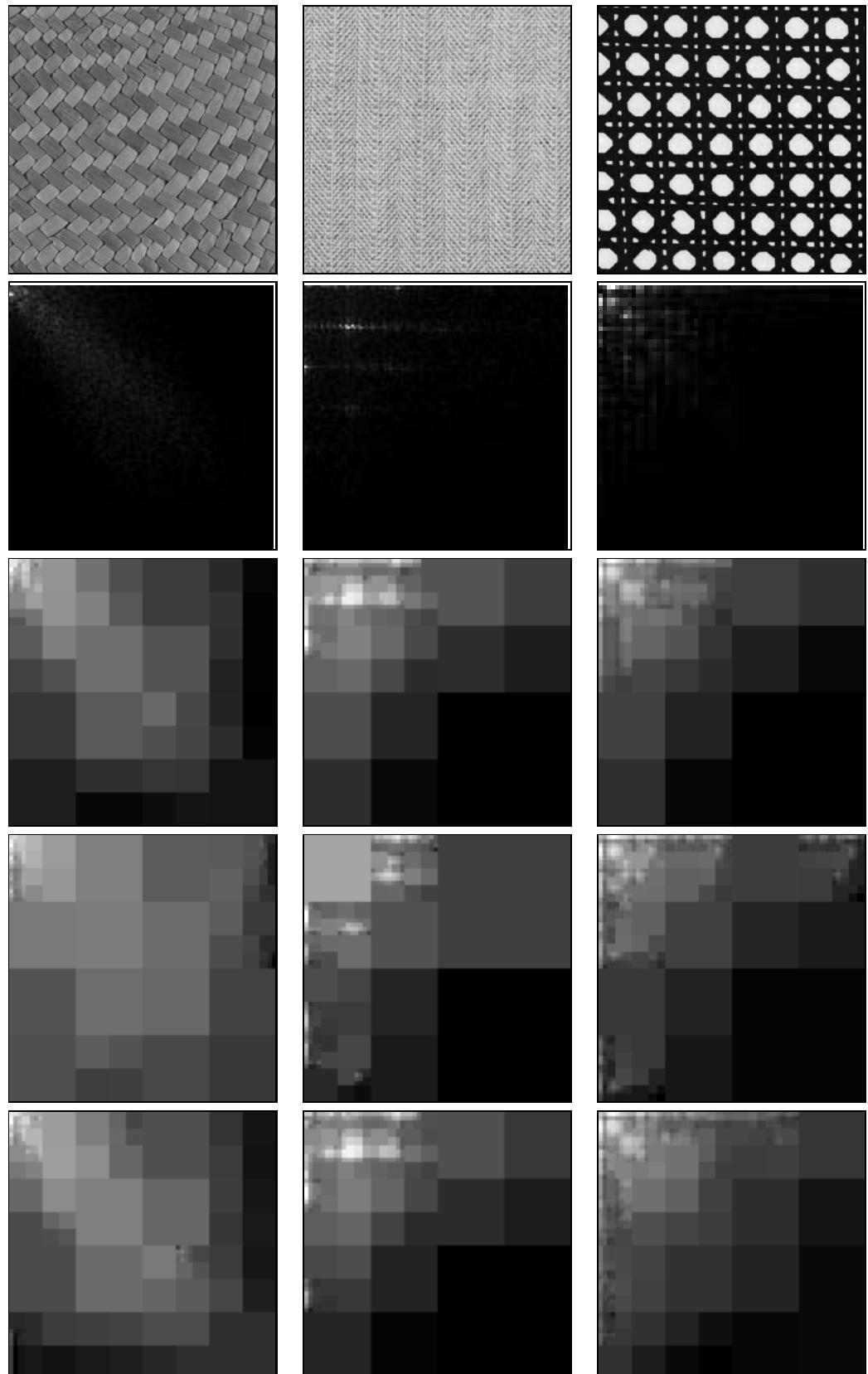


Figure 5.23: Original textures Fabric0000 (left), Herring (middle) and D102 (right); Folded Fourier magnitude images (row2); Adaptive decompositions generated using a mother wavelet of type Daubechies4 (row3), Haar (row4) and Coiflet1 (row5).

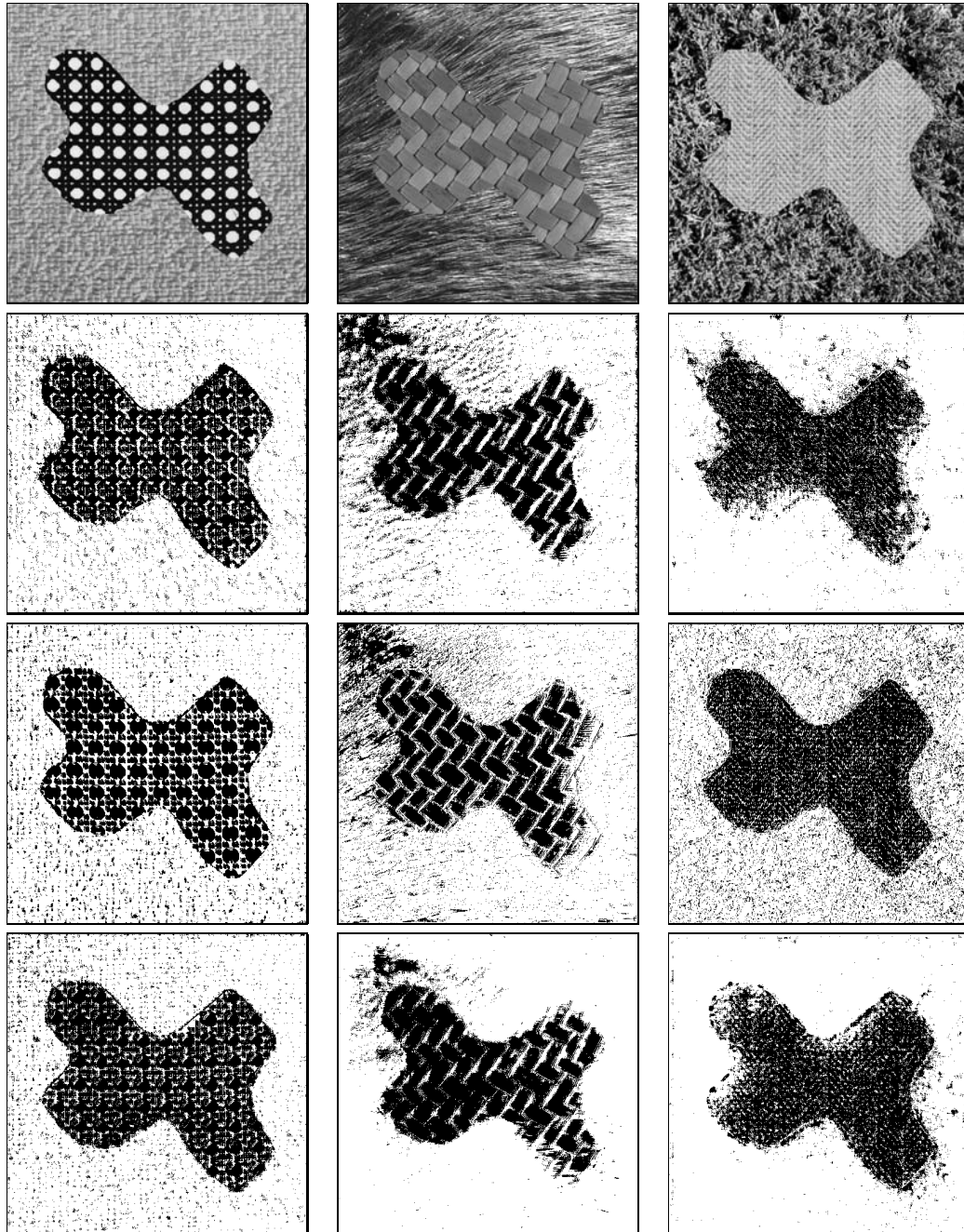


Figure 5.24: Original mosaics of Brodatz textures D102 and Raffia (left), Fabric0000 and Fabric0004 (centre) and Herring and Leaves0012 (right); Pixelwise classification results obtained using models trained down to level 6 with a penalty of $\beta = 64000$ and mother wavelet of type Daubechies4 (row2), Haar (row3) and Coiflet1 (row4).

Chapter 6

Conclusion and Future Work

This chapter provides a brief overview of the work carried out along with a discussion on the contributions and shortcomings of the models developed. We highlight two new avenues of research which should extend the scope of the models and, in doing so, improve the accuracy of the texture classification scheme.

6.1 Summary of the Work

In this thesis, we have addressed the problem of texture description. Texture is an important attribute of the appearance of many semantic entities of interest in a wide variety of images. The ability to describe it therefore contributes significantly to scene understanding. Features computed from textural information in an image form an integral part of many segmentation and classification methods. We have chosen to adopt a probabilistic approach to the problem as it provides us with a coherent framework within which to work.

Although texture appears in finite regions in the images we process, it does not explicitly depend on the size or shape of these regions. Based on this observation, we have chosen to describe texture via a distribution on the space of infinite images which captures the variations present in the texture that give rise to its defining structure. A Gaussian form was chosen for this distribution which, along with the requirement of translation invariance, produced a distribution for texture that is diagonal in the Fourier basis.

Starting with this measure on the space of infinite images, we derived a distribution for texture on a finite region of interest via marginalisation. The computational complexity associated with calculating the resulting probability of the texture on the region required that we diagonalise the inverse covariance of the distribution on the region. This was made possible by placing a restriction on the form of the inverse covariance, namely that it must be piecewise constant on a dyadic partition of the frequency domain.

Due to the local compact nature of the frequency support of wavelet packet basis functions, the elements of such a dyadic partition correspond to subbands in a wavelet

packet decomposition of the textured image. If the wavelet packet basis functions also form a basis for the region of interest, then they satisfy the conditions needed in order to diagonalise the inverse covariance of the distribution on the region. They therefore provide us with the means to calculate the probability of the texture on the region.

Learning the parameters of a wavelet packet texture model involves training it on sample patches of the texture. These patches were chosen to be dyadic in shape so that the wavelet packet basis functions would form a basis on them. The underlying structure of the texture was then captured by decomposing the patches where necessary as determined by the probability measure. The extent to which the patches are decomposed is controlled by a penalty that promotes a relatively smooth partition of the frequency space, with fine divisions permitted only in areas which have a lot of activity. This results in an adaptive wavelet packet decomposition of the texture.

The trained texture models were applied to the supervised segmentation of several types of images: natural images from the Brodatz album, textured scenes cropped from high resolution remote sensing images, and images of wildlife in their natural habitat. The segmentation results presented in chapter 5 demonstrate the suitability of the models for the task of texture analysis and clearly support the use of a wavelet packet based probabilistic framework for the classification of a variety of different textured images.

6.2 Understanding What the Adaptive Models Achieve

The texture models developed in this thesis are based on a partitioning of the Fourier domain that corresponds to a wavelet packet decomposition of the textured image. They could equally well be expressed in terms of a standard wavelet basis. Considering what this means throws light on what the models achieve. Although the inverse covariance of a model is (approximately) diagonal in the adapted wavelet packet basis chosen by the training procedure, it will not in general be diagonal in the standard wavelet basis. Mathematically, this is clear. Intuitively, the reason is the following. Many textures possess periodic correlations between pixels at intermediate frequencies that extend over many wavelengths [24]. The standard wavelet basis, however, links the notions of scale and frequency, so that the range of correlations is of the order of the inverse of the frequency of the wavelet (i.e. one wavelength). Correlations between pixels over a range corresponding to many wavelengths can thus only be captured in a standard wavelet basis by introducing interactions between coefficients. This is typically achieved through the use of tree models, as discussed in chapter 2, in which coefficients interact via their parents, grand-parents, and so on.

In contrast, the adaptive bases presented here capture these long-range correlations using wavelet packet bases that can represent correlations between pixels over many wavelengths while remaining independent, with no need for interactions between the coefficients themselves. For Gaussian models, this type of description is always possible,

since the inverse covariance can always be diagonalised, at least in principle. The problem is that the basis elements may not have compact spatial support, which renders them unsuitable for segmentation. The procedure presented in this thesis can thus be thought of as finding the basis that diagonalises an inverse covariance that must simultaneously be learnt, subject to probabilistic restrictions on its form that ensure the more or less compact spatial support of the basis elements.

The adaptive nature of the bases overcomes the computational burden associated with fixed depth full wavelet packet decompositions. This is because the adaptive decompositions do not unnecessarily divide up the entire frequency domain, but instead only partition those areas which contain the characteristic frequencies present in the texture.

They also provide a better description of the texture. If certain wavelets are independent then this is clearly best expressed through the use of one large region of constant f , rather than four or more regions each with similar, but not quite equal f .

6.3 Future Work

Although the models have proved successful for the purposes of texture description, there are several issues which remain unresolved. These are addressed below.

6.3.1 Selecting the Mother Wavelet

A comparison of the Fourier transform of a texture to the adaptive decompositions produced using different mother wavelets, such as that carried out in section 5.5, suggests that certain wavelets generate “better” decompositions than others. By this we mean that their resulting decompositions match more closely to the structure present in the Fourier transform.

These different decompositions were tested in section 5.5 to see if they produced different classifications of the same mosaics. The results shown in figure 5.24 suggest that the filter length of the mother wavelet plays a significant role in the analysis and classification of macrot textures. This observation, along with the difference in the adaptive decompositions produced by the different wavelets, as shown in figure 5.23, indicates the need for further research on the topic of mother wavelet selection.

This research should not only focus on selecting between orthogonal wavelets such as those used in this thesis. It should also examine the performance of wavelets which have attractive properties such as increased directional selectivity and translation invariance. Complex wavelet packets could be used for this purpose as they improve on regular wavelets in both these senses. The possibility of designing new wavelets which match the structure present in a given texture is also an option.

6.3.2 The Need for a More Complex Distribution

Although the assumption of a Gaussian distribution yields models which adapt quite well to the structure of the texture being analysed, the simplicity of the distribution has several drawbacks.

As discussed throughout this thesis, the requirement of translation invariance is absolutely essential in any texture classification scheme. With our choice of a Gaussian distribution, this requirement is easily satisfied as shown in section 3.2.2. A drawback of the resulting translation-invariant Gaussian models, however, is that they have no way of taking into account the relative phase of Fourier components, and yet phase is critical for texture structure. For phase information to be included in the distribution, a more complicated model is required.

Another important requirement for texture models is that of rotation invariance, however the models presented in this thesis do not have that property. One can impose rotation invariance on a Gaussian distribution, but the results are not that interesting since the inverse covariance must now be a function only of the magnitude of the frequency. Non-trivial rotation invariance can be achieved by mixtures over rotations or by otherwise increasing the complexity of the model.

The analysis of the adaptive wavelet packet subband statistics carried out in section 4.5 unveiled the presence of double peaked histograms. This discovery contradicts the notion that all wavelet packet subbands should be modelled by unimodal distributions. This points toward an obvious extension to our modelling assumption of a Gaussian distribution. The subbands which exhibit this bimodal behaviour cannot be modelled by a Gaussian or generalised Gaussian distribution. Instead a more complicated model is needed.

All three of these modelling issues can be treated in future work by extending the Gaussian model through the addition of quartic terms to the exponent. The quartic model proposed in section 4.5 presents one possible way to address this extension.

Changes in both the form of the distribution and in the mother wavelet, can be readily incorporated into the probabilistic framework which we have developed in this thesis. These changes should result in a more adaptive modelling procedure which chooses the most appropriate analysis tool and distribution for the texture being described.

Appendix A

Wavelets

This appendix provides a brief overview of wavelets. For a more detailed explanation of the subject, the reader is referred to [2, 49, 60, 79].

A.1 The Wavelet Transform

Let $f \in L^2(\mathbb{R})$ be a function in the space of square integrable functions. Expanding f in terms of a wavelet basis allows one to analyse the function locally in both space and frequency. The family of wavelets which make up the fundamental elements of this expansion are generated through a series of translations and dilations of a single function $\psi \in L^2(\mathbb{R})$, known as the *mother wavelet*. The mother wavelet is a function with zero mean

$$\int_{-\infty}^{+\infty} \psi(x) dx = 0 \quad (\text{A.1})$$

which is normalised, $\|\psi\| = 1$.

For the continuous case, the expansion elements are determined via the following expression:

$$\psi_{s,t}(x) = \frac{1}{\sqrt{|s|}} \psi\left(\frac{x-t}{s}\right) \quad (\text{A.2})$$

where $s, t \in \mathbb{R}$ and $s \neq 0$. These elements are also normalised, *i.e.* $\|\psi_{s,t}\| = 1$. The parameters s and t are known as the *scaling* and *translation* parameters, respectively. Figure A.1 shows examples of such expansion elements.

The *continuous wavelet transform* of the function f is then given by

$$\begin{aligned} Cf(s, t) &= \langle f, \psi_{s,t} \rangle \\ &= \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(x) \psi^*\left(\frac{x-t}{s}\right) dx \end{aligned}$$

where ψ^* denotes the complex conjugate of ψ . One may think of the *wavelet coefficient*, $Cf(s, t)$, as a measure of similarity or correlation between the function f and the analysing wavelet at position t and scale s , with a large coefficient representing a high degree of similarity.

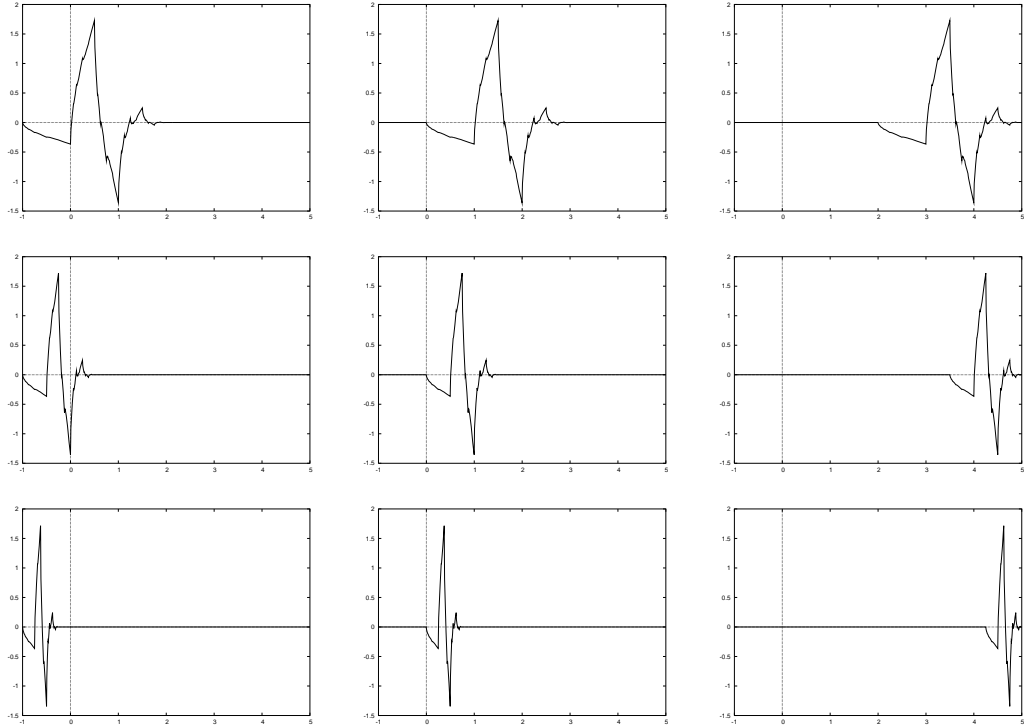


Figure A.1: Scaled and translated versions of the Daubechies2 [22] mother wavelet.

As demonstrated by Mallat [60], each wavelet $\psi_{s,t}$ has a local support in both the spatial and frequency domain. This can be seen in figure A.2 which shows the Heisenberg boxes corresponding to two wavelets from the same family but which are at different scales and positions. It is this space-frequency localisation property of wavelets which has made them so popular in signal and image processing.

Throughout this thesis we have used the discrete form of the wavelet transform in our calculations. The main difference between this type of analysis and that described above is that the scaling and translation parameters are no longer continuous. Instead they are integers and, furthermore, they are restricted to a set of dyadic values:

$$s = 2^j \quad (\text{A.3})$$

$$t = k 2^j = k s \quad (\text{A.4})$$

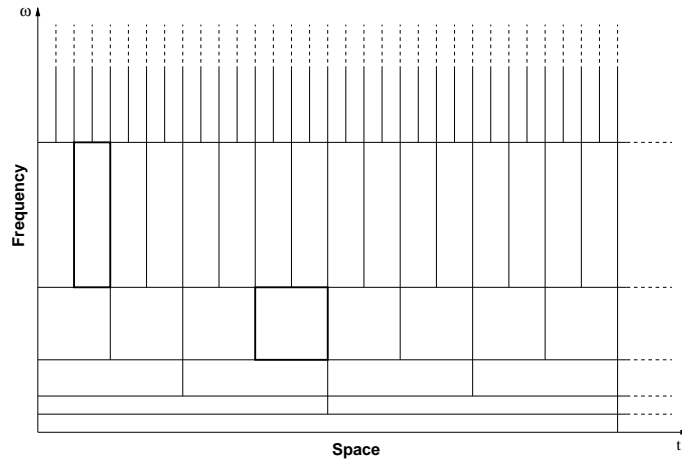


Figure A.2: Space-frequency localisation of wavelet basis functions.

where $(j, k) \in \mathbb{Z}^2$. Hence for the discrete case equation (A.2) becomes

$$\psi_{j,k}[x] = \frac{1}{\sqrt{2^j}} \psi(2^{-j}x - k) . \tag{A.5}$$

A.2 A Multiresolution Analysis Framework

Performing a wavelet analysis of a function $f \in L^2(\mathbb{R})$ is best explained within the framework of *multiresolution analysis*. A multiresolution approximation of the space $L^2(\mathbb{R})$ is formally defined to be a sequence of closed subspaces $\{V_j\}_{j \in \mathbb{Z}}$ of $L^2(\mathbb{R})$ which has the following properties:

- The subspaces are nested:

$$\cdots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \cdots$$

- The dilation property:

$$v(x) \in V_j \Leftrightarrow v(2x) \in V_{j-1} \quad \forall j \in \mathbb{Z} .$$

- The translation property:

$$v(x) \in V_0 \Leftrightarrow v(x - 2^j k) \in V_0 \quad \forall k \in \mathbb{Z} .$$

- The subspaces provide an approximation of the space $L^2(\mathbb{R})$:

$$\bigcup_{j \in \mathbb{Z}} V_j \text{ is dense in } L^2(\mathbb{R}) \text{ and } \bigcap_{j \in \mathbb{Z}} V_j = \{0\} \quad .$$

- There exists a *scaling function* $\phi \in V_0$ whose set of translates, $\{\phi(x-k)\}_{k \in \mathbb{Z}}$, form an orthonormal basis for the subspace V_0 .

From the above properties, we can see that any function in the subspace V_1 can be expressed as a linear combination of the basis functions of the subspace V_0 . This is captured in the following expression:

$$\frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right) = \sum_{k=-\infty}^{+\infty} h[k] \phi(x-k) \quad (\text{A.6})$$

which is known as the *dilation equation*. This equation describes the relationship between the dilation of the function ϕ by 2 and the complete set of its translates. Here $h[k]$ is a discrete filter, known as the *conjugate mirror filter*, whose coefficients are given by

$$h[k] = \left\langle \frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right), \phi(x-k) \right\rangle. \quad (\text{A.7})$$

We can also state the more general result that for each scale $j \in \mathbb{Z}$, the set of translations

$$\left\{ \phi_{j,k}(x) = \frac{1}{\sqrt{2^j}}\phi(2^{-j}x - k) \right\}_{k \in \mathbb{Z}} \quad (\text{A.8})$$

forms an orthonormal basis for the subspace V_j .

An operator P_j is defined such that $P_j f$ is the projection of the function $f \in L^2(\mathbb{R})$ onto the subspace V_j . This projection represents an approximation of f at scale 2^j . Due to the nature and nesting order of the subspaces, information about f is lost when we move between the approximations $P_{j-1}f$ and $P_j f$. This missing information or *detail* about the function f which is present at scale 2^{j-1} , but which is missing at scale 2^j , is contained in W_j which is known as the *detail space at scale* 2^j . W_j is the orthogonal complement of V_j in V_{j-1} :

$$V_{j-1} = V_j \oplus W_j \quad (\text{A.9})$$

where \oplus denotes the direct sum. From equation (A.9), we can see that any function in the subspace V_{j-1} can be expressed as the sum of a function in the detail space W_j and a function in the approximation space V_j .

It can be shown [60] that there exists a function ψ whose set of translations forms an orthonormal basis for W_0 . Using the same reasoning as in the case of the scaling

function, an element of the detail space W_1 can be written as a linear combination of the elements in the subspace V_0 . Hence, there must exist another discrete filter g such that ψ is given by

$$\frac{1}{\sqrt{2}}\psi\left(\frac{x}{2}\right) = \sum_{k=-\infty}^{+\infty} g[k] \phi(x-k) \quad (\text{A.10})$$

where

$$g[k] = \left\langle \frac{1}{\sqrt{2}}\psi\left(\frac{x}{2}\right), \phi(x-k) \right\rangle. \quad (\text{A.11})$$

The set of all dilations and translations $\{\psi_{j,k}\}_{(j,k) \in \mathbb{Z}^2}$ forms an orthonormal basis for $L^2(\mathbb{R})$. This function ψ is the mother wavelet we introduced at the beginning of the appendix.

When a function f is expanded via a multiresolution analysis, there must be a *level* or scale at which the analysis stops. This reference level, which we will denote by J , may be arrived at naturally by the length of the signal being analysed or it may be selected by the user. Whichever way it is chosen, once J has been selected we may express the function f as the sum of all the details up to scale J plus its approximation at scale J :

$$f(x) = A_J + \sum_{j \leq J} D_j. \quad (\text{A.12})$$

The conjugate mirror filters h and g represent a low-pass and a high-pass filter, respectively. In practice, the discrete wavelet transform is carried out by convolving these two filters with the original signal in order to generate the approximation to the signal at a given level and the detail representation of the signal at that level.

Appendix B

Completing the Square in $\phi_{\bar{R}}$

The expression for the energy in equation (3.16) can be viewed as containing the following three terms in $\phi_{\bar{R}}$:

$$\begin{aligned}
 \text{Quadratic Term} & : \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle \\
 \text{Linear Term} & : \langle \phi_R | F_{R\bar{R}} | \phi_{\bar{R}} \rangle + \langle \phi_{\bar{R}} | F_{\bar{R}R} | \phi_R \rangle \\
 \text{Constant Term} & : \langle \phi_R | F_{RR} | \phi_R \rangle
 \end{aligned}$$

To complete the square in $\phi_{\bar{R}}$ we take the following steps:

- Preserve the quadratic term $\langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle$
- Introduce a linear term $\langle \phi_{\bar{R}} + F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + F_{\bar{R}R}\phi_R \rangle$
- Cancel the extra $F_{\bar{R}\bar{R}}$ term $\langle \phi_{\bar{R}} + (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R \rangle$

Expanding this last expression yields

$$\begin{aligned}
 E(\phi) & = \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle + \langle (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle + \\
 & \quad \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R \rangle + \langle (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}} | (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R \rangle \\
 & = \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle + \langle (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}}\phi_{\bar{R}} \rangle + \\
 & \quad \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}}(F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R \rangle + \langle (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}}(F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R \rangle \\
 & = \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} \rangle + \langle (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}}\phi_{\bar{R}} \rangle + \\
 & \quad \langle \phi_{\bar{R}} | F_{\bar{R}\bar{R}} | \phi_R \rangle + \langle (F_{\bar{R}\bar{R}})^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}}\phi_R \rangle. \tag{B.1}
 \end{aligned}$$

Examining the second term of equation (B.1) in detail gives

$$\begin{aligned}
\langle F_{\bar{R}\bar{R}}^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}\bar{R}}\phi_{\bar{R}} \rangle &= \langle \pi_{\bar{R}}Fi_{\bar{R}}^{-1}\pi_{\bar{R}}Fi_R\phi_R | \pi_{\bar{R}}Fi_{\bar{R}}\phi_{\bar{R}} \rangle \\
&= \langle \pi_{\bar{R}}Fi_R\phi_R | i_{\bar{R}}^\dagger F\pi_{\bar{R}}^{\dagger-1} \pi_{\bar{R}}Fi_{\bar{R}}\phi_{\bar{R}} \rangle \\
&= \langle \pi_{\bar{R}}Fi_R\phi_R | \pi_{\bar{R}}Fi_{\bar{R}}^{-1}\pi_{\bar{R}}Fi_{\bar{R}}\phi_{\bar{R}} \rangle \\
&= \langle \pi_{\bar{R}}Fi_R\phi_R | \phi_{\bar{R}} \rangle \\
&= \langle Fi_R\phi_R | \pi_{\bar{R}}^\dagger | \phi_{\bar{R}} \rangle \\
&= \langle i_R\phi_R | F\pi_{\bar{R}}^\dagger | \phi_{\bar{R}} \rangle \\
&= \langle \phi_R | i_{\bar{R}}^\dagger F\pi_{\bar{R}}^\dagger | \phi_{\bar{R}} \rangle \\
&= \langle \phi_R | \pi_{\bar{R}}Fi_{\bar{R}} | \phi_{\bar{R}} \rangle \\
&= \langle \phi_R | F_{R\bar{R}} | \phi_{\bar{R}} \rangle
\end{aligned} \tag{B.2}$$

which is exactly the second term in equation (3.16).

Therefore, we have found that the extra piece of the energy puzzle is the final term of equation (B.1) which when expanded becomes

$$\begin{aligned}
\langle F_{\bar{R}\bar{R}}^{-1}F_{\bar{R}R}\phi_R | F_{\bar{R}R}\phi_R \rangle &= \langle \pi_{\bar{R}}Fi_{\bar{R}}^{-1}\pi_{\bar{R}}Fi_R\phi_R | \pi_{\bar{R}}Fi_R\phi_R \rangle \\
&= \langle \pi_{\bar{R}}Fi_R\phi_R | i_{\bar{R}}^\dagger F\pi_{\bar{R}}^{\dagger-1} \pi_{\bar{R}}Fi_R\phi_R \rangle \\
&= \langle \pi_{\bar{R}}Fi_R\phi_R | \pi_{\bar{R}}Fi_{\bar{R}} \rangle^{-1} \pi_{\bar{R}}Fi_R\phi_R \rangle \\
&= \langle \phi_R | i_{\bar{R}}^\dagger F\pi_{\bar{R}}^\dagger \pi_{\bar{R}}Fi_{\bar{R}}^{-1}\pi_{\bar{R}}Fi_R | \phi_R \rangle \\
&= \langle \phi_R | \pi_{\bar{R}}Fi_{\bar{R}}\pi_{\bar{R}}Fi_{\bar{R}}^{-1}\pi_{\bar{R}}Fi_R | \phi_R \rangle \\
&= \langle \phi_R | F_{R\bar{R}}F_{\bar{R}\bar{R}}^{-1}F_{\bar{R}R} | \phi_R \rangle.
\end{aligned} \tag{B.3}$$

We can therefore complete the square in the following manner

$$\begin{aligned}
 E(\phi) &= \langle \phi_{\bar{R}} + F_{\bar{R}\bar{R}}^{-1} F_{\bar{R}R} \phi_R | F_{\bar{R}\bar{R}} | \phi_{\bar{R}} + (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} \phi_R \rangle \\
 &\quad - \langle \phi_R | F_{R\bar{R}} (F_{\bar{R}\bar{R}})^{-1} F_{\bar{R}R} | \phi_R \rangle + \langle \phi_R | F_{RR} | \phi_R \rangle.
 \end{aligned} \tag{B.4}$$

Appendix C

Implementation

The implementation part of our work was carried out using the Java programming language. The software we wrote consists of two main parts: a scheme to train adaptive texture models and a scheme to use these models to segment textured scenes. Both of these schemes share the same software design and large portions of code. For this reason it is sufficient to provide the details for the training part of the implementation only.

C.1 The Java Programming Language

In the field of image processing, Java is not an obvious programming language choice and is frequently overlooked in favour of C or C++. This may be related to the fact that Java has a reputation for being a slow language that is not well suited for computationally intensive tasks.

In order to investigate this performance issue, we performed an experiment to determine how much slower Java was compared to C/C++. We wrote a function which multiplies two 128×128 matrices containing double precision floating point numbers, which is highly representative of the computations carried out by our software. We implemented that function in both Java and C and plotted the execution time against the number of iterations of this operation. The graph can be seen in figure C.1. This produced the surprising result that, after a given number of iterations, Java performs just as well as C, the only difference between the two languages being that the Java program takes longer to start. This is due to the fact that a Java program is not directly executed by the processor of the machine, as is the case for a program written in C, but rather it is interpreted by a *virtual machine*. It is this virtual machine which introduces a significant start-up delay of about $100ms$. However, as our application has running times measured in minutes if not hours, this problem does not affect us. Hence we have shown that Java performs just as well as C for long-running applications of the type we want to develop.

As the performance issue has been resolved for our particular application, it is

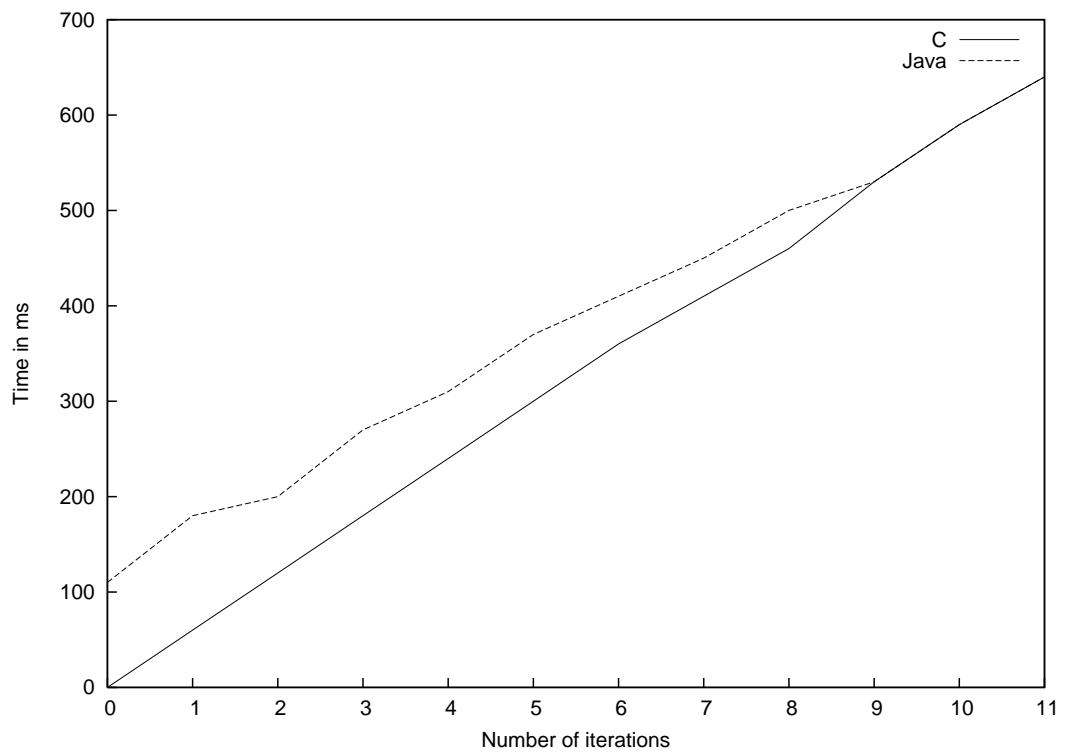


Figure C.1: Comparing Java and C: Time taken to perform a number of iterations of the multiplication of two 128×128 matrices containing double precision floating point numbers.

perhaps worth mentioning some of the advantages of the Java programming language. One of its key features is that a Java program can run unmodified on any computer that has a Java virtual machine installed. This is because, as mentioned above, a Java program is interpreted by a Java virtual machine, which shields the program from any specificities of the underlying hardware or operating system. As a consequence, we were able to run our training and segmentation tasks indifferently on PCs running Linux or Windows, or even on Sun Sparc Stations running the Solaris operating system. This feature proved to be very useful as it allowed us to take full advantage of any available processing power in our work environment.

C.2 The Use of Class Inheritance

As Java is a fully fledged object-oriented language, we were able to make use of powerful techniques, such as inheritance, in our programming. Two examples taken from our code are decomposition trees and multiple wavelets.

The wavelet packet decompositions produced by our training algorithm are represented by trees. The root of the tree is the original image to be decomposed, and the nodes represent the successive transforms applied to the image in order to obtain a set of subbands, which are located at the leaves of the tree. As Java is an object-oriented language, each node in the decomposition tree is an object, *i.e.* an instance of a class. In order to represent all three splits which are considered at each node in the tree, we created eight classes corresponding to the eight transforms which are carried out on each node: low pass on rows, high pass on rows (the combination of these two make up the first split type); low pass on columns, high pass on columns (the combination of these two make up the second split type); low pass on rows and low pass on columns, low pass on rows and high pass on columns, high pass on rows and low pass on columns, high pass on rows and high pass on columns (the combination of these four make up the first split type). In order to be able to perform operations on nodes that do not depend on the type of the node (such as the calculation of the model parameters), we created an abstract class that represents nodes in general. The eight classes are all subclasses of this general node class.

We also used class inheritance for making our software flexible with respect to which mother wavelet we want to use to decompose the texture. Here, as in the above example, we set up an abstract class that represents a general wavelet transform, and then created as many subclasses as mother wavelets used in our analysis.

C.3 Multithreaded Programming

Another very useful feature of Java that we used is *multi-threading*. Java provides us with the ability to start multiple *threads* within a single application. Threads (also known as *lightweight processes*) are parallel executions of different pieces of code that

belong to the same application, much in the same way processes are parallel executions of different applications that run on the same machine. Threads allow us to run independent computations in parallel, while retaining the possibility to specify when and where those computations must wait for each other before moving on.

Threads proved very useful for designing the part of our software that deals with training a texture model from multiple samples extracted from different locations of the original texture patch. Our starting point was the code we had for training a texture model on only one patch. What we did was to create one thread for each patch, and run those training tasks in parallel. However, it is not desirable to have these tasks run fully independently from each other. Depending on the patch considered, the algorithm could make different choices, resulting in different tree structures, which would be impossible to average out into a single model for the texture. This is why we have to make sure that all the trees produced by the different training tasks running in parallel share the same structure, even if the value of the parameters contained in each node of the trees are different.

In order to achieve this result, we choose to synchronise our threads so that, at each node, all the threads wait for each other, compare their results, and decide collectively on which split, if any, should be retained. As described above in section 4.3.1, this decision is made by comparing the penalised probability for the parent node with the penalised probabilities of the child nodes which make up the three possible splits. These probabilities are calculated according to equation 4.3. The contribution from each texture patch is controlled by a separate thread. Once all the threads have arrived at the same point in the algorithm, they combine their information to compute these probabilities.

Using threads for our multiple patch training scheme was a good idea since it allowed us to reuse the original code we had written for training a model on a single patch. In addition, there is an added benefit of using threads. If the machine used has more than one processor, the Java virtual machine will automatically dispatch threads evenly between the processors available, a task that would require considerable time and expertise in other languages. When run on a two-processor PC, we noted that our application took about half the time it took on a similar one-processor machine.

Appendix D

Publications of the Author

R. Cossu, I. Jermyn, K. Brady, and J. Zerubia. Bayesian Model Selection in the Analysis of the Uni- and Multi-Modal Statistics of Adaptive Wavelet Packet Coefficients. *INRIA Research Report*. To appear December 2003.

K. Brady, I. Jermyn, and J. Zerubia. Texture Analysis: An Adaptive Probabilistic Framework. *Proc. IEEE International Conference on Image Processing*. Barcelona, Spain. September 2003.

K. Brady, I. Jermyn, and J. Zerubia. Adaptive Probabilistic Models of Wavelet Packets for the Analysis and Segmentation of Textured Remote Sensing Images. *Proc. British Machine Vision Conference*. Norwich, U.K. September 2003.

K. Brady, I. Jermyn, and J. Zerubia. A Probabilistic Framework for Adaptive Texture Description. *INRIA Research Report No. 4920*.

Bibliography

- [1] A. Arnulf and O. Dupuy. La transmission des contrastes par le système optique de l'oeil et les seuils de contrastes rétinien. *C. R. Seances Acad. Sci. (Paris)*, 250:2257–2759, 1960.
- [2] M. Barlaud. *Wavelets in Image Communication*. Elsevier Amsterdam, 1994.
- [3] J. Beck, A. Sutter, and R. Ivry. Spatial frequency channels and perceptual grouping in texture segregation. *Computer Vision, Graphics, and Image Processing*, 37:299–325, 1987.
- [4] J. Besag. Spatial interaction and the statistical analysis of lattice. *Journal of the Royal Statistical Society, Series B*, 36:192–236, 1974.
- [5] A. Bhalerao and N. Rajpoot. Discriminant feature selection for texture classification. In *British Machine Vision Conference*, Norwich, England, September 2003.
- [6] A. C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.
- [7] F. W. Campbell and J.J. Kulikowski. Orientation selectivity of the human visual system. *Journal of Physiology*, 187:437–445, 1966.
- [8] F.W. Campbell and D.G. Green. Optical and retinal factors affecting visual resolution. *Journal of Physiology*, 181:437–445, 1965.
- [9] F.W. Campbell and R.W. Gubisch. Optical quality of the human eye. *Journal of Physiology*, 186:558–578, 1966.
- [10] F.W. Campbell and J.G. Robson. Application of Fourier analysis to the visibility of gratings. *Journal of Physiology*, 197:551–566, 1968.
- [11] T. Chang and C.C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Trans. on Image Processing*, 2(4):429–441, 1993.

- [12] R. Chellappa and S. Chatterjee. Classification of textures using Gaussian Markov random fields. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 33(4):959–963, 1985.
- [13] H. Choi and R. Baraniuk. Multiscale image segmentation using wavelet-domain hidden markov models. *IEEE Trans. on Image Processing*, 10(9):1309–1321, 2001.
- [14] J.M. Coggins. *A Framework for Texture Analysis Based on Spatial Filtering*. PhD thesis, Michigan State University, 1982.
- [15] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
- [16] R. R. Coifman and M. V. Wickerhauser. Entropy-based methods for best basis selection. *IEEE Trans. on Information Theory*, 38(2):713–718, 1992.
- [17] R.G. Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, 37:35–46, 1991.
- [18] R.G. Congalton and K. Gresn. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. Lewis Publishers, Washington D.C., 1998.
- [19] R.W. Conners, C.W. McMillin, K. Lin, and R.E. Vasquez-Espinosa. Identifying and locating surface defects in wood: Part of an automated lumber processing system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5:573–583, 1983.
- [20] G. C. Cross and A. K. Jain. Markov random field texture models. *Trans. on Pattern Analysis and Machine Intelligence*, 5:25–39, 1983.
- [21] M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Trans. on Signal Processing*, 1998.
- [22] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.
- [23] J. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856, 1980.
- [24] G. Van der Wouwer. *Wavelets for Texture Analysis*. PhD thesis, University of Antwerp, May 1998.
- [25] R.L. Devalois, D.G. Albrecht, and L.G. Thorell. Spatial-frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559, 1982.
- [26] P. Dewaele, P. Van Gool, and A. Oosterlinck. Texture inspection with self-adaptive convolution filters. In *International Conference on Pattern Recognition*, pages 56–60, Rome, Italy, November 14-17 1988.

- [27] M.N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Trans. on Image Processing*, 11(2):146–158, February 2002.
- [28] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. J. Wiley and Sons, 1973.
- [29] K. Etemad and R. Chellappa. Separability based tree structured local basis selection for texture classification. In *IEEE International Conference on Image Processing*, volume 1, pages 441–444, 1994.
- [30] N. Fatemi-Ghomi, P.L. Palmer, and M. Petrou. Performance evaluation of texture segmentation algorithms based on wavelets. In *Workshop on Performance Characteristics of Vision Algorithms, European Conference on Computer Vision*, Cambridge, England, April 1996.
- [31] N. D. Fletcher and A. N. Evans. Minimum distance texture classification of SAR images using wavelet packets. In *IEEE International Geoscience and Remote Sensing Symposium*, Toronto, Canada, 2002.
- [32] J. M. Francos, A. Z. Meiri, and B. Porat. A unified texture model based on a 2-D Wold like decomposition. *IEEE Trans. on Signal Processing*, 41:2665–2678, 1993.
- [33] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *International Congress of Mathematics*, pages 1496–1517, 1986.
- [34] M.A. Georgeson. Spatial Fourier analysis and human vision. In N.S. Sutherland, editor, *Tutorial Essays, A Guide to Recent Advances*, volume 2, chapter 2. Lawrence Erlbaum Associates, Hillsdale, NJ, 1979.
- [35] G. Gimel'farb. Supervised texture segmentation by maximising conditional likelihood. In *Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 169–184, Sophia Antipolis, France, September 2001. Springer. LNCS 2134.
- [36] P. Goel and B. Vidakovic. Wavelet transforms as diversity enhancers, 1995. Institute of Statistics and Decision Sciences, Duke University, NC. Discussion Paper 95–04.
- [37] L. Van Gool, P. Dewaele, and A. Oosterlinck. Texture analysis: Anno 1983. *Computer Vision, Graphics and Image Processing*, 29:336–357, 1985.
- [38] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, 1980.

- [39] A. Grossman and J. Morlet. Decompositions of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15:723–736, 1984.
- [40] R. Gupta and P. Undrill. The use of texture analysis to delineate suspicious masses in mammography. *Physics in Medicine and Biology*, 40:835–855, 1995.
- [41] R.M. Haralick. A texture-context feature extraction algorithm for remotely sensed imagery. In *In Proceedings of the 1971 IEEE Decision and Control Conference*, pages 650–657, December 1971.
- [42] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979.
- [43] R.M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Trans. on Systems, Man and Cybernetics*, 3(6):610–621, 1973.
- [44] M. Hassner and J. Sklansky. The use of Markov random fields as models of texture. *Computer Graphics and Image Processing*, 12:357–370, 1980.
- [45] R. Hu and M.M. Fahmy. Texture segmentation based on a hierarchical Markov random field model. *Signal Processing*, 26(3):285–305, 1992.
- [46] D. Hubel and T. Wiesel. *Receptive Fields*, binocular interaction and functional architecture in the cat’s visual cortex. *Journal of Physiology of London*, 160:106–154, 1962.
- [47] A. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [48] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.
- [49] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analysis. *SIAM Review*, 36(3):377–412, 1994.
- [50] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, March 1981.
- [51] B. Julesz. A theory of preattentive texture discrimination based on first-order statistics of textons. *Biological Cybernetics*, 41(2):131–138, 1981.
- [52] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1186–1190, 1993.
- [53] A. Laine and J. Fan. Frame representations for texture segmentation. *IEEE Trans. on Image Processing*, 5(5):771–780, May 1996.

- [54] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.
- [55] F. Liu. *Modeling Spatial and Temporal Textures*. PhD thesis, MIT, 1997.
- [56] S. Livens. *Image Analysis for Material Characterisation*. PhD thesis, University of Antwerp, 1998.
- [57] S. Livens, P. Scheunders, G. Van de Wouwer, and D. Van Dyck. Wavelets for texture analysis, an overview. In *Proc. ICIP and its Applications*, pages 581–585, Dublin, Ireland, 1997.
- [58] S. Mallat. Multifrequency channel decomposition of images and wavelet models. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37:2091–2110, 1989.
- [59] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 11(7):674–693, 1989.
- [60] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [61] B. B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman, San Francisco, 1983.
- [62] A. Mojsilovic, M. V. Popovic, and D. M. Rackov. On the selection of an optimal wavelet basis for texture characterization. *IEEE Trans. on Image Processing*, 9(12):2043–2050, December 2000.
- [63] Izumi Ohzawa. http://www.bpe.es.osaka-u.ac.jp/ohzawa-lab/ohzawa_e.htm.
- [64] C. Ramananjara, O. Alata, and M. Najim. 2-d wold decomposition: new parameter estimation approach to evanescent field spectral supports. In *EUSIPCO*, Tampere, Finland, 2000.
- [65] T. Randen and J. H. Husoy. Filtering for texture classification: A comparative study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.
- [66] T. Randen and J.H. Husoy. Multichannel filtering for image texture segmentation. *Optical Engineering*, 33:2617–2652, August 1994.
- [67] A. R. Rao. *A Taxonomy for Texture Description and Identification*. Springer-Verlag, New York, 1990.
- [68] T.R. Reed and J.M. Hans du Buf. A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics and Image Processing: Image Understanding*, 57(3):359–372, 1993.

- [69] T. J. Rubio, A. Bandera, C. Urdiales, and F. Sandoval. A hierarchical context-based textured image segmentation algorithm for aerial images". In *Texture02*, pages 113–122, Copenhagen, Denmark, 2002.
- [70] N. Saito, R.R. Coifman, F.B. Geshwind, and F. Warner. Discriminant feature extraction using empirical probability density estimation and a local basis library. *Pattern Recognition*, 35:2841–2852, 2002.
- [71] O.H. Schade. Optical and photoelectric analog of the eye. *Journal of the Optical Society of America*, 46:721–739, 1956.
- [72] P. Scheunders, S. Livens, G. Van de Wouwer, P. Vautrot, and D Van Dyck. Wavelet-based texture analysis. *International Journal on Computer Science and Information Management*, 1(2):22–34, 1998.
- [73] C. W. Shaffrey, N. G. Kingsbury, and I. H. Jermyn. Unsupervised image segmentation via Markov trees and complex wavelets. In *IEEE International Conference on Image Processing*, Rochester NY, September 23–25 2002.
- [74] M. Singh and S. Singh. Spatial texture analysis: A comparative study. In *15th International Conference on Pattern Recognition*, Quebec, August 2002.
- [75] J. Strand and T. Taxt. Local frequency features for texture classification. *Pattern Recognition*, 27(10):1397–1406, 1994.
- [76] N.V. Swindale. Visual cortex: A cat's-eye view of the visual system. *Current Biology*, 7(6):387–389, June 1997.
- [77] M. Tuceryan and A. K. Jain. *Texture Analysis*, chapter 2.1. World Scientific, 1998.
- [78] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. on Image Processing*, 4(11):1549–1560, November 1995.
- [79] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall, 1995.
- [80] T.P. Weldon, W.E. Higgs, and D.F. Dunn. Gabor filter design for multiple texture segmentation. *Optical Engineering*, 35(10):2852–2863, October 1996.
- [81] J.S. Weszka and A. Rosenfeld. An application of texture analysis to materials inspection. *Pattern Recognition*, 8:195–199, 1976.
- [82] M.V. Wickerhauser. Lectures on wavelet packet algorithms. Technical report, Department of Mathematics, Washington University, St. Louis, Missouri, November 1991.

- [83] S. Wilson and J. Zerubia. Unsupervised segmentation of textured satellite and aerial images with bayesian methods. In *European Signal Processing Conference on Image Processing*, Toulouse, France, September 2002.
- [84] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Springer, 2003.
- [85] S. C. Zhu, Y. N. Wu, and D. B. Mumford. Frame: Filters, random fields and maximum entropy - towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):1–20, March/April 1998.