Towards Efficient Texture Classification and Abnormality Detection

Amirhassan Monadjemi

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Computer Science.

October 2004

Abstract

One of the fundamental issues in image processing and machine vision is texture, specifically texture feature extraction, classification and abnormality detection. This thesis is concerned with the analysis and classification of natural and random textures, where the building elements and the structure of texture are not clearly determinable, hence statistical and signal processing approaches are more appropriate. We investigate the advantages of multi-scale/multi-directional signal processing methods, higher order statistics-based schemes, and computationally low cost texture analysis algorithms. Consequently these advantages are combined to form novel algorithms.

We develop a multi-scale/multi-directional Walsh-Hadamard transform for fast and robust texture feature extraction, where scale and angular decomposition properties are integrated into an ordinary Walsh-Hadamard transform, to increase its texture classification performance. We also introduce a highly accurate Gabor Composition method for texture abnormality detection which is a combination of a signal processing and a statistical method, namely Gabor filters and co-occurrence matrices. Furthermore, to overcome the practical drawbacks of traditional classification approaches, that require an extensive training stage, we introduce a method based on restructured eigenfilters for texture abnormality detection within a novelty detection framework. This demands only a minimal training stage using a few normal samples.

The proposed schemes are compared with commonly used texture classification methods on different image sets, including a high resolution outdoor scene database, samples of the VisTex colour texture suite, and randomly textured normal and abnormal tiles. The results are then analysed in order to evaluate texture classification performance, based upon accuracy, generality and computational costs.

Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree.

Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

SIGNED: DATE:

Acknowledgements

First of all, I would like to thank my supervisors, Professor Barry T. Thomas and Dr. Majid Mirmehdi for their unlimited support, clear guidance, novel ideas and brilliant discussions since I started my education here in Bristol. I do appreciate their efforts and concerns, that have improved my academic abilities over the last four years.

I am also grateful to the University of Bristol and the department of computer science, the head of department Professor David May and all departmental staff for their help. I am in particular grateful to Stuart for proof reading, the department's computer administrators Rob and Oliver for their support, and my friends Angus, Janko, Phil, James, Jason and Mike for their time and precious suggestions.

The Iranian Ministry of Science, Research and Technology (IMSRT) must also be acknowledged for the scholarship that I was awarded, and the Iranian students advisory in London for their assistance.

I owe special thanks to Bristol and Bristolians, such a lovely and lively city, with kind, polite and generous people. I have had an enjoyable time here, with Iranian, British, and many international friends.

While I was studying here, my grandfather, grandmother and father passed away one by one. I was their eldest grandson and only son. Sometimes it makes me so sad that I was not there while they were in a critical condition and wonder if my presence could have changed anything or not. However, I believe that they are more happy that I dealt with my further education. May God bless them.

Finally, I do not know how I can thank my family, my respectable mother to whom I owe many things, my small son, Rastin, to whom I owe a seashore trip and a lot of weekend activities, and my dearest wife, Atefeh, whose infinite support, incredible understanding and ultimate kindness are immeasurable. I am certain that I could not have finished this study if she were not as remarkable as she is. When we arrived here, we started a new life, alone, in a completely new environment, and could only rely on each other in our nuclear family. She is totally perfect, in particular a perfect partner.

List of Publications

- A. Monadjemi, M. Mirmehdi and B. T. Thomas. Restructured Eigenfilter Matching for Novelty Detection in Random Textures. In: *Proceedings of the 15th British Machine Vision Conference, BMVC 2004*, pages 637 – 646, September 2004.

- A. Monadjemi, M. Mirmehdi and B. T. Thomas. Fast texture classification in high resolution colour textures. In: *Proceedings of the 2nd Iranian Conference on Machine Vision, Image Processing and Applications*, pages 94 – 101, February 2003.

- A. Monadjemi, B. T. Thomas and M. Mirmehdi. Classification in High Resolution Images with Multiple Classifiers. In: *IASTED Visualization, Imaging and Image Processing, VIIP 2002 Conference*, J. J. Villanueva, editor, pages 417 – 421, September 2002.

- A. Monadjemi, B. T. Thomas and M. Mirmehdi. Speed v. Accuracy for High Resolution Colour Texture Classification. In: *Proceedings of the 13th British Machine Vision Conference, BMVC 2002*, P. L. Rosin and D. Marshall, editors, pages 143 – 152, BMVA Press, September 2002.

- A. Monadjemi, B. T. Thomas and M. Mirmehdi. Experiments on High Resolution Images Towards Outdoor Scene Classification. In: *Proceedings of the Seventh Computer Vision Winter Workshop*, Horst Wildenauer and Walter Kropatsch, editors, pages 325 – 334, Vienna University of Technology, February 2002.

To my dearest wife,

late father, and precious mother

Contents

List of Figures

List of Tables

Chapter 1

Introduction

You love what you see, and you see what you love.

(An Old Iranian Proverb)

1.1 Background and Motivation

For humankind, vision is the most important resource of information, hence the most important sense. Amongst several vision-based activities, *object recognition* and *classification* are regular, basic, and immediate acts. When one picks up a desired book from a table, an object recognition task has been implicitly performed: choosing a particular book within a scene full of other objects, possibly other books. In many applications, it would be decisively useful if we managed to develop an automatic visual pattern recognition system to assist or replace the human operator, for instance, a fast fingerprint identification system, a system for converting handwritten texts to computer text files, face recognition for security systems, outdoor scene object classification to help visually disabled people, and surface inspection of industrial products. These examples, all have something in common: to find the most important visual *properties* or *features* of an object that make it distinguishable from others. These properties can be colour, shape, edges, and texture, to name a few.

In a typical pattern recognition or object classification process, the first step is the extraction of features or key properties of objects (i.e. mapping from the real world to the feature space). The next step is classification of objects according to their features (i.e. mapping from the feature space to the classification space). The human brain is an excellent *classifier* which can successfully classify objects in noisy environments even without significant features. However, we still cannot expect the same performance from our artificial classifiers. Therefore, to work towards a successful classification, extracted features of different objects must show adequate separation in the feature space.

Figure 1.1 illustrates the structure of a traditional pattern recognition system. The two main stages, *feature extraction* and *classification*, eventually map the input object into one of the K classes of the classification space.

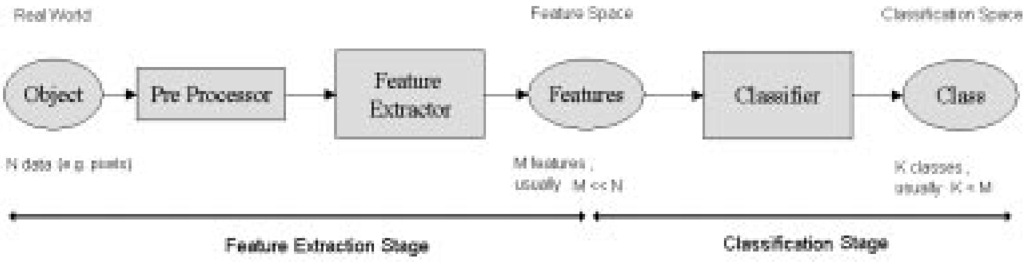


Figure 1.1: A pattern recognition system.

Huge efforts in the field of automatic pattern recognition during the last few decades have improved the overall performance of automatic recognition systems. However, even in constrained tasks, such as automatic registration of car number plates or handwritten character recognition, the lack of efficiency, particularly in robustness and flexibility, is still an important issue. In other words, even though a recognition system A performs well in the recognition and classification of pattern set α under given conditions γ, it will not guarantee that the probability of successful classification, $P(A)$, on other patterns or under other conditions would be high too:

$$P(A) = \mathcal{F}(A, \alpha, \gamma) \tag{1.1}$$

To conclude, some effort is still needed in the field of pattern recognition to increase the quality and performance of pattern recognition systems. This thesis considers the field of *texture analysis and classification*, and its application in automatic pattern recognition as the main subject of its study.

1.2 Overview

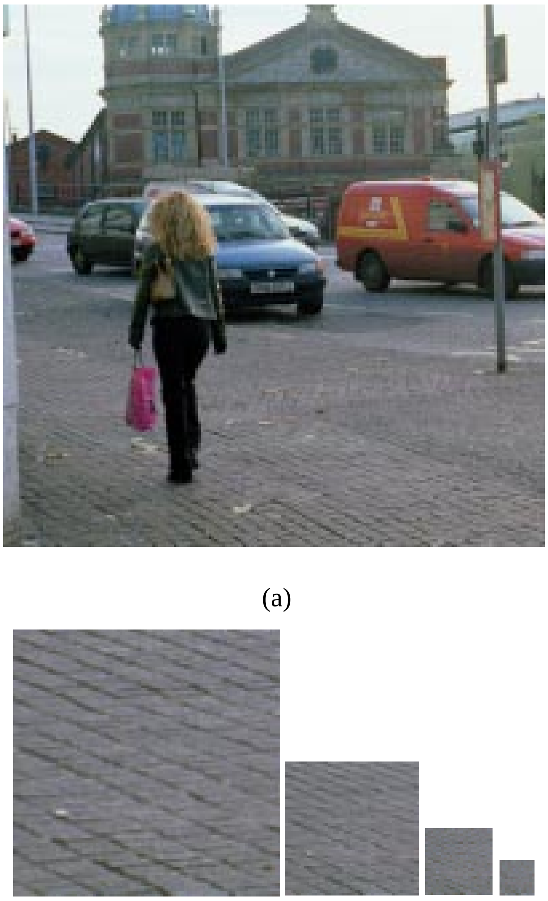
> *"The development of computational formalisms for segmenting, discriminating and recognising image texture projected from visible surfaces are complex and interrelated problems. An important goal of any such formalism is the identification of easily computed and physically meaningful image features which can be used to effectively accomplish those tasks." [16]*

In recent years, the computer vision research group at the University of Bristol has developed a neural network based system for classifying images of typical outdoor scenes to an area accuracy of approximately 90% [23]. The system is trained with features extracted from segmented regions of a large image database with images of typically 512 × 512 resolution. One of the issues investigated in this thesis is whether there is any advantage in utilising higher resolution images in outdoor scene object classification. Compared to ordinary images, different objects in higher resolution photos show more explicit textural properties. Again, in a classification task, by employing higher resolution images we will be able to extract larger patches of different objects. Therefore, methods applied (e.g. filtering) can use a wider range of spatial frequencies or spatial distances. This is particularly useful in texture analysis where essential characteristics of a texture, such as patterns and edges, are mapped on a rather broad range of spatial frequencies. Figure 1.2(a) illustrates a high resolution outdoor scene image. Figures 1.2(b) to (e), show a pavement patch in four successive resolutions, demonstrating declining textural detail of the pavement.

Recent research on the human visual system suggests that receptive field neurons in the human visual cortex show orientation-selective and spatial-frequency-selective properties [70]. This justifies the popular use of multi-scale and multi-directional (MSMD) schemes in image processing, for example in texture analysis, where textures usually show an obvious MSMD structure. We propose and investigate a novel version of the Walsh-Hadamard transform, called the *Directional Walsh-Hadamard transform* or DWHT in the context of a MSMD framework. The Walsh-Hadamard transform is one of the fastest and computationally cheapest transforms. The proposed DWHT can be precise and cost-effective in texture analysis applications. To evaluate the DWHT method, its performance is compared with the Gabor filter which is a widely used MSMD algorithm. Colour features are also employed in our outdoor scene object classification experiments. We introduce two hue-like and saturation-like colour features and compare them with colour features extracted from standard colour spaces HLS and Lab. The proposed chromatic features show good classification accuracy and speed as well.

The experiments performed on the outdoor scene images were part of a large scale project which dealt with wearable computers and supportive tools for partially blind people. For the time being, it is not feasible to embed high resolution imaging tools in such systems. However, as hardware facilities improve in time, this may become practical.

Furthermore, the fast and cheap DWHT, as proposed here, is feasible and it is worthwhile to compare it with more costly algorithms (e.g. Gabor filtering) under real circumstances. Hence, further experiments and comparisons are reported in this thesis performed on the standard texture test suite VisTex to measure the reliability and generality of results applied to images of a more typical resolution.



(a)

(b) (c) (d) (e)

Figure 1.2: A high resolution 2048 × 2048 pixel outdoor scene image (a), and a pavement patch in four successive resolutions, from the highest 256 × 256 pixels (b) to the lowest 32 × 32 pixels (e).

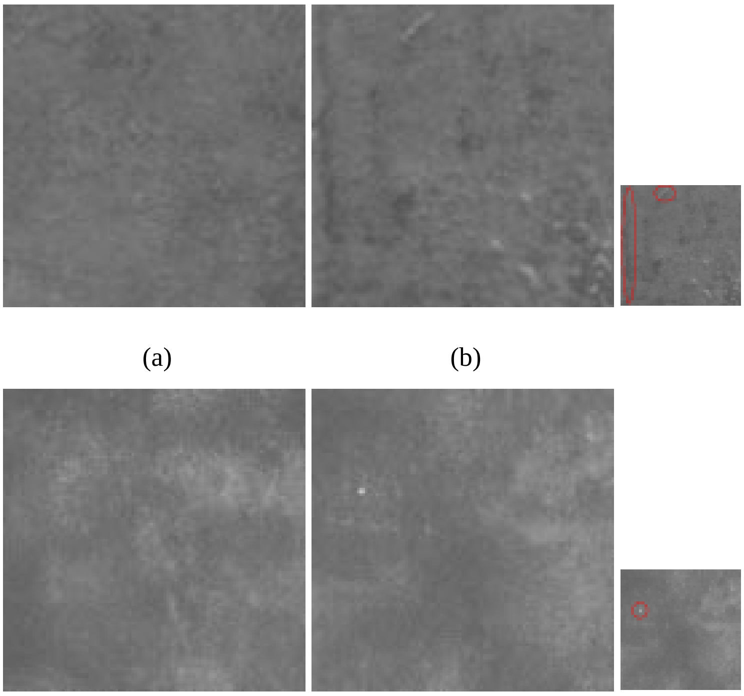
Attention is also paid to another field of texture analysis and classification: Detection of abnormalities in randomly textured ceramic tiles. Quality ranking of tiles is an essential stage in the tile manufacturing industry and development of an automatic surface inspection and defect detection system would have an impressive impact on the overall performance of a tile production plant. Figure 1.3 shows normal and abnormal samples of two textured tiles selected from our tile database.

We shall use the term ‘textural abnormality’ to refer to all possible defects, such as cracks or broken edges, colour or water drops, shading problems and so on. Using this definition, any defect is an unexpected change in the typical texture of a tile. Therefore, we emphasise on texture abnormality detection methods, and review, develop and test many texture abnormality detection algorithms on our tile data set which includes several types of randomly textured tiles. Experimental methods are based on statistical analysis (e.g. co-occurrence matrix and local binary pattern, LBP) or signal processing (e.g. DWHT, Gabor filters, PCA and directional discrete cosine transform (DDCT)). Also a new Gabor Composition scheme (GC) is introduced and implemented. The proposed GC scheme, which is in fact a combination of Gabor filtering and co-occurrence analysis, is on average the best of the tested algorithms.

In our texture classification and defect detection experiments described in Chapters 3 and 4, we employed two different classifiers: a back propagation neural network (BPNN) and a K-nearest neighbourhood (KNN). In a move away from such traditional approaches, in the final part of the thesis we develop a new *novelty detection* (ND) method for texture abnormality detection. The most important advantage of novelty detection in industrial inspection is its independence from defective samples. In other words, while ordinary classifiers need both normal and abnormal samples for a successful training, a novelty detector only employs normal samples.

The proposed algorithm reconstructs a given texture twice, once using a subset of its own eigenfilter bank, and once again using a subset of a reference eigenfilter bank, and measures the reconstruction error as the level of novelty. We then present an improved reconstruction, generated by structurally matched eigenfilters through rotation, negation, and mirroring. Experiments on tile defect detection show that this method can perform very well.

The two major applications that we dealt with (outdoor scene object classification and randomly textured tile defect detection), required a balance between accuracy and the computational



(a) (b)

(c) (d)

Figure 1.3: Two pairs of normal and abnormal tiles from two different types. Top: A normal (a) and an abnormal (b) `PRODO`. Bottom: A normal (c) and an abnormal (d) `KIS`. The defective areas have been highlighted in the small images on the right.

costs. Performance results for all experiments (when appropriate) is presented, and the overall performance of the proposed methods suggests that they can provide a good balance between accuracy and computation cost.

1.3 Contribution

The contributions of this thesis are:

- A novel multi-scale/multi-directional Walsh-Hadamard transform, DWHT, which is fast and easy to implement, and hence is suitable for realtime applications.

- Two easy to compute chromatic features based on the definition of hue and saturation. These features can be used in colour texture classification.
- A Gabor Composition method for detection of abnormalities in random textures. The algorithm highlights the defective textures by combining Gabor filtering and co-occurrence analysis.
- An eigenfilter based reconstruction method for texture novelty detection. The proposed method utilises *restructured* eigenfilters to reconstruct the texture, and then considers the reconstruction error as the indicator of abnormality.

1.4 Thesis Layout

This thesis is divided into 6 chapters. After the introduction part, **Chapter 2** provides a general review of texture analysis literature and the methods employed in this thesis. Definitions of a texture and related terms are reviewed to provide a clearer approach to the subject of the study. It also contains a review on surface inspection and texture abnormality detection. Finally, methods used such as neural networks and principal component analysis are briefly introduced.

In **Chapter 3**, methods for feature extraction and classification of objects in high resolution colour images are presented. Textural features are obtained from a novel multi-band and directional Walsh-Hadamard transform, as well as simple chromatic features that correspond to hue and saturation in the HLS colour space.

In **Chapter 4**, a study in normal/abnormal textures classification experiments is presented. The two proposed methods (DWHT and GC) are applied and compared in terms of accuracy and speed against other established and optimised texture classification methods, such as Gabor filters and co-occurrence matrices on a data set of normal and defective textured ceramic tiles.

In **Chapter 5**, a new eigenfilter-based novelty detection approach to find abnormalities in random textures is presented. The method is accurate and fast, and amenable to implementation on a production line.

The thesis is concluded in **Chapter 6**.

Chapter 2

Texture Analysis and Classification: Background and Methods

2.1 Introduction

In this chapter, we briefly review the field of texture and texture analysis. We begin with the definition of texture and related terms in Section 2.2. Then, diverse approaches to texture analysis and classification are discussed in Section 2.3. New methods for random texture analysis are introduced in Section 2.4. Section 2.5 briefly reviews some previous studies in colour texture processing. Section 2.6 provides an overview to texture inspection and abnormality detection. Finally, Section 2.7 summarises the methods used in this thesis.

2.2 Texture: Definitions

‘Texture’ is a widely used and implicitly understandable term, however as many other intuitively known phenomenon, there is no precise definition. In the Webster dictionary, texture is defined as “the character of a surface as determined by the arrangement, size, quality, and so on” or “the arrangement of the particles or constituent parts of any material as it affects the appearance or feel of the surface”. Some other specific and technical definitions found in machine vision literatures are “discrete 2D stochastic field with a given governing joint probability density function” [97] or “repetitive arrangement of a unit pattern over a given area” [101]. Humans usually describe a given texture by words like fine, coarse, smooth, rough and so on. These attributes are again instinctually obvious, however still relative and not easily measurable [113].

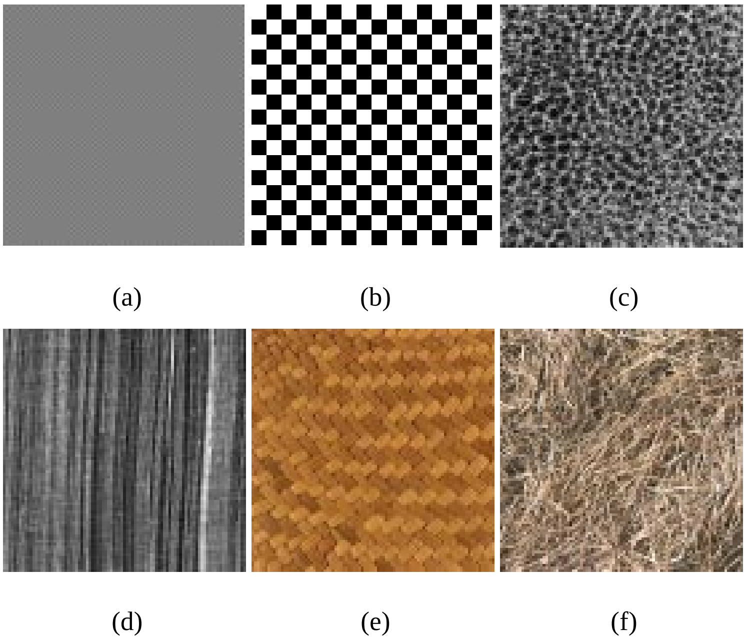
As with many other analyses, a reasonable approach to describe a texture could be extraction and definition of its *primitives* or *elements*, which usually are referred as *textons* [127, 131], along with the description of the inter-primitives relations. We can refer to the internal properties of a primitive (e.g. intensity or colour of the pixels) as the *tone* and spatial inter-primitive relationship as the *structure* [44, 113]. Consequently a countable set of primitives with distinguishable tones and their structure describe the texture. However, for many natural textures, it is not very easy to determine the primitive set and the structure.

Textures could be categorised according to their strength or cohesion feature. A *constant* texture, is constant, slowly changing or approximately periodic. In a *strong* texture, the primitive set is well defined and the structure is rather regular. In other words, elements and spatial relations between them are clearly determinable. While in a *weak* texture definition of a crisp set of primitives is relatively more difficult and spatial correlation between primitives is also low. An extremely weak texture could be considered as a *random* texture [113]. Again, textures can be categorised as *fine* or *coarse*. In a fine texture, primitives are small and the contrast between primitives are high. In contrary, in a coarse texture primitives are relatively large. However, all the definitions above are relative, particularly for natural textures. Also, texture is a property of area, therefore texture measures are dependent on the size of the observation (i.e. patch size) and also the resolution [97].

As an example, Figure 2.1(a) and (b) illustrate that the fineness and coarseness are scale-dependent attributes. In fact, the coarser checkered texture (b) is a patch of the finer (a) after 16 times zooming-in. Figure 2.1(c) and (d) show two natural textures selected from the pseudo-standard Brodatz texture album [18]. Although (c) is more regular than (d) and can be assumed as a strong texture, defining a clear set of primitives as the building blocks of (c) is still difficult. For a weaker texture like (d), it is almost impossible to determine primitives and structure in current resolution. Figures 2.1(e) and (f) are two natural textures from the VisTex texture suite [69]. Again it is not clear how one can define primitives of a weak/random texture like (f), while it is an easier task for (e).

Figure 2.2 depicts a fuzzy-like separation of different textures according to their strengths and the corresponding degrading/increasing properties. While for a constant texture primitives and the structure are well-defined and strength is high, for random textures they are ill-defined and low.

Texture analysis covers a wide range of applications: medical image analysis, scene understanding, remote sensing, textured surface inspection, document processing and many more. Next, we categorise different approaches to texture analysis, with special attention to texture classification. Our study emphasises on the lower level texture processing. Although during recent years an obvious shift of interest from low level to high level vision has occurred in machine vision, low level processes are still an active field of study. High level processes are not independent from low levels, and there are still a lot of unanswered questions in the field of low level vision and image processing [31].



(a) (b) (c)

(d) (e) (f)

Figure 2.1: Examples of artificial and natural textures. (a) and (b): A similar ‘checkered’ artificial texture in high and low resolution representations. (c) and (d): Two natural textures from Brodatz album, (c) is D111 and (d) is D105. (e) and (f): Two colour natural textures from VisTex set. (e) is a fabric and (f) is a grass.

2.3 Texture Analysis and Classification: Different Approaches

Sonka *et al* [113] state that there are two main approaches to texture analysis: statistical and syntactic. They consider auto correlation, discrete image transform, ring/wedge filtering, grey level co-occurrence matrices (GLCM), (or dependency matrices [66]), and mathematical morphology as popular statistical texture analysis methods, and shape chain grammar and primitive grouping as syntactic methods. In a more comprehensive categorisation, Tuceryan and Jain [116] distinguish four different approaches to texture analysis: statistical, geometrical, model-based and signal processing approaches. Using the later categorisation, geometrical (e.g. Voronoi tessellation or region growing) and model-based approaches (e.g. Markov ran-

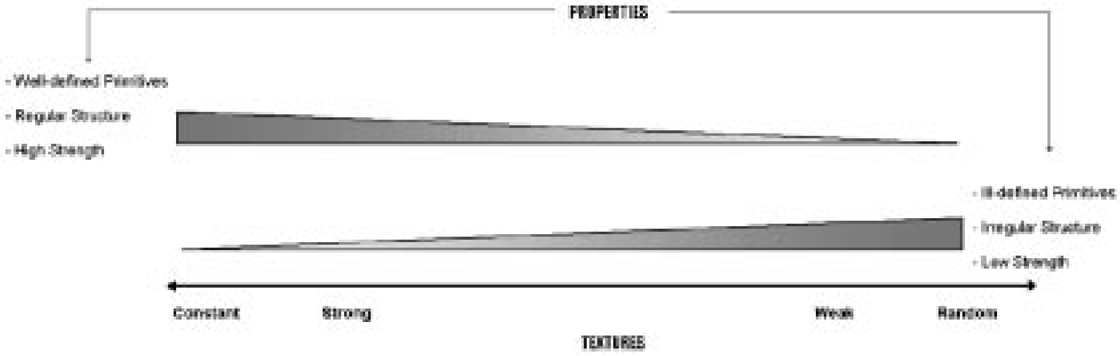


Figure 2.2: Definition of different textures and their properties.

dom fields or fractals) are not of interest in this thesis. Therefore, we only review statistical and signal processing approaches and their advantages and disadvantages.

2.3.1 Statistical Approaches

Statistical texture analysis methods deal with the distribution of grey levels (or colours) in a texture. The first order statistics and pixel-wise analysis are not able to efficiently define or model a texture. Therefore, statistical texture analysis methods usually employ higher order statistics or neighbourhood (local) properties of textures. The most commonly used statistical texture analysis methods are co-occurrence matrices, autocorrelation function, texture unit and spectrum, and grey level run-length [49, 113, 116].

Co-occurrence Matrices: Introduced by Haralick [45], GLCM is one of the earliest texture analysers which is still of interest in many studies. Since the beginning of the 70's many researchers have studied GLCM theory and have practically implemented it in a wide range of texture analysis problems.

GLCM is a model that can explicitly represent the higher order statistics of an image, just like ordinary histograms which represent the first order statistics of images. For an N-grey level image, x, the GLCM which captures the second order statistics and presents them in $N \times N$ matrices, is defined as:

$$\Phi_{d,\theta}(i,j) = \sum_{u=1}^{U} \sum_{v=1}^{V} \rho(x(u,v), x(u',v'), i, j) \tag{2.1}$$

where the image size is $U \times V$, d and θ are distance and direction between pixel pair $< x(u,v), x(u',v') >$ and ρ is:

$$\rho(x(u,v), x(u',v'), i, j) = \begin{cases} 1 & \mathrm{If\ x(u,v) = i\ \ and\ \ x(u',v') = j} \\ 0 & \mathrm{other\ wise} \end{cases} \tag{2.2}$$

In fact $\Phi_{d,\theta}(i,j)$ shows the number of occurrence of grey level pair $< i, j >$ between pixels at d distance and θ direction of each other. For instance, the expression below illustrates a given 4×4 image and one of its GLCM matrix with $d = 1$ and $\theta = 0^\circ$.

$$x = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix} \quad \Rightarrow \quad \Phi_{1,0^\circ}(x) = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix} \tag{2.3}$$

There is no generally accepted solution for optimising d and θ, however, having $d = 1$ and $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ is typical. The next step is usually extracting more condensed texture features by applying some appropriate functions on Φ. GLCM and its parameter setting and functions will be discussed in detail later in Section 4.3.3.

There are several reports on relatively successful implementations of GLCM in texture analysis and classification, for instance [44, 66, 100]. Moreover, recently Partio *et al* [94] utilised

GLCM to retrieve rock textures, where GLCM features performed better than Gabor wavelet-based features. Also Clausi [24] employed GLCM to classify SAR images. Clausi also reviewed several former implementations of GLCM, mostly on the field of remote sensing, and posed certain questions about their algorithms and results, in particular the methods used for parameter optimisation. Again, the role of grey level quantisation on the GLCM performance were discussed in that study.

Autocorrelation (AC) function: The AC function is defined as:

$$AC_{\Delta_u,\Delta_v}(x) = \frac{\sum_{u=1}^{M}\sum_{v=1}^{M} x(u,v)x(u+\Delta_u, v+\Delta_v)}{\sum_{u=1}^{M}\sum_{v=1}^{M} x^2(u,v)} \tag{2.4}$$

where x is the $M \times M$ image, Δ_u and Δ_v, are horizontal and vertical displacements. The AC function can assess the regularity and fineness/coarseness of the texture. The autocorrelation function of a coarse texture drops off slowly and vice versa. Again, the autocorrelation function of a regular texture exhibits clear peaks and valleys. Although it is possible to find some different artificial textures with a similar autocorrelation function, this does not necessary rule out the utility of an AC feature set for natural texture classification [97]. In general however, the AC function is not considered a highly effective and popular texture classification tool.

Texture unit and spectrum (TUS): Introduced by He and Wang [47], TUS firstly replaces the texture's pixels with *texture units* (TU), which are functions of a rather small neighbourhood around each pixel (e.g. 3×3), and then computes the distribution (e.g. histogram) of TUs over the mapped image as the *texture spectrum*. Many of the proposed neighbourhood functions are in fact a mixture of simple logical operators and weighted summation of neighbourhood pixels. For instance, a pixel can be replaced by sum of its brighter neighbour pixels. He and Wang employed their method in texture classification and unsupervised segmentation, and textural filtering, however, the excessive dimensionality of feature space (e.g. 6561 features in [47]) limited the method's practicality.

Local binary pattern (LBP) was introduced by Ojala *et al* [92] as a TUS-based grey level shift invariant texture descriptor. The basic LBP operator considers a 3×3 neighbourhood of a pixel, then these 8 border pixels will be replaced either by 1, if they are larger than or equal to the central pixel or by 0 otherwise. Finally, the central pixel will be replaced with a summation of the binary weights of border pixels in the LBP image and the 3×3 window slides to the next pixel.

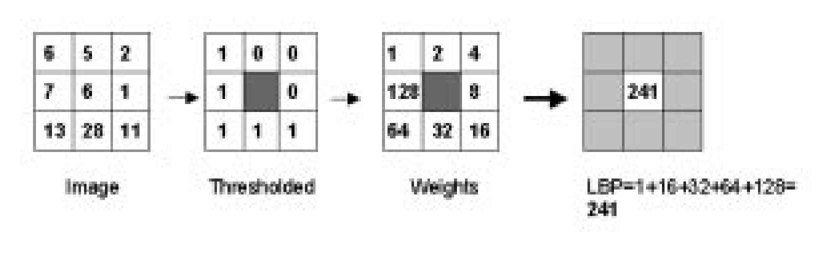


Figure 2.3: Computing the basic 3×3 LBP (From [80]).

It is possible to develop the basic LBP into various neighbourhood sizes and distances [93]:

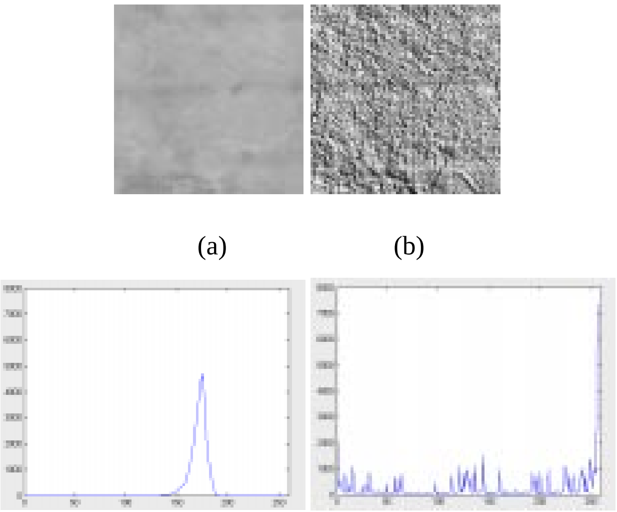
$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{2.5}$$

where $s(\cdot)$ is the sign function:

$$s(x) = \begin{cases} 1 & , x \geq 0 \\ 0 & , x < 0 \end{cases} \tag{2.6}$$

g_p and g_c are grey levels of border pixels and central pixel respectively, and P is the number of pixels in the neighbourhood.

In this case, if we set $(P = 8\ ,\ R = 1)$, we obtain the basic LBP (see Figure 2.3.1). Luminance changing cannot affect signed differences $g_p - g_c$, hence LBP is grey level shift invariant. Whereas ordinary LBP is not rotation invariant, it is possible to modify it to a rotation invariant version [93]. Typically the 256-bin histogram of the LBP is considered as the texture descriptor. However, when a $P > 8$ is used, the LBP range exceeds far beyond $2^8 = 256$ and it may be necessary to select a subset of P to decrease the maximum value of LBP. Figure 2.4 shows a textured ceramic tile, its basic LBP map and the corresponding histograms.



(a) (b)

(c) (d)

Figure 2.4: A textured ceramic tile (a), its basic LBP map (b) and corresponding histograms ((c) and (d)).

We will utilise Local Binary Patterns algorithm later in Section 4.3.2 in a texture defect detection experiment.

Grey level run-length or primitive-length (GLRL): In this method, the primitive set is defined as the maximum set of continuous pixels of the same grey level, located in a line. The length of primitives (run-lengths) in different directions can then be used as the texture descriptors. A longer run-length implies a coarser texture and vice versa, also a more uniformly distributed run-length implies a more random texture and vice versa. Statistics of the primitives can be computed as the texture features. For example, let $B(g,r)$ be the number of primitives of the length r and grey level g, N the number of grey levels, and N_r the maximum run-length of the texture. Then K is the total number of runs:

$$K = \sum_{g=1}^{N} \sum_{r=1}^{N_r} B(g,r) \tag{2.7}$$

and texture uniformity measure can be defined as:

$$\frac{1}{K} \sum_{g=1}^{N} \sum_{r=1}^{N_r} B(g,r)^2 \tag{2.8}$$

Primitives should be computed for all grey levels, lengths, and directions, which is a costly process. Again, implementation of GLRL on grey scale textures is not straightforward, since some considerations on quantisation tolerance should be satisfied. Also GLRL has not shown promising results in many texture classification experiments. For instance, in [108], applied on a specified benchmark [88], GLRL performance is the lowest one with around 45% correct classification, and almost 30% less than the AC and GLCM in the same experiment.

2.3.2 Signal Processing Approaches

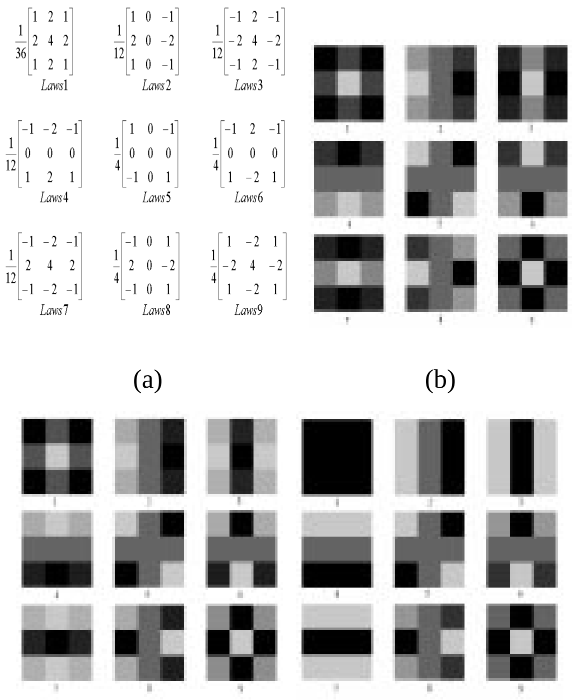
Signal processing approaches cover a wide range of spatial and transform domain filtering, discrete transform domain analysis, and multi-scale/multi-directional (MSMD) methods. Signal processing schemes, which indicate the texture as a 2D digital signal, are very popular and capable of dealing with random as well as regular textures.

Spatial domain filtering: A texture can be considered as a mixture of patterns, therefore characteristics of ‘edges’ and ‘lines’ are key elements to describe any texture. Even a plain or smooth texture can be considered as a texture without any edge. The early attempts to utilise spatial domain filtering as texture descriptor were emphasised on gradient (i.e. line and edge detector) filters such as Robert and Sobel operators [97, 113]. Moreover, Laws [72] proposed his nine 3×3 pixel filter set (see Figure 2.5(a) and (b)) to extract the micro structure of textures. His method concerns filtering the texture by an empirical filter set and measuring the micro structures' *energy* (i.e. standard deviation of the responses). In a later study, Laws successfully employed 5×5 filters and a 15×15 sliding window absolute averaging scheme for texture segmentation [73].

The common term in all spatial domain filtering methods is 2D convolution of the texture with a set of relatively small filters (i.e. filter bank) and then processing the filter responses. It is also possible to implement small size discrete sin (DST), cosine (DCT) or Hadamard filters instead of Laws filters. In a series of works, Unser established a platform of small size spatial domain filters (which he calls *local linear transform*, LLT) for texture analysis and classification [117, 118]. Figure 2.5(c) and (d) illustrate 3×3 DCT and DST filters. Apparent similarity between these filters and Laws filters suggests that all methods use similar principles and may have similar performances.

Eigenfilters (or similarly Karhunen−Loeve transform, KLT) are another alternative for spatial domain texture analysis. Although they look like Laws and other gradient filters (see Figure 2.6), compared to Laws filters they have two additional important features: adaptability and orthogonality. Adaptability means each image has its individual eigenfilter bank which is extracted from its covariance matrix using a principal component analysis (PCA) scheme. Orthogonality means the eigenfilter bank is orthogonal, hence it can decompose an image into a set of uncorrelated detail (or basis) images, and regenerate the image by re-composition of detail images [2]. Details of PCA and eigenfilters will be discussed later in Sections 2.7.1 and 4.3.7. Figure 2.6 depicts a randomly textured ceramic tile (a), its nine 3×3 eigenfilters (b), and detail images (c).

Various authors have suggested that KLT is one of the best texture analysers. For instance, regarding its adaptive nature, Unser [118] considered the KLT as the optimum LLT, and indeed in



(a) (b)

(c) (d)

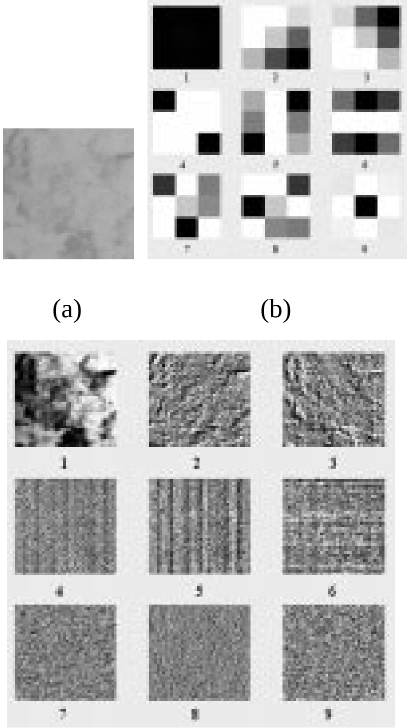
Figure 2.5: Nine 3×3 Laws filters ((a) and (b)), DST filters (c) and DCT filters (d). To increase the visibility, all filters have been equalised.

his experiments KLT performed better than all other local linear transforms. More clarification of the KLT transform and eigenfiltering can be found in [120].

Fourier domain analysis: The Fourier Transform, and its fast version, FFT, is a basic tool for harmonic analysis of images:

$$F(u,v) = \frac{1}{\omega^2} \sum_{x=1}^{M} \sum_{y=1}^{M} f(x,y) e^{\frac{-2\pi j}{\omega}(ux+vy)} \tag{2.9}$$

The complex FFT represents magnitude ($|F(u,v)|$, namely absolute value or power spectrum density) and phase ($\angle F(u,v)$) information of the signal in the frequency domain. Power spectrum density (PSD) is directionally symmetric and represents global frequency contents of an image. Therefore, regarding special attributes of textures, typical PSD analysis (e.g. employ-



(a) (b)

(c)

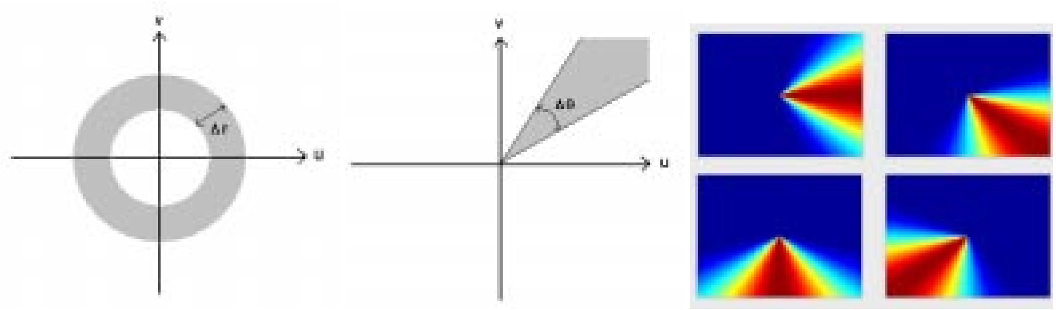
Figure 2.6: 3×3 Filter bank (b) and detail images (c) of a randomly textured tile (a).

ing PSD moments as features) may not be sufficient for extracting efficient texture features. A solution is processing and analysing the FFT output to obtain high performance textural features. For instance, D'Astous and Jernigan utilised detailed measures of the FFT domain for texture discrimination [30]. They proposed two groups of PSD-based features: peak features and power distribution features. Strength, curvature, area and distance-to-centre are examples of their peak features, and difference between vertical/horizontal direction variances, power spectrum eigenvalues and the circularity of the PSD are some of their power distribution features. Chan and Pang studied fabric defect detection by Fourier analysis [19]. They applied FFT domain analysis in x and y directions since many fabric defects occur in those directions. Their proposed features were the first and the second peaks or harmonics of the horizontal and vertical 1D slices of the power spectrum (e.g. $|F(u,0)|$ and $|F(0,v)|$ projections of $|F(u,v)|$). However, while FFT is a very fast transform, many of the proposed subsequent features of the above-mentioned methods are computationally costly.

Ring and Wedge filters (RF/WF) are another commonly used FFT based texture analysis method [53, 113] and can be defined as:

$$\begin{cases} RF(\Delta r) = \sum_{(u^2+v^2)\in\Delta r} |F(u,v)|^2 \\ \\ WF(\Delta\theta) = \sum_{\arctan(\frac{u}{v})\in\Delta\theta} |F(u,v)|^2 \end{cases} \tag{2.10}$$

where $|F(u,v)|$ is the power spectrum. Figure 2.7(a) and (b) show a ring and a wedge filter.



(a) (b) (c)

Figure 2.7: A ring filter (a), a wedge filter (b), and four wedge filters with $\Delta\theta = 45^\circ$ and Gaussian envelope.

A ring filter, which is indeed a symmetric band-pass filter, can reveal the distribution of texture's energy across the frequency domain and measure its fineness/coarseness. A wedge filter in $\Delta\theta$ passes the energy in $\Delta\theta + \frac{\pi}{2}$ orientation, thus can evaluate the directionality of the image. As Figure 2.7(c) illustrates, to decrease the harmful side lobes a Gaussian envelope may cover the filter and smooths its edges [96].

Discrete transforms: Rather than the FFT, it is possible to apply other discrete transforms or harmonic analysers such as discrete cosine transform or Walsh/Hadamard transforms (DCT and WHT respectively) for texture analysis. Each discrete transform has its own advantages and disadvantages. For instance, while FFT is complex, DCT and WHT are real, hence easier to handle. FFT and DCT use sinusoidal kernel functions, whereas WHT uses less accurate but faster square kernels and therefore is easier to implement. However, in the case of WHT or DCT, generally some modification should be carried out on the original transform, to make it more suitable for texture processing. For example, presented in this study, the new DWHT method which will be described later in Chapter 3, is in fact a modified MSMD version of WHT with a better performance in texture classification.

MSMD schemes: Several studies on early stages of the human (and other mammals) visual system (HVS), suggest that we decompose the input image into detail images of various spatial frequencies (scales) and orientations. In other words, retina cells are selective, and different cells respond to different scales and orientations [9, 70]. Inspired by this biological theory, MSMD methods have been developed and tested on texture classification, segmentation and synthesis applications [23, 54, 107]. Although there are some reports on multi-scale LBP [93] and GLCM [122] techniques, MSMD methods are mostly based on either Gabor filters [25, 67, 90], or wavelet analysis [12, 78, 96, 107]. In the spatial frequency domain, a Gabor transform can be interpreted as a windowed or short-time Fourier transform. A Fourier transform is a global frequency content analysis. Instead, a windowed Fourier transform is a local analysis which will be obtained by multiplying the input signal by a window [4]. If the window function is a Gaussian, the transform will be a Gabor transform [116].

A 2D Gabor filter can be defined in both spatial ($G(x,y)$) and spatial-frequency domain ($\mathrm{G}(u,v)$) as:

$$\begin{cases} G(x,y) = e^{-\pi[(x-x_0)^2\sigma_u^2+(y-y_0)^2\sigma_v^2]}.e^{-2\pi j(x_0u+y_0v)} \\ \\ \mathrm{G}(u,v) = e^{-\pi[\frac{(u-u_0)^2}{\sigma_x^2}+\frac{(v-v_0)^2}{\sigma_y^2}]}.e^{-2\pi j[x_0(u-u_0)+y_0(v-v0)]} \end{cases} \tag{2.11}$$

Gabor filter parameters will be discussed later in Section 3.3.1. In the spatial domain, a Gabor filter is a sinusoid wave modulated by a Gaussian envelope. The standard deviation of the Gaussian envelope determines the filter bandwidth, while the direction and frequency of the sinusoid signal tune the direction and frequency of the passing band. Gabor filters in frequency domain are Gaussian bell-shape filters with different horizontal and vertical central frequency and bandwidth, placed in various orientations. Therefore, they are frequency and orientation selective filters. There are two major ways to optimally choose parameters of a Gabor filter: *supervised* and *unsupervised*. In a supervised manner, several sets of parameters are tried to find out the optimum filter (or a few filters) for a given problem. Whereas in an unsupervised manner, a filter bank which spreads throughout the frequency plane is used. The unsupervised method is more general and more popular, however dealing with a filter bank means a higher computational cost and a larger feature space [54]. It is also of importance to optimise Gabor filter bank parameters, namely central frequencies, bandwidths, and directions, and select effective Gabor-based features.

Utilising 1-octave difference between central frequencies is typical and also confirmed by some studies on HVS. This means that for two successive central frequencies in the filter bank, ω_i, and ω_{i+1} we have $log_2(\omega_{i+1}/\omega_i) = 1$ (or $\omega_{i+1} = 2\omega_i$). The bandwidth of higher frequencies is wider than the lower frequencies and half-power bandwidth would be considered as well, where the point of intersection is on half magnitude of two successive filters. This configuration results in a *dyadic* Gabor filter bank [9, 23]. Although biological evidence considers $\Delta\theta = 30^\circ$ for HVS cortex directional resolution [25], many researchers have found $\Delta\theta = 45^\circ$ adequate, e.g. [54].

In their study, Grigorescue *et al* [43] compared a variety of Gabor based texture features. In particular Gabor energy, complex moments and grating cell operator features were evaluated by both Fisher criterion and classification results. The key point of their work was benefits of *post-Gabor processing* using grating cell operators. This operator which is a computational model of a specific type of neuron found in visual cortex of some monkeys, signals the presence of 1D periodicity of particular spatial frequency and orientation in 2D images. To be more specific, a grating cell only responds when a set of at least three bars of a given direction and spacing is present in the receptive field. The response increases with the number of bars but will saturate soon. Classification tests suggest that the grating cell operator performs more effectively than Gabor energy and complex moments in texture segregation. Furthermore, in an effort to separate textures from other parts of image (e.g. edges or contours), the grating cell is the only one which does not give a false positive signal to non-texture regions. Meanwhile many other studies have employed non-linear blob detectors as the post-Gabor processing [54, 103].

Although a restricted Gabor filter bank can also be considered as a wavelet analysis tool, typical wavelets for texture analysis are based on a sequence of spatial domain filters applied on a pyramid-shape multi-scale structure of the image. There are several ways to implement a multi-scale wavelet technique. However, the formal and unified approach which was introduced by Mallat [81] is a well established and popular platform. A wavelet transform decomposes the input signal (e.g. an image) into an orthogonal set of wavelet sub-signals (detail images). There are certain interesting studies on wavelet-based texture classification and synthesis based on Mallat propositions and Gaussian/Laplacian pyramids, in particular Heeger and Bergen's steerable pyramids [37, 48], Portilla and Simoncelli's complex joint statistics [96, 107] and DeBonet's flexible histograms [11, 12].

Figure 2.8 depicts a way of generating multi-scale pyramids and extracting texture features. The input image x is low pass filtered by function f to generate the first Gaussian detail image L_1. If we want to keep 1-octave scaling, f can be a 2-times down sampling function:

$$L_1 = f(x) = 2^{\downarrow}(x) \tag{2.12}$$

and in general,

$$L_n = f(L_{n-1}) = 2^{\downarrow}(L_{n-1}) \tag{2.13}$$

where $2^{\downarrow}$ is 2-times down sampling operator. To obtain the high pass filtered Laplacian detail

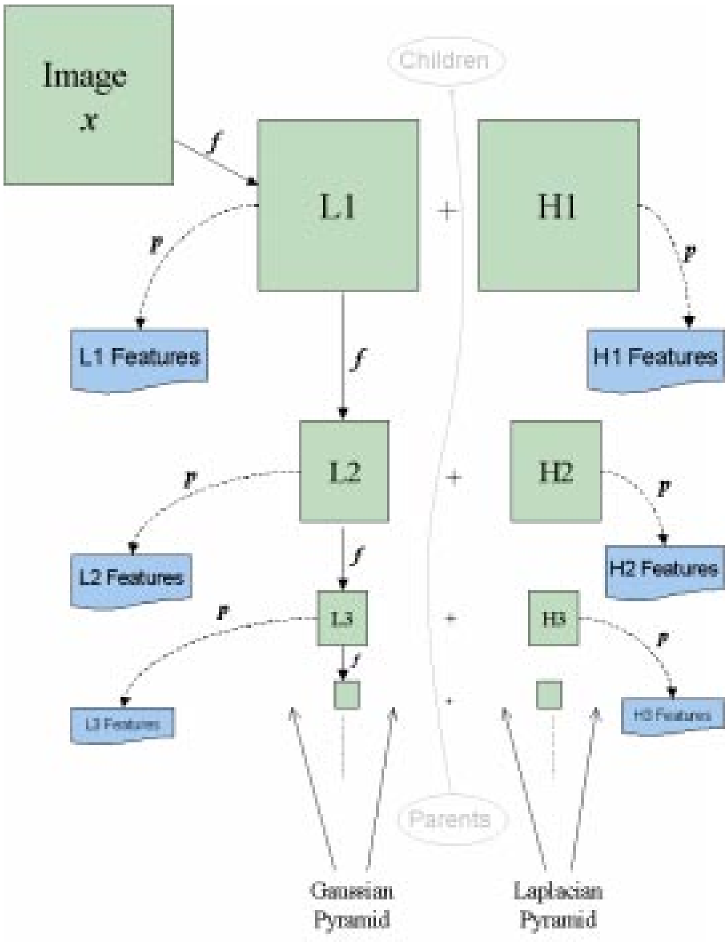


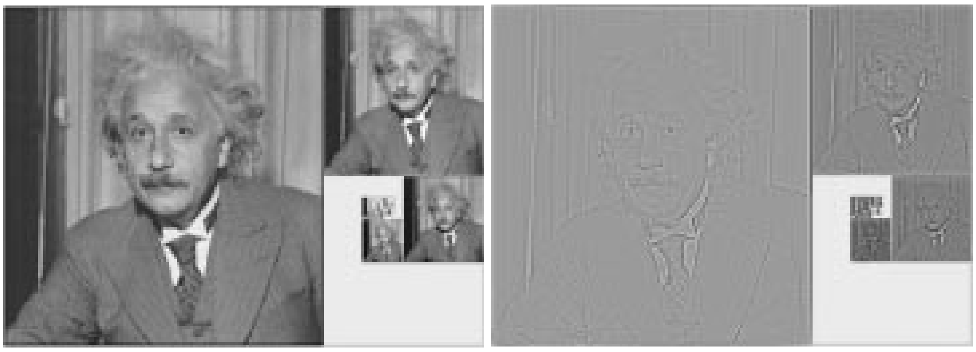
Figure 2.8: Wavelet algorithm and Gaussian/Laplacian pyramids. Parents/children path is defined in Debonet method [11, 12].

images, we can up sample L_n and subtract it from its Gaussian 'child', L_{n-1}:

$$H_n = L_{n-1} - 2^{\uparrow}(L_n) = L_{n-1} - 2^{\uparrow}(2^{\downarrow}(L_{n-1})) \tag{2.14}$$

where $2^{\uparrow}$ is 2-times up sampling operator. Sequences L_n and H_n are indeed different levels of Gaussian and Laplacian pyramids respectively. Figure 2.9 illustrates Gaussian and Laplacian detail images of a test portrait.

Texture feature extraction can be completed by applying directional filters (p_i) on different



(a) (b)

Figure 2.9: Gaussian (a) and Laplacian (b) detail images of a portrait.

levels of L_n and obtaining the responses. As an example, Figure 2.10 depicts 3×3 gradient filters which we utilised for a texture synthesis test. This set contains line detectors in vertical and horizontal direction (p_1 and p_2) and edge detectors in four directions (p_3 to p_6). A simple feature vector then comprises the statistics of H_n along with the statistics of low pass responses $p_i(L_n)$ for all levels and filters. Portilla and Simoncelli suggest that adding joint statistics of different levels and orientations (e.g. cross correlation of $p_i(L_n)$ and $p_{i+1}(L_n)$) can increase the classification performance. DeBonet instead exploits *flexible histograms*, where each bin contains the number of pixels with close *parent structures*. A parent structure is the filter responses of a pixel and all of its parents (i.e. pixels at the same position of the lower resolution levels of the Gaussian pyramid).

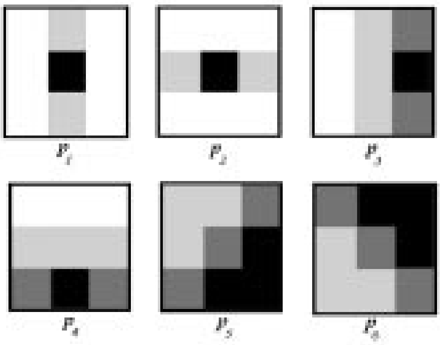


Figure 2.10: Six gradient filters which are employed for wavelet feature extraction.

These works all are applicable in texture synthesis as well, where they aim to produce a perceptually similar, but not copied version of a model texture. Heeger and Bergen start with a random noise and attempt to match its histogram and then sub-band histograms (i.e. histograms of pyramid levels and $p_i(L_n)$) with ones of the model texture to be synthesised iteratively. Portilla and Simoncelli employ a recursive procedure to match the marginal and joint statistics of a random texture and the model level by level, starting from the vertex of pyramid (i.e. lowest resolution). Synthesis mostly begins from the lowest resolution, whereas analysis usually begins from the highest. DeBonet proposes another method of synthesis which again begins from the lowest resolution and randomly swaps the pixels whose parent structures are similar enough, and continues toward the highest resolution.

2.4 Textons in Random Textures: A Different Approach

As mentioned before, modelling of natural and random textures based on the definition and extraction of textons or texture primitives is not a straightforward process. In particular it is not clear how to geometrically define elements of a random texture. However, there is a more stochastic way to model a texture and define its textons, based on a well-prepared set of filter responses and their statistics, clustered in the feature space. This signal processing approach is more suitable than geometrical methods for random textures modelling, since it provides a more operational way to deal with random textons [75, 123]. Amongst several studies which utilised this approach, Leung and Malik in [75], considered a texton as a cluster centre in the filter response space. Their filter bank comprised a few tens of asymmetric MSMD (with different scales and orientations), symmetric low pass and symmetric high pass filters. K-means clustering was then employed to cluster the filter responses into K clusters. Next, cluster centres were assumed as textons and built up a *texton dictionary*. In this study effects of both surface attributes and illumination on generating 3D textures were reviewed and a rotation-invariant method was introduced which represented different viewpoints and lightings. Cula and Dana [29] and Varma and Zisserman [123] utilised a basically similar method, but exploited diverse filter banks and defined texton histograms as the texture feature. In their work textons of a given texture were extracted and labelled using the reference texton dictionary. The Varma and Zisserman algorithm was rotation-invariant, since it exploited an energy-ordered directional filter responses and took the maximum energy. Schmid [105] employed a symmetric multi-scale filter bank and a two-layer constructing model for image retrieval. Her algorithm was also rotation-invariant and showed good performance in image retrieval experiments. We will refer to these schemes later in Section 4.4 as a justification for our proposed method toward texture classification using the innovative Gabor Composition approach.

2.5 Colour Texture analysis

Almost all of the studies on colour texture analysis either deal with colour channels R,G, and B as three individual signals, or transform the texture into another *colour space* (e.g. HLS or $\mathrm{L^*a^*b^*}$) and then process the chromatic planes as well as the intensity one. For instance, Baldrich *et al* [5] established a study on tile classification using colour features in the RGB space. Their work was focused on colour inconstancy detection using a K-means coloured-blob segmentation and there was no signal processing approach in their report. Kittler *et al* [64] on the other hand, utilised both RGB and $\mathrm{L^*a^*b^*}$ (for simplicity we may show it as *Lab*) spaces for colour clustering as a part of their defect detection scheme in colour textures. The initial clustering was carried out in the RGB space, then since inter cluster distances in the RGB space did not fully convey the perceptual distances between colours, clusters were transformed to the *Lab* space where the clusters merged together.

Colour texture classification will be explained further in Section 3.4. Meanwhile, the application of colour-based methods in this thesis is limited. In Chapter 3, we mostly focus on the texture greyscale features, then utilise the chromatic features to enrich the set. Moreover, the tile data set of Chapters 4 and 5 only contains greyscale images, therefore we do not apply any colour texture processing further than the third chapter. HLS and *Lab* colour spaces are discussed in Appendix A

2.6 Texture Inspection and Abnormality Detection

Quality ranking and defect detection of randomly textured ceramic tiles is one of the major topics to be investigated in this thesis. Surface inspection and abnormality detection is a particular case in texture classification, where the algorithm attempts to inspect a surface for possible defects, to classify the input sample as either *normal* or *abnormal*, or to rank its quality. In fact detection of textural abnormalities is a vital part of many systems and applications such as clinical checkup systems [66, 86], surface inspection of industrial products [6, 13, 46], food products inspection [32], and remote sensing [22, 116]. Apart from typical advantages of an automated system in industrial and clinical applications, in some cases such as underwater apparatuses and space crafts automatic inspection is inevitable, since the environment is too hazardous for human operators [22].

2.6.1 How to Detect a Textural Abnormality?

One of the earliest attempts on texture discrimination was Julesz works which started in the early 60s [59]. In a series of studies, she emphasised the fabulous human ability to distinguish between textures and tried to extract some reliable and applicable facts from that process. She stated that the human texture discrimination process could be divided into two categories of *effortless* which is fast and *pre-attentive* (PA) and *need-scrutiny* (NS) which is slow and needs search and focus on the patterns. Figure 2.11 shows one of her classic examples where **'X'**s pop up from background **'L'**s (i.e. pre-attentive), finding **'T'**s requires scrutiny. The later indeed needs the serial shift of attention [60, 127].

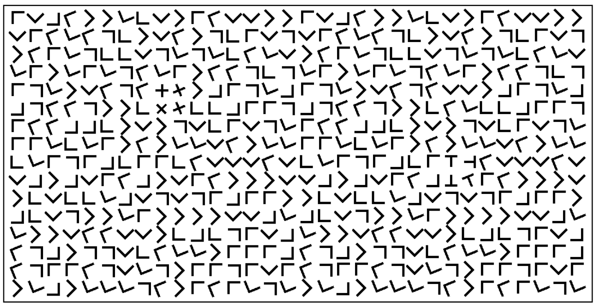
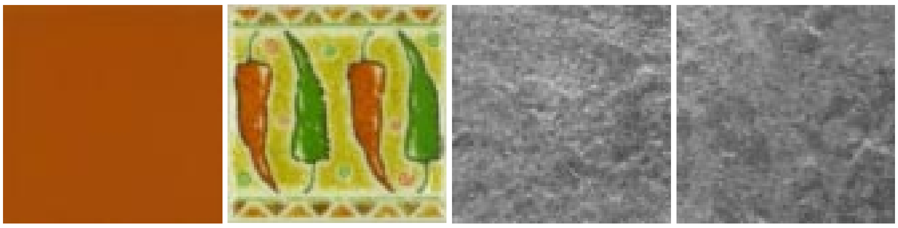


Figure 2.11: An example of pre-attentive and need-scrutiny texture segregation, from [127].

In this example, obviously the 1^{st} order statistics of primitives (i.e. histograms of **'X'**, **'L'** and **'T'**), are similar. Hence higher order statistics must be involved in segregation. Consequently the question to answer would be 'what is the highest N for which it is possible to have identical N^{th} order statistics for a yet distinguishable texture pair?' At first it was claimed that $N = 2$ is the answer. In other words 2^{nd} order statistics could sufficiently describe the differences between textures. However, later Julesz and her colleagues found some stochastic texture pairs with identical 2^{nd} (and even 3^{rd}) order statistics that still were pre-attentively distinguishable [127].

There are some significant differences between normal/abnormal tiles discrimination and stochastic, artificially generated examples of Julesz. Firstly, the threshold between PA and NS is not crisp but fuzzy. Inspection of a textured tile, and many other similar activities, is observer-dependent and environment-dependent. Observer's experience and vision quality or lighting conditions can affect the detection of a particular defect. Secondly, and more importantly, in natural and random textures definition and extraction of texture primitives and the structure is not usually easy.

The major challenge in tile defect detection is in dealing with randomly textured tiles, where the real shape of the tile is not precisely predictable. If the texture is not random, (e.g. Figure 2.12(a) and (b) which are a plain and a figurative tile), we can preserve the model image as a *template* or *reference*. Then we can compensate the intensity and lighting variations (e.g. by normalisation or histogram matching), and viewpoint differences (e.g. by so-called *registration* or matching the corners and edges of model and test images). Consequently, the defect detection can be easily carried out by calculating the difference between the model and test images. The defective areas also will be easily highlighted by this method [46].



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

Figure 2.12: A plain (a) and a figurative (b) tile and two randomly textured tiles, (c) and (d).

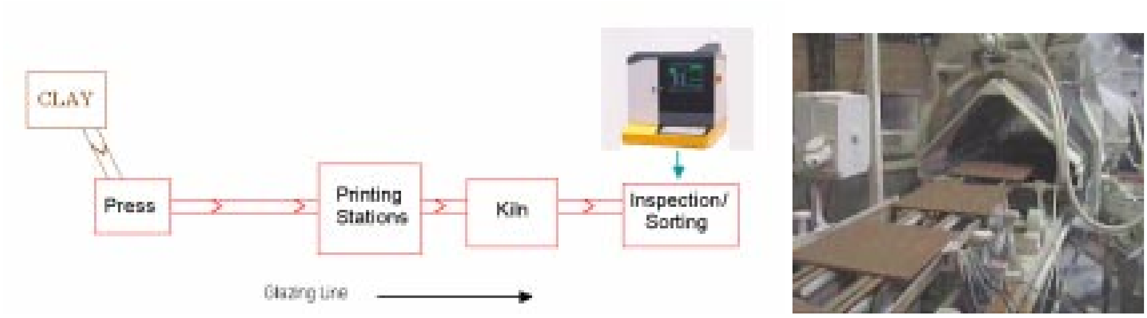
Abnormality detection in a random texture on the other hand, will not be that straightforward. Figure 2.12(c) and (d) are two normal random texture tiles of the same type. It is apparent that they are texturally similar, however no pixel-wise comparison can measure the quality. Instead, we need a texture-wise comparison or measuring the textural properties as the key feature. Rao and Lohse [99] state that humans essentially use three high-level features for texture interpretation: *repetition*, *directionality*, and *complexity*. Repetition and directionality may represent spatial frequency and orientation. Complexity could be related to the consistency of the texture: A strong texture is less complex than a very weak (random) texture. These features can be efficiently measured by signal processing or statistical texture analysis methods to materialise a texture-wise comparison. This fact justifies the wide application of signal processing and statistical methods in random texture classification.

2.6.2 Ceramic Tiles Inspection

As well as many other industries, surface inspection and quality classification is an essential stage in tile manufacturing. Due to the high cost of human inspection, speed of the production line, and repetitious nature of the activity, development of a suitable *automatic defect detection system* (ADDS) would have an impressive impact on the overall performance of a tile production plant. The key point is eventually being at the performance zenith, which guarantees the success in a very competitive market. To be more specific, the advantages of automatic inspection in the tile industry can be listed as:

- Lower inspection costs: In the long term, human inspectors will cost more than an ADDS.
- Less human skill dependent inspection: Precision of the final quality ranking depends on the skill and experience of the inspector. Again, inspection is an iterative and boring task and fatigue can disturb the inspector's performance. An ADDS can provide a more homogenous and consistent inspection process.
- Higher production line speed: The human inspector speed is limited and in many cases the bottle neck of the production line. An ADDS-facilitated production line speed can be increased much further than that limit.
- Workers health risk: The environment of a tile factory is damp and saturated by suspended dirts, and inspectors stay there all the time while the production lines are working. An ADDS decreases the risks of such an unhealthy environment.
- Further development to the other stages of the production line: It is possible to spread a modular ADDS throughout the production line to gain a more effective and robust quality control process. Fault prediction and correction, and recycling the defective materials before the kiln will be some of the advantages of such an advanced system.

Figure 2.13(a) illustrates a schematic of a tile production line and the position of an ordinary ADDS on that. Figure 2.13(b) shows the tiles on a conveyor.



(a) (b)

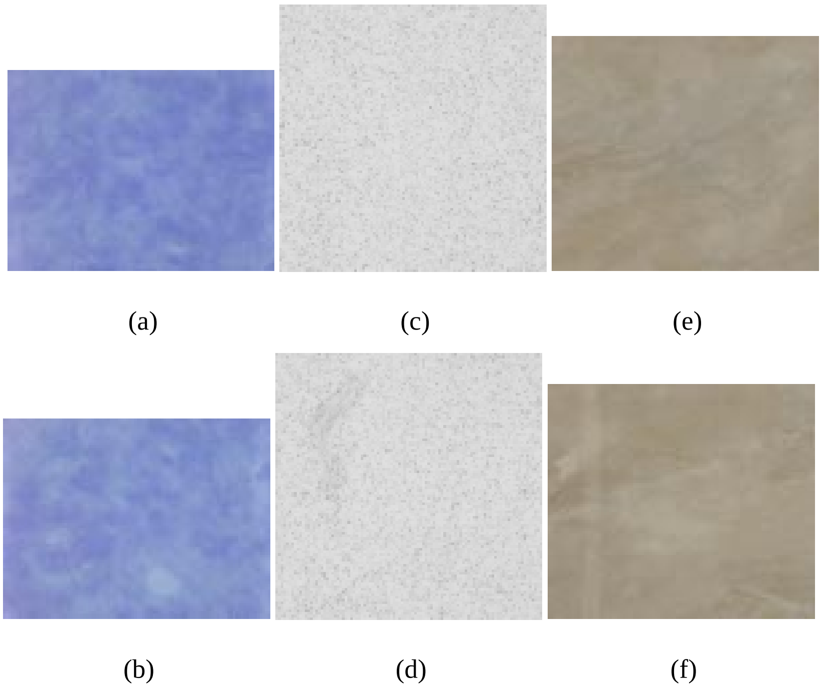
Figure 2.13: A schematic of a tile production line (a) and a picture of tiles on the conveyor belts (b).

Although several manufacturers have introduced their commercial inspection systems (see [74] for some examples) and also there have been massive investigations on that field across academics and research groups (to name a few [5, 13, 14, 28, 79, 84, 111, 121]), it seems that still more efforts and studies are required to achieve high performance, robust and flexible defect detection algorithms and systems.

Defect	**Characteristics**
Broken corners and edges	Physical damages on corners and edges
Colour grading	Changes in colour shades
Cracks	Thin and long random physical defects
Dirt	Small random particles on the surface
Drops	Include colour and water drops
Lines	Wide visible direct lines on tile surface, mostly result of production line bars
Pinholes	Very small holes
Textural problems	Changes in density and shape of patterns

Table 2.1: Typical defects of ceramic tiles.

Table 2.1 illustrates the typical defects for a ceramic tile. Tiles, excluding plain or figurative (or patterned) ones (see Figure 2.12(a) and (b)), are typical examples of random or pseudo-random textures. Any defect changes the expected texture of the tile and hence can be interpreted as a *textural abnormality*. Thus texture analysis is appropriate for normal and abnormal tile discrimination. As an example, Figure 2.14 shows normal and defective tiles of three different types. It can be seen that while normal tiles represent a rather homogeneous texture, defective tiles contain abnormal regions with different textural properties.


(a) (c) (e)

(b) (d) (f)

Figure 2.14: Three pairs of normal/abnormal tiles. Top row: normal, bottom: abnormal.

2.6.3 Previous Studies on Surface Inspection and Tile Defect Detection

Useful reviews on automated visual inspection literature are provided by Chin [22], who covers early works until the beginning of the 80's, and also Bayro-Corrochano [7], and Newman and Jain [91], who cover more recent works until the mid 90's.

Some pioneering work on automatic detection of textural abnormalities was carried out by Kruger *et al* in 1974 [66]. Indeed, in the early 70's it was decided that tens of thousands of chest roentgenogram had to be checked to detect possible coal workers Pneumoconiosis. This massive task was also supposed to be finished in a limited time. That was the main motivation for their work on computer diagnosis of Pneumoconiosis which included texture feature extraction via both co-occurrence matrices (or as they called it 'spatial grey level dependence matrices') and Fourier domain analysis. The co-occurrence matrices were computed on four orientations and four distances, then five pseudo-standard Haralick functions were applied on them. Finally an averaging on different angles was performed to decrease the number of features. Edge information usually concentrates on the higher frequency region of the Fourier domain, also the disease usually affects the image edges. So they employed 32 ring filters to extract Fourier-based textural features and a band-reject wedge filter to diminish the effect of ribs. Then, the normalized energy of rings were selected as the features. The number of grey levels (only 8) and the size of input patches (4×5 inches zonal reproductions of a 14×17 inches roentgenogram) both were relatively low, but justifiable considering the early 70's facilities. A correct decision performance of around 80% was reported. Apart from the limited number of test images, perhaps the most obvious disadvantage of the study might be the assumption of a minor rule for the directionality of the texture compared to the distance (in co-occurrence) and the frequency (in Fourier). In the absence of any reported analogy, it is not clear why directionality has been considered less important than those other two factors.

Hayati *et al* developed a machine vision system for automated surface flaw detection on orbiting space platforms. Their double-staged system firstly compared the grabbed images of the space platform with reference images. Any mismatch triggered the second stage where pattern recognition techniques were exploited to detect and classify the possible flaws. Notable points of their work included a light variability compensation method using a powerful strobe (flash) illumination technique, an iterative matching technique to overcome the misregistration problem between the reference and the inspection image, and a region-based and multi-scale approach to make the flaw recognition process computationally tractable using prior knowledge of image texture [46].

Iivarinen *et al* [52], reported a defect detection system for web surface inspection applicable in distinct processes such as paper, plastic and plywood industries. They applied co-occurrence matrices for feature extraction and a self organizing map (SOM) for segmentation on a windowed web. A notable point of their work is the particular hardware implementation of the system which requires simple feature extraction and segmentation schemes to be used. Hence, for instance only two simple Haralick functions, Mean and Contrast, were applied on the co-occurrence matrices. There is no comment on the utilised directions and distances on that article, however, apparently the devised hardware-oriented co-occurrence computing considers the direction and the speed of the production line to determine a single distance and direction parameters.

Davies *et al* [32] studied on detection of contaminants in food products. They used X-ray images of sealed food bags, which showed strong textural properties, as input, and applied 3×3 Laws filters to extract their textural features. Then a 5×5 scanning window was employed to calculate local energies and build the final texture energy map. A software-based system was also developed for realtime implementation of their foreign object detection scheme.

In [103], Sari-Sarraf and Goddard introduced a vision system for fabric inspection, exploiting multi-scale Wavelet representation to obtain detail images. Bernoulli's rule of combination was then used to recompose the images to highlight the edges and defective regions. Khodaparast and Mostafa [61] repeated a very similar procedure to detect the defective regions in tiles and reported 89% of correct detection. However, due to using rather small number of samples of a single tile model, the generality of results might be questionable.

Similarly, Escofet *et al* in [38] and Kumar and Pang in [68] both implemented the basic idea of wavelet-like Gabor filtering in spatial domain to detect defective textured textiles. Escofet *et al* discussed the algorithm used in details, while Kumar and Pang described the practical implementation more precisely. Decomposition of image into detail images or contrast units (i.e. normalised filter responses) using a Gabor filter bank, applying certain post-processing such as non-linear blob detection and subtraction form a reference feature vector to amplify the 'differences', and composition (fusion) of detail images into a features map were basic stages of these (and in fact many other, e.g. [103]) MSMD schemes. A thresholding operation on the feature map produced the final output where the defects were expected to be highlighted [38].

In [17], Boyd *et al* proposed an inspection method for concrete surfaces using blob-based detection and thresholding in the spatial domain. They employed a perimeter-to-area ratio as a criterion for measuring the blob shape compactness. They suggested that cracks are blobs with high compactness and also utilised a blob-counting scheme to evaluate the consistency of the concrete surface. Conci and Belmiro reported an industrial realtime fabric inspection system [27], focusing on software engineering and implementation aspects of the system. However their categorised-by-defect type and briefly discussed results suggested that amongst three optional defect detection techniques used, thresholding, edge detection and fractal dimension, fractal dimension was the best one. Wang and Asundi studied a computer vision system for wineglass defect inspection [124]. They employed global edge-based Gabor filtering to separate the object of interest (i.e. wineglass) from the background. Then a local Gabor filter was applied to the wineglass image to obtain detail images. Lastly, using 16-grey level co-occurrence matrices, GLCM features of each detail image was extracted and fed to a BPNN classifier to materialise the acceptance/rejection decision.

Valiente *et al* described their tile corner defect detection method in [121]. Their algorithm was based on separation of the tile from the background where the defective corner (i.e. broken corner) was a part of the background. They implemented a histogram subtraction technique which computed the difference between histograms of a pure background and a background/tile patches to threshold between the tile and the background. Difference between expected edges and computed edges at a corner was considered as a defect. After computing this difference in both X and Y directions a simple classifier classified the tiles into three quality ranks from the best to the worst. Outcomes reported showed a good performance on the third and the second quality classes (99.7% and 100% correct classification). Performance for the first quality class however was rather lower and limited to 85.6%. This can be interpreted as a good sensitivity but weaker specificity and perhaps a slight shift in the classifier parameters could balance it in favour of the specificity factor. For further clarification of this subjective classification see [121].

Lopez *et al* studied the registration methods for ceramic tiles [79]. The basic algorithm was edge detection of the test tile, then obtaining the boundary rectangle, followed by a simple geometrical rotation/displacement to map the test image on the reference. They applied two different boundary detection methods: A simpler, faster and more accurate histogram based method and a more complicated, however less accurate Hough transform (HT) based method. The first and the better method used a least square fitting of a straight line on a reduced set of the edge pixels. Unfortunately there was no comment in the paper about possible reasons for the poorer performance of the HT-based method, which on the other hand showed promising results in Costa and Petrou tile registration study [28], where they employed the HT to extract long and straight lines within a tile image. Then a Fourier phase correlation was utilised to register the test and the reference images. Iteratively several displacements $< \Delta x, \Delta y >$ of the test image were tried and the one with the maximum phase correlation with the reference were selected as the registration parameter. In fact, the high similarity between even unregistered test and reference images results in very similar FFT magnitudes, thus only FFT phase was applied as the registration measure.

Smith and Stamp investigated vision techniques for ceramic tile inspection [111]. Their algorithm attempted to analyse complex surfaces which might include 3D topographic features, by separating the topographic and chromatic maps. They reported good feasibility, no need for initial training and being largely pose-invariant as the advantages of their method.

Penaranda *et al* [84] introduced a practical colour machine vision system for ceramic tile inspection. The algorithm contained a simple registration by finding four corners of the test tile using a simple procedure, following a background subtraction. They utilised some special purpose peripherals and a Pentium III CPU to achieve the tile inspection rate of 51 tiles per minute.

In [13], Boukouvalas *et al* used optimal filters to detect abnormal lines and spots in tiles. They also used the Wigner distribution to combine the advantages of both spatial and spatial-frequency domains to detect cracks. In fact, the pseudo Wigner spectrum of each pixel of normal images was calculated and processed to form feature vectors during the training stage. Then in the testing stage, the distance between the feature vectors of train/test images was computed to configure the test image distance map. Unfortunately there was no reported numerical result to evaluate the defect detection precision. Furthermore in [63], the authors presented a method for detecting random texture tile defects consisting of K-means clustering, followed by perceptual merging of clusters in Luv space and morphological analysis. This was computationally expensive, although a promising approach.

Unser and Ade extended their general local transform texture analysis schemes (e.g. [2, 117]) to texture automated inspection in [120]. They proposed an eigenfilter-based feature extraction scheme and a Mahalanobis distance-based decision making procedure.

Chetverikov studied diverse aspects of texture analysis concerning regularity (i.e. approximated period) and defect detection. In [20] he employed a statistical approach together with GLCM to measure the coarseness and regularity of a texture. Also Chetverikov and Hanbury in [21] discussed application of regularity and local orientations as two fundamental structural features in textural defect detection. Their experiments with both Brodatz and TILDA data sets showed promising results.

2.7 Methods

Methods used during this study, namely principal component analysis, two classifiers: artificial neural networks and K-nearest neighbourhood, and novelty detection approach, will be discussed briefly in this section.

2.7.1 PCA, KLT and Eigen-based Decomposition

Principal Component Analysis (PCA) [58] is a popular approach used in pattern recognition studies for reducing problem dimensionality by seeking and eliminating redundant features. It has been applied in a variety of works on texture analysis, for instance [23, 106]. Considering $\vec{x}_i$ a column vector of n features, and M, an $m \times n$ matrix of different observations of $\vec{x}_i$, and finally C_M, the covariance matrix of M, the principal components are the eigenvectors $\vec{e}_j$ of the covariance matrix C_M:

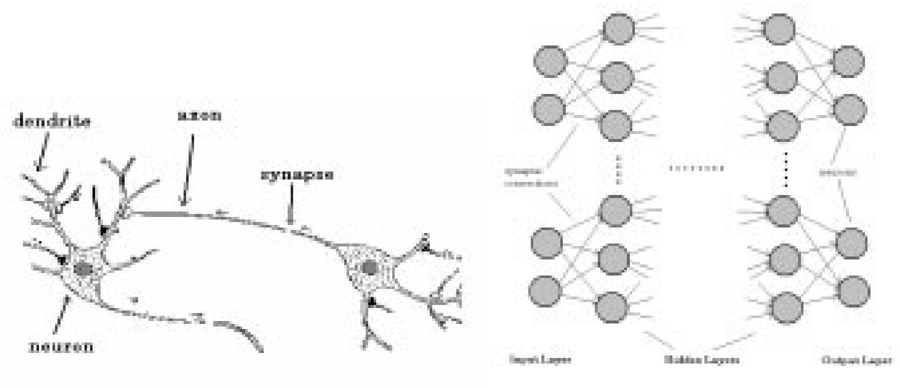
$$C_M \vec{e}_j = \lambda_j \vec{e}_j \tag{2.15}$$

where the eigenvalue, λ_j, is relative to variance of the data across $\vec{e}_j$, hence showing the *importance* of the eigenvectors. In dimensionality reduction tasks, the new p-dimensional feature space is obtained by projection (i.e. element-wise production) of all $\vec{x}_i$ into the first few eigenvectors, sorted on descending eigenvalues.

Extracted from the same hypothesis, Karhunen – Loeve transform, (KLT) and eigenfilter-based methods have been widely used in texture analysis studies [33, 118]. The basic idea underpinning KLT is the employment of eigenfilters as a bank of adaptable filters and obtaining an appropriate set of detail or channel images by projecting the original image onto the bank (i.e. via 2D convolution). The method will be discussed further in Chapters 4 and 5.

2.7.2 Classifiers: Artificial Neural Networks

An artificial neural network (ANN) is a simplified mathematical model of the human brain. The ability of the human brain in learning from experiments, problem solving, and decision making, convinced researchers that an artificial model of the brain, even an extremely simplified one, should be able to perform well in many decision making or computing activities [1, 102]. Figure 2.15(a) depicts a piece of a natural neural system and Figure 2.15(b) shows a multi-layer artificial neural network.



(a) (b)

Figure 2.15: A piece of a natural neural system (a) and a diagram of an ANN (b).

In both natural and artificial NNs, each neuron (i.e. node) accomplishes a very simple task, which is to apply a simple function on its inputs and send the results to its outputs. In an ANN, a node computes the weighted summation of its inputs and compares it with a threshold (or bias) and sends a signal to outputs accordingly:

$$Z = f(\sum_{i=1}^{N} w_i x_i - \beta) \tag{2.16}$$

where Z is the output of the node, x_i is the i_{th} input, w_i the weight (or strength) of the i_{th} connection, and β is the threshold. f can be a simple step function, however *tanh* or *sigmoid* functions (see 2.17) are more commonplace [10, 56].

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.17}$$

A *training* stage is then necessary to optimise the weights. Training indeed conveys the learning-from-example concept of the neural system. A back propagation method is a basic and widely used training algorithm which results in the so-called back propagation neural network (BPNN). This gradient descent method attempts to minimise the difference of the actual and expected output vectors of the network, by thoughtful modification of weights in an iterative procedure [1, 102]. The training algorithm starts with random weights, presents a training sample to the BPNN and calculates the actual outputs and then the error (typically mean square error, MSE) between the expected and actual output vectors. Next, it modifies the weights layer-by-layer from the output toward the input as any modification descends the MSE, and starts the procedure again with these new weights.

During this iterative training procedure MSE ideally should converge to zero. Practically however we may stop the epochs after achieving an adequately small error. Therefore, for training the network we need an adequate number of *labelled* training samples containing all classes, (i.e. samples of all target classes with known expected output vectors). When the training stage is completed, the trained network will be evaluated by presenting a *test* sample set. MSE or the number of correct decision made of the test set show the network performance [10]. Due to their decision making ability, ANNs are appropriate classifiers and in fact have been implemented successfully in many classification applications, to name but a few [23, 39, 67, 89, 92] and [36] where there is a rich list of NNs applications in pattern recognition. There are several NN models each suitable for certain applications. We chose a BPNN with a single hidden layer. The number of nodes in the hidden layer is one of the parameters to be optimised during the tests. It depends on the number of network nodes and training samples and can affect the network properties. For instance, a BPNN with one hidden node is very similar to a linear classifier [10].

When the total number of training/testing samples in the data set is limited, the generality of results obtained by a BPNN classifier could be under question. In such a case, there are some special methods to increase the reliability and generality of the results. Widely-used generalisation methods have different names. Basically however, they are similar. Leave-one-out, Cross-validation, K-folding and Boot-strapping all refer to the same idea: How can we virtually increase the number of training/testing samples?

In a k-fold cross-validation algorithm, a sample set is divided into k subsets of almost equal size. the network will be trained k times, each time leaving out one of the subsets from training and consider it for testing (i.e. $k-1$ subsets for training, 1 subset for testing). If k equals the size of the sample set, this is called *leave-one-out* cross-validation. Cross-validation allows you to use all of the data for training. The disadvantage of cross-validation is that you have to retrain the network k times. Bootstrapping is an improvement on cross-validation that often provides more accurate generalization error approximation at the cost of even more computing time. In the simplest form of bootstrapping, instead of repeatedly analyzing subsets of the data, you repeatedly analyze subsamples of the data. Each subsample is a random sample with replacement from the full sample [65, 104].

2.7.3 Classifiers: K-Nearest Neighbourhood Classifier

A k-nearest-neighbourhood classifier (KNN) is a simple but efficient distance-based classifier. Basically a KNN classifier has a labelled training set which contains examples of all possible n classes of data. When the KNN is presented by a test sample, x, it looks up and finds the k nearest training samples to x. k typically is a small integer and should not be a divisor of n (i.e. $mod(n,k) \neq 0$). Then x will belong to the class which has the most samples amongst the k nearest neighbours [57]. Figure 2.16 shows a 2-feature feature space $F1F2$, and $n = 3$ classes of patterns: $\{black,\ red,\ green\}$. With $k = 7$, the test sample x belongs to *black* class, because there are more *black* samples amongst its 7 nearest neighbours than *red* or *green* (4 vs. 2 vs. 1).

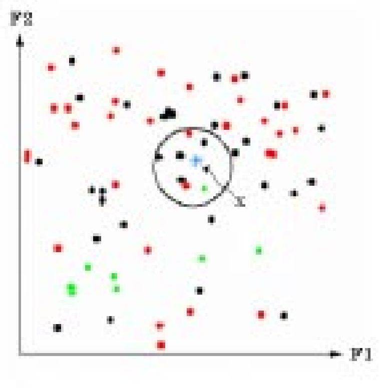


Figure 2.16: KNN classification, k=7 and n=3 classes

To be more specific, distance measurement is mostly carried out in the feature space where the feature vectors of training samples are preserved. Different distance measurements can be employed, however an Euclidian distance on normalised feature vectors usually performs well. For instance normalisation by

$$v_i = \frac{v_i}{max(\mid v_i \mid)} \tag{2.18}$$

where v_i is a vector contains the i_{th} feature of all samples, will bring all the features into $[-1,\ 1]$ range. It gives all the features the same weight in distance measurement. A subtly weighted feature vector however, may increase the classification performance but this also decisively increases the complexity of the classifier design. A simple KNN only should be optimised upon 1 parameter: k, whereas a weighted KNN should be optimised upon number of features $+1$ parameters [57], which may not be practically worthwhile in many cases. A KNN classifier is not statistical in nature, however it can be proven that if the number of training samples moves toward infinity, it will converge to the optimal Bayes classifier [41]. It can also be considered as a *lazy algorithm* which does not need heavy computing. Even its training, in contrary of ANNs, is not real training, but only saving the training samples for further distance measurement.

2.7.4 Novelty Detection

Novelty detection is a different approach to the traditional classification problem. The typical classification approach relies on samples of all classes of patterns due to be classified. In the particular case of abnormality detection, there are two classes of patterns: *normal* and *abnormal*. Although typical classifiers can show a promising accuracy in theory or limited practice, in many applications it is often difficult to acquire an adequate number of defective samples and build up the abnormal data set which also represents the whole *abnormalities space*. While it is much easier to develop an efficient normal data set.

Novelty detection or *concept learning* approach on the other hand, only needs the normal class to be defined. A novelty detector learns the model of normality in the training stage using only normal samples and abnormalities are then identified by testing for novelty against that model. In a geometrical sense, a defective sample shows a considerable distance to the cluster of normal samples [55, 112].

It is common in novelty detection studies to apply auto-associative neural networks (AANN) or self organising maps (SOM). An AANN attempts to recreate the output of the network the same as its input. There is a number of different ways for designing an AANN. The simplest method is to use Principal Component Analysis or a special purpose neural network with a few hidden nodes and equal input/output nodes [112]. For example, Worden [128] utilises an AANN for cracked beam defect detection and Sohn *et al* applies it for hard disk quality inspection [112].

A SOM is an alternative to statistical clustering and is an unsupervised approach. Therefore, it does not need *a priori* information on classes. In most SOM based approaches, similar to statistical clustering, some form of cluster membership value is thresholded to determine whether a sample belongs to a cluster or not. In [52] Iivarinen *et al* implement a SOM for web surface inspection. Also Tolba and Abu-Rezeq utilise a SOM for textile visual inspection

[115]. Statistical methods however, are also popular. In [83], Markou and Singh review both statistical and neural network based novelty detectors. Our work respecting novelty detection however, does not employ any complex classifier. Instead, it maps the texture features into a 1D classification space and utilises a simple single-parameter thresholding scheme. This new texture defect detection algorithm will be presented in Chapter 5.

Chapter 3

Texture Analysis of High Definition Outdoor Scene Images

3.1 Introduction

Outdoor scene object classification using texture analysis is a prime example of a computationally expensive process where there is usually a need for trade-off between speed and accuracy. Examples of such bartering are sometimes inherent in the nature of past works on texture analysis, or less often the subject of explicit analysis [89, 95, 100, 110, 119]. This problem is exacerbated as the size of the image under analysis increases, involving more and more computations. For instance, we tried a frequency space analysis of very high resolution outdoor scene images (4032×2688 pixels) aimed only at increasing the accuracy of texture segmentation. The trade-off issue and the penalties expended by the computational costs were not considered important, and it was found that a 6% increase in accuracy could be achieved, albeit at some considerable computational expense [89]. Here, we show that by using novel faster approaches, similar levels of accuracy can be maintained.

We study computationally lower cost (i.e. faster and easier to implement), precise algorithms for image classification. Past works in grey level texture analysis have increasingly found success in the classification of texture features derived from Gabor filters [25, 40, 54, 100, 110] or wavelets [78, 95, 119]. Gaining their multi-scale and multi-directional properties, the considerable discriminatory power of such features have recently been further strengthened through the use of chromatic features based on such colour spaces as RGB, *Lab* , or HLS [34, 87]. However, as mentioned previously, these methods are computationally high cost.

Our fundamental idea is selecting and analysing a few well-known, high performance feature extractors in each area of texture and colour data, to find out their most important strengths. Consequently a few correspondence lower cost algorithms which attempt to retain those strengths are proposed and compared to the original higher cost methods. Here, we examine the performance of Gabor and *Lab* features as a highly accurate set for colour texture classification against a sometimes slightly less accurate, but much faster set of novel features. These new proposed features are a combination of directed textural features, extracted using the application of Walsh-Hadamard transforms (WHT) to oriented images (called *directional Walsh-Hadamard Transform* or DWHT), as well as chromatic features that correspond to, but more easily computed than, hue and saturation in the HLS or HSI spaces. Gabor features allow us to conveniently capture the low to high frequencies present in the Fourier space of high resolution images in different directions. In the same way, we are proposing a method of capturing these frequencies through the rotation of the image by varying angles before applying the WHT.

WHT is a typical tool in image processing. For instance, Unser [118] used Hadamard matrices along with other local transforms such as DCT and KLT in texture measurement. He applied different small size filters and a filter sliding scheme in the spatial domain to evaluate the effectiveness of these filters in texture analysis. Kim and Cho [62] have also implemented

Walsh functions in a texture segmentation task using 16 different 4 × 4 Walsh matrices as the textural feature extractors. In their recent book, Yarlagadda and Hershey have introduced some new applications of Hadamard matrices, such as error correction coding, signaling (based on a new Hadamard basis) and a signal representation scheme which is capable of representing odd frequency component made signals [50, 129]. To find more examples see Section 2.3.2.

Figure 3.1 illustrates a procedural overview of the experiments to be reported in this chapter. The experiments procedure contains textural and chromatic feature extraction schemes, their appropriate classification performance tests, and the final test with merged textural/chromatic features.

We describe our data set and framework in Section 3.2. In Section 3.3, the Gabor and the DWHT feature sets are described as textural feature extraction schemes. The new chromatic features will be discussed next at Section 3.4. Experimental results using merged texture/colour feature sets on outdoor scenes are presented in Section 3.5. Execution times of different schemes are compared in Section 3.6. Next, in Section 3.7 new tests using pseudo-standard texture suite, VisTex, will be presented. The chapter then is concluded in Section 3.8.

3.2 High Resolution Outdoor Scene Data Set

We have a data set consisting of 724 colour image patches of 128 × 128 pixels extracted from more than a hundred high resolution 4032 × 2688 images of outdoor scenes. These scenes were photographed by a high attribute optical camera using high definition low speed (25 ASA) films and a normal 50mm lens under bright clear (sunny) daylight conditions. Then a high resolution negative scanner was used to digitise the 36 × 24 mm negatives into 4032 × 2688 pixel, 24 bit RGB images. The pixel definition is 112 pixel/mm or 2845 pixel/inch. Figure 3.2 illustrates six images of that set.

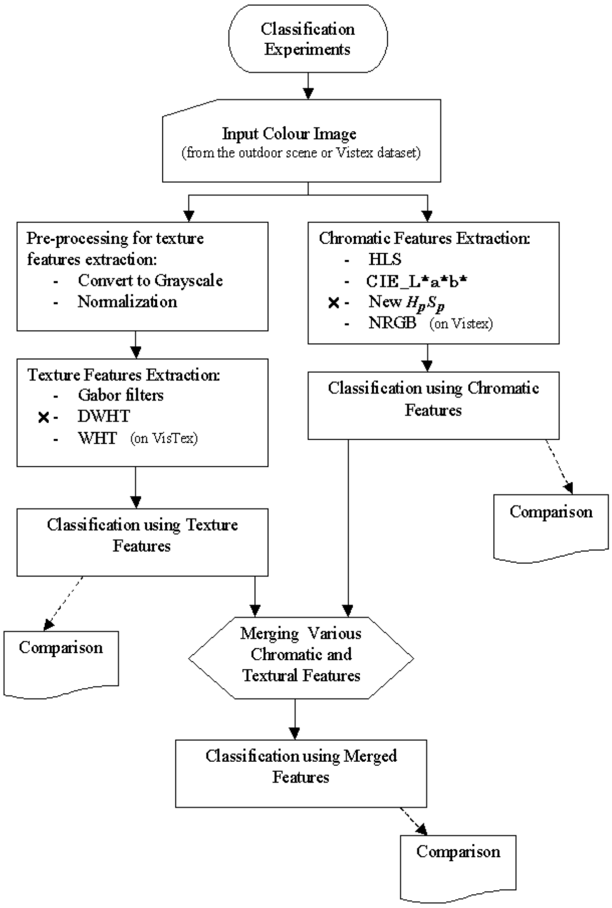


Figure 3.1: An overview of Chapter 3 experiments.

In recent years, the C.V. research group at Bristol University has developed a neural network based system for classifying images of typical outdoor scenes to an area accuracy of approximately 90% [39]. Texture information is represented in this system using Gabor filters. A common problem is that many regions in typical outdoor scenes are too small to allow a significant range of spatial frequencies to be included in the feature set. In [89], we presented a pilot study designed to establish if high resolution images would provide a sufficient increase in texture information to justify the extra computational complexity. We found that a 6% increase in accuracy could be achieved at some considerable computational expense. Here, we show that by using a faster approach, i.e. through our proposed oriented DWHT and chromatic features,

Figure 3.2: Six high resolution outdoor scene images

similar levels of accuracy can be achieved.

The patches come from four categories: `CAR`, `PAVEMENT`, `ROAD` and `TREE`. Figure 3.3 shows some typical examples of our input patches. In brief, the goal of this classification experiments is to classify an unknown test patch into one of those four possible groups. These patches of high resolution images contain shiny, fairly smooth bodies of cars (but including wheels, door-handles, lights etc., rough and coarse surfaces of pavements, fine resolution granularity of road surfaces, and fine and coarse structures within trees and bushes. These provide a wide range of characteristics and frequencies in the data set. Many such frequencies are diminished or lost in lower resolution images (see Figure 1.2-(b)).

3.3 Textural Feature Extractors

In this section, two different feature extraction schemes exploited in this work are discussed. The schemes, namely Gabor filtering and directional Walsh-Hadamard transform, are implemented in a practical image classification task and compared.

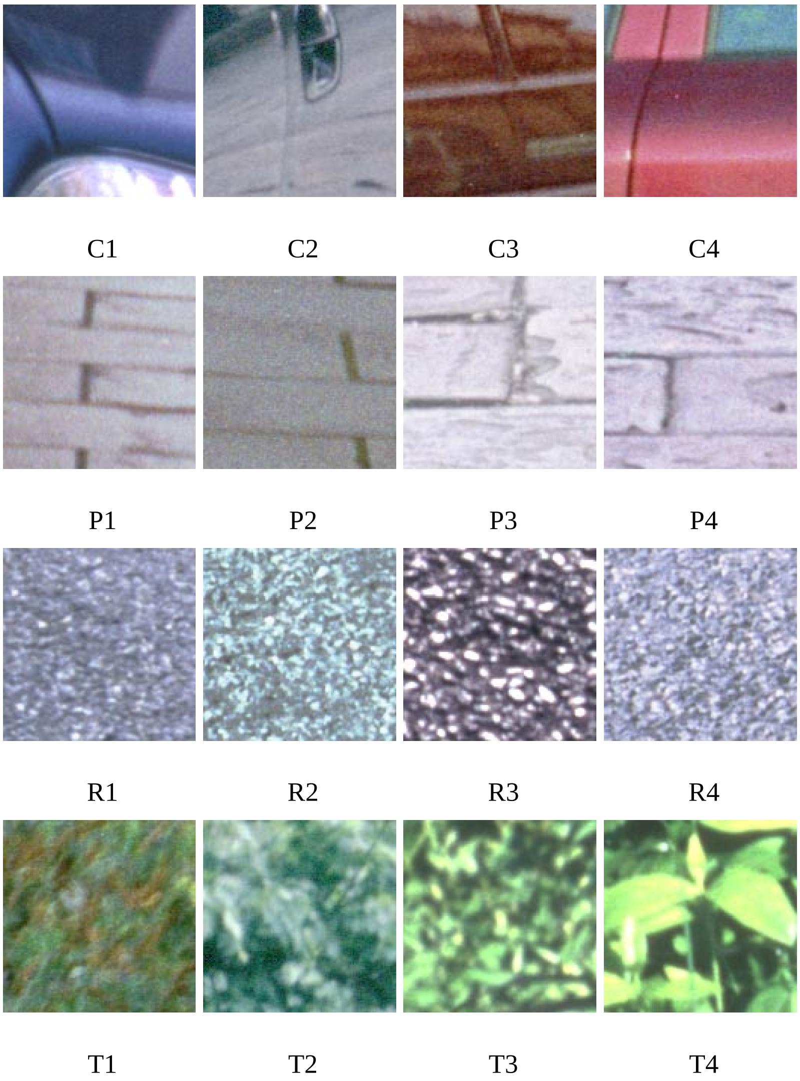


Figure 3.3: Sixteen sample images of four classes, from the top, `CAR`(C1...C4), `PAVEMENT`(P1...P4), `ROAD`(R1...R4), and `TREE`(T1...T4).

3.3.1 Gabor Filters

Gabor filters are widely used for multi-scale/multi-directional analysis in image processing. Specifically, they have shown high performance as feature extractors for texture discrimination and unsupervised texture classification [40, 54, 100, 110, 90, 34]. The important strength of Gabor filters in texture analysis is that they facilitate oriented or directional band pass filtering of the input texture. This allows the filter to extract notable textural features which are directional and, in a frequency sense, band-limited [53, 54, 118, 126]. A basic practical disadvantage in Gabor filtering is their high computational costs. Basically it is possible to define and employ Gabor filters in both spatial and spatial frequency domain, however frequency domain implementation would be faster, particularly for larger images. The Gabor filter in the spatial frequency domain is: (This presentation may seem different from Equation 2.11, however basically they are similar)

$$G(u,v) = e^{-\pi(\frac{u_p^2}{\sigma_x^2}+\frac{v_p^2}{\sigma_y^2})} . e^{-2\pi j(x_0 u + y_0 v)} \tag{3.1}$$

where,

$u_p = (u-\omega_x)*cos(\theta)+(v-\omega_y)*sin(\theta)$ and $v_p = -(u-\omega_x)*sin(\theta)+(v-\omega_y)*cos(\theta)$

are the rotated/displaced coordinates in the frequency plan, ω_x and ω_y are filter central frequencies (i.e. modulation factors) in x and y directions, θ is filter orientation parameter, σ_x and σ_y are filter standard deviations (i.e. band width) in x and y directions, and x_0 and y_0 are horizontal and vertical displacements in the spatial domain. We keep $x_0 = 0$, $y_0 = 0$, and set $\omega_x = \omega_y$, and $\sigma_x = \sigma_y$ in all experiments. Hence the applied Gabor filter is:

$$G(u_c,\sigma_c) = e^{-2\pi(\frac{u_c^2}{\sigma_c^2})} \tag{3.2}$$

In texture analysis applications, usually a set of Gabor filters (called a *Filter Bank*), which contains a few filters with different central frequencies and orientations is employed (see Section 2.3.2). Diverse filtering frequency and orientation of a filter bank will allow it to extract considerable amounts of texture information. However a question to answer is how the optimum filter bank can be configured. Unfortunately, since it seems that there is no general analytical method of optimisation for this, a lengthy procedure of test and evaluation was necessary to determine the best possible filter bank configuration. For example, in current tests we mostly used a Gabor filter bank containing 12 filters arranged in 3 frequency bands and 4 orientations:

$$\text{Gabor Filters}:\ G(\omega_i,\theta_j)\ ,\ \ \omega_i=\{\frac{\Omega_M}{8},\frac{\Omega_M}{4},\frac{\Omega_M}{2}\},\ \ \theta_j=\{0^\circ,45^\circ,90^\circ,135^\circ\} \tag{3.3}$$

Where Ω_M is the maximum possible spatial frequency (i.e. half of the dimension in the FFT domain). Figure 3.3.1 shows that filter bank in the frequency domain. Parameters were declared in keeping with current theories on biological vision system as the frequency bandwidth of 1 octave and half-power bandwidth [23]. Selected central spatial frequencies ($\omega=\frac{\Omega_M}{2},\frac{\Omega_M}{4},\frac{\Omega_M}{8}$), cover the lower 50% of frequency domain with a log_2 scale. In fact in all Gabor filter tests we considered one octave (dyadic) central frequency rate and the half power bandwidth (see Section 2.3.2). Filter responses then will be calculated as:

$$\text{Filter Responses}:\ G(\omega,\theta)=\ |\,\mathrm{F}^{-1}(\mathrm{F}(A)\cdot G(\omega,\theta))\,| \tag{3.4}$$

where $G(\cdot)$ is the filter response, F and F^{-1} are Fourier and inverse Fourier transforms, A is the image and $G(\cdot)$ is the Gabor filter in the frequency domain. Figures 3.5 and 3.6 illustrate the filter responses of two outdoor scene samples, a `CAR`, and a `PAVEMENT`. Apparent diversities amongst multi-band and directional responses suggest that a feature vector based on a few statistics of filter responses (e.g. mean, standard deviation and so on) may adequately segregate the input samples.

Clausi and Jernigan believe that an improper setting could seriously damage the performance of the Gabor functions [25]. Apart from accepted log_2 bandwidth scale throughout the spatial frequency domain, there are several options for central frequencies and directional definition. In our experiments, the above mentioned central frequencies usually provided the maximum accuracy. Starting at $\frac{\omega_M}{2}$ as the highest central frequency, seems to keep the balance between more important lower spatial frequencies and useful higher ones [89, 90]. In fact, Clausi and Jernigan state that rather poor performance of Gabor filters in [114] may be partially due to the exclusion of higher frequency bands [25].

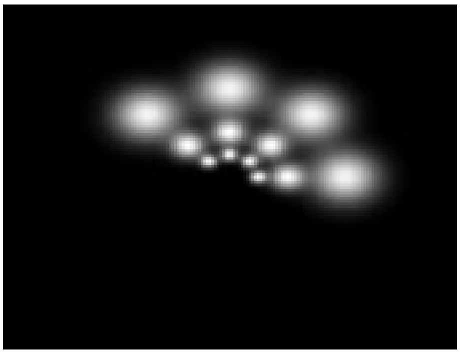


Figure 3.4: Applied Gabor filter bank, 4 lower pass ($\omega = \frac{\Omega_M}{2}$, inner), 4 band pass ($\omega = \frac{\Omega_M}{4}$, middle), and 4 higher pass ($\omega = \frac{\Omega_M}{8}$, outer). Orientations are $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$.

3.3.2 The New Approach: Directional Walsh-Hadamard Transform

In this section, we introduce and describe our novel Hadamard-based texture analyser, directional Walsh-Hadamard transform. Amongst the family of orthogonal linear transforms of time/spatial domain signals, which mostly employ sinusoidal-based kernel functions (e.g. Fourier or Cosine transforms), the Walsh transform is defined as:

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I(x,y)[(-1)^{\psi(u,v,x,y)}] \tag{3.5}$$

where I is the image, N is the image size, and ψ determines the transform's parametric kernel function $-1^{\psi(\cdot)}$. The Walsh transform is one of the exceptions in so-called 'harmonic analysers' which implies sequency-based kernel functions and decomposes the input signal into rectangular wave primitives in the transform domain [8]. The kernel function can be selected from a diverse set of possibilities. For instance, in the Hadamard natural transform (a member of the Walsh family) ψ is:

$$\psi(u,v,x,y) = \sum_{i=0}^{m-1} [b_i(x)b_i(u) + b_i(y)b_i(v)] \tag{3.6}$$

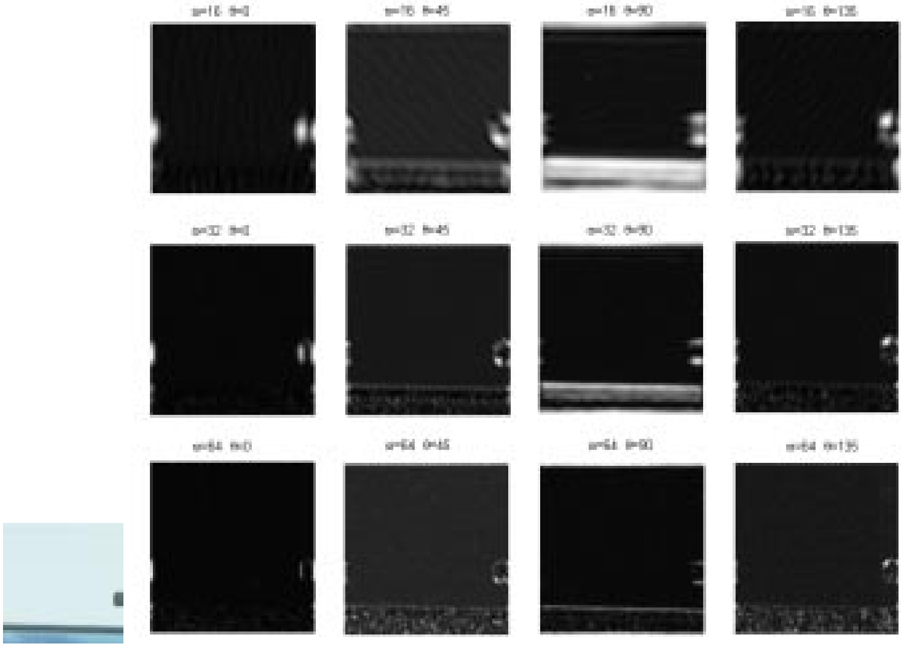


Figure 3.5: Gabor filter responses of a CAR. Left: Input image, Right: filter responses (detail images) G

where $b_i(z)$ is the i^{th} bit of z in binary representation. The Walsh-Hadamard form of *digital frequency* or *sequency*, which is the number of zero-crossings or sign-changing of the signal, is analogous to the frequency in the Fourier transform. However, sequency is twice the size of the frequency of a signal, i.e. $Seq(x) = 2 \times Freq(x)$, where x is a signal in time or spatial domain. As shown in Figure 3.7, the orthogonal set of rectangular waveforms that generate the WHT kernel can only approximate a sinusoidal waveform (c), by weighted summation of their square wave elements (d and e). Hence, we expect WHT features to be a weaker representation of the texture in comparison to those of the FFT or Gabor. For instance, results reported by Unser [118] and Ade *et al* [3], demonstrate the relatively lower performance of WHT in texture classification. Nevertheless, in both cases spatial domain filtering using very small size filters (e.g. 3×3 to 5×5) were applied. Therefore, smoother structure of a, for example, Cosine filter outperforms sharp 0-1 structure of a Hadamard filter (see Figure 3.7 (a) and (b)). This is the typical problem of representation of a signal by rectangular basis functions. When we

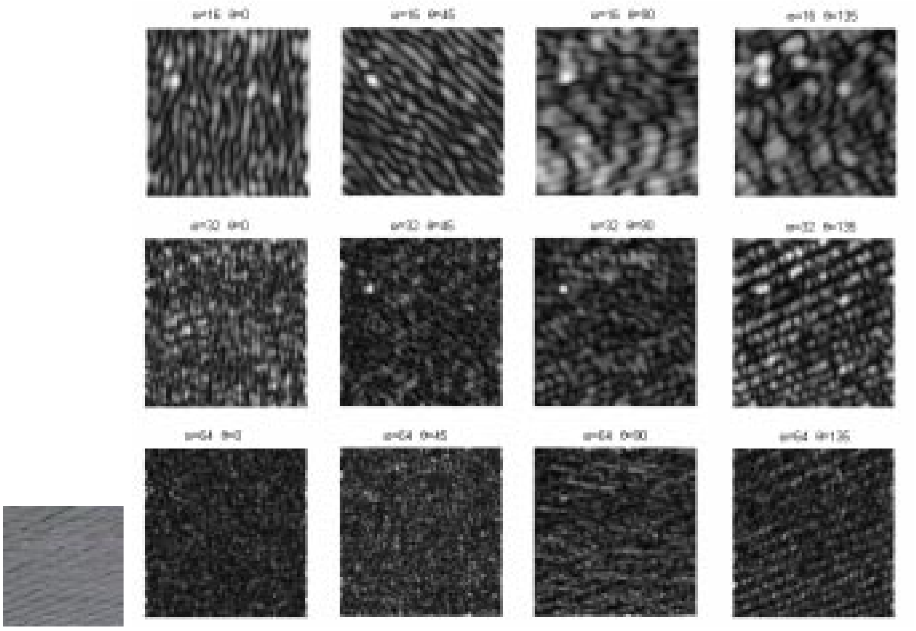
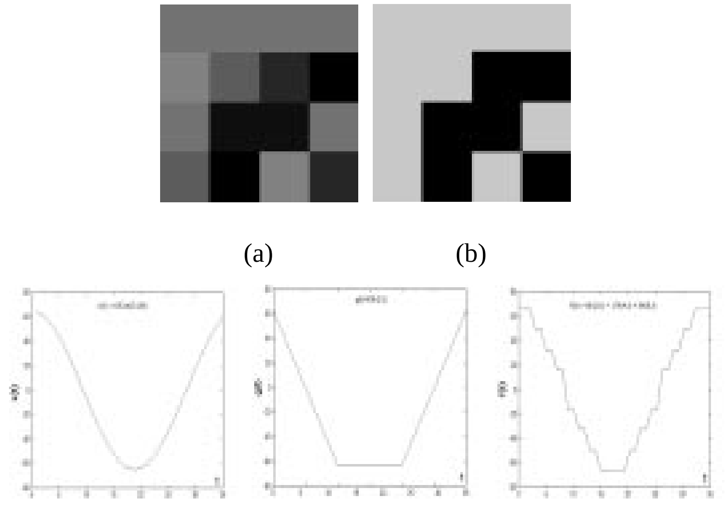


Figure 3.6: Gabor filter responses of a `PAVEMENT`. Left: Input image, Right: filter responses (detail images) G

apply a Hadamard transform and utilise higher sequency ranks however, since higher sequency functions can smooth and moderate the outputs, the representation error degrades. Again in Figure 3.7, in synthesising a sinusoidal signal (a), involvement of higher ranks (here: 8) Walsh functions, decrement the representation error (compare (d) and (e)).

Furthermore, the WHT has important computational advantages. For instance, it is a real (not complex) transform, it only needs addition and subtraction operations, and if the input signal is a set of integer-valued data (as in the case of digital images), we need only use integer operations. Furthermore, there is a fast algorithm for Walsh transforms by simple substitution of the exponential kernel of the Fast Fourier Transform with the $-1^{\Psi(\cdot)}$ kernel of Walsh. The transform matrix, usually referred to as *Hadamard matrix*, can also be saved in binary format resulting in a decrease in memory requirements [42]. Also it is rather easier than other transforms to implement WHT in hardware [35].



(a) (b)

(c) (d) (e)

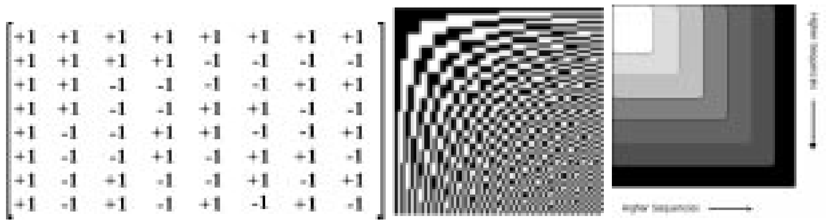
Figure 3.7: 4×4 Cosine (a) and Hadamard (b) filters, and synthesising a given sine wave (c) using Hadamard functions. The Original signal $x(t) = 63cos(0.16t)$ (c), and its two Walsh-Hadamard approximated representations: $g(t) = 63h(2,t)$ (d), and $f(t) = 41h(2,t) + 17h(4,t) + 8h(8,t)$ (d).

Amongst different Walsh transforms (e.g. Dyadic, Natural, Ordered and so on) we applied a *sequency-ordered Hadamard* (SOH) matrix [42, 8] where the rows (and columns) are ordered according to their sequency. In other words, in the first row there are no sign changes, and in the n^{th} row there are *n-1*. As an example, see Figure 3.8 for a rank=3 (or 8×8) SOH matrix (a) and a map of rank=6 (or 64×64) one (b).

The 2D Walsh-Hadamard transform can be defined as:

$$WHT_{2D}(A) = H \times A \times H' \tag{3.7}$$

where A is the image and H and H' are Hadamard matrices of the same size and its transpose. In fact for a Hadamard matrix, H is always equal to H'. The result $WHT(\cdot)$ would be the unique and reversible representation of the input in the sequency domain. Then, as Figure 3.8(c) shows, different row-column bands of $WHT_{2D}(\cdot)$ contain the information of correspondence low/high sequency bands of the input image.



$$\begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix}$$

(a) (b) (c)

Figure 3.8: (a) Sequency-ordered rank=3 (8×8) Hadamard matrix. (b) A map of rank=6 (64×64) SOH. (c) Sequency bands of SOH in a transform domain.

Due to its faster speed and more feasibility in real time applications, we aimed to use the Hadamard transform as a textural feature extractor, the task that Gabor filters usually carry out formidably but in higher computation costs. Inspired from oriented/multi-band structures of Gabor filters, it seems to be useful to develop an *oriented* and *multi-band* Hadamard based feature extraction algorithm which would be capable of extracting textural features in different directions and sequency scales. Traditional Hadamard transforms, (like other 2-dimensional transforms, e.g. Fourier or Cosine) can only extract the vertical and horizontal information. In fact, the normal 2D transform in (3.7) can be implemented as:

$$WHT_{2D}(A) = WHT_{1D}([WHT_{1D}(A)]') \tag{3.8}$$

where $[WHT_{1D}(A)]'$ is the transposed transform matrix. This results in applying a normal 1D transform on rows of the input matrix, transposing the result and again applying the 1D transform on rows. Hence a standard Hadamard transform provides the sequency information of rows and columns of the image. To generate an *oriented WHT*, a direct solution can be rotating the input image before transformation. However, this solution could not be completely practical. For example, whereas any rotation will change the square alignment of the input, eventually we should reshape that to a square matrix before transformation. Hence, we propose a novel oriented Hadamard based features to represent the directionality of texture. In this scheme, the Hadamard matrix remains constant but the image function is *rotated* by $\alpha = \{0°, 45°, 90°, 135°\}$. The rotation is applied to each element in the top row of the image matrix. At border pixels corresponding elements are used from a repeated imaginary version of the same image matrix (i.e. image is vertically and horizontally wrapped around). For example, in the simple 4×4 image matrix below, a 45° rotation at position b gives $\{b, g, l, m\}$ (also see A_{45° in Equation (3.10)) and a 135° rotation at the same position b gives $\{b, e, l, o\}$ (similarly see A_{135° in (3.10):

$$\begin{bmatrix} a & \mathbf{b} & c & d \\ e & f & \mathbf{g} & h \\ i & j & k & \underline{\mathbf{l}} \\ m & n & \underline{o} & p \end{bmatrix} \begin{bmatrix} a & \underline{b} & c & d \\ \underline{e} & f & g & h \\ i & j & k & l \\ \mathbf{m} & n & o & p \end{bmatrix} \tag{3.9}$$

The full rotation set where $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ can be defined for a simple 4×4 image matrix as follows:

$$A_{0^\circ} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \quad A_{45^\circ} = \begin{bmatrix} a & f & k & p \\ \mathbf{b} & \mathbf{g} & \mathbf{l} & \mathbf{m} \\ c & h & i & n \\ d & e & j & o \end{bmatrix}$$

$$A_{90^\circ} = \begin{bmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \\ d & h & l & p \end{bmatrix} \quad A_{135^\circ} = \begin{bmatrix} a & h & k & n \\ \underline{b} & \underline{e} & \underline{l} & \underline{o} \\ c & f & i & p \\ d & g & j & m \end{bmatrix} \tag{3.10}$$

Note that this is not an ordinary geometrical rotation. For example, we create the rows of A_{45° image by considering the pixels that sit in a 45° direction in image A_{0° and so on. This means that the resulting horizontal rows capture the information at the specified angles. In fact it looks more like a pixel rearrangement rather than a geometrical rotation. Of course we extend this concept to vary larger images in this work (e.g. 64×64 or 128×128). The artifacts of repetition (disruption) at the borders are much less significant with larger images than those in the examples above.

Although we have never used other rotations in this study, it is theoretically possible to extend this rotation idea to any desired degree. For instance, (3.11) shows a 22.5° rotated version of A:

$$A_{0^\circ} = \begin{bmatrix} \mathbf{a} & b & c & d \\ e & f & \mathbf{g} & h \\ \mathbf{i} & j & k & l \\ m & n & \mathbf{o} & p \end{bmatrix} \rightarrow A_{22.5^\circ} = \begin{bmatrix} \mathbf{a} & \mathbf{g} & \mathbf{i} & \mathbf{o} \\ b & h & j & p \\ c & e & k & m \\ d & f & l & n \end{bmatrix} \tag{3.11}$$

Other rotation schemes were also tried. Experiments with geometrically rotated images, rotated Hadamard matrices (H_α), or using a wedge of transform domain to explicit directional information in WHT domain (as Figure 3.9 shows), all failed to reach adequate performance levels.

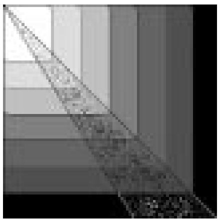


Figure 3.9: A Wedge of WHT matrix. Wedges did not clearly convey the texture's corresponding directional properties.

The manner of our rotations means that after the DWHT transformation we need only extract *row sequency* information, corresponding to the directions used. This reduces (3.8) to a more simplified transform:

$$DWHT_\alpha(A) = A_\alpha \times H' \tag{3.12}$$

We can also ignore, without loss, the column sequency information. Indeed, this takes out the redundancy created by the fact that $A_{0^\circ} = A'_{90^\circ}$ and $H = H'$. We performed a PCA test on the

DWHT feature sets and found no particular preference for any orientation $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ in the feature space. Since we have textures with random directions in the outdoor scene images, it is reasonable to believe that there is no considerable directional preference amongst the set.

$$DWHT_{0^\circ}(A) = A_{0^\circ} \times H' = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} =$$

$$\begin{bmatrix} a+b+c+d & a+b-c-d & a-b-c+d & a-b+c-d \\ e+f+g+h & e+f-g-h & e-f-g+h & e-f+g-h \\ i+j+k+l & i+j-k-l & i-j-k+l & i-j+k-l \\ m+n+o+p & m+n-o-p & m-n-o+p & m-n+o-p \end{bmatrix} \tag{3.13}$$

As (3.13) shows, the operation $DWHT_\alpha(A) = A_\alpha \times H'$ computes and gathers the sequency information of input matrix rows into transformed matrix columns. Hence, the same half transform for a rotated matrix (e.g. A_{45°) will give us the sequency information of pixels with a 45^o orientation, again into the columns of transformed matrix. The transformed matrix columns from left to right correspond to the lower to higher sequency elements. In the Hadamard-based feature extraction procedure, we exploited the above mentioned rotation and transformation for four different orientations, $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$:

$$\begin{cases} DWHT_{0^\circ}(A) & = & A_{0^\circ} \times H' \\ DWHT_{45^\circ}(A) & = & A_{45^\circ} \times H' \\ DWHT_{90^\circ}(A) & = & A_{90^\circ} \times H' \\ DWHT_{135^\circ}(A) & = & A_{0^\circ} \times H' \end{cases} \tag{3.14}$$

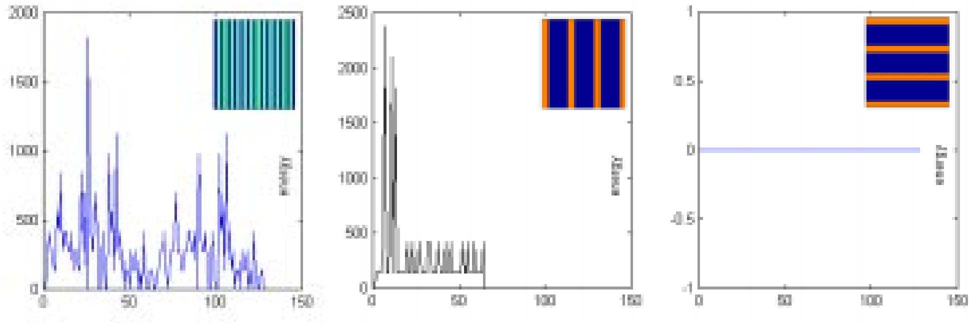
Since the relative arrangement of pixels is of essence in texture analysis [127, 113], sequency based features which represent the number of zero-crossings of pixels in a particular direction can convey a notable amount of textural information. We can measure the DWHT energy in $DWHT_{\alpha}(A)$ as the absolute value of the DWHT output along each column. Columns can be divided into a few groups which represent different *sequency bands*, then a few statistics of each band can be extracted to configure a feature vector with reasonable dimensionality. So, a DWHT output and feature vector can be defined as:

$$H(\alpha, b) = DWHT_{\alpha}(A)|_{i,j}\ , 1 \leq i \leq N\ ,\ j \in b\ ,\ \ and\ \ ,\ F_{DWHT} = M(H(\alpha, b)) \qquad (3.15)$$

where H is the transform's output matrix, N is the matrix size, F is the feature vector, M indicates the applied statistical function, and b is the desired sequency band. Again log_2 or semi-log_2 bandwidth scales could be applied, however we mostly used a simpler $\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}$ division (from the lowest to the highest sequence) for 3-band and a $\frac{1}{4}$ division for 4-band feature sets.

As an example, Figure 3.10 depicts, the sequency representation of a typical fine resolution texture (a) will show more energy in higher sequency bands compared to a coarse resolution texture (b), which indicates the method's spatial frequency sensitivity. Again, the rightmost graph (c) in that figure, illustrates the lack of response of the DWHT transform to the coarse vertical texture when it is rotated (i.e. $A_{90^{\circ}}$), which indicates the method's directional sensitivity.

One main advantage of the proposed Hadamard based feature extraction scheme is that by using larger (e.g. 64×64 or 128×128) Hadamard transforms, rather than the usual 3×3 or 4×4 local filters [118, 62], we can extract higher frequency/sequency information and in fact emphasise on 'global' instead of 'local' features.



(a) (b) (c)

Figure 3.10: From left: Example average energies for fine resolution texture (a), coarse resolution texture (b), and coarse resolution texture at 90° rotation (c). Corresponding textures are shown inside each graph. Energies are computed as the absolute value of the WHT output along each column.

3.3.3 Justification of the DWHT

To justify the proposed DWHT transform, a quick review of some related works could be useful. Boukouvalas *et al* [13] have applied an optimal line filter to detect lines in ceramic tiles which in fact performs two 1D convolutions in the horizontal and vertical directions respectively. Their achievement suggests that although textures are 2D signals, sometimes 1D processes would be significantly effective. In particular, where directional objects are concerned, a 1D process in the appropriate direction would be advisable.

Proposing DWHT, we aim to add MSMD characteristics to an ordinary Hadamard transform. The basic idea behind all MSMD texture analysis methods is scale (i.e. band) and angular (i.e. directional) decomposition of the texture. For instance, in ring/wedge filtering, the intersection of a ring (directional) and a wedge (band-pass) filter reveals the energy of a particular band and orientation of the texture. (There is always a 90° difference between the orientation of the filter and revealed patterns. e.g. a vertical wedge filter highlights the horizontal patterns of the texture) [113]. In a Gabor filter bank, rotated filters decompose the image's energy into different directional detail images, while different central frequencies facilitate the multi-scale analysis [54]. In a wavelet analysis, down/up sampling procedure generates detail images in diverse scales, and directional band-pass filters analyse the distribution of the energy across different orientations in the scaled image [11].

The applied rotation and band separation of the DWHT follows the same idea by a different manner. Rotation brings the different directions of the image into the 1D sequency transform scope, hence performs an angular decomposition. Separation of different bands in the transform domain reveals the information of various sequency bands of the texture. If we again look at (3.13), the first column of the result matrix is of course the summation of rows, which can be interpreted as the DC-term or average. The second column comprises the 1^{st} sequency harmonic of rows (sequence=1). Eventually the fourth column conveys the rows' maximum (here, 3) sequency harmonics. Therefore, after dividing the columns into a few groups (or sequency bands), sequency characteristics of the texture's rows would be analyzable in different scales. If we repeat the procedure for all directions, a comprehensive set of the sequency information of the texture will be obtained. Figures 3.11 and 3.12 represent the outputs of DWHT applied on two different images, a `CAR` and a `PAVEMENT`. The transform domain is depicted in logarithmic scale and for four different orientations, $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ from top left to bottom right. Diversity amongst objects and orientations are considerable and suggest that transform can be effective on discrimination between different images and also has directional sensitivity.

3.4 Colour Feature Extractors

Colour features are gradually playing a more important role in image classification: since colour processing has become feasible for modern hardware in terms of both colour sensing and costs of processing. Nevertheless, having faster algorithms along with maintaining an adequate level of performance is still of importance in terms of real applications.

In many studies, the colour image is considered as a composition of three separate individual colour channels (spectrums) R,G and B [34]. Although it seems to be the simplest way and suitable for digital systems, it has some important disadvantages in colour perception. For instance, coordinations in the RGB space cannot necessarily show the chromatic similarity or contrast of pixels, and in this sense, it is far from human colour perception as well.

Colour is in fact the human eye perception of an object [15]. Therefore some different colour spaces where each pixel can be characterised by features closer to human perception have been introduced. Those three factors are [15, 97]:

- **Luminance:** perception of light intensity or brightness.
- **Hue:** approximately the dominant wavelength in a mixture of light waves, or less formally, perception of dominant nuance or tone by the observer.
- **Saturation:** The perception of colourfulness, purity of colour or lack of mixed white light.

There are some different colour spaces that have been developed based on that idea, namely HLS, HSB, YIQ, YUV and *Lab* [15, 26]. Those models, which mostly use the same idea in different ways and details, can produce both intensity and chromaticity (e.g. hue and saturation) characteristics of pixels, that is very close to the human perception.

Several studies have reported the advantages of using perceptual colour spaces in colour object classification (for instance [34]). However, since the digital input data is mostly in RGB format, to employ other colour spaces we should always carry out a *RGB-to-new space* conversion, which usually is the most costly stage of the colour feature extraction procedure. Hence, trying

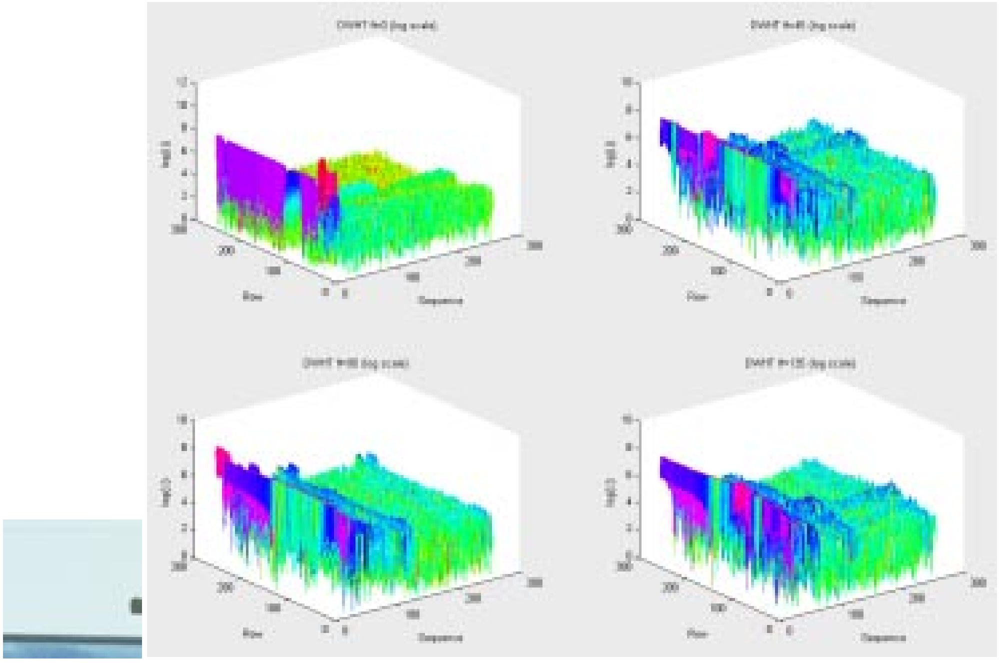


Figure 3.11: DWHT transform of a CAR. Left: Input image, Right: the output in transform domain in log scale. Note the differences between various orientations of the transform domain, and also between Figures 3.11 and 3.12 which represent two different objects.

to find out some RGB-based, easier-to-compute, and inspired-from-perception chromatic features could be a worthwhile challenge. We introduce two new faster-to-compute chromaticity features, called H_p and S_p, and compare them to corresponding chromatic features extracted from standard colour models, HLS and *Lab* . These colour spaces will also be discussed later in Appendix A.

3.4.1 New Chromatic Features: H_p and S_p

The first proposed RGB-based colour feature, H_p, is a mapping from RGB to a hue-like value, where the maximum of (R,G,B) will be mapped to an appropriate $\frac{1}{3}$ division of the possible range [0,255]. In fact, the definition of H_p has been derived from the basic meaning of *Hue*

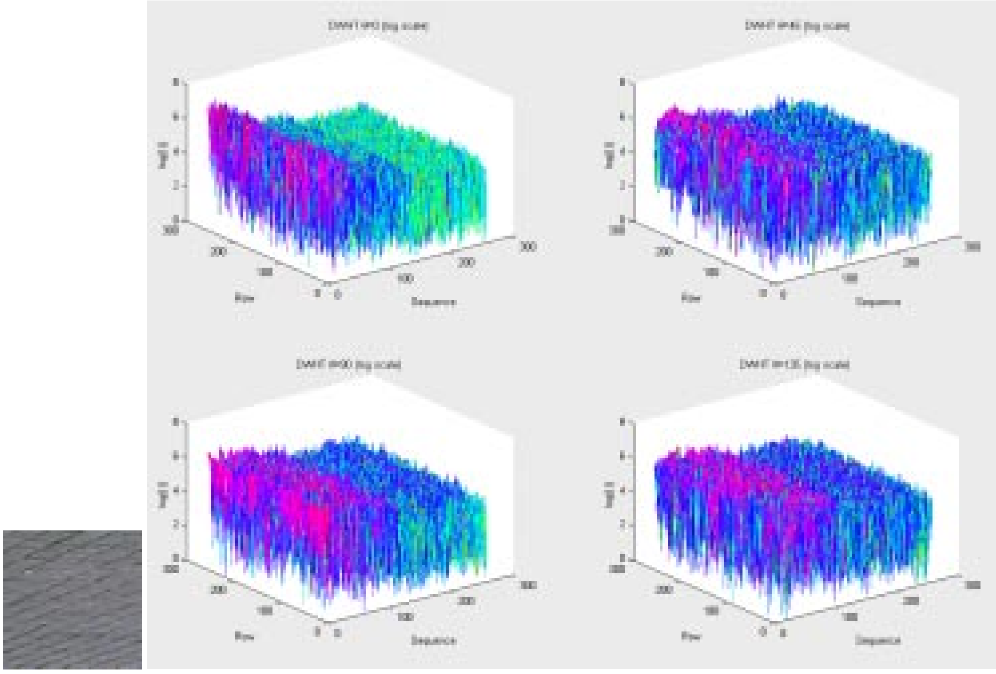


Figure 3.12: DWHT transform of a `PAVEMENT`. Left: Input image, Right: the output in transform domain in log scale. Note the differences between various orientations of the transform domain, and also between Figures 3.11 and 3.12 which represent two different objects.

which is the dominant tone of the object. This definition is not mathematically very close to the standard definition of the Hue in other colour spaces. For example, in the HLS model shown in A.1 (Appendix A), after finding out the maximum of (R,G,B), a factor of difference between other two colours would be computed as the Hue.

The second colour feature, S_p, measures the absolute difference between the maximum value and the average of colours (i.e. intensity), which can be considered as a simple close-to-saturation idea. Again, as (A.1) in Appendix A exhibits, this definition is not completely compatible with the saturation term in the HLS colour space. In fact in our definition of S_p, the role of the minimum spectrum has been eliminated, and the term $Max(R,G,B)+Min(R,G,B)$ has been roughly replaced with the average of R,G and B.

$$H_p = \begin{cases} \frac{R}{3} & \text{if } R \text{ is Maximum} \\ \frac{G+255}{3} & \text{if } G \text{ is Maximum} \\ \frac{B+(2\times255)}{3} & \text{if } B \text{ is Maximum} \end{cases} \qquad S_p = Max(R,G,B) - \frac{R+G+B}{3} \tag{3.16}$$

In performed tests, statistical moments of the chromatic channels or features of an image are calculated as its colour feature vector. After a brief review on standard RGB-to-HLS and RGB-to- *Lab* procedures (see, for instance [85, 15, 26]), proposed features seem to be faster to compute than both HLS-based and *Lab* -based chromatic features (Unless we employ 16 Mega Byte look-up tables for conversions). It is expected due to different background motivations of methods. Whereas we aim to extract simple, fast, and accurate enough colour features for object recognition purposes, standard colour spaces must impose some physical restrictions and also must provide a precise one-to-one (invertible) mapping from RGB space and vice versa. Therefore their procedure is necessarily more complicated. The HLS algorithm tests some parameters and accomplishes some comparisons, and *Lab* performs an intermediate conversion from RGB to CIE-XYZ space. Meanwhile, since they are empirical and approximated models of the hue and saturation, the computation of H_p and S_p is plain and direct. Experimental results presented in the following sections will compare the different aspects of these chromatic features in detail.

Figure 3.13 exhibits an outdoor scene image and its *Lab* (a and b), HLS (hue and saturation) and H_pS_p colour features. The common notable point amongst all features is that they attempt to discriminate different objects inside the scene. Also similarity between HLS-saturation and S_p is considerable.

3.5 Classification Tests

Our experiments consist of classifying the data set introduced in Section 3.2 using:

- Gabor only
- DWHT only
- colour features only
- Gabor with colour features
- DWHT with colour features

We divided our 724 images into a training set of 396, a test set of 160 and a validation set of 168 patches. For classification, we employed a back-propagation neural network (BPNN) classifier with one hidden layer, optimised for the best number of nodes. (for more details see Section 2.7.2).

Pre-processing for textural features included conversion to greyscale and normalisation to compensate the possible tonal effects and making the classification more texture-oriented. To convert the colour images into greyscale a so-called *RGB-to-luminance* function was applied [26]:

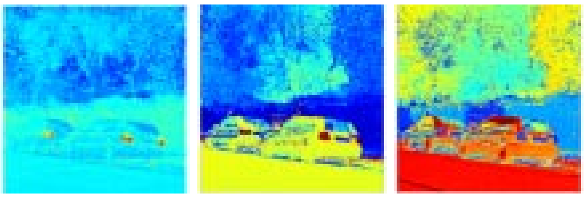
$$A_{GL} = 0.299R_A + 0.587G_A + 0.114B_A \tag{3.17}$$

where A is the input colour image and A_{GL} is the greyscale image.

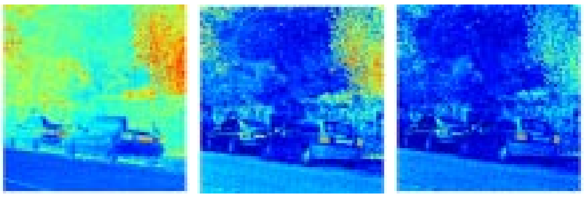
Then, the resulting greyscale image was normalised by:

$$A_{NR} = \frac{A_{GL} - \mu_{A_{GL}}}{\sigma_{A_{GL}}} \tag{3.18}$$

An Outdoor Scene Image



Lab -a HLS-Hue RGB-H_p



Lab -b HLS-Saturation RGB-S_p

Figure 3.13: Chromatic features of different colour spaces extracted from a high resolution outdoor scene image (top).

where $\mu\cdot$ and $\sigma\cdot$ are the mean and standard deviation of the greyscale image.

To evaluate classification performance, we used the Mean Square Error, MSE, as the difference between the ground truth G (i.e. the *expected* outputs of classifiers), and the network classification C (i.e. the *actual* outputs of the classifiers) across N classes. A second metric, the *Classification Accuracy*, CA, was evaluated as the percentage of correct class assignments across the complete labelled test set. MSE and CA are therefore defined as:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N}(G_i - C_i)^2}{N} \qquad \mathrm{CA} = \frac{\text{No. of correct class assignments}}{\text{Total no. of samples (i.e. 160)}} \times 100 \tag{3.19}$$

3.5.1 Classification using Textural Features: Gabor and DWHT

In the first series of tests, we employed our Gabor-based and DWHT-based textural features separately and measured their classification accuracy. Both texture features were extracted from greyscale images. Amongst several Gabor filter banks tested, a 12-filter bank with 3 central frequencies, $\omega = \{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\}$ and 4 orientations, $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ provided the optimum results. Using more central frequencies and also changing the directional resolution to $\Delta\theta = 22.5^\circ$ not only increased the number of features but also decreased the classification accuracy (unlike what Clausi and Jernigan have reported in [25] about advantages of less than 45° directional definition). Our tests indeed confirmed 3 to 5 orientations as the optimum as has been suggested by Smith and Burns [110]). The Gabor feature vector, F_{Gabor}, included 12 mean values of the filter responses (i.e. mean values of $G(\omega, \theta)$ in (3.4)).

The first DWHT feature vector, F^1_{DWHT}, included mean, standard deviation and maximum values of DWHT outputs ($H(\alpha, b)$) in four different directions $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ but without any bands separation (i.e. statistics of 4 orientations in a single sequency scale were calculated). The second DWHT feature set, F^2_{DWHT}, contained mean, standard deviation and maximum values of 4 quarter sequency bands of a non-rotated transform (i.e. statistics of 4 sequency scales without rotation). F^3_{DWHT} comprised mean values of 4 rotated half transforms arranged in 3 sequency bands, $b = \{0 - \frac{1}{4}\ ,\ \frac{1}{4} - \frac{1}{2}\ ,\ \frac{1}{2} - 1\}$ (i.e. means of 4 orientations and 3 sequency scales). Finally, F^4_{DWHT} contained 16 mean outputs of 4 sequency bands (each covering a quarter of the sequency domain) and 4 directions.

$$\begin{cases} F_{Gabor} = \{\mu(G_{\omega,\theta})\} \ , \quad \omega \in \{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\} \ , \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \\ \\ F^1_{DWHT} = \{\mu(H_\theta) \ , \sigma(H_\theta) \ , Max(H_\theta)\} \ , \quad \theta\{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \\ F^2_{DWHT} = \{\mu(H_b) \ , \sigma(H_b) \ , Max(H_b)\} \ , \quad b \in \{0 - \frac{1}{4}, \frac{1}{4} - \frac{1}{2}, \frac{1}{2} - 1\} \\ F^3_{DWHT} = \{\mu(H_{b,\theta})\} \ , \quad b \in \{0 - \frac{1}{4}, \frac{1}{4} - \frac{1}{2}, \frac{1}{2} - 1\} \ , \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \\ F^4_{DWHT} = \{\mu(H_{b,\theta})\} \ , \quad b \in \{0 - \frac{1}{4}, \frac{1}{4} - \frac{1}{2}, \frac{1}{2} - \frac{3}{4}, \frac{3}{4} - 1\} \ , \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \end{cases} \tag{3.20}$$

It is possible to apply some post-processing on the feature vectors (e.g. complex moments, grating cell operators [43] or blob detection on Gabor features [54, 103]). However, since we wanted to compare different algorithms, no post-processing was applied for any of the feature extraction schemes. Table 3.1 shows the best classification results obtained by the Gabor filters and DWHT transforms.

Texture Features	**No. Features**	**MSE**	**CA**
F_{Gabor}	12	0.183	**88.75%**
F^1_{DWHT}	12	0.257	81.88%
F^2_{DWHT}	12	0.269	80.62%
F^3_{DWHT}	12	0.237	85.00%
F^4_{DWHT}	16	0.212	**86.88%**

Table 3.1: Classification results using Gabor and DWHT texture features

The results can be summarised as:

- As expected, the optimum classification accuracy of 88.75%, (i.e. 142 correct classification out of 160) was achieved by Gabor filters. However, the best DWHT performance of 86.88% (F^4_{DWHT}) was close and comparable.

- Both extraction of features from different sequency bands and different directions influenced the DWHT performance. The classification accuracy of the single band feature set (F^1_{DWHT}) was limited to 81.88% and the single direction feature set (F^2_{DWHT}) performance was 80.62%. While (F^3_{DWHT}) with 4 directions and 3 sequency bands achieved a higher accuracy of 85%. Even adding more statistics (σ and maximum) could not compensate the lack of directional and multi-band information.

- In our tests, directionality was marginally more effective than having multi-sequency bands (again, 81.88% vs. 80.62%).

- Employing four sequency bands instead of three, added 1.88% to the classification accuracy of DWHT (85% vs. 86.88%).

Execution time of different algorithms will be discussed later in Section 3.6.

3.5.2 Classification using Chromatic Features

In the second experiment, we applied different chromatic features (H and S of the HLS, a and b of the Lab colour spaces, and RGB-based H_p and S_p, and measured their classification accuracy. As Table 3.2 illustrates, two series of experiments with colour features were carried out. In the first series, only the mean value of each chromatic channel was computed as the feature, while in the second series both the mean and the standard deviation configured the feature vector. Hence, in the first series we had two and in the second, four features per image.

Table 3.2 shows that amongst 2-feature sets of the first test, the best classification performance was achieved by Lab as 84.66%. However this performance was only slightly better than our proposed H_pS_p with CA=82.97%, while both Lab and H_pS_p outperformed HLS at 68% accuracy. Next, in the second series and by using four chromatic features, results in general exhibit higher accuracies for all feature extraction schemes. Again *Lab* was the best one with 91.16%, followed by the H_pS_p set with 87.91%, both were noticeably higher than HLS features with CA=80%. This reiterated that simple features H_pS_p gained a higher classification performance in comparison to the HLS model, but was marginally lower than the precision of *Lab* features. A notable point in this test is the considerable performance of all of colour features in general. We therefore can assume that the homogeneity of colour properties of given objects is relatively high.

Colour Features	No. Features	MSE	CA
Lab (Mean of *a* and *b*)	2	0.221	84.66 %
HLS (Mean of *H* and *S*)	2	0.28	68.00 %
RGB-Based (Mean of H_p and S_p)	2	0.234	82.97 %
Lab (Mean and STD of *a* and *b*)	4	0.13	**91.16%**
HLS (Mean and STD of *H* and *S*)	4	0.241	80.00%
RGB-Based (Mean and STD of H_p and S_p)	4	0.169	**87.91%**

Table 3.2: Classification using colour features *Lab* , HLS and the RGB-based H_pS_p

3.5.3 Classification Using Merged Texture and Colour Features

We then merged our texture features with the colour features and performed classification using all combinations for comparative purposes. The results of these are shown in Table 3.3 and were found to be very close to each other, with the best result in each texture category being Gabor+ *Lab* and DWHT with either H_pS_p or *Lab* features.

The Gabor+ *Lab* features provided the maximum classification accuracy of 94.38%. This was slightly (0.63%) higher than Gabor+H_pS_p as the second best. The DWHT merged with *Lab* or H_pS_p showed close approximation to the maximum accuracy at 93.13%. To conclude, the

Merged Features	**No. Features**	**MSE**	**CA**
Gabor + Lab	16	0.096	**94.38%**
Gabor +HLS	16	0.103	90.00%
Gabor +H_pS_p	16	0.097	93.75%
DWHT + Lab	20	0.094	**93.13%**
DWHT +HLS	20	0.123	89.38%
DWHT +H_pS_p	20	0.105	**93.13%**

Table 3.3: Classification using merged texture and colour features. For all colour spaces above, the features were μ and σ of each colour band used.

difference between the best (Gabor+ Lab) and the proposed (DWHT+H_pS_p) is 1.25%, which qualifies the new schemes, given the computational demands involved.

3.6 Summary of Computational Costs

Previous tests indicated that the proposed lower cost algorithms (DWHT and H_pS_p), along with their simple structures, both performed well in classification experiments. Therefore, if their computational times show notable decreases, they can be reasonably advisable, in particular for realtime applications. To measure and compare the running time of different algorithms, a series of tests were carried out, where both textural (DWHT vs. Gabor) and chromatic (H_pS_p vs. Lab vs. HLS) feature extraction algorithms were run several times, and on different machines. Tables 3.4 and 3.5 depict the average execution time of 20 runs on two different work stations: a PC Pentium III-700 MHz machine and a Sun Ultra-10 Sparc work station.

As Table 3.4 exhibits, the DWHT feature extraction algorithm was 11.46 times faster than Gabor filtering on PC, and 10.29 times on Sun. Meanwhile, both PC Pentium and Sun Sparc are fast, powerful CPUs and we expect more differences on lower level hardware. The chromatic

Textural Algorithm	**PC PIII 700 MHz**	**Sun Sparc**
Gabor	2.75 sec	7.10 sec
DWHT	0.24 sec	0.69 sec
Gabor/DWHT Ratio	11.46	10.29

Table 3.4: Average execution time for texture feature extractions (sec).

Chromatic Algorithm	**PC PIII 700 MHz**	**Sun Sparc**
Lab	0.30 sec	0.50 sec
HLS	0.73 sec	1.24 sec
H_pS_p	0.05 sec	0.11 sec
Lab $/H_pS_p$ Ratio	6.05	4.54
HLS$/H_pS_p$ Ratio	14.6	11.27

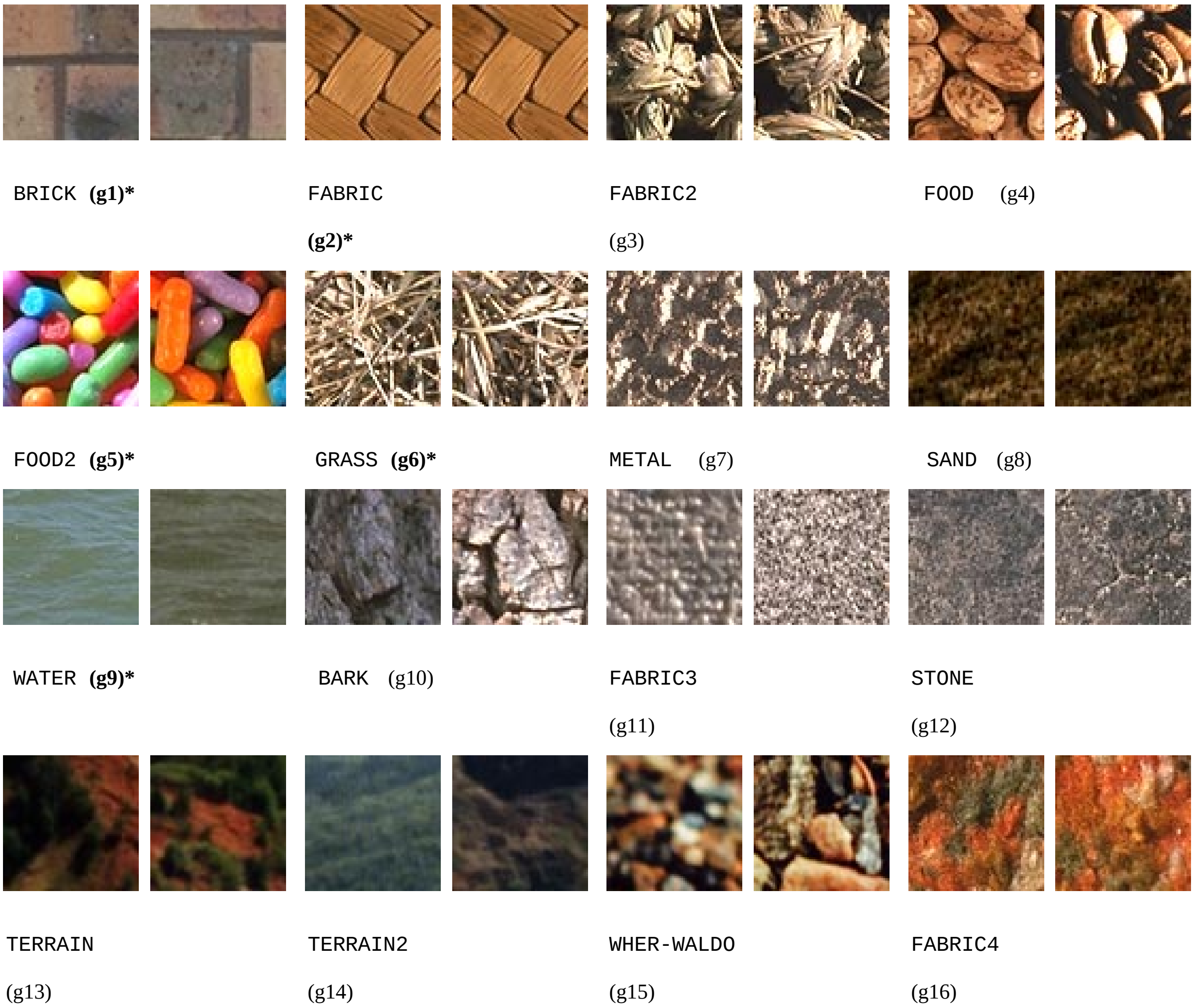
Table 3.5: Average execution time for colour feature extractions (sec).

feature extraction times, presented in Table 3.5, showed that the RGB-based H_pS_p scheme was the quickest, and 6.05(4.54) times faster than Lab and 14.6(11.27) times faster than HLS on PC(Sun). This notable reduction in computation time even on high speed CPUs, proves the considerable lower cost of the proposed features that along with their notable classification accuracy, make them highly plausible for realtime applications. For example, Gabor$+$ Lab needed 3.05 sec to be calculated at 94.38% accuracy, whereas DWHT$+H_pS_p$ needed 0.29 sec at 93.13% accuracy.

3.7 Experiments with VisTex

To evaluate the robustness and generality of the proposed approaches, we repeated the experiments, this time on a pseudo-standard texture database. Amongst a few commonly used texture test suites, (namely Brodatz [18], MeasTex [88], VisTex [69]), VisTex was chosen since it has a rich variety of colour textures and also is easily downloadable via the Internet. Introduced by the MIT Media Lab, the VisTex suite has been gathered to assist in computer vision algorithm evaluation and comparison on a common data set. VisTex contains several colour images of mostly natural textures which have been photographed under real world circumstances to represent real world conditions.

Sixteen different textures from the VisTex database were selected and 192 non-overlapping 64×64 pixel patches were extracted from each, resulting in 3072 patches. The test procedure and circumstances were kept compatible with the former high resolution outdoor scene experiments, apart from adding *Ordinary Hadamard Transform* (OHT) as a new texture feature extractor, and the *Normalised RGB* (NRGB) as a new chromatic feature extractor. Experiments were divided into two stages. Firstly, all textural and chromatic features were tested to segregate all 16 groups of textures. Secondly, to have a more precise evaluation of the advantages of the new directional Hadamard approval on directional textures analysis, a comparative test between OHT and DWHT was arranged. In this test only 5 groups of more directional textures were involved. Figure 3.14 exhibits samples of all 16 groups of tested VisTex textures. the five directional textures (`BRICK` (g1), `FABRIC` (g2), `FOOD2` (g5), `GRASS` (g6), and `WATER` (g9)) of the second experiment are highlighted by $*$. The applied classifier was an ordinary BPNN. Samples of 3072 patches were divided into 50% for training and 50% for testing (i.e. randomly selected 96 training and 96 testing patches per group). To increase the reliability and generality of the outcome, the training and testing subsets were swapped and the classifier was trained and tested once again. The reported results are in fact the average of these two iterations.



BRICK **(g1)***

FABRIC **(g2)***

FABRIC2 (g3)

FOOD (g4)

FOOD2 **(g5)***

GRASS **(g6)***

METAL (g7)

SAND (g8)

WATER **(g9)***

BARK (g10)

FABRIC3 (g11)

STONE (g12)

TERRAIN (g13)

TERRAIN2 (g14)

WHER-WALDO (g15)

FABRIC4 (g16)

Figure 3.14: Some samples of the applied 16 groups of VisTex textures. * indicates the directional textures involved in the second experiment.

3.7.1 Texture-based Classification

For the texture based classification test, we first applied Gabor filters, DWHT and OHT in three frequency/sequency bands, and four directions. The mean values of filter responses and transforms were calculated as the features:

$$\begin{cases} F^1_{Gabor} = \{\mu(G_{\omega,\theta})\} \ , & \omega \in \{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\} \ , \ \theta \in \{0°, 45°, 90°, 135°\} \\ F^1_{DWHT} = \{\mu(H_{b,\theta})\} \ , & b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-1\} \ , \ \theta \in \{0°, 45°, 90°, 135°\} \\ F^1_{OHT} = \{\mu(H_b)\ \} \ , & b \in \{0, \frac{1}{4}, \frac{2}{4}\} \end{cases} \tag{3.21}$$

As Table 3.6 represents, the classification accuracy of the Gabor filters is the highest (80.86%), but only very slightly higher than DWHT (80.21%). OHT however, clearly shows a lower performance of 69.14%.

Texture Features	**No. Features**	**MSE**	**CA**
Gabor Filters (F^1_{Gabor})	12	0.262	80.86%
Directional Hadamard (F^1_{DWHT})	12	0.278	80.21%
Ordinary Hadamard (F^1_{OHT})	3	0.416	69.14%

Table 3.6: Classification results of 16 VisTex textures using mean values of 3 frequency/sequency bands as the texture features.

In the second test, standard deviations of the outputs were inserted to the feature vectors to potentially increase the performances. New feature vectors included 24 features for Gabor and DWHT and 6 features for OHT:

$$\begin{cases} F^2_{Gabor} = \{\mu(G_{\omega,\theta}) \ , \ \sigma(G_{\omega,\theta})\} \ , & \omega \in \{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\} \ , \ \theta \in \{0°, 45°, 90°, 135°\} \\ F^2_{DWHT} = \{\mu(H_{b,\theta}) \ , \ \sigma(H_{b,\theta})\} \ , & b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-1\} \ , \ \theta \in \{0°, 45°, 90°, 135°\} \\ F^2_{OHT} = \{\mu(H_b) \ , \ \sigma(H_b)\ \} \ , & b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-1\} \end{cases} \tag{3.22}$$

As Table 3.7 exhibits, for the first time DWHT showed the highest classification accuracy with 82.72%, marginally improving on the Gabor filters' 82.06% accuracy. OHT is still well behind at 73.40% correct classification rate. This test indicated that it was possible to increase the performance of DWHT even further by choosing appropriate features. Including the standard deviation which in a sense showed the 'normalised energy' of the transform output or filter response [2], had more significant effects on DWHT and in particular OHT than Gabor filters. Several other tests, where higher order moments or different statistics (e.g. max or min value) were added to the feature sets, did not increase the classification accuracy of any of the feature sets at all.

Texture Features	**No. Features**	**MSE**	**CA**
Gabor Filters (F^2_{Gabor})	24	0.253	82.06%
Directional Hadamard (F^2_{DWHT})	24	0.234	82.72%
Ordinary Hadamard (F^2_{OHT})	6	0.409	73.40%

Table 3.7: Classification results of 16 VisTex textures using texture features with 3 frequency/sequency bands.

In the third test, we added the fourth frequency/sequency band to the features. So, for DWHT and OHT the bandwidth was changed to $\frac{1}{4}$. For Gabor filters, we ignored the usual log_2 bandwidth scheme and inserted the fourth central frequency at $\omega = \frac{3\Omega_M}{4}$:

$$\begin{cases} F^3_{Gabor} = \ \{\mu(G_{\omega,\theta}) \ , \ \sigma(G_{\omega,\theta})\} \ , \quad \omega \in \{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \frac{3\Omega_M}{4}\} \ , \ \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \\ F^3_{DWHT} = \ \{\mu(H_{b,\theta}) \ , \ \sigma(H_{b,\theta})\} \ , \quad b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-\frac{3}{4}, \frac{3}{4}-1\} \ , \ \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \\ F^3_{OHT} = \ \{\mu(WHT^{2D}_b) \ , \ \sigma(WHT^{2D}_b)\} \ , \quad b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-\frac{3}{4}, \frac{3}{4}-1\} \end{cases} \tag{3.23}$$

where WHT^{2D} is the output of an ordinary 2D Hadamard transform (see (3.8) for details, and also Figure 3.8(c) for definition of ‘sequency bands’ in this case).

The combination of central frequencies in F^3_{Gabor} provided the best result amongst several examined. For instance, $\{\frac{\Omega_M}{16}, \frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\}$ and $\{\frac{3\Omega_M}{32}, \frac{3\Omega_M}{16}, \frac{3\Omega_M}{8}, \frac{3\Omega_M}{4}\}$ exhibited 2.41% and 1.75% less accuracy than F^3_{Gabor} respectively. Once again, the mean and the standard deviation of the outputs were calculated as feature vectors. Table 3.8 depicts the classification results.

Results showed that again the best performance belongs to DWHT, and F^3_{Gabor} stays 2.18% behind that. 77.96% classification accuracy of the OHT was 7.16% lower than DWHT. Involving

Texture Features	**No. Features**	**MSE**	**CA**
Gabor Filters (F^3_{Gabor})	32	0.260	82.94%
Directional Hadamard (F^3_{DWHT})	32	0.194	85.12%
Ordinary Hadamard (F^3_{OHT})	8	0.323	77.96%

Table 3.8: Classification results of 16 VisTex textures using texture features with 4 frequency/sequency bands.

the 4^{th} frequency band at a rather high frequency of $\omega = \frac{3\Omega_M}{4}$ also has added 2.08% to the Gabor filters performance and suggests that in some cases ignoring the typical log_2 arrangement could be beneficial.

In the fourth experiment, we switched to our second VisTex data set which contains five more-directional textures $\{$`BRICK, FABRIC, FOOD2, GRASS, WATER`$\}$. This data set has 960 images. This test aimed to evaluate the effectiveness of the DWHT on the directional textures more clearly. Also, a Principal Components Analysis has been applied on F^3_{DWHT} to decrease its feature space dimensionality from 32 to 8, to facilitate a fairer comparison between the 8-feature OHT and DWHT. The results, illustrated in Table 3.9, suggested that DWHT performs well in directional texture classification. Its accuracy (97.92%) was 13.13% higher than the OHT's 84.79%. This also indicated that compared to the third experiment with 16 groups of textures, DWHT is even more effective on directional textures. The difference between performances of DWHT and OHT in the third test was 7.16, whilst it was 13.13 in the more directional fourth test. Even after applying the PCA, the 8-feature DWHT is 8.97% more accurate than OHT.

Texture Features	**No. Features**	**MSE**	CA
Directional Hadamard (F^3_{DWHT})	32	0.030	97.92%
Ordinary Hadamard (F^3_{OHT})	8	0.169	84.79%
PCA on F^3_{DWHT} (F^{3p}_{DWHT})	8	0.092	93.76%

Table 3.9: DWHT and OHT performance comparison, applied on more directional VisTex textures.

3.7.2 Colour-based Classification on VisTex

Along with three formerly mentioned colour features, in this test a new *Normalised RGB* [82] feature set was examined. In the normalised RGB (NRGB), normalised values of R, G and B for each pixel are calculated as:

$$r = \frac{R}{R+G+B} \quad , \quad g = \frac{G}{R+G+B} \quad , \quad b = \frac{B}{R+G+B} \tag{3.24}$$

The NRGB feature vector is then formed by the mean and standard deviation of *r*,*g* and *b* channels:

$$F_{NRGB} = \{\mu(r) \, , \, \sigma(r) \, , \, \mu(g) \, , \, \sigma(g) \, , \, \mu(b) \, , \, \sigma(b)\} \tag{3.25}$$

The feature extraction schemes for HLS, *Lab* and H_pS_p are identical to that mentioned in Section 3.5.2. Table 3.10 illustrates the classification results, where *Lab* shows the best performance with CA=74.28%. The second best is H_pS_p with 72.07%. Classification accuracy of NRGB and HLS are considerably lower at 65.82% and 64.71%.

Colour Features	**No. Features**	**MSE**	**CA**
Lab (Mean and STD of a, b)	4	0.395	74.28%
HLS (Mean and STD of H, S)	4	0.515	64.71%
RGB-based (Mean and STD of H_p, S_p)	4	0.389	72.07%
Normalised RGB (Mean and STD of R_n, G_n, B_n)	6	0.401	65.82%

Table 3.10: 16 groups of VisTex textures classification performance using chromatic features

The outcomes of colour feature-based classification on VisTex support the former results of the outdoor scene experiments. Again *Lab* is the most accurate one, while H_pS_p stands marginally behind that and clearly in front of both HLS and the new NRGB. This experiment shows that hue-like and saturation-like H_pS_p outperform non-perceptual NRGB features, and indicates the advantage of even simple perceptual-based chromatic features in colour texture classification. Relatively poor performance of NRGB may also challenge the generality of the results reported by Manian and Vasquez in [82].

3.8 Conclusion

We described novel and fast methods for extraction of both directional texture features using the Walsh-Hadamard-based DWHT transform and simplified hue and saturation-like H_pS_p chromatic features. The methods were applied to high resolution outdoor scenes for colour object classification. The DWHT concept of sequency captured the lower and higher harmonics present in high resolution images very well. The performance of the proposed features were compared, for accuracy and speed, against Gabor and HLS/ *Lab* features. We demonstrated that the performance of the new features were highly comparable at a massively reduced computational cost. Furthermore, similar experiments on a pseudo-standard VisTex texture suite confirmed that DWHT was significantly more effective than the ordinary Hadamard transform and even in some cases slightly more precise than the well-known high performance Gabor filters for texture classification.

The advantages of the proposed Hadamard based feature extraction scheme relies on different factors. First of all, contrary to many previous works which employed local Walsh-Hadamard filters (e.g. 3×3 in [118] or 4×4 in [62]), we used larger patches (e.g. 64×64 or 128×128) and a global transform method. Hence, processed by an ordered Hadamard matrix, a wider range of sequency information is extractable. Moreover, the employment of higher resolution images allows us to use bigger patches of objects and gain a wider range of frequency/sequency. Secondly, the employed rotation scheme covers four main orientations of a texture $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$ and provides an organised set of oriented sequency features which can be evaluated, ranked and utilised in image classification tasks.

The RGB-based H_pS_p colour features have been inspired from perceptual phenomenon, Hue and Saturation. The proposed H_pS_p performed well in our object classification tests and was comparable with the well-known Lab -based colour features. Further experiments where those colour features, along with the NRGB, were applied to the VisTex set, again confirmed the quality of the proposed H_pS_p features. In general, their performance was comparable with the best achieved by Lab and was well in front of HLS and NRGB.

In the following chapter we will apply DWHT once again, along with Gabor filters and some other texture analysis algorithms, this time to detect abnormalities in textured tiles.

Chapter 4

Defect Detection in Textured Tiles

4.1 Introduction

In this chapter we present a study in texture quality classification and abnormality detection. We apply several popular techniques and propose a novel algorithm for texture defect detection and evaluate it on a randomly textured tile data set.

The main focus of this study is to explore statistical and signal processing approaches for abnormality detection. The following methods are investigated and compared which represent a cross-section of state-of-the-art techniques:

- Statistical:
    - Ordinary Histogram
    - Local Binary Pattern
    - Co-occurrence Matrix

- Signal processing:
    - Gabor Filtering
    - Ordinary Walsh-Hadamard Transform
    - Directional Walsh-Hadamard Transform (as proposed in Chapter 3)
    - Ordinary Discrete Cosine Transform
    - Directional Discrete Cosine Transform (similar with DWHT)
    - Eigenfiltering
    - Gabor Composition (the new method proposed in this chapter)

For each algorithm several parameter optimisation tests were also carried out to maximise the performance.

The next section starts with a description of the test framework, including the data set and classifier specifications. The algorithms used and the classification test results are discussed in Section 4.3. The new Gabor Composition method is presented separately in Section 4.4. Then we compare the computational costs and performances of different algorithms in Section 4.5. The chapter will conclude in Section 4.6.

4.2 Classification Tests Framework

4.2.1 Data Set

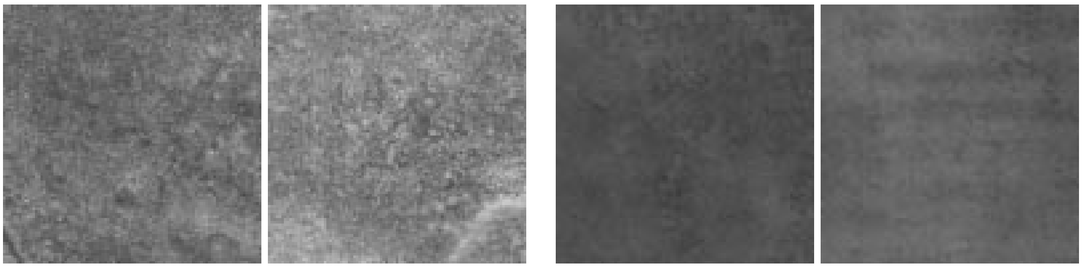
Our randomly textured tile data set (TDS) contained 1883 grey level 256 × 256 pixel images selected from larger (e.g. 1024 × 1024 pixel) tile images. The resolution of the images was 4 pixel/mm (0.25 mm/pixel) and they were grabbed in the factory environment with a prototype inspection system. The classification algorithm attempted to classify input samples into normal and abnormal classes. In reality, the defect detection system could separate the input image into a few non-overlapping 256×256 patches, and considered the tile as normal if all the patches were normal, otherwise the tile would be considered as abnormal.

As Table 4.1 illustrates, the TDS comprised of eleven tile types. Samples of each type were divided into three non-overlapping categories of *Normals*, *Abnormals* and *Templates*.

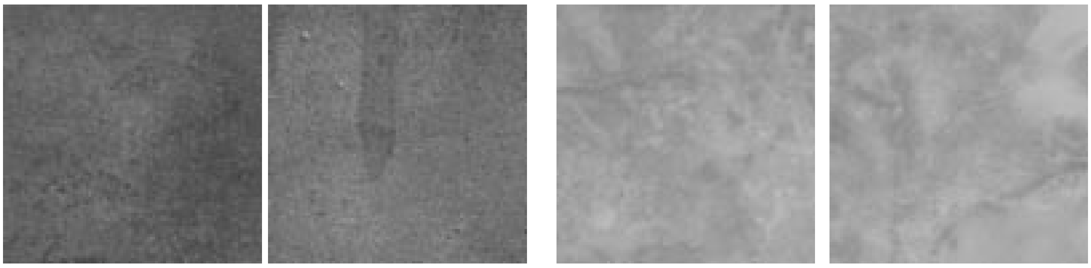
	Number of Samples		
Tile Type	**Normals**	**Abnormals**	**Templates**
`ARDES`	22	22	224
`ARWIN`	40	40	82
`CASA`	72	72	4
`DJZAM`	20	20	170
`DJZUL`	20	20	170
`KIS`	23	23	39
`LRSIDE`	46	46	135
`PRODO`	44	44	6
`PRODT`	38	38	5
`SLTNP`	26	26	161
`SYM`	20	20	145
Total	371	371	1141
			1883

Table 4.1: Tile types and number of samples in the TDS

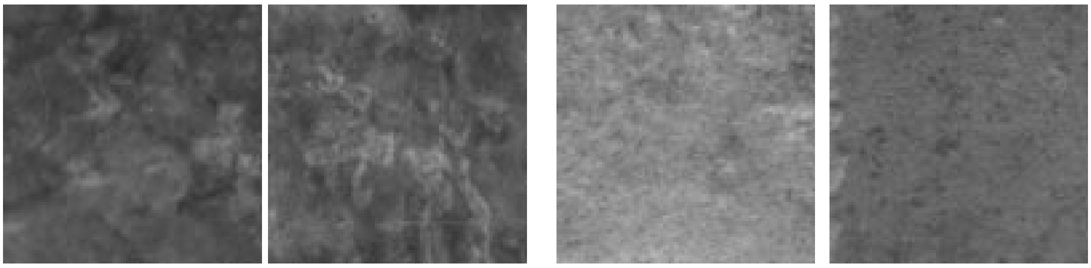
Figures 4.1 and 4.2 illustrate normal and abnormal samples of all eleven types of tiles. As these figures show, the tiles used were all random textures with different character and coarseness. Defects were also decisively diverse in attributes. Unexpected bars in `ARWIN` and `PRODO` samples, small circular brighter regions in LRSIDE and SYM, or local density problems in SLTNP were only some instances of defects in the TDS.



ARDES ARWIN

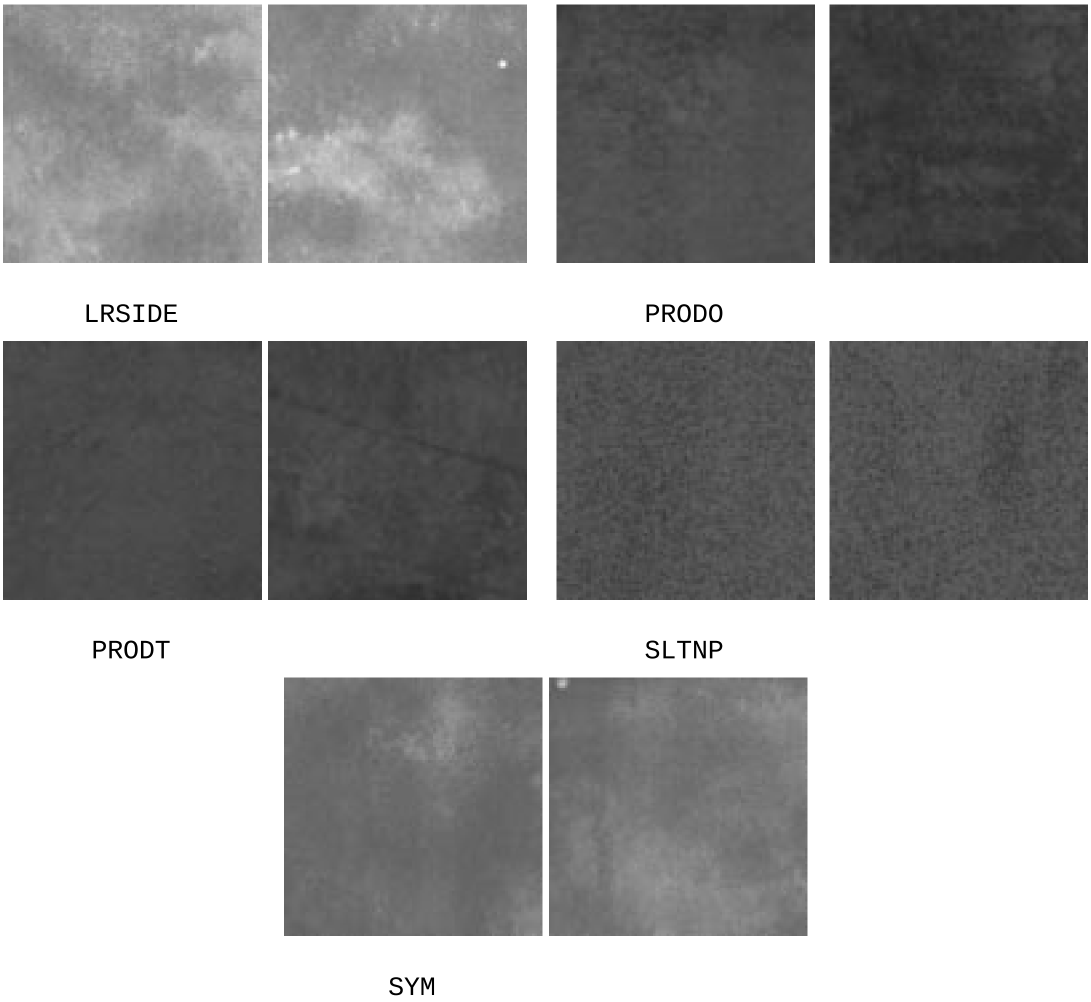


CASA DJZAM



DJZUL KIS

Figure 4.1: Samples of the TDS: normal(left) and abnormal(right) tiles from 11 different models. ARDES: abnormal bottom/right corner, ARWIN: dark horizontal bars, CASA: dark stain, DJZAM: abnormal top/right corner, DJZUL: thin crack-like line at the bottom, KIS: blobs at the left edge (to be continued in 4.2).



LRSIDE

PRODO

PRODT

SLTNP

SYM

Figure 4.2: (Continued from 4.1) Samples of the TDS: normal(left) and abnormal(right) tiles from 11 different models. `LRSIDE:` bright pinhole-like spot, `PRODO:` horizontal bars, `PRODT:` diagonal thin lines, `SLTNP:` regions with denser patterns at the left half, `SYM:` bright spot.

A review of the data set and some sample histograms showed that the image grabber system could provide homogeneous 256×256 image samples with a uniform luminance throughout each. However, in some types there was a slight change in brightness between different samples (e.g. see `CASA` tiles in Figure 4.1). Therefore, a simple normalisation was accomplished throughout the images to compensate the luminance variations:

$$x_p = x - \mu_x \tag{4.1}$$

where x is the input image, μ_x is its mean value, and x_p is the normalised image. This was the only pre-processing carried out on the data set prior to the feature extraction stage.

4.2.2 Classifiers

In this chapter we employed a traditional approach to the normal/abnormal samples classification problem, comprising feature extraction, training the classifier with both normal and abnormal samples and testing the performance. A BPNN and a KNN classifier were separately used to evaluate the performance of the feature extraction algorithms. The BPNN classifier was a single hidden layer network, trained on a type-wise paradigm and optimised on the number of hidden nodes. The number of hidden nodes tried were $n = \{2,3,4,5\}$. We did not test larger n due to the relatively limited number of samples per type in the TDS. The KNN classifier was also trained in a type-wise manner and optimised on the closest neighbourhood parameter $K = \{1,3\}$. Almost always $K = 3$ provided the better results. In general we did not try $K = 5$ or larger, however, when we tested $K = \{1,3,5\}$ for two types with more samples, `CASA` and `LRSIDE`, $K = 3$ was still the optimum choice.

To increase the generality of the classification results, a k-fold cross-validation scheme with $k = 5$ was implemented for both classifiers. For each type, samples were divided into 5 non-overlapping training/testing subsets, where 80% of samples formed the training and the remaining 20% built the testing subsets. The reported results are in fact the average of 5 iterations of the training/testing cross-validation procedure (see Section 2.7.2 for more details on cross-validation algorithms). Classification performance is presented by classification accuracy, CA, or the percentage of correct class assignments across the complete labelled test set:

$$CA = \frac{N_{nn} + N_{aa}}{N_{total}} \times 100 \tag{4.2}$$

where N_{nn}, N_{aa}, and N_{total} are the number of test samples which are classified as normal while they are indeed normal, the number of test samples which are classified as abnormal while they are indeed abnormal, and the total number of test samples respectively.

4.3 Classification Experiments

4.3.1 Ordinary Histograms

We applied histograms to measure the level of complexity of this classification problem, and the validity of assuming all kinds of defects as *textural abnormalities*. If a simple non-textural operator like a histogram shows a high classification performance, the complexity of the classification problem or the validity of that assumption can be questioned. The first histogram feature vector, F_{Hist}^{1}, contained 256 bins of the tile image histogram. As Table 4.2 (columns 2 and 3) illustrates, using the KNN and the BPNN classifiers the average CA were 68.06% and 77.54% respectively, which are not adequate in defect detection applications. Then a PCA was applied to decrease the feature space dimensionality toward something more comparable with other methods that shall be used here. The result was the second histogram feature vector $F_{Hist,PCA}^{n=48}$, which contained the first 48 features of the PCA. Tests showed that $F_{Hist,PCA}^{n=48}$ with 65.45% and 76.34% accuracy on KNN and BPNN is even slightly worse than F_{Hist}^{1}. Next, a PCA-based optimisation test was performed, in which different numbers of PCA features, n, were tried, where $n \in \{16, 26, 36, ..., 246\}$. This was to find out whether or not PCA can increase the classification accuracy at all. The outcomes showed that no PCA feature reduction can improve the performance. Table 4.2 in columns 2 and 3 represents the classification performance of the histogram on different types of the TDS. The average performance of 77.54% achieved by the ordinary histogram suggests that defect detection in TDS is complex. We next show the employment of texture-oriented methods and evaluate their performances.

4.3.2 Local Binary Patterns (LBP)

In this defect detection experiment, we applied basic LBP operators on a 3 × 3 neighbourhood, 8 border pixels ($P = 8$), and single distance ($d = 1$), :

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{4.3}$$

where $s(\cdot)$ is the sign function, g_p and g_c are grey levels of border pixels and central pixel respectively, and P is the number of pixels in the neighbourhood (see Section 2.3.1 for more details).

Although there are some reports on the high performance of LBP in texture classification (e.g. [80, 92, 93]), in our tests LBP did not show promising defect detection accuracy. The performance of 256-feature LBP feature vectors, (F^1_{LBP}), was limited to 78.93% on the KNN and 84.18% on the BPNN classifiers. Again, to find whether or not any improvement in classification performance was obtainable via reduction of the feature space dimensionality (particularly for the BPNN classifier), a PCA analysis was employed on F^1_{LBP}. None of the PCA feature vectors attempted with different numbers of features, n, where $n \in \{16, 26, 36, ..., 246\}$, could improve the classification performance and F^1_{LBP} stayed our best LBP feature vector. Table 4.2 in columns 4 and 5 represents the LBP defect detection performance which was better than the ordinary histogram, however was not adequate for a defect detection application.

4.3.3 Co-occurrence Matrices

Grey level co-occurrence matrices are one of the oldest, and still one of the most commonly used, texture modelling and classification algorithms. For an n-grey level image, x, grey level co-occurrence matrix, $\Phi_{d,\theta}(x)$ is a $n \times n$ matrix which contains the pairwise relations between

	Histogram		**LBP**	
	$\mathrm{CA}_{\mathrm{KNN}}$	$\mathrm{CA}_{\mathrm{BPNN}}$	$\mathrm{CA}_{\mathrm{KNN}}$	$\mathrm{CA}_{\mathrm{BPNN}}$
Tile Type	F^1_{Hist}	F^1_{Hist}	F^1_{LBP}	F^1_{LBP}
`ARDES`	75.00%	72.35%	65.00%	77.50%
`ARWIN`	68.75%	79.25%	86.25%	97.19%
`CASA`	62.86%	78.00%	82.00%	94.29%
`DJZAM`	75.00%	76.75%	80.00%	83.75%
`DJZUL`	55.00%	71.45%	75.00%	73.75%
`KIS`	55.00%	71.25%	75.00%	77.50%
`LRSIDE`	72.00%	84.2%	80.00%	97.00%
`PRODO`	72.50%	81.32%	85.00%	83.75%
`PRODT`	72.50%	80.34%	80.00%	77.50%
`SLTNP`	85.00%	81.75%	80.00%	77.50%
`SYM`	55.00%	76.25%	80.00%	86.25%
Average	68.06%	77.54%	78.93%	84.18%
Variance	0.0098	0.0019	0.0033	0.0073

Table 4.2: Defect detection results using ordinary histograms and LBP methods.

pixel intensities for a particular distance d, and orientation θ. Each element (i, j) of Φ, shows how many pixel pairs with respective intensities i and j exist in the image considering a certain distance, d, and orientation, θ. Φ is therefore a function of three parameters:

$$\Phi_{d,\theta}(x) = GLCM(x, d, \theta) \tag{4.4}$$

The co-occurrence matrices can reveal certain properties of the texture. For example, if larger values have gathered around diagonals of $\Phi_{d,*}(x)$, the texture x is relatively coarse, regarding the distance d [116]. Various combinations of parameters d and θ can generate many different GLCM matrices for a single image. It is also apparent that GLCM matrices cannot be directly used as texture representatives. Therefore, to obtain the texture's feature vector, a few pseudo-standard functions will usually be applied on the GLCM matrices. Haralick *et al* [44] introduced 14 pseudo-standard GLCM functions. Amongst them, we chose 6 popular functions:

$$\begin{cases} f_1 = \text{ Maximum } = Max_{i,j}(\Phi(i,j)) \\ f_2 = \text{ Energy } = \sum_{i,j}\Phi(i,j)^2 \\ f_3 = \text{ Entropy } = -\sum_{i,j}\Phi(i,j)log(\Phi(i,j)) \\ f_4 = \text{ Correlation } = \sum_{i,j}\frac{(i-\mu_i)(j-\mu_j)\Phi(i,j)}{\sigma_i\sigma_j} \\ f_5 = \text{ Inverse Difference Moment } = \sum_{i,j}\frac{1}{1+(i-j)^2}\Phi(i,j) \\ f_6 = \text{ Inertia } = \sum_{i,j}(i-j)^2\Phi(i,j) \end{cases} \tag{4.5}$$

Each proposed function represents certain properties of the texture. Maximum and Energy, (f_1 and f_2), are basic statistics of Φ. Entropy, f_3, measures the texture homogeneity. Correlation function, f_4, is image linearity metric. Linear directional structures in direction θ result in large correlation values in that direction. This can also measure the image coarseness. Inverse Difference Moment (IDM), f_5, measures the extent to which the same tones tend to be neighbours. Inertia (or Contrast), f_6, is a texture dissimilarity measure. The reason behind this selection is, while f_1 and f_2 are basic descriptors of Φ, we believe that f_3 to f_6 are all implicitly defect sensitive. Homogeneity, correlation, linearity, and dissimilarity all may change when a flaw occurs.

We considered four directions $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ in all co-occurrence tests. To find out the nearly optimum distance d, we assumed $d = 1$ as a basic distance and tried two distance sets of *closer* and *farther* distances which built up the first two GLCM feature vectors:

$$\begin{cases} F^1_{GLCM} = \{f_i(\Phi_{d,\theta})\}\ ,\ i \in \{1,...,6\},\ d \in \{1,2,3,4,5\},\ \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\} \\ F^2_{GLCM} = \{f_i(\Phi_{d,\theta})\}\ ,\ i \in \{1,...,6\},\ d \in \{1,20,30,40,50\},\ \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\} \end{cases} \tag{4.6}$$

Each feature vector contained 120 features of 4 directions, 5 distances, and 6 functions. Represented in the first two columns of Tables 4.3 and 4.4, the CA of these two vectors on both classifiers suggest that shorter distances are more effective than farther ones (72.10% vs. 68.93% for KNN and 82.29% vs. 80.91% for BPNN). The accuracy however, is still lower than LBP. Next we attempted to improve the GLCM performance using 3 closer-to-farther *log*-scale distances in a reasonable range of $d = \{1 \cdots 49\}$. This resulted in four more GLCM feature vectors as:

$$\begin{cases} F^3_{GLCM} = \{f_i(\Phi_{d,\theta})\} \ , \ d \in \{1,4,16\} \\ F^4_{GLCM} = \{f_i(\Phi_{d,\theta})\} \ , \ d \in \{1,5,25\} \\ F^5_{GLCM} = \{f_i(\Phi_{d,\theta})\} \ , \ d \in \{1,6,36\} \\ F^6_{GLCM} = \{f_i(\Phi_{d,\theta})\} \ , \ d \in \{1,7,49\} \end{cases} \tag{4.7}$$

For all vectors, $i \in \{1,...,6\}$ and $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, thus each *log*-scale feature vector comprised 72 features. The reason underpinning the selection of the *log*-scale distances is that in natural textures the correlation between two pixels diminishes as their distance is growing. Results in Tables 4.3 and 4.4 suggest that the performance of *log*-scale distances are far better than that of the only-close and only-far, and improve on the LBP algorithm. The best feature vector was F^5_{GLCM} with $d = \{1,6,36\}$, achieving 80.45% and 95.07% correct classification on KNN and BPNN respectively.

Finding the optimum subset of co-occurrence functions is another problem to be solved. Strand and Taxt [114] employed Energy (f_2), Inertia (f_6), Correlation (f_4) and Entropy (f_3). Clausi [24] in his detailed study compared several parameters of the GLCM algorithm. He concluded that a combination of Inertia (f_6), Entropy (f_3) and Correlation (f_4) provides the highest classification accuracy on average. He also believed that utilising all the functions will not increase the performance due to their considerable redundancy. We tried a put-one-aside scheme (or greedy scheme [123]) to find out if elimination of any function can increase (or at least keep) the accuracy of the 6-function F^5_{GLCM}. We put functions aside one by one and repeated the whole classification procedure for the new 5-function feature vector. Elimination of three functions Entropy (f_3), IDM (f_5), and Inertia (f_6) increased the performance marginally. Amongst them, IDM elimination was the most effective one with 0.27% and 0.18% improvement on KNN and BPNN (i.e. $\mathrm{CA}_{\mathrm{KNN}}$=80.72% and $\mathrm{CA}_{\mathrm{BPNN}}$=95.25% respectively). The resulting feature vector is the 5-function, 60-feature F^7_{GLCM}. The put-one-aside scheme was then repeated once again, and this time only elimination of Entropy (f_3) improved the performance to 81.08% and 97.09% for the 4-function, 48-feature F^8_{GLCM}. This is indeed a moderate performance for the KNN and a promising one for the BPNN classifier. No advantage was gained by eliminating any more functions. Tables 4.3 and 4.4 show the results of these GLCM classification tests.

	Distance Optimisation						**Function Optimisation**	
Tile Type	F^1_{GLCM}	F^2_{GLCM}	F^3_{GLCM}	F^4_{GLCM}	F^5_{GLCM}	F^6_{GLCM}	F^7_{GLCM}	F^8_{GLCM}
`ARDES`	83.33%	45.00%	80.00%	80.00%	80.00%	80.00%	83.33%	80.00%
`ARWIN`	70.00%	70.63%	88.75%	85.00%	85.00%	85.00%	85.00%	86.25%
`CASA`	70.14%	63.19%	68.57%	68.57%	71.43%	64.29%	71.43%	67.14%
`DJZAM`	82.50%	87.50%	90.00%	90.00%	100%	90.00%	90.00%	100%
`DJZUL`	92.50%	65.00%	95.00%	95.00%	95.00%	95.00%	90.00%	95.00%
`KIS`	52.50%	59.82%	80.00%	85.00%	80.00%	85.00%	58.33%	80.00%
`LRSIDE`	94.60%	90.34%	96.00%	96.00%	96.00%	90.34%	90.34%	96.00%
`PRODO`	80.65%	76.79%	70.00%	77.50%	75.00%	55.00%	76.19%	62.50%
`PRODT`	47.50%	52.50%	42.50%	62.50%	57.50%	82.50%	90.48%	80.00%
`SLTNP`	53.63%	81.79%	90.00%	90.00%	90.00%	90.00%	92.86%	90.00%
`SYM`	65.77%	65.71%	45.00%	50.00%	55.00%	55.00%	60.00%	55.00%
Average	72.10%	68.93%	76.89%	79.96%	80.45%	79.28%	80.72%	81.08%
Variance	0.0261	0.02018	0.0351	0.0207	0.0223	0.0208	0.0157	0.0210

Table 4.3: Defect detection results using GLCM and the KNN classifier.

To conclude, optimised GLCM features performed effectively in this defect detection experiment. However, it is also of importance to select a well prepared set of distances and functions

	Distance Optimisation						**Function Optimisation**	
Tile Type	F^1_{GLCM}	F^2_{GLCM}	F^3_{GLCM}	F^4_{GLCM}	F^5_{GLCM}	F^6_{GLCM}	F^7_{GLCM}	F^8_{GLCM}
`ARDES`	91.67%	100%	95.00%	90.00%	100%	91.25%	100%	100%
`ARWIN`	85.00%	82.50%	97.50%	100%	98.75%	95.63%	99.38%	94.06%
`CASA`	77.78%	77.78%	98.93%	95.71%	100%	97.86%	97.22%	96.43%
`DJZAM`	100%	100%	95.00%	100%	100%	100%	85.00%	100%
`DJZUL`	90.00%	100%	92.50%	85.00%	80.00%	95.00%	90.00%	90.00%
`KIS`	75.00%	33.33%	91.25%	100%	100%	90.00%	89.88%	100%
`LRSIDE`	100%	95.45%	90.00%	96.00%	92.00%	90.00%	95.64%	90.00%
`PRODO`	66.67%	76.19%	100%	95.63%	95.63%	100%	95.63%	97.50%
`PRODT`	76.19%	61.90%	90.63%	95.00%	95.00%	97.50%	100%	100%
`SLTNP`	92.86%	92.86%	86.25%	95.00%	95.00%	90.00%	100%	100%
`SYM`	50.00%	70.00%	100%	85.00%	85.00%	100%	95.00%	100%
Average	82.29%	80.91%	93.88%	94.70%	95.07%	94.81%	95.25%	97.09%
Variance	0.0229	0.0422	0.0017	0.0032	0.0047	0.0015	0.0025	0.0016

Table 4.4: Defect detection results using GLCM and the BPNN classifier.

to obtain promising results. *Log*-scale distances (e.g. $d = \{1, 6, 36\}$), and four functions: Maximum, Energy, Correlation and Inertia, provided the best results in our experiments. An appropriate *log*-scale distance set represents the correlation information of pixels in short, medium, and long distances (in terms of the image size) all together. The put-one-aside function selection method, allows us to find the optimum subset of GLCM functions. In our experiments, a combination of these optimisations was effective in improving the GLCM performance. The most important disadvantages of GLCM are their relatively high computation costs (time comparison will be presented later in Section 4.6) and its not very promising KNN performance ($\mathrm{CA}_{\mathrm{KNN}}$=81.08%).

4.3.4 Gabor Filters

We wish to investigate whether it is possible to implement an efficient Gabor-based MSMD algorithm for texture abnormality detection. Many studies have reported high performance of Gabor filters in texture classification. Gabor filters also illustrated promising results in our texture classification experiments in Chapter 3. In defect detection tests, we again set the directional resolution to $\Delta\theta = 45^\circ$, and selected one octave (dyadic) central frequencies and half power bandwidth. The first Gabor feature set used, was extracted from a 12-filter bank in lower central frequencies of $\{\frac{\Omega_M}{16}, \frac{\Omega_M}{8}, \frac{\Omega_M}{4}\}$. Ω_M was the maximum frequency in spatial frequency domain (see Section 2.3.2 for details). 12 mean filter responses built the first feature vector F^1_{Gabor}:

$$F^1_{Gabor} = \{\mu(G_{\omega,\theta})\} \ , \quad \omega \in \{\frac{\Omega_M}{16}, \frac{\Omega_M}{8}, \frac{\Omega_M}{4}\} \ , \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \tag{4.8}$$

As Table 4.5 shows, the F^1_{Gabor} performance was not very high and limited to 65.17% for the KNN classifier and 70.46% for BPNN on average. By involving higher central frequencies for further experiments, we then formed the second feature vector, F^2_{Gabor}, comprising filter responses at $\{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\}$, and a third feature vector, F^3_{Gabor}, with central frequencies $\{\frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \Omega_M\}$.

$$\begin{cases} F^2_{Gabor} = \{\mu(G_{\omega,\theta})\} \ , \quad \omega \in \{\frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}\} \ , \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \\ F^3_{Gabor} = \{\mu(G_{\omega,\theta})\} \ , \quad \omega \in \{\frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \Omega_M\} \ , \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \end{cases} \tag{4.9}$$

The classification performance then increased to $\mathrm{CA_{KNN}}$=82.10% and $\mathrm{CA_{BPNN}}$=88.57% for F^2_{Gabor} and 84.39% and 89.21% for F^3_{Gabor}. Results obtained from other similar central frequency arrangements, such as $\{\frac{3\Omega_M}{16}, \frac{3\Omega_M}{8}, \frac{3\Omega_M}{4}\}$ or even non-dyadic setting $\{\frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \frac{3\Omega_M}{4}\}$, were lower than F^3_{Gabor}.

Next, further accuracy was obtained by involving more filter response statistics in F^4_{Gabor}, which contained mean, standard deviation and maximum value of filter responses (36 features):

$$F^4_{Gabor} = \{\mu(G_{\omega,\theta}),\ \sigma(G_{\omega,\theta}),\ max(G_{\omega,\theta})\}\ ,\ \ \omega \in \{\frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \Omega_M\}\ ,\ \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \tag{4.10}$$

F^4_{Gabor} provided the best classification accuracy of 84.87% with KNN and 91.22% with BPNN, (i.e. respectively 0.47% and 2.01% better than F^3_{Gabor}). Table 4.5 illustrates the overall performance of our Gabor filters in tile defect detection.

Using either more frequency bands (e.g. 5 bands $\{\frac{\Omega_M}{16}, \frac{\Omega_M}{8}, \frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \frac{3\Omega_M}{4}\}$, or smaller directional definition (e.g. $\Delta\theta = 30^\circ$ or 22.5°) or more statistics (e.g. skewness and kurtosis), did not increase the overall CA. Although for some types of tiles using 5 frequency bands increased the accuracy (e.g. almost 5% more accuracy on KNN on type `ARDES`), in general we either must employ the better three central frequencies of F^3_{Gabor} and F^4_{Gabor}, or apply a type-by-type frequency optimisation scheme.

Results showed that in general having three rows of filters in the low, middle and high frequencies were essential and sufficient for successful defect detection, and in contrast to some other applications higher frequency filters were of importance (see Table 4.5 and compare F^1_{Gabor} with for instance F^2_{Gabor}). Optimisations successfully improved the Gabor filter performance and added almost 20% to its classification accuracy on both classifiers (i.e. from 65.17% to 84.87% on KNN and from 70.46% to 91.22% on BPNN). Also, we illustrated that finer directional definition or higher order moments did not necessarily increase the performance of a Gabor-based texture defect detector. This suggested that applied directions, central frequencies, and statistics could sufficiently cover the features of normal and abnormal tiles. Gabor filters accuracy on KNN is 3.79% better than GLCM, on the BPNN classifier however, it is on average 5.87% less accurate than GLCM.

	CA$_{\text{KNN}}$				CA$_{\text{BPNN}}$			
Tile Type	F^1_{Gabor}	F^2_{Gabor}	F^3_{Gabor}	F^4_{Gabor}	F^1_{Gabor}	F^2_{Gabor}	F^3_{Gabor}	F^4_{Gabor}
`ARDES`	89.58%	80.00%	80.00%	85.00%	83.33%	87.50%	90.00%	85.00%
`ARWIN`	45.00%	85.00%	86.25%	85.00%	48.13%	97.81%	91.25%	98.75%
`CASA`	58.33%	98.57%	98.57%	98.57%	63.19%	92.86%	98.57%	97.14%
`DJZAM`	62.50%	70.00%	45.00%	75.00%	72.50%	62.50%	60.00%	75.00%
`DJZUL`	65.00%	60.00%	85.00%	65.00%	65.00%	60.00%	85.00%	85.00%
`KIS`	86.31%	100%	100%	100%	82.50%	73.75%	85.00%	100%
`LRSIDE`	71.78%	92.00%	96.00%	100%	65.44%	93.00%	94.00%	100%
`PRODO`	54.76%	95.00%	77.50%	95.00%	72.80%	100%	92.50%	85.00%
`PRODT`	57.44%	77.50%	95.00%	70.00%	57.26%	95.63%	90.00%	90.00%
`SLTNP`	38.69%	70.00%	85.00%	75.00%	53.87%	87.50%	90.00%	97.50%
`SYM`	87.50%	75.00%	80.00%	85.00%	90.00%	87.50%	90.00%	90.00%
Average	65.17%	82.10%	84.39%	84.87%	70.46%	88.57%	89.21%	91.22%
Variance	0.0291	0.0172	0.0233	0.0156	0.0197	0.0127	0.0112	0.0067

Table 4.5: Defect detection results using Gabor filters.

4.3.5 Directional Walsh-Hadamard Transform

The modified directional Walsh-Hadamard transform was the next feature extraction method tested. DWHT showed high speed and accuracy in outdoor scene and VisTex classification tests in Chapter 3. So it is also of interest to implement and evaluate the proposed DWHT method in tile defect detection. DWHT applies rotated (i.e. rearranged) input matrices and a 1D row-wise sequence analysis to encapsulate a MSMD texture analysis. Figure 4.3 depicts a `DJZAM` tile, its 45° rotated version, their 1D Hadamard transforms, and their average of transform matrices columns. Angular decomposition is completed by adding two other rotated input matrices (90 ° and 135°) and scale decomposition is carried out by individual analysis of different sequency bands (i.e. separated vertical bands at the transform matrix). Figures 4.3(c) and (d) show the separation of three and four individual sequency bands applied in these experiments. Statistics of these bands eventually build the DWHT feature vectors.

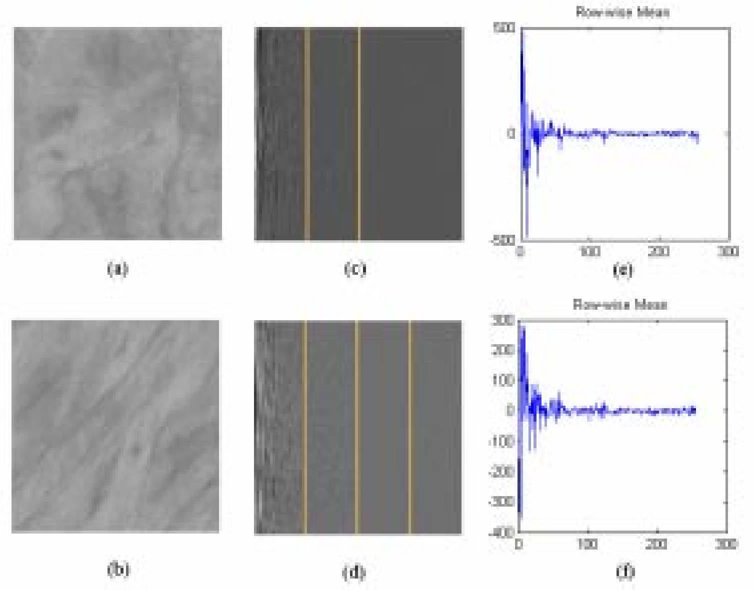


Figure 4.3: A `DJZAM` tile, (a), its 45° rearranged version, (b), their 1D Hadamard transforms, (c) and (d), and the average of transform matrices columns, (e) and (f). Three and four separated sequency bands are also shown in (c) and (d).

The first feature vector experimented with, F^1_{DWHT}, contained the mean values of 4 directions and 4 sequency bands (column-wise quarters of the transform matrix). Next, we decreased the sequency bands into three (Figure 4.3(c)) but concurrently added maximum value and standard deviation of the sequency bands to the second feature vector, F^2_{DWHT}, to see which one was more effective on the defect detection performance. The third feature vector F^3_{DWHT} comprised 4 sequency bands and 3 statistics all together.

$$\begin{cases} F^1_{DWHT} = \{\mu(H_{b,\theta})\ ,\ \ b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-\frac{3}{4}, \frac{3}{4}-1\} \\ F^2_{DWHT} = \{\mu(H_{b,\theta})\ ,\ \sigma(H_{b,\theta})\ ,\ Max(H_{b,\theta})\}\ ,\ \ b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-1\} \\ F^3_{DWHT} = \{\mu(H_{b,\theta})\ ,\ \sigma(H_{b,\theta})\ ,\ Max(H_{b,\theta})\}\ ,\ \ b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-\frac{3}{4}, \frac{3}{4}-1\} \end{cases} \tag{4.11}$$

For all DWHT feature vectors, $\theta \in \{0°, 45°, 90°, 135°\}$. The number of features was 16 for

F^1_{DWHT}, 36 for F^2_{DWHT}, and 48 for F^3_{DWHT}. As Table 4.6 illustrates, the defect detection performance of F^1_{DWHT} is 74.58% and 89.15% on KNN and BPNN. Involving 2 more statistics (σ and Max), added 1.55% to $\mathrm{CA_{KNN}}$ and only 0.32% to $\mathrm{CA_{BPNN}}$. The best DWHT performance was obtained by F^3_{DWHT} which outperformed F^2_{DWHT} at 82.62% and 95.58% accuracy for KNN and BPNN. It suggests that an individual higher sequency band and more statistics are both necessary for higher performance DWHT-based defect detection. Compared to the best Gabor filter results, DWHT performance was slightly lower on the KNN (82.62% vs. 84.87%), however it was considerably higher on the BPNN classifier (95.58% vs. 91.22%). DWHT performance was also 1.51% lower than GLCM on BPNN, but 1.54% higher on the KNN classifier.

	$\mathrm{CA_{KNN}}$			$\mathrm{CA_{BPNN}}$		
Tile Type	F^1_{DWHT}	F^2_{DWHT}	F^3_{DWHT}	F^1_{DWHT}	F^2_{DWHT}	F^3_{DWHT}
`ARDES`	87.50%	87.50%	85.00%	91.15%	93.42%	97.50%
`ARWIN`	78.13%	79.38%	86.25%	99.69%	99.84%	100%
`CASA`	86.81%	92.36%	88.57%	98.09%	97.57%	99.64%
`DJZAM`	85.00%	85.00%	85.00%	82.50%	88.57%	87.50%
`DJZUL`	50.00%	50.00%	55.00%	69.38%	73.75%	87.50%
`KIS`	67.56%	63.99%	100%	95.61%	93.68%	93.75%
`LRSIDE`	90.06%	88.92%	94.00%	94.88%	91.12%	98.00%
`PRODO`	79.23%	81.61%	90.00%	88.04%	91.73%	100%
`PRODT`	65.00%	61.01%	70.00%	79.87%	87.50%	87.50%
`SLTNP`	71.13%	85.12%	90.00%	96.43%	79.54%	100%
`SYM`	60.00%	62.50%	65.00%	85.00%	87.50%	100%
Average	74.58%	76.13%	82.62%	89.15%	89.47%	95.58%
Variance	0.0167	0.0200	0.0183	0.0086	0.0057	0.0030

Table 4.6: Defect detection results using DWHT features.

4.3.6 Discrete Cosine Transform

DCT is a commonly used transform in image and texture processing, most seen in image compression and coding applications [125]. Its characteristics are rather similar with the FFT (it also uses sinusoidal kernels), and Hadamard (it also is a real transform). There were two goals in our DCT experiments. Firstly, to measure its texture defect detection performance, and secondly, and more importantly, to copy the DWHT idea in a directional DCT, called DDCT, and compare its performance with the ordinary DCT. A 2-dimensional DCT is defined as:

$$C(A) = \frac{2}{\sqrt{MN}} T(u) T(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} A(x,y) cos(\frac{(2x+1)u\pi}{2M}) cos(\frac{(2y+1)v\pi}{2N}) \tag{4.12}$$

where A is a $M \times N$ image, and

$$T(x) = \begin{cases} \frac{1}{\sqrt{2}} & if\ x = 0 \\ 1 & otherwise \end{cases} \tag{4.13}$$

The DCT can also be represented in matrix form. Figure 4.4 illustrates a 64 × 64 DCT matrix. Although smoother, it is comparable with the Hadamard transform matrix of Figure 3.8(b).

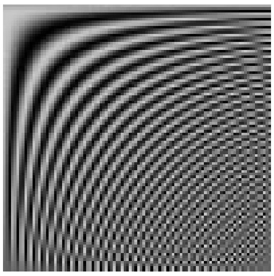


Figure 4.4: A 64×64 DCT matrix. Although smoother, it is comparable with the 64×64 SOH matrix in Figure 3.8(b), Section 3.3.2.

DCT features were statistics of the transform matrix. The first DCT feature vector, F^1_{DCT}, contained the maximum, mean and standard deviation of $C(A)$ (3 features). Its performance was relatively low and limited to 62.51% with the KNN and 73.27% with the BPNN classifier (see Table 4.7). Next we tried a multi-band scheme on the DCT transform matrix (no rotation). This F^2_{DCT} feature vector comprised three statistics of four separated bands:

$$F^2_{DCT} = \{\mu(C_b)\ ,\ \sigma(C_b)\ ,\ max(C_b)\}\ ,\quad b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-\frac{3}{4}, \frac{3}{4}-1\} \tag{4.14}$$

The multi-band, 12-feature F^2_{DCT} showed better accuracy and achieved 77.48% and 89.84% correct classification respectively with KNN and BPNN.

We repeated the DWHT procedure after replacing the Hadamard matrix with a DCT one, resulting in a directional DCT. A DDCT feature vector, F^1_{DDCT}, was extracted next with three statistics, four bands, and four directions (48 features in total).

$$F^1_{DDCT} = \{\mu(C_{b,\theta}),\, \sigma(C_{b,\theta}),\, max(C_{b,\theta})\}\ ,\ \ b \in \{0-\frac{1}{4}, \frac{1}{4}-\frac{1}{2}, \frac{1}{2}-\frac{3}{4}, \frac{3}{4}-1\}\ ,\ \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \tag{4.15}$$

F^1_{DDCT} performed better than both DCT feature vectors, with 82.15% correct classification for the KNN and 95.14% for the BPNN classifier. The 4.67% and 5.30% respective difference between DCT and DDCT performances showed the advantage of DDCT in texture abnormality detection (see Table 4.7). Meanwhile, DDCT outcomes are quite close, although marginally behind the DWHTs. In fact DWHT is 0.47% and 0.44% better than DDCT on the KNN and the BPNN classifiers. Given the basic similarity between DCT and WHT, such close performances are expected. However, it also suggests that in comparison to WHT, in larger size transform domain processes, DCT loses its advantages, whereas reportedly it outperforms WHT in spatial domain texture processing using smaller size filters (see Section 2.3.2 and also [118]).

4.3.7 Eigenfiltering

As mentioned before, in texture processing the first order statistics (e.g. histograms) are not sufficient, but second (or higher) order statistics are necessary. Basically, a texture may be processed either in the spatial domain, where, for instance, co-occurrence matrices can express pixel relations, or in the spatial frequency domain, where, for instance, directional wedge-ring or Gabor filters can be employed. The question is however, which method would be the optimal one.

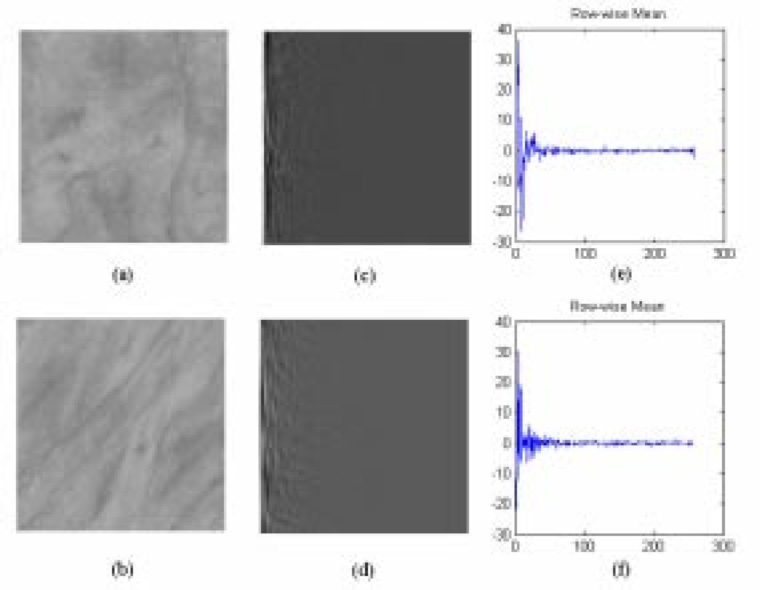


Figure 4.5: A `DJZAM` tile (a), its 45° rearranged version (b), their 1D DCT transforms ((c) and (d)), and average of transform matrices columns ((e) and (f))

Generally, if a local approach instead of a global approach is of interest, and 1^{st} and 2^{nd} order statistics are sufficient for modelling a texture, then co-occurrence matrices will be powerful analytical tools for that texture. However, this method is computationally expensive. So, if we are looking for a more condensed but still powerful representation of pixel neighbourhood analysis, PCA and eigenfilters are a good alternative [2, 33].

Using $n \times n$ sliding windows, $n \times n$ local neighbourhoods of the image can be extracted and rearranged as different observations of data into a $k \times n^2$ matrix (k is the number of sliding windows). The *neighbourhood size* is typically in the range $n = 3, 5, 7$. The covariance matrix is then computed and the eigenvectors and eigenvalues are obtained:

$$C(x) = E[(x - \mu_x)(x - \mu_x)^T] \tag{4.16}$$

$$(C(x) - \lambda_x I)e = 0 \tag{4.17}$$

where, μ_x is the mean value, I is the unity matrix, λ is the eigenvalue and e is the eigenvector

	$\mathrm{CA_{KNN}}$		$\mathrm{CA_{BPNN}}$	
Tile Type	F^2_{DCT}	F^1_{DDCT}	F^2_{DCT}	F^1_{DDCT}
`ARDES`	50.00%	85.00%	80.00%	90.00%
`ARWIN`	88.75%	91.25%	100%	100%
`CASA`	80.00%	77.14%	95.71%	98.57%
`DJZAM`	60.00%	85.00%	75.00%	95.00%
`DJZUL`	65.00%	50.00%	85.00%	85.00%
`KIS`	100%	100%	100%	100%
`LRSIDE`	96.00%	96.00%	100%	98.00%
`PRODO`	72.50%	77.50%	92.50%	95.00%
`PRODT`	80.00%	77.50%	90.00%	95.00%
`SLTNP`	90.00%	75.00%	75.00%	90.00%
`SYM`	70.00%	80.00%	95.00%	100%
Average	77.48%	82.15%	89.84%	95.14%
Variance	0.0244	0.0214	0.0094	0.0025

Table 4.7: Defect detection results using DCT and DDCT.

matrix. A $n \times n$ rearrangement of the eigenvectors could be interpreted as a bank of adapted filters of the same size, which optimally cover all $n \times n$ relations of the test image pixels. Detail images can be obtained by 2D spatial domain convolution of the test image by the members of the eigenfilter bank:

$$D_i^A = A \otimes F_i^A \quad , \quad 1 \leq i \leq n^2 \tag{4.18}$$

where A is the input image and F_i^A and D_i^A are its i^{th} eigenfilter and detail image. So, the process provides n^2 detail images of each input [2, 98]. (See Figure 4.7). Assuming a $n \times n$ neighbourhood of an image as different observations of adjacent data, Ade [2] shows that a $n \times n$ neighbourhood covariance is very similar to Haralick's Correlation (f_4) function:

$$f_4 = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)\Phi_{d,\theta}(i,j)}{\sigma_i \sigma_j} \tag{4.19}$$

where μ_i and μ_j are means of the rows and columns, and σ_i and σ_j are the standard deviations of the rows and columns. This is applied on an ordinary co-occurrence matrix of the corresponding distances, $\Phi_{d,\theta}$ [44]. In fact a $n \times n$ neighbourhood covariance covers all $2 \leq d \leq n$ distances. As Figure 4.6 exhibits, a 3×3 neighbourhood admits 13 different spatial relationships between a pixel pair. All of those relationships are reflected in eigenfilters. Therefore, together all n^2 eigenfilters convey the proper structure of the original texture with respect to a $n \times n$ basic structure. Ordered by their eigenvalues, Figure 4.7(b) exhibits 3×3 eigenfilters of a given tile, and show that apart from the first filter which is a simple ‘weighted average’, the eigenfilter bank comprises mostly of gradient filters which are not rotation invariant.

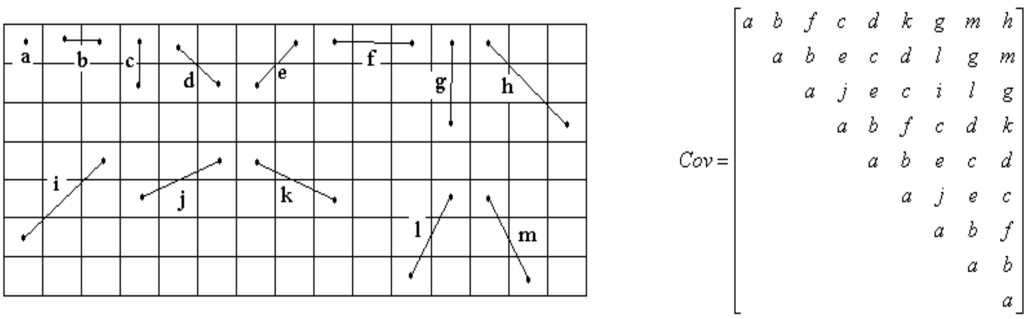


Figure 4.6: Possible relations between pixel pairs in 3x3 patches (left) and the covariance matrix (right). (From [2]).

The filters’ orthogonality is of importance as well, because it builds up uncorrelated detail images (i.e. filter outputs) ordered by their role in building the original image (i.e. eigenvalues). These together describe and can reconstruct the image studied. Next, we need to use those filter outputs as features of the texture, either by measuring certain statistics (typically variance, which can be assumed as a normalised energy measure) and carry out some typical classification, or apply reconstruction/synthesis processes. If results are not satisfactory, we move towards larger neighbourhood sizes (N) or try different window shapes (e.g. cross, diamond, stripe, etc)[44]. In Figure 4.7, the relation between an eigenfilter and its detail images (Figure 4.7(b) and (c)), and the coverage of the major part of the image information by the first eigenfilters are notable.

We started the experiments with the smaller neighbourhood size N =3 × 3. The first eigenfilter feature vector, F_{EF}^1, contained mean values of 9 detail images. The classification accuracy was $\mathrm{CA}_{\mathrm{KNN}}$=77.59% and $\mathrm{CA}_{\mathrm{BPNN}}$=91.63% which was considerable for 9 features. Then the standard deviation $\sigma(D_i)$ was inserted to the feature vector. So, F_{EF}^2 included 18 features per image.

$$\begin{cases} F_{EF}^1 = \{\mu(D_i)\}\ ,\ i \in \{1 \cdots 9\}\ ,\ N = 3 \times 3 \\ F_{EF}^2 = \{\mu(D_i), \sigma(D_i)\}\ ,\ i \in \{1 \cdots 9\}\ ,\ N = 3 \times 3 \end{cases} \tag{4.20}$$

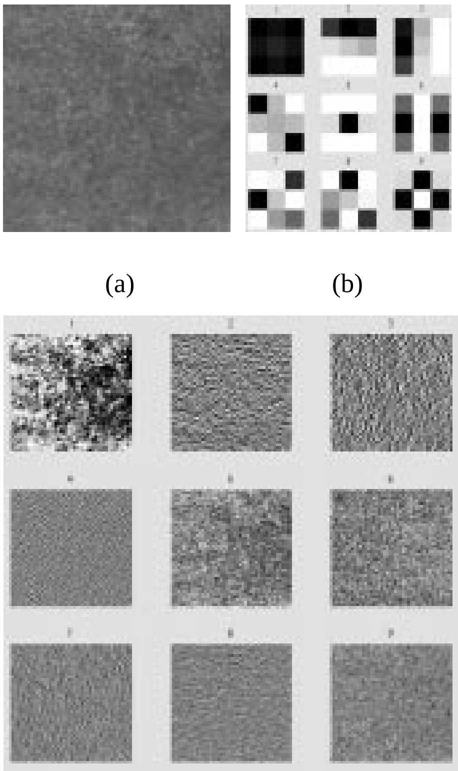
The classification performance using F_{EF}^2 increased to $\mathrm{CA}_{\mathrm{KNN}}$=80.87% and $\mathrm{CA}_{\mathrm{BPNN}}$=94.21% respectively. Adding the maximum, the minimum, or higher order moments did not improve the performance further.

Larger neighbourhood sizes were tried next, where the relations between farther pixels were also captured, hence improved detection rates were expected. The first series of experiments contained N =5 × 5, 7 × 7, and 9 × 9. The results suggested that the maximum accuracies of $\mathrm{CA}_{\mathrm{KNN}}$=85.30% and $\mathrm{CA}_{\mathrm{BPNN}}$=95.7% were provided by N =7 × 7. Table 4.8 represents these tests results.

$$\begin{cases} F_{EF}^3 = \{\mu(D_i), \sigma(D_i)\}\ ,\ i \in \{1 \cdots 25\}\ ,\ N = 5 \times 5 \\ F_{EF}^4 = \{\mu(D_i), \sigma(D_i)\}\ ,\ i \in \{1 \cdots 49\}\ ,\ N = 7 \times 7 \\ F_{EF}^5 = \{\mu(D_i), \sigma(D_i)\}\ ,\ i \in \{1 \cdots 81\}\ ,\ N = 9 \times 9 \end{cases} \tag{4.21}$$

We found that the results peaked at 7 × 7 and continuously decreased for sizes greater than that. This can be reasoned by the lack of inter-pixel correlation in farther distances.

Also, another problem can be the ordering of filters. We typically rely on the eigenvalues for ordering and matching the filters and detail images. It will be shown later in Chapter 5 that this can be problematic, in particular for larger neighbourhood sizes. If we compare eigenfiltering and GLCM results, we will see that the best GLCM performance is achieved by the distance



(a) (b)

(c)

Figure 4.7: Filtering procedure for an ARDES tile (a), its 3×3 eigenfilters (b) and detail images (c). To increase the visibility, all filters and detail images have been equalised.

set $d = \{1, 6, 36\}$. The first two distances, $1, 6$ are covered by 7×7 eigenfilters, the farther distance at 36 however, is very difficult to be covered. Even a quite smaller N $=13 \times 13$ would be impractical, since handling of $13 \times 13 \times 2 = 338$ features is not easy and the computing time of 13×13 eigenfiltering procedure is also almost 78 times longer than 3×3. Although on average the 98-feature F_{EF}^{4} is the most accurate feature vector, the optimum N is different from type to type. For instance, considering both classifiers, for `LRSIDE` the 5×5 eigenfilter is the most accurate one, while for `ARWIN` 3×3 is the best. Therefore, if it is practical, filter size optimisation for each tile type can increase the defect detection performance.

Moreover, it was attempted to find out which members of the eigenfilter bank were the most

	$\mathrm{CA}_{\mathrm{KNN}}$				$\mathrm{CA}_{\mathrm{BPNN}}$			
Tile Type	F_{EF}^{2}	F_{EF}^{3}	F_{EF}^{4}	F_{EF}^{5}	F_{EF}^{2}	F_{EF}^{3}	F_{EF}^{4}	F_{EF}^{5}
ARDES	70.00%	65.00%	80.00%	80.00%	100%	100%	98.75%	90.00%
ARWIN	98.75%	96.25%	96.25%	96.25%	96.88%	86.88%	96.56%	97.50%
CASA	74.29%	82.86%	98.57%	82.86%	97.50%	96.43%	97.50%	99.64%
DJZAM	85.00%	75.00%	80.00%	85.00%	100%	80.00%	93.75%	91.25%
DJZUL	55.00%	70.00%	80.00%	70.00%	82.50%	100%	96.25%	100%
KIS	100%	90.00%	95.00%	90.00%	100%	87.50%	95.00%	93.75%
LRSIDE	94.00%	98.00%	96.00%	94.00%	100%	100%	93.00%	86.50%
PRODO	85.00%	95.00%	100%	90.00%	92.50%	100%	93.75%	99.38%
PRODT	77.50%	85.00%	82.50%	77.50%	90.63%	88.75%	98.13%	99.38%
SLTNP	75.00%	90.00%	60.00%	75.00%	100%	100%	90.00%	87.50%
SYM	75.00%	85.00%	70.00%	95.00%	76.25%	100%	100%	82.50%
Average	80.87%	84.74%	85.30%	85.06%	94.21%	94.51%	95.70%	93.40%
Variance	0.0180	0.0117	0.0169	0.0077	0.0066	0.0053	0.0009	0.0039

Table 4.8: Defect detection results using eigenfilters. N =3 × 3 to N =9 × 9 filter response statistics were used as features.

effective in the abnormality detection process. Table 4.9 represents the filter-wise classification performance of 3 × 3 eigenfilter bank on the KNN classifier. This table clearly shows that on average the first gradient filters (here: F2, F3, F4) were the most effective. The reason can be the dominant role of these filters in texture determination. They convey the most important characteristics of a texture, compared to the low pass filter (F1) or the last gradient filters (i.e. bar detectors, edge detectors and so on) with smaller eigenvalues (F5 onward). A similar test on N =5 × 5 filters also illustrated that the most effective filters are F2 and F4, however the difference between their performance was less than the difference between 3 × 3 filters' performances.

Tile Type	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**
`ARDES`	60.00%	90.00%	60.00%	75.00%	40.00%	50.00%	90.00%	80.00%	70.00%
`ARWIN`	80.00%	77.50%	78.75%	98.75%	72.50%	62.50%	71.25%	47.50%	65.00%
`CASA`	34.29%	62.86%	82.86%	64.29%	48.57%	48.57%	71.43%	74.29%	61.43%
`DJZAM`	75.00%	60.00%	75.00%	40.00%	75.00%	45.00%	70.00%	55.00%	80.00%
`DJZUL`	55.00%	50.00%	75.00%	75.00%	45.00%	50.00%	75.00%	60.00%	60.00%
`KIS`	40.00%	100%	50.00%	95.00%	100%	60.00%	50.00%	30.00%	60.00%
`LRSIDE`	72.00%	100%	80.00%	98.00%	98.00%	92.00%	40.00%	50.00%	58.00%
`PRODO`	77.50%	92.50%	80.00%	80.00%	70.00%	92.50%	85.00%	77.50%	97.50%
`PRODT`	65.00%	85.00%	65.00%	92.50%	50.00%	70.00%	70.00%	70.00%	75.00%
`SLTNP`	80.00%	70.00%	60.00%	70.00%	60.00%	45.00%	75.00%	55.00%	50.00%
`SYM`	30.00%	80.00%	65.00%	80.00%	70.00%	50.00%	80.00%	75.00%	80.00%
Average	63.88%	78.79%	70.66%	78.85%	65.9%	61.557%	69.77%	59.93%	67.69%
Variance	0.0348	0.0277	0.0114	0.0304	0.0405	0.0306	0.0207	0.0242	0.0180

Table 4.9: Comparison between different filters performance for a 3 × 3 eigenfilter bank.

4.4 Gabor Composition Method

Up to this stage, a few of the applied algorithms have performed well on the BPNN-based defect detection tests (e.g. GLCM with $\mathrm{CA}_{\mathrm{BPNN}}$=97.09%). The KNN performance meanwhile was limited to at most $\mathrm{CA}_{\mathrm{KNN}}$=85.30% for the 7 × 7 eigenfilter feature vector. Given that KNN is more practical than BPNN in some industrial applications due to its much simpler training phase, we propose a new and advanced feature extraction method mainly to increase the KNN performance. This method, called *Gabor Composition* (GC), is based on a combination of Gabor filtering and GLCM.

Major studies toward computing texture components (or more general, image components) can be divided into either discriminative or generative models. In discriminative models, a pyramid of filters with various scales and orientations is convolved with the original image, then for each pixel a feature vector of filter responses is configured. If there are some distinguishable repeated 'structures' in the image, it will be justifiable to believe the pixels' feature vectors must form clusters. Then, after finding the cluster centers via clustering (e.g. using k-means or fuzzy c-means), image icons will be generatable by a pseudo-inverse transform applied on the centers. In generative models, an over-complete dictionary of local image bases and their transforms is built up. Then each image would be 'generated' by linear superposition of some bases (i.e. detail images) selected from the dictionary [123, 131].

Although both approaches can determine texture structures, they seem essentially different. In a generative model, the image is reconstructed by superposition of a number of detail images, where the number of those detail images is usually tens of times smaller than the number of pixels. In contrast in discriminative models, pixels are represented by feature vectors and the number of features is usually tens of times larger than the number of pixels [131]. For example, DeBonet's flexible histogram is a discriminative model which replaces a pixel with its relatively large 'parent structure' vector [11, 12].

4.4.1 The Method

The proposed GC method, attempts to build up a generative model of texture using a Gabor decomposition/composition approach. Figure 4.8 illustrates the different stages of a general Gabor decomposition-based image feature extraction procedure. Firstly, the input image is decomposed into detail images using an appropriate Gabor filter bank. Then we can either use detail images' features as the texture descriptors, or re-compose them to a new feature map using a composition function Γ. In the first method, spectral histogram, detail images GLCMs, or moments (as we carried out in previous Gabor filtering experiments) of the detail images can be utilised. In the second method, the composition function, Γ, should be able to properly amplify the desired attributes of the input image in the feature map. In this application, these desired attributes are possible abnormalities. Γ also should match the final analysis selected.

To analyse the feature map, we may compute its statistics, extract its histogram or, as we propose here, apply a GLCM process on it. In particular, when the input image is a texture, the feature map will eventually show texture appearance too, therefore in such cases the texture-oriented GLCM is a suitable approach. The promising defect detection performance of Gabor filters (see Section 4.3.4), suggests that they can effectively decompose the texture energy (i.e. information) into various frequency scales and directions (i.e. detail images). Then a proper function (e.g. a quantisation function), is applied to reduce the redundancy and consequently dimensionality of data in detail images. In the GC algorithm, Gabor filtering and quantisation together reduce the data dimensionality. If this dimensionality reduction succeeds to dampen the irrelevant and redundant information, e.g. the background texture, as well as to amplify desired information, e.g. defects, the performance of GLCM as the final feature extractor improves. Consequently a highly accurate classification is attainable.

The proposed GC algorithm traces the following steps to complete a texture feature extraction procedure:

- **Pre-Gabor-processing:** As before, only a simple $x_p = x - \mu_x$ is applied to compensate possible luminance changes and increase the textural aspects of experiment.
- **Gabor filtering:** The Gabor filter bank which provided the best classification results in previous tests, F^4_{Gabor} (see Section 4.3.4), is used again. This filter bank consists of twelve filters in three frequency bands and four directions:

$$G_{\omega,\theta}\ ,\ \ \omega \in \{\frac{\Omega_M}{4}, \frac{\Omega_M}{2}, \Omega_M\}\ ,\ \ \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \tag{4.22}$$

- **Post-Gabor-processing:** We do not utilise typical post-processing steps such as blob-detection [54, 103]. Instead, a quantisation algorithm is used which quantises a detail

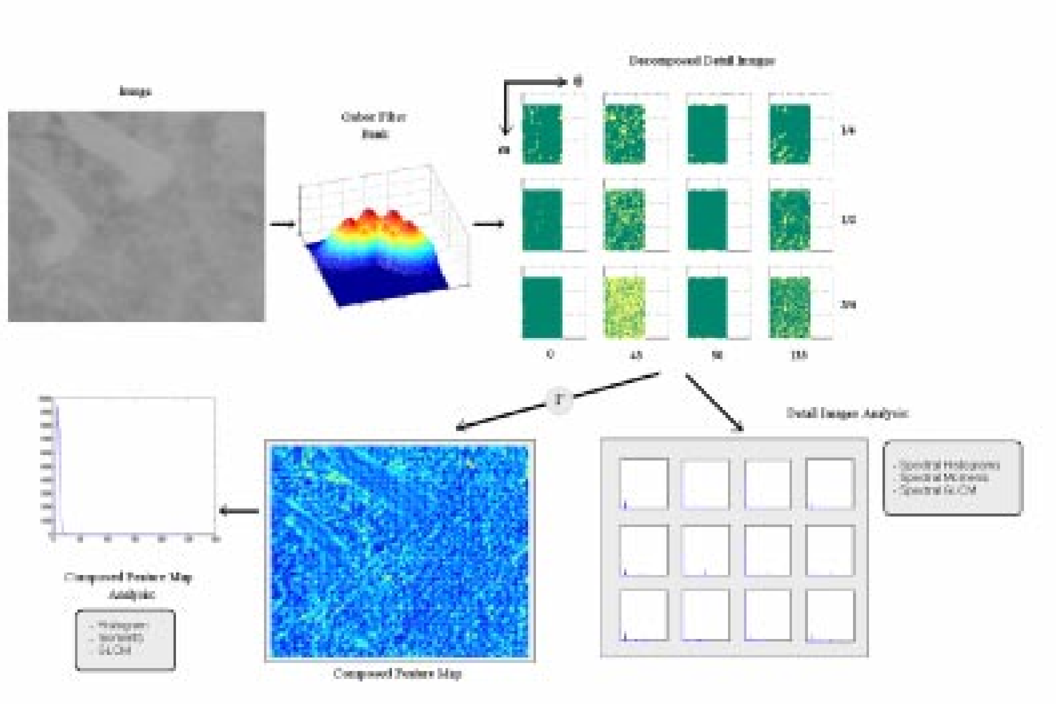


Figure 4.8: Gabor-based decomposition/composition procedure.

image into a few (e.g. 2 to 4) different levels. Blob detection is useful as a non-linear operator to highlight blobs in an image, whereas we employ GLCM analysis at the final stage. Quantisation decreases the number of grey levels in a detail image, hence the final composed feature map is in fact a Gabor-filtered version of the input, presented in less grey levels and GLCM then deals with fewer grey levels.

Before quantisation, a simple linear normalisation is carried out to bring the detail images magnitude into $[0,1]$ range:

$$x_p = \frac{x - min(x)}{max(x) - min(x)} \tag{4.23}$$

where x and x_p are the original and normalised images. Then a quantisation on n equidistant thresholds which generates a $(n+1)$ grey level detail image is implemented as:

$$\text{FOR} \;\; k \in \{0,1,\cdots,n-1\} \quad \text{IF} \;\; \frac{k}{n} \leq x(i,j) \leq \frac{k+1}{n} \quad \text{THEN} \;\; y(i,j) = \; k \;\; , \;\; \forall i,j \in \{1,\cdots,M\} \tag{4.24}$$

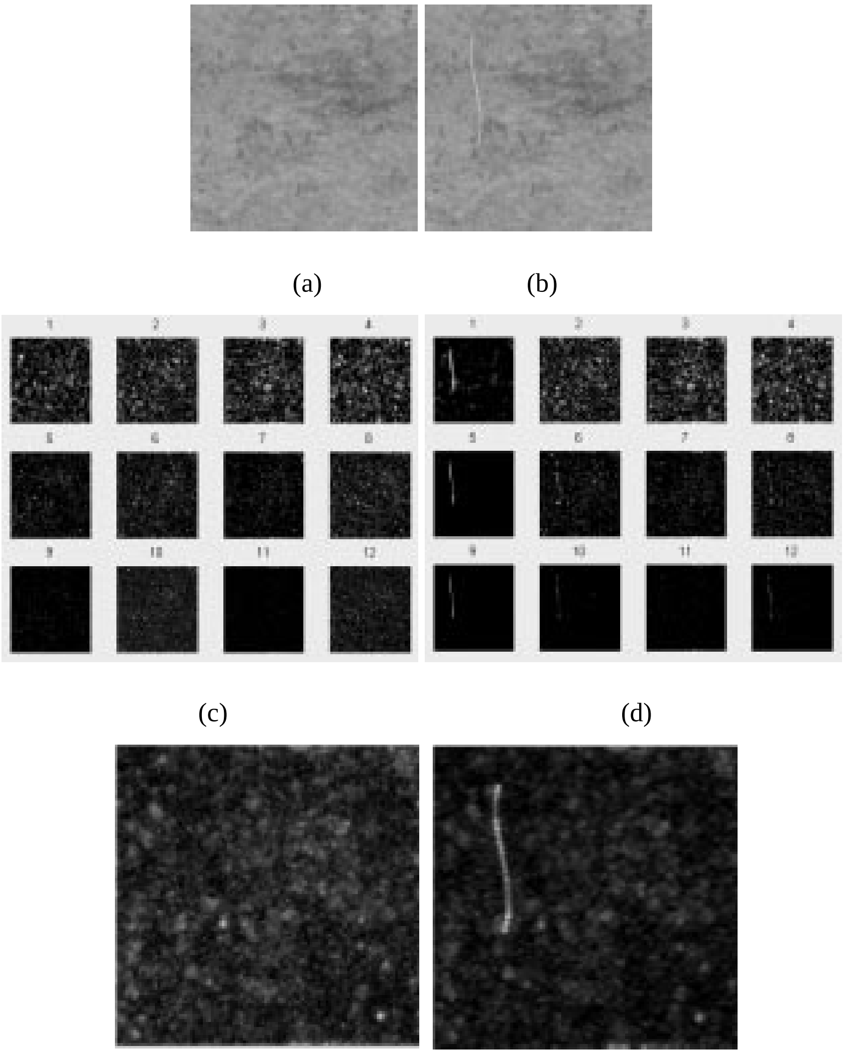
where x is the $M \times M$ normalised input image and y is the quantised $(n+1)$ grey level output and the operation is performed on all pixels of x.

- **Composition:** The easiest way to re-compose the quantised detail images and generate the feature map is simple addition. Meanwhile other fusion functions such as Bernoulli's rule of combination are also applicable [103]. Post-Gabor-processing and composition, together build our Γ function.
- **Feature extraction:** As the final stage, the composed feature map is fed to a GLCM algorithm to extract the texture's features. Basically the GLCM feature extraction applied is similar to that used previously in Section 4.3.3, however a different parameter set may be applicable.

4.4.2 Justification

Similar ideas have been proposed before in the defect detection literature. For example, Sari-Sarraf and Goddard [103] employed a wavelet-based method for on-loom fabric defect detection. They used a MSWAR wavelet for MSMD analysis of textured fabrics and generating detail images, and a Bernoulli function for fusion of appropriately conditioned detail images. Finally, they measured the local coarseness and global homogeneity of the test texture to detect defects. Their method is capable of defect localisation too, and showed promising accuracy and feasibility. In [71] Latif-Amet *et al* also studied a combination of wavelets and GLCM for fabric texture defect detection. Their method, called *sub-band domain co-occurrence matrices* (in contrast to typical spatial domain co-occurrence matrices), contained a wavelet transform on a texture, followed by applying four GLCM functions (namely Entropy, Contrast, Energy and IDM), on wavelet sub-bands (i.e. detail images). A Mahalanobis distance was then employed to measure the abnormality of the test texture.

Figure 4.9 illustrates a `KIS` tile and its artificially defected version (a thin bright vertical curve was added). Detail images of both textures and recomposed feature maps are also presented. The defective region is clearly highlighted in the less-redundant recomposed feature map of the defective tile in 4.9(f), which suggests the efficiency of the GC algorithm.



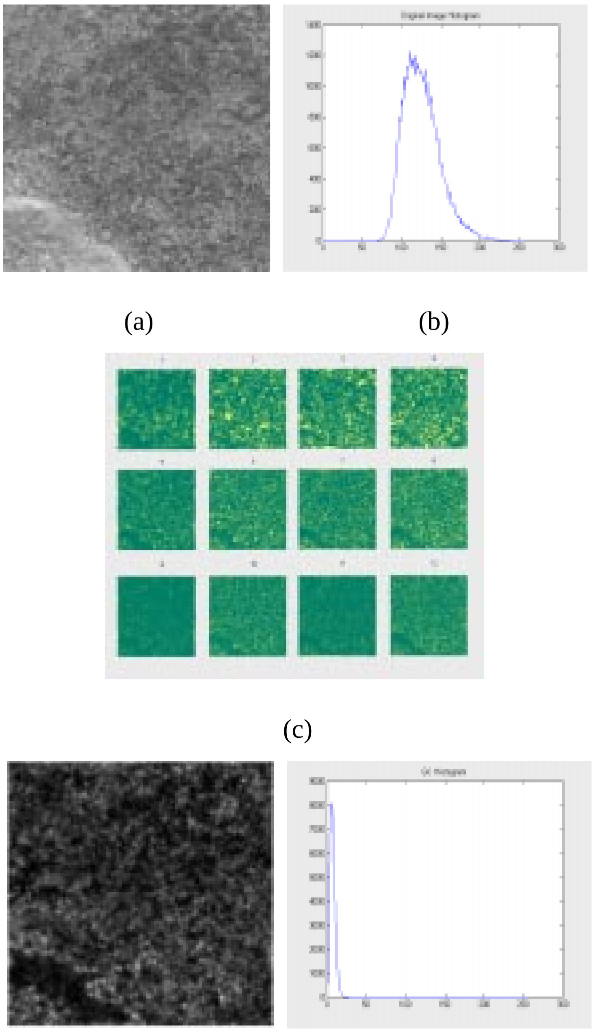
(a) (b)

(c) (d)

(e) (f)

Figure 4.9: A normal `KIS` tile (a), and its artificially defective version (b), their respective detail images, (c) and (d), and feature maps, (e) and (f). Note the highlighted defective region in (f).

Figures 4.10 and 4.11 depict the GC method concepts, applied on an `ARWIN` and a `SYM` tile. An original texture is filtered by a 12-filter Gabor filter bank resulting in corresponding detail images. Then the detail images are combined and the feature map is built. The feature map histogram (e) suggests a considerable reduction of grey levels compared to the original image histogram in (b), which means reduction of dimensionality in the feature space (e.g. smaller co-occurrence matrices). In both examples the defective area is clearly visible in the feature map which shows that along with detection of defects, GC is also capable of defect localisation.

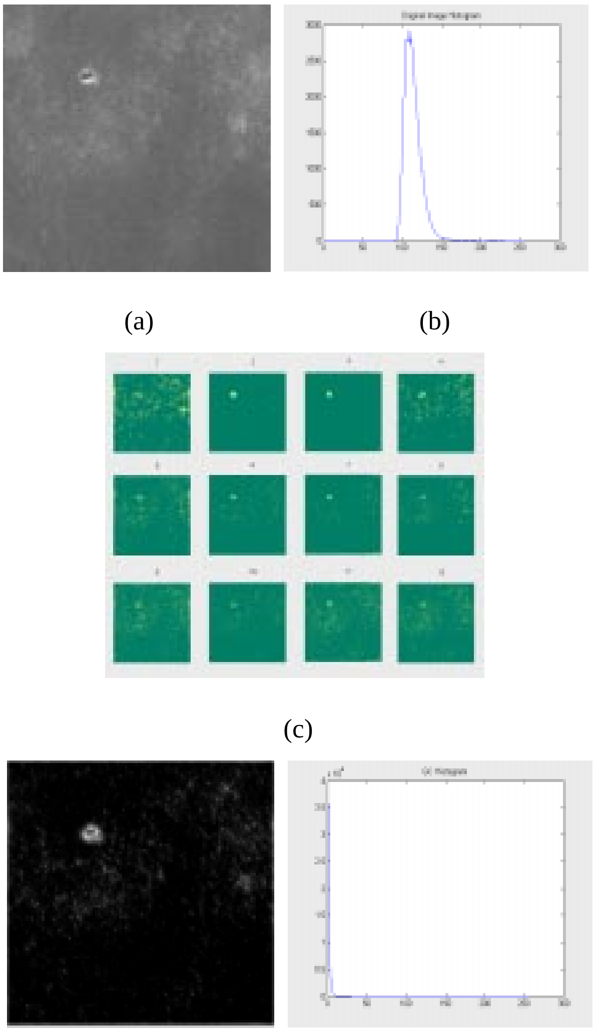


(a) (b)

(c)

(d) (e)

Figure 4.10: An `ARWIN` tile (a), with the histogram (b), detail images (c), generated feature map (d), and the feature map histogram (e).



(a) (b)

(c)

(d) (e)

Figure 4.11: A `SYM` tile (a), with the histogram (b), detail images (c), generated feature map (d), and the feature map histogram (e).

4.4.3 Experiments

For the first GC feature vector, F_{GC}^{1}, we applied twelve filters, two quantisation thresholds (QT) at QT= $\{0.33, 0.66\}$, and the simple composition-by-addition scheme. The result was a 25-grey level feature map. In fact, using a m-filter filter bank and n quantisation thresholds, resulted in a $(m \times n) + 1$ grey level feature map:

$$0 \leq y(i, j) \leq m \times n \tag{4.25}$$

where y is the feature map. Then four GLCM matrices for four directions ($\Delta\theta = 45^\circ$), and one distance ($d = 1$) were computed. Next, we employed those four GLCM functions which had provided the best classification accuracy in previous experiments (namely Maximum, Energy, Correlation and Inertia, see Section 4.3.3) and extracted

$$\theta \times d \times \text{ functions } = 4 \times 1 \times 4 = 16$$

features of F_{GC}^1.

The results however, were not promising at $\mathrm{CA}_{\mathrm{KNN}}$=76.10% and $\mathrm{CA}_{\mathrm{BPNN}}$=88.75%. These were obviously lower than the previous high performance methods (e.g. respectively 4.98% and 8.34% lower than GLCM). To improve the performance, we then tried more quantisation thresholds. There were four thresholds $QT = \{0.2, 0.4, 0.6, 0.8\}$, and hence five levels, in F_{GC}^2, which generated a 49-grey level feature map. Other parameters were kept similar to F_{GC}^1. The F_{GC}^2 performance was considerably improved at $\mathrm{CA}_{\mathrm{KNN}}$=80.79% and $\mathrm{CA}_{\mathrm{BPNN}}$=92.44%. Although clearly better than F_{GC}^1, it was still outperformed by the GLCM method in Section 4.3.3.

Next we attempted to discard some detail images from the composition procedure. Each texture has a certain energy distribution throughout its detail images after Gabor decomposition, depending on its frequency and directional characteristics. Again, a certain defect is traceable in some detail images more than others, also depends on its characteristics. Therefore a well prepared exclusion of a few detail images prior to the composition can decrease the redundancy, as well as highlight the defects further. To optimally select a subset of detail images, Latif-Amet *et al* discarded a detail image if its energy (i.e. $\sum_{u,v} D_i^2(u, v)$) was significantly lower than the maximum detail image energy of the same scale. Regarding the diminished energy of higher frequencies, it would not be reasonable to compare the energy of detail images of different scales [71]. Meanwhile, Sari-Sarraf and Goddard considered the background texture attenuation as the more important factor. Hence, if a given detail image had an energy decisively higher than the others (of the same scale) but almost equal to that of a corresponding detail image in a defect-free reference texture, it would be excluded from the composition process. They believed this rejected the detail images which were captured by the faultless background texture [103].

We, on the other hand, employ a different approach. The template subset of the TDS comprises of defect-free reference samples of all tile types and does not overlap the normal subset (see Table 4.1). For each type, the average of detail image energies is computed throughout the template and preserved, to be compared later with detail image energies of the test tile. In our algorithm, detail images with the closest energy to their counterpart template average are excluded from the composition process. A defect changes the energy of detail images, hence by eliminating detail images with minimum distance to the ‘average of good textures’, we increase the presence of defects, and reduce the background texture simultaneously. Using the average energy of templates is easier than finding an appropriate image in the template set as the reference. We will refer to this problem once again and discuss it in more detail in Chapter 5 as a part of our novelty detection study.

We added the detail image exclusion scheme to the algorithm to obtain the third GC feature vector F_{GC}^{3}. This performed better than former GC feature vectors and achieved 85.9% and 96.43% correct classification on KNN and BPNN respectively. This was a promising improvement of 5.11% and 3.99% higher than F_{GC}^{2}. The optimum number of detail images to be excluded was found by trial and error. We evaluated the performance of 1 to 6 detail images exclusion (out of 12) and realised that the optimum result on average was obtained by exclusion of 3 detail images (i.e. using 9 detail images in composition). Therefore the composed feature map contained $9 \times 4 + 1 = 37$ grey levels. Other parameters (e.g. thresholds, GLCM functions and so on) were kept similar to F_{GC}^{2}, so F_{GC}^{3} still had 16 features. Now, although the

$\mathrm{CA_{BPNN}}$ was marginally lower than the GLCM (just 0.66%), the GC method using F_{GC}^{3} was the best on the KNN classification with 0.6% higher accuracy. In fact, the GC algorithm using F_{GC}^{3} attributes is one of the best defect detection algorithms developed in this study so far.

Next, addition was replaced with Bernoulli's rule of combination (BRC) [130] as the composition function. The BRC is defined as:

$$S = 1 - (1 - x_1)(1 - x_2)(1 - x_3) \tag{4.26}$$

where x_1, x_2, and x_3 are three input signals and S is their Bernoulli combination. The equation is expandable to more inputs. Compared to addition, BRC almost follows one of the inputs, while others possess relatively low values, i.e. it roughly follows the maximum input. This can be useful in our GC abnormality detection method where defects (in particular, defective areas boundaries) usually generate high magnitudes in detail images. An ordinary addition implies averaging, thus can diminish the presence of these high-magnitude defect boundaries in the composed feature map [103, 130]. Figure 4.12 illustrates a simple representation of BRC applied on two linear signals $a = -x$ and $b = 0.5x + 0.3$. As the graph d shows, in contrary to addition c, BRC almost follows the major input signal and is minimum while inputs are equal. If we assume the larger parts of signals a and b as possible defective regions, summation c does not explicitly show that region of the signal b, while the BRC function, d, illustrates two separate maximum points for possibly defective regions of both a and b.

The next GC feature vector, F_{GC}^{4}, which used BRC, reached the highest GC performance of $\mathrm{CA_{KNN}}$=88.35% and $\mathrm{CA_{BPNN}}$=97.02%. It was in fact slightly lower than the best BPNN performance presented by GLCM (97.09%), however, it was the most promising KNN classification result that we obtained in our experiments, and was 3.05% better than the second best KNN classification performed by eigenfiltering. Further efforts to improve the accuracy were not successful. For instance, when in one test three GLCM distances $d = \{1, 6, 36\}$ were employed, the performance of 48-feature F_{GC}^{5} decreased to $\mathrm{CA_{KNN}}$=86.03% and $\mathrm{CA_{BPNN}}$=94.74% re-

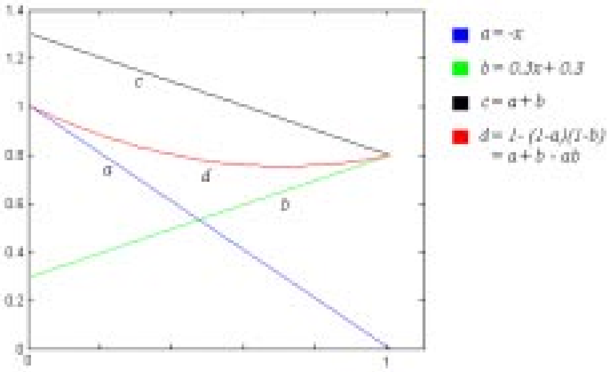


Figure 4.12: Bernoulli's rule of combination applied on two simple signals (a) and (b). (c) is $a+b$ and (d) their BRC or $a+b-ab$.

spectively. Also, a series of peaks-position based feature extraction algorithms that were applied on the composed feature map failed to obtain adequate classification accuracy. Tables 4.10 and 4.11 illustrate the GC algorithm results.

To conclude, the combination of Gabor filtering and GLCM analysis in the new Gabor Composition algorithm provided the maximum classification accuracy on the KNN classifier. GC also can localise the defects in its feature map before final feature extraction. Its promising performance illustrates the potential for improving a statistical method's performance using a signal processing method as pre-processor and data conditioner.

4.5 Computational Costs and Performance Comparison

In any industrial application, as well as accuracy, computing cost is another factor that should be considered. Table 4.12 illustrates the overall ranking of the results of textured tile classification experiments on both KNN and BPNN classifiers. Table 4.13 depicts the average execution time of the more promising feature extraction algorithms after 40 runs on a PC Pentium III-700

Tile Type	F_{GC}^{1}	F_{GC}^{2}	F_{GC}^{3}	F_{GC}^{4}	F_{GC}^{5}
`ARDES`	40.00%	35.42%	60.00%	60.00%	60.00%
`ARWIN`	70.00%	88.75%	88.75%	88.75%	87.50%
`CASA`	67.14%	95.83%	97.14%	98.57%	92.86%
`DJZAM`	70.00%	70.00%	90.00%	90.00%	85.00%
`DJZUL`	75.00%	85.00%	85.00%	90.00%	85.00%
`KIS`	90.00%	91.96%	95.00%	95.00%	90.00%
`LRSIDE`	70.00%	90.15%	84.00%	82.00%	86.00%
`PRODO`	92.50%	92.62%	97.50%	80.00%	97.50%
`PRODT`	82.50%	77.98%	77.50%	100%	82.50%
`SLTNP`	95.00%	73.51%	75.00%	97.50%	90.00%
`SYM`	85.00%	87.50%	95.00%	90.00%	90.00%
Average	76.10%	80.79%	85.9%	88.35%	86.03%
Variance	0.0244	0.0294	0.0138	0.0129	0.0092

Table 4.10: Defect detection results using GC algorithm and the KNN classifier.

MHz machine.

Amongst the statistical approaches tested, GLCM provided the best classification accuracy on BPNN (97.09%), but not a good one on KNN (81.08%). Computationally however, GLCM was a moderate algorithm with 0.132 sec execution time per tile. GC provided the most promising results on KNN (88.35%) and the second best on BPNN (97.02%). Its execution however, needed 0.198 sec per tile which made it a costly algorithm. Although GC was in fact a combination of Gabor filtering and GLCM, its computing time was less than the summation of those two methods (0.198 vs $0.132+0.114=0.246$), since the GLCM algorithm was applied on 256-grey level images and utilised 3 distances, whereas GC was applied to 37-grey level images and a single distance.

Tile Type	F^{1}_{GC}	F^{2}_{GC}	F^{3}_{GC}	F^{4}_{GC}	F^{5}_{GC}
`ARDES`	88.75%	85.00%	100%	100%	100%
`ARWIN`	80.31%	97.50%	98.75%	97.19%	98.13%
`CASA`	93.93%	96.83%	100%	99.64%	100%
`DJZAM`	82.50%	95.00%	100%	95.00%	87.50%
`DJZUL`	76.25%	95.00%	90.00%	100%	88.75%
`KIS`	95.00%	95.00%	90.00%	95.00%	95.00%
`LRSIDE`	92.00%	90.00%	92.00%	88.50%	89.00%
`PRODO`	95.63%	92.50%	100%	98.75%	98.13%
`PRODT`	93.13%	90.00%	100%	96.25%	96.86%
`SLTNP`	95.00%	95.00%	100%	97.50%	98.75%
`SYM`	83.75%	85.00%	90.00%	99.38%	90.00%
Average	88.75%	92.44%	96.43%	97.02%	94.74%
Variance	0.0047	0.0019	0.0023	0.0011	0.0024

Table 4.11: Defect detection results using GC algorithm and the BPNN classifier.

The proposed DWHT was the fastest algorithm. With 0.009 sec execution time, it was 12.6 times faster than the next fastest that was Gabor filter. Its classification performance (i.e. $\mathrm{CA}_{\mathrm{KNN}}$=82.62% and $\mathrm{CA}_{\mathrm{BPNN}}$=95.58%) was also good and gave it the fourth best rank on both classifiers. Thus, in a realtime texture analysis application DWHT is a reasonable choice. In our experience, although DDCT outperformed DCT, it stood behind DWHT on both classification accuracy and computing time (not reported in Table 4.13). Eigenfiltering was another interesting signal processing method. It was amongst the most accurate algorithms (the second on KNN and third on BPNN), however the slowest one as well. The execution time for smaller N s were much lower than 7×7. For instance, execution time for N =3×3 was only 0.025 sec. However, it was still around 2.78 times more than DWHT.

Figure 4.13 shows the average execution time of different algorithms, where the advantage of

		Classifier			
Algorithms	**NF**	$\mathrm{CA}_{\mathrm{KNN}}$	**Rank**	$\mathrm{CA}_{\mathrm{BPNN}}$	**Rank**
Ordinary Histogram	256	68.06%	8	77.54%	8
LBP	256	78.93%	7	84.18%	7
GLCM	48	81.08%	6	97.09%	**1**
Gabor Filter	36	84.87%	3	91.22%	6
DWHT	48	82.62%	4	95.58%	4
DDCT	48	82.15%	5	95.14%	5
Eigenfilter $N = 7 \times 7$	98	85.30%	2	95.70%	3
GC	16	88.35%	**1**	97.02%	2

Table 4.12: Summary of tile classification experiments. NF is the number of features.

Algorithms	**Running Time (sec)**	**Rank**
GLCM	0.132	3
Gabor Filter	0.114	2
DWHT	0.009	**1**
Eigenfilter $N = 7 \times 7$	0.223	5
GC	0.198	4

Table 4.13: Different algorithms' running time (sec).

DWHT is visible. Figure 4.14 illustrates the classification performance of different algorithms used, averaged on all types of tile. The error bars show the standard error:

$$SE = \sqrt{\frac{\sigma^2}{n}} \tag{4.27}$$

where σ^2 is variance and n is the number of tile types.

4.6 Conclusion

We implemented three statistical (ordinary histogram, LBP, and GLCM) and five signal processing schemes (Gabor filter, DWHT, DDCT, Eigenfilter, and GC) in randomly textured tile abnormality detection experiments, using two different classifiers: KNN and BPNN. To obtain the best outcomes, a series of parameter optimisation steps was carried out on each algorithm which successfully improved their performances (e.g. by almost 20% for Gabor filters). The ordinary histogram was employed only to measure the complexity of the defect detection problem, while all other algorithms applied were texture based. The GLCM illustrated promising performance on BPNN. The eigenfiltering algorithm, optimised on the filter size, showed high accuracy on both classifiers. However, on average the proposed GC algorithm was the most accurate method. In particular a well-tuned GC was very effective and accurate on the simpler KNN classifier. The GC method showed also good generality across various tile types, apart from type `ARDES` on the KNN classifier, where possibly the coarse structure of the texture demanded a different parameter setting. The procedure of the GC algorithm, (i.e. Gabor filtering, quantisation, exclusion, and Bernoulli's combination), decreased the redundancy as well as highlighted the defective regions (in particular their boundaries) in the feature map. Consequently the GLCM could extract more effective features for texture classification.

The proposed DWHT was the fastest algorithm while also showed a high classification accuracy. Therefore it can be declared as a reasonable choice for realtime applications.

The proposed classification algorithms carry two significant shortcomings. Firstly, they rely on a lengthy training stage, particularly in the case of the BPNN classifier. Secondly, they depend on the availability of a substantial number of defective samples, that moreover, should cover the full range of possible defects. These are not always available. In the next chapter we attempt to overcome these disadvantages using an effective novelty detection scheme which is both accurate and independent of defective samples.

Figure 4.13: Execution time of different algorithms.

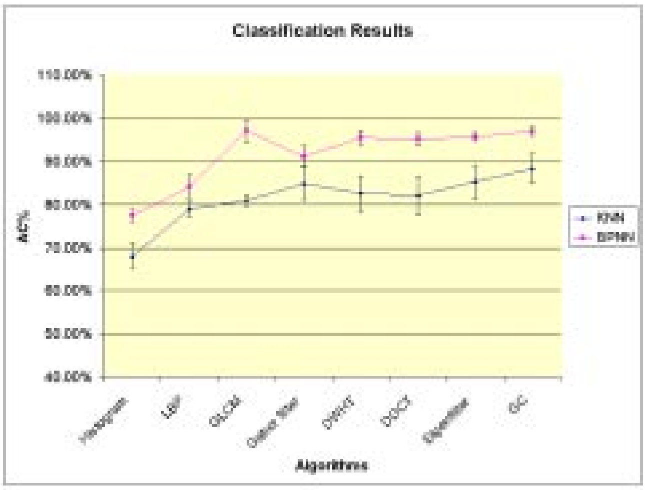


Figure 4.14: KNN and BPNN classification overall results.

Chapter 5

A New Eigenfilters-Based Method for Abnormality Detection

5.1 Introduction

A typical approach to normal-abnormal pattern classification comprises feature extraction and then training a classifier with feature vectors of both negative and positive samples (e.g. normal and defective tiles). Consequently the trained classifier will be used to classify unknown input samples. By employing well aimed pre and post-processing steps, efficient features, and powerful classifiers, we can expect good classification performance from such a typical scheme. However, this traditional method suffers from a few practical drawbacks. Firstly, in real cases the number of abnormal samples is usually much lower than the number of normal samples. (e.g. how many examined patients have a particular kind of cancer?) Therefore, the classifier usually cannot be facilitated with enough abnormal training samples. Secondly, in many cases abnormalities and defects are diverse and unpredictable. So even a large set of abnormal samples may not necessarily cover all the possible forms of defects. In other words, in a given N-dimensional feature space, whereas the normal samples may build well defined and separated clusters, we may see many ill-defined regions of abnormal samples which are scattered across the feature space. However, all the abnormal samples should show a common characteristic: a considerable distance to normal clusters. This ‘distance’ is the base of a different approach to normal/abnormal sample classification which is called *novelty detection* or *concept learning* [55]. There are several novelty detection schemes developed for a variety of applications, for instance [51, 55, 109] (for more examples see Section 2.7.4). However, the basic idea behind almost all of them is measuring a distance toward a set of positive reference samples, here called the *template* set, as the level of novelty of the input sample. Measurement is usually carried out in the feature space and the template should adequately cover the possible positive samples set.

In this study, we develop a new eigenfilter based novelty detection scheme to segregate abnormal textured tiles. The method is based on the reconstruction of the test image twice: once by its own eigenfilter bank and once again by a template tile’s eigenfilter bank, and measuring the reconstruction error as the level of novelty.

In the next section, after a brief review of PCA analysis and eigenfilters, the proposed method and the results of the first series of experiments is detailed. Section 5.3 describes an improved filter’s structure oriented method and the results of the second series of tests concerning this new method. The study is concluded in Section 5.4.

5.2 The Proposed Method and the First Experiments

5.2.1 PCA Analysis and Eigenfilters: Background

PCA and eigenfiltering have been previously discussed in Sections 2.7.1 and 4.3.7. Here we only look over the notations again. Using the covariance matrix of the $k \times n^2$ rearrangement of a given image A (n^2 is the neighbourhood size or N), eigenfilters of A would be computed and utilised to generate n^2 detail images:

$$D_i^A = A \otimes F_i^A \quad , \quad 1 \leq i \leq n^2 \tag{5.1}$$

Where F_i^A and D_i^A are the i^{th} eigenfilter and detail image of A. Figure 5.1 illustrates a tile image, its 25-filter N =5×5 eigenfilter bank, and resulted detail images. Two characteristics of eigenfilters are of importance and helpful in our proposed novelty detection method: *adaptability* and *orthogonality*. Adaptability means the eigenfilter bank of different images are different and each eigenfilter conveys the certain attributes of the image, regarding the eigenfilter size. Orthogonality means we can successfully reconstruct the original image by composition of its detail images [2]. The eigenfilter bank decompose the image into an orthogonal set of detail images. The process of decomposition however, is image dependent. Also, eigenfilters and detail images of similar images are close to each other (considering distances in Euclidian spaces) and vice versa. We use all these factors to establish our eigenfilter based novelty detection algorithm.

5.2.2 Data Set

The applied data set included eight diverse types of textured ceramic tiles. Samples of each type were divided into four non-overlapping categories of *abnormal*, *normal*, *template* (reference,

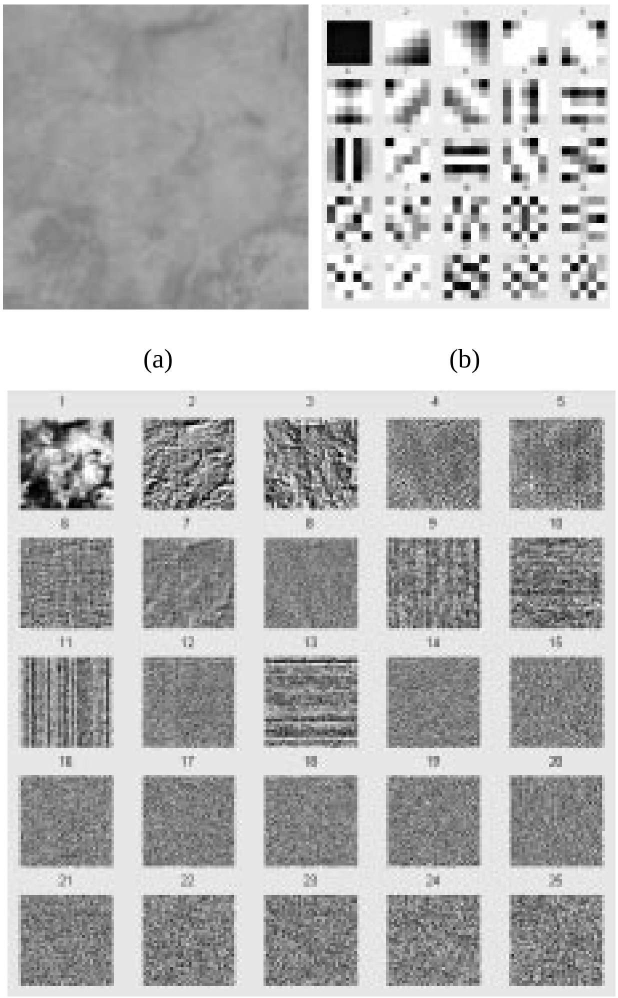
T), and *training* (parameter estimation, P). The number of samples in the normal and abnormal categories was kept equal, and all the remaining good samples were used to build the template and the training sets. As Table 5.1 shows, typically for each tile type in this study the template set was much larger than the training set.

	Number of Samples			
Tile Type	**Normal**	**Abnormal**	**Template**	**Training**
`ARDES`	22	22	224	20
`ARWIN`	40	40	82	15
`DJZAM`	20	20	170	20
`DJZUL`	20	20	170	20
`KIS`	23	23	39	10
`LRSIDE`	46	46	135	20
`SLTNP`	26	26	161	20
`SYM`	20	20	145	20

Table 5.1: Tile types and number of samples

5.2.3 The Method

Initially, we compute and store the eigenfilters and detail images of all the reference images in our *template* set T. Then the eigenfilters of a new tile image A, are computed. We search the template set to find the most similar image to A. This template member is referred as M. When textures are involved, usually methods such as pixel-by-pixel comparison, histogram or power spectrum differences, are not effective to evaluate the similarity (i.e. distance, in a given Euclidian space). Instead, textural properties can be more helpful.



(a) (b)

(c)

Figure 5.1: Filtering procedure for a DJZAM tile (a), its 5×5 eigenfilters (b), and detail images (c). To increase the visibility, all filters and detail images have been equalised.

For example, the most similar textures could be the pair with the closest co-occurrence matrices or the closest MSMD filter responses [44, 54, 127]. Since eigenfilters preserve the textural characteristics, the distance between eigenfilter responses (i.e. detail images) could be used as a textural similarity metric. For such comparisons, the widely applied χ^2 distance function is used [77, 76, 123]. χ^2 is in fact a normalised (or relative) distance between two vectors regardless of their elements' magnitude and can be defined as:

$$\chi^2(A,M) = \sum_{i=1}^{N} \frac{(D_i^A - D_i^M)^2}{|D_i^A| + |D_i^M|} \tag{5.2}$$

where D^A and D^M are detail images of A and M using (5.1). Hence M would be the member of template with the minimum χ^2 distance to A. However, considering the adaptability of the eigenfilters, it is also possible to implement an eigenfilter vs. eigenfilter similarity measurement. Given F^A and F^M as any two eigenfilters that are to be compared, then:

$$\chi^2(A,M) = \sum_{i=1}^{N} \frac{(F_i^A - F_i^M)^2}{|F_i^A| + |F_i^M|} \tag{5.3}$$

In fact, we tried both distance between detail images (DBD) and distance between filters (DBF) schemes and the comparative results are discussed later in Section 5.2.4. We also applied some other distance functions, for instance MSE and vectors angle. On average, χ^2 showed the highest performance in these experiments.

Next, the test image, A, is reconstructed twice: once by a subset of its own eigenfilters and once again by a subset of the selected M's eigenfilters (resulting in R_A and R_M). The number of filters in the subset is naturally important and is dealt with in Section 5.2.6. Reconstruction could be carried out by simple addition of the detail images or Bernoulli's rule of combination [103]. In these experiments, BRC did not show any advantage to simple addition, hence subsequently only addition-based reconstruction was used:

$$R_A = \sum_i D_i^A \quad , \quad R_M = \sum_i D_i^M \tag{5.4}$$

The error between the reconstructed pair is considered as the level of abnormality of test image A. An error larger than a given threshold, Υ, is considered as a sign of a defect on the tile surface texture:

$$\Delta E = |R_A - R_M| > \Upsilon \quad \Rightarrow \text{ DEFECT} \tag{5.5}$$

Figure 5.2 specifies the proposed abnormality detection algorithm.

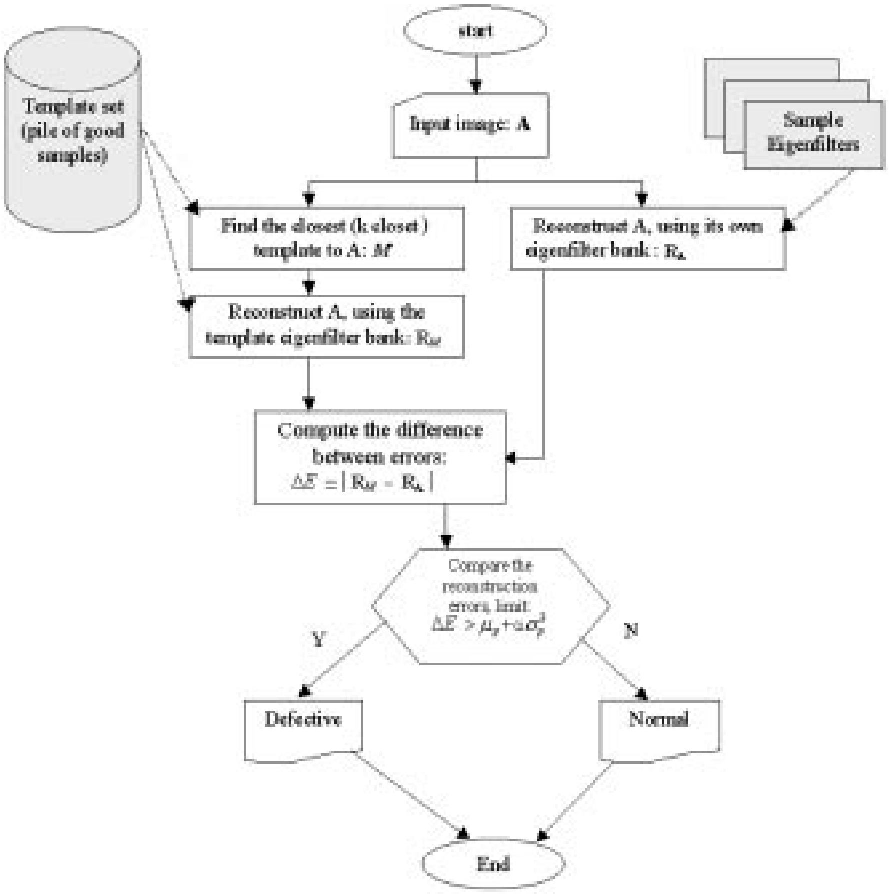
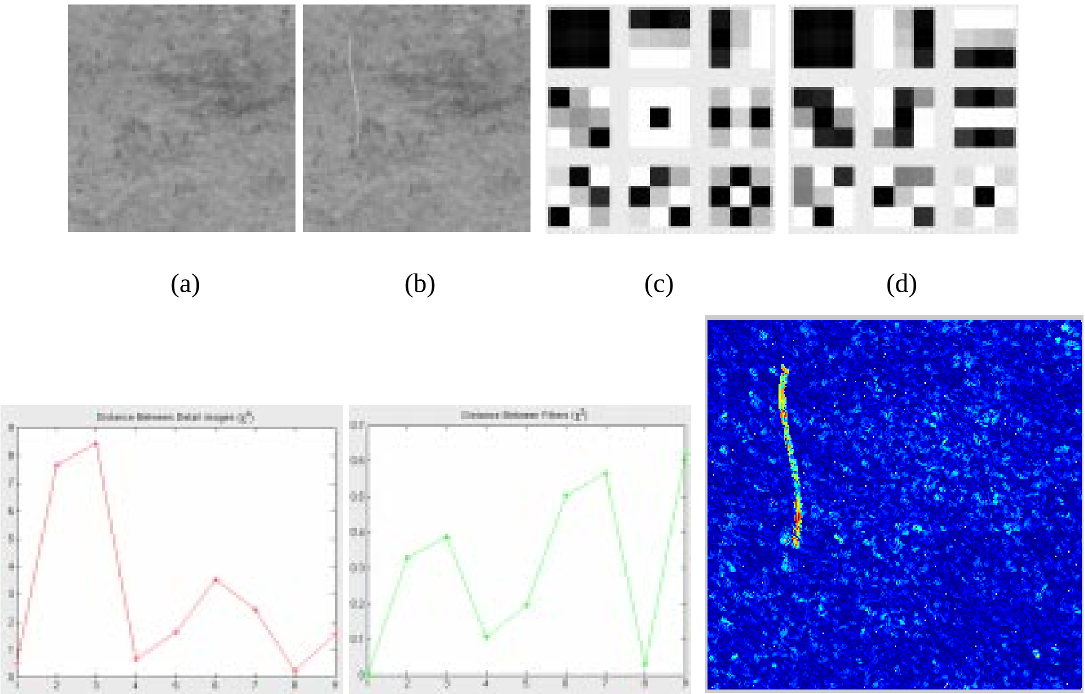


Figure 5.2: Eigenfilter-based abnormalities detection algorithm

Figure 5.3 illustrates a `KIS` tile and its artificially defective version (round grey area in the bottom right). The 3 × 3 eigenfilter banks are different due to the flaw. Graphs 5.3(e) and 5.3(f) respectively show the χ^2 distance between the eigenfilter banks and the detail images. Figure 5.3(g) presents the reconstruction error map using filters 2 to 7. Note that in the reconstructed map the defective area boundary shows a higher error level. Figure 5.4 illustrates the same procedure for another `KIS` tile, where this time 7 × 7 filters are exploited. Again the defective area in the reconstruction error map (g) shows a higher level of error. In this example filters 25 to 49 were employed for reconstruction.



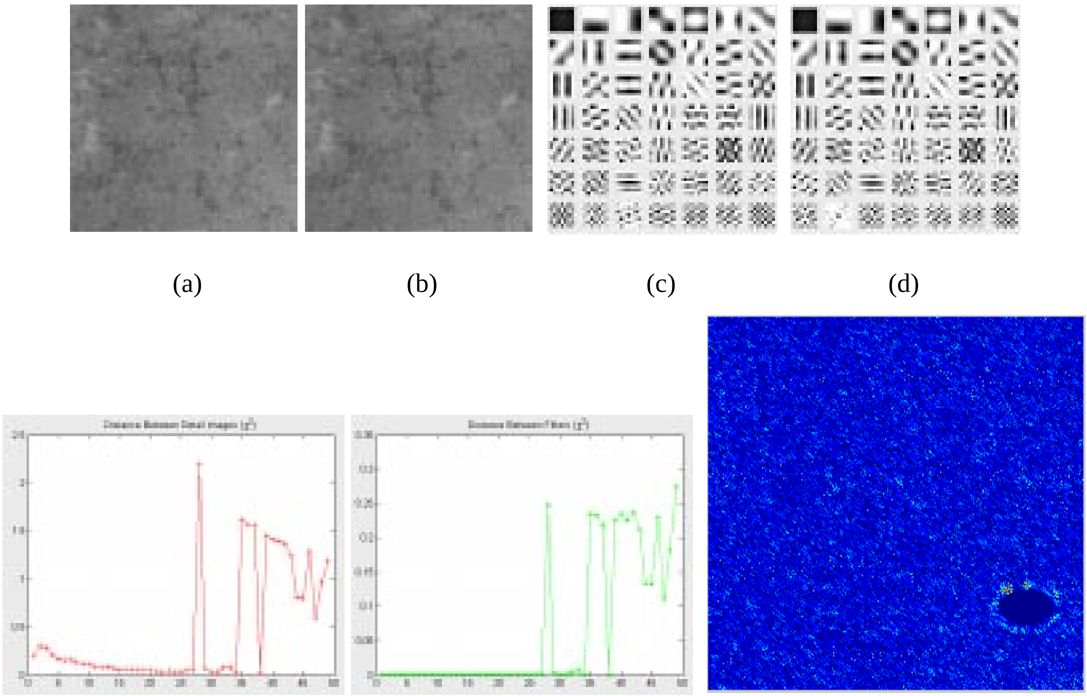
(a) (b) (c) (d)

(e) (f) (g)

Figure 5.3: A normal `KIS` tile (a), and its artificially defected version (b), their respective 3×3 eigen-filter banks ((c) and (d)), the χ^2 distance between filters (e) and detail images (f), and the reconstruction error map (g).

5.2.4 The First Experiments

In the first series of experiments, presented in Table 5.2, three different neghibourhood sizes, $\mathcal{N}$ $=3\times3$, 5×5 and 7×7, and -inspired by KNN classifiers- a different number of closest templates were tried. The second column shows the outcomes of the DBF scheme, and the third column presents the outcomes of the DBD scheme. Results suggest that the best classification performance of 85.32% was achieved by using a 3×3 neighbourhood, three closest Ms, DBD, and reconstruction using 6 (out of 9) detail images. In fact for all tried $\mathcal{N}$, the accuracy of DBD was higher than DBF. However, it was achieved at the expense of slightly more elaborate computations. The DBF method for $\mathcal{N}$ $=3\times3$ took 0.501s on a 700 MHz PC, while it took



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

(e) (f) (g)

Figure 5.4: A normal `KIS` tile (a), and its artificially defected version (b), their respective 7 × 7 eigenfilter banks ((c) and (d)), the χ^2 distance between filters (e) and detail images (f), and the reconstruction error map (g).

0.538*s* for the DBF. The higher accuracy of DBD means it will be considered as the preferred scheme in all further experiments. Degrading performances prevented us from testing larger N s. As an exception, and only for $N = 3 \times 3$, in this test employing 3 closest Ms provides the optimum results. In such cases, average distance to a few (here: 3) closest Ms would be computed. In all other tests, the best performance is obtained by a single closest M. Optimum results shown are achieved by employing a subset of detail images in the reconstruction stage. 6 detail images are involved for $N = 3 \times 3$, 12 (out of 25) for 5 × 5, and 32 (out of 49) for 7 × 7. The method used to choose these subsets will be discussed later.

Perhaps one of the first questions to arise about the proposed method is how effective this particular reconstructional algorithm is. In other words, why cannot the distance between detail images be used as the normality/abnormality metric? The last row of Table 5.2 illustrates the outcomes of a detail images-based classification effort with all circumstances kept similar with the tests above, but without the reconstruction phase and by utilising all the detail images as the feature of novelty. Reduced classification accuracy of 12.91% (from 85.32% to 72.41%) shows that the hypothesis of reconstruction via two different filter banks and using a subset of filters, is effective and increases the overall accuracy. This comparison shows that while eigenfilter is successful as an ordinary classifier (see Section 4.6 and Table 4.12), to apply it as a promising novelty detector, we need to develop a new eigenfilter-based scheme. The slight diversity of eigenfilters of A and M, amplifies the differences of the reconstructed pair, and the amplification magnitude will be higher when the test image A is defective.

Neighbourhood Size (N)	**CA Using DBF**	**CA Using DBD**	**No. Filters Involved**	**No. Ms Involved**
3×3	81.12%	85.32%	6	3
5×5	77.62%	81.25%	12	1
7×7	74.22%	79.90%	32	1

Classification Without Reconstruction

Neighbourhood Size (N)	**CA Using DBF**	**CA Using DBD**	**No. Filters Involved**	**No. Ms Involved**
N =3×3	69.67%	72.41%	all	1

Table 5.2: Results of the first series of experiment. DBD and DBF are distance between filters and detail images respectively, and No. Ms is the number of involved closest templates.

The proposed algorithm looks simple and straightforward, however, certain fundamental issues arise out of the proposed method and implemented tests that we now attempt to deal with. For instance, how the optimum threshold (Υ), or the optimum subset of filters/detail images for reconstruction can be established.

5.2.5 Finding the Optimum Υ

The simplest way to choose the Υ is considering the maximum reconstruction error of the training set as that threshold. However this can slightly decrease the sensitivity of the novelty detector. So, the choice of Υ is determined through a simple type-by-type training or parameter estimation stage. Initially, we apply the proposed algorithm on the training set P (which contains only 'good' samples) and obtain the reconstruction errors. Then the mean (μ_P) and the standard deviation (σ_P) of the reconstruction errors are computed. The optimum threshold Υ is assumed to be an α weighted deviation from the mean:

$$\Upsilon = \mu_P + \alpha\sigma_P \tag{5.6}$$

Thus, any unseen tiles with reconstruction error $\Delta E > \Upsilon$ will be considered as abnormal. Furthermore, we continue the parameter estimation stage to determine the optimum value for α using k-fold cross-validation. Here a 4-fold cross-validation was employed, where 75% of samples were used for parameter estimation, and 25% for testing the performance. The result is taken as the average of four iterations of that procedure on non-overlapping subsets.

As an example, Fig. 5.5 depicts the distributions of ΔE for normal, abnormal and template sets of a specific type of tile. In this case, the template's ΔE statistics are $\mu_P = 0.24$, and $\sigma_P = 0.11$ (Fig. 5.5(top)). The test data reconstruction errors are plotted in Fig. 5.5(bottom). The cross-validation algorithm estimates the optimum separation parameter set as $\Upsilon = 0.36$, from $\alpha = 1.12$, on the normalised ΔE axis. The subsequent correct classification rate for this example was 95.0%.

Next, we describe how the near-optimal number of eigenfilters are selected to reconstruct an image, whether for routine comparison of unseen tiles against the template set T or to determine the optimum value of Υ as just described.

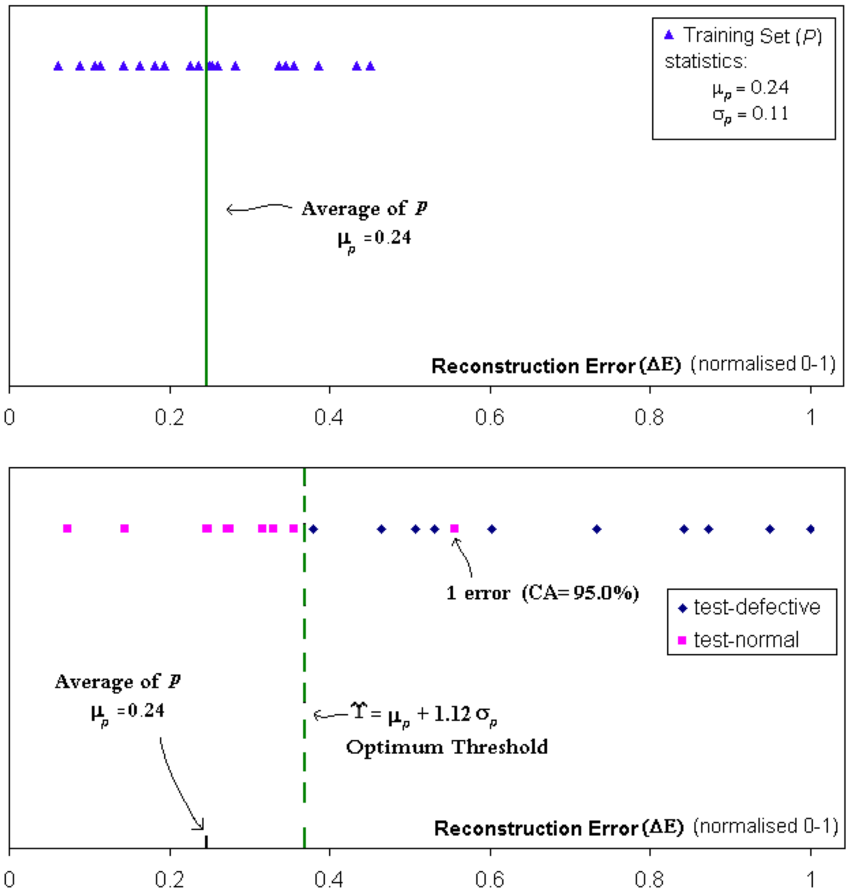


Figure 5.5: Reconstruction error (ΔE) distribution for (top) training set P, (bottom) normal and abnormal test samples. The training set parameters (μ_P, σ_P) are used in computing the optimum threshold for the test samples. The ΔE axis has been normalised to lie in the range [0-1].

5.2.6 Finding The Optimum Subset of Eigenfilters

As mentioned before, eigenfilters are an orthogonal set, therefore any reconstruction using a complete set of detail images would be error free. Also, a few filters (and consequently detail images) of a defective tile may convey the information of the defect. So, the resulting reconstruction error using those filters may provide the optimum discrimination. However, it would be difficult to find the optimum subset. The first reasonable selection could therefore be the filters with highest eigenvalues. Nevertheless our experiments have shown that this was not the best option, as the correct classification achieved was limited to 72.92%. This lack of performance is expected since the defective area is usually a small portion of the whole tile, and therefore cannot affect the first eigenvectors which convey the basic structure of the image. Consequently, employing a subset of the smallest-eigenvalue filters performed even worse with 69.91% correct classification.

Another option could be measuring the distance between pairs of counterpart detail images in M and the test image A. Reasonably, pairs with maximum distance may convey the differences, i.e. abnormalities information, whereas closer pairs may convey the similarities. However, this assumption has failed during tests, when a subset of a few closer filters provided a better classification result than a subset of farther ones. In particular the specificity (SPC) factor of the farther filters was lower than the closer filters, which suggests that we need the closer filters too to keep the SPC adequate. Table 5.3 exhibits the results and suggests that whereas a subset of 6 closer filters reach up to 85.32% correct classification, maximum accuracy of the farther subset is limited to 77.13%. Moreover, the farther subset shows a considerably lower specificity, (0.761 vs. 0.596), which means that this selection puts more normal tiles wrongly into the abnormal category. However, as expected, the sensitivity (SNS) of the farther set is marginally better than the closer (0.954 vs. 0.946), which shows that it can reveal the defects slightly more clearly.

Filter Subset	**No. of Filters**	**CA**	**SNS**	**SPC**
Closer	6	85.32%	0.946	0.761
Farther	6	77.13%	0.954	0.596

Table 5.3: Comparison between closer and farther subsets performances.

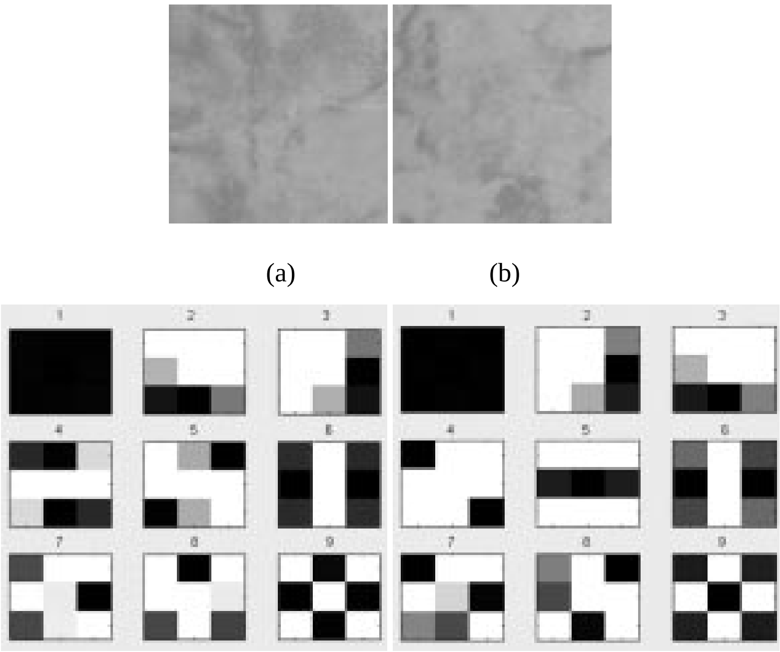
Although the classification accuracy of 85.32% is relatively acceptable, we must try to increase it and also find out why the smallest N has provided the best performance. In the following sections we will explain an improved filter-structure based method to achieve a higher performance.

5.3 Improvement Through Matching by Structure

In the proposed algorithm, when we want to measure the similarity between two textures or the distance between two filters, we implicitly sort the filters or detail images by the eigenvalues, and then compare the corresponding pairs. However, this scheme may have some disadvantages. For example, Figure 5.6, which depicts 3×3 eigenfilter banks of two `DJZAM` tiles, suggests that an ordinary comparison (e.g. by using χ^2) of filter pairs while all have been sorted by eigenvalues, may show an incorrect distance between two wrongly matched filters. Consequently the computed overall distance between two textures would not be reliable. For instance, on that figure, edge detector filter pairs 2 and 3 (in (a) and (b)) are 90° rotated versions of each other. Therefore it might be more reasonable to match the 2^{nd} filter of (a) with the 3^{rd} filter of (b) and vice versa. As another example, pair 8 are complement (negative) of each other, however, both are pixel detectors. In such cases, χ^2 definitely cannot be applied to measure the structural similarity between gradient filters. So, the calculated χ^2 distance (either between filters or between detail images) may not be very precise.

Another drawback of the current algorithm is its rotation-variant property, where rotation of a tile will rotate the filters and consequently disrupt the distance measurement. Uneven surfaces of many tiles and slanted lighting of the image grabber system amplify the effects of rotation on the texture. As an example, Figure 5.7 shows an `ARDES` tile, its 90° rotated version and their 3×3 and 5×5 eigenfilter banks. Rotation has affected almost all of the filters. (e.g. filters 2,3 and 5 to 8 in 3×3 (c,d) and 2,3 and 5 to 13 in 5×5 (d,e)).

Furthermore, Figure 5.7 also reveals that finding some sort of similarity or relation amongst larger 5×5 filters is much more difficult than smaller 3×3 ones, perhaps due to the consid-



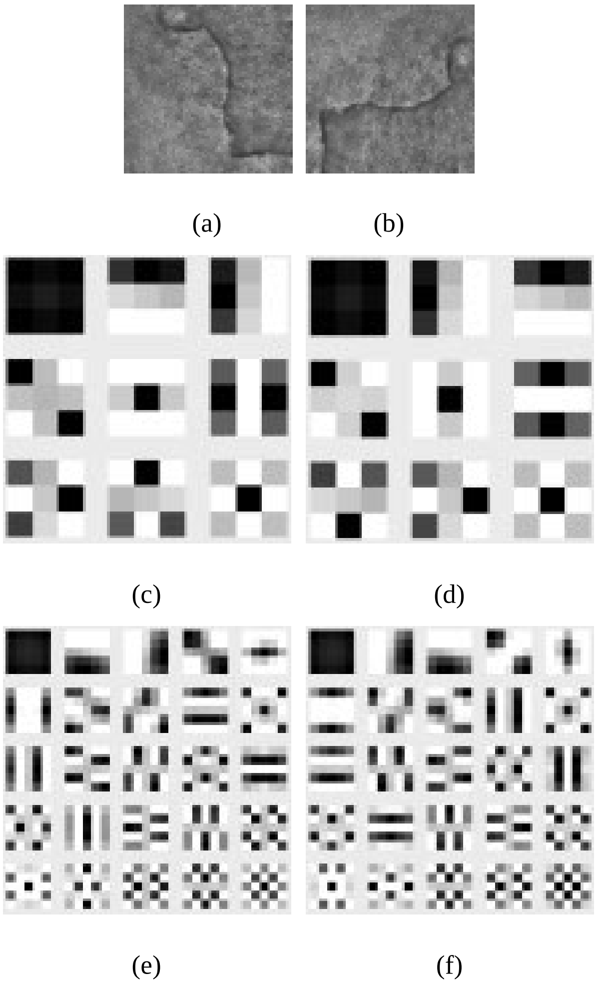
(a) (b)

(c) (d)

Figure 5.6: 3×3 Filter banks of two DJZAM tiles.

erably higher diversity of 5×5 matrices. Therefore the usual *matching-by-eigenvalue* (MBE) scheme would be even less effective for larger neighbourhood sizes. That can be a reason for having lower performance while enlarging N in the first experiments (see Section 5.2.4 and Table 5.2). Again, as Figure 5.8 exhibits, indeed the difference between eigenvalues λ_j , $j = 2 \ to \ n^2$, are relatively small and adjacent λ_j and λ_{j+1} are very close together. Hence, after a slight change in the texture, a swap between adjacent filters will not be unexpected.

To summarise, we need a more effective and rotation-invariant scheme to match the filter pairs, in order to replace the current MBE method. Hence we propose an alternative comparison scheme which involves the filter's structure. This scheme will allow *all the filters* of M to compete as the possible counterpart of the i^{th} filter of the test image, A, regardless of their eigenvalues. In the new scheme, the distance δ, between two specific filters would be computed



(a) (b)

(c) (d)

(e) (f)

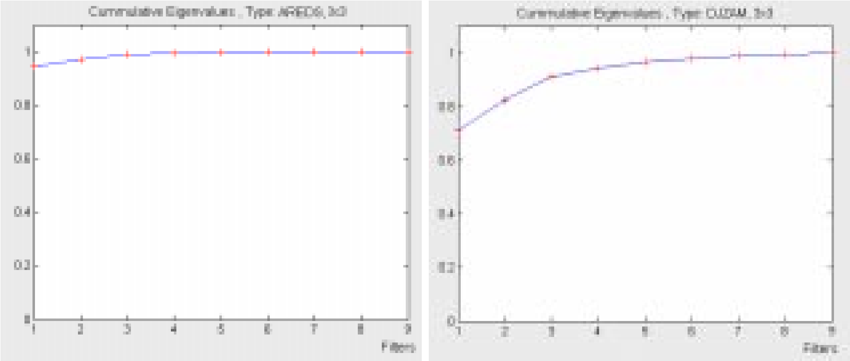
Figure 5.7: Effect of a 90° rotation of a tile on eigenfilters. The majority of filters have also been rotated.

as:

$$\delta_{F_i,G_j} = min(\ \chi^2(F_i,G_j)\ ,\ \chi^2(F_i,G_j^{\rightleftarrows})\ ,\ \chi^2(F_i,G_j^{\uparrow\downarrow})\ ,\ \chi^2(F_i,\overline{G_j})\ ,\ \chi^2(F_i,G_j^{\theta})\ ) \tag{5.7}$$

where,

$$\chi^2(F_i,G_j^*) = \frac{(F_i - G_j^*)^2}{|F_i| + |G_j^*|}$$



(a) (b)

Figure 5.8: Cumulative eigenvalues of two types: ARDES (left) and DJZAM (right)

F and G are eigenfilters of given A and M, and $G_j^{\rightleftarrows}$ and $G_j^{\uparrow\downarrow}$ are vertically and horizontally mirrored (i.e. ‘flipped’) matrices, $(\overline{G_j})$ is the complement (i.e. ‘negative’) of the matrix, and G_j^{θ} indicates the θ degrees rotated version of the input.

Mirroring is implemented by swapping the columns or rows of G. Complementing is performed by using the mean value of the filter μ_X as the origin:

$$\overline{X} = -(X - \mu_X) + \mu_X \tag{5.8}$$

Then, two filters with minimum distance are paired. We refer to this improved approach as *matching by structure* or MBS. MBS consequently affects both finding M and selecting the optimum filter subset, where the number of filters in the subset used for computing the reconstructed image is then worked out as before (see Section 5.2.6).

Table 5.4 presents the performance of the MBS method, where there is a slight improvement (see Table 5.2) for smaller neighbourhood size of $N = 3 \times 3$, (85.32% vs. 86.71%), and considerable improvements for larger sizes $N = 5 \times 5$ (81.25% vs. 91.19%) and 7×7 (79.90% vs. 91.46%), then the accuracy slightly degrades for 9×9 down to 90.74%. In fact during the tests we realise that for the smaller 3×3 size, MBE and MBS are quite similar. However, the new scheme compensates the filter matching problem of larger N s up to 7 × 7, and the classifier can now gain the advantages of the grown neighbourhood sizes. All the reported best performances have been achieved by utilising the farther filter subsets. In fact, the classification accuracy of closer subsets were limited to 83.97% at most for $N = 7 \times 7$. This means that MBS has considerably increased the SPC factor of the farther subset compared to the previous MBE method. Utilising MBS, now all the optimum results (CA, SNS and SPC) are attained with $N = 7 \times 7$.

Neighbourhood Size	CA	SNS	SPC	No. Filters Involved	No. Ms Involved
3 × 3	86.71%	0.898	0.835	5	1
5 × 5	91.19%	0.969	0.853	14	1
7 × 7	91.46%	0.972	0.855	26	1
9 × 9	90.74%	0.966	0.850	43	1

Table 5.4: Classification performance using matched-by-structure filters

Although $N = 7 \times 7$ has the maximum CA, it is only 0.27% higher than 5 × 5. Also the larger the size, the more time consuming the algorithm. Figure 5.9 outlines the computing time of the algorithms for different N and matching schemes. Tests were run on a PC Pentium III 700 MHz computer several times and then averaged. Results reveal that regarding an enlarging N, whereas the increase rate of computing time for the MBE method is rather low (from 0.53 to 2.31 seconds), the increase rate for the MBS method is very high (from 0.74 to 69.23 seconds). It is not unexpected, since the MBS applies much more matrix operations. Trying larger N s is deemed unnecessary, due to both decrement in accuracy and huge increment in computing time.

Table 5.5 presents the detailed results of the MBS method for different N for all our types of tile. Except for case `SLTNP`, both $N = 5 \times 5$ and 7 × 7 always achieve better classification accuracy than $N = 3 \times 3$. By the time we get to a 9 × 9 neighbourhood, a decline in the accu-

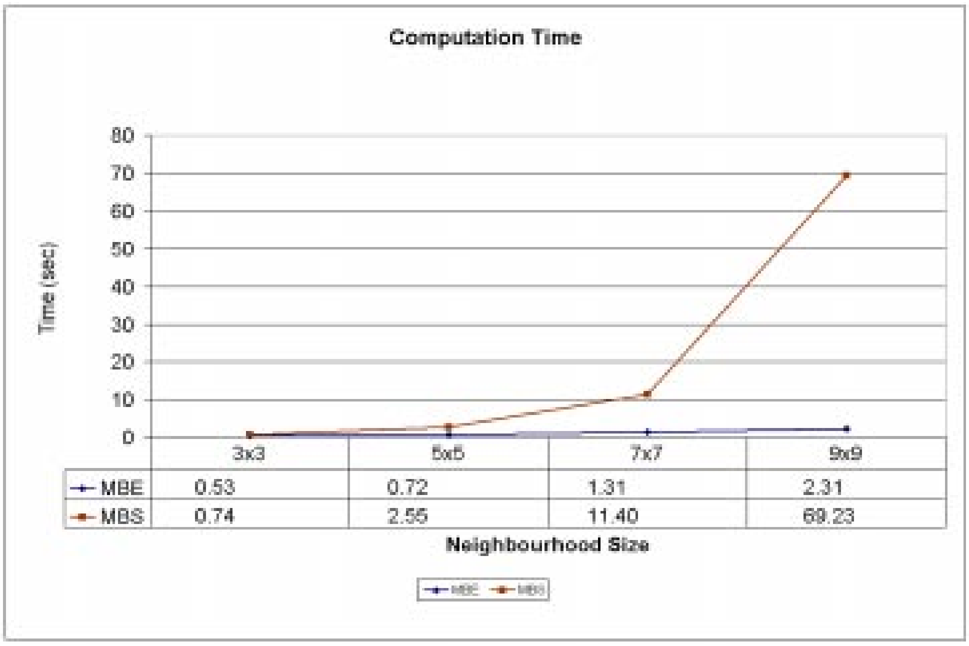


Figure 5.9: Average computing time comparison for the MBS method.

racy can be observed, thus showing N =7 × 7 as the optimal window on average. However, the N =5 × 5 case achieves a close average to the N =7 × 7 case, while also getting better individual accuracies for `LRSIDE` and `SLTNP` tile textures, all at much lower computational costs. Perhaps optimising N for a given type in the training stage could improve the overall classification performance even more. For instance, as this table shows, N =3 × 3 and 5 × 5 can compensate the 7 × 7 lack of specificity on particular types `LRSIDE` and `SLTNP` (e.g. 0.646 for 7 × 7 vs 0.706 and 0.922 for 5 × 5 and 3 × 3 on `SLTNP`).

Moreover, it was claimed that the MBS method is decisively less rotation variant. To prove that, applying both the MBE and MBS methods, the distance between eigenfilters and detail images of several tiles and their 90° rotated versions were computed, averaged, and compared. Table 5.6 presents the results and suggests that MBE is clearly less capable of revealing decisive textural similarity between rotated textures compared to MBS, since distances computed by MBS are many times smaller than MBEs (between 8.1 and 1563.5, depending on the case).

	$N = 3 \times 3$			$N = 5 \times 5$			$N = 7 \times 7$			$N = 9 \times 9$		
Type	**CA**	**SNS**	**SPC**	**CA**	**SNS**	**SPC**	**CA**	**SNS**	**SPC**	**CA**	**SNS**	**SPC**
`ARDES`	82.45%	0.852	0.797	83.72%	0.874	0.800	87.54%	0.875	0.875	84.56%	0.832	0.859
`ARWIN`	76.29%	0.850	0.675	87.24%	0.919	0.825	89.84%	0.965	0.831	87.33%	0.943	0.804
`DJZAM`	100%	1	1	100%	1	1	100%	1	1	99.81%	0.998	0.998
`DJZUL`	79.29%	0.792	0.792	100%	1	1	100%	1	1	99.81%	0.999	0.998
`KIS`	88.00%	0.899	0.861	93.55%	0.978	0.893	97.81%	1	0.956	94.27%	0.998	0.887
`LRSIDE`	85.29%	0.934	0.771	86.76%	1	0.735	81.72%	1	0.624	81.97%	0.976	0.712
`SLTNP`	92.22%	0.922	0.922	85.34%	1	0.706	80.80%	0.969	0.646	85.30%	0.983	0.682
`SYM`	90.15%	0.935	0.868	92.89%	0.988	0.869	93.98%	0.968	0.910	92.88%	0.998	0.859
μ	86.71%	0.898	0.835	91.19%	0.969	0.853	91.46%	0.972	0.855	90.74%	0.966	0.850
σ^2	0.0058	0.0042	0.0099	0.0041	0.0023	0.012	0.006	0.0018	0.022	0.0048	0.0033	0.0135

Table 5.5: Classification accuracy of different tile types for different neighbourhood sizes N, using MBS. μ and σ^2 are mean and variance.

Therefore, even having the rotated version of a test tile in the template, there is no guarantee that MBE can select that as the M, while MBS will most probably find it and can ensure a more reliable textural similarity measurement.

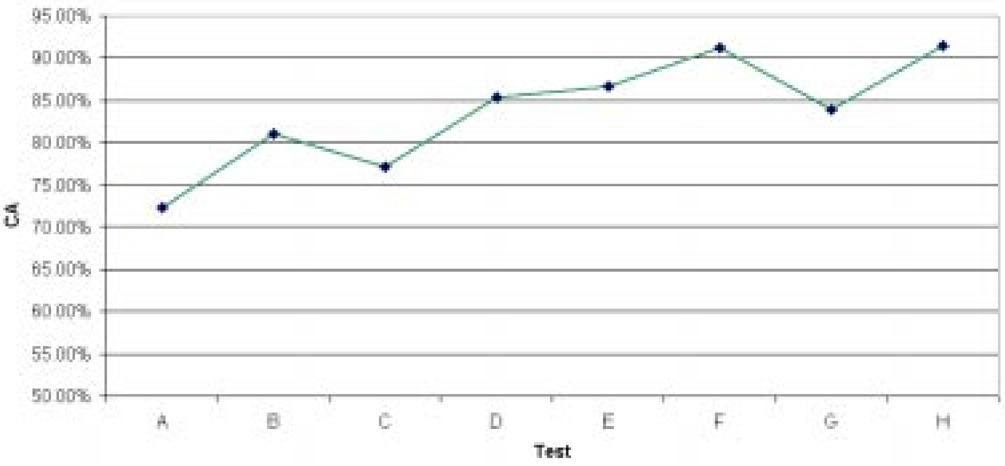
	χ^2 **- DBF**		χ^2 **- DBD**	
N	**MBE**	**MBS**	**MBE**	**MBS**
3×3	0.2461	0.0019	0.6254	0.0004
5×5	0.5409	0.0669	0.7358	0.0010

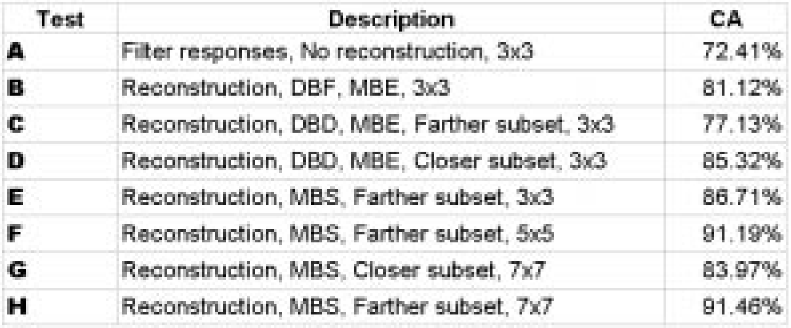
Table 5.6: Average distances between tile images and their 90° rotations.

5.4 Conclusion

The proposed eigenfilter-based novelty detection method showed significant performance and robustness in tile defect detection. Like other novelty detection schemes, its most important advantage is its relatively low dependence on abnormal samples. Moreover, the modified MBS scheme has added accuracy and flexibility compared to the original algorithm. The MBS is rotation invariant and capable of a more accurate normal/abnormal segregation. Figure 5.10 presents a conclusion of the various tests performed and their results. The first proposal is the obvious advantage of the proposed method compared to a simple filter responses distance measurement scheme (Test **A** vs. **B**, **C**, and **D**). It also suggests better performance of the closer subset compared to the farther one when the MBE scheme is used (Test **C** vs. **D**), and in contrast advantages of farther subsets when the MBS scheme is employed (Test **G** vs. **H**). Finally, a combination of MBS, farther subset, and $N = 7 \times 7$ neighbourhood size attains the optimum result of 91.46% correct classification (Test **H**).

To conclude, the eigenfilter-based novelty detection algorithm emphasises on the adaptability of eigenfilters and the similarity between the eigenfilters and detail images of similar textures. It also utilises the orthogonality of eigenfilters to reconstruct the test texture, where we expect more reconstruction error in the case of abnormal textures. To overcome the matching problem of eigenfilters and also to develop a rotation invariant defect detector, we introduce and exploit the MBS method, which shows promising detection performance, particularly on neighbourhoods of size $N = 5 \times 5$ and 7×7. Defect detection performance across all tested tile types is high (e.g. between 83.72% and 100% for $N = 5 \times 5$, see Table 5.5), which given the diversity of types suggests promise for the applicability of the method in the detection of textural abnormalities in general.





Test	Description	CA
A	Filter responses, No reconstruction, 3x3	72.41%
B	Reconstruction, DBF, MBE, 3x3	81.12%
C	Reconstruction, DBD, MBE, Farther subset, 3x3	77.13%
D	Reconstruction, DBD, MBE, Closer subset, 3x3	85.32%
E	Reconstruction, MBS, Farther subset, 3x3	86.71%
F	Reconstruction, MBS, Farther subset, 5x5	91.19%
G	Reconstruction, MBS, Closer subset, 7x7	83.97%
H	Reconstruction, MBS, Farther subset, 7x7	91.46%

Figure 5.10: A summary of various experiments outcomes

Chapter 6

Conclusions and Further Work

6.1 Summary

This thesis developed several efficient approaches to the texture classification problem, concerning natural and random textures in particular, with efficiency defined in terms of classification accuracy, computation costs, robustness and flexibility, and practicality. Experiment results on two texture-based applications (outdoor scenes object classification and tile defect detection) using three data sets (outdoor scenes, VisTex, and the tile image data set) all suggested that DWHT is a reliable, precise, and low cost texture feature extractor that can be applied on a wide range of texture analysis problems. Its classification performance was comparable with well-known Gabor filters when it was applied on outdoor scene and VisTex sets (e.g. 88.75% vs. 86.88% on outdoor scene tests). DWHT also obtained very good results in tile abnormality detection tests (e.g. 95.58% with the BPNN classifier). While its accuracy was usually lower than a pure Gabor filtering method, DWHT always was around ten times or more faster than other algorithms in all experiments. In addition, we developed a directional cosine transform algorithm (DDCT), which compared to its ordinary counterpart showed higher accuracy on tile abnormality detection (e.g. 89.84% vs. 95.14% on the BPNN classifier). The promising performance of DDCT showed that the angular/scale decomposition scheme of DWHT is also applicable on other similar kernel functions and transforms.

We also introduced two simple chromatic features H_p and S_p, which were inspired from the definitions of hue and saturation in standard colour spaces. The hue-like and saturation-like H_pS_p features performed well on both high resolution outdoor scenes and VisTex data sets. Their classification accuracy outperformed NRGB and HLS, but stood slightly lower than Lab . For instance, in VisTex tests, while Lab was the best feature set with 74.28% correct classification, H_pS_p accuracy of 72.07% put it in the second place, but much higher than HLS and NRGB with respectively 64.71% and 65.82% correct classification rates. The H_pS_p method was also very faster than the rest, for example 6.05 times faster than Lab on a PC PIII machine. We generally utilised colour features and texture features together to achieve higher classification performances in outdoor scenes experiments.

In the texture abnormality detection study, the DWHT was also proposed as a viable alternative. Initially, more traditional algorithms such as co-occurrence matrices, LBP, Gabor filters, and eigenfilters were experimented with. Excellent accuracy levels of 97.09% and 95.70% correct classification were obtained for GLCM and eigenfiltering respectively on the BPNN classifier. The best GLCM performance achieved after a series of optimisations on distances and the co-occurrence functions involved. In the case of eigenfiltering, the larger neighbourhood size of 7×7 provided the maximum accuracy. However, both algorithms were amongst the most time-consuming methods. As a fast and highly accurate alternative, DWHT with 95.58% correct classification on BPNN can be used. DWHT was almost 14.6 times faster than GLCM and 24.7 times faster than the 7×7 eigenfilter. Another statistical scheme tested, LBP, did not perform well and its accuracy was limited to $\mathrm{CA}_{\mathrm{KNN}}$=78.93% and $\mathrm{CA}_{\mathrm{BPNN}}$=84.18%

The implementation of the KNN classifier in real applications is easier than the BPNN, due to its relatively simple training stage. The best KNN performance however, was limited to 85.30% of the eigenfilter algorithm. In order to increase the KNN performance, we opted for the Gabor filter as the signal conditioner for a GLCM feature extractor and developed the proposed Gabor Composition algorithm. In this method, a Gabor filter bank was applied to amplify the defects and attenuate the background texture simultaneously, resulting in a composed feature map which later was fed to a GLCM process to extract the final feature vector. We then completed the proposed GC method with a detail image exclusion module and a Bernoulli combination module to further enhance its performance. GC increased the KNN accuracy to 88.35% and was on average the most precise algorithm in our defect detection tests.

Next, we developed an eigenfilter-based novelty detection algorithm for texture abnormality detection applications. The proposed algorithm utilised a template set and adaptable eigenfilter banks in its reconstructional algorithm to highlight the differences between normal and abnormal textures. The proposed novelty detection method achieved a promising level of accuracy and robustness in textured tile abnormality detection tests. We introduced two versions of our algorithm, namely matched-by-eigenvalue (MBE), and matched-by-structure (MBS). The re-structural MBS showed higher performance and flexibility relative to the MBE algorithm, which ordered and matched eigenfilters according to their eigenvalues, i.e. a typical approach. MBS was also rotation invariant and illustrated better performance on larger neighbourhood sizes. MBS achieved 91.46% correct classification in our novelty detection experiments.

We also gathered a survey of texture analysis, classification and defect detection studies during the literature review. A broad range of studies concerning texture analysis were assessed to build up a reliable background for this work. Review of several surface inspection projects and the industrial applications of texture defect detection were also an important part of our reviews. Also, two main texture databases were developed during this work. A set of outdoor scene high resolution images and corresponding labelled patches of different objects, and a categorised tile images data set.

6.2 Concluding Remarks

The aim of this study was the development of more efficient texture classification methods, where computation costs, classification accuracy, and practical specifications were traded off and balanced against each other: Dealing with high resolution images requires fast algorithms to compensate their heavy processing cost. Therefore, we proposed the fast DWHT and H_p/S_p scheme. In industrial applications however, a compromise between accuracy, speed, and practicality was of interest. Therefore, we proposed the high performance GC and novelty detection eigenfilter-based algorithms.

Multi-scale/multi-directional algorithms (e.g. Gabor filters or wavelets) model the basic characteristics of textures. Specifically they can extract repetition and directionality which are two key features of a random texture, thus they can analyse and classify textures accurately. To extract repetition and directionality, they perform an angular and scale decomposition on the texture and generate a multi dimensional detail image space. Feature extraction and other analysis in the detail image space is more efficient than the pattern space. Computationally however, they are complicated and time consuming. On the other hand, fast sequency-based transforms (e.g. Hadamard), are quick and easy to implement, but less accurate in texture classification. We model the characteristics of MSMD texture analysis algorithms in a modified Walsh-Hadamard transform, called DWHT. In the DWHT algorithm, a special purpose rotation and a band-wise analysis integrate the basic angular and scale decomposition properties of MSMD schemes with the ordinary Walsh-Hadamard transform. While Gabor filters and the wavelet transform explicitly decompose the image into angular and scale sub bands, DWHT presents an implicit transform domain decomposition: Sequency harmonics of pixels in different orientations are extracted and categorised into a few low-to-high sequency bands. Nevertheless, the proposed DWHT exhibits the advantages of both the Walsh-Hadamard transform (computation efficiency) and MSMDs (high accuracy) in texture classification. The promising performance of DWHT on a diverse set of texture classification and defect detection applications emphasises its generality. This method can be used for realtime texture analysis in particular. A similar approach can be implemented on other harmonic analysers such as DCT or Haar transform.

The interpretation of the hue and saturation in forms of the proposed chromatic functions, H_p and S_p, also gains both simplicity and the performance of perceptual colour representation. The intention is not developing a real and precise colour space, but extracting significant colour features for pattern classification purposes. Therefore, employing these colour features is reasonable, in particular for realtime and embedded systems.

The comparative study for the detection of abnormalities in randomly textured tiles reveals the advantages and disadvantages of different statistical and signal processing approaches. Also optimisation tests applied on each approach illustrate the substantial role of parameter optimisation on the overall classification performance. With respect to its accuracy and fast execution time, DWHT is the practically favorite method, while GLCM and eigenfiltering provide better precision on the BPNN classifier, at the cost of longer execution times. We also propose the Gabor Composition algorithm to increase the performance of the more feasible KNN classifier. GC decomposes the image into sub-band detail images using a Gabor filter bank, then attempts to highlight the defective regions and their boundaries in a recomposed feature map, using exclusion and Bernoulli's combination techniques. GC decreases the redundancy as well as highlights the defective region in the texture-like resulted feature map, so a simple GLCM algorithm can efficiently extract the normal and abnormal texture features. GC shows excellent classification performance on almost all of the tile types of the data set, which illustrates its scope for the detection of abnormalities on random textures. However, it is again of importance to optimise all the parameters concerning the characteristics of the test texture.

In practical applications, such as the tile industry, novelty detection is more suitable than ordinary classification methods for quality inspection, since it is independent of defective samples. The orthogonality and adaptability of eigenfilters lead us to develop an innovative eigenfilter-based novelty detection scheme. The idea underpinning the proposed method is mutual reconstruction of the test image by its own and the template eigenfilter bank. We expect less success (i.e. larger reconstruction error) in the reconstruction of a defective texture rather than a flawless texture, when a template of flawless samples is used as the reference. The proposed novelty detection scheme illustrates promising results on all types of tile tested. This can suggest the generality of the method in the field of texture abnormality detection.

6.3 Contributions

The main contributions of this study were:

- A novel multi-scale and multi-directional Walsh-Hadamard transform, DWHT, as a fast and accurate method with potential applications in realtime systems.
- Two low cost chromatic features, H_pS_p, based on the definition of hue and saturation in standard colour spaces, efficient for use in colour texture classification.
- A new Gabor Composition based method, GC, devised for detection of abnormalities in random textures.
- An eigenfilter based reconstruction method for tile inspection within a novelty detection framework.

6.4 Further Work

This thesis illustrated effective texture feature extraction and classification techniques which were applied on various test frameworks and procedures. Meanwhile, there are some essential aspects where the methods' performances can be enhanced. We therefore propose the list below as possible outlines for further work.

- **DWHT:**
    - It may be useful to replace the ordinary SOH kernel of DWHT with a slanted Hadamard transform to determine if this can improve the overall performance.
    - It also may be worthwhile to employ and test the Gaussian sequency band envelopes instead of current ideal (hard) separators on both DWHT and DDCT. It is also possible to apply dyadic bandwidths to DWHT-like algorithms.
    - A wavelet-like multi-scale approach instead of current multi-scaling via the band separation method may be of interest. We may down-sample the input texture and apply the DWHT on each stage of the Gaussian pyramid and compare the results with the current method. In this case, there will be no need to separate the sequency bands. Rotation however, still will be performed as before.
- **GC:**
    - Using a GC feature map as the platform, a novelty detection approach could be developed based on measuring the distances between small blocks of the feature map, and a reference set containing the 'normal' feature map blocks.
    - It would also be of interest to develop a GC-based defect localisation method, also utilising the GC feature map.
- **Eigenfilter-based classification and ND scheme:**

  - Applying a multi-scale eigenfilter-based method could be of interest where the input image is down-sampled, then the eigenfilter-based analysis and reconstruction algorithm will be applied on different scales of that multi-scale input pyramid. Although it seems computationally heavy to apply PCA on every single stage of the image pyramid, we may be able to find the optimum stage or stages for a given texture during a training phase.
  - Alternatively, to use diverse mask shapes instead of typical $n \times n$ squares to detect a particular abnormality can be a reasonable option for further eigenfiltering studies.
  - We can attempt to establish a relation between texture characteristics (e.g. coarseness) and the optimum eigenfilter size and subset, applied in eigenfilter-based texture analysis.
  - We can try the MBS method in traditional eigenfilter-based classification schemes. MBS in particular may increase the classification performance of the larger eigenfilters, if it is utilised to order and match detail images instead of the typical eigenvalue-based approach.

- **General:**
  - Texture analysis algorithms usually have several parameters to be tuned (e.g. central frequencies and directional resolution in Gabor filters). Automated optimisation of a texture analyser's parameters with respect to the given texture characteristics, could be an essential development in texture processing. It also can be extended to the automatic selection of the optimum method for analysing (e.g. detection of abnormalities) of a particular texture, again regarding its properties. For instance, automatic selection of either GLCM or Gabor filters considering the test texture, can significantly increase the overall abnormality detection performance.

Bibliography

[1] H. Abdi, D. Valentin, and B. Edelman. *Neural networks*. SAGE Publications, 1999.

[2] F. Ade. Characterization of textures by eigenfilters. *Signal Processing*, 5:451–457, 1983.

[3] F. Ade, N. Lins, and M. Unser. Comparison of various filter sets for defect detection in textiles. In *Proceedings of International Conference on Pattern Recognition, 1984*, volume I, pages 428–431, 1984.

[4] R. Azencott, J. Wang, and L. Younes. Texture classification using windowed fourier filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):148–153, February 1997.

[5] R. Baldrich, M. Vanrell, and J. Villanueva. Texture-colour features for tile classification. In *EUROPTO/SPIE Conference on Colour and Polarisation Techniques in Industrial Inspection*, 1999.

[6] A. Baykut, R. Meylani, S. zdemir, A. Eril, and A. Ertzn. Comparative evaluation of texture analysis algorithms for defect inspection of textile products. In *Proceedings of IEEE SIU, 1997*, volume 2, pages 557–563, 1997.

[7] E. Bayro-Corrochano. Review of automated visual inspection 1983 to 1993. *Proceedings SPIE*, 2055:128–172, 1993.

[8] K. Beauchamp. *Applications of Walsh and related functions*. Academic Press, 1984.

[9] J. Bergen and M. Landy. Computational modelling of visual texture segregation. In M. Landy and J. Movshon, editors, *Computational Models of Visual Processing*, pages 253–271. M. I. T. Press, 1991.

[10] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[11] J. De Bonet. Novel statistical multiresolution techniques for image synthesis, discrimination, and recognition. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1997.

[12] J. De Bonet, P. Viola, and J. Fisher III. Flexible histograms: A multiresolution target discrimination model. In *Image Understanding Workshop 1998*, pages 867–873, 1998.

[13] C. Boukouvalas, J. Kittler, R. Marik, M. Mirmehdi, and M. Petrou. Ceramic tile inspection for colour and structural defects. In *Proceedings of Advanced Materials and Processing Technologies*, pages 390–399, August 1995.

[14] C. Boukouvalas, G. Vernazza, F. De Natale, G. De Toni, J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, P. Le Roy, and R. Salgari. An integrated system for quality inspection of tiles. In *Proceedings of the International Conference on Quality Control by Artificial Vision*, pages 49–54, 1997.

[15] D. Bourgin. Color Space FAQ. Retrieved 1 Feb 2004 from the World Wide Web: http://www.scarse.org/docs/color-faq.html, 1995.

[16] A. Bovik. Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Transactions on Signal Processing*, 39(9):2025–2043, 1991.

[17] A. Boyd, C. Connolly, and A. Starr. Surface defect and texture identification in construction materials. In *6th International Conference in Image Processing and Its Applications*, volume 2, pages 507–510, 1997.

[18] P. Brodatz. *Textures: A photographic album for artists and designers*. Dover Publishing Co., Toronto, Canada, 1966.

[19] C. Chan and G. Pang. Fabric defect detection by fourier analysis. *IEEE Transactions on Industry Applications*, 36(5):1267–1276, 2000.

[20] D. Chetverikov. Measuring the degree of texture regularity. In *Proceedings of International Conference on Pattern Recognition, 1984*, volume 1, pages 80–82, 1984.

[21] D. Chetverikov and A. Hanbury. Finding defects in texture using regularity and local orientation. *Pattern Recognition*, 35(10):2165–2180, 2002.

[22] R. Chin. Automated visual inspection techniques and applications: A bibliography. *Pattern Recognition*, 15(4):343–357, 1982.

[23] A. Clark. *Region classification for the interpretation of video sequences*. PhD thesis, Department of Computer Science, University of Bristol, 1999.

[24] D. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing*, 28(1):4562, 2002.

[25] D. Clausi and M. Jernigan. Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849, 2000.

[26] P. Colantoni. Color Space Transformations. Retrieved 1 Feb 2004 from the World Wide Web: http://colantoni.nerim.net/download/colorspacetransform98.pdf, 2003.

[27] A. Conci and C. Belmiro. A system for real-time fabric inspection and industrial decision. In *14th International Conference on Software Engineering and Knowledge Engineering*, pages 707–714, 2002.

[28] C. Costa and M. Petrou. Automatic registration of ceramic tiles for the purpose of fault detection. *Machine Vision and Applications*, 11:225–230, 2000.

[29] O. Cula and K. Dana. Compact representation of bidirectional texture functions. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1041–1047, 2001.

[30] F. D'Astous and M. Jernigan. Texture discrimination based on detailed measures of the power spectrum. In *Proceedings of International Conference on Pattern Recognition, 1984*, volume 1, pages 83–86, 1984.

[31] E. Davies. *Machine vision: theory, algorithms, practicalities*. Academic Press, 1990.

[32] E. Davies, D. Patel, and A. Johnston. Crucial issues in the design of a real-time contaminant detection system for food products. *Real-Time Imaging*, 1:397–407, 1995.

[33] J. Devaux, P. Gouton, and F. Truchetet. Aerial colour image segmentation by karhune-loeve transform. In *Proceedings of International Conference on Pattern Recognition 2000*, volume 1, pages 309–312, 2000.

[34] A. Drimbarean and P. Whelan. Experiments in colour texture analysis. *Pattern Recognition Letters*, 22(10):1161–1167, 8 2001.

[35] H. Eghbali. Image enhancement using a high sequency ordered Hadamard transform filtering. *Computer and Graphics*, 5:23–29, 1980.

[36] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks, a review. *Pattern Recognition*, 35:2279–2301, 2002.

[37] T. El-Maraghi. An implementation of heeger and bergen's texture analysis/synthesis algorithm. Department of Computer Science, University of Toronto, Retrieved 1 Aug 2001 from the World Wide Web: http://www.cs.toronto.edu/ tem/2522/texture.html, 1997.

[38] J. Escofet, R. Navarro, M. Millan, and J. Pladellorens. Detection of local defects in textile webs using Gabor filters. *Optical Engineering*, 37(8):2297–2307, 1998.

[39] M. Everingham, B. Thomas, T. Troscianko, and D. Easty. Neural-network virtual reality mobility aid for the severely visually impaired. In *Proceedings of the 2nd European Conference on Disability, Virtual Reality and Associated Technologies*, pages 183–192, September 1998.

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

[41] K. Fukunaga. Statistical pattern classification. In *Handbook of Pattern Recognition and Image Processing*, pages 3–32. Academic Press, 1986.

[42] R. Gonzalez and P. Wintz. *Digital image processing*. Addison Wesley, 1987.

[43] S. Grigorescue, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *IEEE Transactions on Image Processing*, 11(10):1160–1167, 2002.

[44] R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–803, May 1979.

[45] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

[46] S. Hayati, J. Balaram, H. Seraji, W. S. Kim, K. Tso, and V. Prasad. Remote surface inspection system. In *Proceedings of IEEE International Conference on Robotics and Automation 1993*, volume 3, pages 875–882, May 1993.

[47] D. He and L. Wang. Textural filters based on the texture spectrum. *Pattern Recognition*, 24(12), 1991.

[48] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 229–238, 1995.

[49] L Hepplewhite. *Computationally Efficient Texture Methods For Classification, Segmentation and Automated Visual Inspection*. PhD thesis, Brunel University, Middlesex, The UK, 1998.

[50] J. Hershey and G. Saulnier N. Al-Dhahir. New Hadamard basis. *Electronics Letter*, 32(5):429–430, 1996.

[51] S. Hickinbotham and J. Austin. Novelty detection in airframe strain data. In *Proceedings of International Conference on Pattern Recognition, 2000*, volume 2, pages 536–539, 2000.

[52] J. Iivarinen, K. Heikkinen, J. Rauhamaa, P. Vourimaa, and A. Visa. A defect detection scheme for web surface inspection. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(6):735–755, 2000.

[53] A. Ikonomopoulos and M. Unser. A directional approach to texture discrimination. In *Proceedings of International Conference on Pattern Recognition, 1984*, volume 1, pages 87–89, 1984.

[54] A. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[55] N. Japkowicz, C. Myers, and M. Gluck. A novelty detection approach to classification. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* pages 518–523, 1995.

[56] B. Jelinek. Neural network application pattern classification problems. Retrieved 1 Sept 2002 from the World Wide Web: http://www.isip.msstate.edu/publications/courses/ece-8443/papers/2001/, 2002.

[57] Y. Jiang and Z. Zhi-Hua. Editing training data for KNN classifiers with neural network ensemble. In *Proceedings of IEEE International Symposium in Neural Networks, 2004*, August 2004.

[58] I. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[59] B. Julesz. Visual pattern discrimination. *IRE Transaction on Information Theory*, 8:84–92, 1962.

[60] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981.

[61] A. Khodaparast and A. Mostafa. On line quality control of tiles using wavelet and statistical properties. In *Proceedings of the 2nd Iranian Conference on Machine Vision and Image Processing*, pages 153–159, February 2003.

[62] D. Kim and D. Cho. Texture segmentation using Walsh spectrum and modified ART2. Department of Computer Science, Ewha womans University, Korea, Retrieved 1 April 2002 from the World Wide Web: http://www.icspat.com/papers/92mfi.pdf, 2000.

[63] J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, and J. Song. Detection of defects in colour texture surfaces. In *Proceedings of IAPR Conference on Machine Vision Applications 1994*, pages 558–567, 1994.

[64] J. Kittler and F. Roli. Foreword on multiple classifier systems. In *Multiple Classifier Systems,* pages 5–6. Springer-Verlag, 2000.

[65] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1995.

[66] R. Kruger, W. Thompson, and A. Turner. Computer diagnosis of pneumoconiosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 4(1):40–49, 1974.

[67] A. Kumar. Neural network based detection of local textile defects. *Pattern Recognition*, 36(7):1645–1659, 2003.

[68] A. Kumar and G. Pang. Defect detection in textured materials using optimized filters. *IEEE Transactions on Systems, Man, and Cybernetics*, 32(5):553–570, 2002.

[69] MIT Media Lab. VisTex: Vision Texture database. Retrieved 1 Feb 2002 from the World Wide Web: http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html, 2002.

[70] M. Landy. Texture perception. In G. Adelman, editor, *Encyclopedia of Neuroscience*. Elsevier North-Holland, 1996.

[71] A. Latif-Amet, A. Ertuzun, and A. Ercil. An efficient method for texture defect detection: sub-band domain co-occurrence matrices. *Image and Vision Computing*, 18:543–553, 2000.

[72] K. Laws. Texture energy measures. In *Image Understanding Workshop, DARPA 1979*, pages 47–51, November 1979.

[73] K. Laws. Rapid texture identification. *SPIE Vol. 238 Image Processing for Missile Guidance*, 238:376–380, 1980.

[74] V. Lebrun. Quality control of ceramic tiles by machine vision. Asian Ceramics: Ceramics manufacturing equipment guide special publication, Retrieved 1 June 2004 from the World Wide Web: http://www.ulg.ac.be/mica/marble/page12.html, 2001.

[75] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *International Conference on Computer Vision*, pages 1010–1017, 1999.

[76] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391, 1995.

[77] X. Liu and D. Wang. Texture classification using spectral histogram. Technical Report OSU-CISRC-7/00-TR17, Department of Computer and Information Science, The Ohio State University, 2000.

[78] S. Livens, P. Scheunders, G. Van de Wouwer, and D. Van Dyck. Wavelets for texture analysis, an overview. In *Proceedings of IPA 1997 Conference, IEE Pub. No. 443*, volume 1, pages 581–585, 1997.

[79] F. Lopez, F. Acebron, J. Valiente, and E. Perez. A study of registration methods for ceramic tile inspection purposes. In *Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volume 1, pages 145–150, 2001.

[80] T. Maenpaa, M. Pietikainen, and T. Ojala. Texture classification by multi-predicate local binary pattern operators. In *Proceedings of International Conference on Pattern Recognition, 2000*, 2000.

[81] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[82] V. Manian and R. Vasquez. Approaches to color- and texture-based image classification. *Journal of International Society for Optical Engineering (SPIE)*, 41(7):1480–1490, 2002.

[83] M. Markou and S. Singh. Novelty detection: a review. *Signal Processing*, 83(12):2481–2521, 2003.

[84] J. Penaranda Marques, L. Briones, and J. Florez. Color machine vision system for process control in the ceramics industry. *Journal of International Society for Optical Engineering (SPIE)*, 3101:182–192, 1997.

[85] The MathWorks. Matlab online reference. Retrieved 1 Feb 2004 from the World Wide Web: http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml.

[86] M. Melloul and L. Joskowicz. Segmentation of microcalcifications in x-ray mammograms using entropy thresholding. In *Proceedings of the 16th International Congress on Computer-Assisted Radiology and Surgery*, pages 490–495, 2002.

[87] M. Mirmehdi and R. Perissamy. Perceptual image indexing and retrieval. *Journal of Visual Communication and Image Representation*, 13(4):460–475, 2002.

[88] MIT Vision and Modelling Group. MeasTex Texture Data Suite. Retrieved 1 Feb 2002 from the World Wide Web: http://www.cssip.uq.edu.au/staff/meastex/meastex.html, 1998.

[89] A. Monadjemi, B. Thomas, and M. Mirmehdi. Experiments on high resolution images towards outdoor scene classification. In *Proceeding Computer Vision Winter Workshop, CVWW 2002*, volume 1, pages 325–334, 2002.

[90] O. Nestares, R. Navarro, J. Portilla, and A. Tabernero. Efficient spatial-domain implementation of a multiscale image representation based on Gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, 1998.

[91] T. Newman and A. Jain. A survey of automated visual inspection. *Computer Vision and Image Understanding*, 61(2):231–262, 1995.

[92] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distribution. *Pattern Recognition*, 29(1), 1996.

[93] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 2002.

[94] M. Partio, B. Cramariuc, M. Gabbouj, and A. Visa. Rock texture retrieval using gray level co-occurrence matrix. In *Proceedings of 5th Nordic Signal Processing Symposium*, 2002.

[95] O. Pichler, A. Teuner, and B. Hosticka. A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition*, 29(5):733–742, May 1996.

[96] J. Portilla and E. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1), 2000.

[97] W. Pratt. *Digital image processing*. John Wiley and Sons, 1991.

[98] T. Randen and J. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.

[99] A. Rao and G. Lohse. Identifying high level features of texture perception. *Computer Vision, Graphics and Image Processing*, 55(3):218–233, 1993.

[100] T. Reed and J. Du Buf. A review of recent texture segmentation and feature extraction techniques. *CVGIP Image Understanding*, 57(3):359–372, 1993.

[101] A. Rosenfeld and A. Kak. *Digital picture processing*. Academic Press, 1982.

[102] D. Rumelhart, B. Widrow, and M. Lehr. The basic ideas in neural networks. *Communications of the ACM*, 37(3), 1994.

[103] H. Sari-Sarraf and J. Goddard. Vision system for on-loom fabric inspection. *IEEE Transactions on Industry Applications*, 35(8):1252–1259, 1999.

[104] W. Sarle. What are cross-validation and bootstrapping? FAQs, Retrieved 1 Sept 2003 from the World Wide Web: http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html, 2003.

[105] C. Schmid. Constructing models for content-based image retrieval. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45, 2001.

[106] C. Setchell and N. Campbell. Using colour texture features for scene understanding. In *7th. International Conference on Image Processing and its Applications*, pages 372–376. Institution of Electrical Engineers, 1999.

[107] E. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proceedings of the IEEE International Conference on Image Processing, ICIP'98,* volume 1, pages 62–66, 1998.

[108] M. Singh and S. Singh. Spatial texture analysis: A comparative study. In *Proceedings of International Conference on Pattern Recognition, 2002,* volume 1, pages 676–679, 2002.

[109] S. Singh and M. Markou. An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):396–407, 2004.

[110] G. Smith and I. Burns. Measuring texture classification algorithms. *Pattern Recognition Letters*, 18(14):1495–1501, December 1997.

[111] M. Smith and R. Stamp. Automated inspection of textured ceramic tiles. *Computer in Industry*, 43:73–82, 2000.

[112] H. Sohn, K. Worden, and C. Farrar. Novelty detection using auto-associative neural network. In *Proceedings of 2001 ASME International Mechanical Engineering Congress and Exposition*, November 2001.

[113] M. Sonka, V. Hlavac, and R. Boyle. *Image processing analysis and machine vision*. International Thomson Computer Press, 1996.

[114] J. Strand and T. Taxt. Local frequency features for texture classification. *Pattern Recognition*, 27(10):1397–1406, 1994.

[115] A. Tolba and A. Abu-Rezeq. A self-organizing feature map for automated visual inspection of textile products. *Computers in Industry*, 32:319–333, 1997.

[116] M. Tuceryan and A. Jain. Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248. World Scientific, 1998.

[117] M. Unser. Local linear transforms for texture analysis. In *Proceedings of International Conference on Pattern Recognition, 1984*, volume 1, pages 1206–1208, 1984.

[118] M. Unser. Local linear transform for texture measurements. *Signal Processing*, 11:61–79, 1986.

[119] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, 1995.

[120] M. Unser and F. Ade. Feature extraction and decision procedure for automated inspection of textured materials. *Pattern Recognition Letters*, 2(3):185–191, 1984.

[121] J. Valiente, F. Acebron, and F. Lopez. A ceramic tile inspection system for detecting corner defects. In *Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volume 2, pages 213–218, 2001.

[122] K. Valkealahti. Texture classification with single- and multiresolution co-occurrence maps. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(4):437–452, 1998.

[123] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of European Conference on Computer Vision 2002*, volume 3, pages 255–271, 2002.

[124] J. Wang and A. Asundi. A computer vision system for wineglass defect inspection via gabor-filter-based texture features. *Information Sciences*, 127(3-4):157–171, 2000.

[125] A. Watson. Image compression using the discrete cosine transform. *Mathematica Journal*, 4(1):81–88, 1994.

[126] T. Weldon, W. Higgins, and D. Dunn. Gabor filter design for multiple texture segmentation. *Optical Engineering*, 35(10):2852–2863, 10 1996.

[127] D. Williams and B. Julesz. Filters versus textons in human and machine texture discrimination. In *Neural Networks for Perception*, volume 2, pages 145–175. Academic Press Inc., 1992.

[128] K. Worden. Structural fault detection using a novelty measure. *Journal of Sound and vibration*, 201(1):85–101, 1997.

[129] R. Yarlagadda and J. Hershey. *Hadamard matrix analysis and synthesis*. Kluwer Academic Publishers, 1997.

[130] Y. Zhang, Y. Sun, H. Sari-Sarraf, and M. Abidi. Impact of intensity edge map on segmentation of noisy range images. *Proceedings SPIE*, 3958:260–269, 2000.

[131] S. Zhu, C. Guo, Y. Wu, and Y. Wang. What are textons? In *Proceedings of the 7th European Conference on Computer Vision, 2002*, volume 4, pages 793–807, 2002.

Appendix A

Colour Spaces

A.0.1 HLS Colour Space

HLS directly implements simple definitions of Hue, Luminance and Saturation to build up its 3D colour space. Considering a colour disk of all spectrums, Hue is an angle in a colour disk, Luminance is an average of R,G, and B, and saturation is the distance from its central axis. Figure A.1 shows a graphical representation of those informal definitions. We applied expressions below for RGB-to-HLS conversion. There are several other RGB-to-HLS algorithms available [15, 97], however they are only slightly different in details.

$$\begin{cases} Hue = \begin{cases} \frac{G-B}{Max(R,G,B)-Min(R,G,B)} & \text{if R is Maximum} \\ \frac{R-B}{Max(R,G,B)-Min(R,G,B)} & \text{if G is Maximum} \\ \frac{R-G}{Max(R,G,B)-Min(R,G,B)} & \text{if B is Maximum} \end{cases} \\ Luminance = \frac{Max(R,G,B)+Min(R,G,B)}{2} \\ \\ Saturation = \frac{Max(R,G,B)-Min(R,G,B)}{Max(R,G,B)+Min(R,G,B)} \end{cases} \tag{A.1}$$

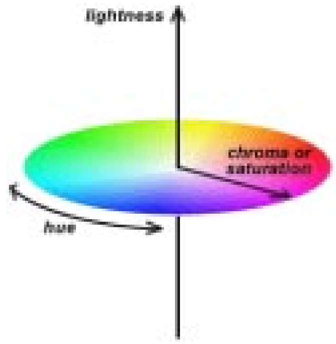


Figure A.1: HLS Colour disk.

A.0.2 *Lab* Colour Space

Lab was introduced in 1976 by CIE (Commission Internationale de l'Eclairage) to provide a more uniform and accurate model of colours. In this color space you use three components: *L* is the luminance, *a* and *b* are respectively red/blue and yellow/blue chromatic data. This color space is defined regarding the intermediate CIE-XYZ color space.

$$\begin{cases} L = & \begin{cases} 116(\frac{Y}{Y_0})^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_0} > 0.008856 \\ 903.3(\frac{Y}{Y_0}) & \text{if } \frac{Y}{Y_0} \leq 0.008856 \end{cases} \\ a = & 500[f(\frac{X}{X_0}) - f(\frac{Y}{Y_0})] \\ b = & 200[f(\frac{Y}{Y_0}) - f(\frac{Z}{Z_0})] \end{cases} \tag{A.2}$$

with,

$$f(t) = \begin{cases} t^{(\frac{1}{3})} & \text{if } \frac{Y}{Y_0} > 0.008856 \\ 7.787 \times t + \frac{16}{116} & \text{if } \frac{Y}{Y_0} \leq 0.008856 \end{cases}$$

and,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad X_0 = 0.9505 \ , \quad Y_0 = 1 \ , \quad Z_0 = 1.0887 \ ,$$

Lab parameters (e.g. XYZ matrix and $\{X_0, Y_0, Z_0\}$ constants) depend on the chromaticity of the reference white point. The expressions above are according to the $D65$ reference point which we employ in this study [15, 26, 97].

Lab chromatic features, *a* and *b*, have direct relations with hue and chroma (i.e. saturation):

$$\begin{cases} Hue = \ \arctan(\frac{b}{a}) \\ Chroma \ (Saturation) = \ \sqrt{a^2 + b^2} \end{cases} \tag{A.3}$$