

Defect Detection in Random Colour Textures

Xianghua Xie



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.

February 2006

45,000 words

Abstract

Automated surface inspection for quality control has largely employed graylevel image processing techniques, for example in textile and wafer inspection. There are rising demands in the quality control industry for colour analysis to fulfil its vital role in visual inspection, e.g. in ceramic tile manufacturing. This thesis is concerned with developing texture analysis techniques in application to the detection of abnormalities in colour texture surfaces, in particular ceramic tile surfaces on which patterns are regularly of a random nature. These abnormalities can be divided into two categories: colour tonality defects and textural abnormalities.

Colour tonality refers to global chromatic appearance of a surface. Its variation from surface to surface may be understated, but becomes significant once the surfaces are placed together. In the first part of this thesis, a multidimensional histogramming method is presented to detect subtle tonality changes by incorporating local chromatic information into global chromatic characteristics. PCA is used to reveal the nonlinear noise interference introduced by the imaging system. Vector directional processing and reference eigenspace feature selection are proposed to obtain salient colour tonality representation. The method is evaluated on a dataset with groundtruth, and compared against an existing state-of-the-art method.

Textural quality inspection involves the detection and localisation of various chromatic and textural imperfections. This thesis suggests that although some textures have a random appearance, there are textural primitives that govern the global appearance. A novel two-layer generative model is proposed to represent an image or a family of images. In this model, random (or regular) texture images in the first layer are assumed to be generated from a collection of texture exemplars, or texems, in the second layer. A bottom-up texem generation process is proposed based on pixel neighbourhoods. This local contextual analysis using texems is applied to a large set of graylevel ceramic tile images, in which graylevel analysis is sufficient to detect textural abnormalities. Then, different schemes are explored to extend graylevel texems to colour images. This results in two different formulations and inference procedures with different computational complexity.

Spatial detection and localisation accuracy of the proposed methods is measured and compared using texture collage images. The texem model is also compared against the multiscale, multidirectional Gabor filter for defect detection. Both the colour tonality inspection and textural defect detection methods are implemented in novelty detection schemes to cope with the variety and unpredictable nature of defective samples.

An application of the colour texem model to medical image analysis, involving the detection of abnormalities in tympanic membrane images, is briefly discussed. Finally, a further extension of the texem model to perform segmentation on inkjet printed ceramic tile surfaces is presented.

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:

To my parents and my wife

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Majid Mirmehdi, for his guidance, inspiration, discourse, and encouragement throughout my research, and for his enormous patience at the stage of my writing-up. Without his effort, it would have been impossible for me to finish my study. I am also grateful to my second supervisor, Prof. Barry Thomas, who passed away before I completed, for his inspiration and conversation.

I am also grateful to Universities UK for the Overseas Research Scholarship (ORSAS) which I received, to European Commission project MONOTONE for financial support, and to the Department of Computer Science for the excellent research environment and facilities. Thanks also to Fernando López for the colour tonality tile dataset, and the Avon Longitudinal Study of Parents and Children (ALSPAC) Bristol for the tympanic membrane images.

I would also like to thank all those in the department who have made my time in Bristol so enjoyable: Annie Yao, James Revel, Ronghua Yang, Amir Monadjemi, Michael Evans, Roger Lo, Ali Osareh, and Pedro Carmona.

Finally, I would like to thank my parents and my wife for their encouragement and support throughout my time at university.

Contents

List of Figures	iv
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Overview	3
1.2.1 Colour tonality inspection	3
1.2.2 Textural defect detection	5
1.3 Contributions	7
1.4 Thesis Layout	8
1.5 Notation	9
2 Background	10
2.1 Introduction	10
2.2 Textural Defect Detection	11
2.2.1 Statistical approaches	11
2.2.2 Structural approaches	15
2.2.3 Signal processing approaches	18
2.2.4 Model based approaches	22
2.2.5 Discussion	24
2.3 Colour Tonality Inspection	26
2.4 Colour Texture Analysis	28
2.5 Classification and Novelty Detection	30
2.5.1 Visual inspection via supervised classification	30
2.5.2 Visual inspection via novelty detection	31
2.6 Conclusions	32
3 Image Datasets and Pre-processing	34
3.1 Image Datasets	34
3.2 Accuracy, Sensitivity, and Specificity	38

3.3	Image Pre-processing	41
3.3.1	Luminance correction via histogram specification	41
3.3.2	Parametric luminance correction	42
3.3.3	Non-parametric luminance correction	44
3.4	Summary	46
4	Colour Tonality Inspection	47
4.1	Introduction	47
4.2	Colour Shade Inspection	49
4.2.1	Noise analysis	49
4.2.2	Vector directional median and LCV	51
4.2.3	Distribution comparison in eigenspace	53
4.3	Method Summary and Comments	56
4.4	Implementations and Results	61
4.5	Conclusions	65
5	Defect Detection using Local Contextual Analysis	67
5.1	Introduction	67
5.2	Texture Exemplars for Defect Detection on Random Texture	69
5.2.1	The texem model	69
5.2.2	Learning texems	70
5.2.3	A simple multiscale approach	73
5.2.4	Novelty detection	74
5.2.5	Method summary and comments	75
5.2.6	Experimental results	78
5.2.7	Summary	85
5.3	Defect Detection using Graylevel Texems in Colour Image Eigenchannels	85
5.3.1	Computing the reference eigenspace	87
5.3.2	Generating multiscale texems in image eigenchannels	87
5.3.3	Novelty detection and defect localisation	88
5.3.4	Experimental results	89
5.3.5	Conclusions	96
6	Colour Texems	97
6.1	Introduction	97
6.2	Colour Texem Model	98
6.2.1	Learning colour texems	98
6.2.2	Training and testing	103
6.3	Results: Ceramic Tile Application	105

6.4	Results: Evaluation using VisTex Collages	111
6.5	Alternative Applications	115
6.5.1	Detecting defects in tympanic membrane images	116
6.5.2	Ink-dot image segmentation	118
6.6	Summary	122
7	Conclusions and Future Work	124
7.1	Conclusions	124
7.2	Contributions	127
7.3	Future Work	128
	Bibliography	132
A	EM Algorithm for Graylevel Texems using Gaussian Mixture Model	145
B	List of Publications	149

List of Figures

1.1	Textural defects - first row: a typical textile texture with normal and defective samples. The burl defect is circled in the last image; second row: a normal tile image and an abnormal tile image with missing print. The defective region is highlighted in the last image of the row.	2
1.2	Example ceramic tiles with different colour shades - from left: The first two images in each row belong to the same colour shade, the last one is an example of off-shade. Images in the first row are plain-coloured with uniform appearance. Images in the second row are fixed pattern with marble textures. Images in the final row also decorated with marble textures but are visually random. The colour shade difference shown in the last row is much more subtle than those of the first two rows.	4
1.3	Examples of textural defects on ceramic tile surfaces (different families of tiles) - first row: normal samples; second row: defective samples with ink-drops, a print error, and a cloud of pin holes (from left to right).	6
2.1	Calculating LBP code and a contrast measure.	15
2.2	A three-level generative model: an image I is a linear addition of some image bases selected from a base dictionary Ψ , including Gabor and Laplacian of Gaussians. The base map is further generated by a smaller number of textons selected from a texton dictionary Π	17
2.3	A star texton configuration.	18
2.4	The frequency response of the dyadic bank of Gabor filters. The maximum amplitude response over all filters is plotted. Each filter is represented by one centre-symmetric pair of lobes in the illustration. The axes are in normalised spatial frequencies.	20
2.5	Epitome - from left: Original colour image, its 32×32 epitome, and its 16×16 epitome.	24
2.6	A flow chart of the colour histogram-based colour tonality inspection method. . .	27

3.1	Colour shade examples from the COLSHADE dataset - left column: tile images with standard normal colour shade; middle column: tile images with minor colour shade problem; right column: tiles with relatively larger colour shade off-set.	35
3.2	Examples of grayscale tile images with textural defects. The left image of each pair of images is a normal sample, while the right one is defective - from top left to bottom right: a random texture with a print error, a fine sand-like texture with physical defects, a marble texture with a small crack, a light texture with a print error, a cloud-like texture with pin holes, a marble texture with broken corner, a miss-registration problem, and a pattern irregularity issue in a grid pattern.	37
3.3	Examples of colour tile images with various textural and/or chromatic defects. The left image of each pair is a normal sample, while the right one is abnormal - from top left to bottom right: a cloud-like texture with a diffused defective region, a subtle printing error in the bottom right corner in an oriented texture, a major printing error along the edge of a tile decorated with dense patterns, several defective regions introduced by alien ink during printing, a thin and long crack across a random texture, a print error in a sand-like random texture, a cluster of pin holes in a marble texture, a missing print defect in a busy pattern, a print defect in a marble texture, and a few physical bumps in a lightly textured tile image.	39
3.4	ViTex sample images (captured from materials such as foods, bark, water, fabric, sand, and brick).	40
3.5	The complete set of collage images used for novelty detection evaluation.	40
3.6	Luminance correction using histogram specification - from left: original defect-free tile image and its corrected version; original defective sample and its luminance corrected image.	42
3.7	A white tile with nearly uniform reflectance captured by the imaging system and its horizontal intensity profiles across the tile (the median values at each pixel position are plotted in red curve).	43
3.8	Cosine-fourth based correction - left: The blue line is the median profile of all the horizontal intensity profiles of the white surface, and the red curve is the Chi-square non-linear fitting result; right: The red line is the flattened cosine-fourth curve and the blue one is the corrected median intensity profile.	44
3.9	Luminance correction using a template median profile - curve in blue colour is the luminance profile before correction; curve in red colour is the profile after luminance compensation.	45

3.10	Luminance correction - left: A plot of the average intensities of those 200 images of a uniform coloured tile captured with no change in the imaging system before (in blue colour) and after (in red colour) luminance correction; right: A plot the standard deviations across the whole image before (in blue colour) and after (in red colour) correction.	45
3.11	Cosine fourth based and template profile based luminance correction - First row: luminance correction on a uniform coloured surface. From left: Original tile image (a dark strip across the tile is indicated by a red arrow), cosine fourth based correction result, and a median template profile based correction result. Second row shows the results on a highly textured tile surface.	46
4.1	Two examples of ceramic tiles with different colour shades from dataset COL-SHADE - from left: The first two images in each row belong to the same colour shade, the last one in each row is an example of off-shade.	48
4.2	Image noise analysis showing the three eigenchannels. The noise is highly visible in the third channel. The images have been scaled for visualisation purposes. . .	50
4.3	Perspective representation of the RGB colour cube.	52
4.4	Refined LCV selection - (a) and (b) show two clusters of vectors with similar directionalities; (c) and (d) show the selected LCVs (in red) according to [161]; (e) and (f) show the selected LCVs (in red) using the proposed method.	53
4.5	Vector directional smoothing using a 5×5 window - top row: original tile image and its smoothed version; bottom row: enlarged patches (marked in white from the top row) showing detail of the smoothing effect.	54
4.6	Flow chart of proposed method.	56
4.7	Image reconstruction - top: the original image, its reconstruction, and their MSE difference - bottom: the three eigenchannels of the reconstructed tile. The last channel shows texture structure, instead of being dominated by noise (cf. Figure 4.2).	58
4.8	Another example of tile image - top row: The original tile image, the reconstructed tile image, and the difference between them using MSE - middle row: the three eigenchannels of the original tile image - bottom row: the three eigenchannels of the reconstructed tile image.	59
4.9	Reconstruction of an off-shade tile image - top row: The original off-shade tile image, the reconstructed tile image based on an eigenspace derived from a normal shade tile image, and the difference between them using MSE - middle row: the three eigenchannels of the original off-shade tile image - bottom row: the three eigenchannels of the reconstructed tile image.	60
4.10	Testing results using eigenspace features with various window sizes.	64

4.11	Colour tonality results comparison - A graphical plot of testing results using RGB colour histogram (RGB), $L^*a^*b^*$ colour space (LAB), vector directional smoothing (VDS), and eigenspace features (ESF).	65
5.1	Example marble tiles from the same family whose patterns are different but visually consistent.	70
5.2	Twelve 5×5 texem means extracted from the top-left image in Figure 5.1. (Images are enhanced for better visualisation.)	73
5.3	An illustration of the two-layer structure of the texem model and its bottom-up learning procedure.	76
5.4	Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 5$, the joint contribution of all resolution levels, and the final defective regions superimposed on the original image.	79
5.5	Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 5$, the joint contribution of all resolution levels, and the final defective regions superimposed on the original image.	79
5.6	Defect detection - first row: original images, second row: superimposed defective regions on original images, from left - surface defect, small bump, and missing print.	80
5.7	Defect detection - first row: original defective images, second row: superimposed defective regions on original images, from left - small material bumps, misprints, and misprint along the whole right edge.	80
5.8	Defect detection - first row: original defective images, second row: superimposed defective regions on original images, from left - printing defect, single surface bump, and a cluster of bumps.	81
5.9	Defect detection - first row: original images from different types of tiles, second row: superimposed defective regions on original images, from left - small missing print, misprint in the top-right and bottom-right corners, and three pin holes; third row: another three example images also from different tile families, last row: superimposed defective regions on original images, from left - broken corner, structural defects with complex shapes, and a thin crack and a small surface undulation.	82
5.10	Detecting defects in regular patterns - first column: original defective image and superimposed defective regions; second column: closeup views of normal and abnormal dot patterns; third and fourth columns: two defective samples of a different regular texture with pattern irregularities and superimposed defective areas. . . .	83

5.11	Performance comparison - first column: two samples with large and subtle defects; second column: novelty detection results using epitomes - which produced many false positives or failed to locate the true defects; third column: novelty detection results using texems which successfully detected all the true defects.	84
5.12	Flow chart of the proposed method.	86
5.13	Novelty detection comparison on a collage of 2 texture images - first row: Original collage image, novelty detection result in RGB channels, and novelty detection result in image eigenchannels; second row: RGB individual channels; third row: Novelty detection results in each corresponding RGB channel; fourth row: Image eigenchannels of the collage image; last row: Novelty detection results in each individual eigenchannel.	90
5.14	Novelty detection comparison on a collage of 5 texture images - first row: Original collage image, novelty detection result in RGB channels, and novelty detection result in image eigenchannels; second row: RGB individual channels; third row: Novelty detection results in each corresponding RGB channel; fourth row: Image eigenchannels of the collage image; last row: Novelty detection results in each individual eigenchannel.	91
5.15	Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 4$, and the final defective regions superimposed on the original image.	94
5.16	Defect localisation- detecting subtle pin holes as shown in the first two columns, and clusters of pin holes as shown in the last column.	94
5.17	Defect localisation - detecting missing print (first column), physical defect (second column), and printing error (last column).	95
6.1	Six 9×9 texems extracted from the images shown in Fig. 5.1. First row: the means of texems. Second row: the corresponding covariance images (Images are enhanced for viewing purposes).	98
6.2	Eight 7×7 texems extracted from the image on the left. The first two rows on the right side show the μ_k , and the last two rows show the corresponding ω_k	103
6.3	Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, 3, 4$, and the final defective regions superimposed on the original image.	106
6.4	Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 4$, and the final defective regions superimposed on the original image.	106

6.5	Defect localisation - The examples given in the top row show a granite-like random texture. Defects shown in the next row include various sizes of print errors. The next three images show a cloud-like random texture with loosely placed texture primitives. Defects shown in the last row, from left to right, include cracks, surface defect, and misprints.	108
6.6	Defect localisation - The examples given in the top row are from the same tile family richly decorated with macro-structure patterns. Defects shown in the next row, from left to right, include print error, missing print, and thin surface crack. The next three images show a texture with oriented patterns. Defects shown in the last row, from left to right, include misprint, missing print, and subtle surface crack.	109
6.7	Defect localisation - The first row shows example images from three different tile families with different chromato-textural properties. Defect shown in the next row, from left to right, include misprint, dimps, and bumps. The third row shows graylevel examples from three different tile families. Defects shown in the last row, from left to right, include surface hole, missing print, and surface bumps. . .	110
6.8	Defect localisation comparison: left column - original textures with print errors, middle column - results using graylevel texems, right column - results using colour texems.	111
6.9	Novelty detection comparison on collage images - see text for details.	113
6.10	A comparison plot for Escofet's Gabor filter bank based method, graylevel texem analysis in RGB channels, graylevel texem analysis in image eigenchannels, and colour texem analysis without factorisation of data.	114
6.11	Snake segmentation and colour normalisation - first row: Two normal and two abnormal TM images; second row: Snake segmentation results on TM images to localise TM regions; third row: Colour normalisation - from left to right: reference TM without highlights, pre and post-normalisation of a normal and a defective TM.	116
6.12	Defect localisation - from top left: initial TM image, after segmentation and colour normalisation, defect probability maps at all 4 scales, fused defect map, and detected defects (and saturated highlights).	117
6.13	Defect localisation examples.	117
6.14	Flow chart of the Tympanic Membrane abnormality detection.	118
6.15	One-dimensional analogy of the quadtree structure (empty circles denote leaf nodes).	119
6.16	Unsupervised ink dot image segmentation - first row: original microscopic ink dot image with one ink pattern and the hand-labelled result; second row: the colour texem based segmentation result and the error map; third row: original microscopic ink dot image with two ink pattern and the hand-labelled result; fourth row: the colour texem based segmentation result and the error map.	120

7.1	Unsupervised image segmentation.	130
7.2	The patch size on the region boundary is preferred to be smaller such that only one type of texture is included.	130

List of Tables

2.1	Inexhaustive list of textural defect detection methods	12
2.2	Some histogram similarity measurements.	13
3.1	Colour shade tile dataset: COLSHADE.	36
3.2	Grayscale tile images for defect detection: GRAYSET.	36
3.3	Colour tile images for defect detection: COLOURSET	38
4.1	Testing results of colour histogram methods (values are %)	61
4.2	Testing results of colour histogram methods in $L^*a^*b^*$ space (values %).	62
4.3	Testing results of vector directional smoothing (values are %)	62
4.4	Testing results of comparison in feature eigenspace (values are %)	63
5.1	GRAYSET results using graylevel texems (values are %)	83
5.2	Novelty detection comparison using graylevel texems in image RGB channels and image eigenchannels (values are %).	92
5.3	COLOURSET results using graylevel texems in image eigenchannels (values are %)	95
6.1	COLOURSET results using colour texems (values are %)	111
6.2	Novelty detection comparison on VisTex collage images (values are %).	112
6.3	Testing results of ink dot image segmentation accuracy (values are %)	121

Chapter 1

INTRODUCTION

One of the essential benefits of computer vision is the reduction or replacement of human labour. Amongst many, visual surface inspection has been one of the major applications of computer vision since the early 1980s in order to reduce tedious and laborious human effort. In this thesis, we investigate and develop colour texture analysis techniques to improve the efficiency of the process of surface inspection.

1.1 Motivation

Computer vision techniques have been used in detecting defects or imperfections on a variety of surfaces, such as textile [143, 84], wood [118, 151], steel [184], semiconductor [197], wafer [186], paper [99], ceramic tiles [79, 110, 155, 17, 40, 114], meat [149], leather [182], aircraft [53], and even curved and complex surfaces [86]. The inspection is concerned with identifying regions or surfaces of products that deviate from normal samples according to certain criteria, e.g. pattern regularity or colour.

The work in this thesis arises from MONOTONE [116, 172], which was a European Commission (EC) funded project investigating the overall manufacturing process in the ceramic tile industry. The ceramic tile manufacturing process is highly automated with the exception of the visual inspection stages. Human intervention is still required to keep stringent quality control. Manual visual inspection takes place in several stages, such as post-printing and post-firing (or baking the tile biscuit). This involves identifying a variety of cosmetic and physical imperfections, such as surface cracks, corner chips, printing smudges, holes, undulations, and chromato-textural irregularities. One other important, but less immediately obvious, defect is the uniformity of the chromatic properties of the final tile surface, known as colour shade uniformity.

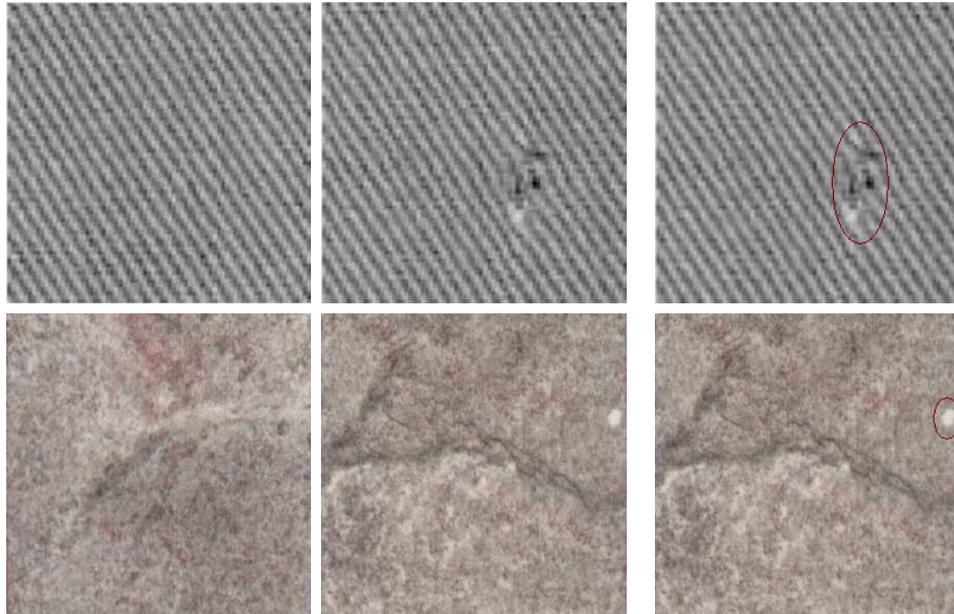


Figure 1.1: Textural defects - first row: a typical textile texture with normal and defective samples. The burl defect is highlighted in red in the last image (*images are adapted from [12]*); second row: a normal tile image and an abnormal tile image with missing print. The defective region is highlighted in the last image.

Ceramic tile production has also been modernised so that it can produce almost any natural or artificial textures. Unlike some materials, such as textiles, steel, and wafers that display complex patterns but appear visually regular on a large scale, printed ceramic tiles may display very complex patterns that are random in appearance. Detecting subtle defects, such as small printing defects and chromatic non-uniformity, on such material surfaces turns out to be rather difficult [79]. For example, in Figure 1.1, the textile image has a dominant and regular periodic pattern, and finding the defect is relatively easy. However, the printed tiles, in the second row, exhibit random appearance with macro-structures. The random pattern, although from the same family, appears different from one surface to another. The thin and long macro-structures in both tile images are not defects, but the light missing print region on the right side is indeed a defect. This defect only occupies a small area of the entire tile image and is potentially much more difficult to detect.

Another notable difference compared to other material surfaces is that ceramic tiles are usually richly decorated with colours. Visual surface inspection tasks are often adequately dealt with using graylevel images captured by monochrome cameras. However, due to increasing processing power and availability of relatively inexpensive colour cameras, there are good prospects for more accurate inspection using colour when appropriate. Furthermore, some defects are chromatic defects by nature as mentioned earlier, so the use of colour is not only justified but of paramount importance.

One approach to defect detection is to first collect a sufficient number of normal and abnormal samples. Image analysis is then performed on all the data resulting in separate feature sets describing normal and abnormal samples. Then, an unseen sample is analysed and classified into known categories. This approach has been widely practised in visual inspection, e.g. [2, 81, 131]. However, in some applications, defects are unpredictable. For example, in ceramic tile production, chromato-textural properties of the final product can be affected by a variety of internal and external factors that are difficult to control, such as variations in colour ink pigments, humidity, and temperature. These effects can lead to a variety of local or global defects, chromatically, texturally, or structurally, which are very difficult to predict due to the nonlinearity in the production chain and complex mechanical and chemical processes at the printing and firing stages. In these circumstances, the ability to detect *novelties* is more desirable than traditional supervised classification techniques which require a good knowledge of both normal and defective samples.

In summary, the main objective of this work is to propose reliable colour texture analysis techniques to detect a variety of surface defects, including colour tonality and various textural imperfections, in a novelty detection framework. The techniques should be able to discriminate subtle colour shade differences and to localise defective regions. Furthermore, it is important that this can be carried out on random textures.

1.2 Overview

1.2.1 Colour tonality inspection

The colour tonality problem, a *global* chromatic defect, has been largely neglected in the literature until very recently. Also known as colour shade [15, 191], it is concerned with chromatic characteristics in terms of overall visual impression across a tile and from one tile to the next. It is also a significant production quality factor.

Figure 1.2 demonstrates three different families of tiles with a colour tonality problem. The colour printed on the last tile surface of each row is slightly different from those on the first two. Any changes in the colour shade, however subtle, will still become significant once the tiles are placed together. Such assessment of tile surfaces for constant colour tonality is one of the key problems in the manufacturing process; it is also tiresome and difficult when inspection is carried out manually. Note the problem is compounded when the surface of the object is not just plain-coloured, but textured. Indeed, the human inspector finds it increasingly difficult to inspect colour tonality problems as the complexity of the (random) patterns increase, and manufacturers have long sought to automate the process. The images in the second row in Figure 1.2 are fixed patterns with a

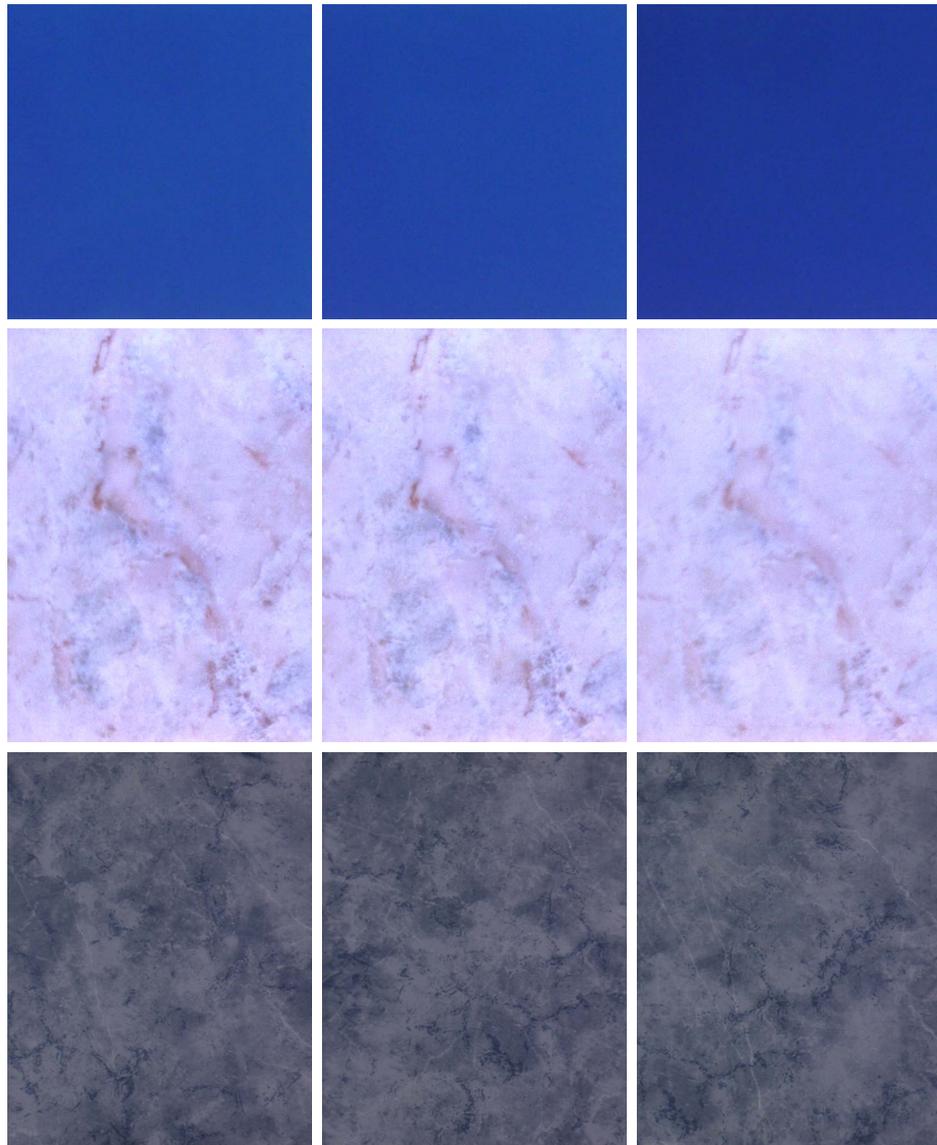


Figure 1.2: Example ceramic tiles with different colour shades - from left: The first two images in each row belong to the same colour shade, the last one is an example of off-shade. Images in the first row are plain-coloured with uniform appearance. Images in the second row are fixed pattern with marble textures. Images in the final row also decorated with marble textures but are visually random. The colour shade difference shown in the last row is much more subtle than those of the first two rows.

relatively large shade difference. However, normal and abnormal randomly textured colour shades can exhibit only very subtle differences not easily visible to the human eye. The final row in Figure 1.2 shows a particularly difficult example.

Existing methods for tonality inspection include wavelet analysis in various image channels [97, 4]

and statistical colour blob analysis in segmented image stacks [6]. Histogram statistics, such as first and second order moments [128], and even higher order moments [95, 96] are also used in colour shade analysis. In [15], colour tonality comparison was treated as colour histogram distribution comparison. A more detailed review is given in Chapter 2.

The first part of this thesis is concerned with developing methods to differentiate subtle shade differences in random textures. A method which combines local chromatic features with global features in characterising colour shades is proposed. The features are then refined using Principal Component Analysis (PCA) and vector order processing to obtain salient and more robust chromatic features for shade inspection. A comparative study with experimental results on 456 tile images from three different manufacturers is presented.

1.2.2 Textural defect detection

There are a variety of chromato-textural defects, such as missing print, printing smudges, mis-registration, and pattern irregularity, as well as physical damage, e.g. holes, cracks, undulations, broken corners, and scratches. All these types of defects exhibit different textural properties, *locally*, against normal background texture. Thus, all these surface defects are referred to as local textural defects even if they are physically introduced. Figure 1.3 shows several types of textural defects in three different families of tile surfaces.

As mentioned earlier, defect detection is a process of determining if a surface deviates from a given set of specifications. It involves classifying the surfaces, not just globally, but going further to localise the defective regions. Also, in visual inspection all the defective surfaces are usually considered as a single class, defect class, as it is a positive verification process. However, those defective samples are likely to form multiple classes in terms of surface properties and would need to be further categorised[†]. Localising defects on textured surfaces should also be viewed as different from texture segmentation which is concerned with splitting an image into texturally homogeneous regions. Neither the normal regions nor the defective regions have to be texturally uniform. Thus, texture segmentation, image classification, and visual inspection are related but distinct.

A variety of statistical techniques have been investigated to perform defect detection tasks on a range of different texture surfaces, e.g. graylevel co-occurrence matrices [168], local binary patterns (LBPs) with self organising maps [99], and first and higher order statistics of texels [196, 62]. Signal processing techniques have also been widely explored in application to defect detection,

[†]While we propose methods in this thesis that do localise defects, we do not categorise the defects. This would require further work and analysis of the defective regions which is outside the scope of this work.

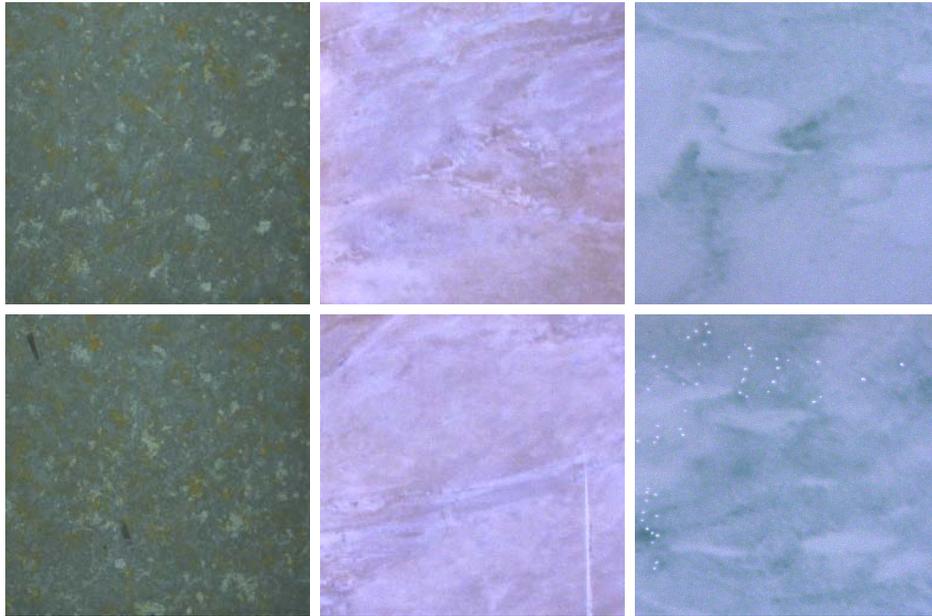


Figure 1.3: Examples of textural defects on ceramic tile surfaces (different families of tiles) - first row: normal samples; second row: defective samples with ink-drops, a print error, and a cloud of pin holes (from left to right).

such as Fourier-domain analysis [21], wavelet decomposition [143], and various filtering based approaches including eigenfilters [2, 114] and Gabor filters [49, 83]. For those materials that exhibit a high degree of regularity and periodicity, such as textiles and wafer, template-based methods have also proved useful for defect detection [186]. A detailed review is given in Chapter 2.

The second part of the thesis focuses on local contextual models. Although the images can display very complex and random patterns, there are visual primitives that govern the consistency within each texture family. In fact, there are several works in the literature that explicitly learn the primitives, e.g. texton [72, 198] and fractal models [103], or model the local neighbourhood relationship, such as Markov Random Field (MRF) [90]. In this work, a semi-parametric model is proposed to implicitly learn the visual primitives or textural elements. The model assumes each image is generated from a superposition of various sizes of image patches taken from a small set of representatives, i.e. a linear combination with added variations at each pixel position. Each representative patch and its corresponding variance matrix is referred to as a *texture exemplar*, or simply *texem*. The texems can be viewed as implicit representations of visual primitives, since image patches with different sizes may encapsulate multiple fundamental micro-structures or only a portion of it. Theoretically, there is no limit to the size of a texem. It can be extremely small, e.g. a single pixel, or as large as the image itself. This flexibility allows the model to avoid dedicated efforts in searching for the appropriate patch size which is often difficult and image dependent.

In this thesis, the derivation of the texems in graylevel images, and then colour images using different inference procedures, is discussed. In application to defect detection on ceramic tile surfaces, the texem model is fit into a novelty detection scheme as (a) defects are usually unpredictable and (b) usually only knowledge of good samples is available. The method is tested on ceramic tile images captured at different manufacturing sites for the MONOTONE project, and is also evaluated on other colour texture datasets. This method should also be applicable to other surfaces, e.g. wood and textile. Indeed, its application to defect detection in other applications, such as medical imaging, is demonstrated. To determine if the texem model can be suitable for texture segmentation, it is also extended to an unsupervised image segmentation application.

1.3 Contributions

The main contributions of this thesis are summarised as follows:

1. *Colour Tonality Inspection.* A multidimensional histogram method to inspect colour tonality on colour textured surfaces is proposed. Comparison in the noise dominated chromatic channels is error prone. Vector-ordered colour smoothing is therefore performed and then a PCA-based reconstruction of a query tile based on an automatically developed reference tile eigenspace is generated. Histograms of local feature vectors are then compared for tonality defect detection. The results were published in [191, 192].
2. *The Texem Model and its Application to Defect Detection.* A novel two layer generative model, named the *texem* model, is developed to represent graylevel images and used for defect detection based on normal samples in a novelty detection framework. This work was published in [190].
3. *Texem Analysis in Image Eigenchannels* The graylevel texem model is extended to deal with colour images by transforming the images into uncorrelated channels, in which the texems are then computed. This work was published in [188].
4. *Colour Texems.* The texem model is extended to deal with colour images using a new formulation so that there is no need for vectorisation of image patches and transformation of colour images into separate channels. The spatial and inter-spectrum interactions are taken into account simultaneously. The texem model thus can process either scalar-valued or vector-valued data. A paper [189] has been submitted and is under review at the time of writing.

Additionally, this thesis includes the following minor contributions: (a) Efficient and effective luminance correction methods for both radial and line scan cameras are developed. (b) Two alternative applications of the colour texem model are investigated, for which promising results are reported. The first is to detect abnormalities in tympanic membrane images, published in [187], which involves snake segmentation and colour texem analysis. The second is to segment colour ink dots printed on tile surfaces in which the colour texem model is extended from novelty detection to unsupervised image segmentation. The segmentation accuracy is measured against manual labelling.

1.4 Thesis Layout

The rest of the thesis is organised as follows.

Chapter 2 - Literature Review: provides an overview of visual inspection techniques and describes the texture analysis methods that are related to the proposed approaches.

Chapter 3 - Image Datasets and Pre-processing: describes the datasets used in this thesis and their pre-processing to correct luminance inhomogeneity.

Chapter 4 - Colour Tonality Inspection: presents a novel method in detecting colour tonality defects on ceramic tile surfaces that can discriminate subtle colour shade differences. A comparative study is also presented.

Chapter 5 - Defect Detection using Local Contextual Analysis: A novel graylevel texem model focusing on local contextual analysis is introduced in application to detecting defects in random textures using a novelty detection framework. Its extension to deal with colour images by processing in image eigenchannels is also presented. This factorisation scheme is compared with RGB channel separation on texture collage images.

Chapter 6 - Colour Texems: describes an extension of the graylevel texem model to colour texems. Its application to defect detection and evaluation on other colour texture datasets is presented. The method is compared with a popular filtering based method and a statistical approach. Also, an alternative application of the colour texems to medical image analysis involving the detection of abnormalities in colour tympanic membrane images is presented. Furthermore, the texem model is extended to perform unsupervised image segmentation in colour ink-dot images captured on tile surfaces.

Chapter 7 - Conclusions and Future Work: concludes the thesis with discussions of the proposed methods and possible extensions, in particular, image segmentation using colour texems.

1.5 Notation

The following notations are used in this thesis:

image	\mathbf{I}
image patch	\mathbf{Z}
vector, matrix	bold letter, e.g. feature vector \mathbf{f}
magnitude of vector \mathbf{f}	$ \mathbf{f} $
diagonal element of matrix \mathbf{A}	$\text{diag}(\mathbf{A})$
texem	\mathbf{m}
mean of texem	μ
variance of texem	ω
a set of texems	\mathcal{M}
graylevel texem parameter set	θ
colour texem parameter set	Θ
estimated parameter	letter with a hat, e.g. $\hat{\theta}$
segmentation class label	c
expectation	$E[.]$
pixel coordinate	(x, y)
displacement vector	$\mathbf{d} = (dx, dy)$
image index i	\mathbf{I}_i
patch index i	\mathbf{Z}_i
texem index k	\mathbf{m}_k
eigenvector index i	\mathbf{e}_i
image eigenchannel index i	$\mathbf{I}_{\mathbf{e}_i}$
eigenvector matrix	\mathbf{E}
pixel index j in image patch	$\mathbf{Z}_{j,i}$: j th pixel position in the i th image patch
pixel index j in texem mean	$\mu_{j,k}$: j th pixel position in the k th texem mean
element index j in texem variance	$\omega_{j,k}$: j th element in the k th texem variance
total image pyramid level	l
image pyramid index i	$\mathbf{I}^{(i)}$
ordered colour vector index i	$\mathbf{f}^{(i)}$
probability function of random variable x	$p(x)$
Gaussian convolution	G_σ with standard deviation σ
down-sampling operator	S^\downarrow
forward PCA transform	\overrightarrow{PCA}
backward PCA transform	\overleftarrow{PCA}
normalised cross correlation	NCC

Chapter 2

BACKGROUND

2.1 Introduction

Visual inspection of randomly textured colour surfaces inevitably involves colour texture analysis and pattern classification. The former is mainly concerned with feature representation and extraction, as well as data perception and modelling. The latter consists of pattern representation, cluster analysis, and discriminant analysis. The aim of this chapter is to review appropriate texture analysis techniques for the purposes of visual inspection and to consider decision making schemes that are able to discriminate the features extracted from normal and defective regions. Whenever appropriate, the relationship between the reviewed methods and proposed approaches will be discussed.

As mentioned in Chapter 1, the inspection problem dealt with in this thesis can be divided into two parts: colour textural defect detection and colour tonality inspection. Detecting textural defects has attracted much attention in the past two decades. However, inspecting random textured surfaces, such as printed ceramic tiles, remains a challenging task. We give an overview of the state-of-the-art in textural defect detection, including its achievements and challenges. Colour tonality is also a very important quality factor in ceramic tile production. The relevant works in this area are also discussed in this chapter.

A significant differentiating factor in visual inspection approaches is that of supervised classification versus novelty detection. For applications where both normal and defective samples can be easily obtained and pre-defined, supervised classification based approaches are usually favoured. However, when defects are unpredictable and defective samples are unavailable, novelty detection is more desirable. Both of these approaches will be reviewed and compared in the context of defect detection.

The rest of the chapter is organised as follows. As the techniques reviewed in textural defect detection generally include those that have been used in colour tonality inspection, textural defect detection and related texture analysis techniques are first discussed in Section 2.2. Colour tonality inspection is then reviewed later in Section 2.3. The ability for a texture analysis technique to be extendible to deal with colour images is particularly important in this thesis. Thus, colour texture analysis is separately discussed in Section 2.4. Section 2.5 compares classification oriented approaches with novelty detection based approaches. Finally, Section 2.6 summaries this chapter.

2.2 Textural Defect Detection

Texture is one of the most important characteristics in identifying defects or flaws. It provides important information for recognition and interpolation. In fact, the task of detecting defects has been largely viewed as a texture analysis problem. Features with large inter-class variations and small intra-class variations are sought to better separate differing textures. Much effort has been put into extracting useful texture features. As it is not practical to provide an exhaustive survey of all texture features in this limited space, this section concentrates on those techniques that have been widely used in texture analysis or demonstrate good potential for application to automatic inspection.

With reference to several survey papers [58, 181, 174, 178, 138, 170], we categorise texture analysis techniques used for visual inspection four ways: statistical approaches, structural approaches, signal processing approaches, and model based approaches. As already noted, colour texture analysis is separately discussed later. Table 2.1 shows a summary list of some of the key texture analysis methods that have been applied to defect detection. Clearly, statistical and signal processing approaches have been very popular.

2.2.1 Statistical approaches

Statistical texture analysis methods measure the spatial distribution of pixel values. They are well rooted in the computer vision world and have been extensively applied to various tasks. A large number of statistical texture features have been proposed, ranging from first order statistics to higher order statistics. Amongst many, histogram statistics, co-occurrence matrices, autocorrelation, and local binary patterns have been applied to visual inspection.

Table 2.1: Inexhaustive list of textural defect detection methods

Approach	Method	References
Statistical	1. Histogram properties	[168, 75, 119]
	2. Co-occurrence matrix	[150, 64, 149, 123, 63, 85]
	3. Local binary pattern	[63, 119, 118, 99, 100]
	4. Other graylevel statistics	[29, 145, 166, 81, 31]
	5. Autocorrelation	[185, 62]
	6. Morphological operations	[79, 155, 102]
	7. Edge Detection	[182]
	8. Registration-based	[94, 183, 40, 186]
Structural	1. Primitive measurement	[79, 155]
	2. Skeleton representation	[25]
Signal processing	1. Spatial domain filtering	[2, 173, 117, 197, 123, 84, 54, 114]
	2. Frequency domain analysis	[137, 193, 26, 48, 162, 21, 164]
	3. Joint spatial/spatial-frequency	[20, 68, 49, 77, 143, 184, 85, 82, 167, 163, 12, 83, 113, 104, 146, 195, 165, 93]
Model-based	1. Fractal models	[36, 37]
	2. Random field model	[34, 124, 123, 7, 130]
Colour texture analysis for defect detection		[79, 155, 166, 100, 54, 165]

A Histogram properties

Commonly used histogram statistics include range, mean, geometric mean, harmonic mean, standard deviation, variance, and median. Table 2.2 lists some similarity measurements of two distributions [15, 142, 89], where x_i and y_i are the number of events in bin i for the first and second datasets, respectively, \bar{x} and \bar{y} are the mean values, n is the total number of bins, and $x_{(i)}$ and $y_{(i)}$ denote the sorted (ascending order) indices. Note EMD is the Earth Mover's Distance.

Despite their simplicity, histogram techniques have proved their worth as a low cost, low level approach in various applications, such as [158, 15, 134]. They are invariant to translation and rotation, and insensitive to the exact spatial distribution of the colour pixels. These characteristics make them ideal for use in application to colour shade discrimination, e.g. [15]. The accuracy of histogram based methods can be enhanced by using statistics from local image regions [18]. Simple histogram moments, such as mean and standard deviation, from subblocks were used for defect classification [168]. In chapter 4, histogram statistics from eigenspace features are used to discriminate the colour shade difference amongst tiles.

Table 2.2: Some histogram similarity measurements.

Measurement	Formula
L_1 norm	$L_1 = \sum_{i=1}^n x_i - y_i $
L_2 norm	$L_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
Mallows or EMD distance	$M_p = \left(\frac{1}{n} \sum_{i=1}^n x_{(i)} - y_{(i)} ^p\right)^{1/p}$
Bhattacharyya distance	$B = -\ln \sum_{i=1}^n \sqrt{x_i y_i}$
Matusita distance	$M = \sqrt{\sum_{i=1}^n (\sqrt{x_i} - \sqrt{y_i})^2}$
Divergence	$D = \sum_{i=1}^n \left((x_i - y_i) \ln \frac{x_i}{y_i} \right)$
Histogram intersection	$H = \frac{\sum_{i=1}^n \min(x_i, y_i)}{\sum_{i=1}^n x_i}$
Chi-square	$\chi^2 = \sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i + y_i}$
Normalised correlation coefficient	$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$

B Co-occurrence matrices

Spatial graylevel co-occurrence matrices (GLCM) [59] are one of the most well-known and widely used texture features. These second order statistics are accumulated into a set of 2D matrices, $\mathbf{P}(r, s|\mathbf{d})$, each of which measures the spatial dependency of two graylevels, r and s , given a displacement vector $\mathbf{d} = (dx, dy)$. The number of occurrences (frequencies) of r and s , separated by distance \mathbf{d} , contributes the (r, s) th entry in the co-occurrence matrix $\mathbf{P}(r, s|\mathbf{d})$. Let \mathbf{I} denote a $w \times h$ image. A co-occurrence matrix is given as:

$$\mathbf{P}(r, s|\mathbf{d}) = ||\{(x_1, y_1), (x_2, y_2) : \mathbf{I}(x_1, y_1) = r, \mathbf{I}(x_2, y_2) = s\}|| \quad (2.1)$$

where $(x_1, y_1), (x_2, y_2) \in w \times h$, $(x_2, y_2) = (x_1 \pm dx, y_1 \pm dy)$ and $||\cdot||$ is the cardinality of a set. Texture features, such as energy, entropy, contrast, homogeneity, and correlation, are then derived from the co-occurrence matrix. Several works have reported using co-occurrence matrices to detect defects, such as [150, 168, 64]. For example in [64], Iivarinen *et al.* applied co-occurrence texture features to detecting defects in paper web where the normal textures have characteristic frequency.

Co-occurrence matrix features can suffer from a number of shortcomings. It appears there is no generally accepted solution for optimising \mathbf{d} [170, 113]. The number of graylevels is usually reduced in order to keep the size of the co-occurrence matrix manageable. It is also important to ensure the number of entries of each matrix is adequate to be statistically reliable. For a given displacement vector, a large number of features can be computed, which implies dedicated feature selection procedures. In a comparative study by Özdemir *et al.* in [123], the co-occurrence matrix method showed poor performance in detecting textural defects in textile products compared to other techniques such as MRF and filtering-based methods. Iivarinen [63] found co-occurrence

matrices and the local binary pattern (LBP) operator had similar performance in detecting defects, while LBP was more efficient.

C Autocorrelation

The autocorrelation feature is derived based on the observation that some textures are repetitive in nature, such as textiles. It measures the correlation between the image itself and the image translated with a displacement vector, $\mathbf{d} = (dx, dy)$. Formally, autocorrelation is given as:

$$\rho(\mathbf{d}) = \frac{\sum_{x=0}^w \sum_{y=0}^h \mathbf{I}(x, y) \mathbf{I}(x + dx, y + dy)}{\sum_{x=0}^w \sum_{y=0}^h \mathbf{I}^2(x, y)}. \quad (2.2)$$

Textures with strong regularity will exhibit peaks and valleys in the autocorrelation measure. This second order statistic is clearly sensitive to noise interference. Higher order statistics, e.g. [39, 62], have been investigated, for example, Huang and Chan [62] used fourth-order cumulants to extract harmonic peaks and demonstrated its ability to localise defects in textile images and Wood [185] used autocorrelation of subimages to detect textile defects. Nevertheless, the autocorrelation function is generally considered as unsuitable for random textures with irregularly arranged textural elements.

D Local binary patterns

The LBP operator was first introduced by Ojala *et al.* [121] as a shift invariant complementary measure for local image contrast. It uses the graylevel of the centre pixel of a sliding window as a threshold for surrounding neighbourhood pixels. Its value is given as a weighted sum of thresholded neighbouring pixels.

$$\mathcal{L}_{P,R} = \sum_{p=0}^{P-1} \text{sign}(g_p - g_c) 2^p, \quad (2.3)$$

where g_c and g_p are the graylevels of centre pixel and neighbourhood pixels respectively, P is the total number of neighbourhood pixels, R denotes the radius, and $\text{sign}(\cdot)$ is a sign function such that

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2.4)$$

Figure 2.1 shows an eight-neighbours LBP calculation. A simple local contrast measurement, $\mathcal{C}_{P,R} = \sum_{p=0}^{P-1} \text{sign}(g_p - g_c) g_p$, is calculated as a complement to the LBP value in order to char-

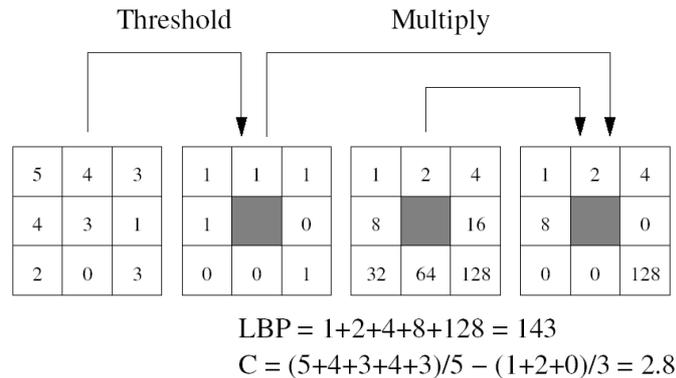


Figure 2.1: Calculating LBP code and a contrast measure (*image adapted from [98]*).

acterise local spatial relationships, together called LBP/C [121]. Two-dimensional distributions of the LBP and local contrast measures are used as texture features.

The LBP operator is relatively invariant with respect to changes in illumination and image rotation (for example, compared to co-occurrence matrices), and computationally simple [98]. Ojala *et al.* demonstrated good performance of LBP in texture classification. It has been applied to defect detection on ceramic tile surfaces [113], wood [119, 118], and real-time inspection [99]. In Chapter 6, the proposed texem model is compared against LBPs in detecting novelties in VisTex image collages.

2.2.2 Structural approaches

In structural approaches, texture is characterised by *texture primitives* or texture elements, and the spatial arrangement of these primitives [178]. Thus, the primary goals of structural approaches are firstly to extract texture primitives, and secondly to model or generalise the spatial placement rules. The texture primitive can be as simple as individual pixels, a region with uniform graylevels, or line segments. The placement rules can be obtained through modelling geometric relationships between primitives or learning statistical properties from texture primitives.

Zucker [200] proposed that natural textures can be treated as ideal patterns that have undergone certain transformations. The placement rule is defined by a graph that is isomorphic to a regular or semi-regular tessellation which is transformable to generate variate natural textures. In [38], Connors and Harlow described a texture primitive as a parallelogram represented by two displacement vectors whose directions correspond to the dominant orientations in the texture. Fu [50] considered a texture as a string of a language defined by a tree grammar which defines the spatial placement rules and its terminal symbols are the texture primitives that can be individual pixels,

connected or isolated. Marr [109] proposed a symbolic description, primal sketch, to represent spatial texture features, such as edges, blobs, and bars. Voorhees and Poggio [179] used the Laplacian of Gaussian filter to extract the blobs as texture primitives. Later, Lindeberg [92] extended the primal sketch in multiscale to detect salient blob-like structures. In [178], Vilnrotter *et al.* used edge features as texture elements which were grouped by type. The spatial relationship between elements are computed as grid placement rules. Tuceryan and Jain [169] proposed the extraction of texture tokens by using properties of the Voronoi tessellation of the given image. The local spatial distributions of tokens, such as peaks, line segments, or closed boundaries, are reflected in the shapes of the Voronoi polygons where the interaction between them is geometrically modelled. In [72], Julesz introduced the concept of textons as fundamental image structures, such as elongated blobs, bars, crosses, and terminators (more details later in this section). The textons were considered as atoms of pre-attentive human visual perception. The idea of describing texture using local image patches and placement rules has also been practised in texture synthesis, e.g. [47].

In [25], Chen and Jain proposed a structural approach to identify defects in textile images. The image was first thresholded using histogram analysis and then was mapped into a data structure which represents the skeleton structure of the texture. Statistical measurements were taken from both location and length histograms of the skeleton. These measurements were compared with a pre-defined acceptance range which was learnt from defect-free samples to detect defects. Kittler *et al.* [79, 155] used K-means clustering to split randomly textured tile images into binary stacks, in which blob analysis was performed to measure the primitives. The measurements included size, perimeter fractality, elongatedness, and spatial distribution.

A *Texton*

We now briefly describe the texton model which has received enormous attention in vision understanding and applications. Its concept is related, yet significantly different, to the texem model introduced in Chapter 5.

Textons were first formally presented by Julesz in [72] as fundamental image structures and were considered as atoms of pre-attentive human visual perception. However, there is a lack of rigorous mathematical definition for the texton and the concept remains vague, although there have been a few recent attempts, notably by [87, 198].

In [87], Leung and Malik adopted a *discriminative* model to describe textons. Each texture image was analysed using a filter bank consisting a large number of Gaussian filters (48) with different orientations, scales and phases. Thus, a high dimensional feature vector was extracted at each pixel position. K-means was used to cluster those filter response vectors into a few central vectors which were referred to as textons.

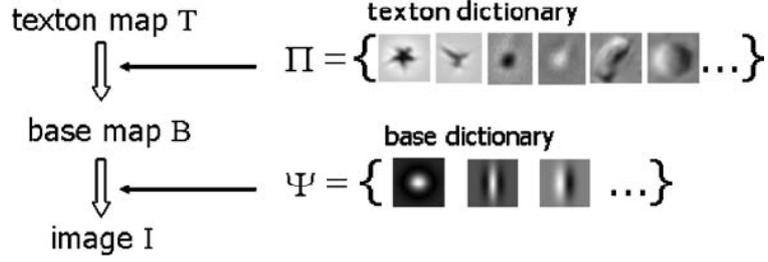


Figure 2.2: A three-level generative model: an image I is a linear addition of some image bases selected from a base dictionary Ψ , including Gabor and Laplacian of Gaussians. The base map is further generated by a smaller number of textons selected from a texton dictionary Π (*image adapted from [198]*).

More recently, Zhu *et al.* [198] argued that textons could be defined in the context of a *generative* model of images. In their three-level generative model, an image I was considered as a superposition of a number of image base functions that were selected from an over-complete dictionary Ψ . These image bases, such as Gabor and Laplacian of Gaussian functions at various scales, orientations, and locations, were generated by a smaller number of texton elements which were in turn selected from a texton dictionary Π . This three-level structure is illustrated in Figure 2.2. An image I is generated by a base map \mathbf{B} which is in turn generated from a texton map \mathbf{T} , i.e:

$$\mathbf{T} \xrightarrow{\Pi} \mathbf{B} \xrightarrow{\Psi} \mathbf{I}, \quad (2.5)$$

where $\Pi = \{\pi_i, i = 1, 2, \dots\}$ and $\Psi = \{\psi_i, i = 1, 2, \dots\}$.

Each texton, an instance in the texton map \mathbf{T} , is considered as a combination of a certain number of base functions with deformable geometric configurations, e.g. star, bird, snowflake as shown in Figure 2.2. This configuration can be illustrated using a texton of a star shape as shown in Figure 2.3. Two different base functions are selected from a base dictionary. One is a radial shape, which can be a Laplacian of Gaussian function, and the other is a bar structure which can be a symmetric Gabor function. These bases are translated, rotated and superpositioned on each other to produce a star texton. By fitting this generative model to observed images, the texton dictionary then is learnt as parameters of the generative model.

Textons, in one form or another, have been used in several applications. In [87], which was referred to earlier, Leung and Malik defined $2D$ and $3D$ textons as cluster centres in filter response space from a single image and a stack of images with representative viewpoints and lighting. These texton representations were then used to classify material images (in grayscale) from the Columbia-Utrecht (CURET) texture database [41]. Schmid [147] employed symmetric base functions to obtain rotation invariant representations for content based image retrieval. Varma and Zisserman [175, 177] performed a similar recognition process, but with different base functions,

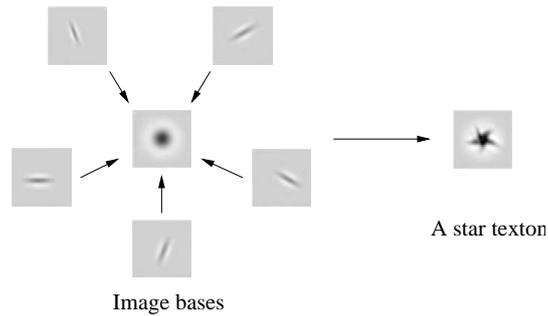


Figure 2.3: A star texton configuration (*image adapted from [198]*).

resulting in smaller filter response space dimension and rotation invariance.

The design and selection of the base functions are fundamentally important in texton based approaches. Also, the use of base functions limits the possibility to deal with colour images. The proposed texem method is related to the texton representation, however, implicit representations without base functions are used instead. More details and comparisons will be given in chapters 5 and 6.

2.2.3 Signal processing approaches

This section will briefly review various signal processing techniques for defect detection. As a common characteristic, most signal processing methods apply filter banks on the image and compute the energy of the filter responses. The methods can be divided into spatial domain, frequency domain, and joint spatial/spatial-frequency domain techniques.

A Spatial domain and frequency domain filtering

Measuring edge strength is one of the earliest attempts to discriminate different textures. In the spatial domain, the images are usually filtered by gradient filters to extract edges, lines, isolated dots, etc. Sobel, Robert, Laplacian, Laws filters have been routinely used as a precursor to measuring edge density. In [101], Malik and Perona designed a bank of differences of offset Gaussian function filters to model pre-attentive texture perception in human vision. Ade [1] proposed eigenfilters, a set of masks obtained from the Karhunen-Lóeve (KL) transform [71] of local image patches, for texture representation.

In [84], Kumar and Pang used linear finite impulse response (FIR) filters to detect defects in textiles. Filter responses from both defect free and defective regions were collected. Then, optimal filters were selected based on discriminant analysis of the filters using objective functions, such

as Mahalanobis-Singh and Fisher criterion. Neubauer [117] exploited three 5×5 FIR filters and performed classification using histograms of features calculated from 10×10 pixel regions. Zhou *et al.* used simple linear filters to capture line-like defects in IC packages. Unser and Ade [173] and recently Monadjemi *et al.* [114] employed eigenfilters in defect detection. The authors argued that unlike other spatial operators, eigenfilters are image dependent and the detailed images are orthogonal to each other.

Many other techniques apply filtering in the frequency domain, particularly when no straightforward kernel can be found in the spatial domain. The image is transformed into the Fourier domain, multiplied with the filter function and then re-transformed into the spatial domain saving on the spatial convolution operation. Ring and wedge filters are some of the most commonly used frequency domain filters. In [33], Coggins and Jain used seven dyadically spaced ring filters and four wedge-shaped orientation filters, which have Gaussian cross sections, for feature extraction. D'Astous and Jernigan [42] used peak features, such as strength and area, and power distribution features, such as power spectrum eigenvalues and circularity, to discriminate textures.

In [162], the authors used the Fourier transform to reconstruct textile images for defect detection. The line patterns in a textile image, supposed to be defects, were taken out by removing high energy frequency components in the Fourier domain using a one-dimensional Hough transform. The differences between the restored image and the original image were considered as potential defects. A similar idea was explored in [164], but low pass filtering was used to remove the periodic information. Chan and Pang [21] extracted harmonic peaks from horizontal and vertical power spectrum slices, based on the observation that defects usually occur in horizontal and vertical directions. However, these methods all rely on the assumption that faultless fabric is a repetitive and regular texture. These methods will not be suitable for defect detection in random textures.

B Joint spatial/spatial-frequency methods

One difficulty with the Fourier transform is that it has relatively poor spatial resolution, as Fourier coefficients depend on the entire image. Many applications require the analysis to be localised in the spatial domain, e.g. in this case defect detection and localisation.

The classical way of introducing spatial dependency into Fourier analysis is through the windowed Fourier transform. If the window function is Gaussian, the windowed Fourier transform becomes the well-known Gabor transform, which can arguably achieve optimal localisation in the spatial and frequency domains [43]. Gabor filters can be categorised into two components: a real part as the symmetric component and an imaginary part as the asymmetric component. The 2D Gabor

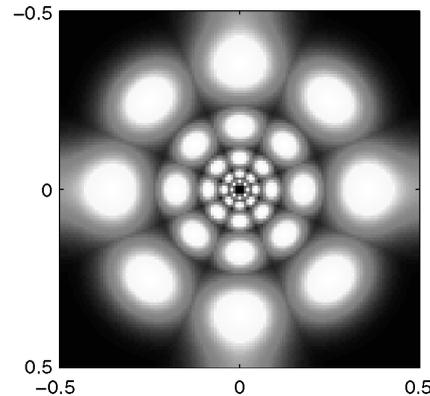


Figure 2.4: The frequency response of the dyadic bank of Gabor filters. The maximum amplitude response over all filters is plotted. Each filter is represented by one centred-symmetric pair of lobes in the illustration. The axes are in normalised spatial frequencies (image adapted from [136]).

function can be mathematically formulated as:

$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \exp(2\pi j u_0 x), \quad (2.6)$$

where σ_x and σ_y define the Gaussian envelope along the x and y directions respectively, u_0 denotes the radial frequency of the Gabor function, and $j = \sqrt{-1}$. Psychophysiological studies, such as [44], have suggested that the human brain performs multi-channel, frequency and orientation analysis on the visual image. These findings have strongly motivated the use of Gabor analysis, along with other multiscale techniques. Turner [171] and Bovik et al. [32] first proposed the use of Gabor filters in texture analysis. Jain and Farrokhnia [65] used it in segmentation and classification of textures with dyadic coverage of the radial spatial frequency range. Figure 2.4 shows the frequency response of the dyadic Gabor filter bank with the centre frequencies $\{2^{-\frac{11}{2}}, 2^{-\frac{9}{2}}, 2^{-\frac{7}{2}}, 2^{-\frac{5}{2}}, 2^{-\frac{3}{2}}\}$, and orientations $\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ [136].

The Gabor filter bank has been extensively studied in visual inspection, e.g. [49, 184, 82, 167, 12, 83, 113, 165]. Kumar and Pang [82] performed fabric defect detection using only real Gabor functions. Later in [83], the same authors used a class of self-similar Gabor functions to classify fabric defects. They also investigated defect detection using only imaginary Gabor functions as an edge detector. For computational efficiency, the fabric samples were analysed using horizontally and vertically projected one-dimensional profiles. In [12], Bodnarova *et al.* applied a Fisher cost function to select a subset of Gabor functions based on the mean and standard deviation of the template (defect-free) feature images to perform textile flaw detection. The filtering responses of those selected Gabor functions were supposed to have compact distributions. Defects were localised by thresholding the filtering responses from an unseen image sample based on the mean

and standard deviation of template filtering responses. Tsai and Wu [167] also performed Gabor filter selection so that the filter response energy of the normal texture, assumed to be homogeneous, was close to zero. Wiltschi *et al.* [184] performed automatic scale selection to preserve channels with maximum energy and directional information. In [49], Escofet *et al.* performed multiscale Gabor filtering in a novelty detection framework. Defect candidates across different scales and orientations were fused together using logical processes. This defect fusion approach was adopted in the proposed novelty detection methods. The proposed texem method is compared against the Gabor filter bank method in Chapter 6.

Carrying similar properties to the Gabor transform, wavelet transform representations have also been widely used for texture analysis, e.g. [143, 85, 146, 195]. Wavelet analysis uses approximating functions that are localised in both spatial and spatial-frequency domain. The input signal is considered as the weighted sum of overlapping wavelet functions, scaled and shifted. These functions are generated from a basic wavelet (or mother wavelet) by dilation and translation. Dyadic transform is one of the most commonly used, however, its frequency and orientation selection are rather coarse. Wavelet packet decomposition [35], as a generalisation of the discrete wavelet transform, is one of the extensions to improve the selectivity where at each stage of the transform, the signal is split into low-pass and high-pass orthogonal components. The low-pass is an approximation of the input signal, while the high-pass contains the missing signals from the approximation. Finer frequency selectivity can be further obtained by dropping the constraints of orthogonal decomposition.

In application to defect detection, Sari-Sarraf and Goddard [143] performed discrete wavelet transforms on fabric images. The detailed images were fused together to produce a feature map in which the normal texture regions, assumed to be homogeneous, had small values. The defects were segmented by thresholds learnt from training templates. The key process was to attenuate the normal regions, and accentuate the defective regions, based on the assumption that normal texture was regular and homogeneous, and defects were those that broke the local homogeneity. Scharcanski [146] also used the discrete wavelet transform to classify stochastic textile textures. Latif-Amet *et al.* [85] extracted co-occurrence and MRF-based features from wavelet transform coefficients for fabric defect detection. Graylevel difference-based features from subbands of the wavelet transform were also applied in classifying fabric defects. Recently, Yang *et al.* [195] used adaptive wavelets resulting in fewer scales compared with the standard wavelet transform. The wavelet functions were adaptively selected based on an objective function measuring the ratio of average energies between defective regions and defect-free regions. The method achieved better performance than the standard wavelet transform, but needed supervised training.

2.2.4 Model based approaches

Model based methods include, among many others, fractal models [103], random field models [90], and the epitome model [70]. In this section, we briefly review these three as they are relevant to the proposed approaches.

A *Fractal models*

Fractals, initially proposed by Mandelbrot [103], are geometric primitives that are self-similar and irregular in nature. Fragments of a fractal object are exact or statistical copies of the whole object and they can match the whole by stretching and shifting.

Fractal dimension is one of the most important features in the fractal model as a measure of complexity or irregularity. Several methods have been developed to estimate the fractal dimension. Pentland [129] used the Fourier power spectral density to estimate the fractal dimension for image segmentation. The image intensity is modelled as 3D fractal Brownian motion surfaces. Gangepain and Roques-Carnes [51] proposed the box-counting method which was later improved by Voss [180] and Keller *et al.* [76]. Super and Bovik [157] have proposed to use Gabor filters to estimate the fractal dimension in textured images. Sarkar and Chaudhuri [144] introduced the differential box-counting (DBC) method, which has also been used in texture analysis (for example [23]). The authors argued that DBC method outperformed in estimating fractal dimension in terms of efficiency, accuracy, and dynamic range compared with those of [129, 51, 76]. Later, Jin *et al.* [69] suggested an improvement in the scale limit of the DBC method. Lacunarity is another important measurement in fractal models. It measures the structural variation or inhomogeneity and can be calculated using the gliding-box algorithm [5].

In [36], Conci and Proenca used box counting to extract fractal features for detecting defects in fabric images. In a comparative study by Ohanian and Dubes [120], the fractal method performed reasonably well against co-occurrence matrices, Gabor filters, and Markov random fields in texture classification. However, it has achieved limited success in real applications. Fractals can have the same fractal dimension but look completely different. Fractal models are mainly suitable for natural textures where self-similarity may hold.

B *Random field models*

MRF theory provides a convenient and consistent way for modelling context dependent entities such as pixels, through characterising mutual influences among such entities using conditional

MRF distributions [90]. The establishment of the equivalence between MRFs and Gibbs distribution [57, 9] provided tractable means for statistical analysis as Gibbs distribution takes a much simpler form. Since then, MRFs have been applied to various applications, such as image segmentation, texture synthesis, texture classification, and image restoration.

In MRF models, an image is represented by a finite rectangular lattice within which each pixel is considered as a site. Neighbouring sites then form cliques and their relationships are modelled in the neighbourhood system. The equivalence theorem between MRF and Gibbs random fields (GRF) provides a general basis for the specification of MRF joint distribution functions. Different distributions can be obtained by specifying the potential functions, such as Gaussian MRF (GMRF) [24] and the FRAME model [199].

Cohen *et al.* [34] partitioned test images into non-overlapping sub-blocks. Each image block was inspected based on statistics derived from defect free samples using GMRFs. Baykut *et al.* [7] implemented this method in a real-time application with a dedicated DSP system. In [123], the authors showed that MRF based methods were competitive in a comparative study against other statistical and spectral based methods in defect detection.

The proposed texem model is related to the MRF model in the sense that both rely on local contextual analysis. However, texems do not explicitly model the *interaction* of local pixels, but model the global distribution constrained in a local context. Additionally, the texem model is a semi-parametric model, while most MRF models are usually fully parametric.

C Epitome model

Recently, Jojic *et al.* [70] introduced a novel model to describe image(s), called the epitome model. The epitome is a small, condensed representation of a given image which contains its primitive shapes and textural elements. The mapping from the epitome to its original pixels is hidden, and several images may share the same epitome by varying the hidden mapping. In this model, raw pixel values are used to characterise textural and colour properties, instead of popular filtering responses. The epitome is derived using a generative model. It is assumed that image patches from the original (large) images are produced from the epitome by copying pixel values from it with added Gaussian noise. Thus, as a learning process various sizes of patches from the image are taken and are forced into the epitome, a much smaller image, by examining the best possible match. The epitome is then updated accordingly when new image patches are sampled. This process iteratively continues until the epitome is stabilised. Figure 2.5 shows an example image and two epitomes at different sizes. We can see that the epitomes are relatively compact representations of the image.

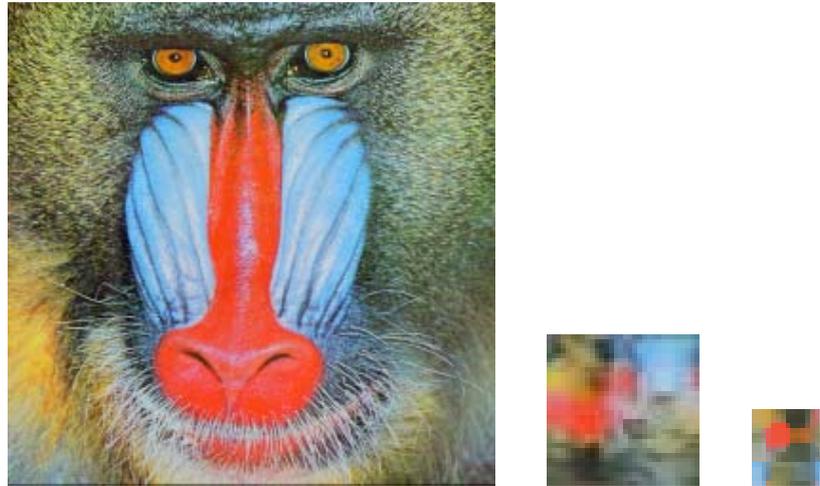


Figure 2.5: Epitome - from left: Original colour image, its 32×32 epitome, and its 16×16 epitome (generated with the software provided by the authors in [70]).

The authors of the epitome model demonstrated its abilities in texture segmentation, image denoising, and image inpainting [70]. Stauffer [156] also used epitomes to measure the similarity between pixels and patches to perform image segmentation. In this thesis, the epitomic appearance model was investigated for detecting textural defects. However, the hidden mapping of the epitome model was found to be impractical for defect detection. More details will be discussed in Chapter 5. Additionally, the spatial relationships are preserved in a limited scale, e.g. see Figure 2.5 where the red and blue region in the original image are connected to each other, while in the epitomic versions (especially in the 16×16 epitome) they are separated from each other.

2.2.5 Discussion

A brief classification of the texture analysis techniques used for defect detection is shown in Table 2.1. As mentioned earlier, the statistical and signal processing based methods have been in favour in terms of the amount of research reported. It is also worth noting that the categorisation of the texture analysis techniques used for defect detection as described above and listed in Table 2.1 is not a crisp classification. There are techniques that combine methods from different categories for texture analysis, e.g. [85] applies co-occurrence measurement on wavelet transformed detail images.

A *Texture analysis*

There are several comparative studies in the literature that evaluate texture analysis methods. It must be noted that different studies use different datasets and possibly different parameter set-

tings. In [120], Ohanian and Dubes compared the fractal model, co-occurrence matrices, the MRF model, and Gabor filtering for texture classification. The co-occurrence features generally outperformed other features in terms of classification rate. However, as pointed out in [170], they used raw Gabor filtered images instead of using empirical nonlinear transformations to obtain texture features. Reed and Wechsler [139] performed a comparative study on various spatial/spatial-frequency representations and concluded that the Wigner distribution had the best joint resolution. However, Pichler *et al.* [133] reported superior results using Gabor filtering over other wavelet transforms. In [22], Chang *et al.* evaluated co-occurrence matrices, Laws texture energy, and Gabor filters for segmentation in natural and synthetic images. Gabor filtering again achieved best performance. Later, Randen and Husøy performed an extensive evaluation of various filtering approaches for texture segmentation in [136]. The methods included Laws filters, ring and wedge filters, various Gabor filters, and wavelet transforms. No single approach was found to be consistently superior to the others on their twelve texture collages. Recently, Singh and Singh [153] compared seven spatial texture analysis techniques, including autocorrelation, co-occurrence matrices, Laws filters, run lengths, and statistical geometrical (SG) features [27]. The SG features performed best in classifying VisTex and MeasTex [154] textures. In the SG method, the image was segmented into several binary stacks depending on the number of graylevels in the image. Then geometrical measurements of the connected regions in each stack were taken as texture features. This approach is similar to that proposed in [79] for defect detection in ceramic tiles.

B Defect detection

Özdemir *et al.* [123] compared six texture features, consisting of MRF, KL transform, 2D Lattice filters, Laws filters, co-occurrence matrices, and a FFT-based method, for detecting textile defects. Texture modelling using a high (9th) order MRF model gave the best detection result. Iivarinen [63] demonstrated LBP and co-occurrence matrices features had similar performance in inspecting textured surfaces. Recently in [113], Monadjemi implemented three statistical (histogram-based, LBP, and co-occurrence matrices) and five signal processing schemes (Gabor filters, directional Walsh-Hadamard transform, discrete cosine transform, eigenfilters, and composition of Gabor filters) for randomly textured ceramic tile abnormality detection. The Gabor filter based composition scheme was found to be the most accurate method with good consistent performance across various tile types.

Although a solid conclusion can not be drawn to determine the best method for either texture analysis or defect detection or both, it is clearly evident that filtering approaches, in particular Gabor filtering, have been more popularly applied in these areas (cf. Table 2.1). However, an attractive idea is to use local neighbourhood pixel relationships to model the texture, e.g. using methods based on the LBP, MRF, or the epitome models. In fact, multi-channel filtering supports

the claim that the joint distribution of neighbouring pixels determines texture appearance, as the joint distribution of pixel values in the filter support window determines the distribution of the filter response [88]. Notably, Varma and Zisserman [176] demonstrated better performance in texture classification using small neighbourhoods than using filter bank-based approaches. Representing texture using primitives is also effective, for example the texton representation. However, due to the difficulties in explicitly deriving primitive representation and associated displacement rules, there are relatively limited works using structural approaches (cf. Table 2.1).

As image textures may often contain both statistical and structural properties, a texture analysis method should be able to represent both types of properties in order to completely describe the texture [62]. Model-based texture analysis methods can generally represent both properties, e.g. [199]. Statistical models and their estimation have recently been an attractive topic, for example [70].

In this thesis, a novel statistical model, the texem model, is proposed to learn texture primitives. Texems are implicit encapsulation of texture primitives based on local neighbourhood relationships.

2.3 Colour Tonality Inspection

In industrial quality inspection of colour texture surfaces, such as ceramic tiles or fabrics, it is also important to maintain consistent colour shade or tonality during production. Colour shade variations, although subtle, can still be discernible once the surfaces are put together. This is therefore another important quality factor in ceramic tile production. Colour tonality inspection must be carried out on both uniform pattern surfaces and randomly textured surfaces, but manual detection is not only tiresome but rather difficult. Problems such as spatial and temporal variation of the illumination may introduce effects which make colour grading even more difficult. Manufacturers have long sought to automate the identification process.

In [6], Baldrich *et al.* segmented the tile image into several stacks using a K-means approach. Then statistical measures were used to represent local and global colour information and segment chromatic and shape characteristics of blobs within each stack. However, this was designed for a specific family of grainy tiles and may not be applicable to other types of randomly textured tiles. In [97], Lumbreras *et al.* used wavelet transforms to assess different colour channels and various decomposition schemes to find appropriate features in order to sort tiles into perceptually homogeneous classes. The feature vectors were classified to the nearest class by using Fisher's linear discriminant function. Similar work has been reported in [4], using wavelet analysis in RGB

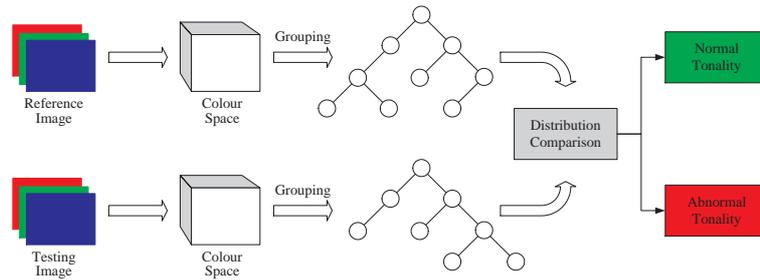


Figure 2.6: A flow chart of the colour histogram-based colour tonality inspection method [15, 16].

channels. The visual perception concerned with in these works, such as [6, 97, 4], include both textural and chromatic properties.

However, there are scenarios in which consistency of chromatic characteristics are as predominantly important as for visual perception, for example [15, 95]. Kauppinen [74] used RGB colour percentile features which were calculated from cumulative histograms to classify wood surfaces. Penaranda *et al.* [128] computed the first and second histogram moments of each channel of the RGB colour space as chromatic descriptors to classify tiles according to visual perception. Very recently, Lopez *et al.* [95, 96] used higher order histogram moments from each channel in $L^*a^*b^*$ colour space to characterise the colour tonality of ceramic tiles. In [13, 14], Boukouvalas *et al.* presented spatial and temporal constancy correction of the image illumination on the surfaces of uniform colour and two-colour (fix) patterned tiles. The luminance and the average colours in image channels, such as red, green, and blue, were used to perform tonality grading. Later in [15], the same authors proposed a colour histogram based method to automatically grade colour shade for randomly textured tiles by measuring the difference between the RGB histograms of a reference tile and each newly produced tile. By quantising the pixel values to a small number of bins for each band and employing an ordered binary tree, the 3D histograms were efficiently stored and compared. Figure 2.6 shows our simulated flow chart of their procedure which is important to our work later in Chapter 5 for comparative purposes. Several measures were investigated to perform the histogram comparison. Normalised cross correlation was found to be the most appropriate one as it gave the most consistent performance and also had a bounded range. This allowed the *a priori* definition of thresholds for colour shade.

Previous works, e.g. [15, 74, 95], show that global measurements, particularly colour histograms and their related statistics are useful in colour tonality defect detection. Smoothing to reduce noise interference (prehistogram computation) has also been found beneficial in colour shade discrimination [16]. In this thesis, colour histogram-based methods have been further explored by incorporating local chromatic features to discriminate subtle colour shade differences. The proposed method is compared against those in [15, 74].

2.4 Colour Texture Analysis

Due to the increasing computational power and availability of colour cameras, there are rising demands to use colour when necessary. There has been a limited but increasing amount of work on colour texture analysis recently (cf. Table 2.1).

Most colour texture analysis techniques are borrowed from methods designed for graylevel images, such as co-occurrence matrices and LBP. This extension of graylevel texture analysis techniques to deal with colour images usually takes one of the following forms:

1. Processing each channel individually by directly applying graylevel based methods [19, 55, 56, 100]: The channels are assumed independent to each other and only the spatial interactions are taken into account.
2. Decomposing image into luminance and chromatic channels [132, 127, 46, 115, 91]: Transforming the colour space so that texture features are extracted from the luminance channel and chromatic features from the chromatic channels, each in a specific manner. The selection of the colour space is usually application dependent.
3. Combining spatial interaction within each channel and interaction between spectral channels [141, 79, 126, 66, 159, 8, 111, 73, 60, 125]: The graylevel texture analysis techniques are applied in each channel, while the pixel interactions between different channels are also taken into account. Also, some works perform global colour clustering analysis, followed by spatial analysis in each individual stack.

Techniques independent of graylevel methods have also been attempted:

4. Using fully three dimensional models to analyse colour textures [70]: The spatial and spectral interactions are simultaneously handled. The main difficulties arise in effectively representing, generalising, and discriminating three dimensional data.

Caelli and Reye [19] processed colour images in RGB channels using multiscale isotropic filtering. Features from each channel were then extracted and later combined for classification. In [55], the author used the KL transform to decorrelate the RGB channels into orthogonal eigenchannels. A recursive MRF model was performed in individual channels for texture segmentation. Later in [56], Haindl and Havlicek used a similar approach for colour texture synthesis. Mäenpää *et al.* [100] measured colour percentiles based on the accumulated histogram in each RGB channel as chromatic features, and co-occurrence matrices and LBP features as textural features to inspect wood surfaces.

Several works transform the RGB colour space to other colour spaces to perform texture analysis so that chromatic channels are separated from the luminance channel, e.g. [132, 127, 46, 115, 91]. In [127], Paschos *et al.* projected the colour images into the xyY colour space. The two chromaticity coordinates (x,y) were combined into one, which provided the chromatic features. Texture features were extracted from the Y channel. Dubuisson-Jolly and Gupta [46] used a multi-resolution simultaneous auto-regressive model to compute the texture features. Very simple colour features were selected from the Yuv colour space. Similarly, Monadjemi *et al.* [115] used hue-like colour features, and Hadamard and Gabor transform texture features to classify outdoor scenes. Liapis *et al.* [91] transformed colour images into the $L^*a^*b^*$ colour space in which discrete wavelet frame transform was performed in the L channel. Local histograms in a and b channels were used as chromatic features. Recently, Tsai *et al.* [165] also transformed colour images into the $L^*a^*b^*$ space, from which two chromatic representations were derived for each pixel colour, hue and chroma (colourfulness). Gabor filtering was then performed in these two channels. The authors argued that processing images in these two chromatic channels only could be resilient to illumination changes. They assumed that defects were chromatically differentiable. However, a large set of defects occur due to intensity irregularities. For example, changes in gray shade will not introduce differences in hue and chroma.

The importance of extracting correlation between the channels for colour texture analysis has been addressed by several authors. One of the earliest attempts was reported in [141]. In [126], Panjwani and Healey devised a MRF model to encode the spatial interaction within colour channels and between colour channels. A similar idea was explored in [73] for unsupervised colour image segmentation. In [66], Jain and Healey used Gabor filters to obtain texture features in each channel and opponent features that capture the spatial correlation between channels. Thai and Healey [159] applied multiscale opponent features computed from Gabor filter responses to model intra-channel and inter-channel interactions. In [111], Mirmehdi and Petrou perceptually smoothed the colour image textures in a multiresolution sense before segmentation. Core clusters were then obtained from the coarsest level and initial probabilities were assigned to all the pixels for all clusters. A probabilistic reassignment was then propagated through finer levels until full segmentation was achieved. Simultaneous auto-regressive models and co-occurrence matrices have also been used to extract the spatial relationship within and between RGB channels [8, 60, 125]. In [79], the authors performed colour clustering, followed by binarised spatial pixel distribution analysis, to identify textural defects in colour ceramic tile images. The colour clustering and binarisation in the spatial domain partially takes into account both spatial and spectral interactions.

There is relatively limited effort to develop fully 3D models to represent colour textures. The 3D data space is usually factorised using one of the approaches mentioned above, then the data is modelled and analysed using lower dimensional methods. However, such methods inevitably suffer from some loss of spectral information, as the colour image data space can only be ap-

proximately decorrelated. As mentioned in Section 2.2.4, the epitome [70] provides a compact 3D representation of colour textures. The image is assumed to be a collection of epitomic primitives relying on raw pixel values in image patches. The neighbourhood of a central pixel in a patch are assumed statistically conditionally independent. A hidden mapping guides the relationship between the epitome and the original image. This compact representation method inherently captures the spatial and spectral interactions simultaneously. The epitome model inspired us to develop the texem model, a compact mixture representation of colour textures. Later in Chapter 5, the epitome model will be used to perform defect detection for comparative purposes.

Visual inspection using colour texture analysis is still largely under-developed in the literature and only a limited number of works have been reported so far. However, the demand for colour visual inspection is rising. In this thesis, several colour techniques with different computational complexities for inspecting tonality and textural faults are investigated. First, a direct extension of graylevel texems with relatively low computational costs is proposed by performing texems based local contextual analysis in decorrelated image eigenchannels. Next, a full 3D model with new formulations and inference procedures is introduced to generate texture exemplars from single or multiple images. The model is then more generally applicable and achieves better results at the cost of more computational power.

2.5 Classification and Novelty Detection

The primary goals of visual inspection are detection and classification. This involves choosing an appropriate decision making scheme which is usually referred to as pattern classification. Generally, this can be divided into supervised classification and unsupervised (or semi-supervised) classification. The following gives a brief review of these two approaches in relation to visual inspection.

2.5.1 Visual inspection via supervised classification

In supervised classification, the input pattern, based on features derived from earlier stages, is identified as a member of a pre-defined known class. This approach has been widely used in visual inspection, e.g. [184, 81, 151, 131, 114, 104, 95, 96].

The K -Nearest Neighbour (KNN) classifier is a simple nonparametric supervised distance-based learning algorithm where the pattern is assigned to the class shared by the majority of the K nearest neighbours. In [95, 96], Lopez *et al.* used KNN to classify ceramic tile surfaces based

on chromatic features extracted from individual channels. The authors also investigated various values of K in terms of classification accuracy. Mandriota *et al.* [104] also applied KNN to classify filter responses and wavelet coefficients to inspect rail surfaces. Contrary to [95, 96], the authors did not find any performance improvement on their dataset by increasing the value K . Wiltsh *et al.* [184] used a parametric minimum distance based classifier to inspect steel images. Latif-Amet *et al.* [85] also used a Mahalanobis distance based parametric classifier. Recently, Pernkopf [131] classified steel surfaces based on data likelihood computed from coupled hidden Markov random fields. In [21], Chan and Pang classified four types of fabric defects by fitting into the expected feature model.

Artificial neural networks have been extensively used in decision making procedures due to their ability to learn complex non-linear input-output relationships. In [81], raw pixel values in textile images were extracted from local neighbourhood as the textural feature for each individual pixel. PCA was then applied to the feature vectors to reduce the feature space dimension. Finally, a feed-forward neural network was used to classify each pixel. Recently, Monadjemi *et al.* [114] applied a back propagation neural network and KNN to classify ceramic tile surfaces using various texture features, such as co-occurrence matrices, LBP, Gabor filtering, eigenfiltering, and discrete cosine transform. They proposed a neural network that generally outperformed the KNN classifier. Another popular network is the Self-Organising Map (SOM), which is mainly used for clustering and feature mapping [67]. Nisknen *et al.* [119, 151] performed SOM based clustering of wood surfaces. However, although the clustering is unsupervised, the labelling of defect-free and defective samples in the SOM map was manually performed.

2.5.2 Visual inspection via novelty detection

In a novelty detection task, the classifier's task is to identify whether an input pattern is part of the data or it is in fact unknown. As for defect detection, it involves assigning a "normal" or "abnormal" label to a pattern (e.g. a surface or a pixel). Contrary to supervised classification, novelty detection only needs the normal samples for training purposes and usually uses a distance measure and a threshold for decision making. Recently, Markou and Singh [107, 108] gave a detailed review of novelty detection approaches, using statistical and neural network-based approaches.

Statistical parametric approaches are commonly used in visual inspection, for example [49, 167, 82, 12, 114]. The fundamental assumption is that the data distribution is Gaussian in nature. Thus, it can be easily statistically modelled based on means and covariances. As misclassifications can not be used as a criterion for the performance of a classifier as in a supervised manner, the available performance measure for novelty detection methods is the probability of false positives, that is rejection of good samples. Increasing the acceptance decision boundary will then obviously

decrease the risk. However, it is also clear that the probability of false negatives depends on the acceptance region. Thus, it is usual to set the minimum acceptance region according to a fixed false positive probability. For example, in a parametric classifier, the decision boundary can be set as $\mu \pm k\sigma$ with $k = 2$ or $k = 3$, which corresponds to 5.0% and 0.3% expected false positive rate. In some applications, the decision boundary is simply set as the maximum range of normal samples in the training stage, e.g. [83, 84].

Probabilistic approaches, e.g. Gaussian mixture models, use kernel functions to estimate general distribution of training patterns. Each pattern is usually represented as a point in a d -dimensional feature space, where d is the length of the feature vector. The parameters of the model are determined by maximising the likelihood of the training data, usually through Expectation Maximisation (EM) algorithms. The objective is then to establish decision boundaries in the feature space and reject patterns that fall in regions of low density. The decision boundaries are determined by the probability distribution of the patterns at training stage. Thus, they can be conveniently computed by examining data likelihoods. In this thesis, two different mixture models are used to measure the pattern likelihoods. Novelty detection is then accomplished by using simple parametric thresholding, determined automatically from training data.

2.6 Conclusions

This chapter established the necessary background in preparation for describing the proposed methods in this thesis. The review of existing visual inspection techniques was divided into two sections. One was concerned with *global* colour shade inspection, and the other with *local* textural defect detection. Traditional classification approaches and novelty detection approaches were compared in the context of defect detection. The research on visual inspection is diversive and ever-changing, however, the following observations can be made.

- Colour shade defect detection should be viewed differently to textural defect detection. It is a global chromatic issue. Among several others, multidimensional histogramming has proved an appropriate approach to this problem. However, chromatic features based on global information only may not be able to discriminate subtle colour shade differences.
- Filter bank based methods have been very popular in textural defect detection, but contextual analysis based on local neighbourhoods is a promising alternative approach.
- Colour texture analysis is increasingly applied in visual inspection, but limited work in this area has so far been reported. The majority of the existing methods decompose the colour image into separate channels and process them independently or with limited interactions.

-
- It is also notable that novelty detection is important in visual inspection where knowledge of defective patterns is usually incomplete and/or unavailable.

Chapter 3

IMAGE DATASETS AND PRE-PROCESSING

Data acquisition and preparation involves the choice of sensor, the pre-processing steps, data selection, and the establishment of the groundtruth. This thesis is based on a large number of ceramic tile images from three different sources, and in one case from three different acquisition systems. The VisTex [112] database has also been used for novelty detection later in chapter 6. This chapter will briefly discuss the image datasets used and their pre-processing, in particular, luminance compensation for ceramic tile images.

3.1 Image Datasets

The first image dataset (COLSHADE) introduced here was used for the colour tonality issue in ceramic tiles. The data was gathered by imaging tiles with a 2048 pixel resolution RGB colour line-scan camera. The image sizes vary from 800×600 pixels to 1000×1000 pixels corresponding to the physical size of the tiles. This tonality or colour shade test dataset comprises eight tile types, totalling 456 tiles, with known groundtruth obtained from manual classification by manufacturing floor experts. Samples of each type were divided into two categories: standard colour shade and off-shade, as listed in Table 3.1.

Figure 3.1 presents several ceramic tile images with normal and abnormal colour shades. The first column in the figure shows tile images with standard normal colour shade. The tile images in the middle column possess visually subtle offset in colour shades, while those in the last column show relatively larger colour shade deviations. Both off-shades are considered as defective and need to



Figure 3.1: Colour shade examples from the COLSHADE dataset - left column: tile images with standard normal colour shade; middle column: tile images with minor colour shade problem; right column: tiles with relatively larger colour shade off-set.

Table 3.1: Colour shade tile dataset: COLSHADE.

Tile Type	Number of Images		
	Normal Shade	Abnormal Shade	Total
1	16	32	48
2	20	40	60
3	20	40	60
4	20	40	60
5	16	32	48
6	20	40	60
7	20	40	60
8	20	40	60
Total	152	304	456

Table 3.2: Grayscale tile images for defect detection: GRAYSET.

No.	Tile Type	Number of Images		
		Normal Sample	Abnormal Sample	Total
1	PRAN1A	52	113	165
2	WRIB1A	42	236	278
3	TF1X1A	161	280	441
4	BM1A	8	7	15
5	LRSIDE	23	23	46
6	TSPI1A	15	334	349
7	CSA1	56	38	94
8	SALZ1A	96	28	124
	Total	453	1059	1512

be detected. The images from different families vary in terms of the layers of textures and the colour tones applied on the surface. Also, the texture decorations range from coarse to fine.

The second (GRAYSET) and the third (COLOURSET) tile datasets were collected for detecting textural defects. The images were captured by three different cameras and setups (each at a different site) as part of the MONOTONE project. The GRAYSET contained grayscale tile images captured by a monochrome line-scan camera in a factory environment in the UK. The spatial resolution of these images were 4 pixels/mm and the images were 512×512 pixels in size. There were eight different types of tiles, totalling 1512 tile images with 453 normal samples and 1059 abnormal samples as shown in Table 3.2.

The COLOURSET data comprised two parts. The first was a collection of 256×256 pixel images grabbed using a RGB colour camera in a laboratory environment in the University of Bristol. The second was a collection of 256×256 pixel images grabbed by a RGB line-scan camera with similar spatial resolution at two different manufacturing sites in Spain using the same inspection

system. As shown in Table 3.3, there were 1018 tile images from ten different families of which around half were defective.

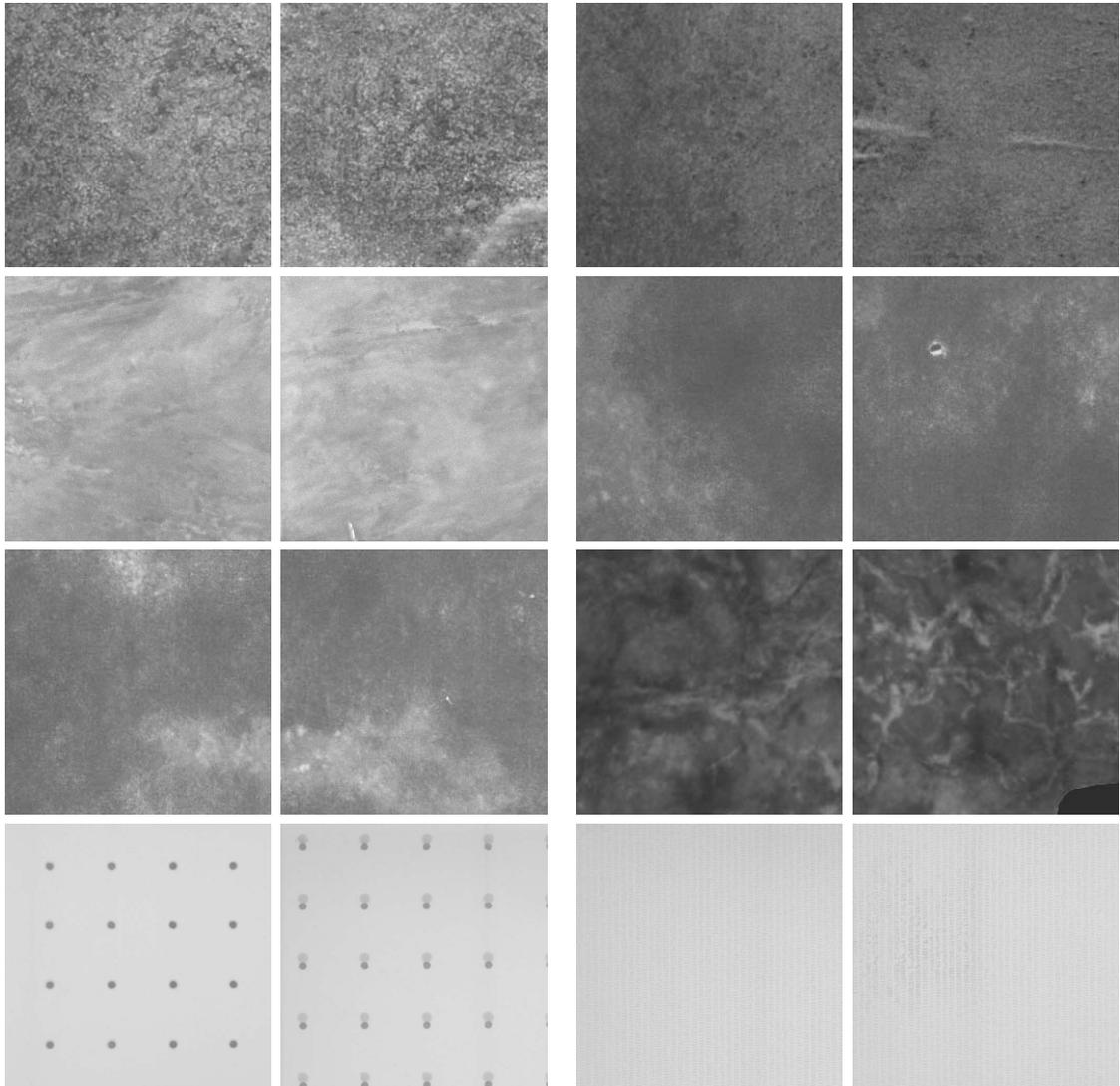


Figure 3.2: Examples of grayscale tile images with textural defects. The left image of each pair of images is a normal sample, while the right one is defective - from top left to bottom right: a random texture with a print error, a fine sand-like texture with physical defects, a marble texture with a small crack, a light texture with a print error, a cloud-like texture with pin holes, a marble texture with broken corner, a miss-registration problem, and a pattern irregularity issue in a grid pattern.

The GRAYSET data contained a wide range of texture decorations, some of which are shown in Figure 3.2. There were surfaces printed with stochastic textures, such as fine sand-like textures, and loose or strong coarse macro textures. There were also some regular patterns. The defects were of physical and textural nature, such as cracks, pin holes, bumps, misprints, missing print

Table 3.3: Colour tile images for defect detection: COLOURSET

No.	Tile Type	Number of Images		
		Normal Sample	Abnormal Sample	Total
1	PRAN	52	130	182
2	ARDES	12	10	22
3	BSAT	40	80	120
4	CSA	56	38	94
5	DJZU	10	10	20
6	GRAN	10	10	20
7	BJUP	40	52	92
8	KNGY	108	31	139
9	SONE	153	46	199
10	WBOT	80	52	132
Total		561	457	1018

and misregistrations.

Figure 3.3 shows several examples from the COLOURSET data with different types of textural and/or chromatic defects, including cracks, printing smudges, and discolourations.

All the ceramic tile images were manually classified into normal or defective categories. This was performed globally and subjectively by experts, and not at the pixel level. Therefore, groundtruth to define localised defect areas is not available. This was not only because the process is very time-consuming but also because it is indeterminate. The defective regions are usually much smaller than normal regions and the pixel based accuracy measurement would be biased towards specificity. Thus, in order to evaluate the spatial accuracy of the defect detection algorithms, the proposed method will be tested on texture collages made from textures in the MIT VisTex texture database [112]. Some example textures from VisTex are shown in Figure 3.4. Up to five different textures were used to produce each image collage. In each case, the background texture was considered as normal texture, while the foreground (disk, square, triangle, and rhombus) was treated as the novelty (defects) to be detected. The foreground was set to occupy 50% of the whole image to allow the sensitivity and specificity measurements not to be biased. Figure 3.5 shows all the 28 collage images used for novelty detection tests.

3.2 Accuracy, Sensitivity, and Specificity

The final result of the proposed method is the classification of previously unseen images into “defective” or “defect-free”. For image collages the accuracy was measured in pixels. The testing

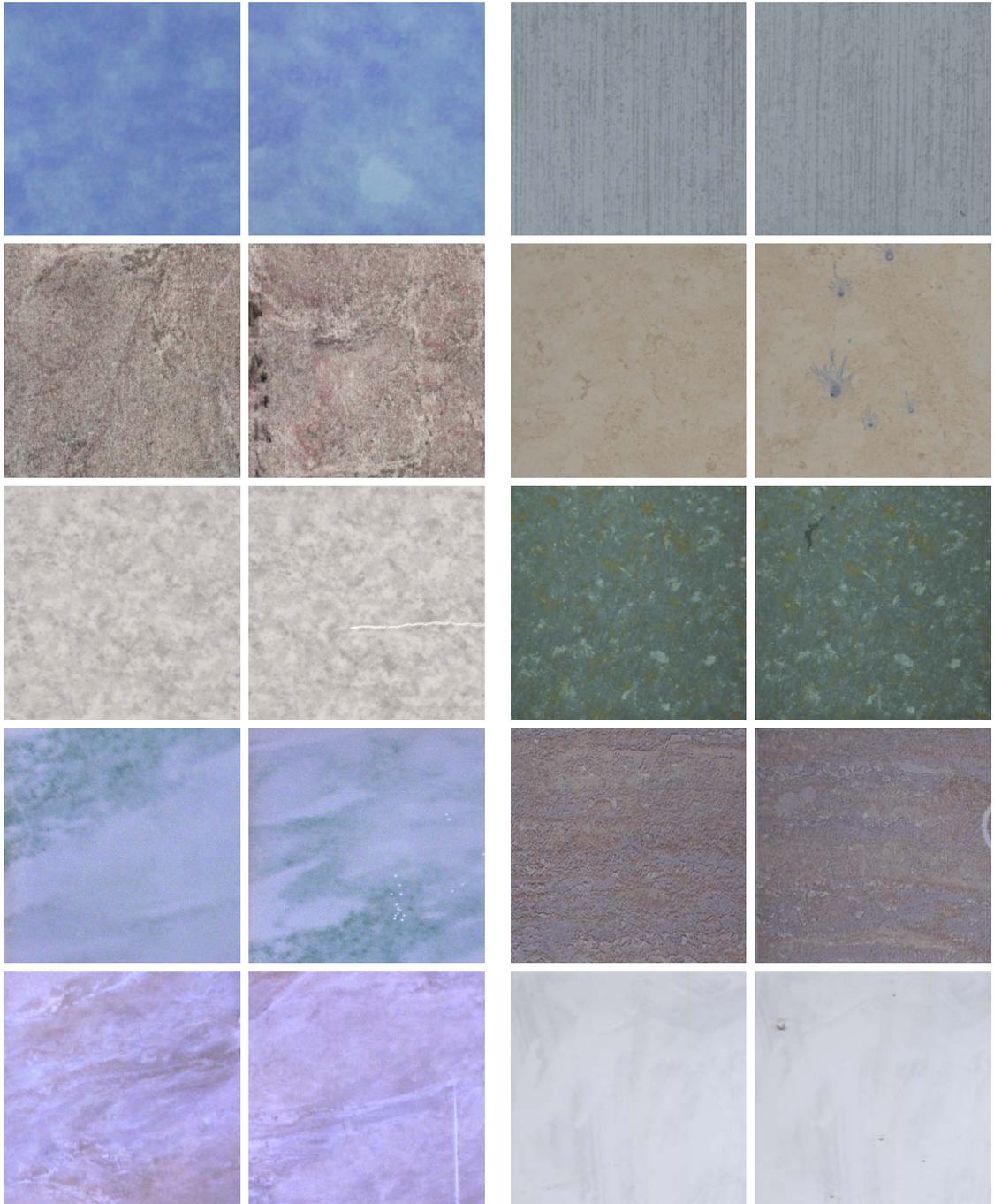


Figure 3.3: Examples of colour tile images with various textural and/or chromatic defects. The left image of each pair is a normal sample, while the right one is abnormal - from top left to bottom right: a cloud-like texture with a diffused defective region, a subtle printing error in the bottom right corner in an oriented texture, a major printing error along the edge of a tile decorated with dense patterns, several defective regions introduced by alien ink during printing, a thin and long crack across a random texture, a print error in a sand-like random texture, a cluster of pin holes in a marble texture, a missing print defect in a busy pattern, a print defect in a marble texture, and a few physical bumps in a lightly textured tile image.



Figure 3.4: ViTex sample images (captured from materials such as foods, bark, water, fabric, sand, and brick).



Figure 3.5: The complete set of collage images used for novelty detection evaluation.

results were quantified using *specificity* to show how accurately defect-free samples were classified, *sensitivity* to show how accurately defective samples were classified, and *accuracy* as the correct classification rate of all samples. They are defined as:

$$\begin{cases} \textit{specificity} = \frac{N_t \cap N_g}{N_g} \times 100\% \\ \textit{sensitivity} = \frac{P_t \cap P_g}{P_g} \times 100\% \\ \textit{accuracy} = \frac{N_t \cap N_g + P_t \cap P_g}{N_g + P_g} \times 100\% \end{cases} \quad (3.1)$$

where P is the number of defective samples, N is the number of defect-free samples, and the subscripts t and g denote the results by testing and groundtruth respectively. These measurements have been used throughout the thesis for evaluation and comparison.

3.3 Image Pre-processing

A major problem in any image inspection application is that of the image acquisition procedure. Due to the well-known *cosine-fourth* fall-off [61], vignetting effect [78], and other nonlinear interferences in the image acquisition chain, such as pupil aberration [3] and lighting non-uniformity, the intensity of the grabbed image is far from uniform.

The images captured at the University of Bristol were obtained by a radial RGB colour camera. It used a 2D illumination field which was difficult to adjust for lighting uniformity. For these images, the histogram specification technique [135] was used to correct the intensity across the images. This is briefly presented in section 3.3.1.

The majority of the tile images were captured by much higher resolution line scan cameras, installed in real factory environments. The behaviour of a line scan camera is significantly different from a radial camera. Sections 3.3.2 and 3.3.3 discuss the luminance correction methods for line-scan cameras using parametric and non-parametric approaches.

3.3.1 Luminance correction via histogram specification

The acquired images from the RGB radial camera were transformed into the CIE $L^*u^*v^*$ colour space to separate the luminance channel from the chromatic channels. The luminance channel was then split into patches with the constraint that the patch was relatively large but intensity was locally approximately homogeneous. The patch in the centre of a selected normal sample was treated as the reference patch based on the assumption that the centre of the sensor was least affected from luminance inconsistency introduced by the *cosine-fourth* effect. The remaining patches were then

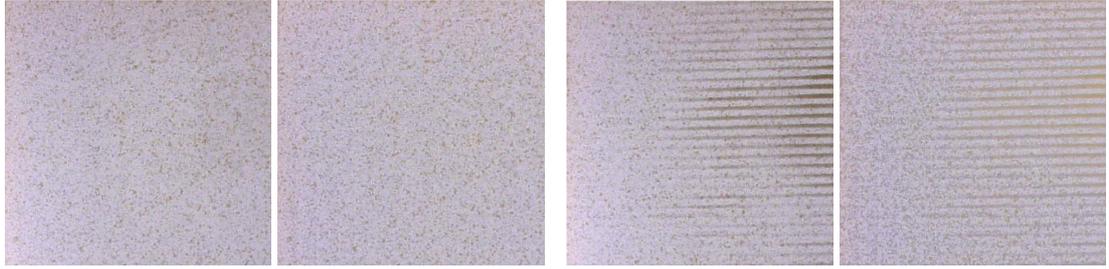


Figure 3.6: Luminance correction using histogram specification - from left: original defect-free tile image and its corrected version; original defective sample and its luminance corrected image.

corrected based on the reference patch using the histogram specification technique. Let $p_r(r)$ and $p_s(s)$ be the reference patch r and sample patch s intensity distribution functions respectively. Histogram equalisation was first used on both functions according to following transforms:

$$\begin{aligned} Q(r) &= \int_r p_r(x) dx \\ S(s) &= \int_s p_s(x) dx \end{aligned} \quad (3.2)$$

Then, by using the inverse transform of the reference patch, the desired intensity distribution is obtained:

$$\hat{s} = Q^{-1}(S(s)). \quad (3.3)$$

Figure 3.6 shows one “defect-free” and one “defective” tile sample and their luminance corrected versions. As the reference histogram is extracted from a normal sample, the correction will force the histograms of defective regions stretch towards the normal sample. As a result, defective regions may appear a little faded away (as can be seen in Figure 3.6). However, the chromaticity and local spatial structure of the defective regions are well-preserved because the correction is only performed in the luminance channel and it is not a pixel based compensation. Note, this method was designed for luminance correction of images from the radial camera used in this work only.

3.3.2 Parametric luminance correction

For the line-scan camera imaging system, two lighting cubes were used to produce a uniform illumination field, which was monitored by a special device along the camera acquisition line. The left image in Figure 3.7 shows a white tile with uniform reflectance imaged by the line-scan camera and a plot of its horizontal intensity profiles at all vertical pixel positions. These horizontal

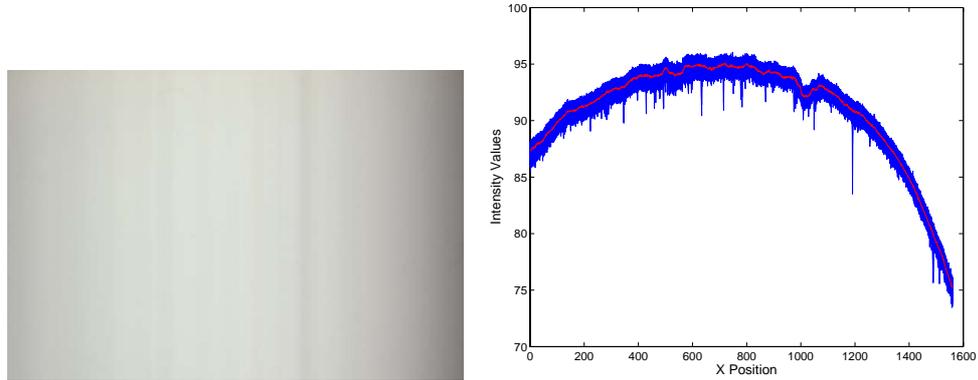


Figure 3.7: A white tile with nearly uniform reflectance captured by the imaging system and its horizontal intensity profiles across the tile (the median values at each pixel position are plotted in red curve).

profiles are parallel to the camera scan line. Figure 3.7 clearly shows that the intensity horizontally across the tile image is far from uniform. It can also be noticed that the temporal variation is very small. The average maximum difference for each pixel position is only about 2 or 3 gray levels across the whole image, however, spatially, the difference can be as large as 27 gray levels. This is mainly due to the well-known *cosine-fourth* fall-off.

The theory can be examined by fitting a cosine-fourth curve to these profiles:

$$y(x) = A \cos^4\left(\tan^{-1}\left(\frac{x_0 - x}{H}\right)\right) + y_0, \quad (3.4)$$

where x denotes the horizontal pixel position, x_0 is the centre of the camera, H is the height of the camera, and A and y_0 are constants. If this curve can fit the profiles well, the luminance inhomogeneity can be conveniently corrected by flattening the cosine-fourth curve. A major advantage would be that this compensation procedure only requires a few camera parameter settings, which are usually fixed during production, and no need for extra measurements. Thus, the Chi-square method was used to find the best fit. The variance of intensity, vertically, at each horizontal pixel position was treated as fitting weight. The Chi-square non-linear fitting result is shown in Figure 3.8 (left) where the red curve is the fitted cosine-fourth curve. The median intensity profile is plotted in blue. The cosine-fourth does not fit well as it loses local accuracy. It is clear that although cosine-fourth was the dominant factor of spatial inhomogeneity, there were several other factors that have to be taken into account, for example vignetting effect. Additionally, dust interference appeared to be critical. The dust in the factory environment can rest on the lens or the protective screen under the camera. The dark strip on the right side of the image in Figure 3.7 left was very likely introduced by dust on the camera lens. It also can be observed from the median profile in Figure 3.8 (left) where an abrupt drop breaks the continuity. Thus, cosine-fourth fitting is not a suitable solution. Other polynomial fittings could be used, however, it is difficult to accurately

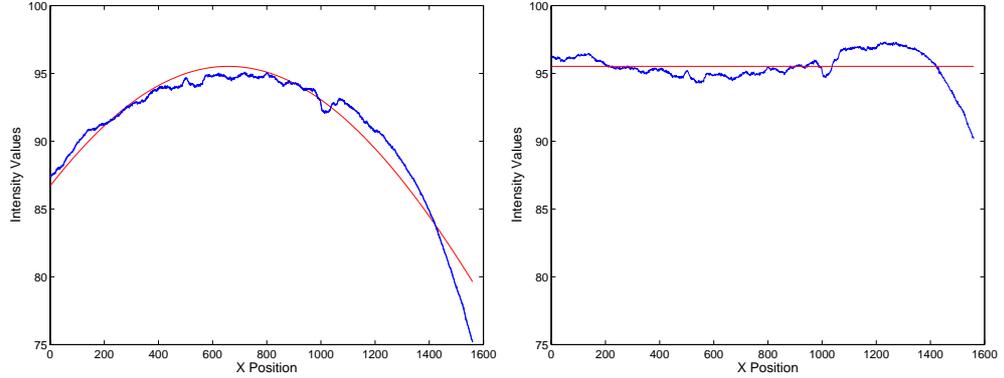


Figure 3.8: Cosine-fourth based correction - left: The blue line is the median profile of all the horizontal intensity profiles of the white surface, and the red curve is the Chi-square non-linear fitting result; right: The red line is the flattened cosine-fourth curve and the blue one is the corrected median intensity profile.

model the unpredictable local changes of the intensity along the scan line.

3.3.3 Non-parametric luminance correction

Instead of using polynomials to model the spatial (horizontal) intensity changes, one can directly rely on the image profiles (as seen in Figure 3.7) extracted from a standard white tile with uniform reflectance. Median values are chosen to be treated as a template profile as they are more robust to outliers than mean values. The surface selected was also ensured to be large enough to cover the full visual field of the line-scan camera. Let L_t denote the template median profile, and L_n the newly scanned pixel intensities. The intensity can then be easily corrected according to its relative position to the camera:

$$\hat{L}_n(x) = L_n(x) \times \frac{\max(L_t)}{L_t(x)}, \quad (3.5)$$

where x is the horizontal pixel position. Figure 3.9 shows an original median profile (in blue) of a sample tile and the corrected version (in red). The original luminance difference across the image was 9.75 graylevels out of 100 on an 800 pixel wide image, while the corrected image showed a 0.28 maximum graylevel difference.

To study the temporal behaviour of the median profile based correction method, 200 uniform colour tiles were imaged without parameter changes in the imaging system. The average intensities and standard deviations across the whole image are plotted in Fig 3.10. The temporal variations before and after correction were consistently small. However, the luminance correction helped to reduce the standard deviation of intensities across the whole image from an average of

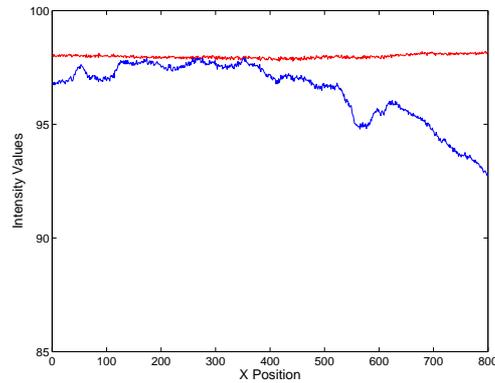


Figure 3.9: Luminance correction using a template median profile - curve in blue colour is the luminance profile before correction; curve in red colour is the profile after luminance compensation.

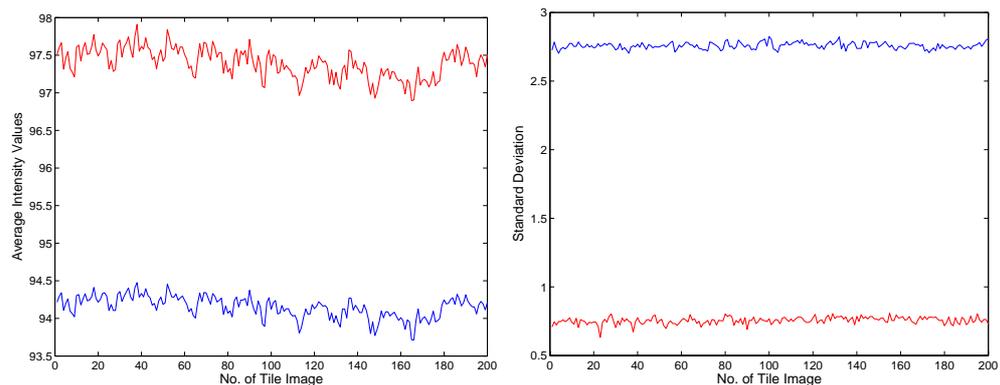


Figure 3.10: Luminance correction - left: A plot of the average intensities of those 200 images of a uniform coloured tile captured with no change in the imaging system before (in blue colour) and after (in red colour) luminance correction; right: A plot the standard deviations across the whole image before (in blue colour) and after (in red colour) correction.

2.76 to 0.76 gray levels. Spatial luminance correction alone was found to be sufficient for the application. However, it is still recommended that a new template profile should be acquired after a certain time period in case of any abrupt changes in the imaging system. Fig 3.11 shows two examples of luminance correction on tile surfaces with uniform colour and densely decorated textures. The profile based method outperformed the cosine fourth based method in that not only was the outcome more uniform but also the dark strip was removed.

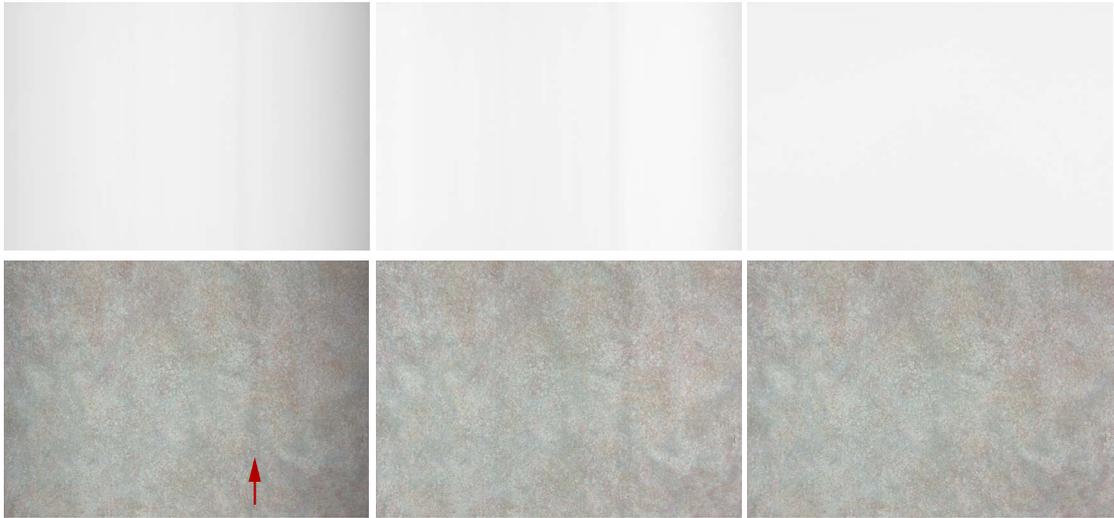


Figure 3.11: Cosine fourth based and template profile based luminance correction - First row: luminance correction on a uniform coloured surface. From left: Original tile image (a dark strip across the tile is indicated by a red arrow), cosine fourth based correction result, and a median template profile based correction result. Second row shows the results on a highly textured tile surface.

3.4 Summary

In this chapter, all the image datasets used in this thesis were introduced. The images range from medium to high resolution and include both monochrome and colour images.

A simple, yet effective and reliable luminance compensation method for line scan cameras was used. It was based on a uniformly coloured sample with near uniform reflectance property. Its median horizontal profile, parallel to the camera scan line, was considered as a template. Compensation coefficients were then calculated and associated with the camera scan line. Newly scanned pixel intensities were then corrected using these coefficients. As only the luminance needs to be compensated, no colour normalisation across the data sets was carried out. Due to its simplicity and efficiency, this approach can be used in real-time applications.

Chapter 4

COLOUR TONALITY INSPECTION

4.1 Introduction

Apart from *textural* faults, such as cracks, pin holes, undulations, misregistration, and misprints, inspecting *chromatic* tonality defects in terms of overall visual impression is also a significant production quality factor. This is concerned with assessing consistency among (tile) surfaces in terms of visual perception.

In ceramic tile production, tonality issues arise when parameter settings or environmental conditions change, such as, ink density, fluid viscosity, and temperature. These changes usually take place very gradually such that colour tonality on the printed surface does not appear to be different from one product to the next. It is difficult for a human inspector to notice the gradual changes and to objectively set a threshold to reject surfaces with off-standard tonality online. The problem is compounded when the surface of the object is not just plain-coloured, but textured. In short, colour shade irregularities on plain or textured surfaces are regarded as defects and manufacturers have long sought to automate the identification process.

A review of existing works on tonality inspection can be found in Chapter 2. The most relevant work to us on the consistency of product colour tonality has been by Boukouvalas et al., for example in [14, 15]. The authors proposed a colour histogram based method to automatically grade colour shade for randomly textured tiles by measuring the difference between the RGB histograms of a reference tile and each newly produced tile. Image quantisation and economic histogram storage were used to efficiently compare histogram distributions. Normalised cross correlation was found to be the most efficient comparison measure amongst several measurements that were studied. It gave the most consistent performance and also had a bounded range which allowed the *a priori* definition of thresholds for each colour shade.



Figure 4.1: Two examples of ceramic tiles with different colour shades from dataset COLSHADE - from left: The first two images in each row belong to the same colour shade, the last one in each row is an example of off-shade.

In fact, colour histograms have proved their worth as a simple, low level approach in many other applications, e.g. [158, 15, 134]. They are invariant to translation and rotation, and insensitive to the exact spatial distribution of the colour pixels. These characteristics make them ideal for use in application to colour shade discrimination, irrespective of the texture pattern. The colours on textured (tile) surfaces are usually randomly or pseudo-randomly applied. However, the visual colour impression of the decoration should be consistent from tile to tile. In other words, the amount of ink and the types of inks used for decoration of individual tiles should be very similar in order to produce a consistent colour shade, but the spatial distribution of particular inks is not necessarily fixed from one tile to the next (see Figure 4.1). Thus, colour histogram based methods are highly appropriate for colour shade inspection tasks.

In this chapter, a multidimensional histogram approach is presented to inspect colour tonality defects on randomly textured surfaces. In particular, the method is designed for discriminating subtle tonality differences which current state-of-the-art methods, such as [14, 15], find difficult to detect. Also, tonality inspection is treated as a novelty detection problem rather than a classification task (see [6, 97, 4, 128] where both the normal and abnormal samples are pre-defined). Another motivation is to develop a more accurate approach, particularly for textured designs.

The proposed method combines local colour distribution with global colour distribution to characterise the colour shade properties as part of the histogrammed data. A reference tile image is selected from a small set of good samples using a voting scheme. Initially, a vector directional processing method is used to compute the *Local Common Vector* amongst neighbouring pixels in the RGB space. This is first used to eliminate local noise and smooth the image. Then, a nine element feature vector is computed for each colour pixel in the image composed of the colour pixel itself, its local common vector, and its local colour variance. To minimise the influence of noise, Principal Component Analysis (PCA) is performed in this 9D feature space. The first few eigenvectors with the largest eigenvalues are selected to form the *reference eigenspace*. The colour features are then projected onto this eigenspace and used to form a multidimensional histogram. By projecting the colour features of an unseen tile onto the same reference eigenspace, a reconstructed image is obtained and histogram distribution comparison can be performed to measure the similarity between the new and the reference tiles.

4.2 Colour Shade Inspection

Figure 4.1 shows two particularly difficult abnormal tonality examples where the left and centre tiles in each row belong to the same colour shade class and are considered normal samples, while the right tile in each row is an “off-shade” example and should be detected as a defect. In developing our method of inspection the degree of noise in the tile images needs to be examined first.

4.2.1 Noise analysis

The problem of noise, which can be introduced in the imaging system chain and at the printing stage, requires special attention beyond facilitating uniform spatial lighting and temporal consistency during image capture. Noise interference will inevitably enlarge the intra-class variations, and make it more difficult to distinguish subtle colour shade differences. To recapitulate, the COL-SHADE dataset tiles were imaged by a 2048 pixel resolution RGB colour line-scan camera. The acquired image size varied from 600×800 pixels to 1000×1000 pixels corresponding to the physical size of the tiles.

To examine the nature of the noise in the image acquisition process, PCA was performed directly in the RGB space on a typical image from the dataset (the leftmost image in the top row of Figure 4.1). The pixel colours were then projected to the three orthogonal eigenvectors and finally mapped back to the image domain to obtain one image for each eigenchannel (see Figure 4.2). The first

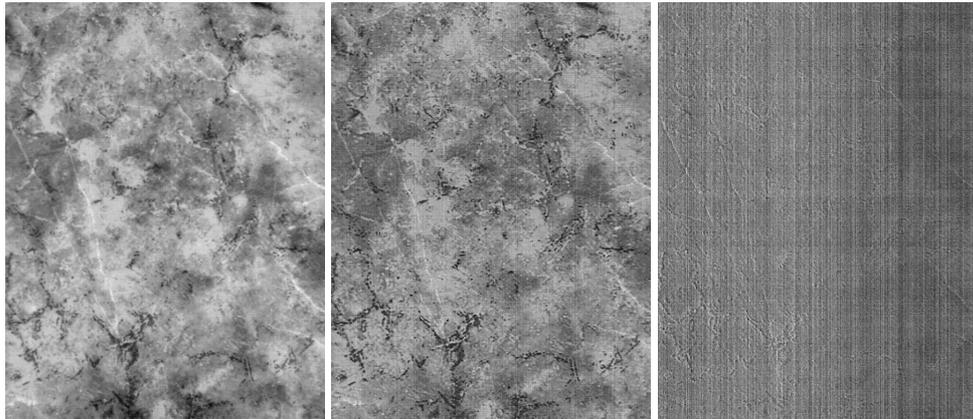


Figure 4.2: Image noise analysis showing the three eigenchannels. The noise is highly visible in the third channel. The images have been scaled for visualisation purposes.

eigenchannel presents the maximum variation in the RGB space, which in most cases is close to the intensity. The other two orthogonal eigenchannels mainly show the chromatic information with the last eigenchannel dominated by image noise. The vertical lines are introduced mainly by spatial variation along the line-scan camera's scan line and the horizontal lines are introduced by temporal variations, ambient light leakage, and temperature variations.

Clearly, the noise is present in all the image channels, but can dominate in (certain) chromaticity channels more than others. Direct comparison in the chromatic channels is likely to be error prone. For colour histogram based methods, each bin has identical weight and the image noise can make the distribution comparison unreliable when the colour shade difference is small, as inter-class difference is heavily diluted. For most tile images, the actual colours only occupy a very limited portion of the RGB space. In other words, variations in chromaticity are much smaller than those of brightness, but can still overwhelm the chromaticity. A variety of smoothing or diffusion methods can be used to explicitly minimise the negative effects of chromatic noise. Vector directional smoothing [161] was found to be an efficient approach for this purpose. Its underlying principle was adopted to compute the Local Common Vector (LCV) as a representation of a pixel neighbourhood.

Other noise can be introduced at the printing stage where subtle temporal inconsistencies usually occur. Further, temporal variations in the imaging system can also contribute to intra-class differences. Thus, more robust colour features, other than RGB alone, are necessary to characterise the colour shade. Decorrelating the image channels using PCA, as demonstrated in Figure 4.2, clearly revealed that the noise appeared in horizontal and vertical lines. Thus, median filtering is more appropriate than averaging. However, for vector-valued data, colour images, vector median has proved more effective than scalar median (in each individual channel) as vector median can not only remove outliers but also preserve edges in colour images [161]. In the next section, we use

vector directional processing to reduce the chromatic noise effect. However, on its own it is still not sufficient to distinguish subtle colour shade differences (this will be illustrated in section 4.4). In this work, LCV is used as the local salient chromatic feature, along with local statistics and pixel RGB colours themselves as an overall colour shade feature vector for colour shade inspection.

Next, the procedure for performing vector directional processing and obtaining the LCV is presented.

4.2.2 Vector directional median and LCV

Following the work in [161], as shown in Figure 4.3, a colour is represented as a vector in the 3 dimensional RGB space with the three primaries, R, G, and B, defined by the axes. The triangular plane connecting the three primaries in the RGB cube is known as the *Maxwell triangle*. The intersection point of a colour vector with the Maxwell triangle, marked as “×” in Figure 4.3, gives an indication of the chromaticity of the colour, i.e. its hue and saturation, in terms of the distance of the point from the vertices of the triangle [135]. As the position of the intersection point only depends on the direction of the colour vector, and not the magnitude, the direction then represents the chromaticity. The angle between any two colour vectors, e.g. between \mathbf{f}_1 and \mathbf{f}_2 in Figure 4.3, represents the chromaticity difference between them. So, the directional median of the set of vectors $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n$ within a window on the image can be considered as the vector that minimises the sum of the angles with all the other vectors in the set. The median is insensitive to extremes; as the vector direction (chromaticity) determines the colour perception, the noise due to the imaging system can be approximately suppressed using this median vector.

Let $\mathbf{I} : \mathcal{R}^2 \rightarrow \mathcal{R}^m$ be the image, a map from a continuous plane to the continuous space \mathcal{R}^m . For a colour image, $m = 3$. A window $W \in \mathcal{R}^m$ with a finite number of pixels is implied in calculating the directional median. The pixels in the processing window W are denoted as $\{\mathbf{c}_i, i = 1, 2, \dots, n\}$. The element $\mathbf{f}(\mathbf{c}_i)$, hereafter referred to as \mathbf{f}_i for convenience, is an m -dimensional vector in the space of \mathcal{R}^m . Thus the vectors in W define the input set $\{\mathbf{f}_i, i = 1, 2, \dots, n\}$. Let α_i be the sum of the angles between the vector \mathbf{f}_i and each of the vectors in the set. Then,

$$\alpha_i = \sum_{j=1}^n \mathcal{A}(\mathbf{f}_i, \mathbf{f}_j), \quad i = 1, 2, \dots, n, \quad (4.1)$$

where $0 \leq \mathcal{A}(\mathbf{f}_i, \mathbf{f}_j) \leq \pi/2$ specifies the angle between vectors \mathbf{f}_i and \mathbf{f}_j in a colour image. Generally, $0 \leq \mathcal{A}(\mathbf{f}_i, \mathbf{f}_j) \leq \pi$, but in the case of colour images, $0 \leq \mathcal{A}(\mathbf{f}_i, \mathbf{f}_j) \leq \pi/2$. Then, the ascending order of all the α s gives

$$\alpha_{(1)} \leq \alpha_{(2)} \leq \dots \leq \alpha_{(k)} \leq \dots \leq \alpha_{(n)}. \quad (4.2)$$

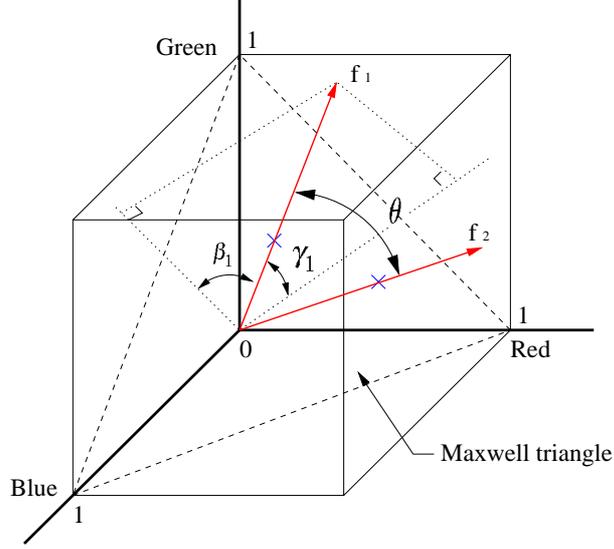


Figure 4.3: Perspective representation of the RGB colour cube.

The corresponding order of the vectors in the set is given by

$$\mathbf{f}^{(1)} \leq \mathbf{f}^{(2)} \leq \dots \leq \mathbf{f}^{(k)} \leq \dots \leq \mathbf{f}^{(n)}. \quad (4.3)$$

The first term in (4.3) minimises the sum of the angles with all the other vectors within the set and is considered as the directional median. Meanwhile, the first k terms in (4.3) constitute a subset of colour vectors which have generally the same direction. In other words, they are similar in chromaticity, but they can be quite different in brightness, i.e. magnitude. Thus, the vector with median magnitude, as suggested in [161], can be selected as the LCV:

$$\mathbf{f} = \arg \min_{\mathbf{f}^{(i)}} \sum_{j=1}^k |(|\mathbf{f}^{(i)}| - |\mathbf{f}^{(j)}|)|. \quad (4.4)$$

The value of k for the number of terms was empirically chosen as $\frac{W}{2}$. Alternatively, an adaptive method [160] can be used to determine its value. Thus, the LCV is computed in a sliding local window to smooth the image.

However, if the first k terms are also similar in brightness, the vector closest to $\mathbf{f}^{(1)}$ needs to be chosen. Thus, this concept is extended by considering the first k terms $\mathbf{f}^{(i)}$, $i = 1, 2, \dots, k$, (i.e. the ones with closest directionality to the median), and defining a simple measure so that the difference between any pair of vectors in the set is computed as

$$(|\mathbf{f}^{(i)}| - |\mathbf{f}^{(j)}|) + \min(|\mathbf{f}^{(i)}|, |\mathbf{f}^{(j)}|) \mathcal{A}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)}), \quad (4.5)$$

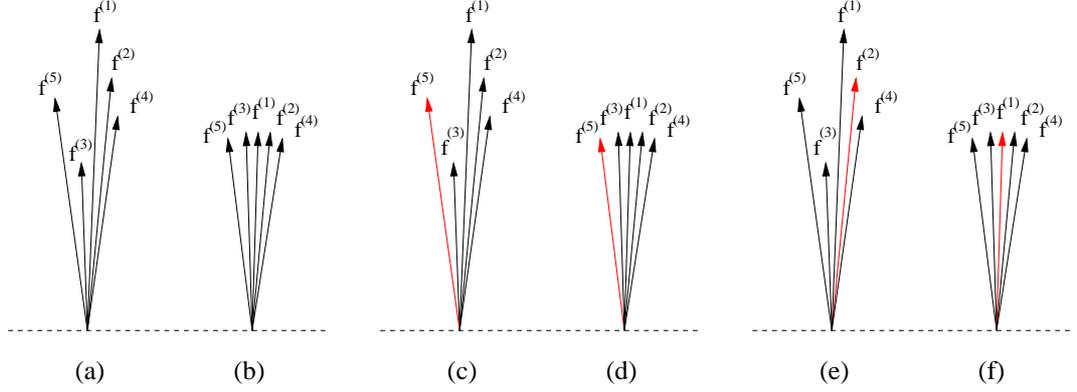


Figure 4.4: Refined LCV selection - (a) and (b) show two clusters of vectors with similar directionalities; (c) and (d) show the selected LCVs (in red) according to [161]; (e) and (f) show the selected LCVs (in red) using the proposed method.

Thus, when the magnitudes vary significantly in the subset $f^{(i)}, i = 1, 2, \dots, k$, the first term of (4.5) will dominate and the vector of median brightness will be selected. On the other hand, if the vectors in the subset have similar brightness, the second term of (4.5) will help select the LCV vector as the vector that has the least sum of chromaticity differences to other vectors. Figure 4.4 shows two clusters of vectors with similar chromaticities (directions). Median magnitude based selection will result in two vectors (as indicated in red) in Figure 4.4(c) and 4.4(d). The proposed combination of magnitude and directionality selection using (4.5) achieves a more accurate representation (as shown in Figure 4.4(e) and 4.4(f)).

For computational efficiency, the median magnitude only approach of [161] was used here as it was found to provide adequate accuracy for our application at lower costs. An example of this is shown in Figure 4.5. The LCV will also be used as a component of the colour feature vector applied to shade comparison as detailed in the next section.

4.2.3 Distribution comparison in eigenspace

Comparing global colour distributions between a reference tile and an unseen tile alone is not always enough, as subtle variations may be absorbed in the colour histograms. The evaluation of local colour distribution becomes a necessity.

Setting up the Reference - A reference tile is selected using a simple voting scheme (details in section 4.4). For any pixel c_i with its colour vector f_i , its brightness is represented by the magnitude $|f_i|$, and its direction (chromaticity) is determined by the two angles β_i and γ_i (that it makes with two of the axes in the RGB cube as shown in Figure 4.3). Thus, a nine-element colour

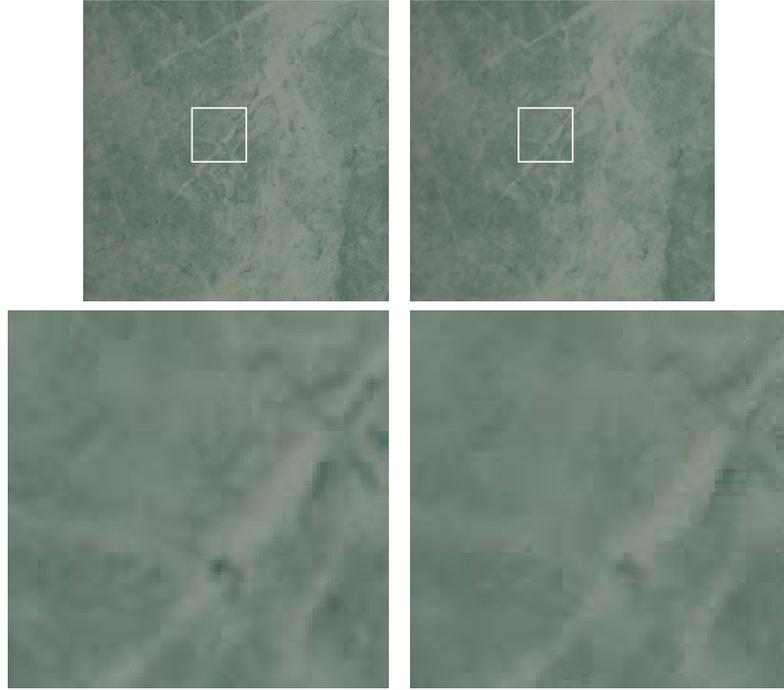


Figure 4.5: Vector directional smoothing using a 5×5 window - top row: original tile image and its smoothed version; bottom row: enlarged patches (marked in white from the top row) showing detail of the smoothing effect.

feature vector can be set up:

$$\mathbf{u}_i = [|\mathbf{f}_i|, \beta_i, \gamma_i, |\mathbf{f}_i|^0, \beta_i^0, \gamma_i^0, \sigma^{|\mathbf{f}_i|}, \sigma_i^\beta, \sigma_i^\gamma]^T, \quad (4.6)$$

comprising the magnitude and directions of the colour pixel itself, its LCV denoted as $(|\mathbf{f}|^0, \beta^0, \gamma^0)$, and the variances of the local colours in brightness $(\sigma_i^{|\mathbf{f}|})$ and chromaticity $(\sigma_i^\beta, \sigma_i^\gamma)$ measured against the LCV. The basis for this representation is that it is designed to encapsulate the local information at pixel-level which in turn can be used in a global multi-dimensional histogramming framework for the entire image.

Next, PCA is performed to determine a reference eigenspace for a reference “image” with each pixel represented by the feature vector \mathbf{u}_i . Let \mathbf{U} be the feature matrix containing q p -dimensional vectors, which are feature vectors derived from the colour tile image. Let w and h specify the dimensions of the image. Then $q = w \times h$, and p is the dimension of the feature space. Let $\bar{\mathbf{u}} = \frac{1}{q} \sum_{\mathbf{u} \in \mathbf{U}} \mathbf{u}$ be the mean vector of \mathbf{U} . The feature matrix \mathbf{U} is then mean-centred by deducting the mean vector $\bar{\mathbf{u}}$. Singular Value Decomposition (SVD) [80] can be used to obtain the matrix of eigenvectors given as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p] \in \mathcal{R}^{p \times p}$. The columns of \mathbf{E} are arranged in descending order corresponding to the eigenvalues $\omega_i, i = 1, 2, \dots, p$. Only $j, j < p$, eigenvectors with the largest eigenvalues are needed to represent \mathbf{U} to a sufficient degree of accuracy determined by a

simple threshold T ,

$$T = \frac{\sum_{i=1}^j \omega_i}{\sum_{i=1}^p \omega_i}, \quad (4.7)$$

with corresponding eigenvectors, $\mathbf{E}_j = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_j]$. The threshold T is usually empirically selected so that statistically significant features are retained. Thus, the resulting number of eigenvectors j is different from one type of texture to another. The subset thresholded with T is referred to as the reference eigenspace $\Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}$, where the colour features are well represented and surfaces with the desired shade should have a similar distribution. Characteristics not included in $\Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}$ are small in variation and likely to be redundant noise.

Colour feature comparison is now possible in this eigenspace for unseen tiles. The reference setup is completed by projecting the original feature matrix \mathbf{U} onto the reference eigenspace $\Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}$ (see examples later in Figures 4.7, 4.8, and 4.9):

$$\mathbf{U}' = \overrightarrow{PCA}(\mathbf{U}, \Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}) = \mathbf{E}_j^T (\mathbf{U} - \bar{\mathbf{u}} \mathbf{J}_{1,q}), \quad (4.8)$$

where $\mathbf{J}_{1,q}$ is a $1 \times p$ unit matrix consisting of all 1s, resulting in $\mathbf{U}' = \{\mathbf{u}'_i \in \mathcal{R}^j, i = 1, 2, \dots, q\}$. Also, the tile image can be reconstructed from this eigenspace through backward projection of a matrix $\mathbf{U}' \in \Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}$ onto the original feature space:

$$\hat{\mathbf{U}} = \overleftarrow{PCA}(\mathbf{U}', \Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}) = \mathbf{E}_j \mathbf{U}' + \bar{\mathbf{u}} \mathbf{J}_{1,q}. \quad (4.9)$$

Verifying Unseen Surfaces - For an unseen tile image, the same feature extraction procedure is performed to obtain the colour feature matrix \mathbf{V} . However, \mathbf{V} is then projected onto the reference eigenspace $\Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}$, resulting in \mathbf{V}' , i.e.

$$\mathbf{V}' = \overrightarrow{PCA}(\mathbf{V}, \Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}). \quad (4.10)$$

Note PCA is not performed on Y . This projection provides a mapping of the new tile in the reference eigenspace where defects will be expected to stand out. Finally, multidimensional histogram comparison is performed to measure the similarity between \mathbf{U}' and \mathbf{V}' in the reference eigenspace $\Phi_{\bar{\mathbf{u}}, \mathbf{E}_j}$. In [15], Boukouvalas *et al.* found that for comparing distributions of such kind the Normalised Cross Correlation (*NCC*) measure performs best as it is bounded in the range $[-1..1]$ and easily finds partitioning which assigns only data with acceptable correlation to the same class. For pairs of quantities $(\mathbf{u}'_i, \mathbf{v}'_i)$, $i = 1, 2, \dots, n$, then,

$$NCC = \frac{\sum_i (\mathbf{u}'_i - \bar{\mathbf{u}}') (\mathbf{v}'_i - \bar{\mathbf{v}}')}{\sqrt{\sum_i (\mathbf{u}'_i - \bar{\mathbf{u}}')^2} \sqrt{\sum_i (\mathbf{v}'_i - \bar{\mathbf{v}}')^2}}, \quad (4.11)$$

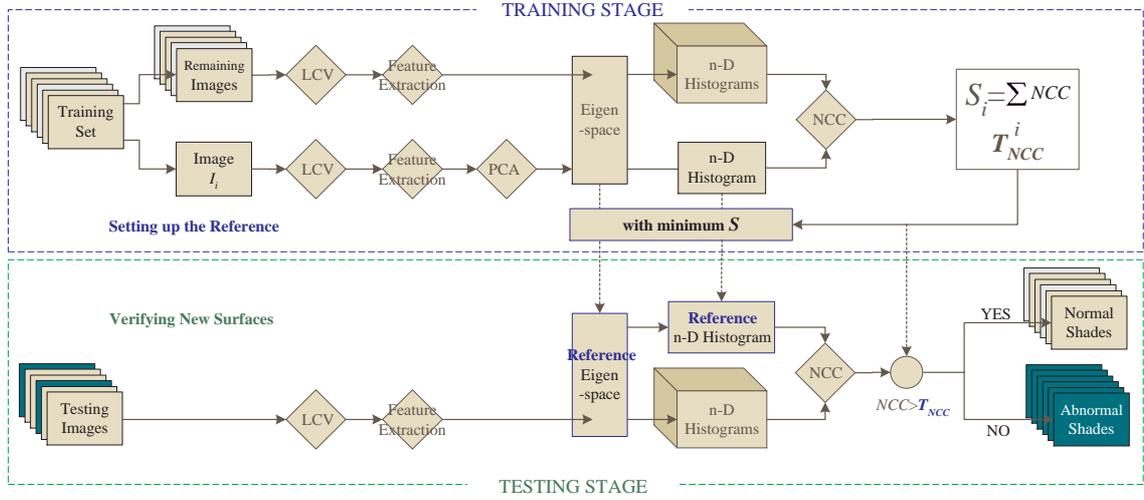


Figure 4.6: Flow chart of proposed method.

where $\bar{\mathbf{u}}'$ and $\bar{\mathbf{v}}'$ are the respective means. The NCC represents an indication of what residuals are to be expected if the data are fitted to a straight line using least squares. When a correlation is known to be significant, NCC lies in a pre-defined range and the partition threshold is easy to choose.

Direct multidimensional histogram comparison is computationally expensive, however for tile images, the data usually only occupies a small portion of the feature space. Thus, only those bins containing data are stored in a binary tree structure as shown in Figure 2.6 in Chapter 2.

4.3 Method Summary and Comments

The method presented is now summarised with the aid of Figure 4.6. The whole procedure comprises two stages: a training stage and a testing stage. The goal of the training stage (see top section of Figure 4.6) is to select the best representative of all the training images whose corresponding eigenspace and multidimensional histogram is treated as the reference data in the testing stage. The inspection starts with the selection of a reference tile using a voting scheme. First, a small number of good samples are each treated as a reference tile and compared with each other. Vector directional processing (as detailed in section 4.2.2) is used to compute the LCV. Then a nine element feature vector is extracted for each pixel, followed by PCA. Those eigenvectors with the largest eigenvalues (thresholded using T) are used to form the reference eigenspace. Colour features are then projected into this eigenspace as in (4.8), which results in a multidimensional

histogram for each tile image. Normalised cross correlation is performed to examine the similarity. Thus, for each tile image, a similarity measure is obtained, quantified by NCC , against each of the other tile images in the training set. The one with the least sum of NCC measures will be treated as the “golden sample”.

This entire process is illustrated in the top section of Figure 4.6 resulting in

$$\mathbf{I}_{\text{ref}} = \arg \min_{\mathbf{I}_i} \sum NCC, \quad (4.12)$$

where \mathbf{I}_{ref} is the selected reference image. In a practical scenario, this stage would be carried out on the first few good tiles manually selected on the production line. It also allows a threshold on the NCC value to be chosen to classify normal/abnormal colour shades. Let R_{NCC} denote the NCC value range for the reference tile. Then the threshold is selected as $T_{NCC} = 1 - \gamma R_{NCC}$. Here, γ was empirically chosen as 1.2. Alternatively, if a relatively larger number of training samples are available, the threshold can be $T_{NCC} = \mu_{NCC} - \gamma \sigma_{NCC}$ where $(\mu_{NCC}, \sigma_{NCC})$ are the mean and standard deviation of the NCC values for the reference tile, and $\gamma = 2$ or 3 in common practise.

The selection of the reference eigenspace is important. The selected reference eigenspace should not only represent the chromatic features of defect-free samples very well, but also be able to produce compact feature distributions for all training samples so that defective samples can be effectively detected. In other words, the NCC values for all training samples in this eigenspace should fall in a compact region close to 1. The selection process as described above ensures the selected eigenspace is the best representation in this context from all available individual eigenspaces. In fact, it can be viewed as the median eigenspace with respect to NCC measurement. One may obtain an eigenspace by projecting all the training images into the feature space followed by dimensionality reduction using PCA. The resulting eigenspace is then likely more sensitive to outliers. Thus, there is no guarantee that this eigenspace will result in a compact distribution of NCC scores. On the other hand, the proposed scheme explicitly searches for an eigenspace that produces a compact NCC distribution of training samples and yet models the whole variations in the training set.

The lower section of Figure 4.6 illustrates the testing stage, in which at first the LCV and feature vector of an unseen, novel tile image are extracted as in (4.6). These features are then projected into the reference eigenspace derived from the “golden sample” using (4.9). Finally, the NCC measure can be applied as an indication of the similarity between the test image and the reference, which is then used in comparison to T_{NCC} to classify the tile.

For comparison, and to demonstrate the function of equations (4.8) and (4.9), the tile image can be reconstructed by mapping the colour features in the eigenspace back to the RGB space. Taking the

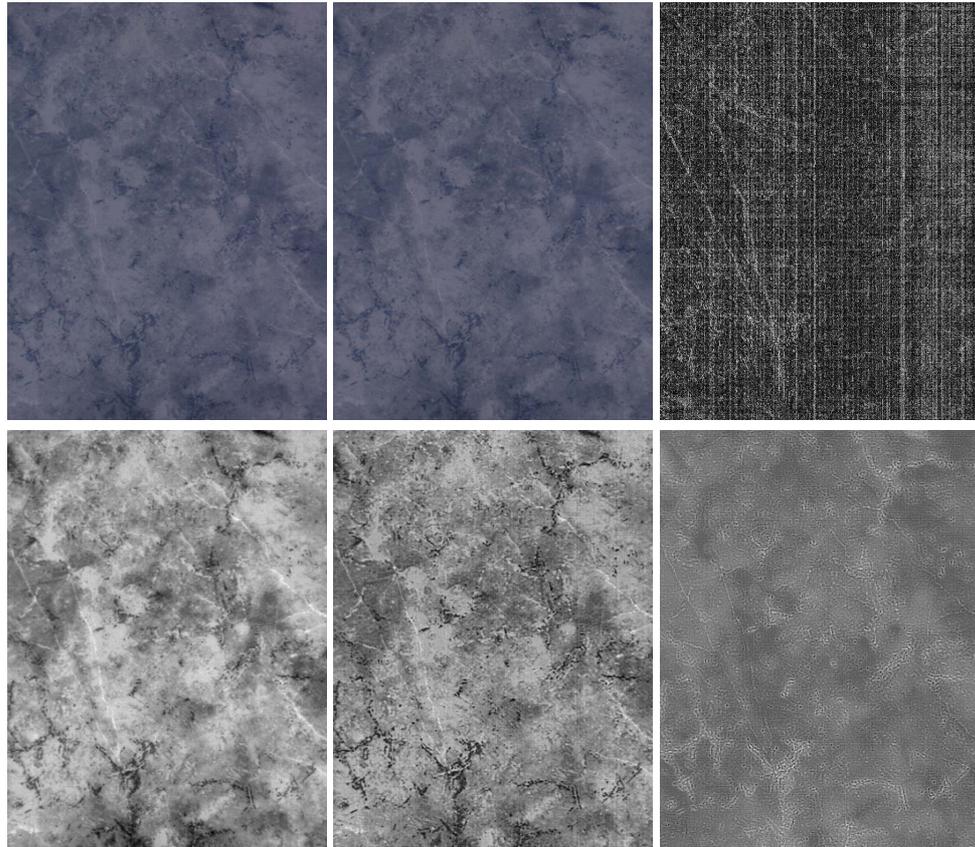


Figure 4.7: Image reconstruction - top: the original image, its reconstruction, and their MSE difference - bottom: the three eigenchannels of the reconstructed tile. The last channel shows texture structure, instead of being dominated by noise (cf. Figure 4.2).

leftmost image in Figure 4.1 as the reference image providing \mathbf{U} , the reconstructed colour features are $\hat{\mathbf{U}} = \overleftarrow{PCA}(\mathbf{U}', \Phi_{\mathbf{u}, \mathbf{E}_j})$. Then taking the first three feature elements $[[\mathbf{f}_i], \beta_i, \gamma_i]^T$, converting to RGB representation $[r_i, g_i, b_i]^T$, and mapping back to the image domain gives the reconstructed tile image, as shown in Figure 4.7(top-centre) along with the Mean Square Error (MSE) between the original and the reconstructed images, in Figure 4.7(top-right). This clearly shows the noise associated with the less significant eigenvalues. Next, noise analysis in the reconstructed image is performed (as in Section 4.2.1 and in Figure 4.2) showing that its third channel is *much less* noisier (bottom row of Figure 4.7). The reconstruction shows that reliable, salient colour features are obtained by the proposed method.

Figure 4.8 shows another example of a different type of tile pattern, which has a little less intensity variation and a bit more chromatic variation than the running example case used so far. The first row shows a good sample that is a reference tile \mathbf{U} , the reconstructed image $\hat{\mathbf{U}}$ which is projected back from its own (reference) eigenspace $\Phi_{\mathbf{u}, \mathbf{E}_j}$, and the reconstruction MSE error map, respectively. The second row shows the three eigenchannels of the original tile image. The textural

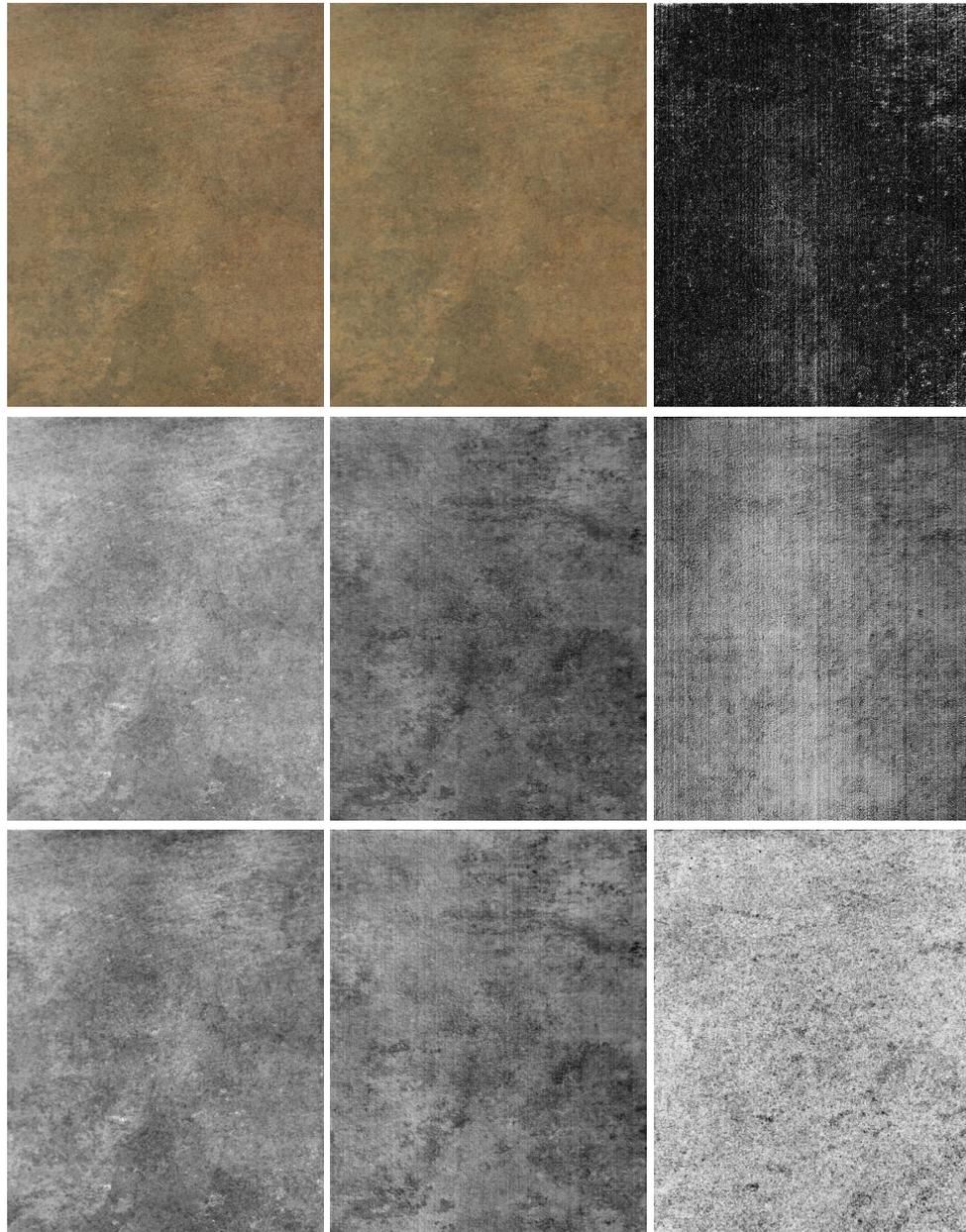


Figure 4.8: Another example of tile image - top row: The original tile image, the reconstructed tile image, and the difference between them using MSE - middle row: the three eigenchannels of the original tile image - bottom row: the three eigenchannels of the reconstructed tile image.

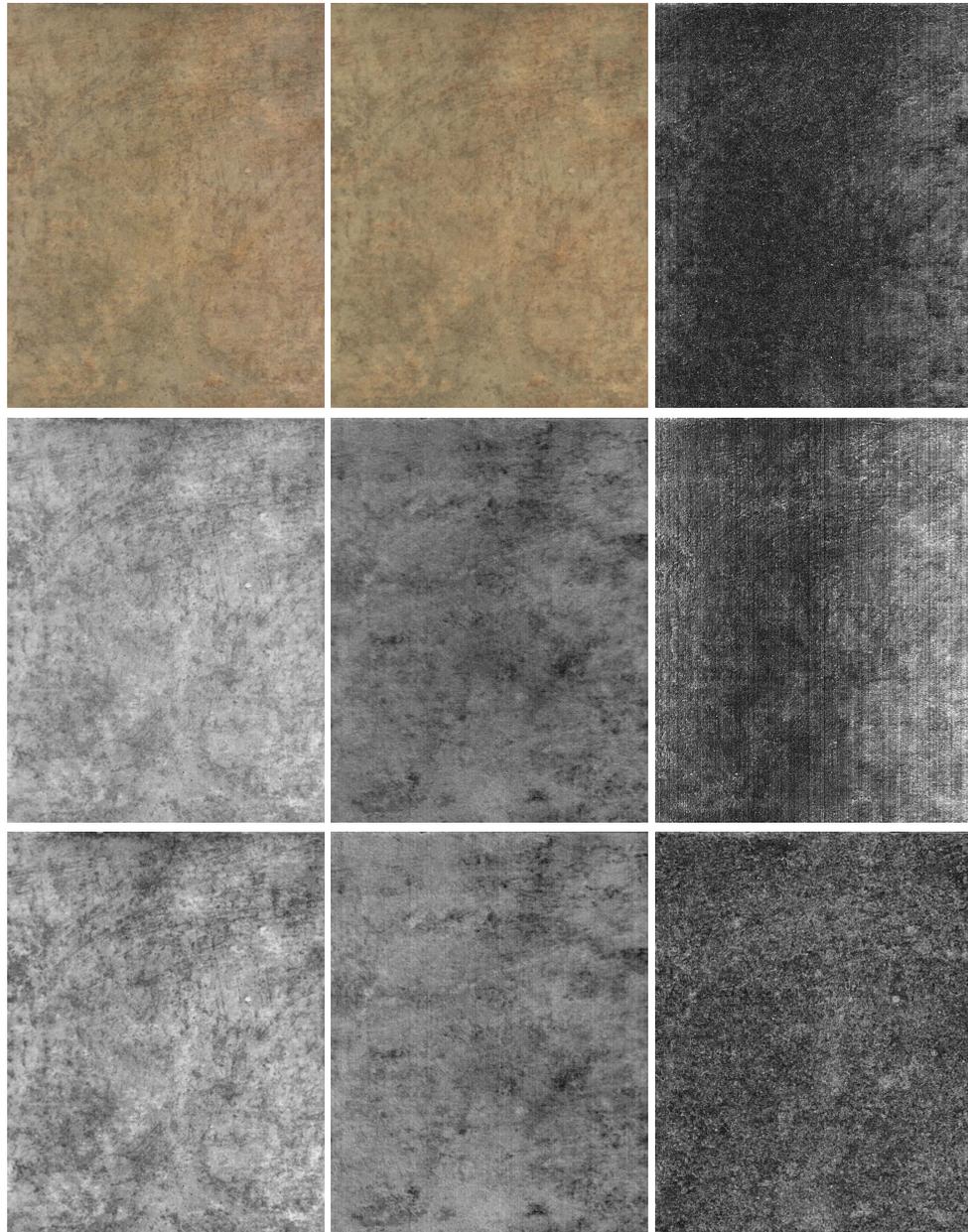


Figure 4.9: Reconstruction of an off-shade tile image - top row: The original off-shade tile image, the reconstructed tile image based on an eigenspace derived from a normal shade tile image, and the difference between them using MSE - middle row: the three eigenchannels of the original off-shade tile image - bottom row: the three eigenchannels of the reconstructed tile image.

Table 4.1: Testing results of colour histogram methods (values are %s)

Tile Type	RGB Colour Space		
	specificity	sensitivity	accuracy
1	93.8	96.9	95.8
2	90.0	95.0	93.3
3	95.0	95.0	95.0
4	90.0	87.5	88.3
5	93.8	84.4	87.5
6	90.0	80.0	83.3
7	90.0	87.5	88.3
8	80.0	72.5	75.0
Overall	90.1	87.2	88.2

information of the last eigenchannel is swamped by the image noise. However, the eigenchannels of the reconstructed tile image, shown in the last row, clearly exhibit the textural structure in less noisy detail.

The tile image shown in Figure 4.9 is in fact an off-shade defect case of the normal shade example in Fig 4.8. The reconstruction, shown in the top-centre, is based on the reference eigenspace $\Phi_{\mathbf{u}, \mathbf{E}_j}$ derived from the reference tile image in Fig 4.8, i.e. $\hat{\mathbf{V}} = \overleftarrow{PCA}(\mathbf{V}', \Phi_{\mathbf{u}, \mathbf{E}_j})$. It is evident that the eigenchannels of the reconstructed tile image are much less consumed by image noise. However, its distribution in the nine-dimensional feature space differs from that of the reference tile image. Thus the NCC measure stands a good chance in discriminating the difference.

4.4 Implementations and Results

The COLSHADE dataset comprised eight tile types, totalling 456 tiles, with known groundtruth obtained from manual classification by experts. Within each set, one-third of the tiles were standard colour shade and two-thirds off-shade (see Chapter 3 for more details). The proposed method was compared with a standard histogram-based method, with and without using vector directional preprocessing. The $L^*a^*b^*$ colour histogram was also investigated. Results were quantified using *specificity* to show how accurately good tiles are classified, *sensitivity* to show how accurately defective tiles are classified, and *accuracy* as the correct classification rate of all tiles. The mathematical definitions of specificity, sensitivity, and accuracy are given in Chapter 2.

The inspection started with the selection of a reference tile using a voting scheme amongst the best of the first few tiles in the production run (visually confirmed to be of high quality) as described in Section 4.3. The threshold T_{NCC} was chosen during this process as the criteria for colour shade variation.

Table 4.2: Testing results of colour histogram methods in $L^*a^*b^*$ space (values %s)

Test Num.	$L^*a^*b^*$ Colour Space		
	specificity	sensitivity	accuracy
1	93.8	96.9	95.8
2	95.0	92.5	93.3
3	95.0	92.5	93.3
4	90.0	90.0	90.0
5	93.8	84.4	87.5
6	80.0	85.0	83.3
7	95.0	82.5	86.7
8	85.0	70.0	75.0
Overall	90.8	85.2	87.9

Table 4.3: Testing results of vector directional smoothing (values are %s)

Tile Type	Window size: 3×3			Window size: 5×5		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
1	100	96.9	97.9	100	100	100
2	95.0	95.0	95.0	95.0	95.0	95.0
3	100	97.5	98.3	100	97.5	98.3
4	100	80.0	86.7	85.0	92.5	90.0
5	93.8	87.5	89.6	93.8	87.5	89.6
6	90.0	90.0	90.0	95.0	90.0	91.7
7	95.0	92.5	93.3	85.0	92.5	90.0
8	80.0	72.5	75.0	80.0	75.0	76.7
Overall	94.1	88.8	90.6	91.5	91.1	91.2
Tile Type	Window size: 7×7			Window size: 9×9		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
1	100	96.9	97.9	100	96.9	97.9
2	95.0	95.0	95.0	95.0	95.0	95.0
3	100	95.0	96.7	100	95.0	96.7
4	85.0	92.5	90.0	70.0	100	90.0
5	87.5	87.5	89.6	87.5	87.5	87.5
6	95.0	90.0	91.7	95.0	90.0	91.7
7	85.0	92.5	90.0	85.0	92.5	90.0
8	75.0	75.0	75.0	75.0	77.5	76.7
Overall	90.1	90.5	90.4	88.2	91.8	90.6

The results of four experiments were compared. In the first, named experiment RGB, the work by Boukouvalas *et al.* [15] was followed. In the second, the same experiment was carried out, but the $L^*a^*b^*$ colour space was used instead of the RGB colour space. This experiment thus was called LAB. Next, in the third experiment, the first experiment was repeated but this time with vector directional smoothing. Colour shade inspection was then based on features extracted from vector

Table 4.4: Testing results of comparison in feature eigenspace (values are %s)

Tile Type	Window size: 3×3			Window size: 5×5		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
1	100	100	100	100	100	100
2	95.0	97.5	96.7	100	95.0	96.7
3	100	95.0	96.7	100	97.5	98.3
4	95.0	92.5	93.3	90.0	95.0	93.3
5	100	96.9	97.9	93.8	100	97.9
6	95.0	92.5	93.3	90.0	92.5	91.7
7	95.0	90.0	91.7	95.0	92.5	93.3
8	85.0	82.5	83.3	85.0	82.5	83.3
Overall	95.4	93.1	93.9	94.1	94.1	94.1
Tile Type	Window size: 7×7			Window size: 9×9		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
1	100	100	100	100	100	100
2	95.0	97.5	96.7	95.0	97.5	96.7
3	100	100	100	100	100	100
4	95.0	90.0	91.7	95.0	90.0	91.7
5	93.4	100	97.9	93.8	100	97.9
6	95.0	92.5	93.3	85.0	97.5	93.3
7	95.0	90.0	91.7	95.0	90.0	91.7
8	85.0	90.0	88.3	80.0	90.0	86.7
Overall	94.7	94.7	94.7	92.8	95.4	94.5

representations. This experiment was named VDS. In the final experiment, called ESF, the full proposed method was applied using eigenspace features. The selection of template samples, the training and testing stages for all four experiments were the same as described in Section 4.2.3.

Experiment RGB: First, an RGB colour histogram based method similar to that described in [15] was performed. The colour images were projected from the image plane into the RGB space to produce the 3D histogram. For efficiency, each bin of the histogram was converted into a unique integer so that the 3D histogram could be economically stored in a binary tree. Thus the experiment from [15] was recreated here for comparative purposes. Table 4.1 shows the results of the RGB colour histogram based method, providing an average accuracy of 88.2%.

Experiment LAB: Other colour spaces were also considered. The $L^*a^*b^*$ colour space was particularly chosen as it can achieve perceptual uniformity and is popularly used. Table 4.2 shows the results of tonality inspection in the $L^*a^*b^*$ colour space, where histogram comparison was not as good as in the RGB colour space, with an average of 87.9%. This was mainly because the a and b chromatic channels were influenced by image noise. Also, the main characteristic of transforming into $L^*a^*b^*$, perceptual uniformity, was no longer advantageous in distribution

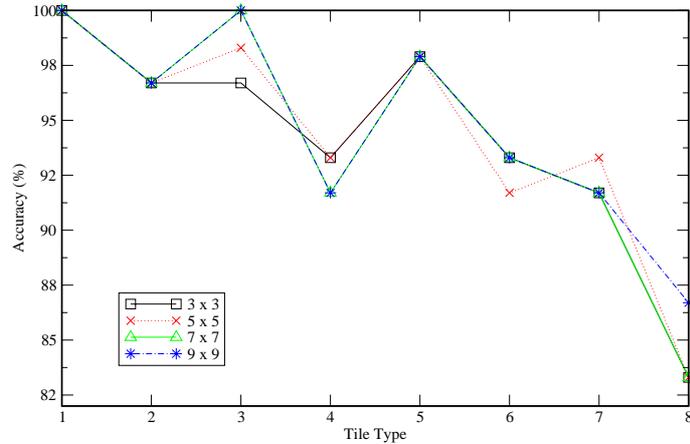


Figure 4.10: Testing results using eigenspace features with various window sizes.

comparison as no distance measurements were taking place.

Experiment VDS: In this experiment, vector directional smoothing was applied, and the $(|\mathbf{f}|, \beta_i, \gamma_i)$ representation of the colour pixel was used as colour tonality feature. Hence, the global histogram comparison is based on the distribution of magnitudes and vector directions. The results are presented in Table 4.3. Different window sizes were tested, from 3×3 to 9×9 with the best results achieved using a window size of 5×5 at 91.2% accuracy. However, it is notable that each texture has its own optimum window size due to different texture characteristics, such as coarseness. An improvement of 3.0% with window size 5×5 was achieved on the RGB experiment. Smoothing the colour images proved to be beneficial as it decreased the negative effects introduced by noise in chromaticity.

Experiment ESF: In the last experiment, when the full 9D feature vector was used and the proposed method was implemented in full, significantly better tonality defect detection results were obtained as shown in Table 4.4. By incorporating the local colour information and comparing the dominant colour features using a high dimensional histogram based method, an overall accuracy of 94.7% was achieved using a 7×7 window. In comparison to [15], this signifies an improvement of 6.5% in overall accuracy with window size 7×7 on our dataset.

Figure 4.10 shows a plot of the results of the proposed method using different window sizes. Since the textures in the different tile types are of various nature in terms of coarseness and density, clearly each performs better with a particular window size. Here it is concluded that on average, and in the interest of computational costs, a window size of 3×3 can be employed at all times.

The results of tile set 8 were consistently lower than those of others for all the four experiments. This is due to the nature of the texture printed on these tiles which have slightly larger intra-class variations. Using multiple “golden samples” may further improve the performance.

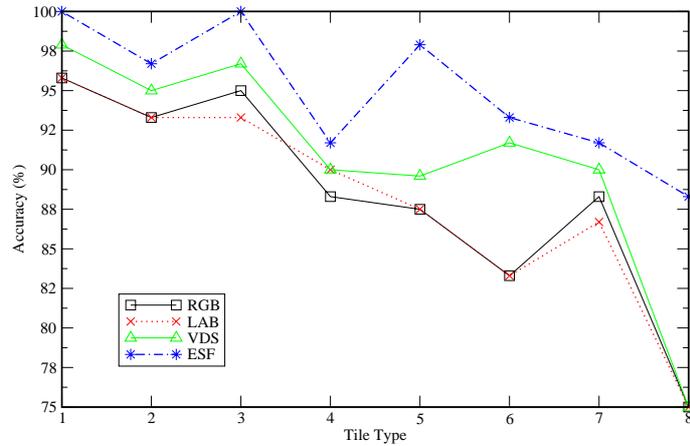


Figure 4.11: Colour tonality results comparison - A graphical plot of testing results using RGB colour histogram (RGB), $L^*a^*b^*$ colour space (LAB), vector directional smoothing (VDS), and eigenspace features (ESF).

Figure 4.11 shows performance comparison between the RGB and $L^*a^*b^*$ colour histogram based methods, vector directional smoothing only, and the full proposed method. Smoothing the data proved beneficial. The proposed method consistently outperformed the rest across all the tile images.

For practical implementation the inspection needs to run at approximately 1 second/tile depending on the size of the tile. Currently, the bottleneck in the system is in the LCV computation. Optimised 3D histogramming using ordered binary trees requires just less than 1 second per 1000×1000 pixel tile. The proposed method requires a computational time in the order of 20 seconds/tile at present: 0.98 seconds for its histogramming, 18 seconds for LCV computation and smoothing, and 0.94 seconds for NCC computation. This was computed on an AMD Athlon XP Processor (1.4GHz) with 512M memory. However, the computation time can be greatly reduced using dedicated hardware and optimised software, e.g. we project that by implementing the accelerating coding scheme mentioned in [160], the computational costs for finding the LCV in a 7×7 window can result in a saving of 82% without compromising accuracy.

4.5 Conclusions

An automatic colour tonality defect detection algorithm for randomly textured surfaces was presented. The shade problem is defined here as visual perception in colour, not in texture. The image noise, introduced by the imaging system, was revealed through eigenchannel analysis and was found mainly to affect the chromatic channels. For colour tonality inspection, the difference between images is very subtle and comparison in the noise dominated chromatic channels

is error prone. Vector-ordered colour smoothing was performed and a localised feature vector at each pixel position was extracted as a new representative colour feature. The resulting histogram represented an encapsulation of local and global information. PCA was performed on this multi-dimensional feature space of an automatically selected reference image to obtain reliable colour shade features, resulting in a reference eigenspace. Then unseen product images were projected onto this eigenspace and compared for tonality defect detection using histogram comparison. The chromatic channels of the reconstructed image were found to be much less dominated by noise. The proposed method was compared and evaluated on a data set with groundtruth. A window size as small as 3×3 gave an overall accuracy of 93.9%. However, the increase in accuracy comes at a computational cost which is hoped will be overcome through more optimised code and faster hardware. The proposed method should be applicable to other flat colour textured surfaces where the tonality issue is of importance, for example, wood and textiles.

Chapter 5

DEFECT DETECTION USING LOCAL CONTEXTUAL ANALYSIS

5.1 Introduction

Detecting textural defects is concerned with identifying regions that deviate from normal samples according to certain criteria, e.g. pattern regularity or colour. As mentioned earlier in Chapter 1, this detection process should be viewed differently to (a) texture segmentation, which is concerned with splitting an image into homogeneous regions, and (b) texture classification, in which texture classes are pre-defined. Neither the normal regions nor the defective regions have to be texturally uniform. For example, a surface may contain totally different types of defects which are likely to have different textural properties. On the other hand, a defect-free sample should be processed without the need to perform “segmentation”, no matter how irregular and unstationary the texture.

The detailed review in Chapter 2 pointed out that filter bank based approaches have been widely used in texture analysis and textural defect detection. In particular, Gabor filters have been applied, as for example shown in [49, 83, 113, 165], due to their ability to analyse texture by achieving optimal joint localisation in the spatial and frequency domains.

However, the supremacy of filter bank based methods for texture analysis have been challenged by several authors. They advocate that by examining the distribution of local measurements, textures with even global structures can be discriminated. Some examples of such works, already touched on or described in Chapter 2, are now mentioned again in support. In [176], Varma and Zisserman argued that a large variety of signals (e.g. textures) can be analysed by just looking at small neighbourhoods. They used 7×7 patches to generate a texton based representation and achieved

better performance than the filtering based methods they compared against when classifying material images from the Columbia-Utrecht database. In [122], Ojala *et al.* also advocated the use of local neighbourhood processing in the shape of local binary patterns as texture descriptors. These generate a binary code that characterises the local texture pattern by thresholding a neighbourhood by the graylevel value of its centre. Other works based on local pixel neighbourhoods are those which apply MRF models. For example in [34], Cohen *et al.* [34] partitioned testing images into non-overlapping sub-blocks. Each image block was inspected based on statistics derived from defect-free samples using Gaussian Markov Random Fields (GMRF). In [123], the authors showed that MRF based methods were competitive in a comparative study against other statistical and spectral based methods in defect detection. Jojic *et al.* [70] defined the epitome as a miniature, condensed version of an image containing the constitutive elements of its shape and textural properties needed to reconstruct the image. The epitome also relies on “raw” pixel values to characterise textural and colour properties rather than popular filtering responses. An image is defined by its epitome and a smooth, *hidden* mapping from the epitome to its image pixels.

Local contextual analysis has also been applied in visual inspection. Chetverikov and Hanbury [30] argued that regularity (periodicity) and anisotropy (local orientation) were two of the fundamental structural properties of the visual world. The authors examined the periodicity of the autocorrelation function in polar representation of a texture image as a measurement of pattern regularity, and computed the dominant local orientation based on the image gradient. They applied these two measurements to detecting local structural defects in textures with strong displacement rules, such as textiles and webs. In [79], Kittler *et al.* used blob analysis in image stacks, generated from colour clustering, and morphological processing to identify defective regions. Blobs from training samples were statistically measured as local structural properties. Regions in testing samples with measurements deviating from known distributions were then classified as defective. Recently, Kumar [81] directly used local pixel neighbourhoods to classify textile defects.

Inspired by the success of non-filtering local neighbourhood approaches for both general texture analysis and defect detection, in this chapter new approaches to detecting and localising defects in random textured surfaces are presented. The methods proposed here support the assertion made by the above works on the reliability of local contextual representation for texture analysis. The emphasis to defect detection is shifted in particular. Firstly, in Section 5.2, a novel semi-parametric *texem* model is introduced as a two layer generative model for graylevel image representation. This shown to be a suitable tool in texture defect inspection within a novelty detection framework which eliminates the need for lengthy training stages particularly when the range of defects are unknown. Next, in Section 5.3, the work is extended to detect defects in randomly textured colour surfaces.

5.2 Texture Exemplars for Defect Detection on Random Texture

A new approach to detecting defects in random textures is presented which requires only a very few defect-free samples for unsupervised training. We propose each product image is divided into overlapping patches of various sizes. Then, density mixture models can be applied to reduce groupings of patches to a number of textural exemplars, referred to here as *texems*, characterising the means and covariances of whole sets of image patches. Through a process of Expectation Maximisation (EM), the maximum likelihood estimate of the mixture density parameters can be found. The *texems* are generated using multiscale analysis to reduce the computational costs. Finally, novelty detection is performed by applying the lower bound of normal samples data likelihoods on the multiscale defect map of an image under inspection to localise defects.

In Sections 5.2.1 to 5.2.4, the proposed method is presented, including the learning of *texems*, the multiscale approach, and the novelty detection stage. Section 5.2.5 presents a brief overview of the *texem* model and its relationship to other texture analysis techniques, namely the *texton* and *epitome* models. Experimental results are given in Section 5.2.6. Section 5.2.7 concludes this section and points to the further development of this method and its extension to deal with colour images.

5.2.1 The *texem* model

In an application such as ceramic tile production, the images under inspection may appear different from one surface to another due to the random texture patterns involved. However, the visual impression of the same product line remains consistent. In other words, there exist textural primitives that impose consistency within the product line. Figure 5.1 shows several example tile images from the same class (or production run) decorated with a marble-like texture. Each tile has different features on its surface, but they all still exhibit a consistent visual impression. One may collect enough samples to cover the range of variations and this approach has been widely used in texture classification and defect detection, e.g. for textile defects in [81]. It usually requires a large number of non-defective samples and lengthy training stages. This is not necessarily practical in a factory environment.

Instead of the traditional supervised classification approach, we learn, in an unsupervised fashion, textural primitive information from a very small number of training samples. These representations are named *texture exemplars* or *texems*. They occur at various sizes and encapsulate the texture or visual primitives of a given image. Similar to the work in [70], each surface texture is considered to be produced by putting together a certain number of sub-image patches of various

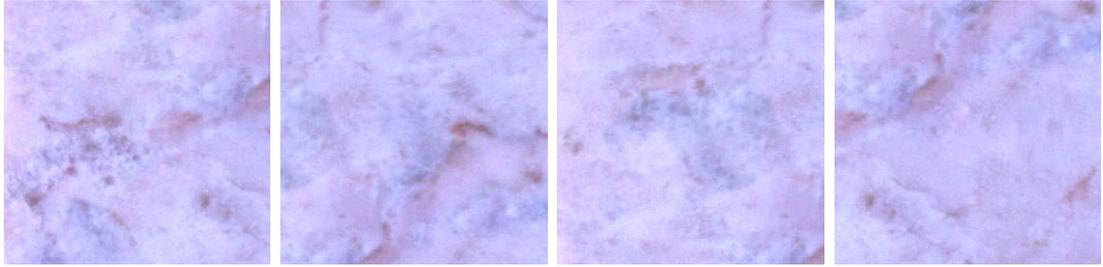


Figure 5.1: Example marble tiles from the same family whose patterns are different but visually consistent.

sizes, possibly overlapped. As the images of the same product contain the same textural elements, one product image can be generated from the patches extracted from other images. Thus, for a few given samples a large number of patches of various sizes can be obtained (which can in turn generate a large set of new images with the same visual impression). However, it is computationally prohibitive to perform defect detection based on such a large number of patches. Also, the patches themselves contain lots of redundant information. The number of patches can be reduced by learning a relatively small number of primitive representatives, i.e. texture exemplars or texems.

Next, a two-layer generative model is presented, in which an image at the first layer is assumed to be generated by superposition of a small number of image patches of various sizes from the second layer with added noise at each pixel position. These various size image patches along with their variances are called texems. The generation process can be naturally modelled by mixture models with a bottom-up procedure.

5.2.2 Learning texems

In this section, the process of extracting texems from a single sample image is detailed with each texem containing some of the overall textural primitive information. For graylevel images, a Gaussian mixture model is used to obtain the texems in a simple and efficient manner. In brief, a defect-free image is broken down into overlapping patches of various sizes and group similar sized patches into a multidimensional space, dependent on the patch size, and describe the clusters found using a Gaussian mixture model. The representative texture exemplars are then learnt through an EM algorithm applied on the mixture density parameters. Then, as we are interested in localising the defective regions, a small patch at each pixel position of an unseen image is extracted and classified using the set of texems obtained at the training stage.

Each texem, denoted as \mathbf{m} , is defined by a mean, $\boldsymbol{\mu}$, and a corresponding covariance matrix, $\boldsymbol{\omega}$, i.e. $\mathbf{m} = \{\boldsymbol{\mu}, \boldsymbol{\omega}\}$. The original image \mathbf{I} is broken down into a set of P patches $\mathbf{Z} = \{\mathbf{Z}_i\}_{i=1}^P$, each

containing pixels from a subset of image coordinates. The shape of the patches can be arbitrary, but in this study square patches of size $d = N \times N$ is used. The patches may overlap and can be of various sizes, e.g. as small as 5×5 to as large as required (however, for larger window size one should ensure there are enough samples to populate the feature space). We assume that there exist K texems, $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^K$, $K \ll P$, for image \mathbf{I} such that each patch in \mathbf{Z} can be generated from a texem. In other words, the original image \mathbf{I} can be reconstructed by the texems with a certain reconstruction error.

To learn these texems the P patches are projected into a set of high dimensionality spaces. The number of these spaces is determined by the number of different patch sizes and their dimensions are defined by the corresponding value of d . Each pixel position in a patch contributes one coordinate of a space. Each point in a space corresponds to a patch in \mathbf{Z} . Then each texem represents a class of patches in the corresponding space. Each class is considered as a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\omega}_k$, which corresponds to \mathbf{m}_k in the spatial domain. Thus, the probability for patch \mathbf{Z}_i given that it belongs to the k th texem \mathbf{m}_k , is:

$$p(\mathbf{Z}_i|\mathbf{m}_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{Z}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k)\right\}. \quad (5.1)$$

Since all the texems \mathbf{m}_k are unknown, we need to compute the density function of \mathbf{Z}_i given the parameter set θ by applying the definition of conditional probability and summing over k ,

$$p(\mathbf{Z}_i|\theta) = \sum_{k=1}^K p(\mathbf{Z}_i|\mathbf{m}_k) \alpha_k, \quad (5.2)$$

where $\theta = \{\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\omega}_k\}_{k=1}^K$ is the graylevel texem parameter set containing α_k , which is the *prior* probability of the k th texem constrained by $\sum_{k=1}^K \alpha_k = 1$, the mean $\boldsymbol{\mu}_k$, and the covariance $\boldsymbol{\omega}_k$. Then, deriving the parameters through optimising the data log-likelihood expression of the entire set \mathbf{Z} , given by

$$\log p(\mathbf{Z}|K, \theta) = \sum_{i=1}^P \log p(\mathbf{Z}_i|\theta) = \sum_{i=1}^P \log \left(\sum_{k=1}^K p(\mathbf{Z}_i|\mathbf{m}_k) \alpha_k \right). \quad (5.3)$$

Hence, the objective is to estimate the parameter set θ for a given number of texems. The EM [45] technique can be used to find the maximum likelihood estimate of the mixture density parameters from the given dataset \mathbf{Z} for all K texems. That is to find $\hat{\theta}$ where

$$\hat{\theta} = \arg \max_{\theta} \log(\mathcal{L}(\theta|\mathbf{Z})) = \arg \max_{\theta} \log p(\mathbf{Z}|K, \theta). \quad (5.4)$$

EM can be derived in many different ways, such as the missing data approach or lower bound maximisation. We derive the algorithm by treating the given data as incomplete, closely following [10]. A more detailed description of the EM algorithm can be found in Appendix A (this also serves as the basis for the proposed mixture model later in Chapter 6). The two steps of the EM stage are as follows. The E-step involves a soft-assignment of each patch \mathbf{Z}_i to texems, \mathcal{M} , with an initial guess of the true parameters, θ . This initialisation can be set randomly, although K-means is used to compute a simple estimate with K set as the number of texems to be learnt. The value of K can be determined empirically, or optimised based on a certain objective function which is discussed later in Chapter 6, Section 6.4. The intermediate parameters are denoted as $\theta^{(t)}$, where t denotes the iteration step. The probability that patch \mathbf{Z}_i belongs to the k th texem may then be computed using Bayes rule:

$$p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) = \frac{p(\mathbf{Z}_i | \mathbf{m}_k, \theta^{(t)}) \alpha_k}{\sum_{k=1}^K p(\mathbf{Z}_i | \mathbf{m}_k, \theta^{(t)}) \alpha_k}. \quad (5.5)$$

The M-step then updates the parameters by maximising the log-likelihood function as given in (5.4). The new estimates are denoted by $\hat{\alpha}_k$, $\hat{\mu}_k$, and $\hat{\omega}_k$:

$$\begin{aligned} \hat{\alpha}_k &= \frac{1}{P} \sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}), \\ \hat{\mu}_k &= \frac{\sum_{i=1}^P \mathbf{Z}_i p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}, \\ \hat{\omega}_k &= \frac{\sum_{i=1}^P (\mathbf{Z}_i - \hat{\mu}_k)(\mathbf{Z}_i - \hat{\mu}_k)^T p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}. \end{aligned} \quad (5.6)$$

The E-step and M-step are iterated until the estimations are stabilised or the rate of improvement of the likelihood falls below a pre-specified convergence threshold. Then, the texems can be easily obtained by projecting the learnt means and covariance matrices back to the patch representation space. Various sizes of texems can be used and they can overlap to ensure they capture sufficient textural characteristics.

Figure 5.2 illustrates twelve 5×5 texem means extracted from one of the images shown in Fig 5.1. They are arranged, from top-left to bottom-right, according to their decreasing order of *priors* α_k .

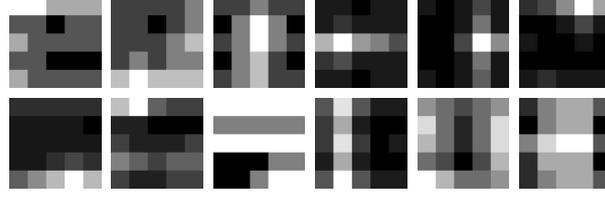


Figure 5.2: Twelve 5×5 texem means extracted from the top-left image in Figure 5.1. (Images are enhanced for better visualisation.)

5.2.3 A simple multiscale approach

In order to capture sufficient textural properties, texems can be from as small as 3×3 to larger sizes such as 20×20 . However, the dimension of the space patches \mathbf{Z} are transformed into will increase dramatically as the dimension of the patch size d increases. This means that a very large number of samples and high computational costs are needed in order to accurately estimate the probability density functions in very high dimensional spaces, [152], forcing the procurement of a large number of training samples.

Instead of generating variable-size texems, fixed size texems are learnt in multiscale. This will result in (multiscale) texems with a very small size, e.g. 5×5 . Besides computational efficiency, exploiting information at multiscale offers other advantages over single-scale approaches. Characterising a pixel based on local neighbourhood pixels can be more effectively achieved by examining various neighbourhood relationships. The corresponding neighbourhood at coarser scale obviously offers larger spatial interactions. Also, processing at multiscale excludes the selection of one optimal resolution, which is often data dependent. A simple multiscale approach by using a Gaussian pyramid is sufficient.

Let us denote $\mathbf{I}^{(n)}$ as the n th level image of the pyramid, $\mathbf{Z}^{(n)}$ as all the image patches extracted from $\mathbf{I}^{(n)}$, l as the total number of levels, and S^\downarrow as the down-sampling operator. Then,

$$\mathbf{I}^{(n+1)} = S^\downarrow G_\sigma(\mathbf{I}^{(n)}), \quad \forall n, n = 1, 2, \dots, l-1, \quad (5.7)$$

where G_σ denotes the Gaussian convolution. The finest scale layer is the original image, $\mathbf{I}^{(1)} = \mathbf{I}$. We then extract multiscale texems from the image pyramid using the method presented in the previous section except \mathbf{m} has a fixed size. Similarly, let $\mathbf{m}^{(n)}$ denote the n th level multiscale texems and $\theta^{(n)}$ the parameters associated at the same level, which will then be used for novelty detection at the corresponding level of the pyramid.

During the EM process, the stabilised estimation of a coarser level is used as the initial estimation

for the finer level, i.e.

$$\hat{\theta}^{(n,t=0)} = \theta^{(n+1)}. \quad (5.8)$$

This helps speed up the convergence and achieve a more accurate estimation. Then, the probability of a patch $\mathbf{Z}_i^{(n)}$ belonging to texems in the corresponding n th scale is

$$p(\mathbf{Z}_i^{(n)}|\theta^{(n)}) = \sum_{k=1}^{K^{(n)}} p(\mathbf{Z}_i^{(n)}|\mathbf{m}_k^{(n)})\alpha_k^{(n)}, \quad (5.9)$$

where $\theta^{(n)}$ represents the parameter set for the level n , $\mathbf{m}_k^{(n)}$ is the k th texem at the n th pyramid level, and $p(\mathbf{Z}_i^{(n)}|\mathbf{m}_k^{(n)})$ is a multivariate Gaussian distribution function as shown in (5.1) with parameters associated to texem set, \mathcal{M} .

5.2.4 Novelty detection

Once the texems are obtained from a single training image, the minimum bounds of normal samples in each resolution level can be worked out automatically in order to perform novelty detection. A small set of defect-free samples (e.g. 4 or 5 only) are arranged within a multiscale framework, and patches with the same texem size are extracted. The minimum probability of a patch $\mathbf{Z}_i^{(n)}$ at level n across the training images is treated as the lower bound of the data likelihood, denoted as $\Lambda^{(n)}$:

$$\Lambda^{(n)} = \min(p(\mathbf{Z}_i^{(n)}|\theta^{(n)})), \quad \forall \mathbf{Z}_i^{(n)} \in \mathbf{Z}^{(n)}, \quad (5.10)$$

where $p(\mathbf{Z}_i^{(n)}|\theta^{(n)})$ is given by (5.9). Alternatively, a statistical parametric approach can be used assuming that the likelihood value distribution is close to Gaussian or using clustering method to separate the distribution into individual near-normal distributions, as opted for later in Chapter 6. This completes the training stage in which with only a very few non-defective images, one can determine the texems and an automatic threshold for marking new image patches as “defective” or “defect-free”.

In the testing stage, the image under inspection is again layered into a multiscale framework and patches at each pixel position (x, y) at each level n are examined against the learnt texems. The probability for each patch is then calculated, $p(\mathbf{Z}_i^{(n)}|\theta^{(n)})$, and compared to the minimum data likelihood, $\Lambda^{(n)}$, at the corresponding level. Let $Q^{(n)}(x, y)$ be the probability map at the n th

resolution level. Then, the potential defect map, $\mathcal{D}^{(n)}(x, y)$, at level n is:

$$\mathcal{D}^{(n)}(x, y) = \begin{cases} 0 & \text{if } Q^{(n)}(x, y) \geq \Lambda^{(n)} \\ \Lambda^{(n)} - Q^{(n)}(x, y) & \text{otherwise.} \end{cases} \quad (5.11)$$

The $\mathcal{D}^{(n)}(x, y)$ indicates the probability of there being a defect. The information coming from all the resolution levels then need to be combined to build the certainty of the defect at position (x, y) . The method described in [49] is followed here which combines information from different levels of a multiscale pyramid and reduces false alarms. It assumes that a defect must appear in at least two adjacent resolution levels for it to be certified as such. Using a logical AND, implemented through the geometric mean, of every pair of adjacent levels, we initially obtain a set of combined maps as:

$$\mathcal{D}^{(n,n+1)}(x, y) = [\mathcal{D}^{(n)}(x, y)\mathcal{D}^{(n+1)}(x, y)]^{1/2}. \quad (5.12)$$

Please note that each $\mathcal{D}^{(n+1)}(x, y)$ is scaled up to be the same size as $\mathcal{D}^{(n)}(x, y)$. This operation reduces false alarms and yet preserves most of the defective areas. Next, the resulting $\mathcal{D}^{(1,2)}(x, y)$, $\mathcal{D}^{(2,3)}(x, y)$, ..., $\mathcal{D}^{(l-1,l)}(x, y)$ are combined in a logical OR, as the arithmetic mean, to provide a final map for the defects detected across all the scales:

$$\mathcal{D}(x, y) = \frac{1}{l-1} \sum_{n=1}^{l-1} \mathcal{D}^{(n,n+1)}(x, y), \quad (5.13)$$

where $\mathcal{D}(x, y)$ contains the joint contribution of all the resolution scales and marks the defects in the test image. Figures 5.4 and 5.5 illustrate this defect fusion process.

5.2.5 Method summary and comments

The texem model is motivated from the observation that in random texture surfaces of the same family, the pattern may appear to be different in textural manifestation from one sample to another, however, the visual impression and homogeneity remains consistent. This suggests that the random pattern can be described with a few textural primitives.

In the texem model, the image is assumed to be a superposition of image patches with various sizes and even various shapes. The variation at each pixel position in the construction of the image is embedded in each texem. Thus, it can be viewed as a two-layer *generative* statistical model. The image \mathbf{I} , as the first layer, is generated from a collection of texems \mathcal{M} as the second layer, i.e. $\mathcal{M} \rightarrow \mathbf{I}$. In deriving the texem representations from an image or a set of images, a bottom-up learning process can be used as presented in this chapter. The images are split into overlapping

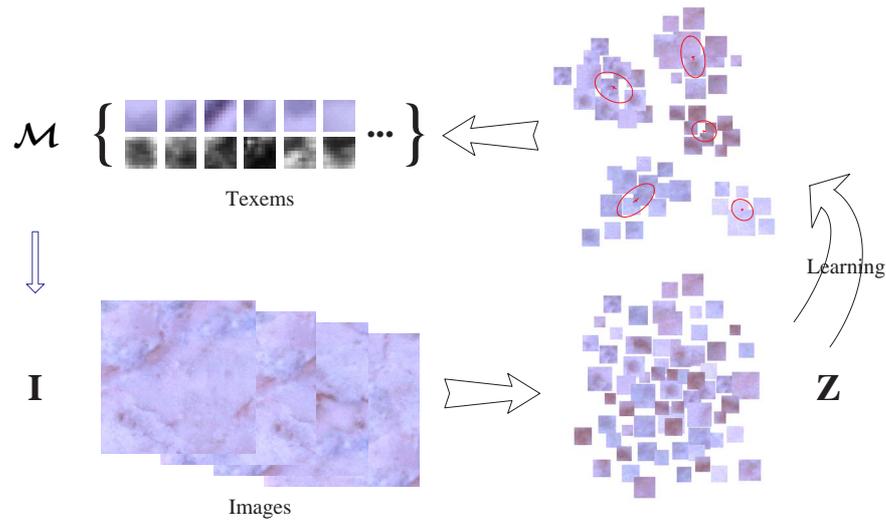


Figure 5.3: An illustration of the two-layer structure of the texem model and its bottom-up learning procedure.

patches \mathbf{Z} with various sizes, followed by a generalisation procedure to obtain the texems. Here, Gaussian mixture modelling is used. However, other alternatives may also be applied. In fact, later in Chapter 6 different mixtures are used for texem inference. Figure 5.3 illustrates the two-layer structure and the bottom-up learning procedure.

Relationship to Textons - As outlined in Chapter 2, Section 2.2.2, three layer *generative* [198] and *discriminative* [87] models have been introduced to describe textons (cf. Figure 2.2 and Figure 5.3). The texem model is similar to the well-known texton model in the sense that both try to characterise textural images by using micro-structures. On the other hand, the texem model is significantly different from the texton model:

- *It relies directly on sub-image patches instead of using base functions.* Selection of the base functions is important in the texton model in order to obtain meaningful textons. However, the design of a bank of base functions is non-trivial and likely to be application dependent. Enormous efforts have gone into explicitly extracting visual primitives (textons), such as blobs. They generally suffer from finding optimum representations, e.g. the window size (for example see [176]). In the proposed model, each texem is an encapsulation of texture primitive(s). Not using base functions also makes the proposed method more convenient to deal with multi-spectral images, e.g. colour images, as will be demonstrated later.
- *It is an implicit, rather than an explicit, representation of primitives.* Texems are implicit representations, which makes them more flexible as they come at different sizes, while textons are explicit representations. For example, if the texem size reduces to a single pixel,

it becomes histogram analysis. If it is the same size as the input images, then the problem turns into image template analysis. In general, each of the texems may contain multiple textural primitives which describes a group of image patches. This implicit representation at various sizes avoids the difficulties of explicitly finding the best primitive representation.

In [176], Varma and Zisserman directly used image patches to derive textons by performing K-means clustering in the feature space based on patch vectorisation (a brief discussion of this was given earlier in Section 5.1). The proposed method in this chapter is similar to this work as we too use “raw” pixel values and patch vectorisation. However, the texem model is a two layer generative model and the use of the Gaussian mixture modelling makes it much more convenient to quantitatively measure the data similarity. It is also worth noting the importance of using multiscale or various size of patches in learning textural primitives, besides its simplicity. In [176], the authors investigated the effect of different patch sizes and found that a large neighbourhood size generally achieved better results. In our proposed method, we advocate multiscale analysis to alleviate the difficulties associated with scale selection.

Relationship to the Epitome - The epitome can be used to generalise multiple images while preserving textural details against over-smoothing [70]. The large number of patches extracted from tile images can be condensed into an epitomic representation for each family of tiles. However, as the mappings between the epitome and the training images are hidden and the mappings from the testing images can not be possibly obtained as *a priori* knowledge, we would need to consider all possible mappings from the epitome at both the training and testing stages. This involves the examination of different patch sizes at each pixel location in the epitome. Although the epitome is much smaller than the original image, it is still much larger than the patches themselves. Hence, it will be computationally very expensive to perform defect detection directly based on the epitome. Thus, rather than forcing the textural properties to be condensed into a single epitome, we learn multiple epitomic-like representations (i.e. texems) that generalise the family of texture images without using any implicit mappings. The large number of patches with various sizes can then be described using a very small number of representatives with explicit pixel position correspondence. Additionally, the spatial relationships are preserved in a limited spatial resolution, e.g. in Figure 2.5 the red and blue region in the original image are connected to each other, while in the epitomic versions (especially in the 16×16 epitome) they are separated from each other.

Relationship to fractals - The fractal model is based on the observation of self-similarity present in natural objects. As reviewed in Chapter 2, fractal dimension, as a measure of complexity or irregularity, and lacunarity, as a measure of structural variation or inhomogeneity, are the two most important features for texture analysis. Texems share with fractals the concept that textures can be characterised by constitutive elements. However, the texem model does not uphold the assumption of self-similarity.

5.2.6 Experimental results

The proposed method was applied to the GRAYSET dataset which contained a variety of tile families with different types of defects including physical damage, pin holes, textural imperfections, multi-print pattern mis-registrations, and many more. Details of the tile dataset are given in Chapter 3. A small number of defect-free samples (five to ten depending on the size of the dataset) were used for training, at which stage the texems and the lower bound data likelihoods $\Lambda^{(n)}$ at each resolution scale were learnt. The number of texems at each resolution level were empirically set to 12, and the size of each texem was set to 5×5 pixels. The number of multiscale levels was $l = 5$. These parameters were fixed throughout the experiments.

The first two examples given here are to demonstrate the multiscale detection process and the defect fusion scheme. Figure 5.4 shows a random texture example with a defect in the lower right region introduced by a printing problem. The potentially defective regions detected at each resolution level n , $n = 1, 2, \dots, 5$, are marked on the corresponding images in Figure 5.4. It can be seen that the texems show good sensitivity to the defective region at different scales. As the resolution progresses from coarse to fine, additional evidence for the defective region is gathered. This evidence is then combined, shown in Figure 5.4 (second right-bottom row), to produce the defect map \mathcal{D} , i.e. (5.13). The darker the colour, the more certain the pixel is defective. The final image shows the defects superimposed on the original image. As mentioned earlier, the defect fusion process can eliminate false alarms, e.g. see the extraneous false defect regions in level $n = 5$ which disappear after the operations in (5.12) and (5.13). Figure 5.5 illustrates the same stages for another example. Note in these examples the evidence for the smaller defects only begins to appear from level $n = 3$ with finer resolutions. The final defects all correspond to true defects.

Figure 5.6 shows three more random texture examples, from the same family of tiles as in Figure 5.4. The defective regions only occupy a very small percentage of the entire surface but are still localised well by the texems. The first example shows a surface that has a physical defect. The defect in the second image is a small bump which is hardly discernible. The last image contains a missing print defect with a round shape.

More examples of different random textures from two tile families are shown in Figures 5.7 and 5.8. In each family of patterns, the textures are varying but of the same visual impression. The images in Figure 5.7 are from a set of tiles with coarse textures. The proposed method could find from very small surface defects to large variable shaped defects such as the thin, long defect running along the whole edge of the tile (due to a shifted print head) as shown in the last image in the first row of Figure 5.7. The last two examples have relatively higher contrast and this made the defective regions easier to detect at coarser levels. As detected defect regions usually appear

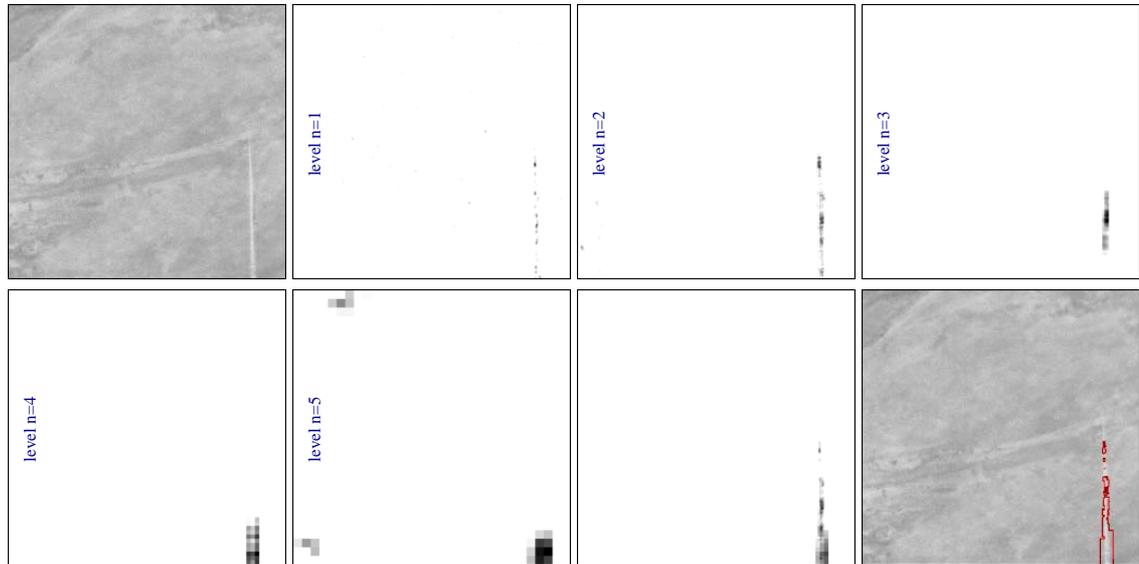


Figure 5.4: Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 5$, the joint contribution of all resolution levels, and the final defective regions superimposed on the original image.

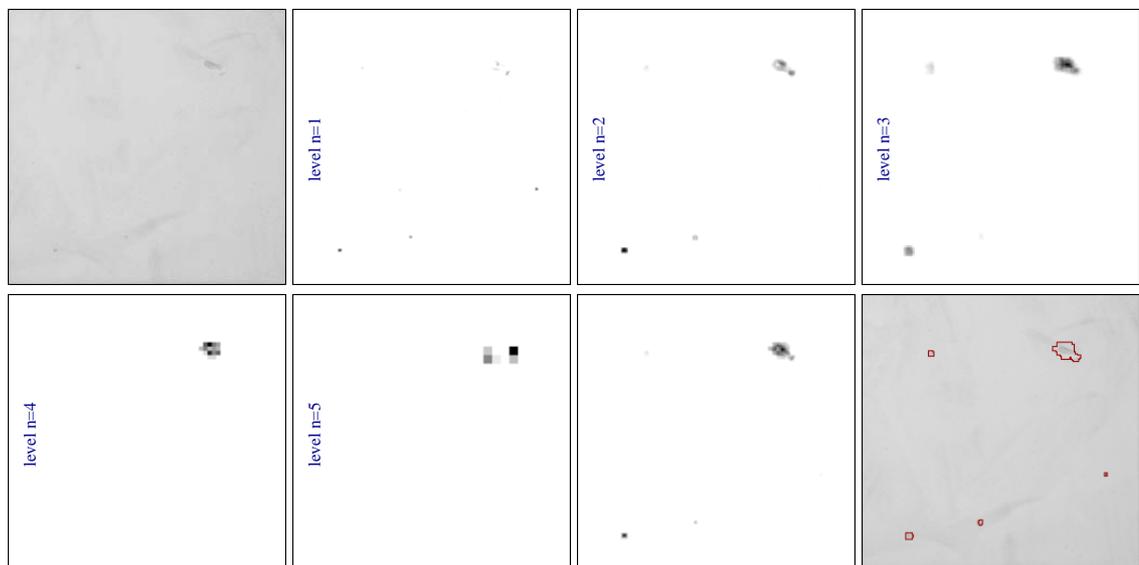


Figure 5.5: Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 5$, the joint contribution of all resolution levels, and the final defective regions superimposed on the original image.

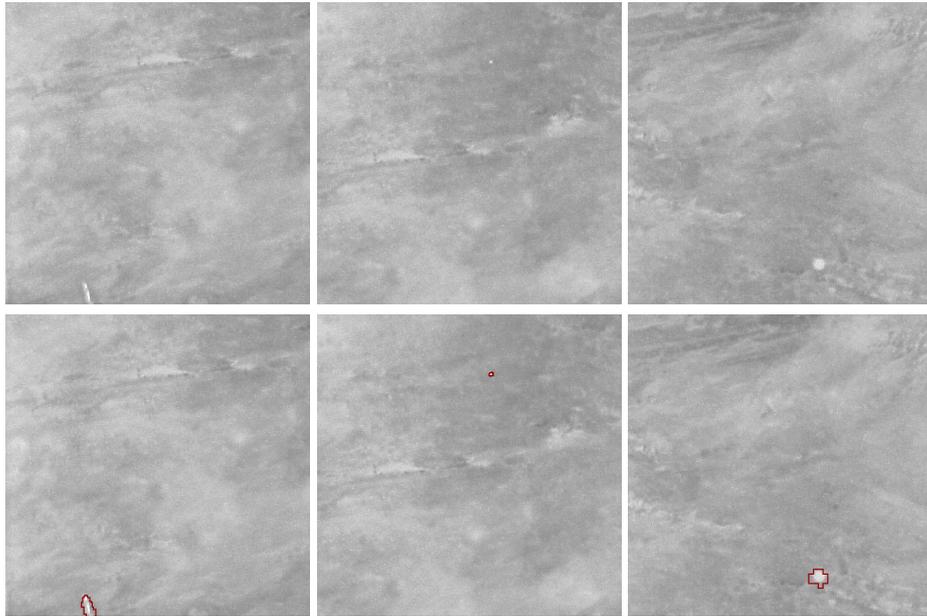


Figure 5.6: Defect detection - first row: original images, second row: superimposed defective regions on original images, from left - surface defect, small bump, and missing print.

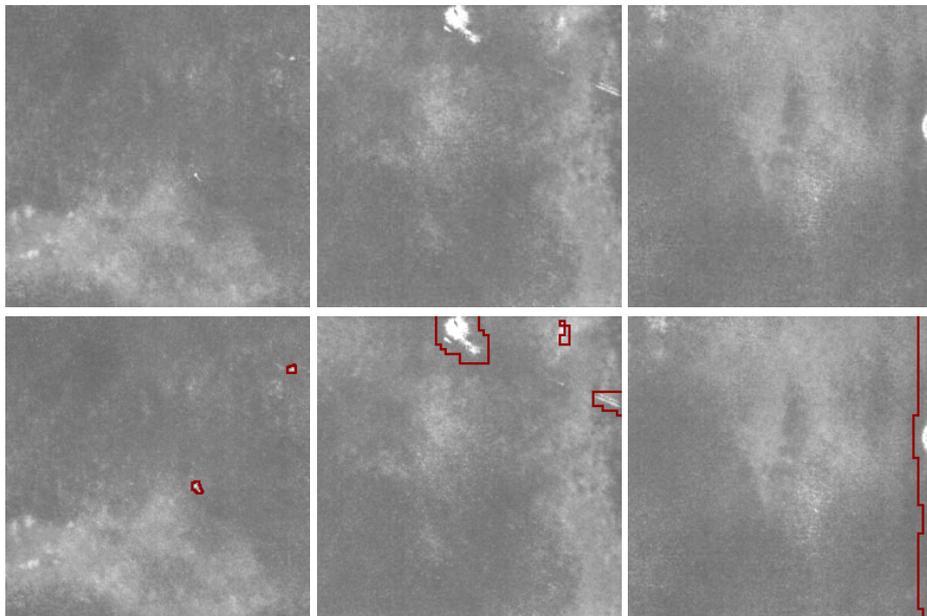


Figure 5.7: Defect detection - first row: original defective images, second row: superimposed defective regions on original images, from left - small material bumps, misprints, and misprint along the whole right edge.

larger than they actually are at coarser levels (e.g. see the 5th level in Figure 5.4), then after fusing these regions from all the resolution levels, the final localised regions appear slightly larger than

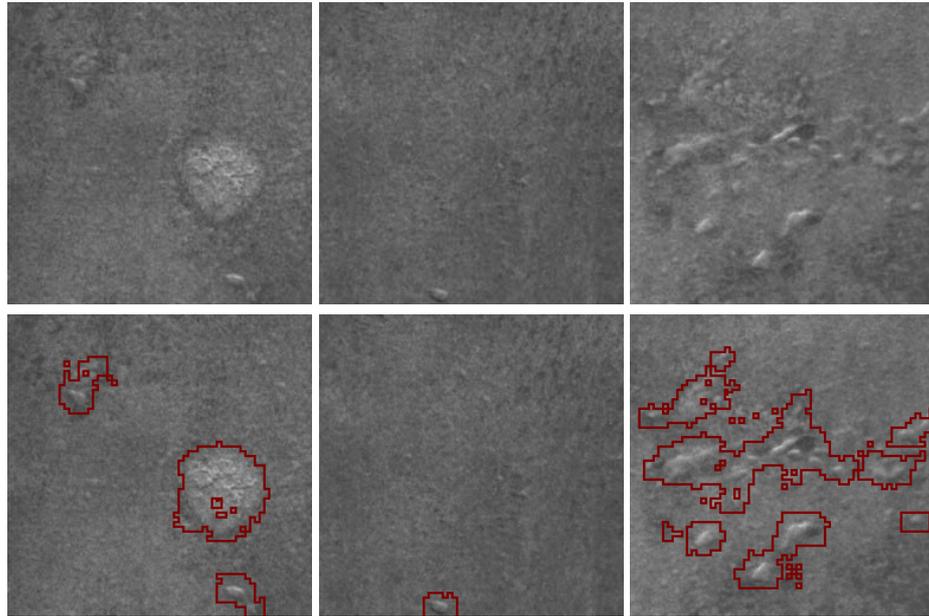


Figure 5.8: Defect detection - first row: original defective images, second row: superimposed defective regions on original images, from left - printing defect, single surface bump, and a cluster of bumps.

the actual defective regions. The images in Figure 5.8 are from a set of tiles with much finer sand-like textures. The first image from this set shows printing defects. A single small surface bump appears in the second image, while the last one contains a cluster of bumps.

Figure 5.9 shows more examples from all kinds of different tile types. There are cloud-like loose textures, marble textures with macro-structures, granite-like fine texture with large contrast, and even periodic pattern with horizontal lines. There are also a variety of defects, such as missing print, pin holes, broken corner, cracks, and surface undulation. All the defects were successfully detected.

All of the examples given so far have had random appearances. The proposed method can also detect abnormalities in regular patterns. For example, the first image in the first row of Figure 5.10 shows three incompletely printed dots at the top-left corner of the regular pattern. Each dot is composed by one larger, lighter dot as background and one smaller, darker dot positioned in the centre. This can be seen more closely by examining a zoomed-in instance of some good dots and the three bad dots in the second column of Figure 5.10. In another regular pattern example, the tiles carried a vertical grid-like pattern. Incorrect print and smudge defects damaging the local pattern regularity were correctly detected as illustrated in the third and last columns in Figure 5.10 where both the original and superimposed defects are shown.

The examples shown illustrate the ability of texems in detecting and localising small or large

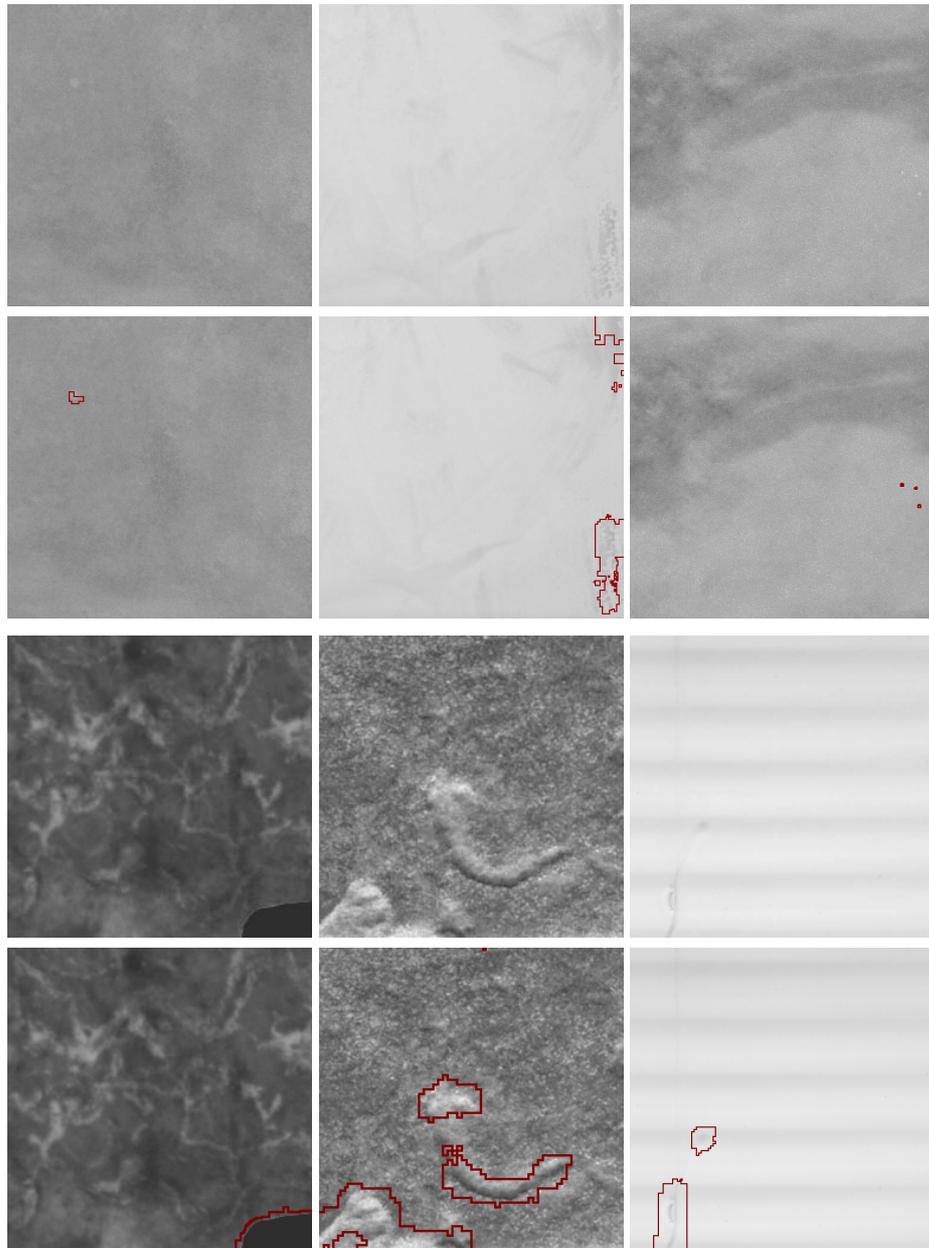


Figure 5.9: Defect detection - first row: original images from different types of tiles, second row: superimposed defective regions on original images, from left - small missing print, misprint in the top-right and bottom-right corners, and three pin holes; third row: another three example images also from different tile families, last row: superimposed defective regions on original images, from left - broken corner, structural defects with complex shapes, and a thin crack and a small surface undulation.

defects on highly textured surfaces. As patches are extracted from each pixel position at each resolution level, a typical *training stage* involves examining over 0.25 million patches (for a 512×512 image) to learn the texemes in multiscale. This takes around 7 minutes on a 2.8GHz Pentium

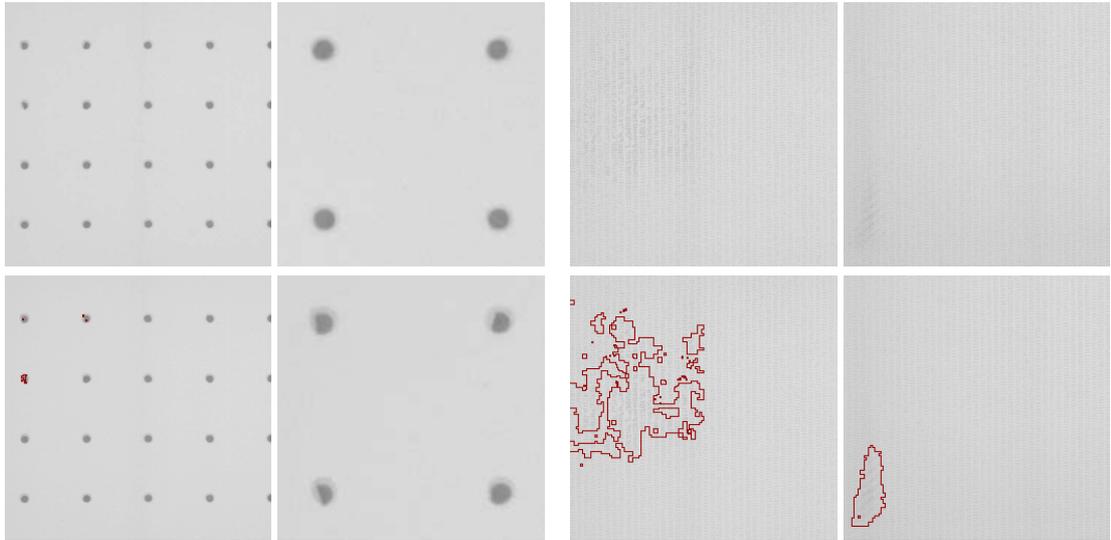


Figure 5.10: Detecting defects in regular patterns - first column: original defective image and superimposed defective regions; second column: closeup views of normal and abnormal dot patterns; third and fourth columns: two defective samples of a different regular texture with pattern irregularities and superimposed defective areas.

Table 5.1: GRAYSET results using graylevel texems (values are %s)

No.	Tile Type	specificity	sensitivity	accuracy
1	PRAN1A	100	97.3	98.7
2	WRIB1A	90.5	99.2	94.8
3	TF1X1A	91.9	96.1	94.0
4	BM1A	87.5	100	93.8
5	LRSIDE	95.6	100	98.3
6	TSPI1A	66.7	99.4	83.0
7	CSA1	92.9	100	96.4
8	SALZ1A	90.6	75.0	82.8
Overall		89.5	95.9	92.7

4 Processor running Linux with 1GB RAM to obtain the texems and to determine the thresholds for novelty detection. The testing stage is much faster, requiring about 20 seconds to inspect one tile image. The computation time of the proposed method can be greatly reduced by examining every other pixel (or less) instead of each pixel position, both at the training and testing stages. For example, examining one in four pixels (i.e. one pixel in a 2×2 neighbourhood) at the testing stage will result in about 5 seconds to inspect one tile, which shows its potential for real-time applications.

Since the defects in the dataset are difficult to manually localise, there is no groundtruth in terms of

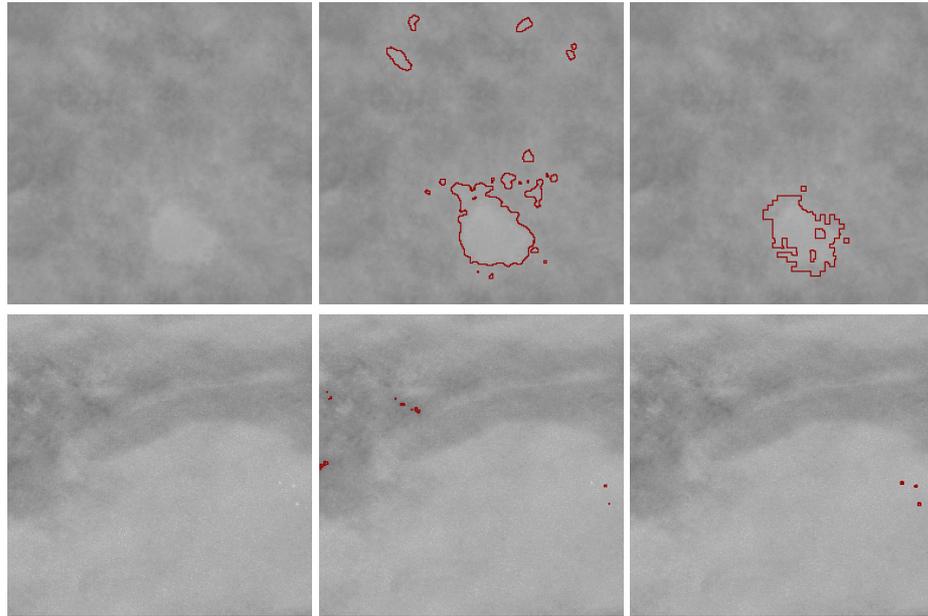


Figure 5.11: Performance comparison - first column: two samples with large and subtle defects; second column: novelty detection results using epitomes - which produced many false positives or failed to locate the true defects; third column: novelty detection results using texems which successfully detected all the true defects.

the spatial accuracy. However, the images were labelled as “defective” or “defect-free” by experts. Thus, the results are presented by classification rate.

The proposed method was evaluated across 1512 tiles from eight different families of textures. Very good results were obtained with sensitivity at 95.9%, specificity at 89.5%, and overall accuracy at 92.7%. The classification results are shown in Table 5.1.

Next, we compared against the epitome model in particular as the proposed method revolves around similar concepts (Section 5.2.5 gives a detailed comparison). The epitome was applied for texture segmentation (with software provided by the authors of [70]), however, we extended it into a similar framework as the proposed method for better comparative analysis. Hence, after generating the multiscale version of an image, an appearance epitome was learnt using 5×5 image patches at each scale, resulting in epitomes varying from 7×7 to 24×24 . Using these multiscale epitomes, novelty detection was performed as before (see Section 5.2.4) in the testing stage. This involved finding a match in the epitome for an image patch under inspection. As the epitome was still larger than the patch itself and there were numerous comparisons across the image, the detection procedure was computationally very expensive. In fact it costs the epitome based method several hours to perform training or testing. The proposed texems approach is many times faster.

Two results are shown in Figure 5.11. The examples show that the epitome was more prone to

false alarms and less specific in accuracy of localisation. Later in Chapter 6 the texem model is compared against a Gabor filter bank based defect detection method.

5.2.7 Summary

A novel statistical generative model for image representation was introduced. An image is assumed to be generated from the superposition of primitive image representations, texems. The texem model directly uses the “raw” pixel values, instead of popular filtering responses. A bottom-up process with Gaussian mixture modelling was presented to derive the texem representations from a single or a set of images.

The texem based local contextual analysis was used to perform automatic defect detection and localisation for random textures on graylevel images. The proposed method only trained on a very small number of defect-free samples, from which primitive textural information and variation range were obtained. Multiscale analysis was used to save computational cost and to capture sufficient textural information. Defect detection and localisation were performed in a novelty detection framework. Logical processes were used and proved to eliminate false alarms yet preserve defective regions. Testing results on eight different types of tiles with various texture characteristics, totalling 1512 surfaces, showed good performance for the proposed texem based approach.

The texems presented here may only be applicable to graylevel images, as they involve patch vectorisation. However, they can be extended to colour analysis, for example performing texem analysis in individual channels or modifying the inference procedure to derive the texems.

5.3 Defect Detection using Graylevel Texems in Colour Image Eigenchannels

Learning texems from multispectral images implies more complex higher dimensional models. However, if we are willing to sacrifice some information we can instead approximately represent the data with a set of 2D models which are more simple and have fewer parameters to be determined. For example, texems in each channel of an RGB image could be learnt separately and use the entire set of texems for defect detection. Separate channel processing has been widely practised in the literature (Chapter 2 gives a detailed review of colour texture analysis techniques). However, in order to retain as much information as possible after factorising the image data, the PCA is used to obtain three *decorrelated channels*. All the resulting channels must be used to ensure any possible defective regions are captured, but instead of performing PCA on each training

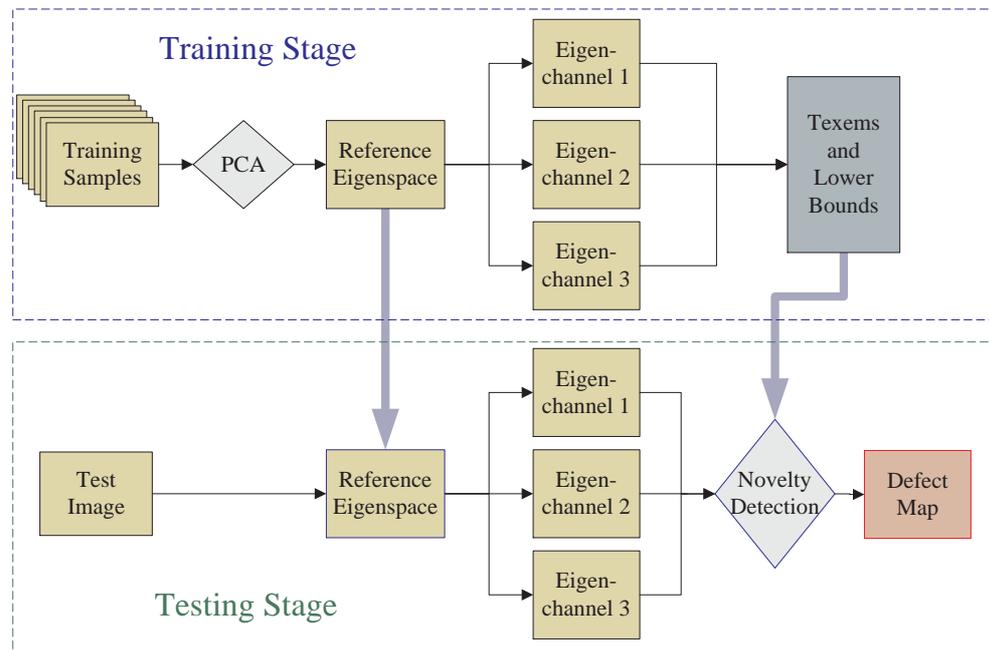


Figure 5.12: Flow chart of the proposed method.

image independently, a single reference texture eigenspace is generated. Details will be given in Section 5.3.1. In Section 5.2, a Gaussian Mixture Model (GMM) is devised to learn texems on graylevel images. Here, the same GMM model is used, which will be briefly reviewed in Section 5.3.2 to learn the texems in each individual PCA transformed channel of a single representative defect-free texture. Later in Section 5.3.3 shows how to combine the information across channels in defect detection using a similar logical process to that described in Section 5.2.4.

The proposed method may be summarised as follows. A reference eigenspace from training samples is derived and the image data is transformed according to this eigenspace into separate channels. Patches in sample images are grouped into clusters, dependent on the patch size, using GMM. The representative texems in each decorrelated channel are learnt through an EM algorithm applied on the model parameters. Then, as we are interested in localising the defective regions, the local neighbourhood of each pixel of the unseen image is examined, after projecting it too into three eigenchannels using the same reference texture eigenspace, and comparing against the set of texems obtained at the training stage. The information from each channel is combined together to produce the final defect map. This process is shown schematically in Fig. 5.12 and will be described in detail next.

5.3.1 Computing the reference eigenspace

To extend texems to colour images, for simplicity, PCA is performed to transform the red, green, and blue channels into three decorrelated channels for better representation. As mentioned already, the patterns on each image within the same texture family can still be different, hence the individually derived principal components can also differ from one image to another. Furthermore, defective regions can affect the principal components resulting in different eigenspace responses from different training samples. Thus, instead of performing PCA on each training image separately, a single eigenspace is generated from several training images; a reference eigenspace in which normal samples are represented (see Fig. 5.12). All the images under inspection will be projected onto this eigenspace. Thus, the transformed channels share the same principal components.

Each colour pixel is denoted as $\mathbf{c}_i = [r_i, g_i, b_i]^T$. Let \mathbf{C} be the matrix containing q three dimensional vectors made up of the pixels from several defect-free samples, e.g. 5 in this work. Let $\bar{\mathbf{c}} = \frac{1}{q} \sum_{\mathbf{c} \in \mathbf{C}} \mathbf{c}$ be the mean vector of \mathbf{C} . Then, PCA is performed on the mean-centred colour feature matrix \mathbf{C} to obtain the eigenvectors $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$, $\mathbf{e}_j \in \mathcal{R}^3$. Singular Value Decomposition again can be used to obtain these principal components. The colour feature space determined by these eigenvectors is referred to as the reference eigenspace $\Psi_{\bar{\mathbf{c}}, \mathbf{E}}$, where the colour features are fully represented. The tile images are then projected onto this reference eigenspace:

$$\mathbf{C}' = \overrightarrow{PCA}(\mathbf{C}, \Psi_{\bar{\mathbf{c}}, \mathbf{E}}) = \mathbf{E}^T (\mathbf{C} - \bar{\mathbf{c}} \mathbf{J}_{1,q}), \quad (5.14)$$

where $\mathbf{J}_{1,q}$ is a $1 \times q$ unit matrix consisting of all 1s. The texems are then learnt in the resulting three eigenchannels.

5.3.2 Generating multiscale texems in image eigenchannels

As before, each texem, \mathbf{m} , is defined by a mean, $\boldsymbol{\mu}$, and a corresponding covariance matrix, $\boldsymbol{\omega}$, i.e. $\mathbf{m} = \{\boldsymbol{\mu}, \boldsymbol{\omega}\}$. Each eigenchannel \mathbf{I}_{e_j} , $j = 1, 2$, or 3 , is broken down into a set of P patches $\mathbf{Z}_{e_j} = \{\mathbf{Z}_{i,e_j}\}_{i=1}^P$, each containing pixels from a subset of image coordinates. Various sizes of patches ($d = N \times N$) can be used. The Gaussian mixture model is then used to learn the K texems, $\mathcal{M}_{e_j} = \{\mathbf{m}_{k,e_j}\}_{k=1}^K$, in each eigenchannel, as described in the Section 5.2.2. Thus, the probability density function for patch \mathbf{Z}_{i,e_j} is given by:

$$p(\mathbf{Z}_{i,e_j} | \mathbf{m}_{k,e_j}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_{k,e_j}|}} \exp\left\{-\frac{1}{2} (\mathbf{Z}_{i,e_j} - \boldsymbol{\mu}_{k,e_j})^T \boldsymbol{\omega}_{k,e_j}^{-1} (\mathbf{Z}_{i,e_j} - \boldsymbol{\mu}_{k,e_j})\right\}. \quad (5.15)$$

The mixture model in eigenchannel e_j is then formulated as:

$$p(\mathbf{Z}_{e_j} | K_{e_j}, \theta_{e_j}) = \sum_{i=1}^P p(\mathbf{Z}_{i,e_j} | \mathbf{m}_{k,e_j}, \theta_{e_j}) \alpha_{k,e_j}, \quad (5.16)$$

where $\theta_{e_j} = \{\alpha_{k,e_j}, \boldsymbol{\mu}_{k,e_j}, \boldsymbol{\omega}_{k,e_j}\}_{k=1}^K$ is the parameter set. The EM algorithm is again applied to derive the parameters resulting in the following iterative estimates:

$$\begin{aligned} \hat{\alpha}_{k,e_j} &= \frac{1}{P} \sum_{i=1}^P p(\mathbf{m}_{k,e_j} | \mathbf{Z}_{i,e_j}, \theta_{e_j}^{(t)}), \\ \hat{\boldsymbol{\mu}}_{k,e_j} &= \frac{\sum_{i=1}^P \mathbf{Z}_{i,e_j} p(\mathbf{m}_{k,e_j} | \mathbf{Z}_{i,e_j}, \theta_{e_j}^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_{k,e_j} | \mathbf{Z}_{i,e_j}, \theta_{e_j}^{(t)})}, \\ \hat{\boldsymbol{\omega}}_{k,e_j} &= \frac{\sum_{i=1}^P (\mathbf{Z}_{i,e_j} - \hat{\boldsymbol{\mu}}_{k,e_j})(\mathbf{Z}_{i,e_j} - \hat{\boldsymbol{\mu}}_{k,e_j})^T p(\mathbf{m}_{k,e_j} | \mathbf{Z}_{i,e_j}, \theta_{e_j}^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_{k,e_j} | \mathbf{Z}_{i,e_j}, \theta_{e_j}^{(t)})}. \end{aligned} \quad (5.17)$$

where

$$p(\mathbf{m}_{k,e_j} | \mathbf{Z}_{i,e_j}, \theta_{e_j}^{(t)}) = \frac{p(\mathbf{Z}_{i,e_j} | \mathbf{m}_{k,e_j}, \theta_{e_j}^{(t)}) \alpha_{k,e_j}}{\sum_{k=1}^K p(\mathbf{Z}_{i,e_j} | \mathbf{m}_{k,e_j}, \theta_{e_j}^{(t)}) \alpha_{k,e_j}}. \quad (5.18)$$

A similar multiscale scheme is again also used in order to capture sufficient textural properties at affordable computational costs.

5.3.3 Novelty detection and defect localisation

A small set of defect-free samples (e.g. 4 or 5 only) are arranged within a multiscale framework as before, and patches with the same texem size in each eigenchannel are extracted. Once the texems are obtained, minimum bounds of normal samples can be found automatically at each resolution level across the eigenchannels. The probability of a patch $\mathbf{Z}_{i,e_j}^{(n)}$ in the eigenchannel e_j belonging to texems of the same channel in the corresponding n th scale is:

$$p(\mathbf{Z}_{i,e_j}^{(n)} | \theta_{e_j}^{(n)}) = \sum_{k=1}^{K^{(n)}} p(\mathbf{Z}_{i,e_j}^{(n)} | \mathbf{m}_{k,e_j}^{(n)}) \alpha_{k,e_j}^{(n)}, \quad (5.19)$$

where $p(\mathbf{Z}_{i,e_j}^{(n)} | \mathbf{m}_{k,e_j}^{(n)})$ is a multivariate Gaussian distribution function as shown in (5.15). The lower bound, $\Lambda_{e_j}^{(n)}$, of the data likelihood is then obtained by examining the distribution of $p(\mathbf{Z}_{i,e_j}^{(n)} | \mathbf{m}_{k,e_j}^{(n)})$ of all patches at level n and channel e_j , $\mathbf{Z}_{e_j}^{(n)}$, across the training images. The same technique described in Section 5.2.4 for graylevel images can be used.

In the testing stage, the image under inspection is again layered into a multiscale framework and patches at each pixel position (x, y) at each level n are examined against the learnt texems. The probability for each patch is then calculated according to (5.19) and compared to the lower bound, $\Lambda_{e_j}^{(n)}$, at the corresponding level and channel. Let $Q_{e_j}^{(n)}(x, y)$ be the probability map at the n th resolution level. Then, the potential defect map, $\mathcal{D}_{e_j}^{(n)}(x, y)$, at level n is:

$$\mathcal{D}_{e_j}^{(n)}(x, y) = \begin{cases} 0 & \text{if } Q_{e_j}^{(n)}(x, y) \geq \Lambda_{e_j}^{(n)} \\ \Lambda_{e_j}^{(n)} - Q_{e_j}^{(n)}(x, y) & \text{otherwise.} \end{cases} \quad (5.20)$$

Again, it is assumed that a defect must appear in at least two adjacent resolution levels for it to be certified as such. Using a logical AND, implemented through the geometric mean, of every pair of adjacent levels, we initially obtain a set of combined pairs as:

$$\mathcal{D}_{e_j}^{(n,n+1)}(x, y) = [\mathcal{D}_{e_j}^{(n)}(x, y)\mathcal{D}_{e_j}^{(n+1)}(x, y)]^{1/2}. \quad (5.21)$$

This operation reduces false alarms and yet preserves most of the defective areas. Then, the defect candidates from each eigenchannel and each pair $\mathcal{D}_{e_j}^{(n,n+1)}(x, y)$ are combined using logical OR, as the arithmetic mean, to provide a final map for the defects detected across all the scales:

$$\mathcal{D}(x, y) = \frac{1}{3(l-1)} \sum_{n=1}^{l-1} \sum_{j=1}^3 \mathcal{D}_{e_j}^{(n,n+1)}(x, y). \quad (5.22)$$

where $\mathcal{D}(x, y)$ contains the joint contribution of all the resolution scales and marks the defects in the test image.

Note, this defect fusion scheme is slightly different from that for graylevel images (cf. (5.22) and (5.13)).

5.3.4 Experimental results

First, the PCA based image decomposition scheme was compared against standard RGB channel separation using graylevel texem analysis. The 28 VisTex texture collage images shown in Chapter 3, were used to examine the spatial localisation accuracy of novelty detection. The texem parameter settings for the eigenchannel based and RGB channel based methods were exactly the same, i.e. twelve 5×5 texems in each scale.

Figure 5.13 and 5.14 show two example collages composed of different numbers of textures. The background images were considered as normal texture, while the foreground regions with various

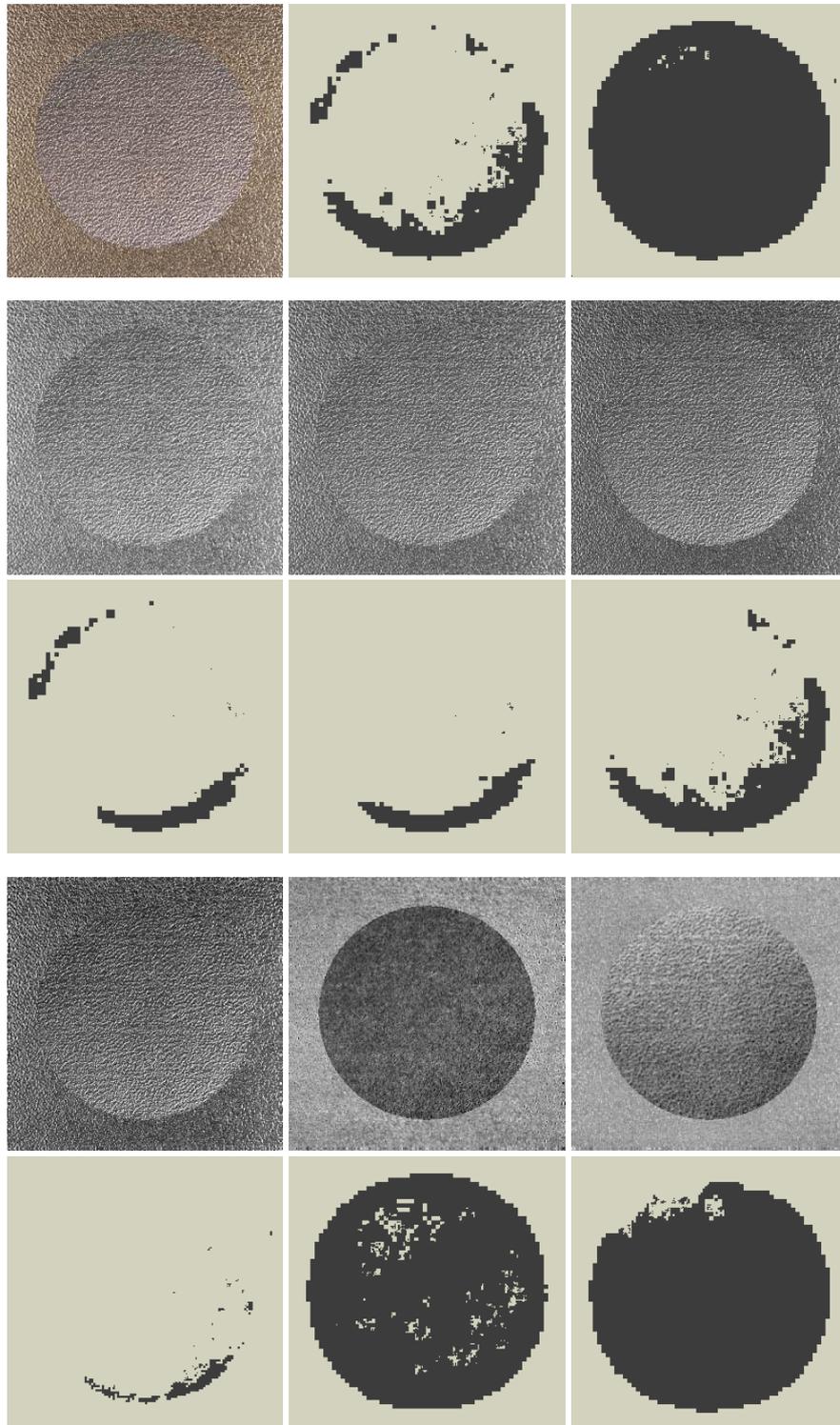


Figure 5.13: Novelty detection comparison on a collage of 2 texture images - first row: Original collage image, novelty detection result in RGB channels, and novelty detection result in image eigenchannels; second row: RGB individual channels; third row: Novelty detection results in each corresponding RGB channel; fourth row: Image eigenchannels of the collage image; last row: Novelty detection results in each individual eigenchannel.

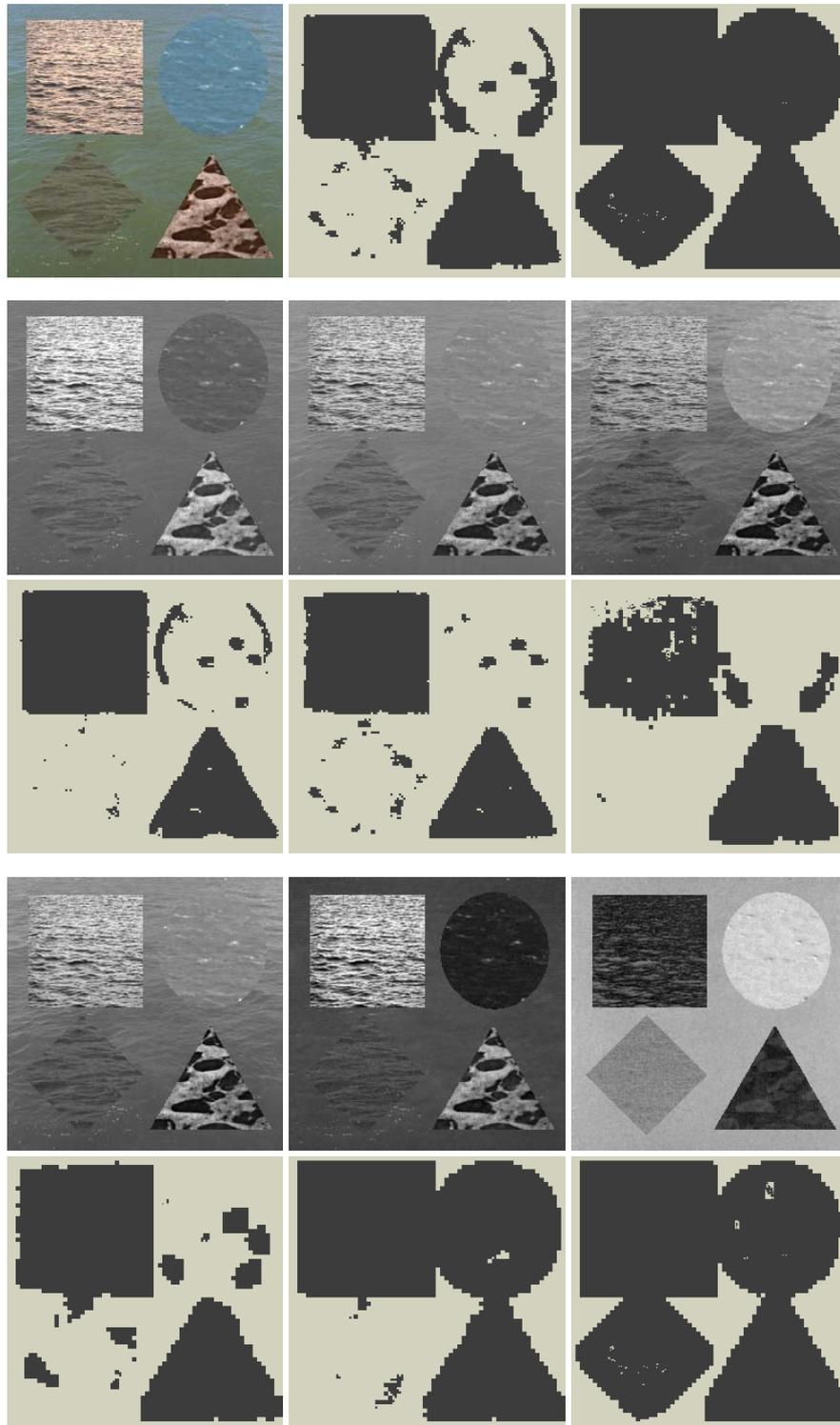


Figure 5.14: Novelty detection comparison on a collage of 5 texture images - first row: Original collage image, novelty detection result in RGB channels, and novelty detection result in image eigenchannels; second row: RGB individual channels; third row: Novelty detection results in each corresponding RGB channel; fourth row: Image eigenchannels of the collage image; last row: Novelty detection results in each individual eigenchannel.

Table 5.2: Novelty detection comparison using graylevel texems in image RGB channels and image eigenchannels (values are %s).

No.	RGB channels			Eigenchannels		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
1	81.7	100	90.7	82.0	100	90.9
2	80.7	100	90.2	80.8	100	90.3
3	87.6	99.9	93.7	82.4	100	91.1
4	94.3	97.2	95.7	93.9	95.7	94.8
5	87.3	30.7	59.3	77.9	99.6	88.6
6	76.6	100	88.2	77.8	100	88.8
7	96.0	93.4	94.7	90.1	98.6	94.3
8	87.8	97.7	92.7	85.6	95.3	90.4
9	85.5	52.0	68.9	76.1	100	87.9
10	92.2	25.2	59.1	77.8	99.2	88.4
11	89.1	33.6	61.6	80.3	97.2	88.6
12	82.5	88.4	85.4	79.5	97.7	88.5
13	93.5	47.8	70.9	93.0	49.0	71.2
14	80.9	99.9	90.3	81.1	100	90.5
15	98.7	55.3	77.2	98.3	74.8	86.7
16	84.5	78.1	81.3	86.5	92.7	89.6
17	75.1	60.8	67.9	62.3	87.9	73.8
18	64.9	69.5	67.2	60.9	91.9	74.8
19	75.1	60.0	67.5	57.0	87.4	72.2
20	83.9	91.8	87.8	85.4	90.0	87.7
21	78.6	97.3	87.8	88.4	98.4	93.4
22	88.5	49.8	69.4	79.5	76.3	77.9
23	98.2	44.5	71.6	96.6	34.8	66.0
24	60.6	69.8	65.2	64.5	86.8	75.7
25	58.7	100	79.4	64.8	99.9	82.3
26	84.1	91.6	87.9	76.5	94.2	85.3
27	73.2	87.8	80.5	64.7	99.9	82.3
28	74.5	88.3	81.4	65.7	94.6	80.1
Overall	82.7	75.4	79.1	78.9	90.8	84.7

shapes were treated as defects which should be detected and localised to simulate the process of novelty detection. The first example collage, i.e. the top-left image in Figure 5.13, is a composition of two steel textures with very similar appearance but slightly different chromatic characteristics. Its separate RGB channels are presented in the second row, from which it can be seen that the channels are highly correlated without much significant difference among them. As the foreground and background textures are very close to each other, the texem analysis in RGB individual channels resulted in failure and all three channels reported similar, non-complementary results as shown in the third row of Figure 5.13. However, the projected transformation of the testing image based on

a reference eigenspace derived from the background image reveals much more information about the texture composite (see the fourth row). The first eigenchannel is closely correlated to image intensity. A lack of chromatic information in this channel, as shown in the last row, only indicated novelties along part of the circular boundary. The next two eigenchannels mainly exhibit chromatic appearance and clearly discriminate the foreground object from the background, which is reflected in the detection results in these channels (see the last row). The final combined detection results of both methods are shown in the first row, next to the original collage. Texem analysis in image eigenchannels significantly outperforms that in RGB channels.

The next example in Figure 5.14 shows a collage made up of five texture images. Again, the individual RGB channel separation did not represent the three dimensional data well. The texture regions in the ellipse and rhombus shapes possess similar chromato-textural properties to the background. Texem analysis failed to localise these two regions in RGB channels. However, PCA based projection gives much better representation. Particularly, the last eigenchannel clearly distinguishes the foreground regions (see the fourth row).

The results of the graylevel texem based novelty detection method applied in RGB image channels and eigenchannels for 28 texture collage images are shown in Table 5.2. The overall accuracy of texem analysis in RGB image channels was 79.1%. While in the image eigenchannels it was 84.7%. In only 4 individual cases (collage number 3, 20, 26, 28) did the proposed eigenchannel texem approach obtain a slightly lower accuracy. Later, in Chapter 6, these results are compared with a Gabor filtering based method and colour texem based novelty detection.

The proposed method was also tested on the COLOURSET dataset comprising ten different families of tile textures, 1018 images in total. The parameter settings for the texem analysis remained the same, i.e. twelve 5×5 texems at each resolution level and around 5 to 10 images in each tile set for training.

Figure 5.15 shows a random texture example with a physical defect on the left side and clusters of pin holes in the lower right region. The potentially defective regions detected at each resolution level across the eigenchannels are also shown, demonstrating that the texems have good sensitivity to defective regions at different scales. As the resolutions progress from coarse to fine, additional evidence for the defective region is gathered, e.g. the pin holes are missing in the defect maps of level 3 and 4 but are detected at finer scales. Also, a few false alarms at level 4 have no further support at other scales. Once again, this shows the combination rules can eliminate false positive regions from the final error map.

Figure 5.16 shows three more tile images from the same family as the image shown in Figure 5.15. The proposed method correctly detected and localised both subtle individual pin holes and clusters of pin holes in a marble texture. Fig. 5.17 shows another three examples but from three different

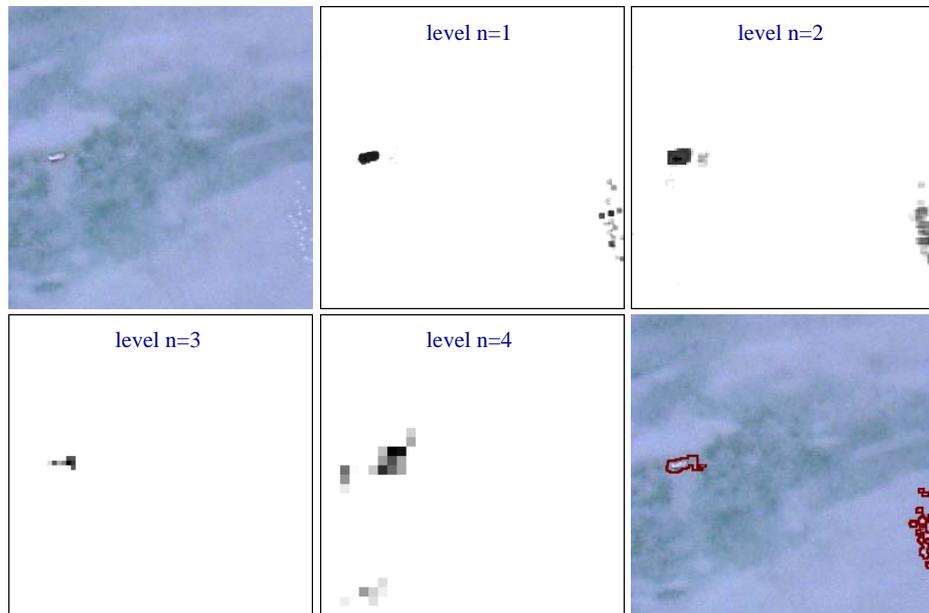


Figure 5.15: Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 4$, and the final defective regions superimposed on the original image.

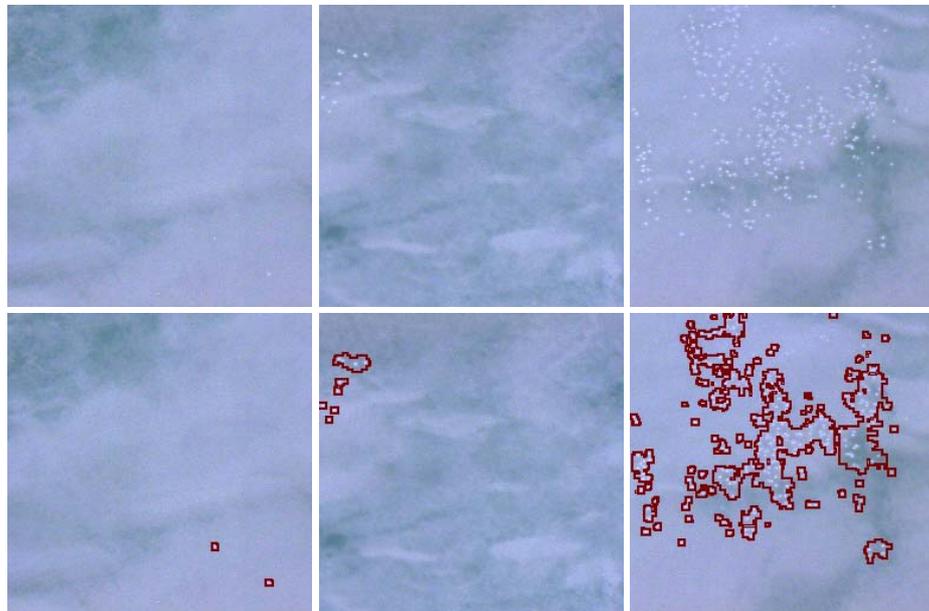


Figure 5.16: Defect localisation- detecting subtle pin holes as shown in the first two columns, and clusters of pin holes as shown in the last column.

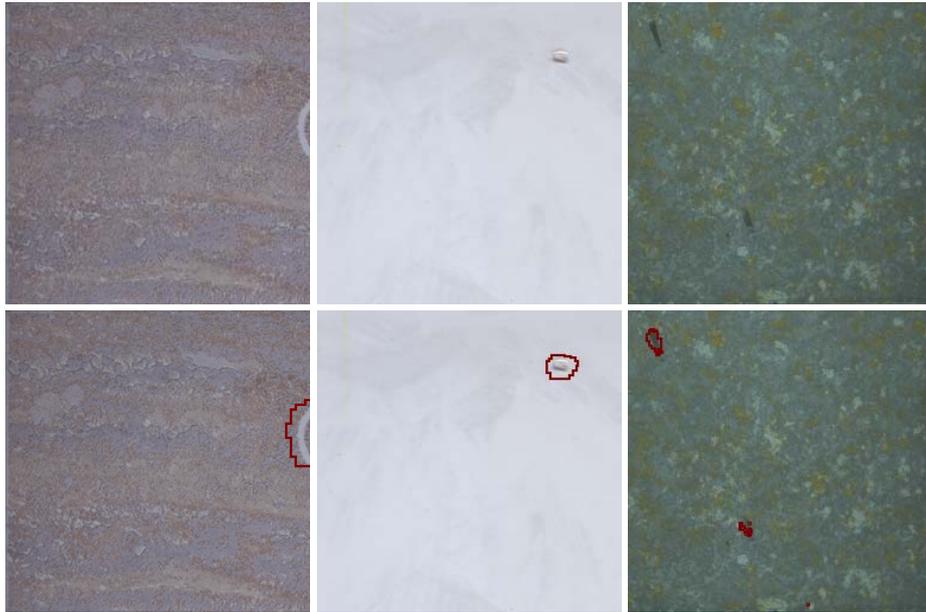


Figure 5.17: Defect localisation - detecting missing print (first column), physical defect (second column), and printing error (last column).

Table 5.3: COLOURSET results using graylevel texems in image eigenchannels (values are %s)

No.	Tile Type	specificity	sensitivity	accuracy
1	PRAN	88.5	88.5	88.5
2	ARDES	83.3	90.0	86.4
3	BSAT	80.0	87.5	85.0
4	CSA	85.7	92.1	88.3
5	DJZU	80.0	90.0	85.0
6	GRAN	80.0	100	88.9
7	BJUP	80.0	88.5	84.8
8	KNGY	86.1	90.3	87.1
9	SONE	84.3	82.6	83.9
10	WBOT	88.8	86.5	88.6
Overall		84.8	88.2	86.4

tile sets, consisting missing ink, physical, and printing defects. The proposed method obtained 86.5% sensitivity, 84.8% specificity, and 88.2% overall accuracy across the COLOURSET dataset as shown in Table 5.3.

5.3.5 Conclusions

In this section, texem based defect detection was extended to deal with colour images by processing the images in separate channels. However, instead of processing in the RGB channels directly, PCA was used to decorrelate them first. A reference eigenspace was obtained by analysing a collection of defect-free samples. Thus, the factorisation of the colour data resulted in a relatively small loss of information. Texem based local contextual analysis was then performed in individual eigenchannels. Defect detection and localisation was based on automatically derived thresholds in a novelty detection framework which required just a few defect-free samples for training. The results demonstrate that the graylevel texem is also a plausible approach to perform colour analysis with relatively economic computational complexity, roughly three times that of graylevel texems on graylevel images.

However, by decomposing the colour image and analysing eigenchannels individually, the inter-channel interactions were not taken into account. There is clear need for a complete three-dimensional model which takes into account both the intra-channel spatial interactions and the inter-channel interactions at the same time.

Chapter 6

COLOUR TEXEMS

6.1 Introduction

In Chapter 5, two different approaches to extend graylevel texems were presented. One was to perform texem analysis in RGB image channels directly, and the other in three decorrelated eigenchannels. The comparative results given in Chapter 5 showed that the decorrelated channel separation performed better.

In this chapter, a simple statistical model to represent colour textures is introduced, which serves as another extension of graylevel texems. The spatial arrangement and interspectral properties of pixels are naturally modelled simultaneously without decomposing images into separate channels. In Chapter 5, the patches were clustered (in grayscale) and GMM was used to learn a relatively small number of texems. However, here a different formulation for texem representation is used with a different inference procedure so that no vectorisation of image patches is required. These new texems then can be generally applied to both scalar and vector-valued data.

Again, novelty detection is performed based on these texems along with automatically derived lower bound data likelihoods of defect-free samples. The overall method is also implemented within a multiscale framework. Then, the results on unlabelled defects on ceramic tiles are presented. For evaluation and validation, novelty detection results based on labelled texture collages from the VisTex database [112] are produced. The results are compared against a popular filter bank approach, as well as graylevel texems with different factorisation schemes. Other applications of colour texems are also presented: (a) abnormality detection in medical images which involves active contour segmentation and colour texem based novelty detection, and (b) unsupervised image segmentation on ink-jet printed tiles which involves colour texem analysis and multiscale fusion.

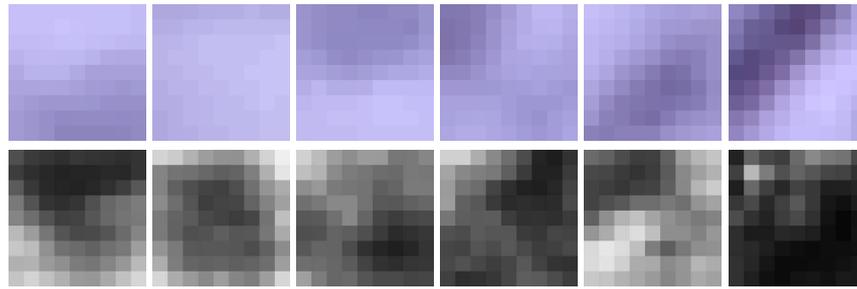


Figure 6.1: Six 9×9 texems extracted from the images shown in Fig. 5.1. First row: the means of texems. Second row: the corresponding covariance images (Images are enhanced for viewing purposes).

6.2 Colour Texem Model

Texems are implicit representations of textural primitives. Images are considered to be generated from superposition of these texems with added noise, as shown in the previous chapter. However, instead of vectorising patches and assuming Gaussian distribution for each texem, Gaussian distribution at each pixel position within each texem is assumed. Various sizes of texems are still necessary to capture sufficient texture properties. Additionally, they give extra flexibility to the texem model. Smaller texem sizes emphasize global information, while larger texems impose more stringent spatial interactions. Again, it becomes clearer when considering the extreme cases given earlier: a single pixel size texem analysis is indeed histogram analysis and full image size texem modelling turns into multiple template analysis.

In brief, the patches of sample images are grouped into clusters, dependent on the patch size, and describe the clusters using a mixture model (different to the standard mixture of Gaussians). The texems are then learnt through an EM algorithm applied on the model parameters. Then, as we are interested in localising the defective regions, a small patch is extracted at each pixel position of the testing image and is inspected using the set of texems obtained at the training stage. A new novelty score is proposed and evaluated to then compute a defect map for each test image.

6.2.1 Learning colour texems

Contrary to the grayscale texem representation described in the previous chapter where each texem was represented by a single multivariate Gaussian function, it is assumed, for colour texems: (a) that pixels are statistically independent for a given texem; and (b) a Gaussian distribution at each pixel position of the texem. The conditional independence assumption among pixels within the local neighbourhood makes the parameter estimation tractable. Each texem \mathbf{m} is now defined

by a vector of means, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_S]$, and a corresponding vector of covariance matrices, $\boldsymbol{\omega} = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \dots, \boldsymbol{\omega}_S]$, i.e. $\mathbf{m} = \{\boldsymbol{\mu}, \boldsymbol{\omega}\}$. Note, $\boldsymbol{\mu}_j$ is a 3×1 colour vector, and $\boldsymbol{\omega}_j$ is a 3×3 matrix characterising the covariance in the colour space. Figure 6.1 illustrates six 9×9 texems extracted from one of the images shown in Figure 5.1. They are arranged according to their descending order of *priors* α_k . Please note that each element $\boldsymbol{\omega}_j$ in $\boldsymbol{\omega}$ is visualised using total variance of $\boldsymbol{\omega}_j$, i.e. $\sum \text{diag}(\boldsymbol{\omega}_j)$, (this is for visualisation purpose only). Similar to the graylevel texem model, the original colour image \mathbf{I} is broken down into a set of P patches $\mathbf{Z} = \{\mathbf{Z}_i\}_{i=1}^P$, each containing pixels from a subset of image coordinates. Square patches of size $d = N \times N$ are used. The patches may overlap and can be of various sizes. It is assumed that there exist K texems, $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^K$, $K \ll P$, with mean colour at each pixel position in $\boldsymbol{\mu}_k$ and corresponding covariance matrix in $\boldsymbol{\omega}_k$ for image \mathbf{I} such that each patch in \mathbf{Z} can be generated from a texem by copying each mean colour with certain added variations governed by the corresponding covariance matrix. Thus, the probability for image patch \mathbf{Z}_i belonging to the k th texem can be given as a joint probability at each pixel position, i.e.

$$p(\mathbf{Z}_i | \mathbf{m}_k) = p(\mathbf{Z}_i | \boldsymbol{\mu}_k, \boldsymbol{\omega}_k) = \prod_{j \in S} \mathcal{N}(\mathbf{Z}_{j,i}; \boldsymbol{\mu}_{j,k}, \boldsymbol{\omega}_{j,k}), \quad (6.1)$$

where S is the pixel grid as before, and $\mathcal{N}(\mathbf{Z}_{j,i}; \boldsymbol{\mu}_{j,k}, \boldsymbol{\omega}_{j,k})$ is a Gaussian distribution over $\mathbf{Z}_{j,i}$ with mean $\boldsymbol{\mu}_{j,k}$ and covariance $\boldsymbol{\omega}_{j,k}$ at the j th pixel position in the k th texem:

$$\mathcal{N}(\mathbf{Z}_{j,i}; \boldsymbol{\mu}_{j,k}, \boldsymbol{\omega}_{j,k}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_{j,k}|}} \exp\left\{-\frac{1}{2}(\mathbf{Z}_{j,i} - \boldsymbol{\mu}_{j,k})^T \boldsymbol{\omega}_{j,k}^{-1} (\mathbf{Z}_{j,i} - \boldsymbol{\mu}_{j,k})\right\}. \quad (6.2)$$

The probability function shown in (6.1) is similar to the epitome model in [70], where conditional independence was also assumed within pixels in a same patch. However, instead of condensing all image patches into a single miniature with a hidden mapping, the following mixture model is used:

$$p(\mathbf{Z}_i | \Theta) = \sum_{k=1}^K p(\mathbf{Z}_i | \mathbf{m}_k) \alpha_k, \quad (6.3)$$

where the parameters are $\Theta = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_K, \mathbf{m}_1, \dots, \mathbf{m}_K)$, and α_k is the *prior* probability of k th texem constrained by $\sum_{k=1}^K \alpha_k = 1$. Since all the texems \mathbf{m}_k are unknown, the parameter set Θ can be determined first by optimising the data log likelihood expression of the entire set \mathbf{Z} . The log-likelihood expression for this density from the data \mathbf{Z} is given by:

$$\log(\mathcal{L}(\Theta | \mathbf{Z})) = \log \prod_{i=1}^P p(\mathbf{Z}_i | \Theta) = \sum_{i=1}^P \log \left(\sum_{k=1}^K p(\mathbf{Z}_i | \mathbf{m}_k) \alpha_k \right) \quad (6.4)$$

Hence, the objective is to estimate the parameter Θ for a given number of texems, K . Again, the EM algorithm is employed to find the maximum likelihood estimate of the mixture model parameters. The reader can refer to (6.14) and (6.15) for final parameter solutions and skip the following, which details the derivation of the parameters, without loss of continuity.

Here, the derivation of the parameter set Θ using the EM technique is briefly presented. Again, the missing data approach is used. The following deduction steps are analogous to that of the standard Gaussian mixture model given in Appendix A, however the component probability density function is now expressed as a product of Gaussians, instead of a single multivariate Gaussian function. As (6.4) contains the log of the sum, it becomes difficult to derive optimised solutions. However, \mathbf{Z} can be treated as observed data and consider that there exist hidden, unobserved, data items $\mathcal{Y} = \{y_i\}_{i=1}^P$ that determine which texem “generated” each patch \mathbf{Z}_i . Thus, the likelihood of the complete data can be written as:

$$\begin{aligned} \log(\mathcal{L}(\Theta|\mathbf{Z}, \mathcal{Y})) &= \log(p(\mathbf{Z}, \mathcal{Y}|\Theta)) = \sum_{i=1}^P \log(p(\mathbf{Z}_i|y_i)p(y_i)) \\ &= \sum_{i=1}^P \log(p(\mathbf{Z}_i|\mathbf{m}_{y_i})\alpha_{y_i}). \end{aligned} \quad (6.5)$$

We first derive an expression for the distribution of the unobserved data and then iteratively approximate the true parameters. It starts with a guess that $\Theta^{(t)} = (\alpha_1^{(t)}, \dots, \alpha_K^{(t)}, \mathbf{m}_1^{(t)}, \dots, \mathbf{m}_K^{(t)})$ are the appropriate parameters for the likelihood $\mathcal{L}(\Theta^{(t)}|\mathbf{Z}, \mathcal{Y})$, where t denotes the iteration step beginning with $t = 0$. Then $p(\mathbf{Z}_i|\mathbf{m}_k^{(t)})$ for each i and \mathbf{m}_k according to $\Theta^{(t)}$ can be computed. As mentioned earlier, α_k is the prior probability of each texem. Thus, using Bayes’ rule, we have

$$p(y_i|\mathbf{Z}_i, \Theta^{(t)}) = \frac{p(\mathbf{Z}_i|\mathbf{m}_{y_i}^{(t)})\alpha_{y_i}^{(t)}}{p(\mathbf{Z}_i|\Theta^{(t)})} = \frac{p(\mathbf{Z}_i|\mathbf{m}_{y_i}^{(t)})\alpha_{y_i}^{(t)}}{\sum_{k=1}^K p(\mathbf{Z}_i|\mathbf{m}_k^{(t)})\alpha_k^{(t)}}. \quad (6.6)$$

Let $\mathbf{y} = (y_1, \dots, y_P)$ be an instance of the unobserved data. Then $p(\mathbf{y}|\mathbf{Z}, \Theta^{(t)})$ is the marginal distribution of the unobserved data, given as:

$$p(\mathbf{y}|\mathbf{Z}, \Theta^{(t)}) = \prod_{i=1}^P p(y_i|\mathbf{Z}_i, \Theta^{(t)}). \quad (6.7)$$

The EM algorithm first finds the expected value of the complete-data log-likelihood $\log p(\mathbf{Z}, \mathcal{Y}|\Theta)$ with respect to the unknown data \mathcal{Y} given the observed data \mathbf{Z} and the current parameter estimates.

Thus, we define

$$\begin{aligned} Q(\Theta, \Theta^{(t)}) &= E[\log p(\mathbf{Z}, \mathcal{Y} | \Theta | \mathbf{Z}, \Theta^{(t)})] \\ &= \sum_{\mathbf{y} \in \Upsilon} \log(\mathcal{L}(\Theta | \mathbf{Z}, \mathbf{y})) p(\mathbf{y} | \mathbf{Z}, \Theta^{(t)}), \end{aligned} \quad (6.8)$$

where $E[\cdot]$ denotes expectation, $k \in 1, \dots, K$, and Υ is the space where \mathbf{y} can draw values from. The equation can be expanded and then simplified as:

$$\begin{aligned} Q(\Theta, \Theta^{(t)}) &= \sum_{k=1}^K \sum_{i=1}^P \log(\alpha_k) p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}) \\ &\quad + \sum_{k=1}^K \sum_{i=1}^P \log(p(\mathbf{Z}_i | \mathbf{m}_k)) p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}), \end{aligned} \quad (6.9)$$

The term containing α_k and the term containing \mathbf{m}_k can be maximised independently. The Lagrange multiplier τ can be introduced with the constraint that $\sum_k \alpha_k = 1$, and solve the following equation:

$$\frac{\partial}{\partial \alpha_k} \left(\sum_{k=1}^K \sum_{i=1}^P \log(\alpha_k) p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}) + \tau \left(\sum_k \alpha_k - 1 \right) \right) = 0. \quad (6.10)$$

Thus α_k is solved by summing both sides of the extended expression of (6.10):

$$\alpha_k = \frac{1}{P} \sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}). \quad (6.11)$$

Then \mathbf{m}_k needs to be solved, which in this case contains $(\boldsymbol{\mu}_k, \boldsymbol{\omega}_k)$. We can take the log of $p(\mathbf{Z}_i | \boldsymbol{\mu}_k, \boldsymbol{\omega}_k)$ to get

$$\begin{aligned} &\sum_{k=1}^K \sum_{i=1}^P \log(p(\mathbf{Z}_i | \boldsymbol{\mu}_k, \boldsymbol{\omega}_k)) p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}) \\ &= \sum_{k=1}^K \sum_{i=1}^P \sum_{j \in S} \left(\log \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\omega}_{j,k}|^{\frac{1}{2}}} - \frac{1}{2} (\mathbf{Z}_{j,i} - \boldsymbol{\mu}_{j,k})^T \boldsymbol{\omega}_{j,k}^{-1} (\mathbf{Z}_{j,i} - \boldsymbol{\mu}_{j,k}) \right) p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}). \end{aligned} \quad (6.12)$$

Thus we need to take partial derivatives of (6.12) with respect to $\boldsymbol{\mu}_{j,k}$ and set it to zero:

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{\mu}_{j,k}} \left(\sum_{k=1}^K \sum_{i=1}^P \sum_{j \in S} \left(\log \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\omega}_{j,k}|^{\frac{1}{2}}} - \frac{1}{2} (\mathbf{Z}_{j,i} - \boldsymbol{\mu}_{j,k})^T \boldsymbol{\omega}_{j,k}^{-1} (\mathbf{Z}_{j,i} - \boldsymbol{\mu}_{j,k}) \right) p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}) \right) \\ &= 0, \end{aligned} \quad (6.13)$$

which solves the parameter $\mu_{j,k}$. In a similar fashion, $\omega_{j,k}$ can be solved.

Hence, the estimation of texems' parameters can be solved according to the following equations:

$$\begin{aligned}
\hat{\alpha}_k &= \frac{1}{P} \sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}), \\
\hat{\mu}_k &= \{\hat{\mu}_{j,k}\}_{j \in S}, \\
\hat{\omega}_k &= \{\hat{\omega}_{j,k}\}_{j \in S}, \\
\hat{\mu}_{j,k} &= \frac{\sum_{i=1}^P \mathbf{Z}_{j,i} p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)})}, \\
\hat{\omega}_{j,k} &= \frac{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}) (\mathbf{Z}_{j,i} - \hat{\mu}_{j,k}) (\mathbf{Z}_{j,i} - \hat{\mu}_{j,k})^T}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)})},
\end{aligned} \tag{6.14}$$

where according to (6.6) $p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)})$ can be written as

$$p(\mathbf{m}_k | \mathbf{Z}_i, \Theta^{(t)}) = \frac{p(\mathbf{Z}_i | \mathbf{m}_k, \Theta^{(t)}) \alpha_k^{(t)}}{p(\mathbf{Z}_i | \Theta^{(t)})} = \frac{p(\mathbf{Z}_i | \mathbf{m}_k, \Theta^{(t)}) \alpha_k^{(t)}}{\sum_{k=1}^K p(\mathbf{Z}_i | \mathbf{m}_k, \Theta^{(t)}) \alpha_k^{(t)}}. \tag{6.15}$$

Then the two stages of EM are as follows. The E-step involves a soft-assignment of each patch \mathbf{Z}_i to texems, \mathcal{M} , according to (6.15) with an initial guess of the true parameters, Θ . This initialisation can be set randomly, although K-means is used to compute a simple estimate with K equal to the number of texems. The M-step then updates the parameter estimations given by (6.14) and (6.15). The E-step and M-step are iterated until the estimations are stabilised or the rate of improvement of the likelihood falls below a pre-specified convergence threshold. Then the colour texem is stated as $\mathbf{m}_k = \{\mu_{j,k}, \omega_{j,k}\}_{j \in S}$.

Various sizes of texems can be used and they can overlap to ensure they capture sufficient textural characteristics. Alternatively, similar to graylevel texems, multiscale texems can be used. The same Gaussian pyramid can be used to generate texems at a more computationally manageable size, such as 5×5 . During the EM process, the stabilised estimation of a coarser level is used as the initial estimation for the finer level, i.e. $\hat{\Theta}^{(n,t=0)} = \Theta^{(n+1)}$, which helps speed up the convergence and achieve a more accurate estimation.

Another texem example is shown in Figure 6.2, where eight 7×7 colour texems are extracted from the ‘‘baboon’’ image. Although a very small patch size was used, the texems still managed to capture the micro structures. Each prior, α_k , can be treated as a measurement of the contribution from each texem. The input image then can be viewed as a superposition of various sizes of means

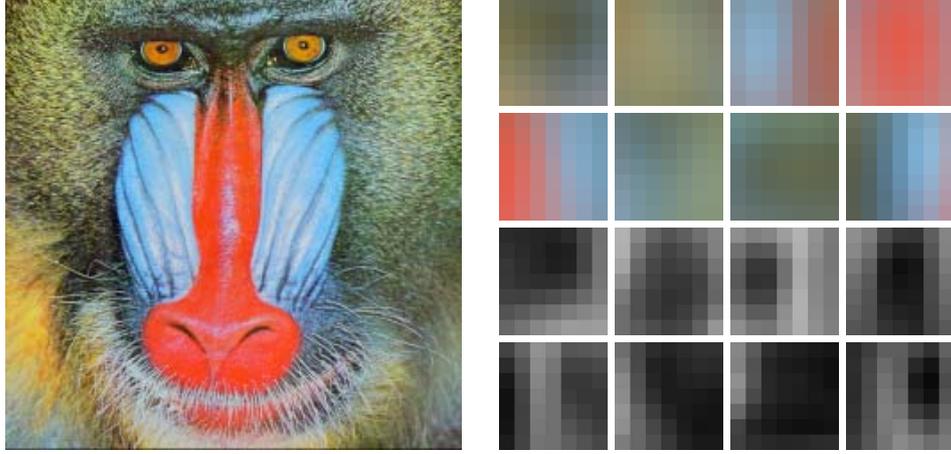


Figure 6.2: Eight 7×7 texems extracted from the image on the left. The first two rows on the right side show the μ_k , and the last two rows show the corresponding ω_k .

μ_k with added variations at each pixel position governed by the corresponding covariance matrices ω_k .

6.2.2 Training and testing

Novelty detection frees the application from having to provide a portfolio of defects within a supervised training stage. In fact, as demonstrated in the previous chapter, texems lend themselves well for performing unsupervised training and testing for novelty detection. This is achieved by automatically determining the minimum bounds of normal samples in each resolution level. This was performed for graylevel texems and it now adapted for colour texems.

For training, a small number of defect free samples (e.g. 4 or 5 only) are arranged within the multiscale framework, and patches with the same texem size are extracted. The probability of a patch $\mathbf{Z}_i^{(n)}$ belonging to texems in the corresponding n th scale is:

$$p(\mathbf{Z}_i^{(n)} | \Theta^{(n)}) = \sum_{k=1}^{K^{(n)}} p(\mathbf{Z}_i^{(n)} | \mathbf{m}_k^{(n)}) \alpha_k^{(n)}, \quad (6.16)$$

where $\Theta^{(n)}$ is the parameter set at level n , $\mathbf{m}_k^{(n)}$ is the k th texem at level n , and $p(\mathbf{Z}_i^{(n)} | \mathbf{m}_k^{(n)})$ is a product of Gaussian distributions as shown in (6.1) with parameters associated to texem set, \mathcal{M} . Based on this probability function, a novelty score function \mathcal{V} can be defined as the negative log likelihood:

$$\mathcal{V}(\mathbf{Z}_i^{(n)} | \Theta^{(n)}) = -\log p(\mathbf{Z}_i^{(n)} | \Theta^{(n)}). \quad (6.17)$$

The lower the novelty score, the more likely the patch belongs to the same family and vice versa. Thus, it can be viewed as a same source similarity measurement. The distribution of the scores for all the patches $\mathbf{Z}^{(n)}$ at level n of the pyramid forms a 1D novelty score space which is not necessarily a simple Gaussian distribution. In order to find the upper bound of the novelty score space of defect-free patches (or the lower bound of data likelihood), K-means clustering is performed in this space to approximately model the space. The cluster with the maximum mean is the component of the novelty score distribution at the boundary between good and defective textures. This component is characterized by mean $u^{(n)}$ and standard deviation $\sigma^{(n)}$. This K-means scheme replaces the single Gaussian distribution assumption in the novelty score space, which is commonly adopted in a parametric classifier in novelty detection, e.g. [114] and for which the correct parameter selection is critical. Instead, dividing the novelty score space and finding the critical component, here called the boundary component, can effectively lower the parameter sensitivity. The value of K should be generally small (we empirically fixed it at 5). It is also notable that a single Gaussian classifier is a special case of the above scheme, i.e. when $K = 1$. The maximum novelty score (or the minimum data likelihood), $\Lambda^{(n)}$ of a patch $\mathbf{Z}_i^{(n)}$ at level n across the training images is then established as:

$$\Lambda^{(n)} = u^{(n)} + \lambda\sigma^{(n)}, \quad (6.18)$$

where λ is a constant (set to $\lambda = 2$ in the experiments in Section 6.3). This completes the training stage in which, with only a very few defect-free images, we determine the texems and an automatic threshold for marking new image patches as defect free or defective.

In the testing stage, the image under inspection is again layered into a multiscale framework and patches at each pixel position (x, y) at each level n are examined against the learnt texems. The probability for each patch and its novelty score are then calculated according to (6.16) and (6.17) and compared to the minimum data likelihood, determined by $\Lambda^{(n)}$, at the corresponding level. Let $Q^{(n)}(x, y)$ be the novelty score map at the n th resolution level. Then, the potential defect map, $\mathcal{D}^{(n)}(x, y)$, at level n is:

$$\mathcal{D}^{(n)}(x, y) = \begin{cases} 0 & \text{if } Q^{(n)}(x, y) \leq \Lambda^{(n)} \\ Q^{(n)}(x, y) - \Lambda^{(n)} & \text{otherwise.} \end{cases} \quad (6.19)$$

Next, the information coming from all the resolution levels must be combined to build the certainty of the defect at position (x, y) . The logical processing used for graylevel texems again is adopted to combine information from different levels of a multiscale pyramid and to reduce false alarms. It assumes that a defect must appear in at least two adjacent resolution levels for it to be certified as such. Using a logical AND, implemented through the geometric mean, of every pair of adjacent

levels, a set of combined maps are then obtained as:

$$\mathcal{D}^{(n,n+1)}(x, y) = [\mathcal{D}^{(n)}(x, y)\mathcal{D}^{(n+1)}(x, y)]^{1/2}. \quad (6.20)$$

Note each $\mathcal{D}^{(n+1)}(x, y)$ is scaled up to be the same size as $\mathcal{D}^{(n)}(x, y)$. This operation reduces false alarms and yet preserves most of the defective areas. Next, the resulting $\mathcal{D}^{(1,2)}(x, y)$, $\mathcal{D}^{(2,3)}(x, y)$, ..., $\mathcal{D}^{(l-1,l)}(x, y)$ are combined in a logical OR, as the arithmetic mean, to provide a final map of the joint contribution of all the defects detected across all the scales:

$$\mathcal{D}(x, y) = \frac{1}{l-1} \sum_{n=1}^{l-1} \mathcal{D}^{(n,n+1)}(x, y). \quad (6.21)$$

Next, two sets of experiments are presented. In the first set, the result of applying the proposed method to detecting defects on ceramic tiles is presented. As expressed before in Chapter 5 we can not compare and evaluate these localised defects against a groundtruth since the defects in the data set are difficult to localise by hand. However, whole tile classification rates, based on overall “defective” and “non-defective” labelling by factory-floor experts is presented. Moreover, in order to evaluate the proposed method, we outline the result of the experiments on texture collages made from textures in the MIT VisTex texture database [112].

6.3 Results: Ceramic Tile Application

The proposed method was applied to a variety of randomly textured tile images with different types of defects including physical damage, pin holes, textural imperfections, and many more. Details of the colour tile dataset COLOURSET is given in Chapter 3. Similar to graylevel texems, only five to ten defect-free samples were used to extract the texems and to determine the upper bound of the novelty scores $\Lambda^{(n)}$. The number of texems at each resolution level was again 12, and the size of each texem was 5×5 pixels. The number of multiscale levels was $l = 4$. These parameters were fixed throughout the experiments on the variety of random texture tile prints used.

First, two examples are given to demonstrate the multiscale detection and fusion process, as well as to illustrate different spatial sensitivities at different scales. Figure 6.3 shows a random texture example with a defect in the upper right region introduced by a printing problem. The potentially defective regions detected at each resolution level n , $n = 1, \dots, 4$, are marked on the corresponding images in Figure 6.3. It can be seen that the texems show good sensitivity to the defective region at different scales. Information is gathered together when propagating through the coarse

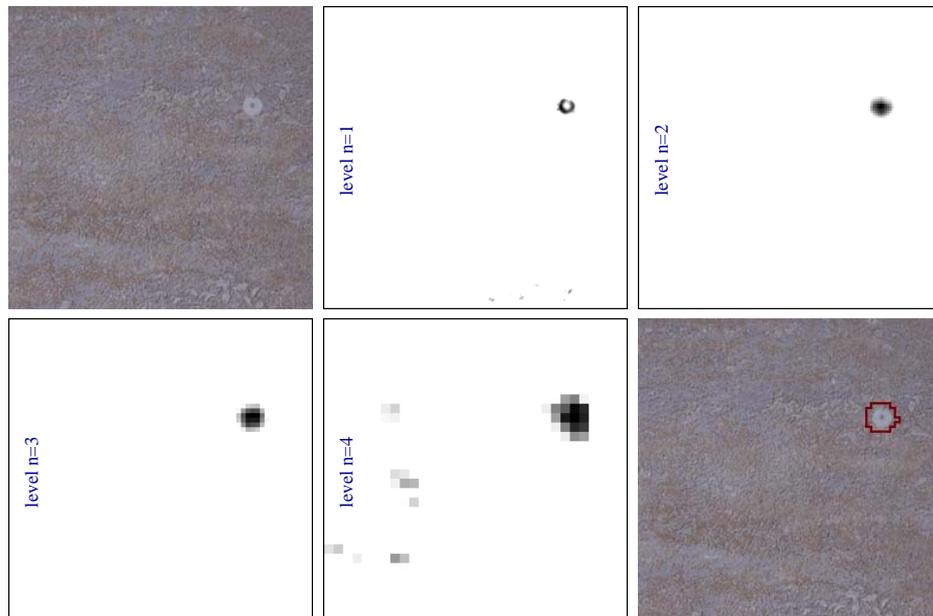


Figure 6.3: Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, 3, 4$, and the final defective regions superimposed on the original image.

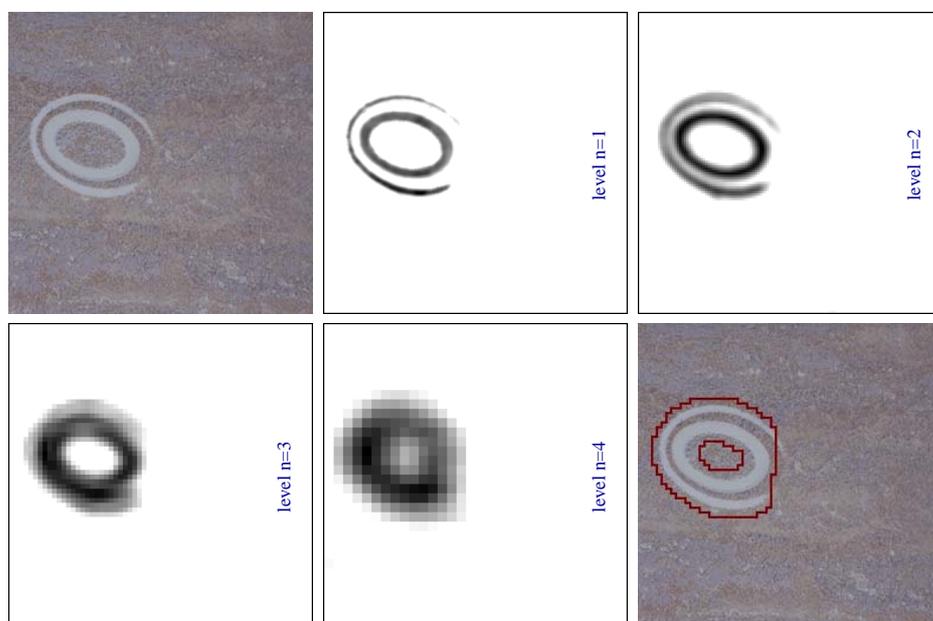


Figure 6.4: Localising textural defects - from top left to bottom right: original defective tile image, detected defective regions at different levels $n = 1, 2, \dots, 4$, and the final defective regions superimposed on the original image.

to fine scales. The final image shows the defect superimposed on the original image. As seen in the previous chapter, the defect fusion process can eliminate false alarms, e.g. see the false alarm regions in levels $n = 1$ and $n = 4$ which disappear after the operations in (6.20) and (6.21). The second example, shown in Figure 6.4, demonstrates that texems capture defective regions with different spatial accuracy given different spatial resolution. At finer scales, the normal pixels falling in the gap between the defective ellipsoid regions were classified as normal, while at coarser levels, these pixels were considered as defective, since their neighbouring pixels contained defective pixels. Thus, the detected defective regions at level 3 and level 4 united those two defective areas into one. The final combination across the scales is given in the last image.

Figure 6.5 shows two families of ceramic tiles, in each of which the textures are varying but have the same visual impression. The first set of images contain a granite-like texture with random appearance. The defects shown are print errors in various shapes and sizes. The proposed method localised the region well. The physical defects shown in the next two examples were accurately localised without any false alarms. The second set shown in Figure 6.5 are cloud-like random textures with loosely placed textured primitives. The method successfully detected structural cracks, surface defect, and chromatic defects of various shapes.

The first set of examples in the top row of Figure 6.6 show a richly decorated texture with crack-like macro-structures. The surfaces are highly textured with defects ranging from obvious misprint along the tile edge, small missing print, and thin surface crack. The chromatic defects shown in the first image were successfully detected. The missing print in the second example is much more difficult to find not only because it is much smaller but also because it tends to blend well in the busy background. Texems found this missing print in the top-right corner, but also indicated a possible defect in the top-centre area which is too subjective to determine. The crack in the last example is even more difficult to localise as it is rather thin and blends well with its background. The next three example images in Figure 6.6 show lightly textured tile surfaces also with a random appearance. The textures exhibit a dominant vertical orientation. The proposed method accurately localised all the defective areas.

Figure 6.7 shows example colour textures from three different tile families with different textural and chromatic characteristics. In each case the proposed method could find structural and chromatic defects of various shapes and sizes. Some example graylevel images and their results are also shown in Figure 6.7 to verify that the new texem framework described in this chapter is equally applicable to scalar valued images.

Graylevel texems were found adequate for most defect detection tasks where defects were still reasonably visible after converting from colour to gray scale. However, colour texems were found to be more powerful particularly for better localisation and when the nature of the chromaticity

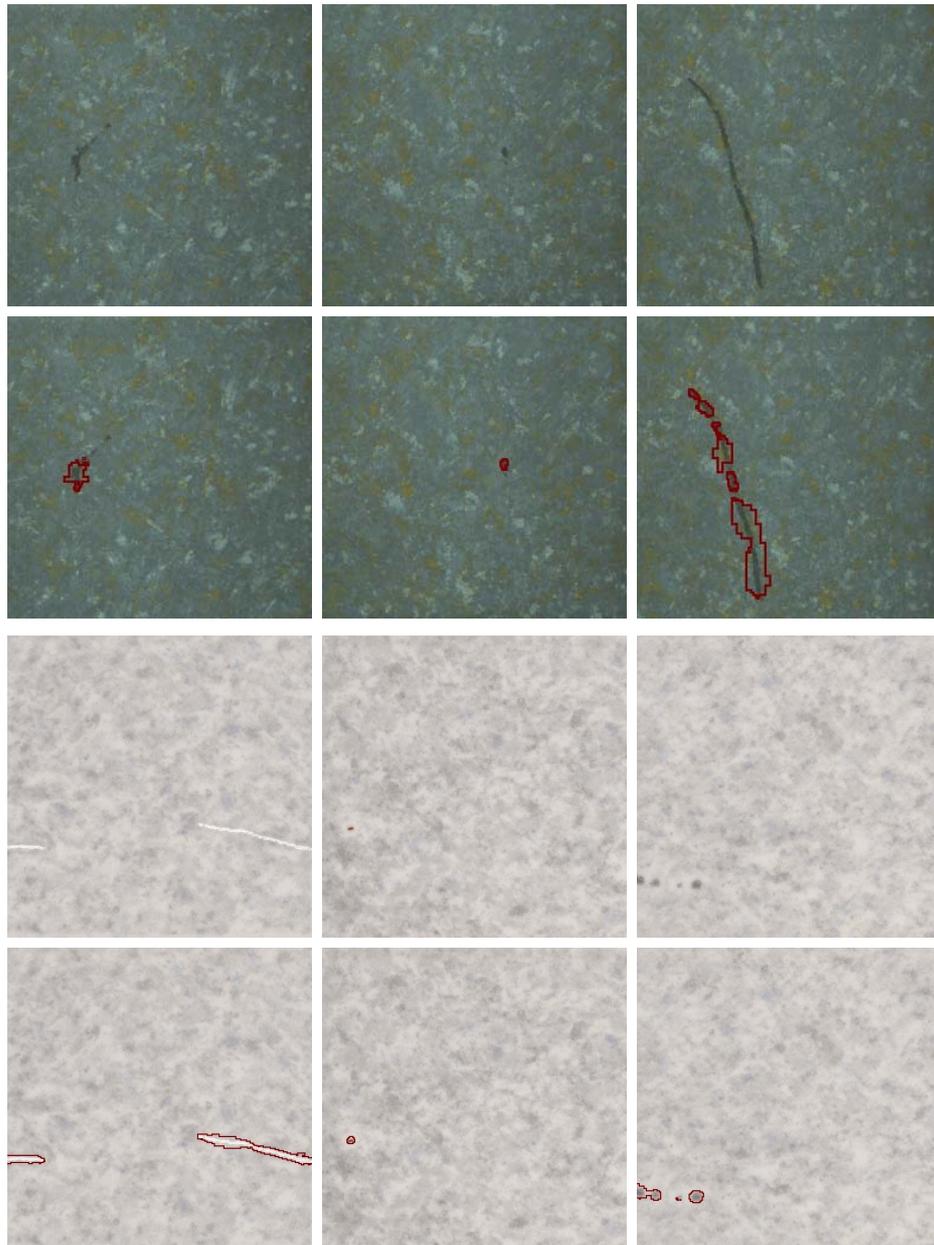


Figure 6.5: Defect localisation - The examples given in the top row show a granite-like random texture. Defects shown in the next row include various sizes of print errors. The next three images show a cloud-like random texture with loosely placed texture primitives. Defects shown in the last row, from left to right, include cracks, surface defect, and misprints.

defect matters. Two examples are given in Figure 6.8. The first shows a tile image with a defective region, which is not only slightly brighter but also less saturated in blue. The colour texem method achieved better results in localising the defect. The second row in Figure 6.8 demonstrates a dif-



Figure 6.6: Defect localisation - The examples given in the top row are from the same tile family richly decorated with macro-structure patterns. Defects shown in the next row, from left to right, include print error, missing print, and thin surface crack. The next three images show a texture with oriented patterns. Defects shown in the last row, from left to right, include misprint, missing print, and subtle surface crack.

ferent type of defect which clearly possesses a different hue from the normal background texture. The colour texems found the affected regions more accurately, while the graylevel texems missed some defective regions altogether.

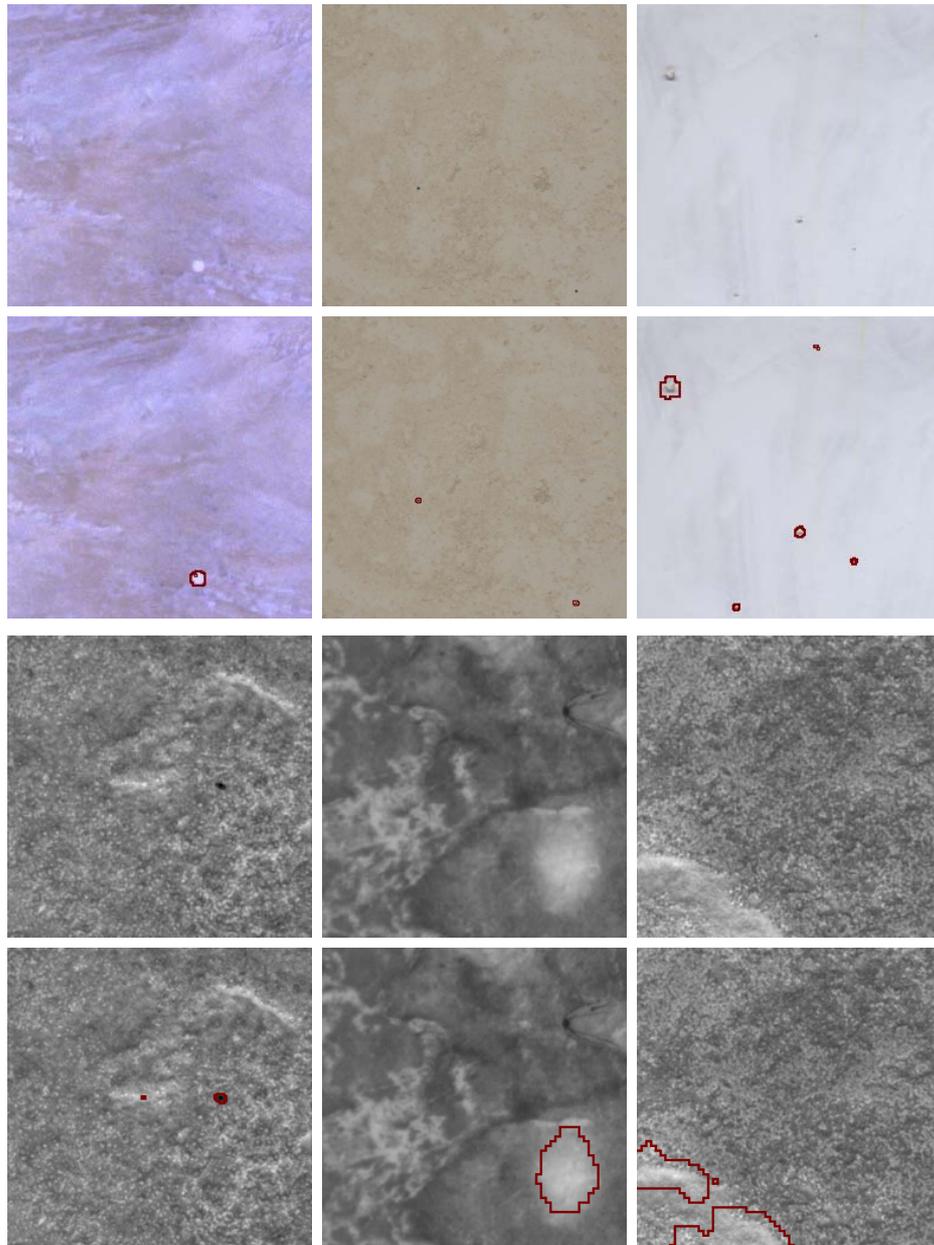


Figure 6.7: Defect localisation - The first row shows example images from three different tile families with different chromato-textural properties. Defect shown in the next row, from left to right, include misprint, dimps, and bumps. The third row shows graylevel examples from three different tile families. Defects shown in the last row, from left to right, include surface hole, missing print, and surface bumps.

The colour texem model was tested on the COLOURSET dataset comprising 1018 tile samples. It obtained a defect detection accuracy rate of 91.1%, with sensitivity at 92.6% and specificity at 89.8% on 10 different families of tiles consisting of 561 normal samples and 457 defective samples

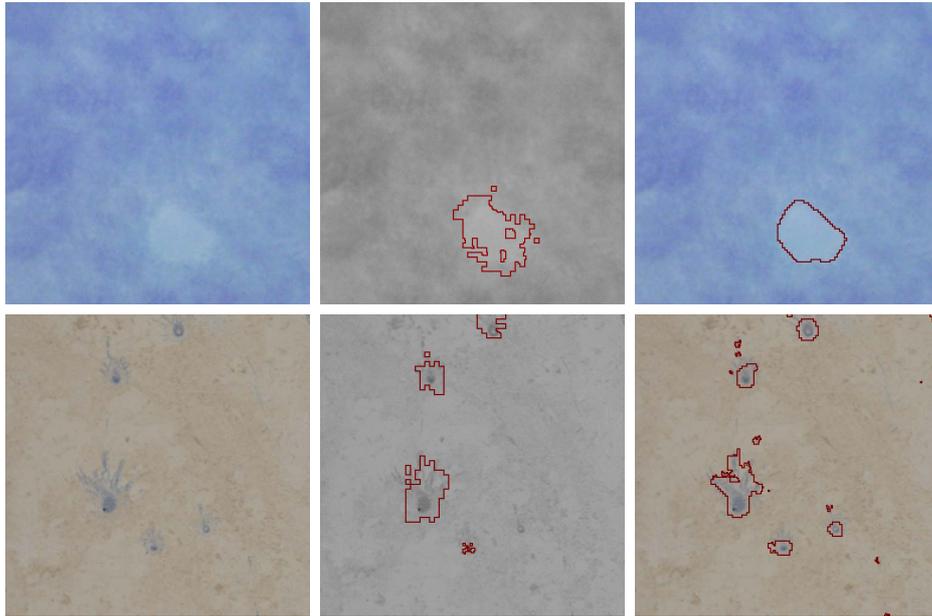


Figure 6.8: Defect localisation comparison: left column - original textures with print errors, middle column - results using graylevel texems, right column - results using colour texems.

Table 6.1: COLOURSET results using colour texems (values are %s)

No.	Tile Type	specificity	sensitivity	accuracy
1	PRAN	90.4	96.2	94.5
2	ARDES	83.3	100	90.9
3	BSAT	90.0	93.8	92.5
4	CSA	89.3	86.9	88.3
5	DJZU	80.0	90.0	85.0
6	GRAN	80.0	100	88.9
7	BJUP	90.0	96.2	93.5
8	KNGY	90.7	96.8	92.1
9	SONE	90.2	82.6	88.4
10	WBOT	91.3	86.5	89.4
Overall		89.8	92.6	91.1

(see Table 6.1).

6.4 Results: Evaluation using VisTex Collages

To evaluate the accuracy of the proposed method, again the set of 28 image collages made up from textures in the MIT VisTex database [112] are used. As detailed in Chapter 3, up to 5 texture

images were used to generate one collage. In each case the background is the learnt texture for which colour texems are generated and the foreground (disk, square, triangle, and rhombus) is treated as the novelty to be detected. Note again, this is not texture segmentation, but rather defect segmentation. The foreground is set to occupy 50% of the whole image to allow the sensitivity and specificity measures have equal weights.

Table 6.2: Novelty detection comparison on VisTex collage images (values are %s).

No.	Escofet's Method			Colour Texems		
	specificity	sensitivity	accuracy	specificity	sensitivity	accuracy
1	95.6	82.7	89.2	91.9	99.9	95.9
2	96.9	83.7	90.3	84.4	100	92.1
3	96.1	61.5	79.0	91.1	99.8	95.4
4	98.0	53.1	75.8	97.0	92.9	95.0
5	98.8	1.5	50.7	92.1	98.8	95.4
6	96.6	70.0	83.4	96.3	98.6	97.4
7	98.9	26.8	63.2	98.6	79.4	89.0
8	91.4	74.4	83.0	89.6	99.8	94.7
9	90.8	49.0	70.1	86.4	100	93.1
10	94.3	7.2	51.2	92.8	99.6	96.2
11	94.6	8.6	52.1	96.3	90.8	93.6
12	86.9	44.0	65.7	88.4	98.8	93.5
13	96.8	71.0	84.0	91.0	91.9	91.5
14	90.7	95.2	93.0	82.5	100	91.1
15	98.4	27.2	63.2	96.5	76.3	86.5
16	95.5	43.0	69.3	96.3	71.2	83.8
17	80.0	56.5	68.2	83.5	98.7	91.1
18	73.9	60.4	67.2	83.9	96.5	90.2
19	84.9	52.0	68.4	90.4	71.3	80.9
20	94.4	52.0	73.2	95.1	88.8	91.9
21	94.0	48.9	71.6	95.8	75.9	85.9
22	95.8	23.4	60.0	92.2	72.0	82.2
23	97.1	35.1	66.5	93.6	67.8	80.9
24	89.4	46.4	67.9	81.6	98.1	89.8
25	82.6	92.9	87.7	88.3	100	93.9
26	94.5	55.3	74.9	94.3	92.2	93.2
27	93.9	36.5	65.2	85.9	98.9	92.4
28	81.2	55.3	68.3	82.0	95.2	88.6
Overall	92.2	50.5	71.5	90.6	91.2	90.9

First, the proposed method was compared against a Gabor filtering based novelty detection method [49]. Gabor filters have been widely used in defect detection, see [49, 83] for typical examples. The work by Escofet *et al.* [49], referred to here as Escofet's method, is the most comparable to ours, as it is (a) performed in a novelty detection framework and (b) uses the same defect fu-

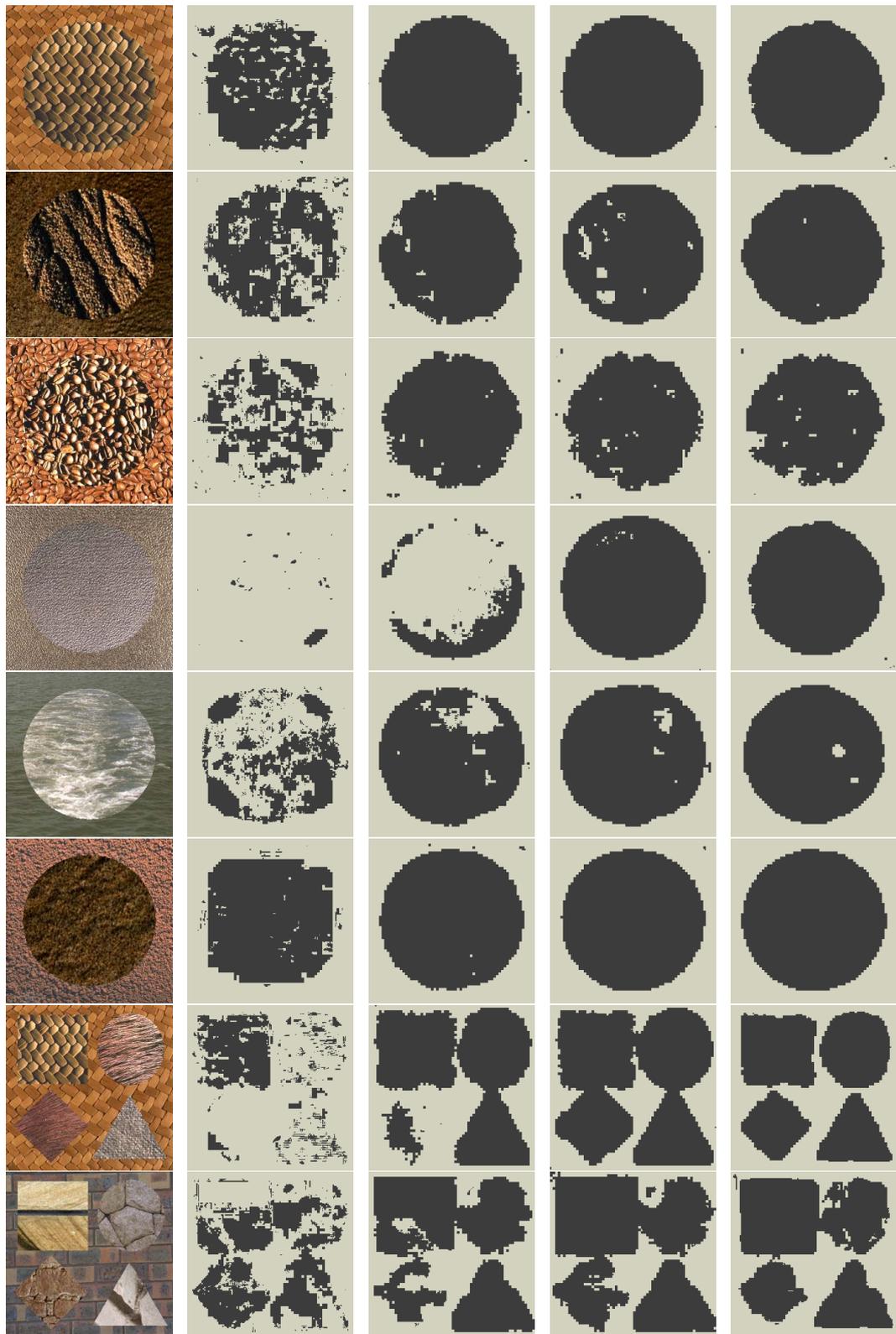


Figure 6.9: Novelty detection comparison on collage images - see text for details.

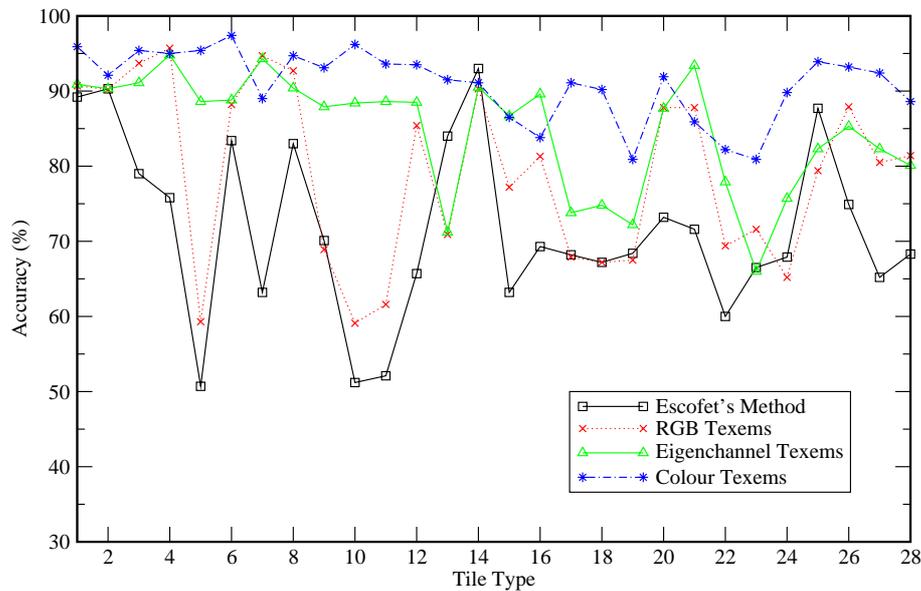


Figure 6.10: A comparison plot for Escofet's Gabor filter bank based method, graylevel texem analysis in RGB channels, graylevel texem analysis in image eigenchannels, and colour texem analysis without factorisation of data.

sion scheme across the scales. Thus, following Escofet's method to perform novelty detection on the synthetic image collages, the images were filtered through a set of 16 Gabor filters, comprising four orientations and four scales. The texture features were extracted from the filtering responses. Feature distributions of normal samples were then used for novelty detection. The same logical process was used to combine defect candidates across the scales. More details can be found in [49] and in Chapter 2. The overall detection accuracy of 71.5% by Escofet's method was significantly lower than the proposed method (see Table 6.2). A more elaborately designed traditional classification scheme and more finely tuned parameters may improve Escofet's Gabor filter based performance, however, the training stage and the detection process will inevitably become lengthier and more difficult to control. Recall that graylevel texem analysis in RGB channels gave an overall accuracy of 79.1% (see Table 5.2) which also outperforms Escofet's filter bank based method. Graylevel texem analysis in decorrelated image eigenchannels improved the overall accuracy to 84.7% (see Table 5.2). For colour texems, an overall detection accuracy of 90.9% with 91.2% sensitivity and 90.6% specificity provides the best set of results (see Table 6.2).

A non-filtering method using LBPs [122] was also investigated for novelty detection. The LBP coefficients were extracted from each RGB colour band. The estimation of the range of coefficient distributions for normal samples and the novelty detection procedures were the same as that described in Section 6.2.2. Results showed that LBP performed very poorly, although a more sophisticated classifier may improve the performance.

Some novelty detection results on those image collages are shown in Figure 6.9. Collage images are shown in the first column, followed by results of Escofet's method in the second column. The third column presents the novelty detection results using graylevel texem analysis in image RGB colour channels. Proposed colour space factorisation using principal component analysis, as described earlier in Chapter 5, achieved better results as shown in the fourth column. The full 3D model of the colour texems, which takes into account interspectral and intraspectral interactions simultaneously, gave the best results as illustrated in the last column.

Figure 6.10 shows a plot of the results for all these four methods. The filtering based approach consistently performed poorer than the colour texems with one exception in test number 14, in which the result was marginally better than that of colour texems. The foreground texture in this collage has a relatively larger chromato-textural difference from the background. All four methods achieved very good accuracy on this collage, above 90%. There were three occasions when graylevel texems in separate channels performed slightly better than colour texems, for collages 7, 16, and 21. In each of these cases a reasonably good channel separation was obtained leading to good discrimination. On average, colour texems performed 19.0% better in overall accuracy than Escofet's method, 11.8% better than graylevel texems in RGB channels, and 6.2% better than graylevel texems in image eigenchannels.

There are two important parameters in the texem model, the size of texems and the number of texems. In theory, the size of the texems is arbitrary, thus, it can easily cover all the necessary spatial frequency range. However, for the sake of computational simplicity, a window size of 5×5 or 7×7 across all scales generally suffices. The number of texems can be automatically determined using model order selection methods, such as Minimum Description Length (MDL) [140], though they are usually computationally expensive. Generally, images with more chromato-textural variations need more texems to represent them. In application to defect detection on ceramic tiles, the random patterns differ from one family to another in terms of chromatic and textural properties. In order to cope with the variations, we used a slightly larger number of texems to generalise the textures. Consistent performance with little variation was found when using 12 to 16 texems. Naturally, the more the number of texems, the higher the computational costs. Thus, in this application 12 texems in each scale was used for over 1000 tile images and collages.

6.5 Alternative Applications

Two other applications of texems are described briefly in this section to illustrate their versatility in dealing with varied problems.

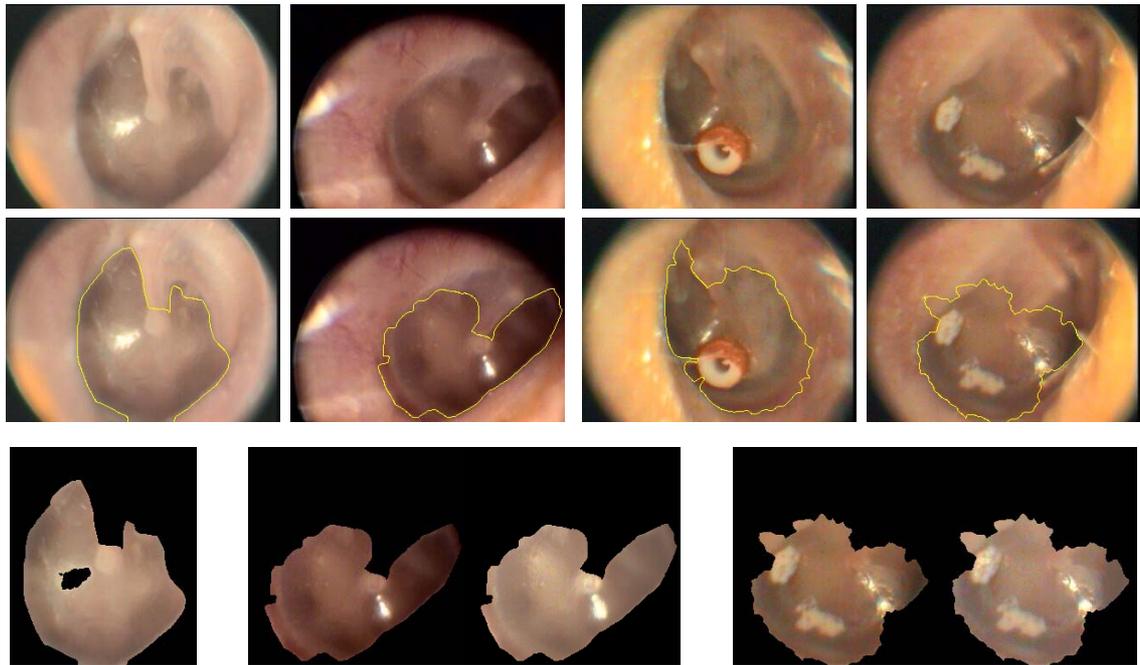


Figure 6.11: Snake segmentation and colour normalisation - first row: Two normal and two abnormal TM images; second row: Snake segmentation results on TM images to localise TM regions; third row: Colour normalisation - from left to right: reference TM without highlights, pre and post-normalisation of a normal and a defective TM.

In section 6.5.1, an application of the colour texem model to finding abnormalities in medical images is demonstrated. These images are significantly different from tile images. Particularly, the illumination condition is very difficult to control and there is quite a lot interference from regions of no interest.

Texems are essentially image representations. It is possible to use them for other purposes than defect detection, for example image segmentation. As each texem is a representation of a group of image patches from the image, it can be treated as the centre of cluster for those image patches and perform segmentation. In section 6.5.2, the colour texems are used to segment microscopic images of ink jet printed tiles for colour dot separation and modelling. As multiscale texems are involved, interscale fusion is used to refine the segmentation.

6.5.1 Detecting defects in tympanic membrane images

Here, a method to detect abnormalities in colour tympanic membrane images using colour texems is briefly presented. The tympanic membrane (TM), also commonly known as the eardrum, is a thin membrane that divides the ear canal from the middle ear. The TM vibrates in response to

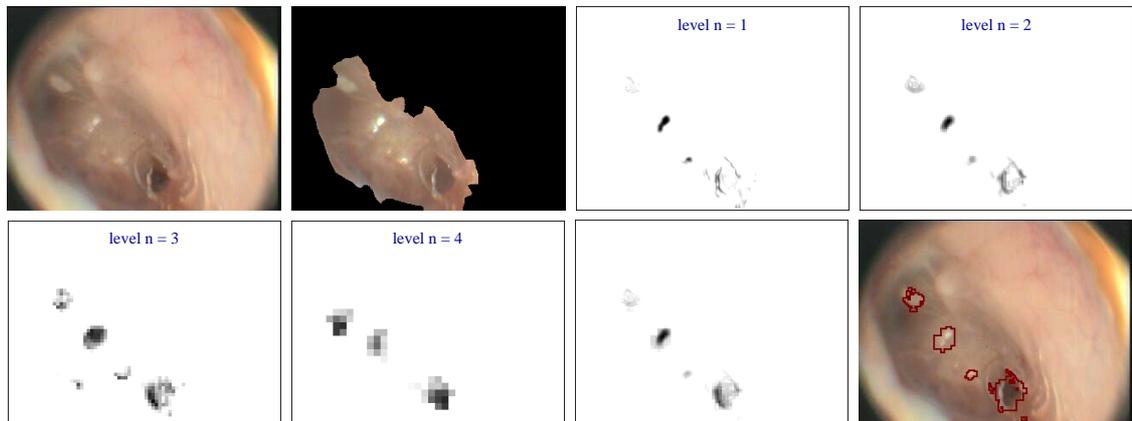


Figure 6.12: Defect localisation - from top left: initial TM image, after segmentation and colour normalisation, defect probability maps at all 4 scales, fused defect map, and detected defects (and saturated highlights).

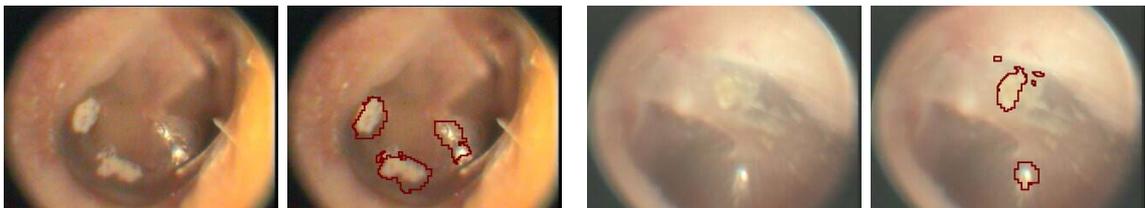


Figure 6.13: Defect localisation examples.

sound, and enables it to be passed along the middle ear bones, through the inner ear and up to the brain. Abnormalities of the TM include perforations, retractions, otitis media (fluid build up behind the TM) and cholesteatoma (infection within the middle ear cleft). These have various consequences for patients if left untreated, including hearing loss and major infection, and so need to be detected early, particularly in children. Examples of normal and abnormal TM are shown in the first row of Figure 6.11 (note the saturated highlight regions originating from the illumination source of the otoscope).

Due to the high variations of skin colour, ear canal geometry, and illumination conditions, the appearance of the tympanic membrane and its surrounding regions are significantly different from one patient to another. It is then important to localise the regions of interests and to perform some pre-processing before actually finding the defects. Thus, a deformable contour model (snake) is used to firstly localise the membranes, as a snake can naturally adapt to shape variations. The generalised version [194] of the well-known gradient vector flow (GVF) snake is adopted to segment regions of interest. However, implicit representation of the generalised GVF (GGVF) using level set techniques [148] is used so that the snake can handle topological changes and represent shapes more accurately. The second row in Figure 6.11 shows some snake localisation results.

The snake segmented TM regions are then processed to minimise colour variations. A reference TM image is selected from the training samples for colour normalisation based on histogram specification, which was also used for luminance correction for ceramic tiles in Chapter 3. Two colour normalisation examples, one for a normal and one for an abnormal example, are shown in the last row of Figure 6.11.

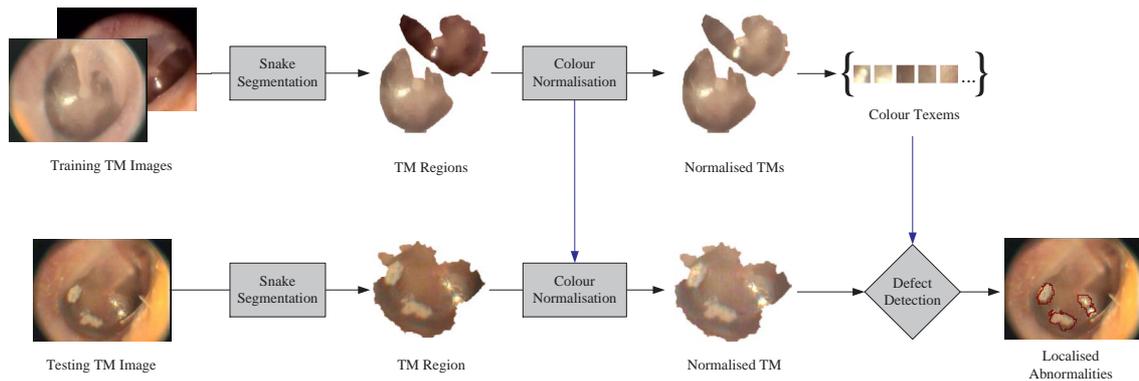


Figure 6.14: Flow chart of the Tympanic Membrane abnormality detection.

Next, colour texems are learnt to represent the surface properties of defect-free TM regions. The same procedure as that described in Section 6.2 is again applied to detect and localise defective regions in unseen TM images. Briefly, the TM images under inspection are also layered into the same multiscale framework and image patches are examined against the learnt texems. Regions with data likelihoods lower than the automatically derived thresholds are assigned as potential defects. Logical combination rules are then applied to fuse the defects, if any, across the scales to produce the final defect regions (see Figure 6.12 for an example). Some experimental results are shown in Figure 6.13. Figure 6.14 illustrates the overall learning and detection procedures.

6.5.2 Ink-dot image segmentation

The tile industry has recently seen advances in ink-jet printing of tiles. This printing technique is faster than traditional methods and more importantly, it can produce very complex patterns. It is important for the tile industry to understand printing and ink characteristics in ink jet printing in order to faithfully reproduce colours and patterns, i.e. to predict the colour appearance based on ink measurements. A full understanding of the relationship can be attempted using colour prediction models. As part of the process, this involves segmenting the ink jet printed tile images of halftone designs, where the images are captured under microscope to achieve the resolution necessary to perceive individual ink dots. Here, the colour texems are used to perform the segmentation task.

It can be recalled from Section 6.2.1 that each image patch from an image has a measurable

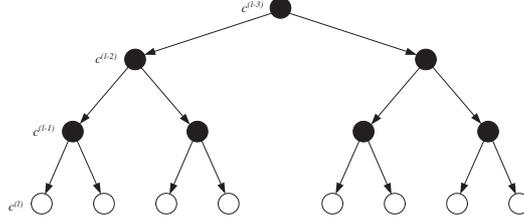


Figure 6.15: One-dimensional analogy of the quadtree structure (empty circles denote leaf nodes).

relationship with each texem according to the posterior, $p(\mathbf{m}_k | \mathbf{Z}_i, \Theta)$, which can be conveniently obtained using Bayes rule as shown in (6.15). Thus, every texem can be viewed as an individual textural class component, and the posterior can be regarded as component likelihood. Each pixel in the image then can be labelled with texems using the component likelihood. However, the partitions at each scale have to be combined across scales to find the final segmentation. Next, the segmentation procedure and interscale fusion using a simple quadtree structure are briefly described.

A Partition and interscale fusion

The ink dot image is first laid out in the multiscale framework as described in section 6.2.1. The colour texems are then obtained by applying the EM based parameter estimation. In order to perform image segmentation, each pixel needs to be assigned a class label, $c = \{1, 2, \dots, K\}$. So, for each scale n , there is a random field of class labels, $C^{(n)}$. The probability of a particular image patch, $\mathbf{Z}_i^{(n)}$, belonging to a texem (class), $c = k, \mathbf{m}_k^{(n)}$, is determined by the posterior probability, $p(c = k, \mathbf{m}_k^{(n)} | \mathbf{Z}_i^{(n)}, \Theta^{(n)})$ simplified as $p(c^{(n)} | \mathbf{Z}_i^{(n)})$, given by:

$$p(c^{(n)} | \mathbf{Z}_i^{(n)}) = \frac{p(\mathbf{Z}_i^{(n)} | \mathbf{m}_k^{(n)}) \alpha_k^{(n)}}{\sum_{k=1}^K p(\mathbf{Z}_i^{(n)} | \mathbf{m}_k^{(n)}) \alpha_k^{(n)}}. \quad (6.22)$$

The class probability at given pixel location $(x^{(n)}, y^{(n)})$ at scale n then can be estimated according to

$$p(c^{(n)} | (x^{(n)}, y^{(n)})) = p(c^{(n)} | \mathbf{Z}_i^{(n)}). \quad (6.23)$$

Note different patch sizes can be used across the scales. Thus, this labelling assignment procedure initially partitions the image in each individual scale. As the image is laid hierarchically, there is inherited relationship among parent pixel and children pixels. Their labels should also reflect this relationship. Next, based on initial labelling, the partitions across all the scales are fused together to produce the final segmentation map.

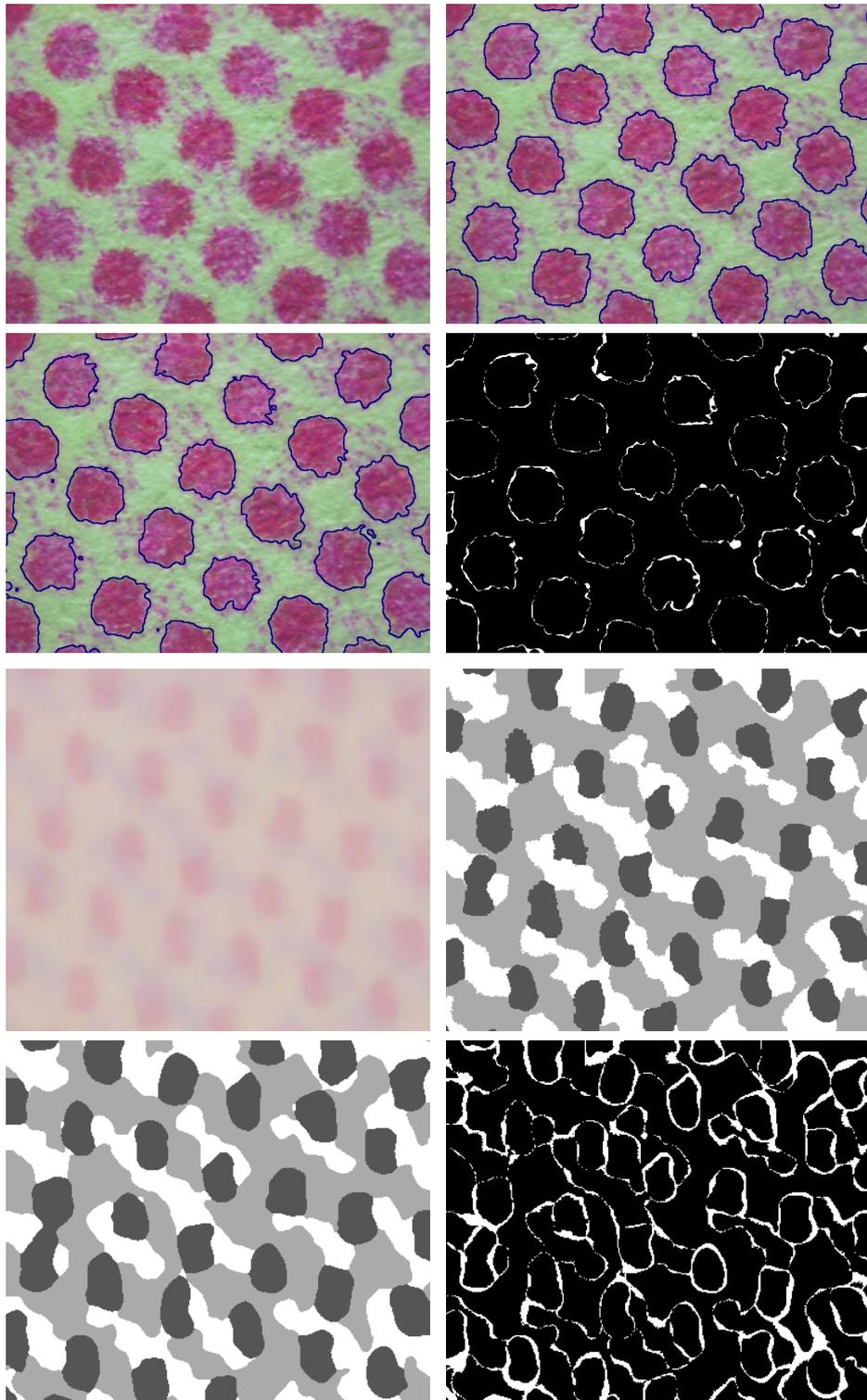


Figure 6.16: Unsupervised ink dot image segmentation - first row: original microscopic ink dot image with one ink pattern and the hand-labelled result; second row: the colour texem based segmentation result and the error map; third row: original microscopic ink dot image with two ink pattern and the hand-labelled result; fourth row: the colour texem based segmentation result and the error map.

Table 6.3: Testing results of ink dot image segmentation accuracy (values are %s)

Image No.	Ink type 1	Ink type 2	Background	Overall
1	98.2	86.0	83.9	87.8
2	96.8	-	90.2	94.0
3	82.1	90.4	86.5	85.8
4	86.0	-	93.6	90.6
5	94.2	-	93.1	93.6
6	88.7	86.5	85.9	87.5
7	97.7	-	95.6	96.9
Overall	-	-	-	90.9

The class labels $c^{(n)}$ are assumed conditionally independent given the labelling in the coarser scale $c^{(n+1)}$. Thus, each label field $C^{(n)}$ is assumed only dependent on the previous coarser scale label field $C^{(n+1)}$. This structure offers efficient computational processing, yet the desirable capability to capture complex spatial dependencies in the segmentation. The label field $C^{(n)}$ becomes a Markov chain structure in the scale variable n , which can be expressed as:

$$p(c^{(n)}|c^{(>n)}) = p(c^{(n)}|c^{(n+1)}), \quad (6.24)$$

where $c^{(>n)} = \{c^{(i)}\}_{i=n+1}^l$ are the class labels at all coarser scales than the n th, and $p(c^{(l)}|c^{(l+1)}) = p(c^{(l)})$ as l is the coarsest scale. The coarsest scale segmentation is directly based on the initial labelling.

A quadtree structure for the multiscale label fields is assumed, and $c^{(l)}$ only contains a single pixel. Figure 6.15 demonstrates the quadtree structure. A more sophisticated context model can be used to achieve better interaction between child and parent nodes, e.g. a pyramid graph model [28]. However, for simplicity and efficiency, the quadtree structure is adopted to model the label structure. The transition probability $p(c^{(n)}|c^{(n+1)})$ can be efficiently calculated using a lookup table. The label assignment at each scale are then updated, from coarsest level to the finest level, according to the joint probability of the data probability and the transition probability:

$$\begin{cases} \hat{c}^{(l)} = \arg \max_{c^{(l)}} \log p(c^{(l)}|(x^{(l)}, y^{(l)})), \\ \hat{c}^{(n)} = \arg \max_{c^{(n)}} \{\log p(c^{(n)}|(x^{(n)}, y^{(n)})) + \log p(c^{(n)}|c^{(n+1)})\} \quad \forall n < l. \end{cases} \quad (6.25)$$

Thus, the image segmentation is refined from the coarsest level down to the finest level with due consideration to both neighbouring and parent pixels. The segmented regions will be smoother and small isolated holes will be filled in.

B Experimental Results

The method was tested on microscopic ink dot images with different halftone patterns, and compared against hand-labelled results. Figure 6.16 shows two example ink dot images printed on tile surfaces. One is a single ink dot image and the other contains two types of ink dots. It was segmented into 2 and 3 classes respectively, each ink-printed region and background ink. The colour texems used were 5×5 , and were learnt from 4 scales for each inkdot image. Each texem represents a cluster of ink patches (or background patches). The image is thus labelled and later refined using the quadtree structure context model. The results are shown in Figure 6.16, which also shows the hand-labelled segmentation and the error maps (the difference between automatic segmentation and manual segmentation).

Table 6.3 shows the statistical results of the automatic segmentation measured based on manual labelling. The test images contain single ink and double ink printed tile surfaces. An overall accuracy of 90.9% was achieved.

6.6 Summary

In this chapter, a different inference procedure for the learning model is introduced to deal with colour texems so that it can be generally applied to both scalar and vector-valued data. A mixture model was used to learn these texems. The method required only a few defect free samples for unsupervised training to detect defects in random colour textures. Multiscale analysis was also used to reduce the computational costs and to localise the defects more accurately. It was evaluated on both synthetic image collages and a large number of tile images with various types of physical, chromatic, and textural defects. The comparative study showed texem based local contextual analysis significantly outperformed a filter bank method and the LBP based texture features in novelty detection. Also, it revealed that incorporating interspectral information was beneficial, particularly when defects were chromatic in nature. The ceramic tile test data was collected from several different sources and had different chromato-textural characteristics. This showed that the proposed work was robust to variations arising from the sources. However, better accuracy comes at a price. The colour texems can be 10 times slower than the grayscale texems at the learning stage. They were also much slower than the Gabor filtering based method but had fewer parameters to tune. The computational cost, however, can be drastically reduced by performing window-based, instead of pixel based, examination at the training and testing stages. Also, there are methods available, such as [52], to compute the Gaussian function, which is a major part of the computation, much more efficiently.

Alternative applications of the colour texem model were also presented. The first one was to detect abnormalities in colour images of tympanic membranes. The process started with region-of-interest segmentation using an active contour model, a geometric GGVF snake. The GGVF snake has been used by many researchers to for segmentation, however, it was implemented in a geometric snake framework using the level set method to gain topological freedom (more details of this can be found in [187]). The extracted membrane regions were then passed to texems for colour texture analysis in order to find defects, if any. The second application was to segment ink jet printed microscopic tile images. As a mixture model was used to derive the colour texems, it was natural to classify image patches based on posterior probabilities. Thus, an initial segmentation of the image in multiscale was obtained by directly using the posteriors. In order to fuse the segmentation from different scales together, the quadtree context model was used to interpolate the label structure, from which the transition probability was derived. Thus, the final segmentation was obtained by top-down interscale fusion. The results were compared against manual labelling to assess the segmentation accuracy. The method may be directly used for colour texture segmentation. However, it can be improved in several ways. Firstly, a more complex context model can be used instead of the quadtree structure in which each pixel label only depends on one class label. Secondly, some natural textures may contain very complex scenes. A texture region may contain multiple textural elements. In other words, multiple texems may be needed to describe a single multi-modal textural region. Thus, texem grouping may be necessary to segment more complex images, e.g. natural scenes, where multi-modal textures often appear.

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The aim of this thesis was to develop texture analysis techniques for defect detection on colour random texture surfaces, in particular ceramic tile surfaces, on which patterns are usually richly decorated and exhibit complex random appearance. The image datasets originated from different sources within the MONOTONE project. The thesis covers the issues of data preparation and image pre-processing, colour tonality inspection, and local chromato-textural defect detection.

Image preprocessing for luminance compensation was necessary for stringent chromato-textural inspection. For each image, a 2D illumination field was generated and inspected for uniformity for both radial and line-scan cameras. However, both dimensions of the illumination field for radial camera were in the spatial domain, whereas the second dimension for that of a line-scan camera was in the temporal domain. For images captured by the radial camera, representative patches were used to correct the image using histogram specification. Parametric cosine-fourth based luminance and nonparametric profile based methods were investigated for luminance correction for the line-scan camera. Experimental results suggested that the profile based approach not only performed faster but also achieved better results, in particular when there were local irregularities that parametric models found difficult to explain and approximate. Other nonlinear polynomial fittings could be used instead of the cosine fourth function, yet it would still rely on a hypothesised model and parameterisation. The median profile based luminance correction turned out to be a simple but effective and reliable method in this application.

Colour tonality issues are perceived as global appearance abnormalities. They are generally concerned with consistency from surface to surface. In the ceramic tile industry, changes in tonality during production is usually gradual and subtle which make it very difficult for prompt human

judgement. Recent research in colour tonality inspection has suggested that using global measurements, such as statistics from the colour histogram, is highly appropriate and can achieve good results, e.g. [15, 16]. The inspection process thus was generally treated as a global colour distribution test. However, given the absence of spatial neighbourhood exploitation in histogram analysis, global statistics alone can struggle in discriminating subtle colour differences.

We proposed that local colour distribution was also important in characterising surface colour tonality. It was also revealed that chromatic noise could largely demolish the tonality differences when using histogram based methods. This noise effect was examined through a PCA process which showed that it was generally introduced by non-uniformity along the line-scan camera acquisition line and temporal inconsistency in the imaging system. Smoothing proved beneficial as it improved the overall performance compared to direct colour tonality comparison. A localised feature vector was then derived by computing the local common vector and its related statistics resulting in a higher dimensional colour tonality feature space. Again, PCA was used to generate a reference eigenspace, in which defect-free tonality was well represented and more reliable chromatic features were obtained. PCA analysis of reconstructed defect-free and defective samples based on the reference eigenspace showed that the reconstructed images were much less affected by noise interference. Improved performance was achieved when using the proposed eigenspace feature vector. The training was based on defect-free samples only, and the testing images were examined with a simple parametric classifier.

There is a considerable body of work on detecting abnormalities in textured images of various types of material surfaces, such as textile and wood. The methods employed in these studies range from statistical, through structural, and signal processing, to model based approaches. However, there is still relatively limited research in defect detection on random textures, particularly random colour textures.

Texture decorations on ceramic tile surfaces usually exhibit complex patterns with random appearance. Filter bank based approaches, as applied in [20, 68, 49, 77, 82, 163, 12, 113, 93], can be adopted to detect defects in such random textures using discriminate analysis based on filter responses. However, difficulties arise when processing 3D data, e.g. colour images. Filtering generally requires factorisation of the 3D space.

Several works have suggested that image patch based local contextual analysis is a promising alternative approach to random texture analysis, such as [70, 177]. They have shown that textures with global structures can be discriminated based on distribution of local measurement. In fact, filtering responses too are generally based on image pixels and their local neighbourhoods which indeed supports the idea of analysing textures based on local spatial interactions. However, works such as [177] proved that analysing textures directly based on local context could perform better without dedicated filter bank design and related parameter tuning.

This work condones local contextual texture analysis and proposed a simple two-layer generative model, the texem model, to represent images of random and regular textures. An image was considered as a superposition of a collection of texems, primitive representations encapsulated in image patches at various sizes. Each texem was described by a matrix containing mean pixel values and corresponding variance that determines the variation at each pixel position in the texem. The texems had to be in various sizes in order to model sufficient spatial interactions. This also alleviated the difficulties in finding the optimum patch size. Alternatively, multiscale fixed-size texems were used to save computational costs.

Texems are implicit texture representations and remove the need to explicitly defining and extracting texture primitives. Each texem encapsulates partial or multiple primitives and is a compact yet comprehensive representation.

In application to defect detection, it seems intuitive that image patches extracted from images within the same texture family share the same texture primitives which govern the visual consistency for that product family. The learnt texems from given samples were then capable of describing the textural characteristics of the whole family, based upon which the inspection process took place by examining the same-source similarity. Thus, the learning process of the generative model was a bottom-up procedure. A few given defect-free samples were split into various size patches which were then condensed into a small number of primitive representations. A multiscale approach was adopted for computational efficiency.

In this study, unsupervised novelty detection was used instead of conventional supervised classification. In the training stage, the lower bounds for the defect-free samples at each scale were automatically derived based on data likelihoods, measured according to the learnt texems. Then, in the testing stage, the data likelihoods at each scale were measured and compared with those of training samples, resulting in multiscale potential defect maps which were later combined into the final defect map using logical processes based on the assumption that true defects should appear in two consecutive scales. Thus, the defect detection task was accomplished based on only defect-free samples without dedicated collections of defective samples.

The learning of the texems can be carried out in various forms dependent on data structure and computational requirement. For graylevel images, the Gaussian Mixture Model was used to abstract the texems from given images. Image patches were projected into a higher dimensional space, in which EM based mixture modelling was performed. Each Gaussian component was treated as a texem representation. The mixture coefficient then can be generally viewed as a measurement of the number of image patches associated to a particular texem.

The extension of the texem model to colour texture analysis was performed via two different approaches. In the first, the colour space was factorised into three channels, from which a reference

eigenspace was established from defect-free samples, a similar approach to that involved in colour shade inspection. Thus, images were decomposed into uncorrelated channels to minimise the loss of information due to data factorisation. Multiscale graylevel texem based defect detection was then carried out in each channel, and defects across scales and channels were fused together afterwards. In the second, a full 3D texem model was derived without data factorisation and patch vectorisation. Texems were also modelled by mixture of components. However, each component was modelled by a product of Gaussians, instead of a single Gaussian, based on the assumption of conditional independence among neighbourhood pixels. Similar multiscale and novelty detection processes were used for defect detection.

The proposed local contextual based defect detection methods were tested on datasets that came from various sources, each of which had its own chromato-textural properties. Good classification accuracy was achieved on these datasets. This demonstrated the robustness of the proposed methods towards variations in different datasets.

In order to evaluate the spatial accuracy of the proposed defect detection methods, a collection of VisTex image collages was generated for novelty detection tests. Experimental results showed that uncorrelated data decomposition was better than direct correlated channel separation. Moreover, handling inter-spectral and intra-spectral interactions simultaneously through the full 3D texem model further improved the overall performance, but at some computational cost. The proposed methods also performed better when compared with a Gabor filter bank based method.

The concept of texems was examined on other applications. An application to abnormality detection in colour eardrum images provided promising results. The method started by segmenting the region of interest using a geometric active contour, followed by colour texem analysis. Texems can also be used for image segmentation and this was demonstrated by segmenting microscopic ink dot images. Each texem was considered as the centre of a cluster of image patches. Thus, the ink dot image was labelled in multiscale based on corresponding colour texems. A quadtree structure was adopted to model the interscale relationship, based on which segmentations in individual scales were fused together to generate the final segmentation map. These alternative applications demonstrate the potential of the texem model in solving texture-related problems.

7.2 Contributions

The contributions of this thesis are summarised as follows:

1. *Colour Tonality Inspection.* A multidimensional histogram method to inspect colour tonality

on colour textured surfaces was proposed. Vector directional processing and PCA based feature extraction were performed to discriminate subtle colour tonality changes.

2. *The Texem Model and its Application to Defect Detection.* A novel two layer generative texem model was proposed to represent images. An efficient graylevel texem learning scheme was developed and used for defect detection. The method was implemented in a novelty detection framework using only defect-free samples.
3. *Texem Analysis in Image Eigenchannels.* An efficient extension of the graylevel texems to defect detection on colour images was developed. A PCA based decomposition scheme was used to obtain decorrelated image channels in which graylevel texem analysis was performed.
4. *Colour Texems.* The texem model was further extended into a full 3D model in which inter-spectral and intra-spectral relationships were simultaneously modelled. This extension resolved the issues related to data factorisation and image patch vectorisation.
5. *Minor Contributions:* (a) Efficient and effective luminance correction methods for both radial and line scan cameras were developed. (b) A medical application of the colour texem model was presented, i.e. detecting abnormalities in tympanic membrane images. (c) An unsupervised image segmentation method was proposed to segment colour ink dot images.

7.3 Future Work

The number of texems at the learning stage was selected empirically in this study. Experimental results showed that 12 texems was about the right number needed for the ceramic tile textures. However, it is possible to automatically determine the number of texems by applying model order selection criteria, e.g. MDL, as part of future work.

Although the main objective of this thesis was not to obtain real-time performance techniques, the computational costs of the proposed methods are relatively high, particularly the colour texem model. As mentioned earlier in Chapter 6, texem generation can be accelerated drastically by examining the test image at lower spatial sampling. Other techniques, such as fast Gaussian computations, e.g. as in [52], are worth investigating to speed up the training and testing.

In Chapter 6, we showed texems could be applied to image segmentation. However, for some natural colour images, a textural region may contain multiple visual elements and display complex patterns. A single texem might not be able to fully represent such a textural region. In this case, texems can be grouped together to represent a “multimodal” texture region. Here, we further

emphasise the potential of texems for image segmentation by briefly demonstrating a possible approach using a simple but effective grouping method proposed by Manduchi [105, 106]. A more elaborate treatment of this approach is worth investigating as part of future work.

The basic strategy is to group some of the texems based on their spatial coherence. The grouping process takes the form:

$$\begin{aligned}\hat{p}(\mathbf{Z}_i|c) &= \frac{1}{\hat{\beta}_c} \sum_{k \in G_c} p(\mathbf{Z}_i|\mathbf{m}_k)\alpha_k, \\ \hat{\beta}_c &= \sum_{k \in G_c} \alpha_k,\end{aligned}\tag{7.1}$$

where G_c is the group of texems that are combined together to form a new cluster c which labels the different texture classes, and $\hat{\beta}_c$ is the new prior for cluster c . The mixture model thus can be reformulated as:

$$p(\mathbf{Z}_i) = \sum_{c=1}^{\hat{K}} \hat{p}(\mathbf{Z}_i|\mathbf{m}_k)\hat{\beta}(c),\tag{7.2}$$

where \hat{K} is the number of texture regions which can be automatically determined using model order selection methods. Equation (7.2) shows that pixel i in the centre of patch \mathbf{Z}_i will be assigned to the texture class c which maximises $\hat{p}(\mathbf{Z}_i|c)\hat{\beta}(c)$, i.e.

$$\begin{aligned}c &= \arg \max_c \hat{p}(\mathbf{Z}_i|c)\hat{\beta}(c) \\ &= \arg \max_c \sum_{k \in G_c} p(\mathbf{Z}_i|\mathbf{m}_k)\alpha_k.\end{aligned}\tag{7.3}$$

The grouping as given in (7.2) is carried out based on the assumption that the posterior probabilities of grouped texems are typically spatially correlated. The spatial coherency of two texem components is measured based on the Maximum Description Criteria (MDC) [105, 106]. For two original mixture models, the descriptiveness D is defined as:

$$\begin{aligned}D &= \sum_{j=1}^K D_j, \\ D_j &= \int p(\mathbf{Z}_i|\mathbf{m}_j)p(\mathbf{m}_j|\mathbf{Z}_i)d\mathbf{Z}_i = \frac{E[p(\mathbf{m}_j|\mathbf{Z}_i)^2]}{\alpha_j},\end{aligned}\tag{7.4}$$

where $E[.]$ is the expectation computed with respect to $p(\mathbf{Z}_i)$. The descriptiveness decreases drastically when well separated texem components are grouped together, but decreases very slowly when spatially correlated texem component distributions merge together. Thus, the texem group-

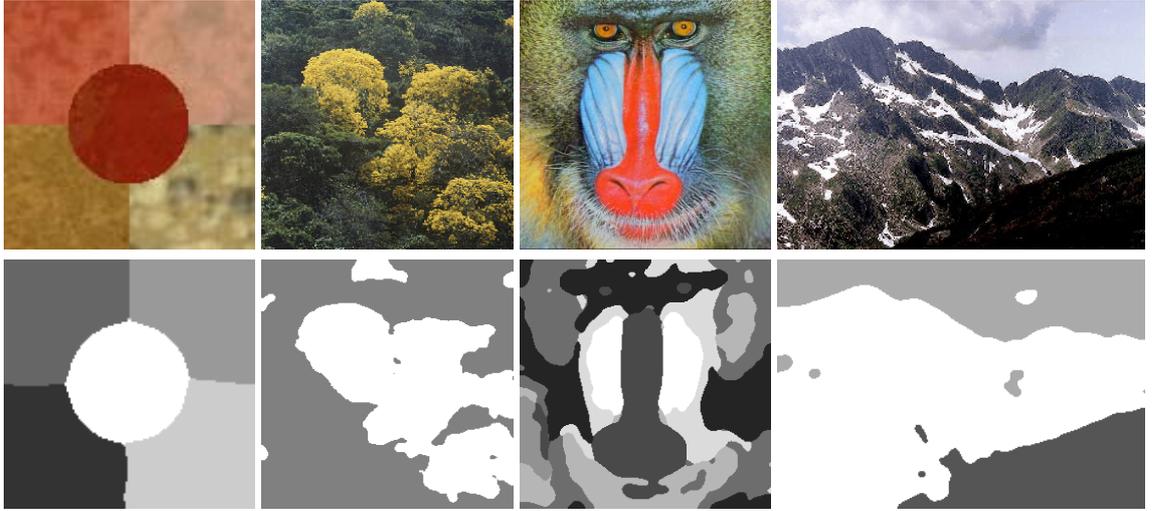


Figure 7.1: Unsupervised image segmentation.

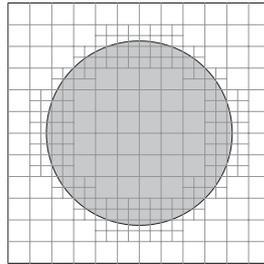


Figure 7.2: The patch size on the region boundary is preferred to be smaller such that only one type of texture is included.

ing should search for smallest change in descriptiveness, ΔD . This strategy is known as MDC. The algorithm proposed in [105] can be followed to greedily group two texem components, \mathbf{m}_j and \mathbf{m}_k , at a time with minimum ΔD_{jk} :

$$\Delta D_{jk} = \frac{\alpha_k D_j + \alpha_j D_k}{\alpha_j + \alpha_k} - \frac{2E[p(\mathbf{m}_j|\mathbf{Z}_i)p(\mathbf{m}_k|\mathbf{Z}_i)]}{\alpha_j + \alpha_k}. \quad (7.5)$$

We can see that the first term in (7.5) is the maximum possible descriptiveness loss when grouping two texems, and the second term in (7.5) is the normalised cross correlation between the two texem component distributions. However, since one texture region may contain different texem components that are significantly different to each other, it is beneficial to smooth the posterior probabilities as proposed in [106] such that a pixel that originally has high probability to just one texem component will be softly assigned to a number of components that belong to the same “multimodal” texture. After grouping, the segmentation is performed according to (7.2). A multi-scale approach as presented in Chapter 6 can be used. Figure 7.1 shows some initial segmentation

results.

The assumption of spatial coherency may not be always valid which will lead to dissimilar texems being grouped together. As part of future work, we shall investigate additional constraints towards texem grouping. The work can be further extended based on the fact that for segmentation purposes the patch size on the region boundary is preferred to be smaller so that only one type of texture is included in order to obtain higher spatial accuracy. A dynamic patch generating scheme based on a split and merge procedure before texem learning could be used as illustrated in Figure 7.2.

Bibliography

- [1] F. Ade, "Characterization of texture by 'eigenfilter'," *Signal Processing*, vol. 5, no. 5, pp. 451–457, 1983.
- [2] F. Ade, N. Lins, and M. Unser, "Comparison of various filter sets for defect detection in textiles," in *International Conference on Pattern Recognition*, vol. 1, 1984, pp. 428–431.
- [3] M. Aggarwal and N. Ahuja, "On cosine-fourth and vignetting effects in real lenses," in *IEEE International Conference on Computer Vision*, vol. 1, 2001, pp. 472–479.
- [4] J. Ai, D. Liu, and X. Zhu, "Combination of wavelet analysis and color applied to automatic color grading of ceramic tiles," in *International Conference on Pattern Recognition*, vol. 3, 2004, pp. 23–26.
- [5] C. Allain and M. Cloitre, "Characterizing the lacunarity of random and deterministic fractal sets," *Physical Review*, vol. A-44, no. 6, pp. 3552–3558, 1991.
- [6] R. Baldrich, M. Vanrell, and J. Villanueva, "Texture and color features for tile classification," in *SPIE Polarization and Color Techniques in Industrial Inspection*, vol. 3826, 1999, pp. 124–135.
- [7] A. Baykut, A. Atalay, A. Ercil, and M. Güler, "Real-time defect inspection of textured surfaces," *Real-Time Imaging*, vol. 6, pp. 17–27, 2000.
- [8] J. Bennett and A. Khotanzad, "Multispectral random field models for synthesis and analysis of color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 327–332, 1998.
- [9] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of Royal Statistical Society*, vol. B-36, pp. 344–348, 1974.
- [10] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models," Tech. Rep., 1998, iCSI-TR-97-021, University of California, Berkeley.
- [11] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [12] A. Bodnarova, M. Bennamoun, and S. Latham, "Optimal Gabor filters for textile flaw detection," *Pattern Recognition*, vol. 35, pp. 2973–2991, 2002.

- [13] C. Boukouvalas, J. Kittler, R. Marik, and M. Petrou, "Automatic grading of ceramic tiles using machine vision," in *IEEE International Symposium on Industrial Electronics*, 1994, pp. 13–18.
- [14] C. Boukouvalas, J. Kittler, R. Marik, and M. Petrou, "Automatic color grading of ceramic tiles using machine vision," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 1, pp. 132–135, 1997.
- [15] C. Boukouvalas, J. Kittler, R. Marik, and M. Petrou, "Color grading of randomly textured ceramic tiles using color histograms," *IEEE Transactions on Industry Electronics*, vol. 46, no. 1, pp. 219–226, 1999.
- [16] C. Boukouvalas and M. Petrou, "Perceptual correction for colour grading of random textures," *Machine Vision and Applications*, vol. 12, pp. 129–136, 2000.
- [17] C. Boukouvalas, G. Vernazza, F. Natale, G. Toni, J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, P. Roy, and R. Salgari, "Automatic system for surface inspection and sorting of tiles," *Journal of Materials Processing Technology*, vol. 82, pp. 179–188, 1998.
- [18] R. Broadhurst, J. Stough, S. Pizer, and E. Chaney, "Histogram statistics of local image regions for object segmentation," in *International Workshop on Deep Structure, Singularities, and Computer Vision*, 2005.
- [19] T. Caelli and D. Reye, "On the classification of image regions by colour, texture and shape," *Pattern Recognition*, vol. 26, no. 4, pp. 461–470, 1993.
- [20] J. Campbell and F. Murtagh, "Automatic visual inspection of woven textiles using a two-stage defect detector," *Optical Engineering*, vol. 37, pp. 2536–2542, 1988.
- [21] C. Chan and G. Pang, "Fabric defect detection by Fourier analysis," *IEEE Transactions on Industry Applications*, vol. 36, no. 5, pp. 1267–1276, 2000.
- [22] K. Chang, K. Bowyer, and M. Sivagurunath, "Evaluation of texture segmentation algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 1999, pp. 294–299.
- [23] B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 72–77, 1995.
- [24] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition 2*, L. Kanak and A. Rosenfeld, Eds. Elsevier, 1985.
- [25] J. Chen and A. Jain, "A structural approach to identify defects in textured images," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, 1988, pp. 29–32.
- [26] P. Chen, T. Liang, H. Yau, W. Sun, N. Wang, H. Lin, and R. Lien, "Classifying textile faults with a back propagation neural network using power spectra," *Textile Research Journal*, vol. 68, pp. 121–126, 1998.

- [27] Y. Chen, M. Nixon, and D. Thomas, "Statistical geometrical features for texture classification," *Pattern Recognition*, vol. 28, no. 4, pp. 537–552, 1995.
- [28] H. Cheng and C. Bouman, "Multiscale bayesian segmentation using a trainable context model," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 511–525, 2001.
- [29] D. Chetverikov, "Structural defects: General approach and application to textile inspection," in *Proceedings of the 15th International Conference on Pattern Recognition*, 2000, pp. 1521–1524.
- [30] D. Chetverikov and A. Hanbury, "Finding defects in texture using regularity and local orientation," *Pattern Recognition*, vol. 35, no. 10, pp. 2165–2180, 2002.
- [31] C. Cho, B. Chung, and M. Park, "Development of real-time vision-based fabric inspection system," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 4, pp. 1073–1079, 2005.
- [32] M. Clark and A. Bovik, "Texture segmentation using Gabor modulation/demodulation," *Pattern Recognition Letters*, vol. 6, pp. 261–267, 1987.
- [33] J. Coggins and A. Jain, "A spatial filtering approach to texture analysis," *Pattern Recognition Letters*, vol. 3, pp. 195–203, 1985.
- [34] F. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 803–809, 1991.
- [35] R. Coifman, Y. Meyer, and V. Wickerhauser, "Size properties of wavelet packets," in *Wavelets and Their Applications*, M. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, Eds. Jones and Bartlett, 1992, pp. 453–470.
- [36] A. Conci and C. Proenca, "A fractal image analysis system for fabric inspection based on box-counting method," *Computational Networks*, vol. 30, pp. 1887–1895, 1998.
- [37] A. Conci and C. Proenca, "A system for real-time fabric inspection and industrial decision," in *14th International Conference on Software Engineering and Knowledge Engineering*, 2002, pp. 707–714.
- [38] R. Connors and C. Harlow, "Towards a structural textural analyzer based on statistical methods," *Computer Graphics and Image Processing*, vol. 12, pp. 224–256, 1980.
- [39] C. Coroyer, D. Declercq, and P. Duvaut, "Texture classification using third order correlation tools," in *IEEE Signal Processing Workshop on High-Order Statistics*, 1997, pp. 171–175.
- [40] C. Costa and M. Petrou, "Automatic registration of ceramic tiles for the purpose of fault detection," *Machine Vision and Applications*, vol. 11, pp. 225–230, 2000.
- [41] K. Dana, B. Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, 1999.
- [42] F. D'Astous and M. Jernigan, "Texture discrimination based on detailed measures of the power spectrum," in *International Conference on Pattern Recognition*, 1984, pp. 83–86.

- [43] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, pp. 847–856, 1980.
- [44] R. De Valois, D. Albrecht, and L. Thorell, "Spatial-frequency selectivity of cells in macaque visual cortex," *Vision Research*, vol. 22, pp. 545–559, 1982.
- [45] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series A (Statistical Methodology)*, vol. 39, pp. 1–39, 1997.
- [46] M. Dubuisson-Jolly and G. A., "Color and texture fusion: Application to aerial image segmentation and GIS updating," *Image and Vision Computing*, vol. 18, pp. 823–832, 2000.
- [47] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *IEEE International Conference on Computer Vision*, 1999, pp. 1033–1038.
- [48] J. Escofet, S. Millan, H. Abril, and E. Torrecilla, "Inspection of fabric resistance to abrasion by Fourier analysis," in *Proceedings of SPIE*, vol. 3490, 1998, pp. 207–210.
- [49] J. Escofet, R. Navarro, M. Millán, and J. Pladellorens, "Detection of local defects in textile webs using Gabor filters," *Optical Engineering*, vol. 37, no. 8, pp. 2297–2307, 1998.
- [50] K. Fu, *Syntactic Pattern Recognition and Applications*. Prentice-Hall, 1982.
- [51] J. Gangepain and C. Roques-Carmes, "Fractal approach to two dimensional and three dimensional surface roughness," *Wear*, vol. 109, pp. 119–126, 1986.
- [52] L. Greengard and J. Strain, "The fast Gauss transform," *SIAM Journal of Scientific Computing*, vol. 2, pp. 79–94, 1991.
- [53] P. Gunatilake, M. Siegel, A. Jordan, and G. Podnar, "Image understanding algorithms for remote visual inspection of aircraft surfaces," in *SPIE Machine Vision Applications in Industrial Inspection*, vol. 3029, 1997, pp. 2–13.
- [54] H. Habib, M. Yousaf, and M. Mohibullah, "Modified laws energy descriptor for inspection of ceramic tiles," in *National Conference on Emerging Technologies*, 2004, pp. 137–140.
- [55] M. Haindl, "Texture segmentation using recursive markov random field parameter estimation," in *Scandinavian Conference on Image Analysis*, vol. 2, 1999, pp. 771–776.
- [56] M. Haindl and V. Havlicek, "A simple multispectral multiresolution Markov texture model," in *International Workshop on Texture Analysis and Synthesis*, 2002, pp. 63–66.
- [57] J. Hammersley and P. Clifford, "Markov field on finite graphs and lattices," 1971, unpublished.
- [58] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [59] R. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

- [60] O. Hernandez and A. Knotanzad, "Color image segmentation using multispectral random field texture model & color content features," *Journal of Computer Science and Technology*, vol. 4, no. 3, pp. 141–146, 2004.
- [61] B. Horn, *Robot Vision*. The MIT Press, 1986.
- [62] Y. Huang and K. Chan, "Texture decomposition by harmonics extraction from higher order statistics," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 1–14, 2004.
- [63] J. Iivarinen, "Surface defect detection with histogram-based texture features," in *SPIE Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*, vol. 4197, 2000, pp. 140–145.
- [64] J. Iivarinen, J. Rauhamaa, and A. Visa, "Unsupervised segmentation of surface defects," in *International Conference on Pattern Recognition*, vol. 4, 1996, pp. 356–360.
- [65] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, pp. 1167–1186, 1991.
- [66] A. Jain and G. Healey, "A multiscale representation including opponent color features for texture recognition," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 124–128, 1998.
- [67] A. Jain, P. Robert, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.
- [68] W. Jasper, S. Garnier, and H. Potapalli, "Texture characterization and defect detection using adaptive wavelets," *Optical Engineering*, vol. 35, pp. 3140–3149, 1996.
- [69] X. Jin, S. Ong, and Jayasooriah, "A practical method for estimating fractal dimension," *Pattern Recognition Letters*, vol. 16, pp. 457–464, 1995.
- [70] N. Jojic, B. Frey, and A. Kannan, "Epitomic analysis of appearance and shape," in *IEEE International Conference on Computer Vision*, 2003, pp. 34–42.
- [71] I. Jolliffe, *Principal Component Analysis*. Springer-Verlag, 1986.
- [72] B. Julesz, "Textons, the element of texture perception and their interactions," *Nature*, vol. 290, pp. 91–97, 1981.
- [73] Z. Kato, T. Pong, and S. Qiang, "Unsupervised segmentation of color textured images using a multi-layer MRF model," in *IEEE International Conference on Image Processing*, 2003, pp. 961–964.
- [74] H. Kauppinen, "A two stage defect recognition method for parquet slab grading," in *15th International Conference on Pattern Recognition*, vol. 4, 2000, pp. 803–806.
- [75] H. Kauppinen, H. Rautio, and O. Silvén, "Non-segmenting defect detection and SOM based classification for surface inspection using color vision," in *SPIE Conference on Polarization and Color Techniques in Industrial Inspection*, vol. 3826, 1999, pp. 270–280.

- [76] J. Keller, S. Chen, and R. Crownover, "Texture description and segmentation through fractal geometry," *Computer Vision, Graphics, and Image Processing*, vol. 45, pp. 150–166, 1989.
- [77] S. Kim, M. Lee, and K. Woo, "Wavelet analysis to defect detection in weaving processes," in *Proceedings of IEEE International Symposium on Industrial Electronics*, vol. 3, 1999, pp. 1406–1409.
- [78] R. Kingslake, *Optical System Design*. Academic Press, 1983.
- [79] J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, and J. Song, "Detection of defects in colour texture surfaces," in *IAPR Machine Vision Applications*, 1994, pp. 558–567.
- [80] W. Krzanowski, *Principles of Multivariate Analysis: A User's Perspective*. Oxford University Press, 1988.
- [81] A. Kumar, "Neural network based detection of local textile defects," *Pattern Recognition*, vol. 36, pp. 1645–1659, 2003.
- [82] A. Kumar and G. Pang, "Fabric defect segmentation using multichannel blob detectors," *Optical Engineering*, vol. 39, no. 12, pp. 3176–3190, 2000.
- [83] A. Kumar and G. Pang, "Defect detection in textured materials using Gabor filters," *IEEE Transactions on Industry Applications*, vol. 38, no. 2, pp. 425–440, 2002.
- [84] A. Kumar and G. Pang, "Defect detection in textured materials using optimized filters," *IEEE Transactions on System, Man, and Cybernetics—Part B: Cybernetics*, vol. 32, no. 5, pp. 553–570, 2002.
- [85] L. Latif-Amet, A. Ertuzun, and A. Ercil, "An efficient method for texture defect detection: Subband domain co-occurrence matrices," *Image and Vision Computing*, vol. 18, no. 6-7, pp. 543–553, 2000.
- [86] M. Lee, C. Silva, E. Croft, and Q. Wu, "Machine vision system for curved surface inspection," *Machine Vision and Applications*, vol. 12, pp. 177–188, 2000.
- [87] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.
- [88] E. Levina, "Statistical issues in texture analysis," Ph.D. dissertation, University of California, Berkeley, 2002.
- [89] E. Levina and P. Bickel, "The Earth Mover's distance is the Mallows distance: Some insights from statistics," in *International Conference on Computer Vision*, 2001, pp. 251–256.
- [90] S. Li, *Markov Random Field Modeling in Image Analysis*. Springer, 2001.
- [91] S. Liapis, E. Sifakis, and G. Tziritas, "Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms," *Journal of Visual Communication and Image Representation*, vol. 15, no. 1, pp. 1–26, 2004.

- [92] T. Lindeberg, "Detecting salient blob-like image structures and scales with a scale-space primal sketch: A method for focus-of-attention," *International Journal of Computer Vision*, vol. 11, no. 3, pp. 283–318, 1993.
- [93] J. Liu and J. MacGregor, "Estimation and monitoring of product aesthetics: Application to manufacturing of "engineered stone" countertops," *Machine Vision and Applications*, pp. –, 2005.
- [94] 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*, vol. 1, 2001, pp. 145–150.
- [95] F. Lopez, J. Valiente, R. Baldrich, and M. Vanrell, "Fast surface grading using color statistics in the CIE Lab space," in *Iberian Conference on Pattern Recognition and Image Analysis (LNCS 3523)*, vol. 2, 2005, pp. 666–673.
- [96] F. Lopez, J. Valiente, and J. Prats, "Surface grading using soft colour-texture descriptors," in *Iberoamerican Congress on Pattern Recognition (LNCS 3773)*, 2005, pp. 13–23.
- [97] F. Lumbreras, J. Serrat, R. Baldrich, M. Vanrell, and J. Villanueva, "Color texture recognition through multiresolution features," in *International Conference on Quality Control by Artificial Vision*, vol. 1, 2001, pp. 114–121.
- [98] T. Mäenpää and M. Pietikäinen, "Texture analysis with local binary patterns," in *Handbook of Pattern Recognition and Computer Vision*, 3rd ed., C. Chen and P. Wang, Eds. World Scientific, 2005, pp. 197–216.
- [99] T. Mäenpää, M. Turtinen, and M. Pietikäinen, "Real-time surface inspection by texture," *Real-Time Imaging*, vol. 9, no. 5, pp. 289–296, 2003.
- [100] T. Mäenpää, J. Viertola, and M. Pietikäinen, "Optimizing color and texture features for real-time visual inspection," *Pattern Analysis and Applications*, vol. 6, no. 3, pp. 169–175, 2003.
- [101] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America, Series A*, vol. 7, pp. 923–932, 1990.
- [102] B. Mallick-Goswami and A. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Textile Research Journal*, vol. 70, pp. 758–762, 2000.
- [103] B. Mandelbrot, *The Fractal Geometry of Nature*. W.H. Freeman, 1983.
- [104] C. Mandriota, M. Nitti, N. Ancona, E. Stella, and A. Distanto, "Filter-based feature selection for rail defect detection," *Machine Vision and Applications*, vol. 15, pp. 179–185, 2004.
- [105] R. Manduchi, "Bayesian fusion of color and texture segmentations," in *IEEE International Conference on Computer Vision*, 1999, pp. 956–962.
- [106] R. Manduchi, "Mixture models and the segmentation of multimodal textures," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 98–104.

- [107] M. Markou and S. Singh, "Novelty detection: A review – part 1: Statistical approaches," *Signal Processing*, vol. 83, pp. 2481–2497, 2003.
- [108] M. Markou and S. Singh, "Novelty detection: A review – part 2: Neural network based approaches," *Signal Processing*, vol. 83, pp. 2499–2521, 2003.
- [109] D. Marr, "Early processing of visual information," *Philosophical Transactions of the Royal Society of London*, vol. B-275, pp. 483–524, 1976.
- [110] M. Mirmehdi, R. Marik, J. Kittler, and M. Petrou, "Structural defect detection in random macro textures," in *Signal Processing VIII, Theories and Applications*, 1996, pp. 1499–1502.
- [111] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 142–159, 2000.
- [112] MIT Media Lab, "VisTex texture database," 1995. [Online]. Available: <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>
- [113] A. Monadjemi, "Towards efficient texture classification and abnormality detection," Ph.D. dissertation, University of Bristol, UK, 2004.
- [114] A. Monadjemi, M. Mirmehdi, and B. Thomas, "Restructured eigenfilter matching for novelty detection in random textures," in *British Machine Vision Conference*, 2004, pp. 637–646.
- [115] A. Monadjemi, B. Thomas, and M. Mirmehdi, "Speed v. accuracy for high resolution colour texture classification," in *British Machine Vision Conference*, 2002, pp. 143–152.
- [116] MONOTONE, "European Commission MONOTONE project," 2002-2005. [Online]. Available: <http://www.monotone.upv.es/>
- [117] C. Neubauer, "Segmentation of defects in textile fabric," in *Proceedings of the 7th International Conference on Pattern Recognition*, vol. 1, 1992, pp. 688–691.
- [118] M. Niskanen, H. Kauppinen, and O. Silvén, "Real-time aspects of SOM-based visual surface inspection," in *Machine Vision Applications in Industrial Inspection*, 2002, pp. 123–134.
- [119] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," in *Proceedings of the 12th Scandinavian Conference on Image Analysis*, 2001, pp. 336–342.
- [120] P. Ohanian and R. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognition*, vol. 25, no. 8, pp. 819–833, 1992.
- [121] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distribution," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

- [122] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [123] S. Özdemir, A. Baykut, R. Meylani, A. Ercil, and A. Ertüzün, "Comparative evaluation of texture analysis algorithms for defect inspection of textile products," in *International Conference on Pattern Recognition*, 1998, pp. 1738–1740.
- [124] S. Özdemir and A. Ertüzün, "Markov random field and Karhunen-Loève transforms for defect inspection of textile products," in *IEEE Conference on Emerging Technologies and Factory Automation*, vol. 2, 1996, pp. 697–703.
- [125] C. Palm, "Color texture classification by integrative co-occurrence matrices," *Pattern Recognition*, vol. 37, no. 5, pp. 965–976, 2004.
- [126] D. Panjwani and G. Healey, "Markov random field models for unsupervised segmentation of textured color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 939–954, 1995.
- [127] G. Paschos, P. Kimon, and P. Valavanis, "A color texture based monitoring system for automated surveillance," *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, vol. 29, no. 1, pp. 298–307, 1999.
- [128] J. Penaranda, L. Briones, and J. Florez, "Colour machine vision system for process control in ceramic industry," in *SPIE New Image Processing Techniques and Applications: Algorithms, Methods, and Components II*, vol. 3101, 1997, pp. 182–192.
- [129] A. Pentland, "Fractal-based description of nature scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 661–674, 1984.
- [130] F. Pernkopf, "3D surface inspection using coupled HMMs," in *Proceedings of the International Conference on Pattern Recognition*, 2004, pp. 298–305.
- [131] F. Pernkopf, "Detection of surface defects on raw steel blocks using Bayesian network classifiers," *Pattern Analysis and Applications*, vol. 7, pp. 333–342, 2004.
- [132] R. Picard and T. Minka, "Vision texture for annotation," *Multimedia System*, vol. 3, pp. 3–14, 1995.
- [133] O. Pichler, A. Teuner, and B. Hosticka, "A comparison of texture feature extraction using adaptive Gabor filter, pyramidal and tree structured wavelet transforms," *Pattern Recognition*, vol. 29, no. 5, pp. 733–742, 1996.
- [134] M. Pietikainen, T. Maenpaa, and J. Viertola, "Color texture classification with color histograms and local binary patterns," in *International Workshop on Texture Analysis and Synthesis*, 2002, pp. 109–112.
- [135] W. Pratt, *Digital Image Processing*. Wiley, 1991.
- [136] T. Randen and J. Husøy, "Filtering for texture classification: a comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, 1999.

- [137] S. Ravandi and K. Toriumi, "Fourier transform analysis of plain weave fabric appearance," *Textile Research Journal*, vol. 65, pp. 676–683, 1995.
- [138] T. Reed and J. Buf, "A review of recent texture segmentation and feature extraction techniques," *Computer Vision, Image Processing and Graphics*, vol. 57, no. 3, pp. 359–372, 1993.
- [139] T. Reed and H. Wechsler, "Segmentation of textured images and gestalt organization using spatial/spatial-frequency representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1–12, 1990.
- [140] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.
- [141] A. Rosenfeld, C. Wang, and A. Wu, "Multispectral texture," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 12, no. 1, pp. 79–84, 1982.
- [142] Y. Ruber, C. Tomasi, and L. Guibas, "The Earth Mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.
- [143] H. Sari-Sarraf and J. Goddard, "Vision systems for on-loom fabric inspection," *IEEE Transactions on Industry Applications*, vol. 35, pp. 1252–1259, 1999.
- [144] N. Sarkar and B. Chaudhuri, "An efficient approach to estimate fractal dimension of textural images," *Pattern Recognition*, vol. 25, pp. 1035–1042, 1992.
- [145] M. Schael, "Texture defect detection using invariant textural features," in *DAGM Symposium*, 2001, pp. 17–24.
- [146] J. Scharcanski, "Stochastic texture analysis for monitoring stochastic processes in industry," *Pattern Recognition Letters*, vol. 26, pp. 1701–1709, 2005.
- [147] C. Schmid, "Constructing models for content-based image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 39–45.
- [148] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1996.
- [149] K. Shiranita, T. Miyajima, and R. Takiyama, "Determination of meat quality by texture analysis," *Pattern Recognition Letters*, vol. 19, pp. 1319–1324, 1998.
- [150] L. Siew, R. Hodgson, and E. Wood, "Texture measures for carpet wear assessment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 92–105, 1988.
- [151] O. Silvén, M. Niskanen, and H. Kauppinen, "Wood inspection with non-supervised clustering," *Machine Vision and Applications*, vol. 13, pp. 275–285, 2003.
- [152] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [153] M. Singh and S. Singh, "Spatial texture analysis: A comparative study," in *International Conference on Pattern Recognition*, vol. 1, 2002, pp. 676–679.

- [154] G. Smith and I. Burns, "Measuring texture classification algorithm," *Pattern Recognition Letters*, vol. 18, pp. 1495–1501, 1997.
- [155] K. Song, J. Kittler, and M. Petrou, "Defect detection in random colour textures," *Image and Vision Computing*, vol. 14, pp. 667–683, 1996.
- [156] C. Stauffer, "Learning a probabilistic similarity function for segmentation," in *IEEE Workshop on Perceptual Organization in Computer Vision*, 2004, pp. 50–58.
- [157] B. Super and A. Bovik, "Localizing measurement of image fractal dimension using Gabor filters," *Journal of Visual Communication and Image Representation*, vol. 2, pp. 114–128, 1991.
- [158] M. Swain and D. Ballard, "Indexing via color histograms," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1990.
- [159] B. Thai and G. Healey, "Modeling and classifying symmetries using a multiscale opponent color representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1224–1235, 1998.
- [160] P. Trahanias, D. Karakos, and A. Venetsanopoulos, "Directional processing of color images: Theory and experimental results," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 868–880, 1996.
- [161] P. Trahanias and A. Venetsanopoulos, "Vector directional filters - a new class of multichannel image processing filters," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 528–534, 1993.
- [162] D. Tsai and C. Heish, "Automated surface inspection for directional textures," *Image and Vision Computing*, vol. 18, no. 1, pp. 49–62, 1999.
- [163] D. Tsai and B. Hsiao, "Automatic surface inspection using wavelet reconstruction," *Pattern Recognition*, vol. 34, pp. 1285–1305, 2001.
- [164] D. Tsai and T. Huang, "Automated surface inspection for statistical textures," *Image and Vision Computing*, vol. 21, pp. 307–323, 2003.
- [165] D. Tsai, C. Lin, and K. Huang, "Defect detection in coloured texture surfaces using Gabor filters," *Imaging Science Journal*, vol. 53, no. 1, pp. 27–37, 2005.
- [166] D. Tsai and Y. Tsai, "Defect detection in textured surfaces using color ring-projection correlation," *Machine Vision and Applications*, vol. 13, pp. 194–200, 2003.
- [167] D. Tsai and S. Wu, "Automated surface inspection using Gabor filters," *The International Journal of Advanced Manufacturing Technology*, vol. 16, pp. 474–482, 2000.
- [168] I. Tsai, C. Lin, and J. Lin, "Applying an artificial neural network to pattern recognition," *Textile Research Journal*, vol. 65, pp. 123–130, 1995.
- [169] M. Tuceryan and A. Jain, "Texture segmentation using voronoi polygons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 211–216, 1990.

- [170] M. Tuceryan and A. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*. World Scientific, 1998, ch. 2, pp. 235–276.
- [171] M. Turner, "Texture discrimination by Gabor functions," *Biological Cybernetics*, vol. 55, pp. 71–82, 1986.
- [172] University of Bristol, "European Commission MONOTONE project University of Bristol website," 2004. [Online]. Available: www.cs.bris.ac.uk/Research/Digitalmedia/projects/monotone/monotone.htm
- [173] M. Unser and F. Ade, "Feature extraction and decision procedure for automated inspection of textured materials," *Pattern Recognition Letters*, vol. 2, no. 3, pp. 185–191, 1984.
- [174] L. Van Gool, P. Dewaele, and A. Oosterlinck, "Texture analysis," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 336–357, 1985.
- [175] M. Varma and A. Zisserman, "Classifying images of materials from images: Achieving viewpoint and illumination independence," in *European Conference on Computer Vision*, 2002, pp. 255–271.
- [176] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 691–698.
- [177] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*, vol. 61, no. 1/2, pp. 61–81, 2005.
- [178] F. Vilnrotter, R. Nevatia, and K. Price, "Structural analysis of natural textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 76–89, 1986.
- [179] H. Voorhees and T. Poggio, "Detecting textons and texture boundaries in natural images," in *International Conference on Computer Vision*, 1987, pp. 250–258.
- [180] R. Voss, "Random fractals: Characterization and measurement," in *Scaling Phenomena in Disordered Systems*, R. Pynn and A. Skjeltorp, Eds. Plenum, 1986.
- [181] H. Wechsler, "Texture analysis - a survey," *Signal Processing*, vol. 2, pp. 271–282, 1980.
- [182] W. Wen and A. Xia, "Verifying edges for visual inspection purposes," *Pattern Recognition Letters*, vol. 20, pp. 315–328, 1999.
- [183] R. Westra, J. Shippen, and N. Freear, "Printing quality control using template independent neurofuzzy defect classification," in *Proceedings of the 7th European Conference on Intelligent Techniques and Soft Computing*, 1999.
- [184] K. Wiltschi, A. Pinz, and T. Lindeberg, "An automatic assessment scheme for steel quality inspection," *Machine Vision and Applications*, vol. 12, pp. 113–128, 2000.
- [185] E. Wood, "Applying Fourier and associated transforms to pattern characterization in textiles," *Textile Research Journal*, vol. 60, pp. 212–220, 1990.
- [186] P. Xie and S. Guan, "A golden-template self-generating method for patterned wafer inspection," *Machine Vision and Applications*, vol. 11, pp. 225–230, 2000.

- [187] X. Xie and M. Mirmehdi, "Detecting abnormalities in tympanic membrane images," in *Medical Image Understanding and Analysis*, 2005, pp. 19–22.
- [188] X. Xie and M. Mirmehdi, "Localising surface defects in random colour textures using multiscale texem analysis in image eigenchannels," in *Proceedings of the 12th IEEE International Conference on Image Processing*, vol. 3, 2005, pp. 1124–1127.
- [189] X. Xie and M. Mirmehdi, "TEXEM: Texture exemplars for defect detection on random textured surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, submitted.
- [190] X. Xie and M. Mirmehdi, "Texture exemplars for defect detection on random textures," in *International Conference on Advances in Pattern Recognition*, 2005, pp. 404–413.
- [191] X. Xie, M. Mirmehdi, and B. Thomas, "Inspecting colour tonality on textured surfaces," in *International Conference on Image Analysis and Recognition*, 2004, pp. 810–817.
- [192] X. Xie, M. Mirmehdi, and B. Thomas, "Colour tonality inspection using eigenspace features," *Machine Vision and Applications*, 2006, in press.
- [193] B. Xu, "Identifying fabric structure with fast Fourier transform techniques," *Textile Research Journal*, vol. 66, pp. 496–506, 1996.
- [194] C. Xu and J. Prince, "Generalized gradient vector flow external forces for active contours," *Signal Processing*, vol. 71, no. 2, pp. 131–139, 1998.
- [195] X. Yang, G. Pang, and N. Yung, "Robust fabric defect detection and classification using multiple adaptive wavelets," *IEE Proceedings Vision, Image Processing*, vol. 152, no. 6, pp. 715–723, 2005.
- [196] X. Zhang and R. Bresee, "Fabric defect detection and classification using image analysis," *Textile Research Journal*, vol. 65, no. 1, pp. 1–9, 1995.
- [197] H. Zhou, A. Kassim, and S. Ranganath, "A fast algorithm for detecting die extrusion defects in IC packages," *Machine Vision and Applications*, vol. 11, pp. 37–41, 1998.
- [198] S. Zhu, C. Guo, Y. Wang, and Z. Xu, "What are textons?" *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 121–143, 2005.
- [199] S. Zhu, Y. Wu, and D. Mumford, "FRAME: Filters, random field and maximum entropy - towards a unified theory for texture modeling," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 1–20, 1997.
- [200] S. Zucker, "Toward a model of texture," *Computer Graphics and Image Processing*, vol. 5, pp. 190–202, 1976.

Appendix A

EM ALGORITHM FOR GRAYLEVEL TEXEMS USING GAUSSIAN MIXTURE MODEL

As before, let \mathbf{Z}_i be the i th image patch, $\mathbf{m}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\omega}_k\}$ the k th texem, S the patch pixel grid with size $d = N \times N$, P the number of image patches, and K the total number of texems. Here, each texem is characterised by a multivariate Gaussian distribution:

$$p(\mathbf{Z}_i|\mathbf{m}_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{Z}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k)\right\}. \quad (\text{A.1})$$

Thus, the texem generative model is derived by solving the parameters of a standard Gaussian mixture model:

$$p(\mathbf{Z}_i|\theta) = \sum_{k=1}^K p(\mathbf{Z}_i|\mathbf{m}_k) \alpha_k, \quad (\text{A.2})$$

where $\theta = \{\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\omega}_k\}_{k=1}^K$ is the graylevel texem parameter set containing α_k , which is the *prior* probability of the k th texem constrained by $\sum_{k=1}^K \alpha_k = 1$, the mean $\boldsymbol{\mu}_k$, and the covariance $\boldsymbol{\omega}_k$.

The log-likelihood expression for this density from the data \mathbf{Z} is given by:

$$\log p(\mathbf{Z}|K, \theta) = \sum_{i=1}^P \log p(\mathbf{Z}_i|\theta) = \sum_{i=1}^P \log\left(\sum_{k=1}^K p(\mathbf{Z}_i|\mathbf{m}_k) \alpha_k\right). \quad (\text{A.3})$$

As it contains the log of sum, it is difficult to optimise [10]. One may make use of Jensen's inequality to convert (A.3) into a form of summing of logs. However, if the information of which

component generated each data point, in this case image patch, was available the problem of determine the texem parameters would be straightforward. Thus, we consider a hypothetical complete data set by assuming \mathbf{Z} is incomplete and there exist unobserved data items $\mathcal{Y} = \{y\}_{i=1}^P$ specifying which texem generated the image patch. The likelihood for this complete data is then given by:

$$\begin{aligned} \log(\mathcal{L}(\Theta|\mathbf{Z}, \mathcal{Y})) &= \log(p(\mathbf{Z}, \mathcal{Y}|\Theta)) \\ &= \sum_{i=1}^P \log(p(\mathbf{Z}_i|y_i)p(y_i)) \\ &= \sum_{i=1}^P \log(p(\mathbf{Z}_i|\theta_{y_i})\alpha_{y_i}). \end{aligned} \quad (\text{A.4})$$

Assuming the knowledge of which texem generated each image patch is available, the complete data log likelihood function can be decomposed into a sum of independent terms [11]. However, the data \mathcal{Y} is hidden. Therefore, we start with a guess that $\theta^{(t)} = (\alpha_1^{(t)}, \dots, \alpha_K^{(t)}, \mu_1^{(t)}, \dots, \mu_K^{(t)}, \omega_1^{(t)}, \dots, \omega_K^{(t)})$ are the parameters of the mixture model and then use these to work out the probability distribution of hidden data. The mixing parameters, α_k can be computed using Bayes' theorem:

$$p(y_i|\mathbf{Z}_i, \theta^{(t)}) = \frac{\alpha_{y_i}^{(t)} p(\mathbf{Z}_i|\theta_{y_i}^{(t)})}{p(\mathbf{Z}_i|\theta^{(t)})} = \frac{\alpha_{y_i}^{(t)} p(\mathbf{Z}_i|\theta_{y_i}^{(t)})}{\sum_{k=1}^K \alpha_k^{(t)} p(\mathbf{Z}_i|\theta_k^{(t)})}. \quad (\text{A.5})$$

Thus, the expectation step of the EM algorithm finds the expected value of the complete data loglikelihood given the observed data and the current parameter estimations:

$$\begin{aligned} Q(\theta, \theta^{(t)}) &= E[\log p(\mathbf{Z}, \mathcal{Y}|\theta)|\mathbf{Z}, \theta^{(t)}] \\ &= \sum_{\mathbf{y} \in \Upsilon} \log(\mathcal{L}(\theta|\mathbf{Z}, \mathbf{y}))p(\mathbf{y}|\mathbf{Z}, \theta^{(t)}) \\ &= \sum_{k=1}^K \sum_{i=1}^P \log(\alpha_k)p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)}) + \sum_{k=1}^K \sum_{i=1}^P \log(p(\mathbf{Z}_i|\theta_k))p(\mathbf{m}_k|\mathbf{Z}_i, \theta^{(t)}). \end{aligned} \quad (\text{A.6})$$

At the maximisation step, the term containing α_k and the term containing θ_k can be maximised independently. Thus, the parameters can be solved by taking the derivative with respect to each parameter and setting it equal to zero. However, for the mixing parameter α_k , we need to take into the constraint that $\sum_k \alpha_k = 1$ by introducing the Lagrange multiplier λ and solving the following

equation:

$$\begin{aligned} \frac{\partial}{\partial \alpha_k} \left(\sum_{k=1}^K \sum_{i=1}^P \log(\alpha_k) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) + \lambda \left(\sum_k \alpha_k - 1 \right) \right) &= 0 \\ \sum_{i=1}^P \frac{1}{\alpha_k} p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) + \lambda &= 0. \end{aligned} \quad (\text{A.7})$$

Summing both sides of (A.7) over k and noticing that $\sum_k \alpha_k = 1$ and $\sum_{k=1}^K p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) = 1$, we obtain $\lambda = -P$ which leads to:

$$\alpha_k = \frac{1}{P} \sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}). \quad (\text{A.8})$$

The texem parameters θ_k then can be solved by taking the log of (A.1) and substituting into the second term in the right side of (A.6):

$$\begin{aligned} &\sum_{k=1}^K \sum_{i=1}^P \log(p(\mathbf{Z}_i | \boldsymbol{\mu}_k, \boldsymbol{\omega}_k)) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) \\ &= \sum_{k=1}^K \sum_{i=1}^P \log\left(\frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_k|}} \exp\left\{ -\frac{1}{2} (\mathbf{Z}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k) \right\} \right) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}). \end{aligned} \quad (\text{A.9})$$

Ignoring any constant and taking the derivative of (A.9) with respect to $\boldsymbol{\mu}_k$ and setting it to zero:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\mu}_k} \left(\sum_{k=1}^K \sum_{i=1}^P \log\left(\frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\omega}_k|}} \exp\left\{ -\frac{1}{2} (\mathbf{Z}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k) \right\} \right) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) \right) &= 0 \\ \frac{\partial}{\partial \boldsymbol{\mu}_k} \left(\sum_{i=1}^P \left(-\frac{1}{2} (\mathbf{Z}_i - \boldsymbol{\mu}_k) \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k) \right) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) \right) &= 0 \\ \sum_{i=1}^P \boldsymbol{\omega}_k^{-1} (\mathbf{Z}_i - \boldsymbol{\mu}_k) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) &= 0. \\ \sum_{i=1}^P (\mathbf{Z}_i - \boldsymbol{\mu}_k) p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) &= 0 \\ \sum_{i=1}^P \mathbf{Z}_i p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) &= \sum_{i=1}^P \boldsymbol{\mu}_k p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) \end{aligned} \quad (\text{A.10})$$

Then leads to the solution of $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^P \mathbf{Z}_i p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}. \quad (\text{A.11})$$

In a similar fashion, we can solve the covariance matrix $\boldsymbol{\omega}_k$:

$$\boldsymbol{\omega}_k = \frac{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) (\mathbf{Z}_i - \boldsymbol{\mu}_k) (\mathbf{Z}_i - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}. \quad (\text{A.12})$$

Finally, the update equations for the texem parameters of the Gaussian mixture model take the following forms:

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_k &= \frac{1}{P} \sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}), \\ \hat{\boldsymbol{\mu}}_k &= \frac{\sum_{i=1}^P \mathbf{Z}_i p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}, \\ \hat{\boldsymbol{\omega}}_k &= \frac{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) (\mathbf{Z}_i - \hat{\boldsymbol{\mu}}_k) (\mathbf{Z}_i - \hat{\boldsymbol{\mu}}_k)^T}{\sum_{i=1}^P p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})}. \end{aligned} \quad (\text{A.13})$$

where $p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)})$ can be written as, according to (A.5)

$$p(\mathbf{m}_k | \mathbf{Z}_i, \theta^{(t)}) = \frac{\boldsymbol{\alpha}_k^{(t)} p(\mathbf{Z}_i | \theta_k^{(t)})}{p(\mathbf{Z}_i | \theta^{(t)})} = \frac{\boldsymbol{\alpha}_k^{(t)} p(\mathbf{Z}_i | \theta_k^{(t)})}{\sum_{k=1}^K \boldsymbol{\alpha}_k^{(t)} p(\mathbf{Z}_i | \theta_k^{(t)})} \quad (\text{A.14})$$

The EM then iterates by using updated parameters as new estimates.

Appendix B

LIST OF PUBLICATIONS

Here is a list of publications that result from this thesis. A short abstract is presented for each publication.

TEXEMS: Texture Exemplars for Defect Detection on Random Textured Surfaces

Xianghua Xie and Majid Mirmehdi

Abstract - We present an approach to detecting and localizing defects in random color textures which requires only a very few defect free samples for unsupervised training. It is assumed that each product image is generated by superposition of various-size image patches with added variations at each pixel position. Those image patches and their variances are referred to here as textural exemplars or texems. Mixture models are applied to obtain the texems using multiscale analysis to reduce the computational costs. Novelty detection on color texture surfaces are performed by examining the lower bound of normal samples likelihoods on the multiscale defect map of an image under inspection to localize defects.

In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, submitted, 2006.

Colour tonality inspection using eigenspace features

Xianghua Xie, Majid Mirmehdi, and Barry Thomas

Abstract - In industrial quality inspection of colour texture surfaces, such as ceramic tiles or fabrics, it is important to maintain a consistent colour shade or tonality during production. We present a multidimensional histogram method using a novelty detection scheme to inspect the surfaces. The image noise, introduced by the imaging system, is found mainly to affect the chromatic channels. For colour tonality inspection, the difference between images is very subtle and comparison in the noise dominated chromatic channels is error prone. We perform vector-ordered colour smoothing and extract a localised feature vector at each pixel. The resulting histogram represents an encapsulation of local and global information. Principal Component Analysis is performed on this multidimensional feature space of an automatically selected reference image to obtain reliable colour shade features, which results in a reference eigenspace. Then unseen product images are projected onto this eigenspace and compared for tonality defect detection using histogram comparison. The proposed method is compared and evaluated on a data set with groundtruth.

In: *Machine Vision and Applications*, volume 16, number 6, pages 364-373, Springer Press, 2006.

Localising Surface Defects in Random Colour Textures using Multi-scale Texem Analysis in Image Eigenchannels

Xianghua Xie and Majid Mirmehdi

Abstract - A novel method is presented to detect defects in random colour textures which requires only a very few normal samples for unsupervised training. We decorrelate the colour image by generating three eigenchannels in each of which the surface texture image is divided into overlapping patches of various sizes. Then, a mixture model and EM is applied to reduce groupings of patches to a small number of textural exemplars, or texems. Localised defect detection is achieved by comparing the learnt texems to patches in the unseen image eigenchannels.

In: *Proceedings of the 12th IEEE International Conference on Image Processing*, pages (III)1124–1127, IEEE CS Press, September 2005.

Texture Exemplars for Defect Detection on Random Textures

Xianghua Xie and Majid Mirmehdi

Abstract - We present a new approach to detecting defects in random textures which requires only very few defect free samples for unsupervised training. Each product image is divided into overlapping patches of various sizes. Then, density mixture models are applied to reduce groupings of patches to a number of textural exemplars, referred to here as texems, characterising the means and covariances of whole sets of image patches. The texems can be viewed as implicit representations of textural primitives. A multiscale approach is used to save computational costs. Finally, we perform novelty detection by applying the lower bound of normal samples likelihoods on the multiscale defect map of an image to localise defects.

In: *Proceedings of the 3rd International Conference on Advances in Pattern Recognition*, pages 404–413, Springer LNCS 3687, August 2005.

Detecting Abnormalities in Tympanic Membrane Images

Xianghua Xie, Majid Mirmehdi, Richard Maw, and Amanda Hall

Abstract - We present a method to detect abnormalities in colour tympanic membrane (eardrum) images. The method first uses a geometric snake to localise the regions of interest and normalise the colour pixels based on a reference image. Then, defect detection and localisation in multiscale is performed using a statistical model. Initial experimental results (presented) suggest excellent potential for in-depth progress and validation.

In: *Proceedings of the 9th Medical Image Understanding and Analysis*, pages 19–22, BMVA Press, July 2005.

Inspecting Colour Tonality on Textured Surfaces

Xianghua Xie, Majid Mirmehdi, and Barry Thomas

Abstract - We present a multidimensional histogram method to inspect tonality on colour textured surfaces, e.g. ceramic tiles. Comparison in the noise dominated chromatic channels is error prone. We perform vector-ordered colour smoothing and generate a PCA-based reconstruction of a query

tile based on a reference tile eigenspace. Histograms of local feature vectors are then compared for tonality defect detection. The proposed method is compared and evaluated on a data set with groundtruth.

In: *Proceedings of the 1st International Conference on Image Analysis and Recognition*, pages 810–817, Springer LNCS 3212, September 2004.