Textural Difference Enhancement based on Image Component Analysis

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Jianning Chi

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

In this thesis, we propose a novel image enhancement method to magnify the textural differences in the images with respect to human visual characteristics. The method is intended to be a preprocessing step to improve the performance of the texture-based image segmentation algorithms.

We propose to calculate the six Tamura's texture features (coarseness, contrast, directionality, line-likeness, regularity and roughness) in novel measurements. Each feature follows its original understanding of the certain texture characteristic, but is measured by some local low-level features, e.g., direction of the local edges, dynamic range of the local pixel intensities, kurtosis and skewness of the local image histogram. A discriminant texture feature selection method based on principal component analysis (PCA) is then proposed to find the most representative characteristics in describing textual differences in the image.

We decompose the image into pairwise components representing the texture characteristics strongly and weakly, respectively. A set of wavelet-based soft thresholding methods are proposed as the dictionaries of morphological component analysis (MCA) to sparsely highlight the characteristics strongly and weakly from the image. The wavelet-based thresholding methods are proposed in pair, therefore each of the resulted pairwise components can exhibit one certain characteristic either strongly or weakly.

We propose various wavelet-based manipulation methods to enhance the components separately. For each component representing a certain texture characteristic, a non-linear function is proposed to manipulate the wavelet coefficients of the component so that the component is enhanced with the corresponding characteristic accentuated independently while having little effect on other characteristics.

Furthermore, the above three methods are combined into a uniform framework of image enhancement. Firstly, the texture characteristics differentiating different textures in the image are found. Secondly, the image is decomposed into components exhibiting these texture characteristics respectively. Thirdly, each component is manipulated to accentuate the corresponding texture characteristics exhibited there. After re-combining these manipulated components, the image is enhanced with the textural differences magnified with respect to the selected texture characteristics.

The proposed textural differences enhancement method is used prior to both grayscale and colour image segmentation algorithms. The convincing results of improving the performance of different segmentation algorithms prove the potential of the proposed textural difference enhancement method.

Keywords: texture difference enhancement; segmentation improvement; human visual perception; morphological component analysis; wavelet-based dictionary.

Acknowledgements

I would like to thank the people from the Imaging, Multimedia and Graphics (IMG) Lab at the University of Saskatchewan for their friendship and support along these years.

First and foremost, I would like to thank Associate Professor Mark Eramian, who supervised me for the last five years. During my Ph.D study, he has become not only a supervisor in research, but also a mentor in my life. I learned a lot from him, not only on image processing and computer vision, but on the scientific theory and process. I appreciate all his contributions of time, ideas and funding to make my Ph.D. study productive and impressive. The enthusiasm he has for his research and teaching motivated me during difficult time in the Ph.D. pursuit. He has provided an excellent example as a successful scientist and professor. I also would like to thank Professor Eric Neufeld, Assistant Professor Ian Stavness and Associate Professor David Cooper, my thesis committee members, for their dedication in reading the thesis, providing ideas and being open-minded to have good debates with.

I would like to thank my Ph.D. colleagues, Xin Yi and Rafizul Haque. They help me a lot by enlightening me with novel ideas on theories, experiments and programs. They also provide constructive suggestions and comments during the writing of this thesis. I also would like to thank Dr. Ekta Walia for her help by providing ideas in research and sharing experiences in technical paper writing. Moreover, the friendship in the whole research group relieves the stressful research.

I would like to thank Gwen Lancaster, Graduate Program Assistant of the Department of Computer Science, for her helps in answering questions about Ph.D. study, informing important issues, and scheduling comprehensive exam, proposal defence and all the committee meetings. I also would like to thank all other staffs in the department for their help and dedication.

Finally, to my family for supporting me in all possible ways and encouraging me to achieve all good things during all my pursuits. I dedicate the achievement to them as a minimal return for what they have done and still do for me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

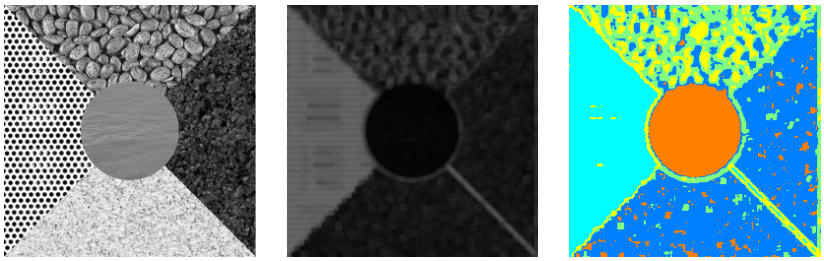
Image enhancement aids in a diverse selection of image processing methods because higher image quality generally results in better processing performance. In this thesis we propose a novel method to enhance textural differences between textures in an image, which is used as a general pre-processing step to improve results for texture-based segmentation algorithms.

1.1 Background

1.1.1 Texture-based segmentation

Texture-based segmentation can be generally defined as partitioning an image into regions of similar texture. Texture-based segmentation methods have recently received more attention because they are relatively adaptive and robust to noise [108]. As a simple example, texture-based segmentation can be implemented by thresholding feature maps of the image [176]. In [172], a group of seed pixels were firstly selected in the original image and regions were grown by appending to each seed those neighbouring pixels that have texture features similar to seed pixels. Rather than choosing seed points, the user can divide an image into a set of arbitrary unconnected regions and then merge the regions [112, 154] with similar texture features. Most of the clustering-based segmentation methods, including hard clustering [68], k-means clustering [148], mean-shift clustering [79], and fuzzy clustering [124], can be adapted to texture-based segmentation: a similarity criteria is defined between pixels with respect to texture features, then similar pixels are grouped together to form clusters.

As discussed above, the performance of texture-based segmentation mostly relies on the textural differences in the image. Therefore, the texture-based segmentation could be difficult when the textural differences are too subtle to differentiate. For example, as shown in Fig. 1.1, the segmentation method fails to partition the image into five different texture regions because the quantitative texture features of the similar regions cannot be reliably distinguished.



(a) example image with multiple textures

(b) feature map of the example image

(c) segmentation of the example image

Figure 1.1: An example of texture-based segmentation. (a) is the example image with five different Brodatz [24] textures in it, (b) is the map of local contrast description [184] of the example image, (c) is the segmentation of the example texture image by applying the k-means clustering [10] to the feature map.

1.1.2 Improvement of texture-based segmentation

One way to achieve better performance of texture-based segmentation is to design more effective segmentation algorithms, including extracting features that describe the differences between each region more clearly [78], or clustering the pixels with similar features more accurately [127]. However, most of the recent segmentation algorithms perform well only for the type of images they are designed to segment [57]. This disadvantage leads to a novel attempt to design image enhancement methods magnifying the differences between regions as the pre-processing steps prior to texture-based segmentation methods. For example, as shown in Fig. 1.2, different regions of the example image are made more different so that the segmentation results are more accurate, closer to the ground truth. Image enhancement magnifies the texture differences without prior-knowledge about the number and kind of textures present in the image or the segmentation algorithm, leading to improved performance of different texture-based segmentation on different types of image.

1.1.3 Image Enhancement

Different image enhancement methods have been proposed to highlight the texture regions with respect to certain features. Some early texture enhancement methods reduce noise or artifacts in the image to highlight the textures indirectly, for example, the median filter [98] and the Weiner filter [104]. However, these conventional filters degrade the textures in addition to removing noise because of their lowpass-filter-like qualities. Discontinuity-preserving filters were developed which mitigate this issue to a certain extent by preserving the image details and local geometries while removing the undesired noise [152]. Filtering based on total variation (TV) was proposed by Rudin et. al. [169] to remove the noise by minimizing the total variation (TV) of the image subject to constraints involving the statistics of the noise. Bilateral filtering proposed by Tomasi et. al. [199] can preserve image details while de-noising by weighting the local Gaussian filter coefficients with their corresponding relative pixel intensities. The non-local means filter was proposed in [25] to estimate a pixel by using the similarities between it and all the other pixels in image to act as

(a) original texture image (b) enhanced texture image

(c) segmentation of (a) (d) segmentation of (b) (e) ground truth

Figure 1.2: Example of enhancing the segmentation algorithm by magnifying the differences between different regions in the image. (a) is the original image, (b) is the image enhanced by the adaptive histogram equalization [157], (c) is the segmentation of the original image using the k-means clustering method [148], (d) is the segmentation of the enhanced image using the same algorithm, and (e) is the ground truth of the segmentation.

weight, and the similarities are computed not from pixels themselves but from their neighbouring window. It can smooth the noise and artifacts in the image while preserving image detail as much as possible. Wavelet-based methods, e.g. VISUShrink [47], BayesShrink [64], and SUREShrink [65] were proposed to remove noise by shrinking coefficients in high-frequency sub-bands not exceeding certain thresholds, while preserving the image textures which are represented by coefficients in high-frequency sub-bands that exceed these thresholds. Recently, the de-blurring methods have become attractive in image enhancement or restoration, especially for removing motion noise. Y. Hacohen et. al. [87] proposed a de-blurring method by using a sharp reference example to a non-rigid dense correspondence (NRDC) model [86]. Tomer Michaeli et. al. [140] proposed a blind de-blurring method using internal patch recurrence for recovering the unknown blur kernel.

Other methods enhance the textures in the image directly. Unsharp masking (UM) was proposed to improve the visual appearance of an image by emphasizing its high frequency contents [158]. However, the highpass-filter-like nature of UM causes enhancement of noise and artifacts in the image as well. The same is true of histogram equalization methods [157]. Therefore, some non-linear methods have been proposed to enhance the textures. Hong et. al. [94] proposed a texture enhancement algorithm that can improve the clarity of ridge and valley structures of fingerprint textures based on the estimated local ridge orientation and frequency. Coherence-enhancing anisotropic diffusion [209] is based on the modifications of partial

differential equations (PDEs) [37] which can preserve strong discontinuities at edges while removing artifacts from smooth regions. Shock filtering [210] is a transformation of anisotropic diffusion which smooths along the coherent texture flow orientations, and reduces diffusivity at non-coherent structures which enhances textural details. In recent research, fractional differential masks [105, 77] proved to be effective in enhancing the edges in the image, therefore enhancing the image textures. These texture enhancement methods have attracted more attention recently because they directly highlighted texture features without losing image details. Unfortunately, current research work has been limited in enhancing the weak edges or contrast of textures in the image.

The conventional methods introduced above have attempted to enhance the "textures" with the "non-textures" unchanged, or smooth the "non-textures" with the "textures" preserved. However, most of these existing enhancement methods had the following disadvantages:

1. the differences between "textures" and "non-textures" were subtle, resulting in the mis-enhancement of the "non-textures" while the mis-suppression of the "textures"; and

2. the high-frequency contents of different types of "textures" were enhanced to the same extent, while the low-different types of "non-textures" were suppressed equally, even though some adaptive methods were applied.

These two drawbacks led to that the quantitative difference between texture descriptors for textures is not much altered, resulting in only aesthetic enhancement but little improvement in image segmentation.

1.2 Motivation and Hypothesis

The research work summarized in Section 1.1 demonstrates the following facts:

- image enhancement can generally improve the performance of different texture-based segmentation methods;

- the existing image enhancement methods cannot effectively emphasize the differences between textures because they manipulate different types of textures identically.

These two facts are followed by the idea that different types of textures should be enhanced separately and differently, so the textural differences can be magnified and the performance of segmentation methods based on these differences can be improved consequently. It naturally follows that image component analysis [85] might be useful for textural differences enhancement, with the following hypotheses:

- images can be considered as the combination of components representing different image characteristics;

- texture components of a given image are independent of each other, so that the manipulation of each component won't affect the manipulations of the others;

- there exist some non-linear image manipulation methods that can accentuate or suppress a texture's own properties.

The possible advantages of this component-analysis-based textural differences enhancement method are:

- unlike the conventional model, an image can be considered as the combination of several components representing different texture characteristics, leading to a novel understanding of the image composition;
- textures with different characteristics are separated into independent components, and by developing the enhancement methods for these components to highlight their own properties, the re-combination of these components will result in an enhanced image where textures are more different with respect to the certain texture characteristics; and
- since the components separated from the image represent some certain texture characteristics, the characteristics found in the image could guide the selection of the texture descriptors to be used with a texture-based image segmentation algorithm.

This thesis investigates how to enhance the image by magnifying the textural differences between the different textures in it. It is intended to be used as a general preprocessing method to be applied prior to texture-based image segmentation.

1.3 Main problems and solution

The above discussion derives the following three main problems:

- What characteristics should be used to describe and separate a given image? There are several existing texture descriptors which can describe the images with different characteristics. However, some descriptors that describe images with low-level characteristics do not reflect textural differences according to human visual perception. While some other descriptors describe images with semantic characteristics, these characteristics are not orthogonal to each other, resulting in the redundancy of the description.
- How can a given image be separated into components representing different texture characteristics? This is the core problem in this work. There are several component analysis methods [139, 61, 49], but these methods can only separate an given image into components representing different directions, oscillated and smooth, or low-level characteristics and coefficients map, and are not able to find image components representing representing higher-level texture characteristics.
- How can a separated component be enhanced to highlight the texture characteristic it represents while not changing other characteristics represented by other components? Since existing methods focus on enhancing low-level characteristics (edges, high-frequency coefficients, high gradients, spatial distances, etc.), high-level characteristics in the images are enhanced to the same extent, resulting in difficulties in differentiating them.

Texture description, image decomposition, and non-linear image enhancement methods can be used to solve the problems listed above. We will:

- study human visual perceptual texture description and select the texture characteristics that can describe the textural differences in the image completely and orthogonally;
- develop morphological component analysis using wavelet-based dictionaries, by which a given image can be separated to pairs of components respectively exhibiting any given texture characteristic strongly and weakly;
- develop a group of wavelet-based manipulation methods, each of which can accentuate one certain texture characteristic exhibited in the image while preserving other characteristics;
- combine the above solutions to propose a uniform framework, as shown in Fig. 1.3, to enhance the textural differences in the image, which can be used as a pre-processing method to improve texture-based segmentation algorithms generally.

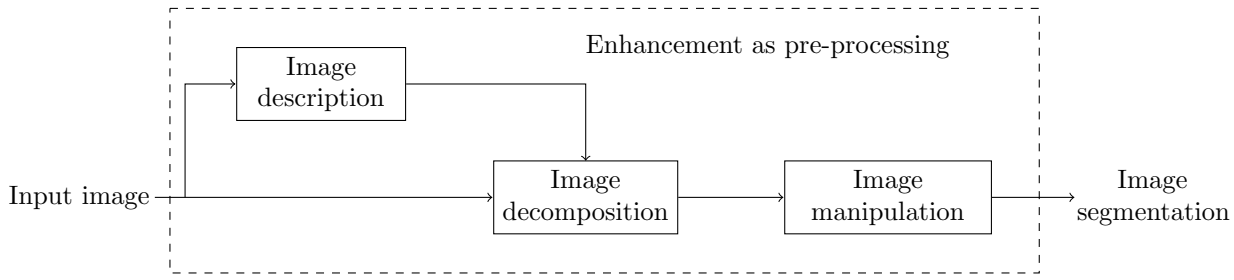


Figure 1.3: The process of our textural differences enhancement method. Human visual perceptual texture characteristics are selected to describe the textural differences in the image. Then image is decomposed into pairwise components exhibiting any of the selected characteristics strongly and weakly. After that, each component is manipulated to accentuate the corresponding characteristic. Re-combining the manipulated components result in the enhanced image where the textural differences are magnified. These steps together function as a uniform pre-processing framework for better texture-based segmentation.

1.4 Outline of the Thesis

Chapter 2 gives a review of the literature on the methods of image description, image decomposition, and image manipulation. For the image description methods, we focus those statistical description methods based on human visual perception, excluding some low-level description methods, e.g., first-order statistics [83], spin image based description [118], or texel-based description [35], because the review is too extensive otherwise. For the same reason, the discussion of image manipulation is limited to non-linear methods for manipulating images.

In chapter 3, Tamura's texture descriptor is studied in detail as a specific human visual perceptual texture description method. Tamura's descriptor is improved to a local texture descriptor, making it especially

suitable for discriminating different textures in the given images. Then the principal component analysis (PCA) algorithm is applied to select key characteristics from the six Tamura's texture characteristics, solving the problem that some of those characteristics are highly correlated to others. Thereby the essential characteristics which make the textures different can be found.

Chapter 4 applies the selected texture characteristics as the basis of image decomposition. Morphological component analysis (MCA) is improved to decompose the image into components representing the different essential texture characteristics. Wavelet-based thresholding methods are proposed and used as the dictionaries of the MCA. Using Tamura's description to measure the intra-component-similarity within one component and inter-component-difference between different components, it is shown the proposed method can provide good performance in separating image into components representing different texture characteristics. For each characteristic, we study and quantify the minimum textural difference that can be decomposed according to the certain characteristic.

The decomposition of the image leads to the idea that different texture characteristics can be highlighted by enhancing the components representing certain characteristics. In chapter 5, wavelet-based image enhancement methods are introduced. These methods enhance the components' own properties by manipulating the wavelet coefficients of the components. Instead of removing or shrinking the wavelet coefficients as conventional methods, the proposed methods manipulate the wavelet coefficients as nonlinear functions of these coefficients themselves, such as adding "multiplicative noise" along the coefficients, leading to an enhancement without loss of image information. After that, enhanced components are re-combined to form the enhanced image, where the textural differences are magnified.

In chapter 6, we discuss the applications of the proposed textural differences enhancement method. We show that by pre-processing with the new enhancement method the results of segmenting pure texture images and natural images by several texture-based segmentation algorithms can be considerably improved.

Chapter 7 ends the thesis with some discussion, conclusions and plans for further research work.

1.5 Publications

The thesis contains work previously published and submitted for publication. The sources are as follows:

- In Chapter 4, the concept of decomposing the image based on texture characteristics is contained in a paper published in *IEEE Transactions on Image Processing* [44].
- In Chapter 4 and 5, the proposed wavelet-based dictionaries for image decomposition and wavelet-based enhancement methods for enhancing the components are contained in a paper submitted to *Computer Vision and Image Understanding* [45].
- In Chapter 6, a paper on the the application of enhancing colour image segmentation algorithm has been accepted and will be published in the proceedings of *ICIP 2016* [46].

Chapter 2

Literature review

In preparation for pursuing the solution to the problem that magnify the textural differences in the images, we have thoroughly reviewed the literature in areas of texture description, image decomposition and image manipulation in detail. We exclude some literature that have little relationship with our objective to avoid a too extensive discussion.

2.1 Texture description

The central problem in many computer vision tasks can be regraded as extracting "meaningful" descriptions from images or image sequences [111]. These descriptions are then used to solve problems such as object recognition, classification or tracking [130, 141, 174, 175]. Different descriptors may reflect different characteristics of the texture, so in our method of manipulating image textures, it is essential to select a certain texture descriptor that is sensitive to some specific texture characteristics that we wish to measure.

Mathematical procedures to characterize texture can be classified into two major categories: statistical and structural approaches. In early research, it has been demonstrated that statistical texture description methods described textures by a quantitative measure of the arrangements of pixel intensities, while structural texture descriptors described a texture as the composition of well-defined texels such as regularly spaced parallel lines. Since structural methods [35, 230] can only describe very regular textures [83], statistical methods have become the mainstream methods in describing arbitrary textures. According to the features extracted from the image, the popular statistical texture descriptors can be categorized as: low-level texture descriptor and high-level texture descriptor.

2.1.1 Low-level texture descriptor

Low-level texture descriptors describe the texture by the intuitive features of the texture itself, including the occurrence and direction of the edges or corners, the shape of the textures, the occurrence frequency of the texture primitives, the variance of the pixel intensities and other primitive level features. The most widely-used low-level image description methods are reviewed as follows.

Model-based descriptor

The model based descriptors typically generate a model of each pixel in the image based on a weighted average of the pixel intensities in its neighbourhood. Then the estimated parameters of the image models are used as textural feature descriptors. Deguchi et. al. [58] proposed the autoregression (AR) texture model, where they "estimated" the pixel intensity as a linear combination of its neighbours' intensities and used the coefficients as the descriptor. Alex P. Pentland [156] proposed the fractal model that related a metric property such as line length or surface area to the elementary length or area used as a basis for determining the metric property. They demonstrated a correlation between texture coarseness and fractal dimension of a texture. Another popular model based descriptor is based on Markov random fields (MRF), which was proposed in [55] by George R. Cross and Anil K. Jain. Similar as Alex P. Pentland did in AR model, George and Anil presented that the intensity of the central pixel was determined by the intensities distribution of its neighbouring pixels, which was a Markov random field distribution. These Markov random field parameters can describe the strength and directions of the clustering of the textures.

Edge-based descriptor

In [195], Sutton, R.N. and Hall, Ernest L. firstly proposed an edge detection based descriptor by calculating the gradient magnitude and direction on the edge pixels by the pixel intensities, then measuring the distribution of the texture edges by a distance-dependent texture description function. Some other texture properties that are derived from first-order and second-order statistics of edge distributions were summarized by Tomita and Tsuji in [200]. Edge-based texture description can provide good performance of describing images consisting of homogeneous textural regions. It can determine image regions of constant texture with no need to assume a priori knowledge about the image, texture types or scale. However, the major disadvantage of the edge-based texture description is that it is implemented after edge detection. If the result of edge detection or gradient calculation is not accurate, the result of texture description is not meaningful either.

Local binary pattern (LBP) descriptor

Local binary pattern (LBP) [150], was proposed by Ojala et. al. to describe the texture by computing the LBP code from comparing a pixel intensity with its neighbours. Based on the original LBP, several methods were proposed to improve the LBP with rotation invariant variance measurement [225], adding local differences features [225], avoiding missing local structure information [107], overcoming the distortion of noise [122], combining with multi-scale analysis [215], better detecting various texture structures as dominant features of images [223] and incorporating the non-uniform patterns [217].

Scale invariant feature transformation descriptor

Scale invariant feature transform (SIFT) is another important intensity based descriptor proposed in [129] by David G. Lowe, which combined a scale invariance region detector and a description based on the gradient distribution in the detected regions. Although SIFT feature description was applied in image matching and indexing, video tracking or moving matching [177], recent works show the possibility of using SIFT to describe textures to improve texture classification or segmentation [222]. Some other methods have been proposed to improve the distinctiveness and robustness of the original SIFT. For example, the gradient computation in original SIFT, which was done with pixel differences, is very sensitive to noisy measurements [144]. In [121], Liao and Liu proposed an improved SIFT descriptor by means of combining the second derivative and the gradient magnitude and introducing the polar histogram orientation bin to improve the distinctiveness and robustness. Another way to improve SIFT is to reduce the high dimensionality of the feature vectors. In [113], Yan and Sukthankar proposed PCA-SIFT by using PCA instead of the histogram to normalize gradient patch. Fewer components required less storage and resulted in a faster analysis, which lead to significant improvement in efficiency. By combining SIFT and Locality Preserving Projection (LPP), a linear version of manifold learning algorithm, Cao et. al. in [33] proposed LPP-SIFT to extract the most discriminative feature so that the dimensionality could be reduced.

Spatial frequency based descriptor

The spatial frequency based descriptors are based on an assumption that textural characteristic directly relates to the spatial frequencies of texture primitives: Fine textures are characterized by higher spatial frequencies, coarse textures are characterized by lower spatial frequencies. One of many related spatial frequency methods evaluated the autocorrelation function of a texture [194], where the decreasing speed of the autocorrelation function around the certain texture primitive (pixel) with increasing distance had a negative correlation with the coarseness of the texture. Shulman et. al. [180] first proposed to use average values of energy in specific rings of the Fourier spectrum as a spatial frequency based texture description features. Features evaluated from the rings can reflect the coarseness of the texture because high energy in large radius rings is one of the character of fine textures and vice versa. Texture description based on Gabor transform, proposed by Huang et. al. [99], overcame the Fourier transform's problem of depending on global information via applying the Gabor filters, which were a group of wavelets with each wavelet capturing energy at a specific frequency and a specific direction. Therefore, the Gabor texture descriptor can measure the coarseness and directionality of the texture in different scales.

2.1.2 High-level texture descriptor

The above low-level texture descriptors lacked in giving significant attention to explore the relationship between the texture characteristics they described and human visual sense. Therefore, image descriptors for

high-level features were proposed to describe how human beings "feel" about the different texture regions.

Gray level co-occurrence matrix (GLCM) texture descriptor

Haralick et. al. [90] defined 14 characteristics based on the gray level co-occurrence matrix (GLCM), which was a two-dimensional array C in which both the rows and the columns were indexed by the set of possible image values V and the value of $C(i, j)$ indicated how many times value i co-occurred with value j in some designated spatial relationship. In the 14 defined characteristics, the energy, entropy, contrast, homogeneity and correlation were the most popular-used to describe a texture from others.

Laws' texture descriptor

Laws in [117] asserted that uniformity, density, coarseness, roughness, regularity, linearity, directionality, frequency and phase played important roles in describing a texture. They presented the Laws' texture energy measures to determine these properties by assessing average gray level, edges, spots, ripples and waves in texture [116], which were derived from the convolution of three simple vectors calculating the averaging, first difference (edge) and second difference (spots) with themselves and each other.

Tomita and Tsuji's texture descriptor

Tomita and Tsuji's work proved an edge detection based descriptor could also describe a wide range of texture characteristics. They summarized the texture density, directionality, coarseness, randomness, linearity and periodicity in [195, 200] by first detecting texture edges then calculating the edge densities in a given area, the entropy of the edge magnitude histogram, the co-occurrences of edge pairs with the same edge direction or perpendicular to the edge directions at constant distances.

M. Amadasun's work

M. Amadasun et. al. [6] developed new measures that corresponded to textural characters, and therefore to visual perception of textures. They defined five basic characters of texture namely as: coarseness, contrast, busyness, complexity and texture strength in terms of spatial changes in intensity. To evaluate the proposed texture characters, they used a ranking experiment where subjects (human and computer) were told to rank the ten textures using each of the five characters then calculated the degree of correspondences between the ranking results by Spearman's coefficient of rank correlation. The results showed very high levels of correspondence of coarseness and busyness between computational and perceptual measurements (0.856 and 0.782 respectively), with little lower levels of those three (contrast: 0.685, complexity: 0.600, strength: 0.600). However, the results for combining all these characters to recognize different textures performed quite unstably, which meant a distance from the proposed description to the human visual perception.

Rao and Lohse's work

Rao and Lohse et al. [163] developed a classification method for perceptual texture. Based on psychological similarity judgements, they constructed a three-dimensional space for texture description. The three orthogonal dimensions were (1) repetitive vs. non-repetitive, (2) high-contrast and non-directional vs. low-contrast and directional, and (3) granular, coarse and low-complexity vs. non-granular, fine and high-complexity. Similar experiments conducted by Cho et al. [48] suggested that the dimensionality of perceptual texture space was at least four. They described four orthogonal attributes, named as coarseness, regularity, contrast and lightness.

K. Fujii's work

K. Fujii et al. [76] presented a set of texture characters (contrast, coarseness and regularity) corresponding to perceptual properties of visual texture based on autocorrelation function (ACF). They measured psychophysical scales for the textural characters then compared these two measurement by the regression coefficients to get a result that how these visual properties were related to particular characters. Their experiments showed that the characteristics they chose and measured by ACF had high correspondences with the perceived texture properties (contrast: 0.89, regularity: 0.86 and coarseness: 0.70). Moreover, the regression coefficients was the first attempt to show how human subjects seemed to be sensitive to the same parameters captured by a certain descriptor (ACF).

Tamura's texture descriptor

H. Tamura et. al. [196] first attempted to compare the correspondence between the texture characteristics and psychological measurements for human subjects. They presented that six textural characteristics had high correlation to human visual perception, which were coarseness, contrast, directionality, line-likeness, regularity and roughness. They developed both computational and psychological measurements for these characteristics and made a comparison between these two measures. From their comparison, they assessed that coarseness, contrast, and directionality had a successful correspondence with human visual perception in describing textures (the correspondence were 0.9, 0.86 and 0.831 respectively), while line-likeness, regularity and roughness had less but still considerable correspondences between computational and psychological measurements (the correspondence were 0.713, 0.719 and 0.652 respectively). As a conclusion from Tamura et. al., coarseness, contrast, directionality would have higher weight in describing textures, but they also admitted that their results needed more simulating experiments and the simple combinations of their features had not simulated the human similarity measurement very well because there was more complex mechanisms existing in the human usage of multiple cues.

Table 2.1 summarizes the image description methods reviewed above, including their categories, capabilities and deficiencies. The literature reviewed here relating to image description is far from complete, and the authors reviewed often emphasize the need for greater attention to this area. No work was found to provide

Method	Category	Capability	Deficiency
Model-based descriptor [58, 156, 55]	Low-level descriptor	describe the texture by the estimated parameters of the image model	• do not reflect the textural differences in human visual sense • lacking in richness to describe a wide variety of textures
Edge-based descriptor [195]	Low-level descriptor	describe the texture by the distribution of the image edges	
LBP descriptor [150]	Low-level descriptor	describe the texture by computing the LBP code from comparing a pixel intensity with its neighbours	
SIFT descriptor [129]	Low-level descriptor	describe the gradient orientation information of the keypoint pixels in different scales	
Tomita and Tsuji's descriptor [200]	High-level descriptor	define density, directionality, coarseness, randomness, linearity and periodicity	• do not provide a local description of textural differences • lacking in deciding orthogonal features for texture description
GLCM descriptor [90]	High-level descriptor	define energy, entropy, contrast, homogeneity, correlation and other 14 characteristics	
Law's descriptor [117]	High-level descriptor	define uniformity, density, coarseness, roughness, regularity, linearity, directionality, frequency and phase	
M. Amadasun's descriptor [6]	High-level descriptor	define coarseness, contrast, busyness, complexity and strength	
Rao and Lohse's descriptor [163]	High-level descriptor	define coarseness, regularity, contrast and lightness	
K. Fujii's work [76]	High-level descriptor	define contrast, coarseness and regularity	
Tamura's descriptor [196]	High-level descriptor	define coarseness, contrast, directionality, line-likeness, regularity and roughness	

Table 2.1: Summary of several popular texture description methods.

a measurement of the degree of correspondence on a wider range of texture properties other than regression. Further, no research was found that demonstrated how to determine the number of orthogonal dimensions of texture characters for different textures. The theoretical methods proposed above for image description is foundational to the hypothesis and research objectives which follow.

2.2 Image decomposition

According to the proposed solution in Chapter 1.3, the texture image needs to be manipulated by modifying components of the texture. Therefore, image decomposition could be a good solution to this problem because it separates the image into different parts. There are three elements for image decomposition methods, which are 1) assumption of input images, 2) assumption of output results, and 3) algorithmic approaches for achieving decomposition.

Assumption of input images

The input images of image decomposition can be obtained from a variety of sources, such as synthetic texture images from different texture database, real world pictures, etc. These images can be classified into the following two types:

- images containing model-specific components [100, 69, 26]: the intensity of each pixel in the image can be considered as the sum of the products of value of some components and their coefficients:

$$I(x,y) = \sum_{k=1}^{K} c_k(x,y)\, s_k(x,y), \tag{2.1}$$

  where $I(x,y)$ is the intensity of the pixel at the position (x,y), $s_k(x,y)$ and $c_k(x,y)$ are the value of the component and the coefficient at (x,y) respectively. A good example is the texture-cartoon model proposed in [26] where an image is considered as the sum of two parts:

$$\inf_{(u,v)\in X_1\times X_2} \{F_1(u) + \lambda F_2(v) : f = u + v\}, \tag{2.2}$$

  where u represents the cartoon part defined as the piecewise smooth image content [139] and v is the texture part containing textures in the image [139].

- images containing components with arbitrary characteristics [119, 60, 59].

Assumption of output results

With different purposes, the results of image decomposition are different. The results can be also categorized into the following two types:

- components corresponding to different features [153, 16]: images are separated into different basis images that display certain features from the original image;

- components matching prior image models [198, 13, 18]: images are decomposed into different images based on models that are learnt before separation.

Algorithmic approaches for decomposition

Given the types of input images and output results, there are two different types of algorithmic approaches for the decomposition:

- component analysis [106, 203, 11]: searching for the comprising components of the input image;
- transformation [219, 43]: using filter masks or other algorithms to transform the images into components as close to the assumption of results as possible.

Based on the above three elements, state-of-the-art methods for image decomposition can be classified as: 1) methods based on mathematical models, 2) methods based on filter banks, and 3) methods based on component analysis. Table 2.2 summarizes the relationship between these kinds of methods and the input, output and strategies that we discussed before.

Method	Input	Output	Strategy
Methods based on mathematical models	Linear combination of several components	Components matching several prior image models	Transformation
Methods based on component analysis	Linear combination of several components	Components corresponding to different features	Component analysis
Methods based on filter banks	Images containing no obvious components	Components corresponding to different features	Transformation

Table 2.2: Characteristics of different image decomposition methods.

2.2.1 Decomposition based on mathematical models

The general concept of this kind of method is that an image can be regarded as composed of several parts following different mathematical models.

At the very beginning of the research in image decomposition, an image was regarded as composed by a structural part, corresponding to the main large objects in the image, and a textural part, containing

fine scale-details, usually with some periodicity and oscillatory nature. Because texture is vague and highly depends on the image scale, the early decomposition methods based on models attempted to model the meaningful structural parts first, and chose different models for textural parts. The most popular model for estimating structural part was total variation proposed by Rudin et. al. [169]. Generally, they solved the function as shown in Equation 2.2. To make better estimation of textural parts, different total variational (TV) based models have been proposed. Here we just introduce some most widely-used models.

Rudin, Osher and Fatemi proposed in [169] a model (ROF) that decomposed an image f into a component u belonging to a bounded variation (BV) function, which meant the total variation of u was finite, and a component v in the L^2 space where the distance of two functions f and g was defined as 2-norm. In their approach the following function was minimized:

$$\inf_{(u,v)\in BV\times L^2/f=u+v}\left(\int |Du| + \lambda \left\|v\right\|_{L^2}^2\right), \tag{2.3}$$

where Du was the total variational model of u, $\left\|v\right\|_{L^2}$ denoted the 2-norm of noise v, which can be modelled as the difference between the input image f and the estimated component u.

In [139], Mayer suggested a new decomposition model based on Banach space G that is composed of distributions f which can be written as:

$$f = \partial_1 g_1 + \partial_2 g_2 = div\left(g\right). \tag{2.4}$$

They also proposed the norm based on G as a model for the oscillatory part as:

$$\left\|v\right\|_G = \inf\left\{\left\|g\right\|_{L^\infty}/v = div\left(g\right),\ g = \left(g_1, g_2\right),\ g_1 \in L^\infty,\ g_2 \in L^\infty,\ \left|g\left(x\right)\right| = \sqrt{\left(\left|g_1\right|^2 + \left|g_2\right|^2\right)\left(x\right)}\right\}. \tag{2.5}$$

The function belonging to G may have a large norm or a small norm. Thus the norm on G was well-adapted to capture the oscillations of an image function in an energy minimization method.

In [149], it was suggested to replace the assumption that v was in L-2 space by that it is in L-1 space, where the distance of two functions was defined as L-1 norm. From their experimental results, Nikolova et. al. showed that their model can preserve the edge of objects from de-noising better than the ROF model and it was well suited for separating "salt and pepper" noise from the smooth regions of the image.

The above methods perform well for synthesized images because the composition of the images is relatively simple. But for natural images, the presence of noise lead researchers to find some new models to differentiate texture and noise. In [139], Meyer et. al. proposed a method for decomposing images into three parts: structures, textures and noise. Each of these were captured by a different model: structures and textures were modelled by functions in BV and G spaces (BV was the bounded variation space and G was defined in 2.4), noise was modelled as a different oscillatory function in G and its G-norm was much smaller than the norm of the textures. The following functions were used:

$$F^{JG}_{\lambda,\mu_1,\mu_2}\left(u,v,w\right) = J\left(u\right) + J^*\left(\frac{v}{\mu_1}\right) + J^*\left(\frac{w}{\mu_2}\right) + \left(2\lambda\right)^{-1}\left\|f - u - \nu_1 v - \nu_2 w\right\|_{L^2}^2, \tag{2.6}$$

where J^* was the adjoint of J, guaranteeing the texture and noise part were in a particular range, defined by

$$J^*\left(\frac{v}{\mu}\right) = \begin{cases} 0, & v \in G_\mu \\ +\infty, & v \in G \setminus G_\mu \end{cases}, \tag{2.7}$$

where

$$G_\mu = \{v \in G \mid \|v\|_G \leq \mu\}. \tag{2.8}$$

Following Meyer's work, several additional models were proposed for distinguishing texture and noise. For example, in [12], Aujol and Chambolle proposed to consider noise as a distribution modelled by functions belonging to Besov space $B^s_{p,q}$. From the experimental results, textures can be better de-noised by this model, which proved a better noise modelling by distributions in the Besov space. But this model didn't take advantage of the adaptivity as the model before. In [82], J. Gilles combined the advantages of both the two models and the experiments showed that the combination of the two properties (Besov space for noise and local adaptivity) increased the quality of the decomposition. Textures were better de-noised and the noise part contained less residual texture by the local adaptive algorithm. Though the noise was not modelled as accurately as which was separated in [12], this method still attracted a great attention since one may care about the texture and structure of an image more than the noise part.

2.2.2 Decomposition based on filter banks

Another category of classical decomposition methods are methods based on filter banks, which transforms the input image without loss of information by a particular filter bank sensitive to some properties of the image. Gaussian pyramid [202] and Laplacian pyramid [3] were two classical methods proposed to decompose image based on different resolutions. The Gaussian pyramid consisted of a sequence of images where each image was a down-sampled, low-pass filtered copy of the previous image. While Laplacian pyramid was a decomposition derived from Gaussian pyramid for each level of this bandpass pyramid represented the difference between successive levels of the Gaussian pyramid. Roberto and Mark proposed the directional filter bank (DFB) image decomposition in [15], which decomposed an image into components representing different directional edges in the image. A 2^m-band DFB was implemented by a tree structure consisting of 'm' levels of two-channel fan filter banks. In [67], the author tried changing the basic fan filter to a quadrature mirror filter (QMF) and using it to the same DFB decomposition structure to get a lower error via filtering the image. Recently, filter banks based on wavelet theory have attracted a great attention for it can decomposition the image both by resolution and directionality. In [7], 2-D wavelet was first expanded from 1-D wavelet transform to decompose an image by analyzing the image at different scales via down-sampling the image. To solve the problem which higher scaling sub-images in wavelet transform have lower resolutions, Fowler et. al. [75] proposed to decompose an image by stationary wavelet transform (SWT), which up-sampled the wavelet in contrast to the traditional wavelet transform, producing full-resolution images at each level of decomposition, leading to the preservation of more image details in the process of decomposition. Donoho et

al. [63] proposed curvelet transform that enabled the possibility to analyze an image with different angular resolutions in a single and effective transform. In [61], Do et. al. proposed a method that can separate the input image into any resolution and any direction called contourlet, which was designed as a combination of the Laplacian pyramid and the directional filter bank.

2.2.3 Decomposition based on component analysis

Methods based on mathematical models and methods based on filter banks both have some disadvantages in decomposing the image into components corresponding to arbitrary characters. Methods based on mathematical models have the problem that the adaptivity of the function is difficult to guarantee because it may be that the whole image cannot be represented by the same combination of the models and the optimal solution (in terms of the model) is often a compromise result. For methods based on filter banks, there are only multi-resolution and multi-directional filter banks used in image decomposition, so if the textures to be processed don't contain some obvious directional features or the contour of the object in the image is not explicit, this kind of methods perform not as well as expected. The following work about decomposing methods based on component analysis have enlightened us a lot in decomposing the image according to arbitrary features of textures.

Independent component analysis (ICA) was originally developed to deal with the blind source separation problem in signal processing [52]. In [49], it was first proposed that ICA could represent an image by the same feature combinations and different coefficients as it did in signal processing. Cichocki et. al. presented that with a suitable selection of the components of ICA, an image can be projected into these basis function directly, which was the target of image decomposition.

The methods of solving an ICA problem was summarized by Hyvarinen in [101] as formulating an objective function that made a choice of the statistical properties of the components and than minimizing or maximizing them, and some optimization algorithms were used to improve the performance of the method in terms of its algorithmic properties. Following this idea, a diversity of ICA algorithms with different objective functions and optimization algorithms were proposed in image processing. Hyvarinen and Oja proposed the FastICA algorithm in [102], which had the advantages: (1) it can directly find independent components of any non-Gaussian distribution using any non-linearity functions, (2) it was parallel distributed, computationally simple and required little memory space, and (3) it was easy to use because it worked automatically after setting the initial parameters. Sparse ICA (SPICA) was proposed in [227] with using the sparsity as the objective function. Huang et al. [97] proposed a L-norm based sparsity as the objective function, then optimized the function by gradient method, finally iterated the result by a Winner-Take-All (WTA) neural network, improving the spasity by 30% and the speed by 40% compared with the classic ICA method.

Morphological component analysis (MCA) was proposed by Starck et at. [189] based on the sparse representation of signals. MCA assumed that each mono-channel signal is the linear mixture of several layers, which were so called morphological components, that are morphologically distinct, for example, sines

and bumps.

A widely-used method in separating an image by MCA was proposed in [23] as solving the minimization problem:

$$\left\{s_1^{opt}, \ldots, s_K^{opt}\right\} = \arg \min_{\{s_1,\ldots,s_K\}} \sum_{k=1}^{K} \|T_k s_k\|_1 + \lambda \left\| s - \sum_{k=1}^{K} s_k \right\|_2^2 + \sum_{k=1}^{K} \gamma_k C_k\left(s_k\right), \tag{2.9}$$

where s_k represented the k-th source of the image signal, T_k was the transformation corresponding to the particular characteristic of the source s_k, s was the observed image, λ was the weighted parameter and C_k implemented constraints on component s_k. $\|\cdot\|$ was a 1-norm, minimizing $\|T_k s_k\|_1$ meant to make the separation sparsest since it measured the number of non-zero entries in the signal matrix s_k after transform T_k. $\left\| s - \sum_{k=1}^{K} s_k \right\|_2^2$ measured the distance between the observed image s and the estimated result $\sum_{k=1}^{K} s_k$ so that a minimum of it meant the decomposed components can reconstruct the image closest to the original one. While minimizing the residue $\sum_{k=1}^{K} \gamma_k C_k\left(s_k\right)$ guaranteed the decomposition was implemented as complete as possible.

Based on the work discussed above, the essential mission of solving MCA image decomposition is to find different dictionaries corresponding to different transformations T_k in Eq. 2.9. Starck et al. [192] applied the Discrete Cosine Transform (DCT) to comprise the dictionary for texture and curvelet transform for the structure part. In [188], Starck et. al. proved that Gabor transform was helpful for extracting textures with different orientations and frequencies from an image and they presented to use a Gabor transform to compute the texture component. MCA was extended in [22] to deal with multichannel data (MMCA) based on the idea proposed in [228] and [84] to build alternative source separation methods by sparsity. The input data X were no longer the simple sum of sources but a set of linear combinations of some sources. MMCA brought a stronger and more robust solution than original MCA as long as the MCA hypothesis was verified for morphologically diverse sources. Another algorithm for multichannel morphological component analysis was generalized morphological component analysis (GMCA), which was proposed in [72]. Fadili et. al. assumed that GMCA was an extension of MMCA that each source was modelled as the linear combination of morphological components where each component was sparse in a specific basis. GMCA was able to produce better separation results than MMCA because it had greater morphological diversity, better sparsity and used sparse, over-complete (redundant) representations. Further, in [189] Starck et. al. found that MCA could help ICA to some extent because ICA aimed at enforcing the independence of the estimated sources and suffered from a certain lack of robustness in the presence of noise while MCA performed well even in the presence of noise for morphologically different sources. They proposed the ICA-MCA method that took advantages of both ICA, which enforced the statistical independence of the estimated sources, and MCA, which gave more contrast between the sources as it extracted the essence of each one.

Table 2.3 summarizes several most typical image decomposition methods, including their categories, capabilities and deficiencies. Image decomposition based on component analysis can provide a more and more adequate and accurate performance so far, but the decomposition is still based on "structure+texture+noise". There is still no convincing work in decomposing a texture into different components according to some

Method	Type	Capability	Deficiency
ROF model [169]	method based on mathematical models	separate the structural parts and texture or noise	can't separate different types of textures
Meyer's model [139]	method based on mathematical models	separate the structural parts and oscillations	can't separate different types of textures
directional filter bank [15]	method based on filter banks	find components representing directional edges	can't separate textures different in characteristics other than directionality
wavelet [7]	method based on filter banks	find components representing low-frequency parts and high-frequency parts with different directions	can't separate textures different in characteristics other than directionality and resolution
curvelet [63]	method based on filter banks	find components smooth apart from singularities along smooth curves	can't separate different types of smooth regions
contourlet [61]	method based on filter banks	find components in any resolution and any direction	can't separate textures different in characteristics other than directionality and resolution
ICA [49]	method based on component analysis	find components representing different pre-learned texture features	can't separate textures different in characteristics other than the pre-set ones
MCA [189]	method based on component analysis	find structural parts, textural parts and noise	can't separate different types of textures

Table 2.3: Summary of several most popular image decomposition methods.

characteristics, in which case the texture cannot be processed by manipulating the components composing the texture. However, all the theoretical methods and practical work proposed above for image decomposition are foundational work for our hypothesis in expressing different characteristics of texture and develop different dictionaries corresponding to characteristics in the decomposition efficiently.

2.3 Image manipulation

Image manipulation is defined as the method of transforming the intensities of the pixels in an image by some functions or operations. Since we consider the components of the texture as a set of independent images, methods of image manipulation can be applied to modify the textural components so as to modify the texture. As the proposed hypothesis, texture can be decomposed into different pairs according to different characters and each sub-image of each pair represents the relative effect of that character (e.g. fine vs. coarse, directional vs. non-directional, high-contrast vs. low-contrast), so we only need to enhance each of the sub-images to increase the difference between different textures with respect to some characteristics. Therefore, popular methods in image manipulation, particularly in image enhancement, are reviewed in the following part.

2.3.1 Linear image manipulation

Some classical intensity transformation methods were proposed to enhance the image contrast and so image details by increasing the range of the intensity values in the image. Image normalization [155] was proposed to transform an n-dimensional grayscale image I with intensity values in the range (Min, Max) into a new image I_N with intensity values in the range $(newMin, newMax)$ by a linear transformation. Histogram equalization was proposed in [229] to remap pixel intensities. It spread the pixel intensities over the entire available dynamic range and made use of as many different intensities as possible and the discrete equalization transformation was expressed as: $s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$ for $k = 0, 1, 2, \ldots, L-1$, where s_k was the output intensity value corresponding to input value r_k. Some other non-linear transformations can also be used to expand the intensity range and image contrast, including logarithmic transformation [4], of which the expression was $v = c \log(1 + u)$, power-law transformation [133], with the expression of $v = c \cdot u^{\gamma}$, and contrast-stretching transformation [216], with the form of $v = \frac{1}{1+(m/u)^E}$.

The intensity transformations described above may cause loss of information because pixels of different intensities might be mapped to the same one. Another disadvantage was that these operations didn't consider the spatial information of the processed pixel and the relationship between the pixel and its neighbours. Since most image features (edge, textures, etc) involved a spatial neighbourhood of pixels, spatial filtering methods that processed all pixels in the neighbourhood as a whole have attracted more attention.

Image derivate filters [182] manipulated the input image to its gradient image, which were proposed to enhance some features of the image. Some famous example of derivative filters were: the first-order derivative filter, Sobel operator [212] and Roberts operator [166] for edge extraction; the second-order derivative filter,

Laplacian operator [207] for edge sharpening or detail enhancement and the Laplacian of Gaussian (LOG) filter, which convolved the Gaussian smoothing filter with the Laplacian filter to reduce the high frequency noise components prior to the different step so as to sharpen the edge without increasing the noise. Sticks filter bank proposed by Czerwinski et al. [56] consisted of a set of $2n-2$ square binary valued linear filter masks of size $n \times n$, each responding maximally to a length n line segment of different orientation. Each mask in a sticks filter bank represented the hypothesis of a line segment in a particular orientation in the maximum likelihood formulation of the line and boundary detection problem. It has proved to be successful in enhancing images for boundary detection because it assigned the maximal average only to the pixels that on the boundary. Gaussian sticks were proposed in [70] to improve the contrast between linear features and background based on uniform sticks. Instead of using uniform distribution along the line orientation, Gaussian sticks applied Gaussian distributed masks along different orientations to build the filter bank. It proved to be able to increase the difference between two textures with respect to some GLCM characteristics.

2.3.2 Non-linear image manipulation

The linear spatial filtering methods discussed above had the disadvantage of loss of adaptivity because of using the same filter mask or set of filter masks to convolve the input image without considering different image structures locally. The following spatial filtering methods based on non-linear processing enlightened our proposed hypothesis a lot.

Coherence-enhancing anisotropic diffusion was proposed by Cottet and Germain [53] as a modification of traditional anisotropic diffusion. The basic idea was to smooth the image anisotropically along the flow field such that gaps can be closed. Cottet et. al. presented that their diffusion method encouraged smoothing along the coherence orientation and was therefore well-suited for closing interrupted textures. Fig. 2.1 shows the effect of enhancing fingerprint texture by coherence-enhancing anisotropic diffusion. Due to its reduced diffusivity at non-coherent structures, the locations of the semantically important singularities in the texture would remain the same.

Shock filtering was another type of morphological image enhancement methods, and most of the current shock filters were based on modifications of Osher and Rudin's formulation in terms of partial differential equations (PDEs) [151]. They created strong discontinuities at image edges, and within a region the filtered signal became flat, therefore, shock filtering can provide good performance where edge sharpening and a piecewise constant segmentation was desired, as shown in Fig. 2.2. J. Weickert et al. [211] proposed the so-called coherence-enhancing shock filters. The filters combined the stability properties of shock filters with the possibility of enhancing flow-like structures. It was achieved by steering a shock filter with the orientation information that was provided by the so-called structure tensor [19, 74, 162]. They reported their filter acted like a contrast-enhancing shock filter perpendicular to the flow direction, while it created a constant signal along the flow direction by applying either a dilation or an erosion process.

Fractional differential operator based method was another important attempt to enhance image textures,

(a) original image (b) enhanced image

Figure 2.1: Texture enhancement by diffusion filtering. (a) is the original fingerprint image, and (b) is the image enhanced by the coherence-enhancing diffusion filter [209]. The texture of the fingerprint is enhanced with more smooth, continuous texture flows and less noise.

(a) original image (b) enhanced image

Figure 2.2: Texture enhancement by shock filtering. (a) is the original image, and (b) is the image enhanced by the coherence-enhancing shock filter [210]. The subtle differences in the blurred texture regions are enhanced with the sharp shocks of colours orthogonal to the directions of the texture flows.

because fractional differential operator was well known for its capability of characterizing fractal-like patterns [165] that typically composed texture regions. Pu et. al. [218] have developed an $n \times n$ fractional differential operator mask after analyzing the geometric and physical properties of different ractional differential operators. By convoluting the mask to the image in different directions, the proposed method could nonlinearly preserve the low-frequency contour feature in the smooth area to the furthest degree, and nonlinearly enhance high-frequency areas including edges and details. Their experiments proved the proposed method enhanced texture better than traditional integral-based differential operators [159]. Pu's work had the problem that texture regions with high self-similarity were not well characterized and it was difficult to find the optimal fixed fractional derivative order matching local texture details. In [96], Hu et. al. proposed an adaptive fractional differential operator mask to overcome this drawback. The proposed method could adaptively determine the fractional order by integrating a non-regular support region so that the texture enhancement performance can be optimized regardless of whether the regions consist of high or low frequency patterns.

(a) original image (b) enhanced image

Figure 2.3: Texture enhancement by fractional differential operator. (a) is the original image, and (b) is the image enhanced by the fractional differential operator [96]. The contour features in the smooth area are preserved while the edges and details are enhanced by convolving the fractional differential operator mask to the image.

Other than spatial filtering, there were also image enhancement methods implemented in transform domain. They first converted the image by different transforming methods, then modified the coefficients in the corresponding domains, and finally transformed the image back to the spatial domain using the modified coefficients. The most widely-used methods in image enhancement were methods based on wavelet transform. The most popular wavelet-based filtering was the soft-thresholding, which was first proposed in [62]. The authors "shrank" or "expanded" the wavelet coefficients above a threshold T by the absolute value of the threshold itself, therefore the noise would be restrained, the details would be enhanced, and the main structure wouldn't be changed. Based on their theory, a diversity of wavelet thresholding methods were proposed according to different choices of threshold and ways of modifying coefficients [95, 173, 206, 128].

The categories, capabilities and deficiencies of the most typical image manipulation methods are listed

Method	Category	Capability	Deficiency
Histogram equalization [229]	linear manipulation	enhance the contrast of the image	degrade the spatial information of the processed pixel and the relationship between the pixel and its neighbours
Gaussian sticks filtering [70]	linear manipulation	enhance the edges with different orientations	degrade the texture details without explicit orientations
Unsharp mask filtering [158]	linear manipulation	sharpen the edges in the image	enhance all types of textures to the same extent
Coherence-enhancing diffusion filtering [53]	non-linear manipulation	enhance the continuity of the texture flows and remove noise	remove some isolated texels as noise
Shock filtering [211]	non-linear manipulation	enhance the subtle differences between textures by adding shocks orthogonal to the directions of the texture flows	over-manipulate the shapes of textures because of the additive shocks
Fractional differential operator filtering [159]	non-linear manipulation	enhance the texture details by finding the fractal-like structures in the image	enhance all the textures using the same fixed fractional derivative order without characterizing different types of textures
Wavelet-based filtering [62]	non-linear manipulation	enhance the texture details by modifying the corresponding high-frequency coefficients	only enhance the directional texture edges by the wavelet coefficients in the corresponding sub-bands

Table 2.4: Summary of several typical image manipulation methods.

(a) original image (b) enhanced image

Figure 2.4: Texture enhancement by wavelet-based shrinkage method. (a) is the original image, and (b) is the image enhanced by the wavelet-based shrinkage method [47]. The noise is restrained and the edges are preserved by shrinking the wavelet coefficients above the threshold.

in Table 2.4. From the discussion above, non-linear manipulation method fits our proposed hypothesis best because it can accentuate the texture features in the image. However, the image manipulation with non-linear filter is still a quite open problem so far. The enhancement methods mentioned above still degrade some parts of the image while enhance the interesting region. Further, these methods focus on how to enhance texture itself as a total but not each particular texture characteristics. However, the theoretical methods proposed above enlighten what we will do in the following chapters.

Chapter 3

Human visual perceptual texture description

3.1 Introduction

As we discussed in Chapter 2.1.2, the human visual perceptual texture descriptors are designed to reflect how human beings perceive different types of textures. However, one drawback of the existing descriptors is the incorrect calculation of the high-level texture characteristics by using the low-level statistical measurements [103]. For example, the Tamura descriptor's directionality was tested in [143] and the results demonstrated that sometimes it gave higher directional score to images having edges with random directions than images with a dominant edge orientation. Moreover, most of the descriptors capture the texture characteristics globally for the given image, so they cannot precisely describe the variances of the texture characteristics in the textural images with multiple textures, e.g., two completely different images each with a 50% black and 50% white pixels have idential gray level histograms [186], therefore the Tamura's contrast, which is based on the variance and the kurtosis of the gray level histogram of the whole image, cannot distinguish between them.

Another shortcoming of the human visual perceptual texture descriptors is they capture more characteristics than necessary. Tamura concluded that coarseness, contrast and directionality were the essential factors in texture and had high potential to distinguish different textures [196]. But this result was based on empirical and subjective experiments, i.e., they let human subjects rank textures with respect to the given characteristics and approximated the correlation between each characteristic. Therefore, it is necessary to design a more objective, mathematical algorithm to reduce the dimensionality of the feature vector extracted from the image and extract the essential characteristics to describe the different textures in the image.

Dimensionality reduction is the process of reducing the number of random variables under consideration to obtain a set of "uncorrelated" principal variables [168]. Most of the dimensionality reduction techniques can be divided into feature extraction and feature selection [160]. Feature extraction approaches transform data in a high-dimensional space to a lower-dimensional space. The transformations can be both linear (such as in principal component analysis (PCA) [109], linear discriminant analysis (LDA) [136], and maximum margin criterion (MMC) [120]), and non-linear (such as kernel PCA, locally linear embedding (LLE) [168], and isomap [197]). However, transformed features usually lack obvious semantic meaning, making it difficult to understand the image and to analyze image components representing different high-level visual characteristics.

Feature selection techniques have proven to be more effective than feature extraction approaches for dimensionality reduction of image features because they pick a subset of the original features rather than to find a mapping that uses all of the original features. Some early feature selection methods have been proposed and applied in different situations. Among them, the regression area has been investigated extensively. In [126], a multi-layer perception is used for variable selection. In [125], step-wise discriminant analysis for variable selection is used as inputs to a neural network that performs pattern recognition of circuitry faults. Other regression techniques for variable selection are described in [93]. In contrast to the regression methods, which lack unified optimality criteria, the optimality properties of PCA have attracted research on PCA-based variable selection methods [138, 109, 114, 115]. However, these methods have the disadvantage of either being too computationally expensive, or choosing a subset of features with redundant information.

In this chapter, we improve the Tamura's texture descriptor by measuring each of the six texture characteristics with novel local metrics so that these measurements reflect the human perception of each characteristics more precisely. The local calculations of the texture characteristics make it possible to distinguish between images with different textures but similar histograms. Then we propose a PCA-based feature selection method exploiting the structure of the principal components of the feature set to find a subset of the original feature vector, where the features reflect the most representative characteristics for the textures in the given image dataset.

3.2 Local Tamura's texture description

3.2.1 Tamura's texture features

In [196], six human visual perceptual texture features were proposed by Tamura, Mori and Yamawaki on the basis of psychological experiments. The six textural features are: coarseness, contrast, directionality, line-likeness, regularity and roughness. The computations of these features are reviewed as follows.

Coarseness

Coarseness relates to spatial distances between obvious variations of gray levels. In other words, it is the measurement of the size of the primitive elements (texels) composing the texture. The computational procedure is defined in the following steps:

Step 1: Calculate averages at every point over neighbourhoods whose sizes are powers of two, e.g., 1×1, 2×2,4×4,..., 32×32. The average over the neighbourhood of size $2^k \times 2^k$ at the pixel (x, y) is:

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i,j)/2^{2k}, \tag{3.1}$$

where $f(i, j)$ is the intensity at (x, y).

Step 2: Calculate differences between pairs of averages corresponding to pairs of non-overlapping neighbourhoods just on opposite sides of each pixel in both horizontal and vertical orientations. For example, the difference in the horizontal case is:

$$E_{k,h}(x,y) = \left| A_k\left(x + 2^{k-1}, y\right) - A_k\left(x - 2^{k-1}, y\right) \right|. \tag{3.2}$$

Step 3: At each pixel, pick the neighbourhood size which generates the highest output value:

$$S_{best}(x,y) = 2^k, \tag{3.3}$$

where k maximizes E in either direction, i.e.,

$$E_k = E_{max} = \max\left(E_1, E_2, \cdots, E_L\right). \tag{3.4}$$

Step 4: Finally, calculate the average of S_{best} over the picture to be a coarseness measurement F_{crs}:

$$F_{crs} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{best}(i,j), \tag{3.5}$$

where m and n are the effective width and height of the image, respectively.

Contrast

In Tamura's work, they assumed four factors that influenced the contrast difference between two texture patterns with different structures:

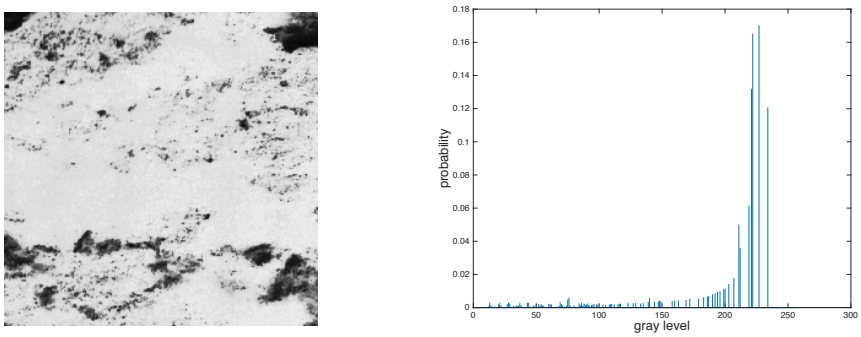
1. dynamic range of gray levels,
2. polarization of the distribution of black and white on the gray level histogram or ratio of black and white areas,
3. sharpness of edges,
4. period of repeating patterns.

However, they approximated only Factors 1 and 2 (to generate quite successful results for their testing texture pattern) as follows:

For Factor 1, they used the variance of pixel intensities σ^2 or standard deviation σ about the mean of the gray levels probability distribution to reflect the dynamic range of the gray levels of the image. However, the resultant value was undesirable for a distribution where a single peak is highly biased to black or white as shown in Fig. 3.1.

For Factor 2, they applied the kurtosis α_4 of the intensity histogram as a measurement of polarization, which is defined as:

$$\alpha_4 = \frac{\mu_4}{\sigma^4}, \tag{3.6}$$



(a) D7 texture in Brodatz database [24] (b) Histogram of the D7 texture in (a)

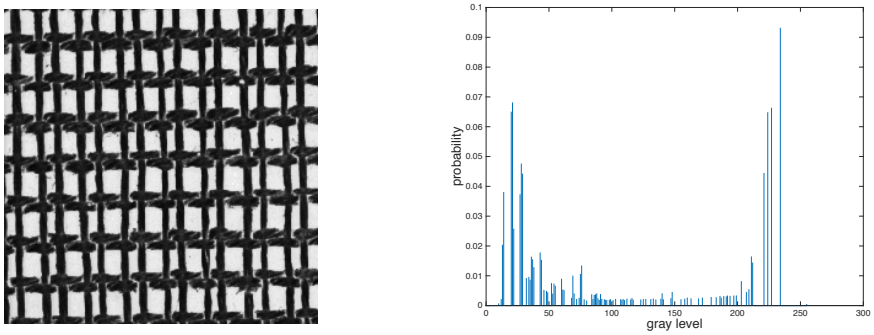
Figure 3.1: Example of the texture with the gray level distribution where a single peak is highly biased to white. (a) is the part of the D7 texture in the Brodatz database [24], (b) is the distribution of the gray levels of the pixels in the image.

where μ_4 is the fourth moment about the mean and σ^2 is the variance. The measurement is normalized with respect to the range so that it can have the minimum value of one in case of twin peaks.

The standard deviation σ and the kurtosis α_4 are combined for the measurement of contrast as follows:

$$F_{con} = \sigma / (\alpha_4)^n , \tag{3.7}$$

where n is a positive number. In their experiments, they varied n from 8, 4, 2, 1, 1/2, 1/4, to 1/8. Experimentally, $n = 1/4$ yielded the best correlation between the human measurement and the computational measurement. This calculation of contrast could reduce the values for distributions with biased peaks while almost preserving those for polarized distributions as shown in Fig. 3.2.



(a) D20 texture in Brodatz database [24] (b) Histogram of the D20 texture in (a)

Figure 3.2: Example of the texture where the gray levels distribute into two polarized peaks. (a) is the part of the D20 texture in the Brodatz database [24], (b) is the distribution of the gray levels of the pixels in the image.

Directionality

In Tamura's descriptor, directionality was measured using the histogram of directional angles of the oriented local edges, i.e., the frequency distribution of oriented local edges against their directional angles. This

method utilizes the fact that a gradient is a vector, so it has both magnitude and direction. In the discrete case, the magnitude $|\Delta G|$ and the local edge direction θ are approximated as follows:

$$\begin{aligned} |\Delta G| &= \left(|\Delta_H| + |\Delta_V|\right)/2 \\ \theta &= \tan^{-1}\left(\Delta_V/\Delta_H\right) + \tfrac{\pi}{2} \end{aligned}, \tag{3.8}$$

where Δ_H and Δ_V are the horizontal and vertical differences measured by the following 3×3 moving window operators:

$$\begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{array} \tag{3.9}$$

The resultant θ is a real number ($0 \leq \theta < \pi$) measured counterclockwise so that the horizontal direction is 0.

The desired histogram H_D is obtained by quantizing θ and counting the pixels with the magnitude $|\Delta G|$ over the threshold t as:

$$H_D\left(k\right) = N_\theta\left(k\right) / \sum_{i=0}^{n-1} N_\theta\left(i\right), \qquad k = 0, 1, \cdots, n-1, \tag{3.10}$$

where $N_\theta\left(k\right)$ is the number of points where $\left(2k-1\right)\pi/2n \leq \theta < \left(2k+1\right)/2n$ and $|\Delta G| \geq t$. Thresholding $|\Delta G|$ by t prevents counting directions of the pixels that are too weak to be regarded as edge pixels.

By computing the sharpness of the peaks of the direction histogram H_D, the directionality was measured quantitatively, defined as follows:

$$F_{dir} = 1 - r \cdot n_p \cdot \sum_{p}^{n_p} \sum_{\phi \in w_p} \left(\phi - \phi_p\right)^2 \cdot H_D(\phi), \tag{3.11}$$

where n_p is the number of peaks, ϕ_p id the p-th peak position in the histogram H_D, w_p is the range of p-th peak between valleys, ϕ is the quantized direction angle (cyclically in modulo 180°), while r is a normalizing factor related to the number of quantization levels of ϕ.

Line-likeness

Tamura defined the word line-likeness as a characteristic of texture that is composed of lines. For this purpose, when the direction and the neighbouring edges' directions for a given edge are nearly equal, they regarded such a group of edge pixels as a line.

To be more specific, a direction co-occurrence matrix was constructed whose element $P_{Dd}(i, j)$ was defined as the relative frequency with which neighbourhoods centred at two pixels separated by a distance d along the edge direction occur on the image, one with the quantized direction i and the other with the quantized direction j. And as they calculated the directionality, it is preferable to ignore directions of trivial edges by using the threshold t.

Therefore, as the measure of line-likeness, the following measurement is defined so that co-occurrences in the same direction are weighted by $+1$ and those in the perpendicular direction by -1:

$$F_{lin} = \sum_{i}^{m} \sum_{j}^{n} P_{Dd}(i, j) \cos[(i-j)\frac{2\pi}{n}] / \sum_{i}^{m} \sum_{j}^{n} P_{Dd}(i, j), \tag{3.12}$$

where P_{Dd} is the $n \times n$ local direction co-occurrence matrix of pixels at distance d.

Regularity For natural textures, it is difficult to measure the degree of irregularity because the natural images are always without any information such as texture element (texel) size or shape. Therefore, Tamura assumed that if any feature of a texture varies over the whole image, the image is irregular. Hence they took partitioned sub-images from the input image and considered the variation of each feature in each sub-image. They took the sum of the variation for each of the four features defined above as the measurement of regularity:

$$F_{reg} = 1 - r(\sigma_{crs} + \sigma_{con} + \sigma_{dir} + \sigma_{lin}), \tag{3.13}$$

where r is the normalizing factor and each σ_{xxx} means the standard deification of F_{xxx}.

Roughness According to the results of their psychological experiments on vision, Tamura emphasized the effects of coarseness and contrast on roughness, and approximated a measurement of roughness by:

$$F_{rgh} = F_{crs} + F_{con}. \tag{3.14}$$

Their intention lied in examining to what an extent such a simple approximation corresponds to human visual perception.

From their own experiments [196] and the following research [134], Tamura's texture descriptor has proved to be successful in reflecting the human visual perception on textures and bridging the semantic gap [226] between low-level features (e.g., numerical vectors) and high-level concepts (e.g., coarseness/fineness, line-like/blob-like). However, all the six texture characteristics from Tamura's descriptor describe the textural image globally, resulting in difficulties in discriminating the different textures in the image. Therefore, in the next chapter, Tamura's descriptor is improved to represent texture features locally, still following the definition and the understanding of the texture characteristics.

3.2.2 Local Tamura's texture features

This chapter will focus on extending Tamura's features for local texture description. When extending descriptors from global to local, a common method is: for each pixel, treat the neighbourhood centred at it as an image, then apply the texture descriptor to this sub-image and assign the feature vector to the centre pixel. However, for some of the Tamura's features, this simple "global to local" method is not applicable because the global features are calculated from the relations among the adjacent image blocks that cannot be defined in a local neighbourhood. Therefore, we research the definition of each feature and calculate them by the low-level features that measures the relation among pixels in the local window centred at each pixel.

Coarseness

Coarseness is defined to measure the size of the texels in the image. As shown in Fig. 3.3(a) and Fig. 3.3(g), larger texels composing the texture result in higher coarseness value. Equally, it can be considered to

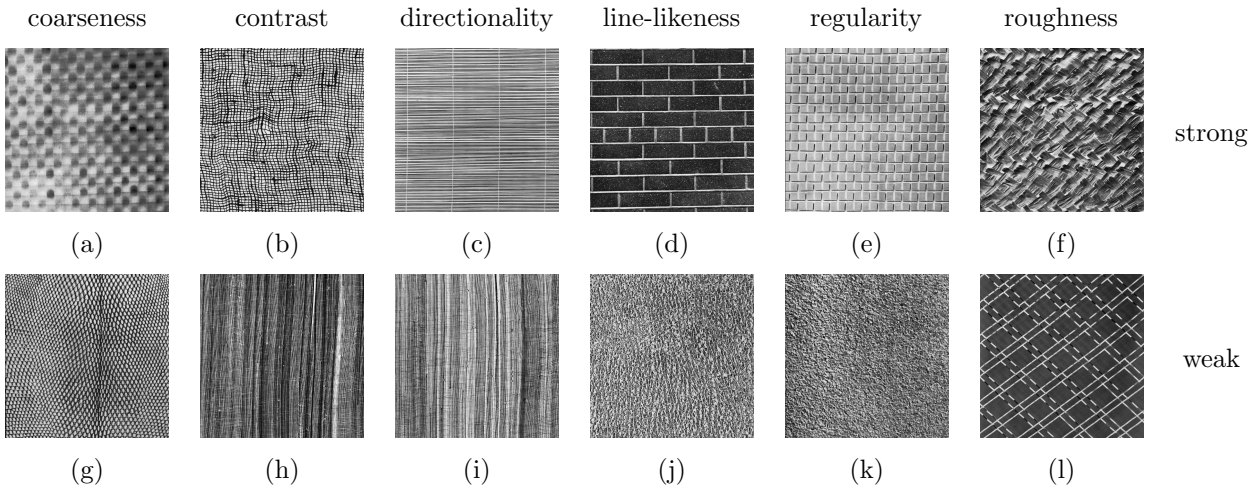


Figure 3.3: Example textures exhibiting the six Tamura's texture features strongly and weakly. Columns (from left to right): textures with high (upper row) and low (lower row) values with respect to coarseness, contrast, directionality, line-likeness, regularity and roughness.

measure the density of the texels in the neighbourhood: the more texels are there in the local window, the smaller sizes the texels have, and the lower the coarseness should be. Therefore, the computation procedure is defined as follows:

Step 1: Calculate the edges in the given image I by applying the Laplacian of Gaussian (LoG) filter [185] to the image and finding the zero crossings, i.e., the edge detection criterion is the presence of a zero crossing in the second derivative with the corresponding large peak in the first derivative. The Laplacian of Gaussian filter is defined as follows:

$$\mathrm{LoG} = \nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.15}$$

where σ is the standard deviation of the Gaussian, which is given as:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3.16}$$

Step 2: Connect the edges by the morphological transformation that sets 0-valued pixels to 1 if they have two non-zero neighbours that are not connected [91], such as

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \text{ becomes } \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{3.17}$$

After the edge connection, the edges that surround the smooth regions as "texels" are mostly detected from the image.

Step 3: Calculate the "texels" image $\widehat{L}$ as the negative transformation of the connected edge image E':

$$\widehat{L} = 1 - E'. \tag{3.18}$$

Then the "texels" are labelled in the image $\widehat{L}$.

Step 4: The coarseness is calculated as the ratio of the number of different texels to the number of the pixels in the neighbourhood:

$$F_{crs} = \frac{\text{diff}(\widehat{L})}{M \times N}, \tag{3.19}$$

where $\text{diff}(\widehat{L})$ calculates the number of different "texels" in the neighbourhood of $\widehat{L}$, M and N are the size of the neighbourhood.

Contrast

Based on the discussion in Chapter 3.2.1 and the observation of the textures with different contrast, as shown in Fig. 3.3(b) and Fig. 3.3(h), the following factors are considered to influence the contrast difference between different texture patterns:

1. the range of gray levels: the textures where the distances between the maximum and minimum gray level is large are more likely to have higher contrast;
2. the concentration of the distribution of the gray level histogram: the textures where the histogram distributes uniformly in a wide range will exhibit higher contrast;
3. the polarization of the distribution of the gray level histogram: a small difference of intensity is negligible if the intensities are biased to 255, while the same small difference matters if the intensities are biased to 0.

Therefore, the contrast of the image can be measured locally as:

$$F_{con} = \frac{r_d}{p}, \tag{3.20}$$

where r_d corresponds to the dynamic range of the gray levels between the maximum intensity I_{max} and the minimum intensity I_{min}:

$$r_d = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}, \tag{3.21}$$

and p combines the skewness $\text{skew}(\cdot)$ and the kurtosis $\text{kurt}(\cdot)$ of the histogram of the pixel intensities in the neighbourhood I_l to measure the concentration and the polarization of the distribution of the intensities:

$$p = \frac{\text{kurt}(I_l)}{1 + \text{skew}(I_l)}. \tag{3.22}$$

Directionality

The concepts of directionality and line-likeness in Tamura's texture descriptor are obscured because the directionality measures the edges with similar directions while the line-likeness measures the coincidence of the edge directions. It can partly explain the incorrectness of the Tamura's descriptor in describing directionality for some textures with random directions [103]. Therefore, in this thesis, the concept of directionality is redefined as the measurement of the mean orientation of all the edges in the local texture. Moreover, the

horizontal orientation is defined to have high directionality value, as shown in Fig. 3.3(c), while the vertical orientation is defined to have low directionality value, as shown in Fig. 3.3(i). Then the directionality of a texture pixel is calculated in its neighbourhood as:

$$F_{dir}(x,y) = \tan^{-1}\left(\frac{\sum_{x=1}^{M}\sum_{y=1}^{N} m(x,y)\cos d(x,y)}{\sum_{x=1}^{M}\sum_{y=1}^{N} m(x,y)\sin d(x,y)}\right), \tag{3.23}$$

where M and N are the size of the neighbourhood, $m(x,y)$ and $d(x,y)$ represent the magnitude and direction of the edge pixel (x,y), which are calculated as:

$$\begin{aligned} m(x,y) &= \sqrt{\delta_x(x,y)^2 + \delta_y(x,y)^2} \\ d(x,y) &= \tan^{-1}(\tfrac{\delta_y(x,y)}{\delta_x(x,y)}) + \tfrac{\pi}{2} \end{aligned}, \tag{3.24}$$

where the $\delta_x(x,y)$ and $\delta_y(x,y)$ represent the pixel-wise horizontal and vertical derivatives using the Sobel operator [183] to the neighbourhood centred at (x,y).

Line-likeness

The line-likeness measures how many of the edges in the neighbourhood have the same or similar directions. Fig. 3.3(d) shows the texture with high line-likeness where most of the edges are in the similar direction, while Fig. 3.3(j) shows the texture with low line-likeness because of the edges with various directions. The directions at the edge pixels $d(x,y)$ are quantized into n direction intervals $[0, \frac{\pi}{n}), [\frac{\pi}{n}, \frac{2\pi}{n}), \ldots, [\frac{(n-1)\pi}{n}, \pi)$. As a result, the line-likeness of the pixel (x,y) is calculated as 1 minus the variance of the local edge orientations:

$$F_{lin}(x,y) = \left(1 - \frac{\sum_{i=x-w}^{x+w}\sum_{j=y-w}^{y+w}\left(D\left(i,j\right) - \mu_D\right)^2}{\left(2w+1\right)^2}\right), \tag{3.25}$$

where $D(x,y)$ is the quantized direction of the edge direction $d(x,y)$, μ_D is the mean value of D in the local window, w is the radius of the window.

Regularity

The regularity of the texture measures the spatially repetitiveness of texels over a certain distance. For a regular texture shown in Fig. 3.3(e), the texel edges repeat every several pixels, and the distance of this repetitiveness is relatively fixed in contrast to an irregular texture as shown in Fig. 3.3(k). Therefore, this pattern can be calculated by the auto-correlation functions of the edge image as follows:

$$F_{reg}(x,y) = 1 - \frac{\max(C)}{\sum_{i=1}^{N}(C(i))}, \tag{3.26}$$

where C is the auto-correlation function of the neighbouring image centred at (x,y):

$$C = \mathrm{CORR}(I_l, I_l), \tag{3.27}$$

where $\mathrm{CORR}(\cdot)$ represents the auto-correlation function, I_l is the local window of the image I. Since the maximum of the auto-correlation function C will only appear when the displacement of the texture is 0, and

the regular texture where there must exist another displacement of the repeating texels will have other local peak values in the auto-correlation function, the ratio of the maximum auto-correlation value to the sum of all the local peak auto-correlation values $\frac{\max(C)}{\sum_{i=1}^{N}(C(i))}$ will be smaller in regular texture than that in irregular one, leading to a larger measurement $F_{reg}(x, y)$.

Roughness

As shown in Fig. 3.3(f), a rough texture means the surface of the texture is uneven and not smooth, and contains many edges, while the texture with quite smooth surface as shown in Fig. 3.3(l) has the low value of roughness. Therefore, the roughness can be approximated by the density of the edges in the neighbourhood:

$$F_{rgh}(x, y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} E(i, j)}{M \times N}, \tag{3.28}$$

where the E represents the edge map of the neighbourhood centred at (x, y) with the size $M \times N$. Similar to the calculation of coarseness, the edge is computed by the Laplacian of Gaussian (LoG) filter.

As discussed above, the six human visual perceptual texture characteristics proposed in Tamura's work have been locally calculated with novel methods from low-level texture features. Therefore, the texture image can be described more precisely according to human visual perception and different types of textures can be better differentiated locally.

3.3 PCA-based key characteristics selection in texture description

From the above discussion, textural images can be locally described by the six improved Tamura's texture features. Using all these features in describing textural images, however, is sometimes redundant because these features are correlated to each other to some extent. In this part, a PCA-based discriminant texture feature selection method is proposed to find the representative characteristics in differentiating textures in the image.

3.3.1 Principal component analysis (PCA) in reducing dimensionality of feature vector

Principal component analysis (PCA) is one of the most popular methods for dimensionality reduction [214]. As described in [80], for a set of data n dimensions, PCA aims to find a linear subspace of dimension p lower than n so that the data points mostly lie in this subspace, which is likely to maintain most of the variability of the original data. The linear subspace can be specified by p orthogonal vectors that form a new coordinate system, called the 'principal components' [205]. The principal components are orthogonal, linear transformations of the original data points, so there can be no more than n of them. However, the expectation is that only $p < n$ principal components are needed to approximate the space spanned by the n original axes [181].

Mathematically, the transformation is defined [1] by a set of p-dimensional vectors of weights or loading $\mathbf{w_{(k)}} = (\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_p})_{\mathbf{(k)}}$ that map each row vector $\mathbf{x_{(i)}}$ of X to a new vector of principal component scores $\mathbf{t_{(i)}} = (\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_k})_{\mathbf{(i)}}$, given by:

$$t_{k(i)} = \mathbf{x_{(i)}} \cdot \mathbf{w_{(k)}}. \tag{3.29}$$

In such a way that the individual variables of $\mathbf{t}$ considered over the data set successively inherit the maximum possible variance from $\mathbf{x}$, with each weights vector $\mathbf{w}$ constrained to be a unit vector. Therefore the full principal components transformation of $\mathbf{x}$ can be given as:

$$T = XW, \tag{3.30}$$

where W is a $p \times p$ matrix whose columns are the eigenvectors of $x^T x$.

3.3.2 Reduction of image feature dimensionality

According to Eq. 3.30, the transformation $\mathbf{T} = \mathbf{XW}$ maps a data vector $\mathbf{x}_{(i)}$ from an original space of p variables to a new space of p variables which are de-correlated over the dataset. However, not all the principal components need to be kept. Keeping only the first L principal components, produced by using only the first L loading vectors, gives the truncated transformation:

$$\mathbf{T}_L = \mathbf{XW}_L, \tag{3.31}$$

where the matrix $\mathbf{T}_L$ now has n rows but only L columns. In other words, PCA learns a linear transformation $t = W^T x,\ x \in R^p,\ t \in R^L$, where the columns of $p \times L$ matrix W form an orthogonal basis for the L features (the components of representation t) that are de-correlated [17]. By construction of all the transformed data matrices with only L columns, this score matrix maximizes the variance in the original data that has been preserved, while minimizing the total squared reconstruction error $\left\|\mathbf{TW}^T - \mathbf{T}_L\mathbf{W}_L^T\right\|_2^2$ or $\|\mathbf{X} - \mathbf{X}_L\|_2^2$.

Following this transformation scheme, PCA has been widely used to reduce the dimensionality of the image features selected by some certain image descriptor [113, 221, 34, 131, 2]. Given an image with size $M \times N$ and a local descriptor that extract p features for each pixel in the image. Then the data matrix $\mathbf{X}$ can be considered as:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}, \tag{3.32}$$

where $n = M \times N$ is the total number of the pixels in the image, p is the dimensionality of the original feature space for each pixel. Then according to Eq. 3.31, only the first L principal components that generate the largest variances are preserved and the newly feature matrix T_L is:

$$\mathbf{T}_L = \begin{bmatrix} y_{11} & \cdots & y_{1L} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nL} \end{bmatrix}, \tag{3.33}$$

where $\{y_{i1} \dots y_{iL}\}$ is the newly feature vector for each pixel in the image, with a reduced dimensionality of the image feature.

3.3.3 PCA-based discriminant feature selection

Given the PCA transformation for image feature vector as described in Eq. 3.32 and Eq. 3.33, the "principal" features are the linear combination of the original features with the coefficients matrix:

$$[Y_1 Y_2 \cdots Y_q] = [X_1 X_2 \cdots X_p] \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1q} \\ k_{21} & k_{22} & \cdots & k_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ k_{p1} & k_{p2} & \cdots & k_{pq} \end{bmatrix}, \tag{3.34}$$

where $Y_1, Y_2, \dots, Y_q$ are the q principal components translated from the p original features $X_1, X_2, \dots, X_p (q \leq p)$ by the coefficients matrix.

Assuming that there are n observations of the image with respect to the p features, then we have n sets of the original features with p-dimension. After PCA transformation, the n observations of each of the q features with the reduced dimensionality of p of the original feature space as follows:

$$Y_i = \begin{bmatrix} k_{1i}X_{11} + k_{2i}X_{12} + \cdots + k_{pi}X_{1p} \\ k_{1i}X_{21} + k_{2i}X_{22} + \cdots + k_{pi}X_{2p} \\ \vdots \\ k_{1i}X_{n1} + k_{2i}X_{n2} + \cdots + k_{pi}X_{np} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} \begin{bmatrix} k_{1i} \\ k_{2i} \\ \vdots \\ k_{pi} \end{bmatrix} = \mathbf{X} \begin{bmatrix} k_{1i} \\ k_{2i} \\ \vdots \\ k_{pi} \end{bmatrix} \tag{3.35}$$

where $i = 1, 2, \dots, q$. Since for every i, the observations of the original features $\mathbf{X}$ are the same, the column $[k_{1i}, k_{2i}, \cdots, k_{pi}]^T$ controls the difference between two principal components Y_i, while the absolute value of the $k_{ri}, r = 1, 2, \dots, p$ represents the weight of the r-th original feature to the i-th principal component.

Differently from the original PCA dimensionality reduction which uses $Y_1, Y_2, \cdots, Y_q$ as the features to describe the image, the PCA-based feature selection aims to find a subset of X' of the original data vector X, so that with the corresponding coefficients k', which is also the subset of the original coefficients k, the output component Y' are also the "principal" components. In another word, the orthogonality of the output Y' should remain close to that of the original principal components Y.

In this thesis, the orthogonality of the component matrix Y is defined based on the cosine dissimilarity between each two components Y_i and Y_j in Y:

$$\text{Ortho}(Y) = \frac{\sum_{i=1}^{q} \sum_{j=1}^{q} \text{Diss}(Y_i, Y_j)}{q^2}, \tag{3.36}$$

where q is the number of components in Y, and the cosine dissimilarity $\text{Diss}(\cdot)$ is defined as:

$$\text{Diss}(Y_i, Y_j) = 1 - \frac{\sum_{p=1}^{n} Y_{ip} Y_{jp}}{\sqrt{\sum_{p=1}^{n} Y_{ip}^2} \sqrt{\sum_{p=1}^{n} Y_{jp}^2}}, \tag{3.37}$$

where n is the number of observations of the image data. However, it is difficult to calculate the dissimilarity of the Y_i and Y_j directly because the n of them are always large as the total number of the image pixels and varies in images with different sizes. Therefore we use the weighting coefficients matrix k to measure the dissimilarity because the columns $[k_{1i}, k_{2i}, \cdots, k_{pi}]^T$ and $[k_{1j}, k_{2j}, \cdots, k_{pj}]^T$ which are able to control the difference between two principal components Y_i and Y_j. Then the orthogonality of the component matrix Y can be calculated by the coefficient matrix k as:

$$\text{Ortho}(Y) = \frac{\sum_{i=1}^{q} \sum_{j=1}^{q} \text{Diss}(k_i, k_j)}{q^2}, \tag{3.38}$$

where q is also the number of columns in k corresponding to the components in the Y, and

$$\text{Diss}(k_i, k_j) = 1 - \frac{\sum_{l=1}^{p} k_{li} k_{lj}}{\sqrt{\sum_{l=1}^{p} k_{li}^2} \sqrt{\sum_{l=1}^{p} k_{lj}^2}}, \tag{3.39}$$

where p is the dimensionality of the input features X. Since p is based on the descriptor used to represent the image, p is a constant number for all the images, which simplifies the calculation of the orthogonality. Therefore, the PCA-based feature selection can be mathematically described as:

$$\{s\} = \arg_s \left\{ \text{Ortho}(k_{(s)}) \approx \text{Diss}(k_{(p)}) \right\}, \tag{3.40}$$

where s is the subset of the original features and the number of the dimensionality of the selected subset is l, while p represents the full set of the original features. The algorithm of the PCA-based feature selection is implemented as Algorithm 3.1:

1. implement the PCA to the data vector consisting of the features of the images in the dataset;
2. find all the possible subsets of the coefficients matrix k. Each subset is formed by selecting the rows from k, corresponding to the selected features;
3. calculate the dissimilarities between every two columns of all the possible subsets, then compute the orthogonality of the components matrices represented by those subsets;
4. compare the orthogonality of the possible matrices, and select the most discriminant features represented by the subset of the coefficients matrix that deriving the output components matrix with the orthogonality closest to the original one.

1. Initialize $L = 1$, the number of selected features, $s_{select} = \{1, 2, \ldots, p\}$, the selected features set, ortho $= 0$, the orthogonality of the principal components computed from the selected features, k is the principal component coefficients matrix with p rows corresponding to the original features.
2. Perform $p - 1$ times:
   - Calculate the set s of possible combinations by choosing L features from the original p features and the number of combinations C_p^L.
   - Initialize $i = 1$, the order of the combination in the set s.
   - 3. Perform C_p^L times:
     - Calculate the orthogonality ortho$_{(i)}$ of the subset consisting of the i-th combination of rows in the principal component coefficients matrix k by Eq. \ref{}.
     - If ortho$_{(i)}$ > ortho, update the orthogonality by ortho = ortho$_i$, update the selected features set by $s_{select} = s_{(i)}$. Else, skip.
     - Update the order i by $i = i + 1$.
   - Update the number of selected features L by $L = L + 1$.
4. Return the selected features set s_{select}.

Algorithm 3.1: The algorithm for PCA-based feature selection.

3.4 Experiments and discussion

To evaluate the performance of the proposed PCA-based feature selection method, the features selected by different methods are compared to classify textures in the given testing texture dataset.

3.4.1 Experimental materials

Image database of the experiment

The 112 original texture images from the Brodatz database [24] are randomly divided into 8 groups with 14 textures in each group, so that the experiment is implemented 8 times to classify the 14 textures in each group. Since there are not enough samples of each texture in the original database, the training and testing samples for each texture are generated from the given texture itself. As Fig. 3.5 shows, the training and testing samples for a given texture are generated as follows:

1. each texture with the size of 512×512 is cut into $8 \times 8 = 64$ patches with the size of 64×64, shown as those in Fig. 3.5(a) and (b). We choose 64×64 as the size of each texture patch because it is a good trade off: size larger than 128×128 would result in too few patches of the original image to generate enough texture samples, while size smaller than 32×32 would result in too shattered patches to generate the samples preserving enough original texture information;

2. the 64 patches are randomly combined into 8 rows and 8 columns then merged into one image. This step is repeated 64 times, resulting in 64 images with the same type of texture. Fig. 3.5(c) and (d) show two example randomly generated texture images;

3. 32 of the generated images are used as the training image set for one texture, the other 32 of the generated images are used as the testing image set.

Comparators of the experiment

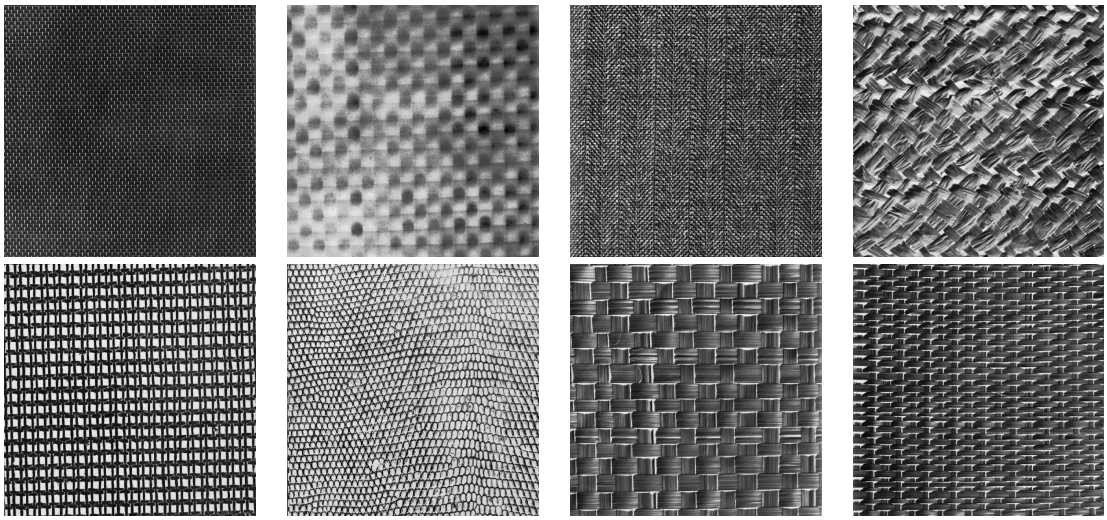


Figure 3.4: The example textures from Brodatz texture database [24].

The following state-of-the-art feature selection methods are used as the comparator to the proposed method in finding discriminant features from the given feature set:

1. the mean square prediction error minimization of PCA method (PCA-MSPEM), which finds the subset of the feature set that minimize the trace of a measure matrix of the covariance matrix of the input feature vector;

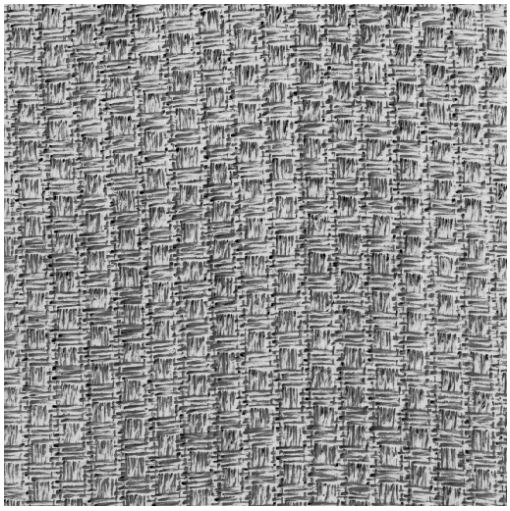
2. the approximation of principal features by using the absolute value of the coefficients of the principal components, which chooses the variables corresponding to the highest coefficients of each of the first q principal components as the principal features from the given ones;

3. the principal feature analysis (PFA) [50], which clusters all the rows in the principal component coefficient matrix by K-Means algorithm, and chooses the feature corresponding to the row vector which is closest to the mean of each cluster as the principal feature.

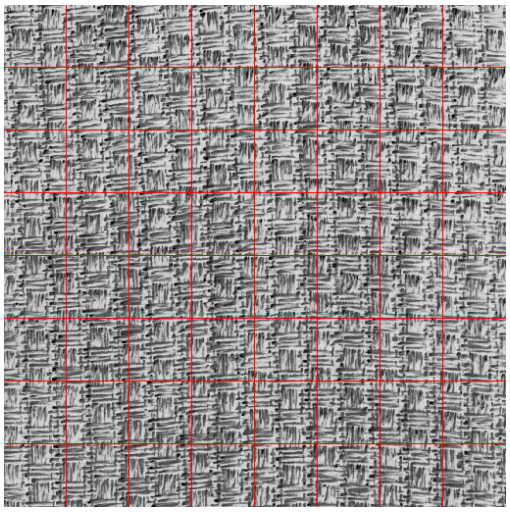
Implementation of the experiment

With the generated training and testing samples of each texture and the given feature selection methods, the experiment of classifying the 14 selected textures is implemented as follows:

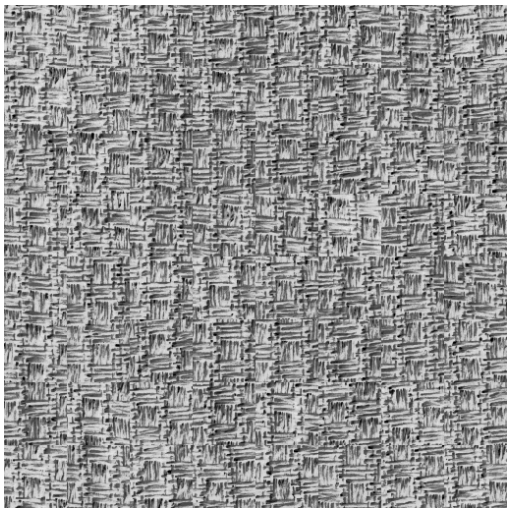
1. the local Tamura's features, consisting of the 6 specific characteristics: coarseness, contrast, directionality, line-likeness, regularity and roughness are calculated in the 7×7 neighbourhood of each pixel in each of the training and testing sample images. 7×7 is a suitable size for the neighbourhood because the size smaller than that would result in incomplete texture information in the neighbourhood, while the size larger than that would result in much more complexity in calculation. The mean value of the feature values for all the pixels in each sample image is considered as the certain feature value for the image, that is, 1×6 feature vector is calculated for each sample image;



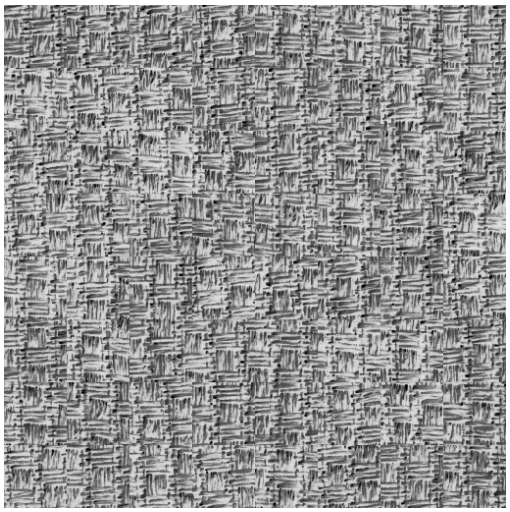
(a) the sample texture



(b) 64 patches of the sample texture



(c) regenerated texture by merging the 64 partitions randomly



(d) another regenerated texture by merging the 64 partitions randomly

Figure 3.5: Example of generating multiple training and testing texture images from a given texture. The texture is cut into $8 \times 8 = 64$ patches with the size of 64×64, then the 64 patches are randomly combined into 8 rows and 8 columns then merged into one texture image. This process is repeated 64 times to get 32 training images and 32 testing images where the textures are the same as the given texture.

2. the most discriminating features are selected from the 6 Tamura's features using the different feature selection methods, then for each sample image a $1 \times q$ ($q \leq 6$) vector of the most discriminating feature for each feature selection method is generated;

3. the $14 \times 32 = 448$ testing samples of 14 textures are classified by various classifiers: (1) Multi-class SVM [213], (2) Random forest (RF) [123], and (3) naive Bayes classifier [145], and the discriminant Tamura's features. The classifiers are trained with the discriminant Tamura's feature vectors of the $14 \times 32 = 448$ training samples;

4. the classification accuracy of each classification is defined as:

$$AR = \frac{TP + TN}{TP + FP + TN + FN}, \tag{3.41}$$

where the traditional TP, TN, FP and FN are described in [5].

5. the classification accuracy of classifying the textures in the dataset with all the features and the principal features selected by different methods are compared to evaluate the effect of different feature selection methods in choosing the most representative features for the textures in a given database.

The experiment described above is implemented 8 times with different groups of textures randomly selected from the Brodatz database [24] to evaluate the general performance of the feature selection methods in discriminate different types of textures.

3.4.2 Experimental results and analysis

The textures in the Brodatz database [24] randomly assigned into Group 1 to Group 8 are shown in Table. 3.1. Table. 3.2 shows the features selected in Group 1 to Group 8 by different methods, together with the comparison of the corresponding running time. Fig. 3.6 to Fig. 3.13 show the classification accuracy of each texture in Group 1 to Group 8 by multi-SVM, naive Bayes and random forest classifier respectively, using the texture features selected by different methods. And Table. 3.3 to Table. 3.5 show the classification accuracies of the textures in Group 1 to Group 8 by different classifiers, using different features selected by different methods. Note that the last three columns in Table. 3.3 to Table. 3.5 show the mean values of the classification accuracies and running times in classifying the textures in Group 1 to Group 8 for a general evaluation of the feature selection performance.

The full-feature description generates the best result in the classification, but with the cost of large computation and running time. The features selected by PCA-MSPEM generates the classification results close to the full-feature classification. However, it preserves 4 or 5 of the 6 original features so the redundancy is still possible to exist. And the computational complexity of finding the principal features is high because the mean square prediction error is calculated for all possible combinations of the q principal features. The approximation based on the magnitudes of coefficients of the principal components is a very intuitive and

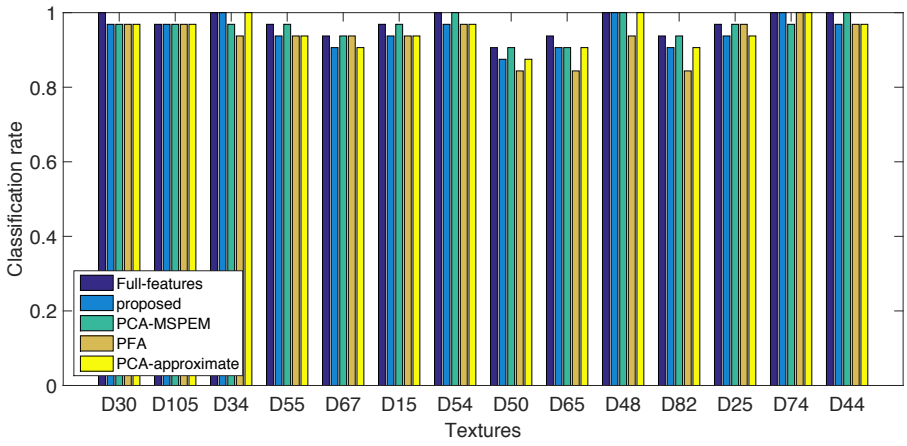
Group	Textures in the group
1	D30, D105, D34, D55, D67, D15, D54, D50, D65, D48, D82, D25, D74, D44
2	D59, D49, D52, D61, D24, D38, D109, D43, D4, D39, D108, D5, D26, D58
3	D89, D101, D93, D2, D60, D80, D20, D18, D106, D95, D35, D75, D66, D86
4	D8, D23, D97, D69, D22, D41, D72, D51, D98, D68, D62, D14, D7, D92
5	D12, D90, D56, D11, D13, D112, D9, D79, D94, D42, D111, D64, D84, D3
6	D107, D40, D6, D29, D99, D36, D110, D78, D16, D27, D37, D83, D103, D87
7	D10, D96, D1, D46, D33, D57, D28, D71, D45, D85, D88, D102, D31, D21
8	D73, D53, D63, D77, D32, D76, D47, D81, D104, D19, D100, D91, D17, D70

Table 3.1: The 8 groups of the Brodatz textures where there are 14 textures randomly assigned in each group.

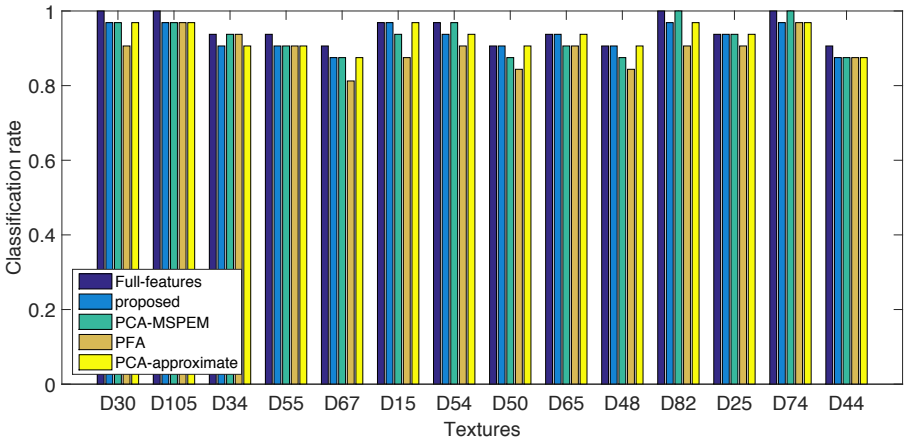
computationally feasible method because it needs no computation other than checking the highest absolute value of the coefficients. But the number of the selected features is the same as that of the principal components, which loses some discriminant features, therefore the performance of classifying textures with these features varies a lot over the whole database, e.g., the classifications of D30, D59, D18, D8, D12, D107, D28, D76 performs worse than those by using all the Tamura's features. The PFA method achieves good performance in selecting the features that are largely spread in the lower dimensional space and good representation of the original data. But it requires the parameter K to be chosen first so it is not adaptive in selecting the features. And the clustering method is high in computational cost when the number of coefficients is large. The proposed method applies the cosine distance as the simpler dissimilarity measurement and finds the feature vectors that are most dissimilar to each other. Therefore, the discriminant features are computed with less computational complexity than other methods and able to represent the most representative features of the original ones, leading to a very close classification performance to the full-feature classification and much shorter running time in both feature selection and classification.

Method	Group 1		Group 2		Group 3		Group 4	
	Features	time(s)	Features	time(s)	Features	time(s)	Features	time(s)
PCA-M	F1, F2, F3, F4, F5	2.74	F1, F2, F3, F4, F5, F6	3.00	F1, F4, F5, F6	3.04	F1, F2, F3, F4	2.84
PCA-A	F1, F2, F4	0.12	F1, F2, F3	0.10	F1, F2, F4	0.14	F2, F3, F4	0.14
PFA	F1, F3, F4	0.10	F1, F2, F5	0.10	F2, F3, F6	0.12	F2, F4, F6	0.12
proposed	F1, F2 F4	0.18	F1, F2, F4	0.16	F1, F4	0.21	F2, F4	0.20
Method	Group 5		Group 6		Group 7		Group 8	
	Features	time(s)	Features	time(s)	Features	time(s)	Features	time(s)
PCA-M	F1, F3, F4, F5, F6	2.58	F1, F2, F4, F5, F6	2.60	F1, F2, F3, F4, F6	2.60	F2, F3, F4, F5, F6	2.70
PCA-A	F1, F2, F3	0.14	F1, F2, F3, F4	0.14	F1, F2, F4	0.14	F2, F3, F4	0.12
PFA	F1, F2	0.10	F2, F4, F5	0.10	F1, F2, F3, F4	0.10	F2, F3, F4	0.08
proposed	F1,F2	0.21	F2,F4	0.20	F1, F2, F4	0.23	F2,F4	0.21

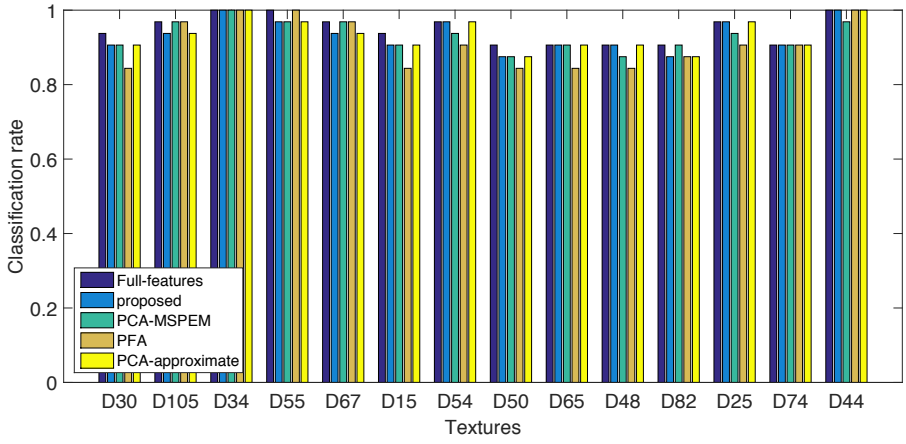
Table 3.2: The essential Tamura's features selected to describe the 8 groups of textures by different feature selection methods. The symbols $F1$, $F2$, $F3$, $F4$, $F5$, $F6$ denote the six Tamura's features: coarseness, contrast, directionality, line-likeness, regularity and roughness respectively.



SVM

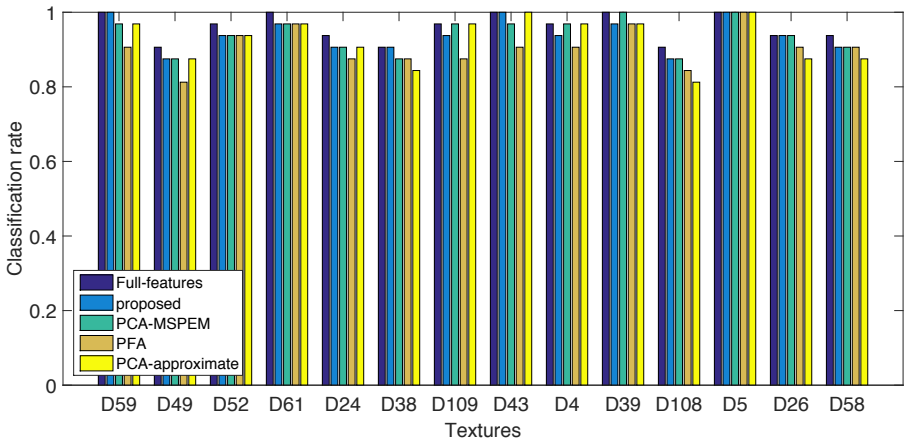


Naive Bayes

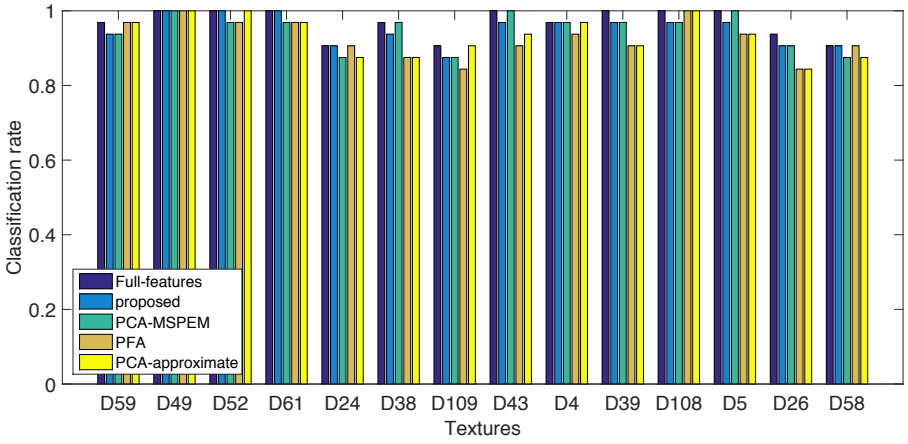


RF

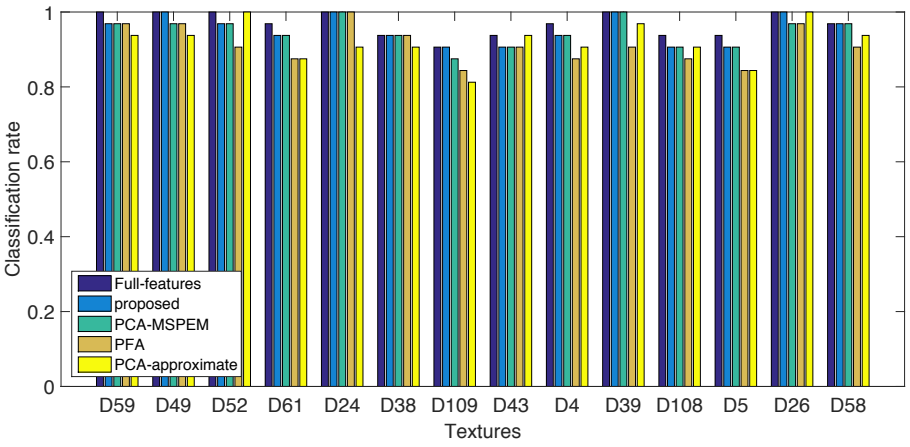
Figure 3.6: The classification rates of the Brodatz textures D30, D105, D34, D55, D67, D15, D54, D50, D65, D48, D82, D25, D74, D44 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM

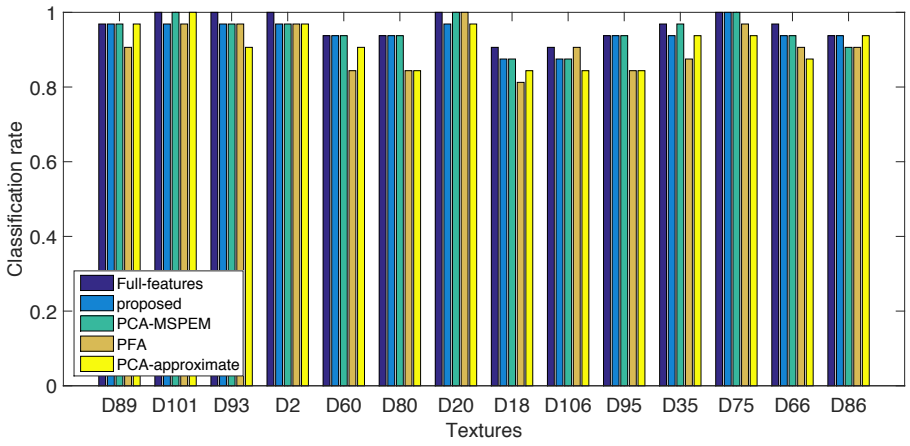


Naive Bayes

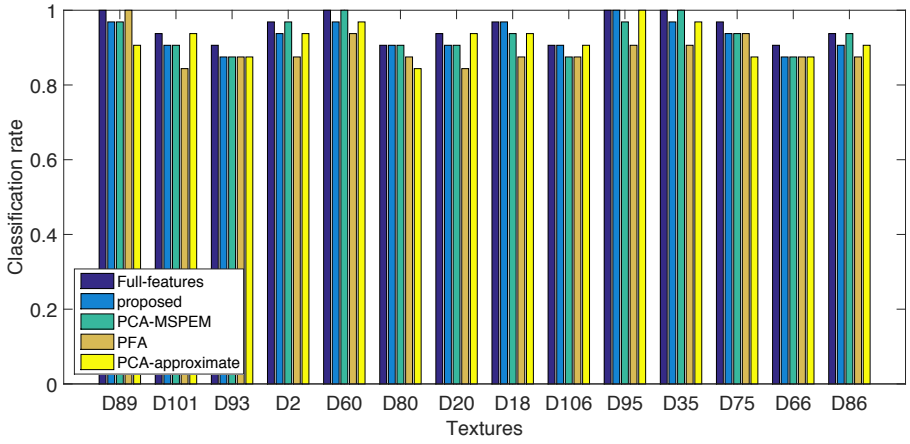


RF

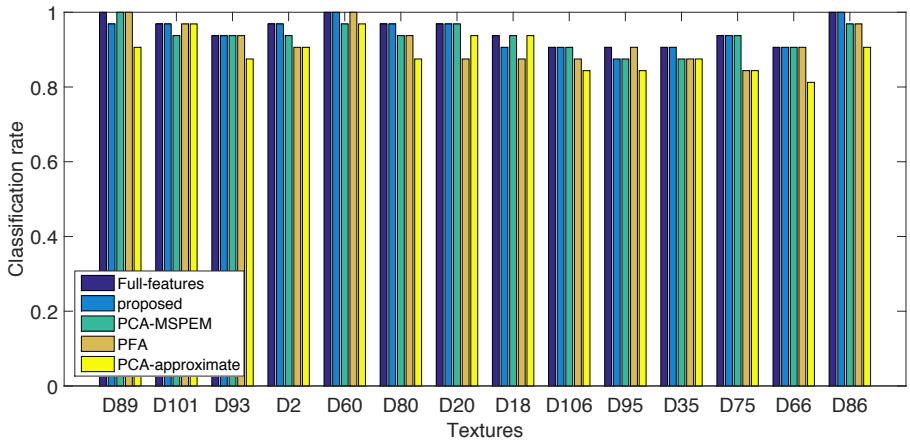
Figure 3.7: The classification rates of the Brodatz textures D59, D49, D52, D61, D24, D38, D109, D43, D4, D39, D108, D5, D26, D58 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM

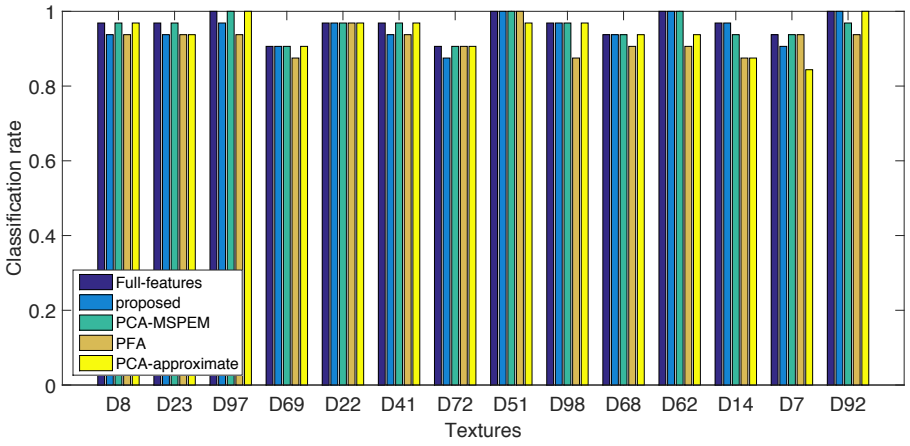


Naive Bayes

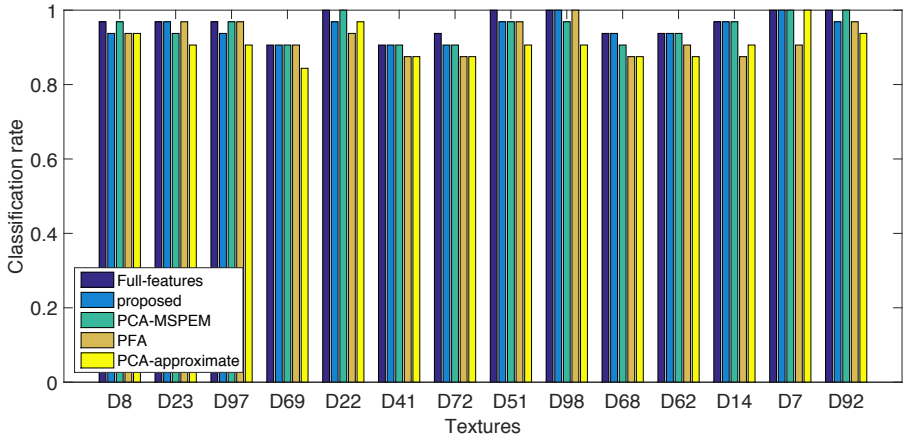


RF

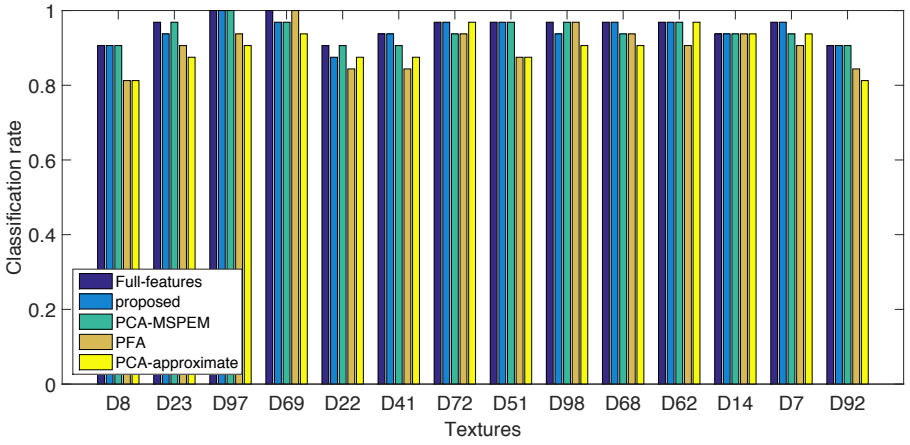
Figure 3.8: The classification rates of the Brodatz textures D89, D101, D93, D2, D60, D80, D20, D18, D106, D95, D35, D75, D66, D86 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM

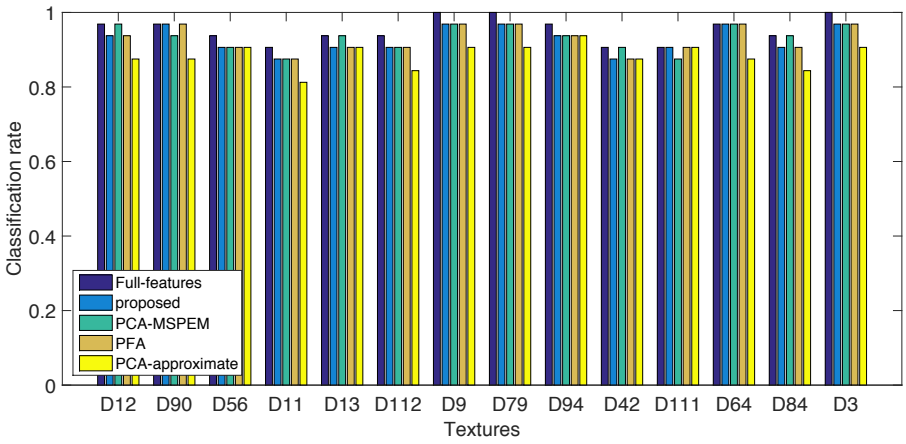


Naive Bayes

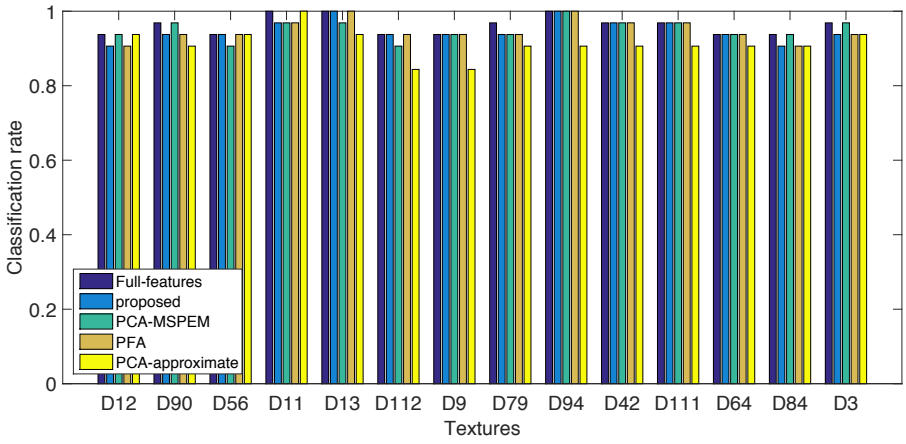


RF

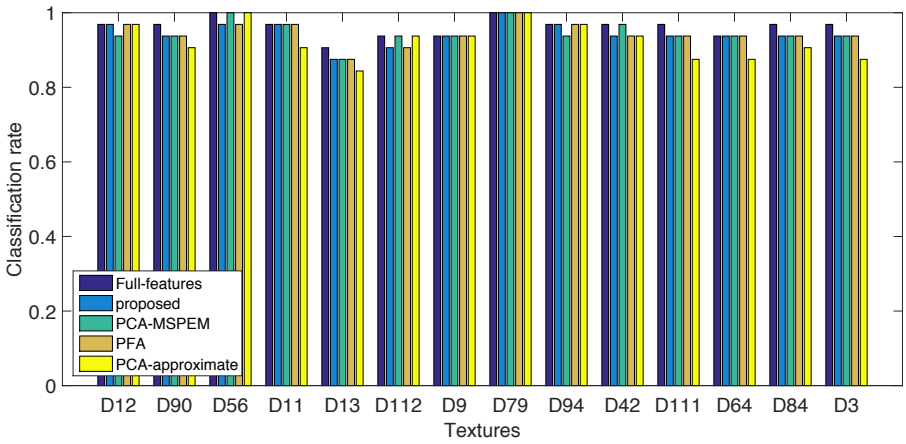
Figure 3.9: The classification rates of the Brodatz textures D8, D23, D97, D69, D22, D41, D72, D51, D98, D68, D62, D14, D7, D92 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM

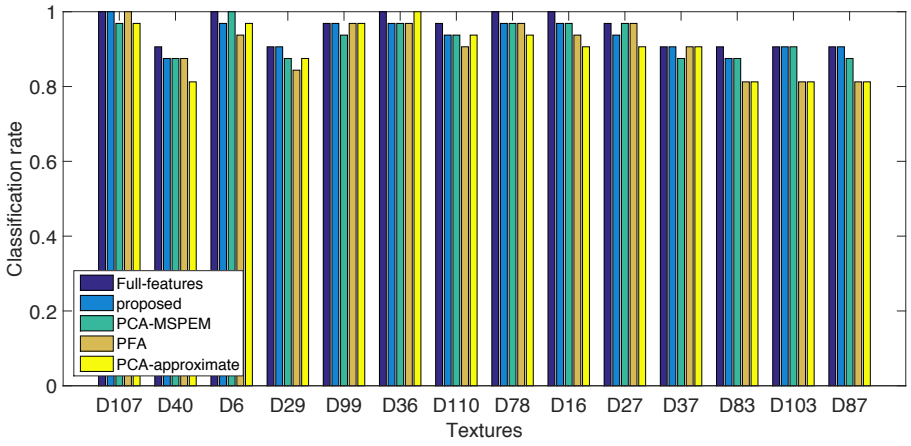


Naive Bayes

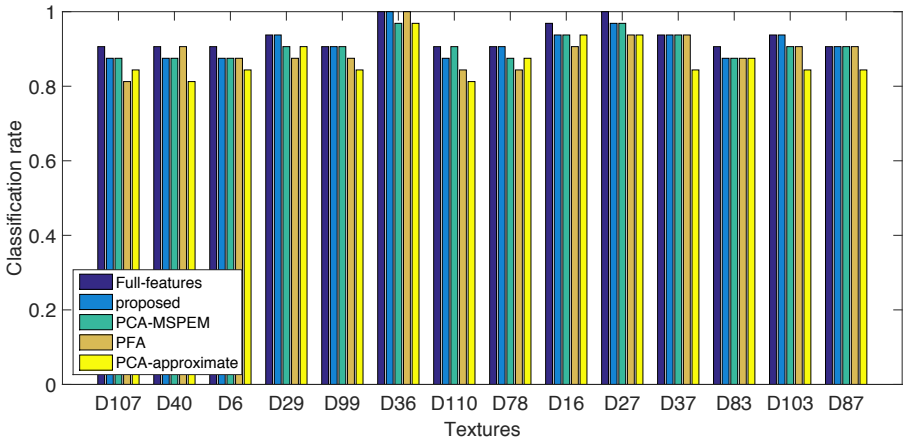


RF

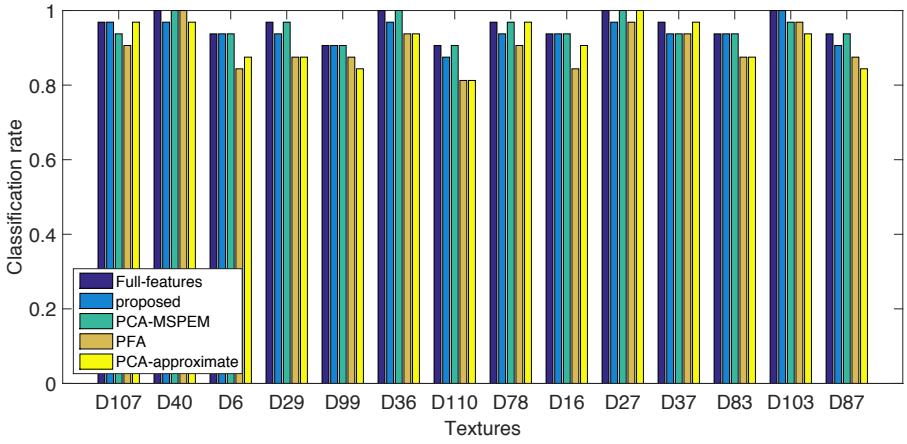
Figure 3.10: The classification rates of the Brodatz textures D12, D90, D56, D11, D13, D112, D9, D79, D94, D42, D111, D64, D84, D3 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM

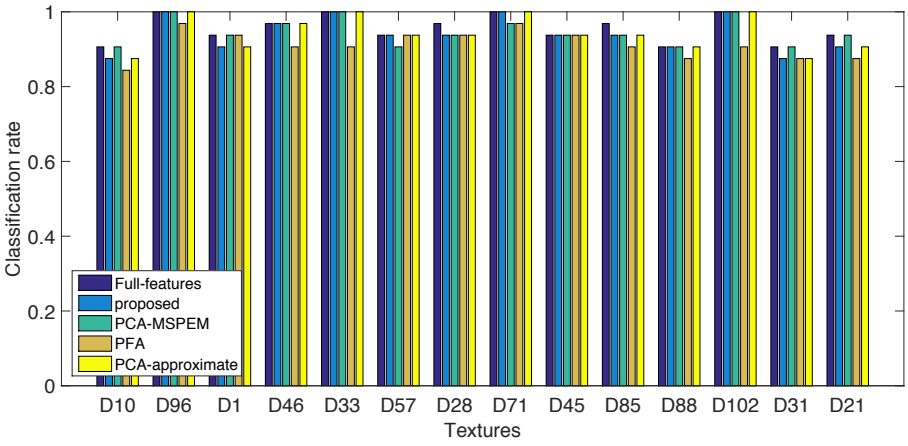


Naive Bayes

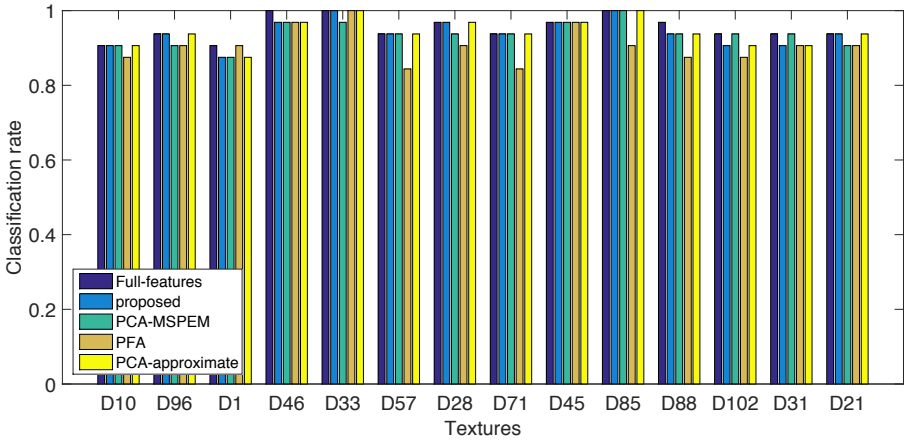


RF

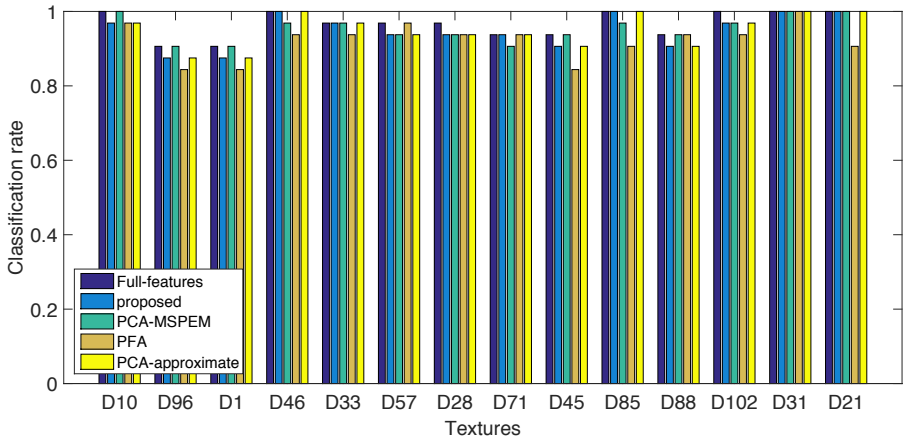
Figure 3.11: The classification rates of the Brodatz textures D107, D40, D6, D29, D99, D36, D110, D78, D16, D27, D37, D83, D103, D87 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM

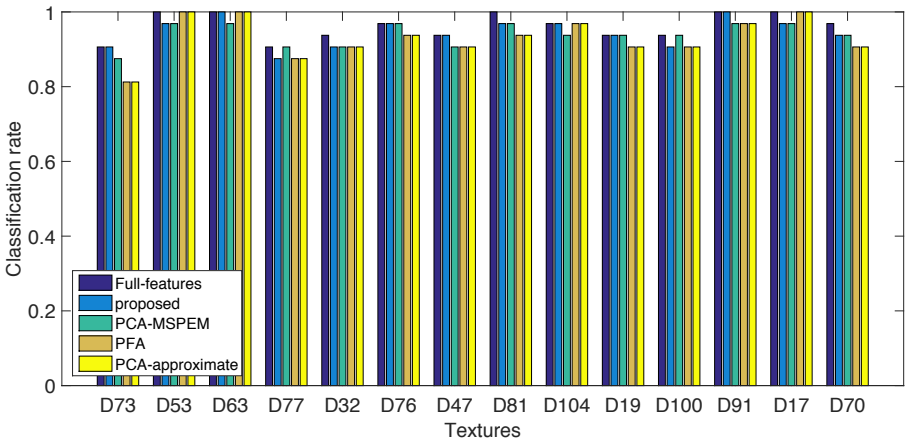


Naive Bayes

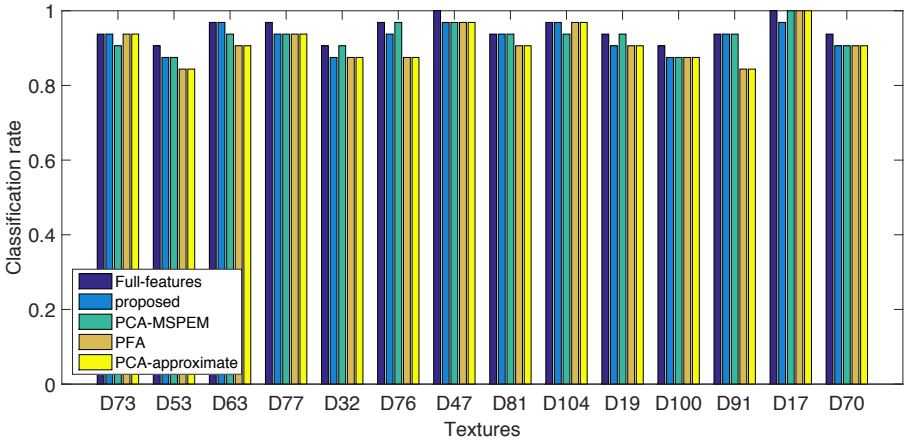


RF

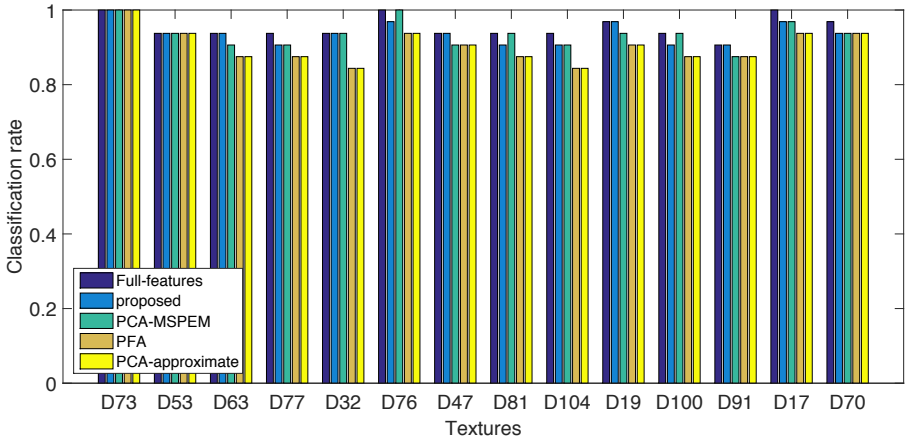
Figure 3.12: The classification rates of the Brodatz textures D10, D96, D1, D46, D33, D57, D28, D71, D45, D85, D88, D102, D31, D21 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.



SVM



Naive Bayes



RF

Figure 3.13: The classification rates of the Brodatz textures D73, D53, D63, D77, D32, D76, D47, D81, D104, D19, D100, D91, D17, D70 by multi-SVM, naive Bayes and random forest classifiers (RF), using the texture features selected by PCA-MSPEM, PCA-approximate, PFA and the propose method.

Method	Group 1			Group 2			Group 3		
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)
original	0.9554	0	131.63	0.9554	0	131.06	0.9598	0	129.89
PCA-M	0.9286	2.74	108.65	0.9286	3.00	110.14	0.9598	3.04	107.88
PCA-A	0.9487	0.12	41.10	0.8951	0.10	41.48	0.9174	0.14	41.10
PFA	0.9174	0.10	41.23	0.8996	0.10	42.54	0.9085	0.12	41.43
proposed	0.9487	0.18	41.10	0.9420	0.16	41.10	0.9554	0.21	40.84
Method	Group 4			Group 5			Group 6		
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)
original	0.9353	0	127.27	0.9442	0	132.51	0.9732	0	133.38
PCA-M	0.9241	2.84	109.81	0.9263	2.58	106.58	0.9554	2.60	109.96
PCA-A	0.8862	0.14	41.01	0.9018	0.14	41.48	0.9263	0.14	109.81
PFA	0.9040	0.12	41.90	0.9420	0.10	41.23	0.9308	0.10	43.05
proposed	0.9241	0.20	40.48	0.9420	0.21	41.23	0.9464	0.20	40.48
Method	Group 7			Group 8					
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)			
original	0.9643	0	135.11	0.9643	0	131.49			
PCA-M	0.9464	2.60	109.96	0.9397	2.70	108.29			
PCA-A	0.9353	0.14	41.10	0.9152	0.12	41.01			
PFA	0.9263	0.10	109.81	0.9152	0.08	41.01			
proposed	0.9353	0.20	41.10	0.9241	0.21	40.48			

Table 3.3: Accuracy and running time of classifying 8 groups of textures by SVM using different subsets of the features. $T1$ and $T2$ are the running time of the feature selection step and classification step respectively. Using the features selected by the proposed method provides the classification accuracy close to that using all the features, with a relatively short running time of the classification that includes the selection of the features ($T1$) and the implementation of the classifier ($T2$).

Method	Group 1			Group 2			Group 3		
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)
original	0.9509	0	131.46	0.9509	0	131.25	0.9509	0	132.33
PCA-M	0.9509	2.74	108.78	0.9509	3.00	109.92	0.9487	3.04	105.07
PCA-A	0.9487	0.12	39.41	0.9241	0.10	39.86	0.9174	0.14	39.41
PFA	0.9152	0.10	39.72	0.9263	0.10	40.11	0.9241	0.12	40.02
proposed	0.9487	0.18	39.41	0.9464	0.16	39.41	0.9375	0.21	38.92
Method	Group 4			Group 5			Group 6		
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)
original	0.9554	0	134.28	0.9621	0	128.92	0.9487	0	131.2
PCA-M	0.9554	2.84	106.58	0.9598	2.58	108.59	0.9420	2.60	107.44
PCA-A	0.9308	0.14	38.43	0.9263	0.14	39.86	0.9286	0.14	106.58
PFA	0.9241	0.12	38.60	0.9598	0.10	38.06	0.9152	0.10	39.19
proposed	0.9554	0.20	38.29	0.9598	0.21	38.06	0.9308	0.20	38.29
Method	Group 7			Group 8					
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)			
original	0.9598	0	132.43	0.9710	0	132.58			
PCA-M	0.9554	2.60	107.12	0.9710	2.70	106.72			
PCA-A	0.9420	0.14	39.41	0.9241	0.12	38.43			
PFA	0.9040	0.10	106.58	0.9241	0.08	38.43			
proposed	0.9420	0.23	39.41	0.9621	0.21	38.29			

Table 3.4: Accuracy and running time of classifying 8 groups of textures by Naive Bayes using different subsets of the features. $T1$ and $T2$ are the running time of the feature selection step and classification step respectively. Using the features selected by the proposed method provides the classification accuracy close to that using all the features, with a relatively short running time of the classification that includes the selection of the features ($T1$) and the implementation of the classifier ($T2$).

Method	Group 1			Group 2			Group 3		
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)
original	0.9598	0	130.99	0.9576	0	131.17	0.9598	0	130.29
PCA-M	0.9487	2.74	106.84	0.9576	3.00	106.70	0.9576	3.04	107.43
PCA-A	0.9487	0.12	38.35	0.9241	0.10	40.01	0.9375	0.14	38.35
PFA	0.9420	0.10	38.39	0.9420	0.10	39.77	0.9397	0.12	39.21
proposed	0.9487	0.18	38.35	0.9509	0.16	38.35	0.9598	0.21	38.11
Method	Group 4			Group 5			Group 6		
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)
original	0.9397	0	130.47	0.9509	0	131.44	0.9576	0	131.97
PCA-M	0.9397	2.84	106.59	0.9509	2.58	107.27	0.9509	2.60	108.96
PCA-A	0.9241	0.14	39.75	0.9286	0.14	40.01	0.9308	0.14	106.59
PFA	0.9353	0.12	40.06	0.9509	0.10	38.95	0.9308	0.10	38.74
proposed	0.9397	0.20	38.18	0.9509	0.21	38.95	0.9487	0.20	38.18
Method	Group 7			Group 8					
	AR	$T1$(s)	$T2$(s)	AR	$T1$(s)	$T2$(s)			
original	0.9487	0	131.88	0.9509	0	131.79			
PCA-M	0.9420	2.60	107.86	0.9487	2.70	109.93			
PCA-A	0.9420	0.14	38.35	0.9353	0.12	39.75			
PFA	0.9297	0.10	106.59	0.9353	0.08	39.75			
proposed	0.9420	0.23	38.35	0.9487	0.21	38.18			

Table 3.5: Accuracy and running time of classifying 8 groups of textures by RF using different subsets of the features. $T1$ and $T2$ are the running time of the feature selection step and classification step respectively. Using the features selected by the proposed method provides the classification accuracy close to that using all the features, with a relatively short running time of the classification that includes the selection of the features ($T1$) and the implementation of the classifier ($T2$).

3.5 Conclusion

In this chapter, we proposed a local Tamura's texture description and a PCA-based feature selection scheme. Texture images can be locally described by the high-level characteristics that fit the human visual perception, including coarseness, contrast, directionality, line-likeness, regularity and roughness. Then a PCA-based feature selection scheme was applied on a given database to find the discriminant features that can differentiate the textures with low redundancy. Experiments were implemented to compare the performance of using the principal features selected by the proposed method and other methods in classifying the Brodatz textures by different classifiers. The experimental results demonstrate that the local Tamura's texture features can describe different textures in the database accurately and the proposed feature selection method can effectively find the features that are essential in differentiating the textures. As a result, the textural differences in the images are able to be measured with respect to these key texture characteristics and the image can be considered as the sum of components representing these key characteristics.

Chapter 4

Texture characteristic based morphological component analysis using wavelet-based dictionaries

As discussed in the last chapter, a given image can be described by several human visual perceptual characteristics, that is, different regions in the image are different in these certain characteristics. As a result, magnifying the textural differences in the image equates to enhancement of the image components representing these characteristics. Therefore, we will discuss how to decompose a given image into components representing texture characteristics in the following parts.

4.1 Introduction

From the literatures reviewed in Chapter 2.2, morphological component analysis (MCA) is suitable for decomposing images into components representing different features. However, the MCA requires improvement both in concept and in dictionary selection to represent different texture characteristics. We will first introduce some background knowledge relevant to our image decomposition method in this chapter.

4.1.1 Morphological component analysis

Morphological component analysis (MCA) has proven successful at separating features contained in an image when these features present different morphological aspects [189]. In a series of recent papers [191, 188, 69, 71], the MCA concept has been developed and shown that MCA can be used for separating the texture and the piecewise smooth component from a given image [188], for in-painting applications [69] or, more generally, for separating several components containing different morphologies. MCA has also been extended to the multichannel case in [21, 20]. The central idea of MCA is to make use of the morphological diversity of the different types of signals contained in the data, and to represent each morphology with a dictionary for which a fast transform is available. Because of recent developments in harmonic analysis, many new multi-scale transforms are now possible to be used as dictionaries [28, 187, 61], increasing the potential versatility of MCA. However, most of the recent of MCA-based image decomposition methods, no matter what dictionaries they make use of, are limited to decomposition of images into "cartoon" and "texture" components, which is still far from the original assumption of the MCA concept that it can decompose

the images into arbitrary morphologically different components. This shortcoming leads us to an overview of recent dictionary selections of MCA, which determine the components separated from the image by the MCA methods.

4.1.2 Dictionaries

A *dictionary* is a transformation representing some certain content type of the image. In recent MCA-based methods, the selection of dictionaries focused on the transformations that can discriminate piecewise smooth content and texture content, according to the purposes of these methods in image inpainting, texture separation, or astronomical image processing, respectively.

For piecewise smooth content, bi-orthogonal wavelet transforms (OWT) have proved to be well suited to sparse representation in the early research [64] because they can represent the directional elements independently [7, 179, 171]. However, the OWT lacks shift invariance since it down-samples the original signals, which causes difficulties by losing information of the structures of the images. Therefore, the undecimated wavelet transform (UWT) was proposed to obtain the desired shift invariance property. UWT improved the OWT by skipping the decimation, implying that it is an over-complete transform represented as a matrix with more columns than rows. However, both the OWT and the UWT present only a fixed number of directional elements without highly anisotropic elements [30], so they were deemed sub-optimal for detecting highly anisotropic features of an image. Isotropic Trous Algorithm was another version of the wavelet transform that decomposes an $N \times N$ image I as a superposition of the form $I(x,y) = c_J(x,y) + \sum_{j=1}^{J} w_j(x,y)$, where c_J is a coarse or smooth version of the original image I and w_j represent the details of I at scale 2^{-j} [193]. Therefore, the algorithm outputs $J+1$ sub-bands of size $N \times N$, which meets the requirement for shift invariance. Moreover, this transform is very well adapted to the detection of isotropic features, which explains the reason of its success for astronomical image processing where data contains mostly (quasi-)isotropic objects, such as stars or galaxies [190]. The ridgelet transform is the application of a 1-D wavelet transform to the angular slices of the Radon transform [30]. It has shown to be effective in representing global lines in an image. For detecting the line segments, a partitioning must be implemented [29], and then a ridgelet transform is applied in each block. The curvelet transform, proposed in [187, 63, 31], enabled the directional analysis of an image at different scales. The curvelet transform may be considered as a combination of the wavelet transform and ridgelet transform: it first decomposes the image into a set of wavelet sub-bands, and then analyzes each band with a local ridgelet transform with different block sizes at different scale levels. Different levels of the multi-scale ridgelet pyramid are used to represent different sub-bands of a filter bank output. To maintain the fundamental property of the curvelet transform, the side-length of the local windows is doubled at every other dyadic sub-band. In another word, the elements of length about $2^{-j/2}$ serve for the analysis and synthesis of the j-th sub-band $\left[2^j, 2^{j+1}\right]$. Therefore, the curvelet transform is also redundant with a redundancy factor of $16J+1$ whenever J scales are employed, and the redundancy provides the shift invariance of curvelet transform. So far, the curvelet transform performs best for the detection of anisotropic

structures, smooth curves, and edges of different lengths, that is, the piecewise smooth content.

For texture content representation, the discrete cosine transform (DCT) [164] is a quite popular choice recently. The DCT is a variant of the discrete Fourier transform (DFT) [92], replacing the complex analysis with real numbers by a symmetric signal extension. The DCT is an orthogonal transform, known to be well suited for first order Markov stationary signals. Its coefficients essentially represent frequency content, similar to the those obtained by Fourier analysis. When dealing with non-stationary sources, DCT is typically applied in blocks. Such is indeed the case in the JPEG image compression algorithm. Choice of overlapping blocks is preferred for analyzing signals while preventing artifacts. A block overlap of 50% results in an over-complete transform with a redundancy factor of 4. The DCT has proven to be appropriate for a sparse representation of either smooth or periodic behaviours. Another popular choice for texture representation is the Gabor transform. Similar to the DCT, the Gabor transform is essentially a localized DFT, where the localization is obtained by windowing portions of the signal in an overlapping fashion [161]. The amount of redundancy is controllable.

Given the goals of the standard MCA methods, the dictionaries for smooth and textured content are sufficient for those goals. However, these conventional dictionaries cannot represent different texture characteristics, which is necessary for finding the elements resulting in textural differences and enhancing them as we describe in later sections. Therefore, it is necessary to find dictionaries that can both represent the components sparsely and reflect different texture characteristics.

4.2 Standard morphological component analysis (MCA)

4.2.1 Assumptions of the MCA model

It was assumed in [189] that the data s can be considered as the sum of K different signals, i.e., $s = \sum_{k=1}^{K} s_k$, where each s_k represents a different type of signal to be decomposed from s. Then the assumptions for the standard morphological component analysis is:

1. For every possible signal s_k, there exists a dictionary (which could be over-complete), $\Phi_k \in M^{N \times L_k}$ (where typically $L_k \gg N$) such that solving:

$$\alpha_k^{opt} = \arg\min_{\alpha} \|\alpha\|_0 \text{ subject to: } s_k = \Phi_k \alpha, \tag{4.1}$$

   leads to a very sparse solution (i.e. $\left\|\alpha_k^{opt}\right\|_0$ is very small). The definition in the above equation is essentially the over-complete transform of s_k, yielding a representation α_k.

2. For every possible signal s_l, solving for $k \neq l$:

$$\alpha_l^{opt} = \arg\min_{\alpha} \|\alpha\|_0 \text{ subject to:} s_l = \Phi_k \alpha, \tag{4.2}$$

   leads to a very non-sparse solution. This requirement suggests that the dictionary Φ_k is able to distinguish between the different types of signals to be separated.

Therefore, the dictionaries Φ_k play the role of discriminants between the different content types. And it is required that dictionaries Φ_k should have a fast transformation T_k ($\alpha_k = T_k s_k$) and reconstruction R_k ($s_k = R_k \alpha_k$) for computational convenience.

For an arbitrary signal s containing K types of signals as a linear combination, it was proposed to seek the sparse representation over the augmented dictionary containing all Φ_k by the following optimization function:

$$\begin{aligned} \left\{\alpha_1^{opt}, \ldots, \alpha_K^{opt}\right\} = \arg \min_{\alpha_1,\ldots,\alpha_K} \sum_{k=1}^{K} \|\alpha_k\|_0 \\ \text{subject to: } s = \sum_{k=1}^{K} \Phi_k \alpha_k \end{aligned} . \tag{4.3}$$

This optimization task is likely to lead to a successful separation of the signal content, based on the assumptions made earlier about Φ_k being very efficient in representing specific phenomena and being highly ineffective in representing other signal types.

4.2.2 Standard "Cartoon + Texture" decomposition using MCA

In [69], the image was assumed to contain two types of content: (1) piece-wise smooth image content containing the non-texture or structure parts of the image, which was called "cartoon" component, and (2) texture or oscillation image content, which was called "texture" component. By considering the input image that contained N total pixels as a $1D$ vector of length N containing two types of signals, the image decomposition can be modelled as a case of signal decomposition shown in Eq. 4.3 where the number of signal types $K = 2$. To model the component X_t containing only texture, it was assumed that there was a transform T_t allowed sparse decomposition:

$$\alpha_t = \mathbf{T}_t X_t, \quad \alpha_t \text{ is sparse.} \tag{4.4}$$

The sparsity was quantified by the small value of the l_0 norm $\|\alpha_t\|_0 = \#\{i : \alpha_t(i) \neq 0\}$. Sparsity of image decomposition measured in the l_0 norm indicates that the texture image X_t is a linear combination of relatively few columns from T_t.

For the transformation T_t, there were two more assumptions proposed in [69]. One is the localization: the representation of the image by transform T_t should be still sparse even if the textures appear only in parts of the image, requiring that the dictionary exploited a multi-scale and local analysis of the image. The other assumption was incoherence: T_t cannot represent the cartoon parts of the image sparsely, implying that the cartoon image cannot be a linear combination of few columns from T_t. Therefore, T_t should be sensitive to texture parts of the image while insensitive to the cartoon parts, differentiating between the two types of image content.

Conversely, the existence of a dictionary T_n was assumed such that the cartoon component X_n can be sparsely represented by the above definition. Similar to the assumptions about T_t, the analysis employed by this dictionary was multi-scale and local in nature, enabling representation of localized pieces of the target content. Moreover, incoherence should be guaranteed such that texture components are represented very

non-sparsely by T_n. Therefore, for an arbitrary image X containing both texture and piece-wise smooth content, the sparse representations and the transformation dictionaries T_t, T_n were sought at the same time by solving the optimization function:

$$\left\{X_t^{opt}, X_n^{opt}\right\} = \arg \min_{\{X_t, X_n\}} \|T_t X_t\|_0 + \|T_n X_n\|_0 \quad \text{subject to: } X = X_t + X_n. \tag{4.5}$$

Based on the assumptions about T_t and T_n being able to sparsely represent one type of content while being highly ineffective in sparsely representing the other, the above optimization task leads to a successful separation of the image with $\alpha_t = T_t X_t$ corresponding to the texture and $\alpha_n = T_n X_n$ corresponding to the cartoon. However, the computational complexity of Eq. 4.5 grows exponentially with the number of columns in the overall dictionary because the problem with l_0 norm is non-convex and intractable. Then the l_0 norm was replaced by l_1 norm as suggested by the basis pursuit (BP) method [41], resulting in the more tractable optimization problem:

$$\left\{X_t^{opt}, X_n^{opt}\right\} = \arg \min_{\{X_t, X_n\}} \|T_t X_t\|_1 + \|T_n X_n\|_1 \quad \text{subject to: } X = X_t + X_n. \tag{4.6}$$

The BP approach was extended for the case that the image was noisy and it cannot be completely decomposed into texture and cartoon parts:

$$\left\{X_t^{opt}, X_n^{opt}\right\} = \arg \min_{\{X_t, X_n\}} \|T_t X_t\|_1 + \|T_n X_n\|_1 \quad \text{subject to: } \|X - X_t - X_n\|_2 \leq \epsilon, \tag{4.7}$$

where ϵ is the noise level in the image X. This solution leads to an approximation of the image decomposition leaving some error to be absorbed by content that was not represented well by either of the dictionaries. An alternative method was proposed to replace the constrained condition in Eq. 4.7 by an unconstrained penalized optimization, providing more accurate sparse representation:

$$\left\{X_t^{opt}, X_n^{opt}\right\} = \arg \min_{\{X_t, X_n\}} \|T_t X_t\|_1 + \|T_n X_n\|_1 + \lambda \|X - X_t - X_n\|_2^2, \tag{4.8}$$

where λ was the penalty coefficient for the noise in the image.

4.2.3 Algorithm to solve the standard morphological component analysis

To solve the optimization problem Eq. 4.8, the following steps were implemented in [189]:

1. two dictionaries were selected first for the cartoon and texture parts of the image. The dictionaries employed transformations that represent either texture or piecewise smooth (cartoon) behaviours of the image. More details about candidate dictionaries were described in [188];

2. with the selected dictionaries T_t and T_n, the Algorithm 1 was proposed to minimize Eq. 4.8 and seek two images X_t and X_n as the texture and cartoon components of the image X.

1. Initialize L_{max}, the number of iterations,
   and threshold $\delta = \lambda \cdot L_{max}$, $X =$ input image,
   $X_t = X_n = 0$. T_t, T_n, R_t, R_n are the forward and inverse
   transforms for the dictionaries, respectively.
2. Perform L_{max} times:
   Update of X_t assuming X_n is fixed:
      - Calculate the residual $r = X - X_t - X_n$.
      - Calculate the transform T_t of $X_t + r$
        and obtain $\alpha_t = T_t(X_t + r)$.
      - Soft threshold the coefficient α_t with
        the δ threshold and obtain $\widehat{\alpha_t}$.
      - Reconstruct X_t by $X_t = R_t\widehat{\alpha_t}$.
   Update of X_n assuming X_t is fixed:
      - Calculate the residual $r = X - X_n - X_t$.
      - Calculate the transform T_n of $X_n + r$
        and obtain $\alpha_n = T_n(X_n + r)$.
      - Soft threshold the coefficient α_n with
        the δ threshold and obtain $\widehat{\alpha_n}$.
      - Reconstruct X_n by $X_n = R_n\widehat{\alpha_n}$.
3. Update the threshold by $\delta = \delta - \lambda$.
4. If $\delta > \lambda$, return to Step 2. Else, finish.

Algorithm 4.1: The algorithm for minimizing Eq. 4.8 [189].

4.3 Wavelet-based texture characteristic morphological component analysis

4.3.1 Model of texture characteristic MCA

Similar to the standard MCA method, the model of texture characteristic morphological component analysis (TC-MCA) is established on the following assumptions:

1. According to i-th feature in the selected texture descriptor, a given image x is the combination of 2 parts, $x = x_{s,i} + x_{w,i}$, where $x_{s,i}$ and $x_{w,i}$ represent the strong and weak aspect of the i-th texture characteristic.

2. For $x_{s,i}$, there exists a dictionary $\Phi_{s,i}$ exploiting the transformation $T_{s,i}$ so that the solution:

$$\alpha_{s,i}^{opt} = \arg\min_{\alpha} \|\alpha\|_1 \text{ subject to: } \alpha_{s,i} = T_{s,i}x_{s,i}, \tag{4.9}$$

   is very sparse, that is , $\|\alpha_{s,i}\|_1$ is very small.

3. For $x_{w,i}$, solving:

$$\alpha_{w,i}^{opt} = \arg\min_{\alpha} \|\alpha\|_1 \text{ subject to: } \alpha_{w,i} = T_{s,i}x_{w,i} \tag{4.10}$$

   leads to a very non-sparse solution or zero-solution, suggesting that the dictionary $\Phi_{s,i}$ is distinguishing between strong and weak aspects of the i-th texture characteristic.

Therefore, for a given texture image I consisting of components representing different texture characteristics, TC-MCA decomposes the image to pairwise components, wherein each pair is composed of components showing the strong and weak aspects of a certain texture characteristic of the image respectively. However,

1. Initialize the number of iterations L_{max}, the parameters
   $\mu_{s,i}$ and $\mu_{w,i}$ of the dictionaries
   $T_{s,i}$ and $T_{w,i}$, $\delta = \lambda \cdot L_{max}$, threshold for stopping
   decomposition, and ϕ is a threshold for updating
   parameters of dictionaries. $s_{s,i} = s_{w,i} = 0$.
2. Perform L_{max} times:
   Update of $s_{s,i}$ assuming $s_{w,i}$ is fixed:
      - Calculate the residual $r = I - s_{s,i} - s_{w,i}$.
      - Calculate the transformation $T_{s,i}$ of $s_{s,i} + r$ and
        obtain $s'_{s,i} = T_{s,i}(s_{s,i} + r)$.
      - Calculate $d = \left\| s'_{s,i} - s_{s,i} \right\|_1$.
      - If $d > \phi$, update $T_{s,i}$ by updating
        $\mu_{s,i}$ with $\mu_{s,i} - \frac{\mu_{s,i}}{L_{max}}$.
        Else, $\mu_{s,i}$ keep the same values.
      - Update $s_{s,i}$ with $s'_{s,i}$.
   Update of $s_{w,i}$ assuming $s_{s,i}$ is fixed:
      - Calculate the residual $r = I - s_{w,i} - s_{s,i}$.
      - Calculate the transformation $T_{w,i}$ of $s_{w,i} + r$ and
        obtain $s'_{w,i} = T_{w,i}(s_{w,i} + r)$.
      - Calculate $d = \left\| s'_{w,i} - s_{w,i} \right\|_1$.
      - If $d > \phi$, update $T_{w,i}$ by updating
        $\mu_{w,i}$ with $\mu_{w,i} - \frac{\mu_{w,i}}{L_{max}}$.
        Else, $\mu_{w,i}$ keep the same values.
      - Update $s_{w,i}$ with $s'_{w,i}$.
3. Update the threshold by $\delta = \delta - \lambda$.
4. If $\delta > \lambda$, return to Step 2. Else, finish.

Algorithm 4.2: The algorithm for minimizing Eq. 4.11.

differently from the traditional MCA decomposition introduced in Chapter 4.2, where "cartoon" and "texture" are two strictly different types of image contents, the boundary between "strong characteristic" and "weak characteristic" in TC-MCA is quite fuzzy because it represents the extent to which one type of texture exhibits the certain texture characteristic. For example, "coarse" and "fine" components are decomposed from the image according to "coarseness". The "coarser" the texture is, the more it is preserved in "coarse" component, while the "finer" the texture is, the more it is preserved in "fine" component. Generally, the decomposition problem is equivalent to the optimization task:

$$\begin{aligned}\left\{s_{s,i}^{opt}, s_{w,i}^{opt}, T_{s,i}^{opt}, T_{w,i}^{opt}\right\} = \arg \min_{\{s_{s,i}, s_{w,i}, T_{s,i}, T_{w,i}\}} & \left\|T_{s,i} s_{s,i}\right\|_1 \\ & + \left\|T_{w,i} s_{w,i}\right\|_1 + \left\|I - s_{s,i} - s_{w,i}\right\|_2^2\end{aligned}, \tag{4.11}$$

where $s_{s,i}$ and $s_{w,i}$ are the components strongly and weakly exhibiting the i-th characteristic of the input image I, $i = 1, 2, \ldots, k$, and $T_{s,i}^{opt}$ and $T_{w,i}^{opt}$ are the filters with optimized parameters as dictionaries representing the strong and weak components of i-th characteristic $s_{s,i}$ and $s_{w,i}$ respectively.

Based on the selected dictionaries, an improved Block-Coordinate-Relaxation method is used to iteratively compute components $s_{s,i}$ and $s_{w,i}$ as well as the parameters of dictionaries $T_{s,i}$ and $T_{w,i}$ in Eq. 4.11. In Algorithm 2, $\mu_{s,i}$ and $\mu_{w,i}$ are the parameter sets of the dictionaries $T_{s,i}$ and $T_{w,i}$ respectively. L_{max} is the maximum number of iterations of decomposition. The parameters in $\mu_{s,i}$ and $\mu_{w,i}$ are decreased uniformly over each iteration to 0 until optimal results are obtained. Note that the decomposition for each texture characteristic is independent of the others so that any number of texture characteristics may be used as desired.

4.4 Wavelet-based thresholding used as the dictionaries of TC-MCA

Considering that all the four characteristics used as the basis of decomposition: coarseness, contrast, line-likeness and directionality, are about measuring the statistics of edges and pixel intensities of the image, we propose to apply the discrete wavelet transform [170] as the basis of the dictionaries for all the four parts of decomposition because:

1. the low-frequency sub-band of DWT contains the information of the distribution of pixel intensities of the image and high-frequency sub-bands contain the information of the distribution of edges of the image, which is suitable for reflecting the coarseness (density of edges) and contrast (variance of pixel intensities);

2. the high-frequency sub-bands of DWT preserve the directions and positions of the edges in the image, which is necessary in measuring directionality (mean of edge directions in a local region) and line-likeness (variance of edge directions in a local region);

Therefore, the i-th strong and weak components $s_{s,i}$ and $s_{w,i}$ are separated from the input image I by the wavelet-based dictionaries $T_{s,i}$ and $T_{w,i}$ respectively:

$$s_{s,i} = T_{s,i}(I), \quad s_{w,i} = T_{w,i}(I), \tag{4.12}$$

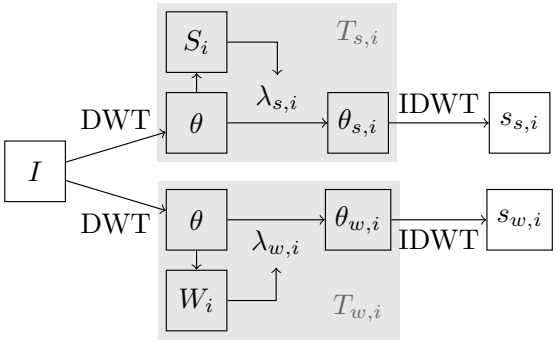


Figure 4.1: Decomposition of the image for the i-th texture characteristic with wavelet-based dictionaries $T_{s,i}$ and $T_{w,i}$. $T_{s,i}$ and $T_{w,i}$ are implemented in three steps: 1) obtain wavelet coefficients θ from input image I, 2) filter the coefficients θ by the shrinkage coefficients $\lambda_{s,i}$ and $\lambda_{w,i}$ (which are dependent on the characteristic maps S_i and W_i) to obtain wavelet coefficients of the strong and weak components, 3) obtain the strong and weak components of the i-th texture characteristic $s_{s,i}$ and $s_{w,i}$ from the inverse wavelet transform (IDWT) of $\theta_{s,i}$ and $\theta_{w,i}$.

Fig. 4.1 shows the implementation of $T_{s,i}$ and $T_{w,i}$. $\theta = \{a, h, v, d\}$ are, respectively, the approximation, horizontal, vertical, and diagonal wavelet coefficients from the discrete wavelet transform $\mathrm{dwt}(\cdot)$ with j scales. The *characteristic maps* S_i and W_i are binary images computed from the components of θ which denote which

areas of the image respectively strongly and weakly exhbit the i-th texture characteristic; these are computed differently for each characteristic (see Chapter 4.4.1 for details). The shrinkage coefficients $\lambda_{s,i}$ and $\lambda_{w,i}$ are defined as the local density of the strong and weak characteristic maps S_i and W_i as follows:

$$\lambda_{s,i}(x,y) = \frac{\sum_{(r,c)\in L} S_i(r,c)}{N}, \quad \lambda_{w,i}(x,y) = \frac{\sum_{(r,c)\in L} W_i(r,c)}{N}, \tag{4.13}$$

where L represents the local neighbourhood centred at (x,y), and N is the number of pixels in the neighbourhood. The shrinkage coefficients $\lambda_{s,i}$ and $\lambda_{w,i}$, $T_{s,i}$ and $T_{w,i}$ separate the wavelet coefficients θ to θ_{s_i} and θ_{w_i} where only wavelet coefficients exhibiting the i-th characteristic strongly and weakly are preserved respectively. The components $s_{s,i}$ and $s_{w,i}$ representing the i-th texture characteristic strongly and weakly are obtained from the inverse wavelet transform idwt$(\cdot)$ of $\theta_{s,i}$ and $\theta_{w,i}$.

Based on the above discussion, the key point to solve the decomposition problem is to find appropriate wavelet thresholding functions for representing each texture characteristic strongly and weakly. The proposed functions are described in the following parts.

4.4.1 Thresholding functions for each texture characteristic

The thresholding functions based on the strong and weak characteristic maps S_i and W_i are shown in Table 4.1. These functions determine the dictionaries $T_{s,i}$ and $T_{w,i}$ for the i-th texture characteristic as shown in Fig. 4.1 and Eq. 4.13. The details of the design of the characteristic maps for each texture characteristic are described below.

	Strong characteristic map	Weak characteristic map
coarseness	$S_1(i,j) = \begin{cases} 1 & \text{if } \lvert h(i,j)\rvert < \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert < \delta_v \ \& \\ & \quad \lvert d(i,j)\rvert < \delta_d \\ 0 & \text{otherwise} \end{cases}$	$W_1(i,j) = \begin{cases} 1 & \text{if } \lvert h(i,j)\rvert > \delta_h \text{ or} \\ & \quad \lvert v(i,j)\rvert > \delta_v \text{ or} \\ & \quad \lvert d(i,j)\rvert > \delta_d \\ 0 & \text{otherwise} \end{cases}$
contrast	$S_2(i,j) = \begin{cases} 1 & \text{if } \Delta x > \delta_x \text{ or} \\ & \quad \Delta y > \delta_y \\ 0 & \text{otherwise} \end{cases}$	$W_2(i,j) = \begin{cases} 1 & \text{if } \Delta x < \delta_x \ \& \\ & \quad \Delta y < \delta_y \\ 0 & \text{otherwise} \end{cases}$
	$\Delta x = \lvert a(x+1,y) - a(x,y)\rvert$ $\Delta y = \lvert a(x,y+1) - a(x,y)\rvert$	$\Delta x = \lvert a(x+1,y) - a(x,y)\rvert$ $\Delta y = \lvert a(x,y+1) - a(x,y)\rvert$
directionality	$S_3(i,j) = \begin{cases} 1 & \text{if } \lvert h(i,j)\rvert > \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert \le \delta_v \\ 0 & \text{otherwise} \end{cases}$	$W_3(i,j) = \begin{cases} 1 & \text{if } \lvert v(i,j)\rvert > \delta_v \ \& \\ & \quad \lvert h(i,j)\rvert \le \delta_h \\ 0 & \text{otherwise} \end{cases}$
line-likeness	$S_4(i,j) = \begin{cases} 1 & \text{if } \lvert h(i,j)\rvert > \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert \le \delta_v \ \& \\ & \quad \lvert d(i,j)\rvert \le \delta_d \\ 1 & \text{if } \lvert h(i,j)\rvert \le \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert > \delta_v \ \& \\ & \quad \lvert d(i,j)\rvert \le \delta_d \\ 1 & \text{if } \lvert h(i,j)\rvert \le \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert \le \delta_v \ \& \\ & \quad \lvert d(i,j)\rvert > \delta_d \\ 0 & \text{otherwise} \end{cases}$	$W_4(i,j) = \begin{cases} 1 & \text{if } \lvert h(i,j)\rvert > \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert > \delta_v \\ 1 & \text{if } \lvert h(i,j)\rvert > \delta_h \ \& \\ & \quad \lvert d(i,j)\rvert > \delta_d \\ 1 & \text{if } \lvert v(i,j)\rvert > \delta_v \ \& \\ & \quad \lvert d(i,j)\rvert > \delta_d \\ 1 & \text{if } \lvert h(i,j)\rvert < \delta_h \ \& \\ & \quad \lvert v(i,j)\rvert < \delta_v \ \& \\ & \quad \lvert d(i,j)\rvert < \delta_d \\ 0 & \text{otherwise} \end{cases}$

Table 4.1: Thresholding functions for representing each texture characteristics for the proposed wavelet-based TC-MCA (WT-TC-MCA). The initial values are selected as: $\delta_h = 0.95\,|h|_{max}$, $\delta_v = 0.95\,|v|_{max}$ and $\delta_d = 0.95\,|d|_{max}$.

Coarseness

As discussed in Chapter 2, coarseness is used to measure the size of the primitives of the texture. Coarse (strong coarseness) texture corresponds to larger primitives in the texture image, while fine (weak coarseness) texture corresponds to smaller primitives. When measured in low-level description, texture is considered as the regular repetition of some elements (called texels). Thus, the edges between those elements in a given neighbourhood can represent different image contents with strong and weak coarseness: coarse areas of the image exhibit fewer, stronger edges surrounding the larger texels, while fine areas of the image have more, weaker edges surrounding the small texels. Therefore, the dictionaries, as well as the corresponding characteristic maps S_1 and W_1, which are used to discriminate coarse and fine contents of the image, are also based on the measurement of the quantity and strength of the edges, which are represented by the values of the wavelet coefficients in high-frequency sub-bands, reflecting the distribution of edges in the image.

The characteristic map for coarse component is defined as follows:

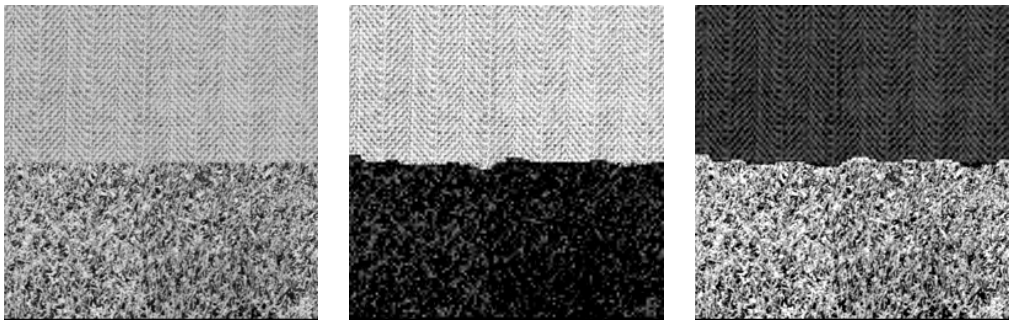
$$S_1(i,j) = \begin{cases} 1 & \text{if } |h(i,j)| < \delta_h \;\&\; \\ & \quad |v(i,j)| < \delta_v \;\&\; \\ & \quad |d(i,j)| < \delta_d \\ 0 & \text{otherwise} \end{cases}, \tag{4.14}$$

where $h(i,j)$, $v(i,j)$, $d(i,j)$ are the wavelet coefficients in high-frequency sub-bands, δ_h, δ_v and δ_d are the corresponding thresholds. In coarse regions, the magnitudes of the coefficients $h(i,j)$, $v(i,j)$, $d(i,j)$ will be lower than the thresholds δ_h, δ_v and δ_d, S_1 has the value of 1 so that the shrinkage coefficient $\lambda_{s,1}$ has the value close to 1, therefore these wavelet coefficients will be preserved. In fine regions, the value of $\lambda_{s,1}$ is close to 0 so these wavelet coefficients will be close to 0 after shrinkage. In the coarse component of the image, as shown in Fig. 4.2(b), most of the coarse region in the image (upper region in Fig. 4.2(a)) is preserved while most of the fine region in the image (lower region in Fig. 4.2(a)) is suppressed.

For the fine component, W_1 is the opposite of S_1 as:

$$W_1(i,j) = \begin{cases} 1 & \text{if } |h(i,j)| > \delta_h \text{ or} \\ & \quad |v(i,j)| > \delta_v \text{ or} \\ & \quad |d(i,j)| > \delta_d \\ 0 & \text{otherwise} \end{cases}, \tag{4.15}$$

where $h(i,j)$, $v(i,j)$, $d(i,j)$, δ_h, δ_v and δ_d are the same as those in Eq. 4.14, so coarse regions will be suppressed. Fig. 4.2(c) shows the fine component of the image, where the fine region is mostly preserved while the coarse region is mostly suppressed.



(a) original (b) coarse component (c) fine component

Figure 4.2: The decomposition of an image by coarseness. (a) is the original image, (b) is the coarse component exhibiting coarseness strongly, and (c) is the fine component exhibiting coarseness weakly.

Contrast

Contrast is the measurement of the change of the pixel intensities of the texture. High-contrast (strong contrast) texture corresponds to that the intensities change rapidly over space within the local neighbourhood, and low-contrast (weak) texture corresponds to a more gradual change in the intensities in the local neighbourhood. Unlike coarseness, the low-level definition of texture contrast is identical to its high-level description. To distinguish the high and low contrast areas of the image, the dictionaries and the characteristic maps S_2 (high-contrast) and W_2 (low-contrast) are based on the magnitudes of the wavelet coefficients in the approximate sub-band reflecting the distribution of pixel intensities of the image.

The characteristic map S_2 for the high contrast component is defined as:

$$S_2(i,j) = \begin{cases} 1 & \text{if } \Delta x > \delta_x \text{ or} \\ & \quad \Delta y > \delta_y \\ 0 & \text{otherwise} \end{cases} . \tag{4.16}$$

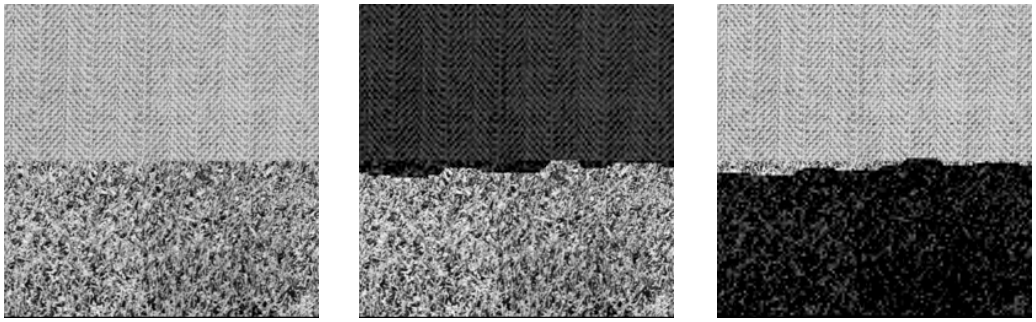
$$\Delta x = |a(x+1,y) - a(x,y)|$$
$$\Delta y = |a(x,y+1) - a(x,y)|$$

In high contrast regions, where the approximate magnitude of one of the first partial horizontal and vertical derivatives Δx or Δy exceed the thresholds δx and δy respectively, the value of S_2 is 1 so $\lambda_{s,2}(x,y)$ are close to 1, therefore these wavelet coefficients are preserved. Low-contrast regions will be suppressed because there are few pixels with large intensity changes and the $\lambda_{s,2}(x,y)$ are close to 0, shown as Fig. 4.3(b).

For the low contrast component, W_2 is defined opposite to the high contrast map S_2:

$$W_2(i,j) = \begin{cases} 1 & \text{if } \Delta x < \delta_x \ \& \\ & \quad \Delta y < \delta_y \\ 0 & \text{otherwise} \end{cases} . \tag{4.17}$$
$$\Delta x = |a(x+1,y) - a(x,y)|$$
$$\Delta y = |a(x,y+1) - a(x,y)|$$

W_2 is 1 only when both of the partial derivatives are below the thresholds, which results in the suppression of high contrast regions as in Fig. 4.3(c).



(a) original (b) high-contrast component (c) low-contrast component

Figure 4.3: The decomposition of an image corresponding to contrast. (a) is the original image, (b) is the high-contrast component exhibiting contrast strongly, and (c) is the low-contrast component exhibiting contrast weakly.

Directionality

Directionality of texture measures the orientation of the local texture. Though the orientation of texture could be anywhere in the range from 0 to π, herein we consider the horizontal texture as the image content with strong directionality while the vertical texture as that with weak directionality. When measured in low-level description, the edge directions are used to represent the directions of the texture. Therefore, we require dictionaries to represent the horizontal and vertical edge directions separately. Since the wavelet coefficients in high frequency sub-bands reflect the occurrence of edges with horizontal, vertical and diagonal directions, the dictionaries for horizontal and vertical texture contents, and the corresponding strong and weak characteristic maps S_3 and W_3 are based on the magnitudes of the wavelet coefficients in horizontal and vertical sub-bands respectively, which are defined as follows:

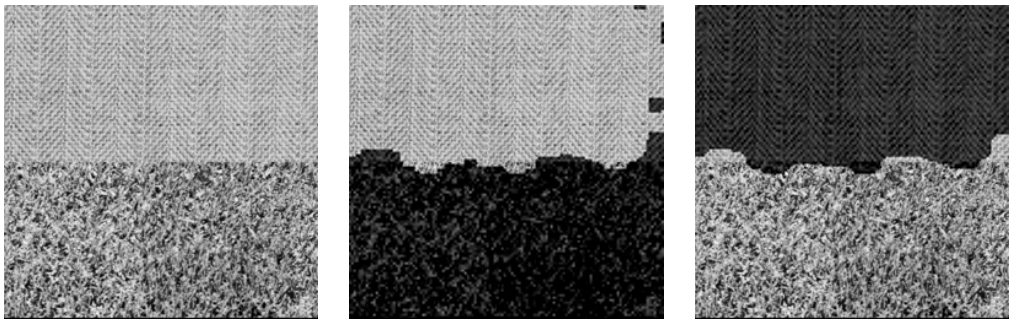
$$S_3(i,j) = \begin{cases} 1 & \text{if } |h(i,j)| > \delta_h \ \& \ |v(i,j)| \le \delta_v \\ 0 & \text{otherwise} \end{cases} , \tag{4.18}$$

and

$$W_3(i,j) = \begin{cases} 1 & \text{if } |v(i,j)| > \delta_v \ \& \ |h(i,j)| \leq \delta_h \\ 0 & \text{otherwise} \end{cases}, \tag{4.19}$$

where $h(i,j)$ and $v(i,j)$ are the wavelet coefficients in horizontal and vertical sub-bands respectively, δ_h and δ_v are the corresponding thresholds.

In regions with horizontal directionality, coefficients are preserved because the horizontal coefficients are larger than the threshold δ_h, vertical coefficients are below the threshold δ_v, and S_3 is 1. In regions with non-horizontal directionality, neither of these conditions are met, and S_3 is zero, suppressing such regions as shown as Fig. 4.4(b). Similarly, W_3 suppresses regions of non-vertical directionality as shown in Fig. 4.4(c).



(a) original (b) horizontal component (c) vertical component

Figure 4.4: The decomposition of an image corresponding to directionality. (a) is the original image, (b) is the horizontal component exhibiting directionality strongly, and (c) is the vertical component exhibiting directionality weakly.

Line-likeness

According to the definition of line-likeness, the line-like texture and the non-line-like texture show high and low coincidence of edge directions in the neighbourhood, respectively. Measured in low-level description, the number of edges with the same direction can be used to differentiate textures with strong and weak line-likeness: line-like contents of the image correspond to the regions where most edges are in the similar direction, while non-line-like contents of the image correspond to the regions where the edge directions vary a lot. Since the high-frequency sub-bands of the wavelet transform are with strong directionality (horizontal, vertical and diagonal), the dictionaries used to discriminate line-like and non-line-like regions are based on the measurement of the number of coefficients in every high-frequency sub-band. Therefore, the characteristic maps for line-like component S_4 and non-line-like component W_4 are defined as:

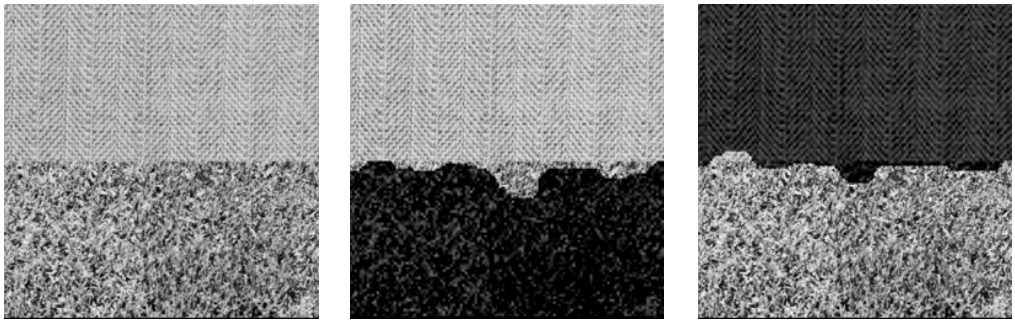
$$S_4(i,j) = \begin{cases} 1 & \text{if } |h(i,j)| > \delta_h \ \& \ |v(i,j)| \leq \delta_v \ \& \ |d(i,j)| \leq \delta_d \\ 1 & \text{if } |h(i,j)| \leq \delta_h \ \& \ |v(i,j)| > \delta_v \ \& \ |d(i,j)| \leq \delta_d \\ 1 & \text{if } |h(i,j)| \leq \delta_h \ \& \ |v(i,j)| \leq \delta_v \ \& \ |d(i,j)| > \delta_d \\ 0 & \text{otherwise} \end{cases}, \tag{4.20}$$

and

$$W_4(i,j) = \begin{cases} 1 & \text{if } |h(i,j)| > \delta_h \ \& \ |v(i,j)| > \delta_v \\ 1 & \text{if } |h(i,j)| > \delta_h \ \& \ |d(i,j)| > \delta_d \\ 1 & \text{if } |v(i,j)| > \delta_v \ \& \ |d(i,j)| > \delta_d \\ 1 & \text{if } |h(i,j)| < \delta_h \ \& \ |v(i,j)| < \delta_v \ \& \ |d(i,j)| < \delta_d \\ 0 & \text{otherwise} \end{cases}, \tag{4.21}$$

where $h(i,j)$, $v(i,j)$, $d(i,j)$ are the wavelet coefficients in high-frequency sub-bands, while δ_h, δ_v and δ_d are the corresponding thresholds.

The line-like characteristic map S_4 is 1 only in regions where there is a large magnitude in only one high frequency sub-band, resulting in a $\lambda_{s,4}$ which preserves coefficients in such regions, as in Fig. 4.5(b). Conversely, the non-line-like characteristic map W_4 is 1 only in regions where there is strong coefficient magnitude in more than one direction, which causes $\lambda_{w,4}$ to preserve non-line-like regions as in Fig. 4.5(c).



(a) original (b) line-like component (c) non-line-like component

Figure 4.5: The decomposition of an image corresponding to directionality. (a) is the original image, (b) is the line-like component exhibiting line-likeness strongly, and (c) is the non-line-like component exhibiting line-likeness weakly.

4.4.2 Incoherence of the texture characteristic dictionaries

The ability of the wavelet thresholding methods with different thresholding functions in highlighting different texture characteristics are described in the above chapter. However, being the dictionaries for representing the certain texture characteristics, these wavelet thresholding methods needs to be incoherent in representing either strong or weak characteristics: the component exhibiting a certain texture characteristic can be considered as the combination of few columns of the corresponding transformation while cannot be reconstructed by the sum of the coefficients of the other transformation. In the following part, we will discuss the incoherence of the dictionaries corresponding to strong characteristics and weak characteristics respectively.

Incoherence of dictionaries corresponding to strong characteristics

According to the assumption of the dictionaries of MCA introduced in Chapter 4.3.1, the incoherence of the dictionaries for strong characteristics representation should be discussed in the following two aspects:

1. the "strong-characteristics" dictionaries can represent the components with strong characteristics sparsely, that is, the components can be considered as the linear combination of few columns of the transformations employed by the "strong-characteristics" dictionaries;

2. the "strong-characteristics" dictionaries **cannot** represent the components with weak characteristics sparsely, that is, the components **cannot** be considered as the linear combination of few columns of the transformations employed by the "strong-characteristics" dictionaries.

Since the transformations exploited by the proposed dictionaries $T_{s,i}$ are all wavelet thresholding methods, the number of columns of the transformations used to compose the components $s_{s,i}$ can be approximated by the number of sub-bands where the wavelet coefficients after thresholding by $T_{s,i}$ are able to reconstruct the component $s_{s,i}$. The transformation $T_{s,i}$ of $s_{s,i}$ is defined as:

$$T_{s,i}s_{s,i}: \begin{cases} [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] = \mathrm{dwt}(s_{s,i}) \\ S_i = f(a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}, \delta_{s,i}) \\ [\widehat{a_{s,i}}, \widehat{h_{s,i}}, \widehat{v_{s,i}}, \widehat{d_{s,i}}] = \lambda(S_i) \cdot [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] \end{cases}, \tag{4.22}$$

where $\mathrm{dwt}(\cdot)$ represents the discrete wavelet transform, S_i is the map for i-th strong characteristic, and $\lambda(\cdot)$ denotes the shrinkage function based on the characteristic map S_i as defined in Eq. 4.13. As discussed in Chapter 4.4, the wavelet coefficients $[a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}]$ of the component $s_{s,i}$ exhibiting the i-th characteristic strongly are mostly preserved by the shrinkage function $\lambda(S_i)$ with a group of calculated thresholds $\delta_{s,i}$:

$$\begin{gathered} S_i = f(a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}, \delta_{s,i}) \approx 1 \\ [\widehat{a_{s,i}}, \widehat{h_{s,i}}, \widehat{v_{s,i}}, \widehat{d_{s,i}}] = \lambda(S_i) \cdot [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] \approx [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] \end{gathered}. \tag{4.23}$$

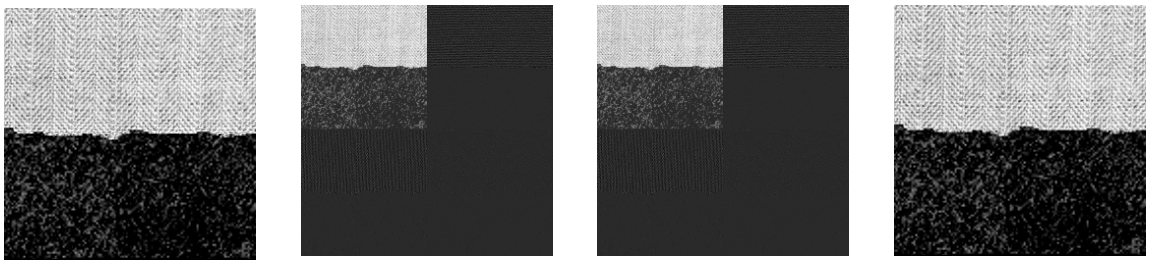
Therefore, the reconstruction of the component $s_{s,i}$ can be implemented as:

$$\mathrm{idwt}(\widehat{a_{s,i}}, \widehat{h_{s,i}}, \widehat{v_{s,i}}, \widehat{d_{s,i}}) \approx \mathrm{idwt}(a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}) = s_{s,i}, \tag{4.24}$$

where $\mathrm{idwt}(\cdot)$ represent the inverse discrete wavelet transformation. The component $s_{s,i}$ can be calculated from the thresholded wavelet coefficients in at most 4 sub-bands. Considering that the magnitudes of the coefficients in some sub-bands are so small that they can be omitted, the reconstruction of the component $s_{s,i}$ can be calculated from the coefficients using less than 4 sub-bands. We will now show that the component $s_{s,i}$ exhibiting i-th characteristic strongly can be represented sparsely by the dictionaries $T_{s,i}$ corresponding to the i-th strong characteristic by using a small number of coefficients thresholded by the transformations exploited in $T_{s,i}$ such that:

$$\|T_{s,i}s_{s,i}\|_0 \leq 4. \tag{4.25}$$

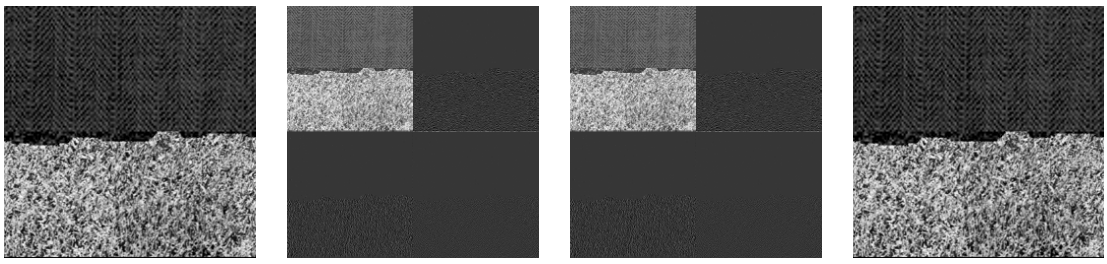
Fig. 4.6 shows the coarse (strong coarseness) component $s_{s,1}$, the wavelet coefficients before and after thresholding by $T_{s,1}$, and the reconstruction of the coarse component with the thresholded wavelet coefficients. Since the magnitudes of both the original coefficients $h_{s,1}, v_{s,1}, d_{s,1}$ and the thresholded coefficients $\widehat{h_{s,1}}, \widehat{v_{s,1}}, \widehat{d_{s,1}}$ of the coarse component are quite small, the component $s_{s,1}$ can be approximately reconstructed by using only the thresholded coefficients in the approximate sub-band $\widehat{a_{s,1}}$, resulting in the l_0 norm of $T_{s,1}s_{s,1}$ being equal to 1.



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

Figure 4.6: The representation of the coarse component by the transformation used in the "strong-coarseness" dictionary $T_{s,1}$. (a) is the coarse component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,1}$, (d) is the coarse component reconstructed by the wavelet coefficients thresholded by $T_{s,1}$.

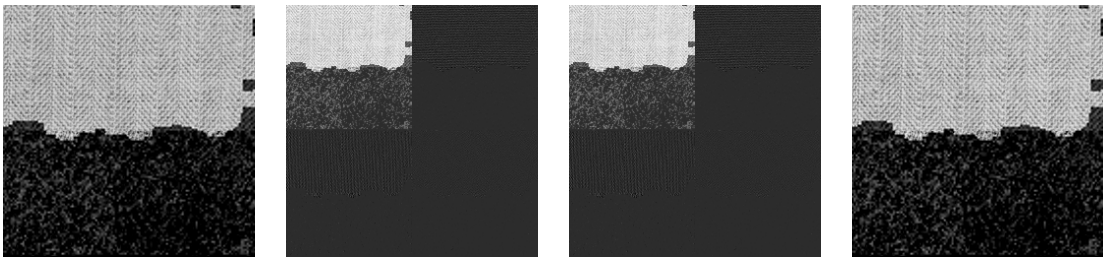
Fig. 4.7 shows the process of representing a high-contrast (strong contrast) component $s_{s,2}$ with the strong contrast dictionary $T_{s,2}$. The magnitudes of $h_{s,2}, d_{s,2}$ and those of $\widehat{h_{s,2}}, \widehat{d_{s,2}}$ are small because the gradients vary most in vertical direction. Then the component $s_{s,2}$ can be reconstructed by only using the approximate and vertical sub-bands, that is, $\|T_{s,2}s_{s,2}\|_0 = 2 < 4$.



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

Figure 4.7: The representation of the high-contrast component by the transformation used in the "strong-contrast" dictionary $T_{s,2}$. (a) is the high-contrast component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,2}$, (d) is the high-contrast component reconstructed by the wavelet coefficients thresholded by $T_{s,2}$.

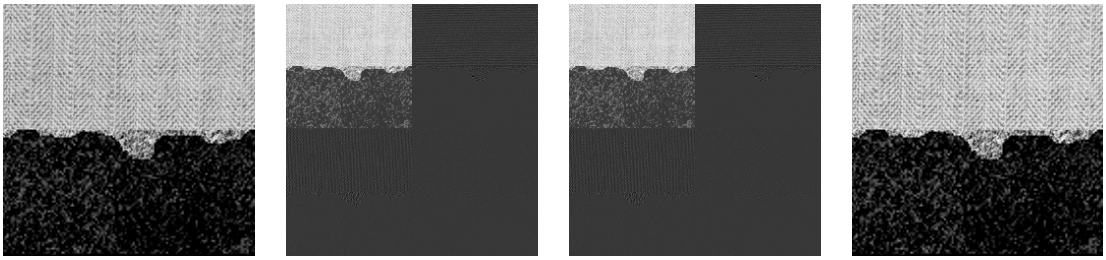
Fig. 4.8 shows the transformation and reconstruction of a horizontal (strong directionality) component $s_{s,3}$ with the strong directionality dictionary $T_{s,3}$. The magnitudes of coefficients $a_{s,3}, h_{s,3}$ are high and mostly preserved. The coefficients in only the approximate and horizontal sub-bands can reconstruct this horizontal component well. Consequently the l_0 norm is: $\|T_{s,3}s_{s,3}\|_0 = 2 < 4$.



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

Figure 4.8: The representation of the horizontal component by the transformation used in the "strong-directionality" dictionary $T_{s,3}$. (a) is the horizontal component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,3}$, (d) is the horizontal component reconstructed by the wavelet coefficients thresholded by $T_{s,3}$.

Fig. 4.9 shows the coefficients of the line-like (strong line-likeness) component $s_{s,4}$ before and after the transformation $T_{s,4}$ corresponding to the strong line-likeness. Since the textures in this example component are mostly in horizontal-like directions, the magnitudes of coefficients $a_{s,4}, h_{s,4}$ are preserved with high values. Therefore, the line-like component $s_{s,4}$ can be considered as the combination of the thresholded coefficients $\widehat{a_{s,4}}, \widehat{h_{s,4}}$ in only the approximate and horizontal sub-bands. Then the dictionary $T_{s,4}$ can represent the component $s_{s,4}$ sparsely with the l_0 norm small: $\|T_{s,4}s_{s,4}\|_0 = 2 < 4$.



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

Figure 4.9: The representation of the line-like component by the transformation used in the "strong-line-likeness" dictionary $T_{s,4}$. (a) is the line-like component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,4}$, (d) is the line-like component reconstructed by the wavelet coefficients thresholded by $T_{s,4}$.

Now we apply the transformations for the strong characteristic components, $T_{s,i}$, respectively to the weak characteristic components $s_{w,i}$ and show that these transformations cannot represent the weak components sparsely. The transformation $T_{s,i}$ of $s_{w,i}$ is:

$$T_{s,i}s_{w,i} : \begin{cases} [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] = \mathrm{dwt}(s_{w,i}) \\ S_i = f(a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}, \delta_{s,i}) \\ [\widehat{a_{w,i}}, \widehat{h_{w,i}}, \widehat{v_{w,i}}, \widehat{d_{w,i}}] = \lambda(S_i) \cdot [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] \end{cases}, \tag{4.26}$$

where $\delta_{s,i}$ is the same as that in Eq. 4.22 because the same transformation is applied to the components. As a result, the wavelet coefficients $[a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}]$ of the component $s_{w,i}$ with weak aspect of the i-th

characteristic are thresholded out by the thresholding function $\lambda(S_i)$ with the thresholds $\delta_{s,i}$ for preserving strong aspect of the i-th characteristic:

$$\begin{gathered} S_i = f(a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}, \delta_{s,i}) \approx 0 \\ [\widehat{a_{w,i}}, \widehat{h_{w,i}}, \widehat{v_{w,i}}, \widehat{d_{w,i}}] = \lambda(S_i) \cdot [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] \approx [0,0,0,0] \end{gathered}. \tag{4.27}$$

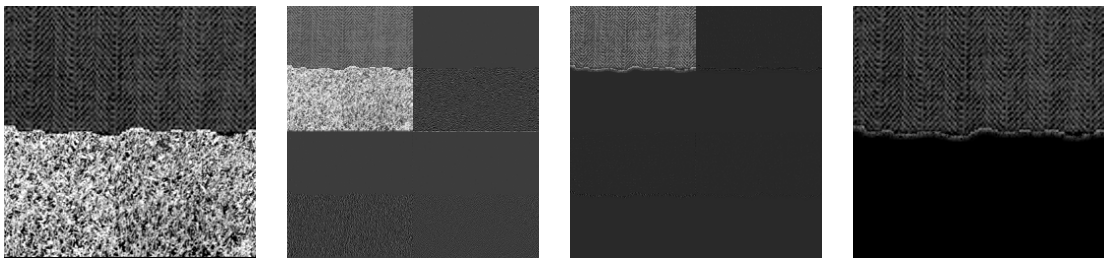
Then the inverse discrete wavelet transform of these thresholded coefficients is:

$$\text{idwt}(\widehat{a_{w,i}}, \widehat{h_{w,i}}, \widehat{v_{w,i}}, \widehat{d_{w,i}}) \approx 0 \neq s_{w,i}. \tag{4.28}$$

As shown above, the component $s_{w,i}$ cannot be reconstructed even by using the thresholded coefficients in all of the sub-bands. Therefore, it can be concluded that the i-th weak characteristic components $s_{w,i}$ cannot be represented by the dictionaries $T_{s,i}$ designed for the i-th strong characteristic sparsely, since using all the coefficients thresholded by the transformations employed by $T_{s,i}$ cannot reconstruct the component $s_{w,i}$:

$$\|T_{s,i} s_{w,i}\|_0 > 4. \tag{4.29}$$

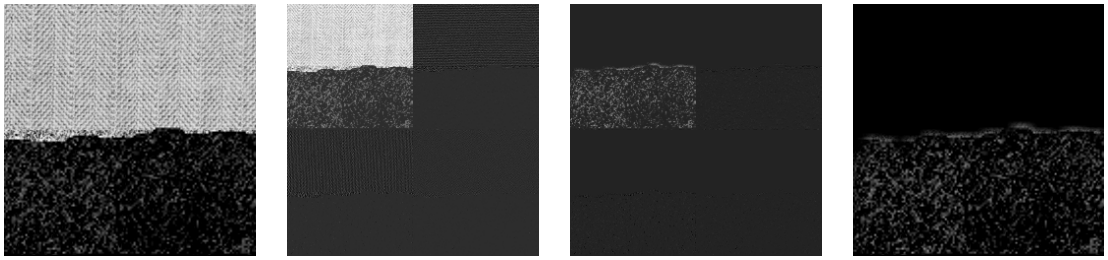
Fig. 4.10 to Fig. 4.13 show the process of transformation, thresholding and reconstruction of the fine (weak coarseness), low-contrast (weak contrast), vertical (weak directionality) and non-line-like (weak line-likeness) components $s_{w,i},\ i = 1,2,3,4$ of the same example image by the same dictionaries $T_{s,i},\ i = 1,2,3,4$ for the strong components. As illustrated in Eq. 4.28, the reconstructions of these components by the coefficients thresholded with $T_{s,i},\ i = 1,2,3,4$ yield nearly untextured components. In the other words, the components $s_{w,i}$ cannot be considered as the combination of few columns of the dictionaries $T_{s,i}$.



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

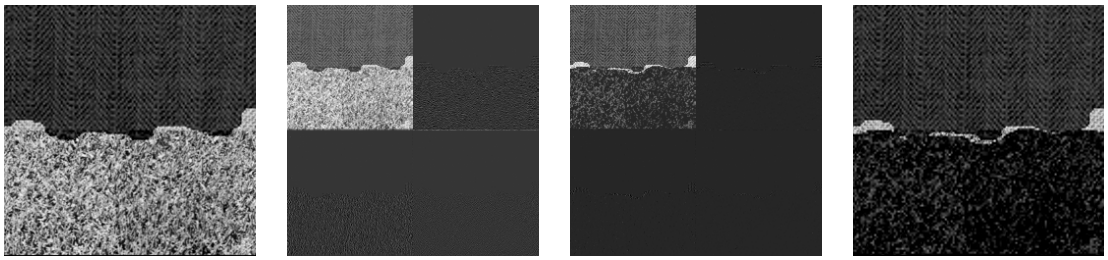
Figure 4.10: The representation of the fine component by the transformation used in the "strong-coarseness" dictionary $T_{s,1}$. (a) is the fine component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,1}$, (d) is the fine component reconstructed by the wavelet coefficients thresholded by $T_{s,1}$.

Based on the above discussion, the proposed dictionaries $T_{s,i},\ i = 1,2,3,4$ can represent the strong characteristic components $s_{s,i},\ i = 1,2,3,4$ sparsely while cannot represent the weak characteristic component $s_{w,i},\ i = 1,2,3,4$ sparsely. Therefore, the dictionary $T_{s,i},\ i = 1,2,3,4$ is incoherent for the decomposition of components exhibiting the i-th characteristic strongly.



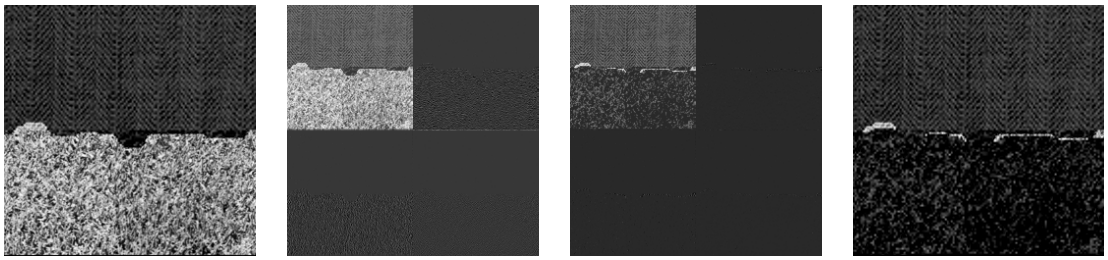
(a) (b) (c) (d)

Figure 4.11: The representation of the low-contrast component by the transformation used in the "strong-contrast" dictionary $T_{s,2}$. (a) is the low-contrast component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,2}$, (d) is the low-contrast component reconstructed by the wavelet coefficients thresholded by $T_{s,2}$.



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

Figure 4.12: The representation of the vertical component by the transformation used in the "strong-directionality" dictionary $T_{s,3}$. (a) is the vertical component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,3}$, (d) is the vertical component reconstructed from the wavelet coefficients thresholded by $T_{s,3}$.



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

Figure 4.13: The representation of the non-line-like component by the transformation used in the "strong-line-likeness" dictionary $T_{s,4}$. (a) is the non-line-like component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{s,4}$, (d) is the non-line-like component reconstructed from the wavelet coefficients thresholded by $T_{s,4}$.

Incoherence of dictionaries corresponding to weak characteristics

Proof of the incoherence of dictionaries corresponding to weak characteristics are the same as those corresponding to strong characteristics, for which also need to discuss the following evidence:

1. the "weak-characteristics" dictionaries can represent the components with weak characteristics sparsely, which means the components can be considered as the combination of few columns of the transformation exploited in the "weak-characteristics" dictionaries;

2. the "weak-characteristics" dictionaries **cannot** represent the components with strong characteristics sparsely, equally saying that the components **cannot** be considered as the combination of few columns of transformations exploited by those "weak-characteristics" dictionaries.

Applying the transformations $T_{w,i}$ to the components $s_{w,i}$ that represent the i-th weak characteristic is described as:

$$T_{w,i}s_{w,i}: \begin{cases} [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] = \mathrm{dwt}(s_{w,i}) \\ W_i = f(a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}, \delta_{w,i}) \\ [\widehat{a_{w,i}}, \widehat{h_{w,i}}, \widehat{v_{w,i}}, \widehat{d_{w,i}}] = \lambda(W_i) \cdot [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] \end{cases}, \tag{4.30}$$

where $\delta_{w,i}$ are the thresholds controlling the thresholding maps W_i so that the wavelet coefficients $[a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}]$ of the components $s_{w,i}$ exhibiting the i-th characteristic weakly are mostly preserved by the thresholding function $\lambda(W_i)$:

$$\begin{gathered} W_i = f(a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}, \delta_{w,i}) \approx 1 \\ [\widehat{a_{w,i}}, \widehat{h_{w,i}}, \widehat{v_{w,i}}, \widehat{d_{w,i}}] = \lambda(S_i) \cdot [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] \approx [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] \end{gathered}. \tag{4.31}$$

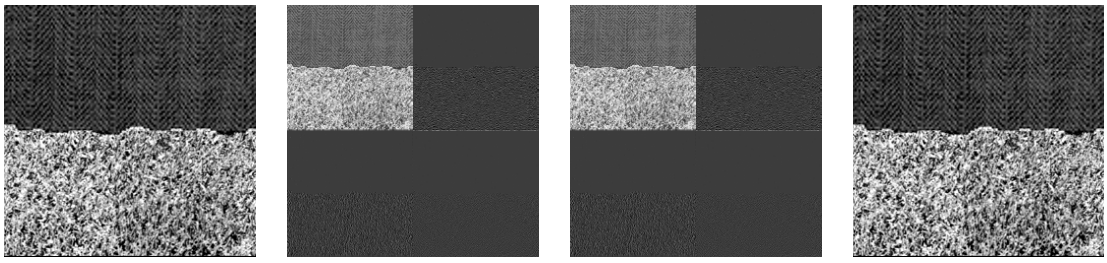
As a result, the component $s_{w,i}$ is close to the inverse discrete wavelet transform of those thresholded wavelet coefficients:

$$\mathrm{idwt}(\widehat{a_{w,i}}, \widehat{h_{w,i}}, \widehat{v_{w,i}}, \widehat{d_{w,i}}) \approx \mathrm{idwt}(a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}) = s_{w,i}. \tag{4.32}$$

Similar to the conclusion for Eq. 4.24, the component $s_{w,i}$ can be reconstructed by using the coefficients thresholded by $T_{w,i}$ in no more than the 4 sub-bands. Therefore, the components $s_{w,i}$ exhibiting the i-th characteristic weakly can be represented by the corresponding dictionaries $T_{w,i}$ sparsely, because only some of the subbands thresholded by the transformation $T_{w,i}$ are needed to reconstruct the component $s_{w,i}$:

$$\|T_{w,i}s_{w,i}\|_0 \leq 4. \tag{4.33}$$

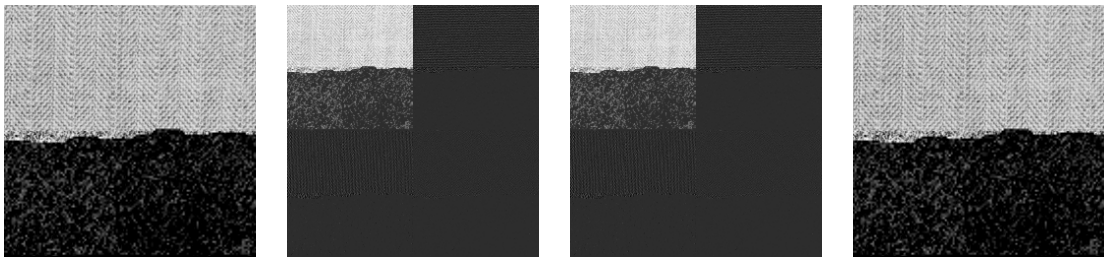
Fig. 4.14 shows the transformation and reconstruction of an example fine (weak coarseness) component $s_{w,1}$. The quantity of edges in the local neighbourhood of the fine component is large and the edges are close to vertical directions, so the coefficients in the approximate sub-band and the vertical sub-band are preserved with high magnitude. Then the component $s_{w,1}$ can be considered as the combination of the approximate and vertical wavelet sub-bands, $\|T_{w,1}s_{w,1}\|_0 = 2 < 4$.



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

Figure 4.14: The representation of the fine component by the transformation used in the "weak-coarseness" dictionary $T_{w,1}$. (a) is the fine component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,1}$, (d) is the fine component reconstructed by the wavelet coefficients thresholded by $T_{w,1}$.

Fig. 4.15 shows the process of representing the low-contrast (weak contrast) component $s_{w,2}$ by the dictionary $T_{w,2}$ corresponding to the weak characteristic. The magnitudes of wavelet coefficients in the three high-frequency sub-bands are low and those of coefficients in the approximate sub-band are preserved and close to the components. Therefore, the component $s_{w,2}$ is close to the inverse discrete wavelet transform of the coefficients only in approximate sub-band, with the l_0 norm: $\|T_{w,2}s_{w,2}\|_0 = 1 < 4$.

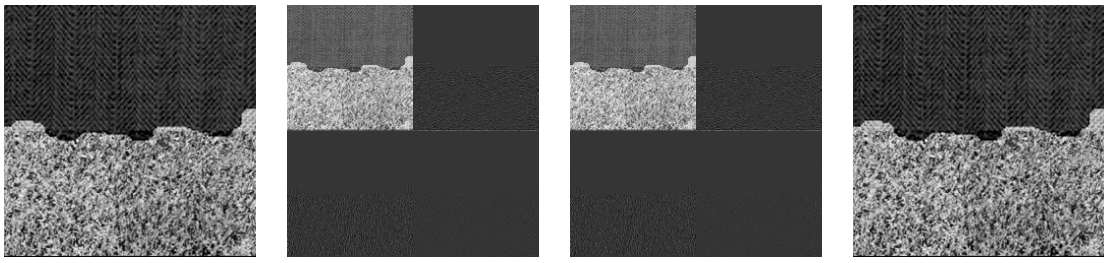


(a) (b) (c) (d)

Figure 4.15: The representation of the low-contrast component by the transformation used in the "weak-contrast" dictionary $T_{w,2}$. (a) is the low-contrast component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,2}$, (d) is the low-contrast component reconstructed by the wavelet coefficients thresholded by $T_{w,2}$.

Fig. 4.16 shows the coefficients of the vertical (weak directionality) component $s_{w,3}$ and the inverse transformation of the coefficients thresholded by the dictionary $T_{w,3}$. The coefficients of the approximate and vertical sub-bands obviously contain most of the information of the vertical edges in the component, and these coefficients are mostly preserved by the thresholding method in $T_{w,3}$. So the reconstruction of the component $s_{w,3}$ only requires the coefficients in the approximate and vertical sub-bands, leading to $\|T_{w,3}s_{w,3}\|_0 = 2 < 4$.

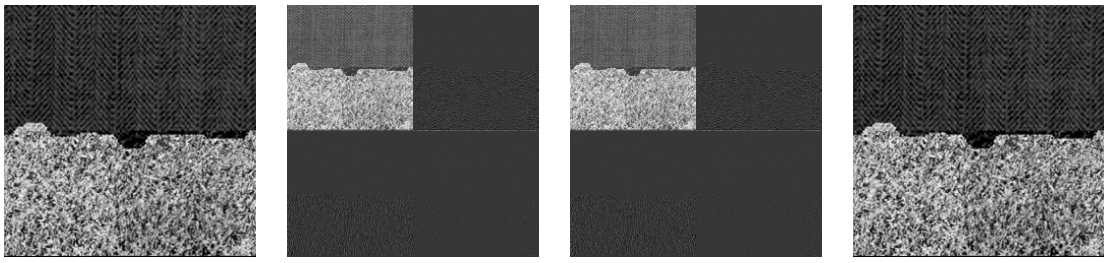
Fig. 4.17 shows the transformation and reconstruction of the non-line-like (weak line-likeness) component $s_{w,4}$ under the weak line-likeness dictionary $T_{w,4}$. After thresholded by the transformation in $T_{w,4}$, most of the wavelet coefficients are preserved and the coefficients in approximate and diagonal sub-bands reflect



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

Figure 4.16: The representation of the vertical component by the transformation used in the "weak-directionality" dictionary $T_{w,3}$. (a) is the vertical component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,3}$, (d) is the vertical component reconstructed by the wavelet coefficients thresholded by $T_{w,3}$.

most of the non-line-like textures in the components. The component $s_{w,4}$ can be sparsely represented by the $T_{w,4}$ by using thresholded coefficients in approximate and diagonal sub-bands, therefore the l_0 norm of the transformation of the component $s_{w,4}$ is small: $\|T_{w,4}s_{w,4}\|_0 = 2 < 4$.



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

Figure 4.17: The representation of the non-line-like component by the transformation used in the "weak-line-likeness" dictionary $T_{w,4}$. (a) is the non-line-like component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,4}$, (d) is the non-line-like component reconstructed by the wavelet coefficients thresholded by $T_{w,4}$.

To prove the remaining properties of incoherence, the transformations for weak characteristics $T_{w,i}$ are applied to the strong components $s_{s,i}$:

$$T_{w,i}s_{s,i}: \begin{cases} [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] = \mathrm{dwt}(s_{s,i}) \\ W_i = f(a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}, \delta_{w,i}) \\ [\widehat{a_{s,i}}, \widehat{h_{s,i}}, \widehat{v_{s,i}}, \widehat{d_{s,i}}] = \lambda(W_i) \cdot [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] \end{cases}, \tag{4.34}$$

where $\delta_{w,i}$ are the same thresholds as those in Eq. 4.30 because the same transformations $T_{w,i}$ are applied to the components. Then the wavelet coefficients $[a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}]$ of the component $s_{s,i}$ representing i-th strong characteristics are filtered by the thresholding functions $\lambda(W_i)$ because the threshold $\delta_{w,i}$ is calculated in Chapter 4.4.1 for preserving weak aspect of the i-th characteristic:

$$\begin{gathered} W_i = f(a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}, \delta_{w,i}) \approx 0 \\ [\widehat{a_{s,i}}, \widehat{h_{s,i}}, \widehat{v_{s,i}}, \widehat{d_{s,i}}] = \lambda(W_i) \cdot [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] \approx [0,0,0,0] \end{gathered}. \tag{4.35}$$

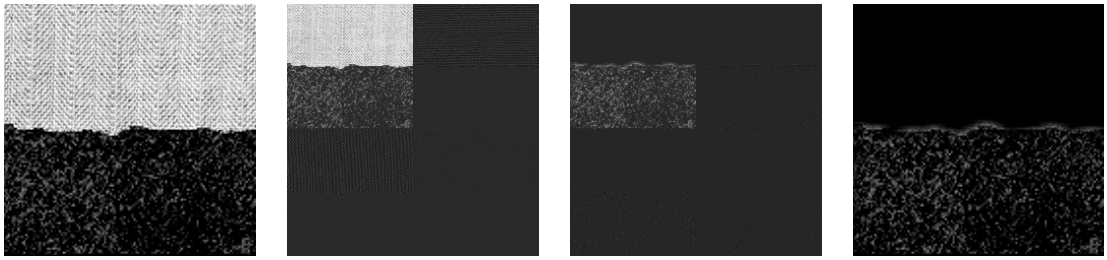
Therefore the inverse discrete wavelet transform of these thresholded coefficients is:

$$\text{idwt}(\widehat{a_{s,i}}, \widehat{h_{s,i}}, \widehat{v_{s,i}}, \widehat{d_{s,i}}) \approx 0 \neq s_{s,i}, \tag{4.36}$$

which illustrates that the component $s_{s,i}$ cannot be reconstructed by the coefficients thresholded by the transformations of dictionaries $T_{s,i}$, even using all the sub-bands. It can be therefore concluded that the components $s_{s,i}$ containing i-th strong characteristics cannot be represented by the coefficients from the dictionaries $T_{w,i}$ designed for the i-th weak characteristics sparsely, because using the coefficients transformed by $T_{w,i}$ in all the sub-bands cannot reconstruct the component $s_{s,i}$:

$$\|T_{w,i} s_{s,i}\| > 4. \tag{4.37}$$

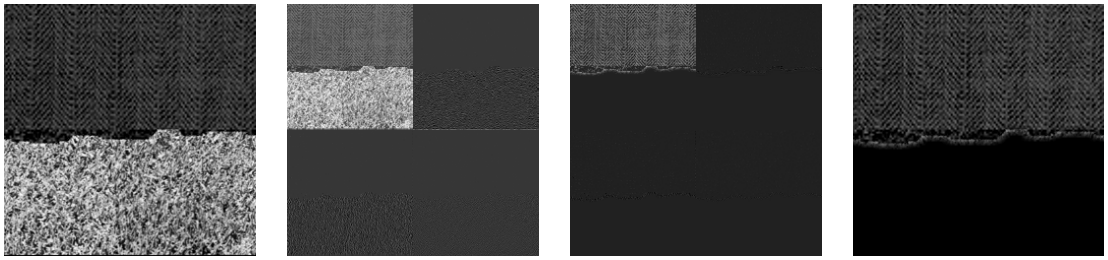
The transformation, thresholding and reconstruction of the coarse (strong coarseness), high-contrast (strong contrast), horizontal (strong directionality) and line-like (strong line-likeness) components $s_{s,i},\ i = 1, 2, 3, 4$ of the same example images with the same dictionaries $T_{w,i},\ i = 1, 2, 3, 4$ for the weak components are shown in Fig. 4.18 to Fig. 4.21. Demonstrated in Eq. 4.36, the magnitudes of the coefficients of these components transformed by $T_{w,i},\ i = 1, 2, 3, 4$ are close to zero, resulting in nearly untextured reconstructions of the components using these coefficients. Therefore the components $s_{s,i}$ cannot be represented as the combination of few columns of the dictionaries $T_{w,i}$.



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

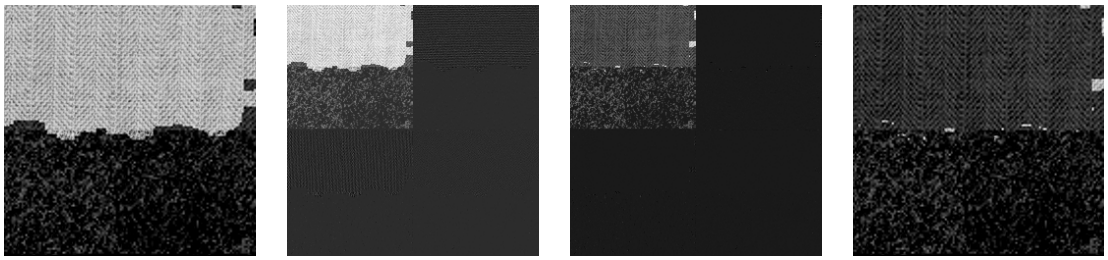
Figure 4.18: The representation of the coarse component by the transformation used in the "weak-coarseness" dictionary $T_{w,1}$. (a) is the coarse component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,1}$, (d) is the coarse component reconstructed from the wavelet coefficients thresholded by $T_{w,1}$.

It is proved above that the proposed dictionaries $T_{w,i},\ i = 1, 2, 3, 4$ can represent the weak characteristic components $s_{w,i},\ i = 1, 2, 3, 4$ sparsely and cannot represent the strong characteristic components $s_{s,i},\ i = 1, 2, 3, 4$ such sparsely. Therefore, the dictionaries $T_{w,i},\ i = 1, 2, 3, 4$ are incoherent in decomposing components exhibiting the corresponding characteristics weakly.



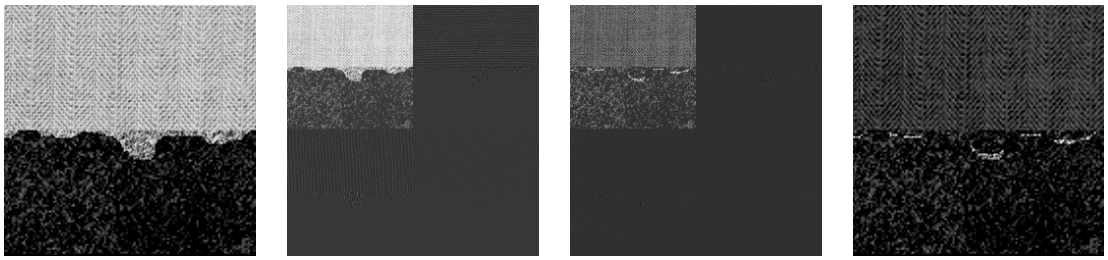
(a) (b) (c) (d)

Figure 4.19: The representation of the high-contrast component by the transformation used in the "weak-contrast" dictionary $T_{w,2}$. (a) is the high-contrast component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,2}$, (d) is the high-contrast component reconstructed from the wavelet coefficients thresholded by $T_{w,2}$.



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

Figure 4.20: The representation of the horizontal component by the transformation used in the "weak-directionality" dictionary $T_{w,3}$. (a) is the horizontal component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,3}$, (d) is the horizontal component reconstructed from the wavelet coefficients thresholded by $T_{w,3}$.



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

Figure 4.21: The representation of the line-like component by the transformation used in the "weak-line-likeness" dictionary $T_{w,4}$. (a) is the line-like component of the image, (b) and (c) are the wavelet coefficients before and after thresholded by the dictionary $T_{w,4}$, (d) is the line-like component reconstructed from the wavelet coefficients thresholded by $T_{w,4}$.

4.5 Experiments and analysis

In this chapter, the wavelet-based texture characteristic morphological component analysis (WT-TC-MCA) is tested and evaluated for its performance in decomposing images into components corresponding to different texture characteristics.

4.5.1 Experimental images

The first image set used in our experiments consists of mosaic texture images created by concatenating textures randomly selected from the Brodatz texture database [24] and the SIPI texture database [208], as shown in Fig. 4.22(a) through (e). The decomposition of this kind of images is relatively simple because each image region contains one type of homogeneous texture that need to be exhibited in the certain component. Therefore, this image set is used as the starting step in our image decomposition evaluation experiments.

In addition to the synthesized mosaic texture images, the natural images from the BSD500 database [135], as shown in Fig. 4.22(f) through (j), are used as the second image set in our experiments. The scenes in the natural images are more complex and the textures are not as homogeneous as those in mosaic images. Therefore, the natural image set is used to evaluate different image decomposition methods more comprehensively and more representative of actual decomposition problems.

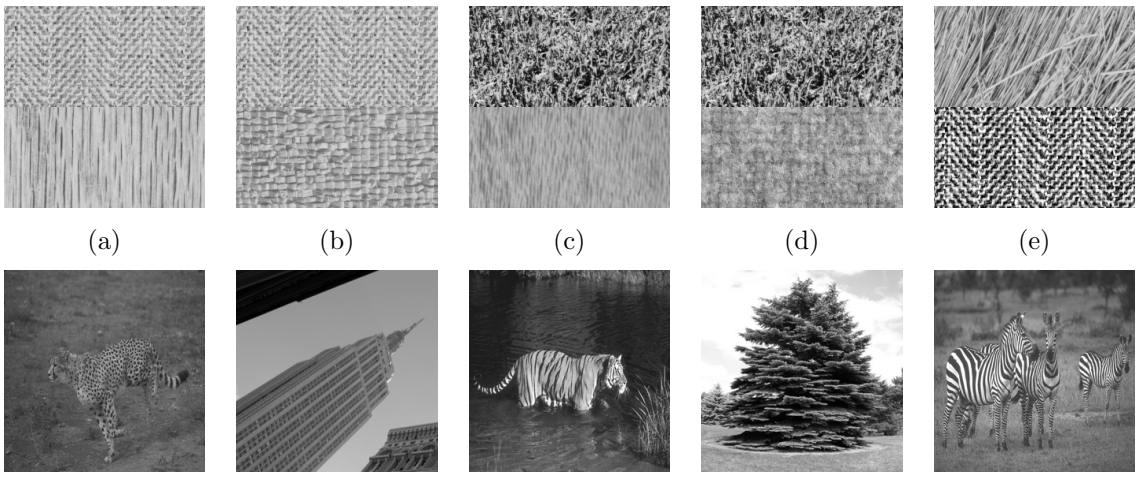


Figure 4.22: Examples of images used in the experiments. Rows: the mosaic texture images synthesized from 4 textures randomly selected from the Brodatz texture dataset [24] and the SIPI texture database [208] (top) and the natural images from the BSD500 database [135] (bottom).

4.5.2 Experimental benchmarks

There are two benchmarks which which we evaluate the performance of the WT-TC-MCA:

1. sensitivity and specificity: they are used to evaluate the accuracy of the decomposition by WT-TC-MCA, that is, how much of the texture component representing a certain texture characteristic can be accurately separated from others in the image;

2. threshold of the decomposable textural difference: it is used to measure the applicability of the WT-TC-MCA, that is, how different the textures in the image must be so that the image can be successfully decomposed.

Sensitivity and specificity

Both the sensitivity and specificity are statistical measurements of the performance of classifiers, and they can be used to evaluate the texture decomposition since the texture decomposition problem in our work is to separate the image into parts representing one texture characteristic "strongly" and "weakly":

- Sensitivity (also called the true positive rate) measures the proportion of positives that are correctly identified as such (e.g., the percentage of texture regions with strong characteristic that are correctly identified as the strong component of the image).

- Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g., the percentage of texture regions with weak characteristic that are correctly identified as the weak component of the image).

Therefore, the sensitivity and specificity are calculated based on the following 4 measurements: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Then the sensitivity and specificity for the decomposition corresponding to each texture characteristic, together with these basic measurements in each textural image was computed by comparing the decomposed result with the ground truth using the metric as follows:

$$\begin{aligned} \text{Sens} &= \frac{TP}{TP+FN} \\ \text{Spec} &= \frac{TN}{TN+FP} \\ TP &= \frac{|\{S(x,y)|\text{MSE}(S_{s,i}(x,y),G_{s,i}(x,y))\leq \text{Th}\}|}{N} \\ TN &= \frac{|\{S(x,y)|\text{MSE}(S_{w,i}(x,y),G_{w,i}(x,y))\leq \text{Th}\}|}{N} \\ FP &= \frac{|\{S(x,y)|\text{MSE}(S_{s,i}(x,y),G_{s,i}(x,y))> \text{Th}\}|}{N} \\ FN &= \frac{|\{S(x,y)|\text{MSE}(S_{w,i}(x,y),G_{w,i}(x,y))> \text{Th}\}|}{N} \end{aligned}, \tag{4.38}$$

where $S_{s,i}$ and $S_{w,i}$ denote the decomposed image component, $G_{s,i}$ and $G_{w,i}$ represent the ground truth image component, $\text{MSE}(S_{s,i}, G_{s,i})$ and $\text{MSE}(S_{w,i}, G_{w,i})$ measure the mean square error (MSE) [204] between the decomposed image components and the corresponding ground truths, Th is the threshold that discriminate the correctly and incorrectly decomposed components: the MSE between the component and the ground truth smaller than the threshold Th means the component is correctly decomposed from the image, otherwise the component is incorrectly decomposed from the image. N is the total number of the pixels in the image.

Threshold of the decomposable textural difference

In addition, we quantify the amount of variation in the measure of a particular texture characteristic in an image that is sufficient to permit decomposition of the image into strong and weak components of that characteristic. The threshold of the decomposable textural difference measures the ability of the WT-TC-MCA to separate similar textures with respect to one certain texture characteristic. For example, to decompose the given image into "strong" and "weak" coarseness components, WT-TC-MCA requires the coarseness of the two components differ by more than some threshold T_{cor}, or the image cannot be decomposed according to coarseness. Therefore, the threshold is calculated by using the following metric:

$$\begin{aligned} T_i &= min\left\{D_{i,j}\right\} \\ D_{i,j} &= \text{Mahalanobis}\left(C_{s,i,j}, C_{w,i,j}\right) \\ C_{s,i,j} &= C_i(s_{s,j}) \\ C_{w,i,j} &= C_i(s_{w,j}) \\ \text{subject to: } \text{Dacc}_{i,j} &= \frac{\text{Sens}(s_{i,j})+\text{Spec}(s_{i,j})}{2} > 0.9 \end{aligned} \quad , \tag{4.39}$$

where $\text{Dacc}_{i,j}$ is the decomposition accuracy calculated based on the decomposition sensitivity and decomposition specificity from Eq. 4.38, $C_{s,i,j}$ is the cluster of the i-th ($i = 1, 2, \ldots, 4$) texture characteristic of the "strong" component in the j-th image from the dataset, $C_{w,i,j}$ is the cluster of the i-th texture characteristic of the "weak" component in the j-th image from the dataset, $D_{i,j}$ is the Mahalanobis distance between the two components in j-th image with respect to the i-th texture characteristic, and the threshold T_i for the i-th characteristic is therefore the minimum distance of all the images in the given dataset.

4.5.3 Experimental methods

As discussed in Chapter 4.5.2, the performance of the wavelet-based texture characteristics morphological component analysis (WT-TC-MCA) in image decomposition includes two aspects: 1) the accuracy of the decomposition, and 2) the applicability of the decomposition.

For the evaluation of the accuracy, the WT-TC-MCA method is herein compared with the conventional morphological component analysis that apply the curvelet transform as the dictionary for the cartoon component and the local discrete cosine transform (LDCT) as the dictionary for the texture component. The experiment is implemented as follows:

1. the images are decomposed by WT-TC-MCA corresponding to different texture characteristics (coarseness, contrast, directionality and line-likeness) and the traditional MCA method to cartoon and texture components;

2. the sensitivity and specificity of the images decomposed by the WT-TC-MCA and the traditional MCA are calculated as Eq. 4.38 and compared for each texture characteristic.

For the evaluation of the applicability of the WT-TC-MCA, another experiment is implemented as follows:

1. the mosaic images are decomposed by the WT-TC-MCA according to different texture characteristics;

2. the thresholds for the decomposable textural difference with respect to each texture characteristic are calculated and compared with range of the certain texture features of all the images in the database.

4.5.4 Experimental results and analysis

Accuracy of the image decomposition

Fig. 4.23 shows the results of the decomposition of an example mosaic texture image according to coarseness, contrast, directionality and line-likeness by the WT-TC-MCA method, compared with the decomposition of the image to piece-wise cartoon and texture by the traditional MCA method. The conventional component analysis method can only decompose the image into smooth and oscillating components (the piece-wise cartoon and texture components). It cannot differentiate the different texture types in the image with respect to different texture characteristics, leading to low sensitivity and specificity when considering strong and weak aspects of the texture characteristic as positive and negative parts of the image. In contrast, the WT-TC-MCA is established based on the new model that an image is the combination of components representing different texture characteristics, and each of the wavelet-based dictionaries can highlight only one texture characteristic sparsely. Therefore, by the WT-TC-MCA method, for each texture characteristic, the image can be separated into two components representing the strong and weak aspects of the characteristic, respectively. In some cases of decomposition, there may exist "wavy boundaries" in both of the components. It is caused by over-suppressing or over-preserving the pixels that are around the "strong-weak" boundary of the input image and are therefore difficult to calculate their shrinkage coefficients. However, these pixels not decomposed well only occur around the boundaries in the synthesized texture images, while most of the pixels in the images are correctly represented in the corresponding components. As a result, according to any arbitrary texture characteristic, the WT-TC-MCA leads to higher sensitivity and specificity values. Table 4.2 shows the average sensitivity and specificity over all testing mosaic images according to different characteristics.

	MCA-CT		WT-TC-MCA	
Characteristic	Sensitivity	Specificity	Sensitivity	Specificity
Coarseness	54.26%	34.40%	**98.83%**	**98,51%**
Contrast	54.53%	40.06%	**97.76%**	**97.60%**
Directionality	45.29%	34.40%	**95.98%**	**97.93%**
line-likeness	63.50%	41.16%	**97.57%**	**97.02%**

Table 4.2: Average sensitivity and specificity of decomposing 435 synthetic images by MCA-CT and WT-TC-MCA. WT-TC-MCA exhibited better average performance than the MCA-CT method.

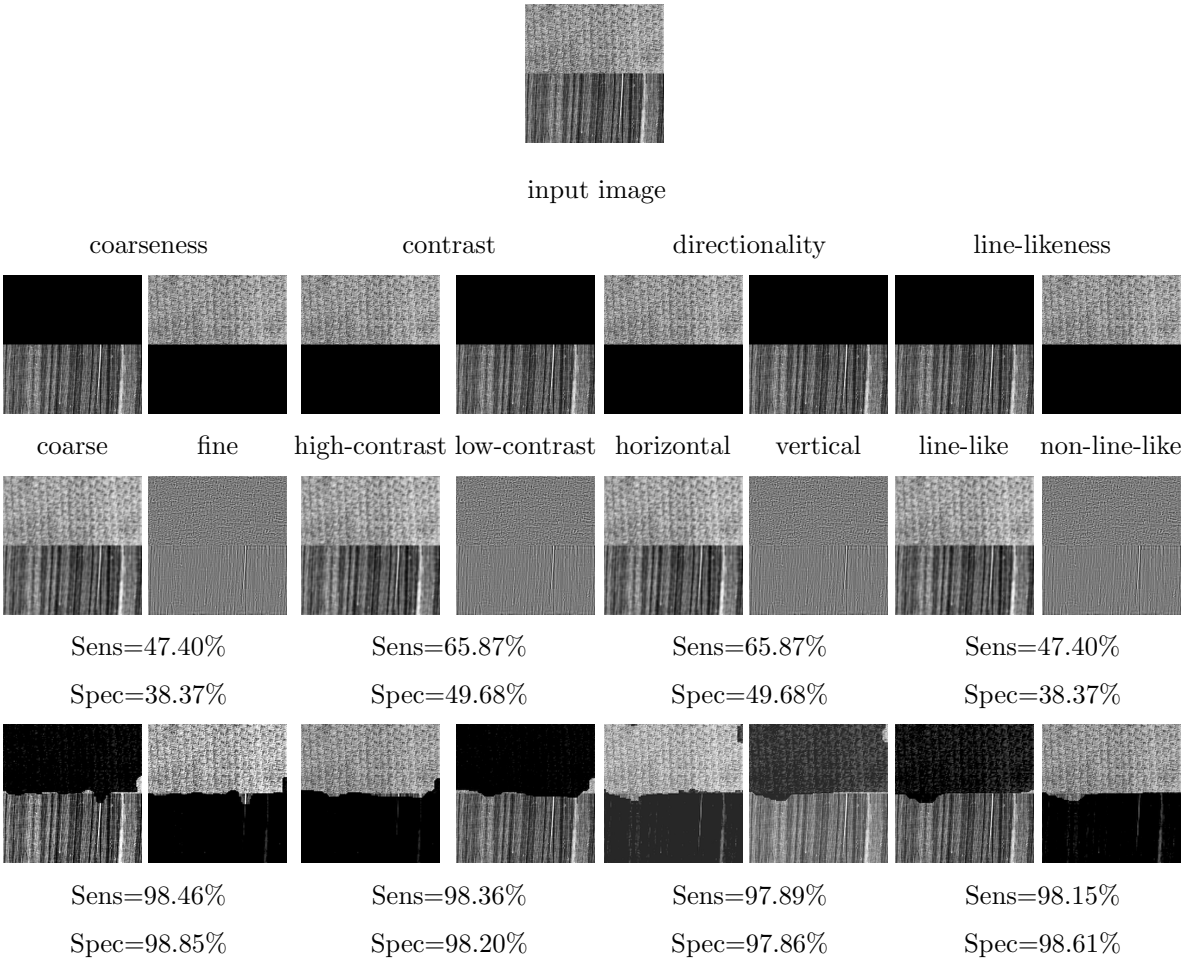


Figure 4.23: The decomposition results of a sample synthetic texture image with two textures in it by different decomposition methods. Rows: decomposition results by different methods: ground truth (top), traditional "cartoon+texture" MCA (middle), WT-TC-MCA (bottom). Columns: decomposition results according to different texture characteristics: coarse, fine, high-contrast, low-contrast, line-like, non-line-like, horizontal, vertical (from left to right). The sensitivity and specificity of the decomposition with respect to each characteristic are calculated as Eq. 4.38 with the threshold $Th = 0.1$.

While for the real-world image, Fig. 4.24 shows the results of decomposing the example real-world image according to coarseness, contrast, directionality and line-likeness by the WT-TC-MCA method, compared with other decomposition methods. Similar to the decomposition results of the mosaic images, the conventional components analysis based method can only separate the structure and textures from the image, without finding the components representing the certain texture characteristics. While the WT-TC-MCA can highlight the regions with the certain texture characteristics separately. Therefore, the sensitivity and specificity of separating the image into strong and weak characteristic components by WT-TC-MCA are higher than those generated by other methods. Table 4.3 shows the average sensitivity and specificity over all testing real-world images according to different characteristics.

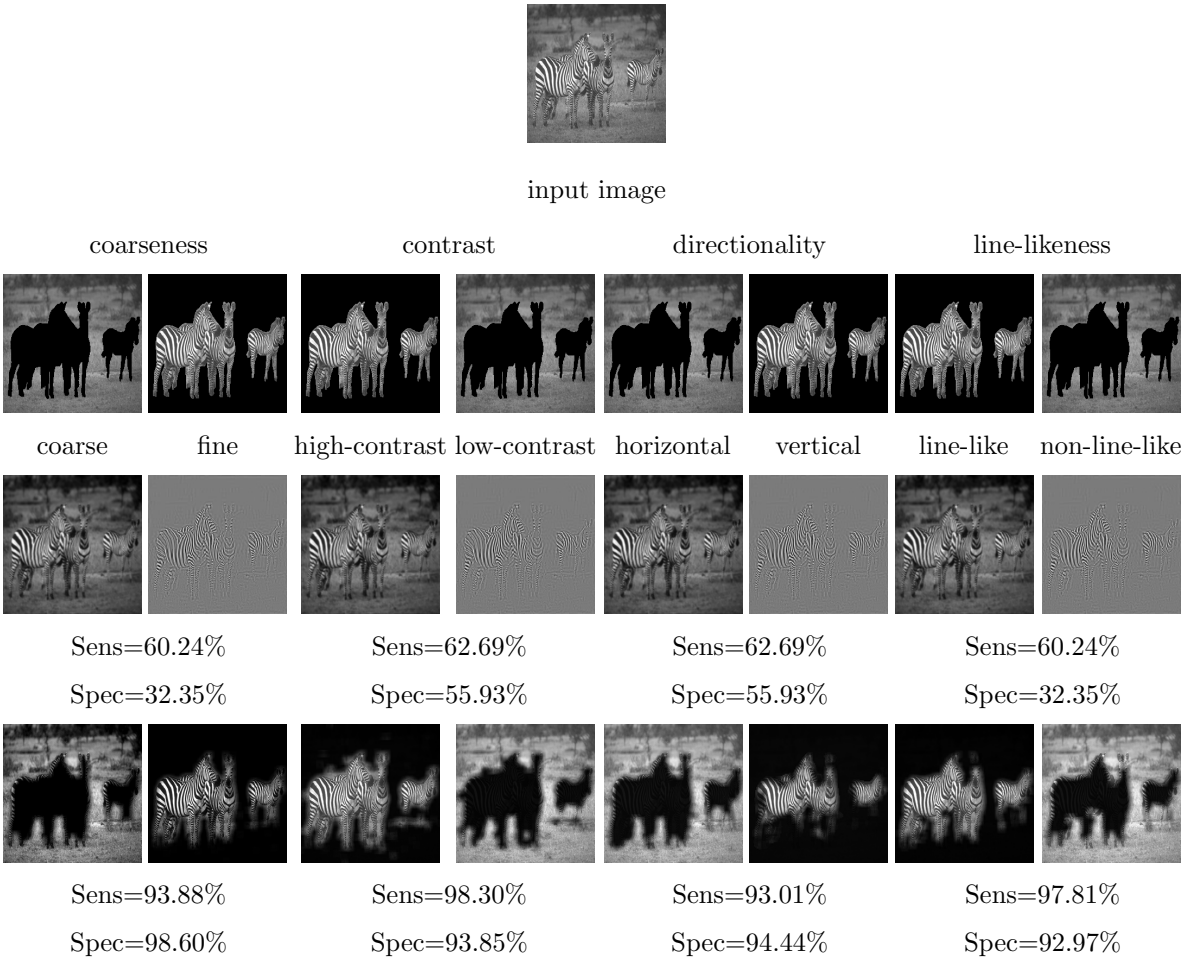


Figure 4.24: The decomposition result of a sample natural texture image with two textures in it by TC-MCA with different dictionary strategies. Rows: decomposition results by different methods: ground truth (top), non-sparse dictionary (middle), wavelet-based dictionary (bottom). Columns: decomposition results according to different texture characteristics: coarse, fine, high-contrast, low-contrast, line-like, non-line-like, horizontal, vertical (from left to right). The decomposition sensitivity and specificity are calculated as Eq. 4.38 with the threshold $Th = 0.1$.

	MCA-CT		TC-MCA-WT	
Characteristic	Sensitivity	Specificity	Sensitivity	Specificity
Coarseness	61.11%	24.50%	**94.67%**	**98.66%**
Contrast	62.33%	36.29%	**97.41%**	**95.04%**
Directionality	63.94%	36.19%	**93.96%**	**93.86%**
Line-likeness	62.71%	24.40%	**97.48%**	**95.09%**

Table 4.3: Average sensitivity and specificity of decomposing 200 real-world images by MCA-CT and WT-TC-MCA. WT-TC-MCA exhibited better average performance than the MCA-CT method.

Applicability of the WT-TC-MCA image decomposition method

Table. 4.4 shows part of the textures in Brodatz and SIPI datasets that compose the mosaic images, together with the value of the texture characteristics of these textures (ranked from high to low, that is, from strong to weak).

Fig. 4.25 to Fig. 4.28 show the decomposition of some example images composed by different textures in Table 4.4 with respect to coarseness, contrast, directionality and line-likeness, respectively. As shown in these examples, images where the textures are very different with respect to the certain texture characteristics are decomposed much more accuratly than those where the texture are quite similar. Fig. 4.29 show four extreme examples, where we apply the WT-TC-MCA method to decompose four images where there is only one single texture. In these examples of unsuccessful decompositions, shown as the bottom rows in Fig. 4.25 to Fig. 4.28 and the cases in Fig. 4.29, textures are fully preserved in either the "strong" components or the "weak" components, or over half of the textures are mistakenly preserved in the opposite characteristics. The reason for these unsuccessful decompositions is that there is little textural difference in the image with respect to coarseness, contrast, directionality and line-likeness, respectively. Therefore we need to find the threshold of the textural differences so that only the images where the textural differences are larger than the threshold can be decomposed by WT-TC-MCA according to the certain texture characteristics.

Fig. 4.30 shows the results of the relationship between the decomposition accuracy and the textural differences (Mahalanobis distance between two clusters of a certain texture feature) over the whole dataset, that is, 435 test pairwise textural images consisting of 30 different textures. We consider that the decompositions with the decomposition accuracies larger than 90% are successful while those with the decomposition accuracies smaller than 90% are unsuccessful. And if the decomposition is successful, we consider the image as a decomposable image. Although the very specific threshold value cannot be calculated from the experimental results, there exists a quite small range of the threshold that the image is decomposable according to each texture characteristic. For coarseness, the threshold is between $[0.79, 0.80]$, which means the images could be decomposed if the textural differences was larger than 0.80 in Mahalanobis distance while the images couldn't be decomposed if the textural difference was less than 0.79 in Mahalanobis distance with respect

Texture	Coarseness	Texture	Contrast	Texture	Line-likeness	Texture	Directionality
	0.8567		0.3254		0.8567		0.4785
	0.8395		0.3240		0.8652		0.4432
	0.8204		0.3135		0.7725		0.4192
	0.7886		0.2927		0.7150		0.3969
	0.7874		0.2901		0.6702		0.3762
	0.7648		0.2390		0.6595		0.3521
	0.7636		0.2357		0.6500		0.3442
	0.7623		0.2242		0.6285		0.3374
	0.7605		0.2068		0.5801		0.3336
	0.7506		0.1984		0.5778		0.3245
	0.7450		0.1966		0.5710		0.3214
	0.7380		0.1885		0.5557		0.2970
	0.7318		0.1866		0.5496		0.2951
	0.7295		0.1851		0.5490		0.2935
	0.7235		0.1692		0.5406		0.2874
	0.7153		0.1637		0.5251		0.2870
	0.7073		0.1574		0.5222		0.2832
	0.7033		0.1548		0.5090		0.2676
	0.6969		0.1547		0.5007		0.2480
	0.6748		0.1537		0.4990		0.2412
	0.6658		0.1415		0.4948		0.2364
	0.6632		0.1346		0.4829		0.2360
	0.6518		0.1316		0.4563		0.2282
	0.6511		0.1283		0.4363		0.2243
	0.6134		0.1218		0.4360		0.2230
	0.6121		0.1191		0.4350		0.2088

Table 4.4: Example textures taken from Brodatz and SIPI texture database used to compose the mosaic texture images in the experiments. The textures are sorted from high to low with respect to coarseness, contrast, line-likeness and directionality respectively.

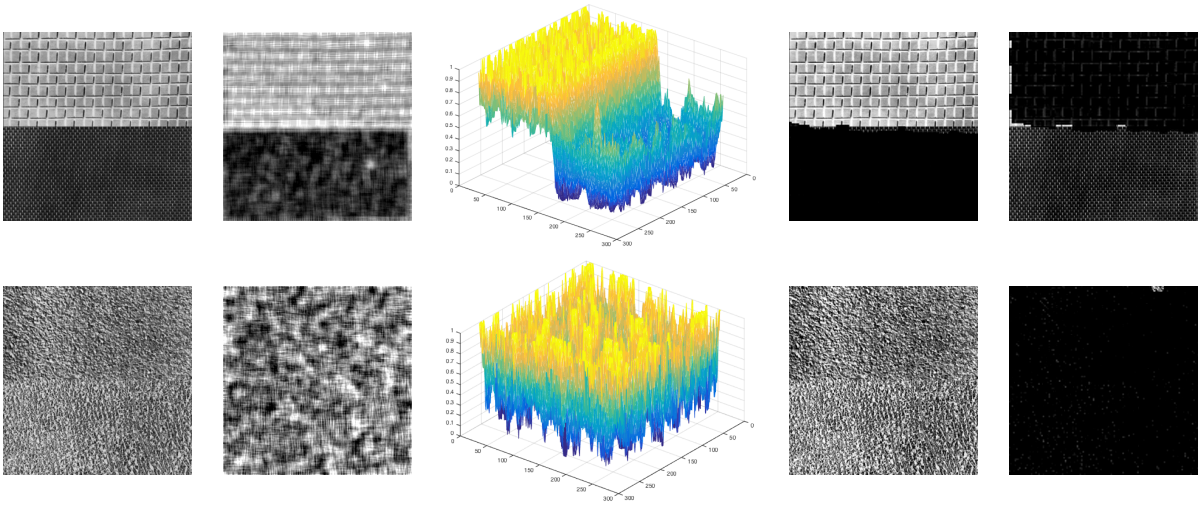


Figure 4.25: The decomposition of the mosaic images with respect to coarseness by the WT-TC-MCA method. Rows (from top to bottom): decomposition of the image where textures are very different in coarseness and decomposition of the image where textures are very similar in coarseness. Columns (from left to right): the input images, the feature maps of the coarseness of the images, the colour mesh plots of the corresponding the feature maps, the "strong-coarseness" component decomposed from the image by WT-TC-MCA and the "weak-coarseness" component decomposed from the image by WT-TC-MCA.

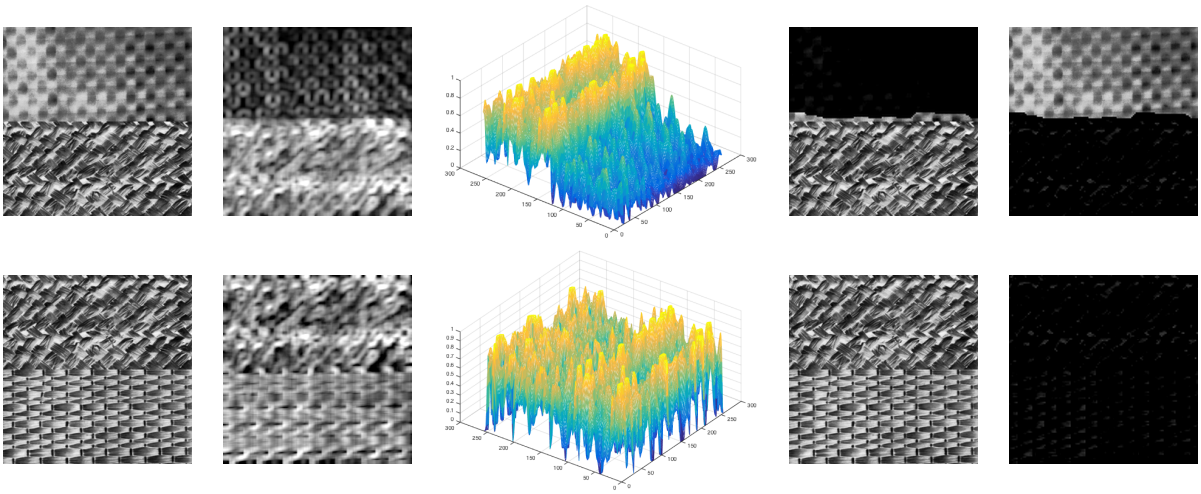


Figure 4.26: The decomposition of the mosaic images with respect to contrast by the WT-TC-MCA method. Rows (from top to bottom): decomposition of the image where textures are very different in contrast and decomposition of the image where textures are very similar in contrast. Columns (from left to right): the input images, the feature maps of the contrast of the images, the colour mesh plots of the corresponding the feature maps, the "strong-contrast" component decomposed from the image by WT-TC-MCA and the "weak-contrast" component decomposed from the image by WT-TC-MCA.

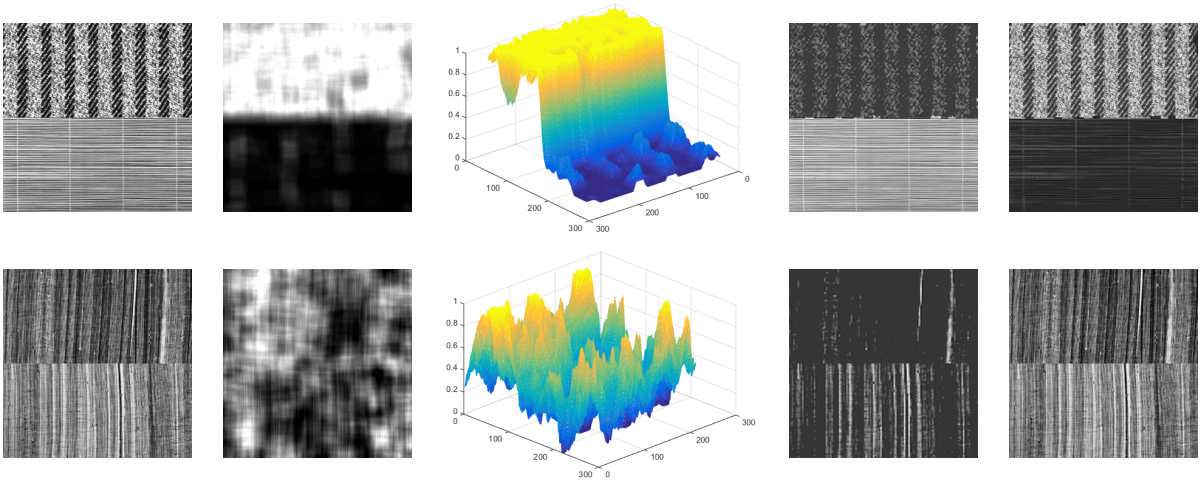


Figure 4.27: The decomposition of the mosaic images with respect to directionality by the WT-TC-MCA method. Rows (from top to bottom): decomposition of the image where textures are very different in directionality and decomposition of the image where textures are very similar in directionality. Columns (from left to right): the input images, the feature maps of the directionality of the images, the colour mesh plots of the corresponding the feature maps, the "strong-directionality" component decomposed from the image by WT-TC-MCA and the "weak-directionality" component decomposed from the image by WT-TC-MCA.

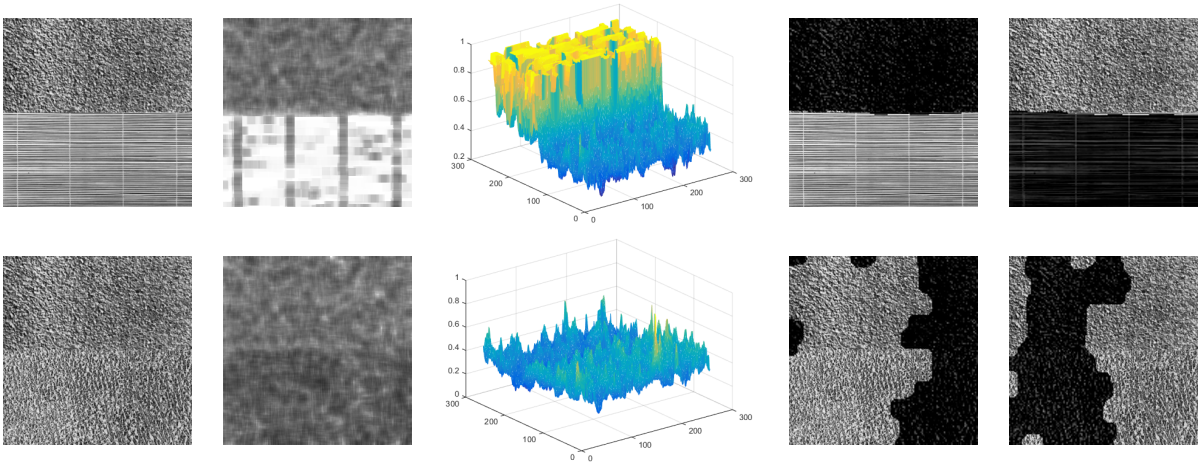


Figure 4.28: The decomposition of the mosaic images with respect to line-likeness by the WT-TC-MCA method. Rows (from top to bottom): decomposition of the image where textures are very different in line-likeness and decomposition of the image where textures are very similar in line-likeness. Columns (from left to right): the input images, the feature maps of the line-likeness of the images, the colour mesh plots of the corresponding the feature maps, the "strong-line-likeness" component decomposed from the image by WT-TC-MCA and the "weak-line-likeness" component decomposed from the image by WT-TC-MCA.

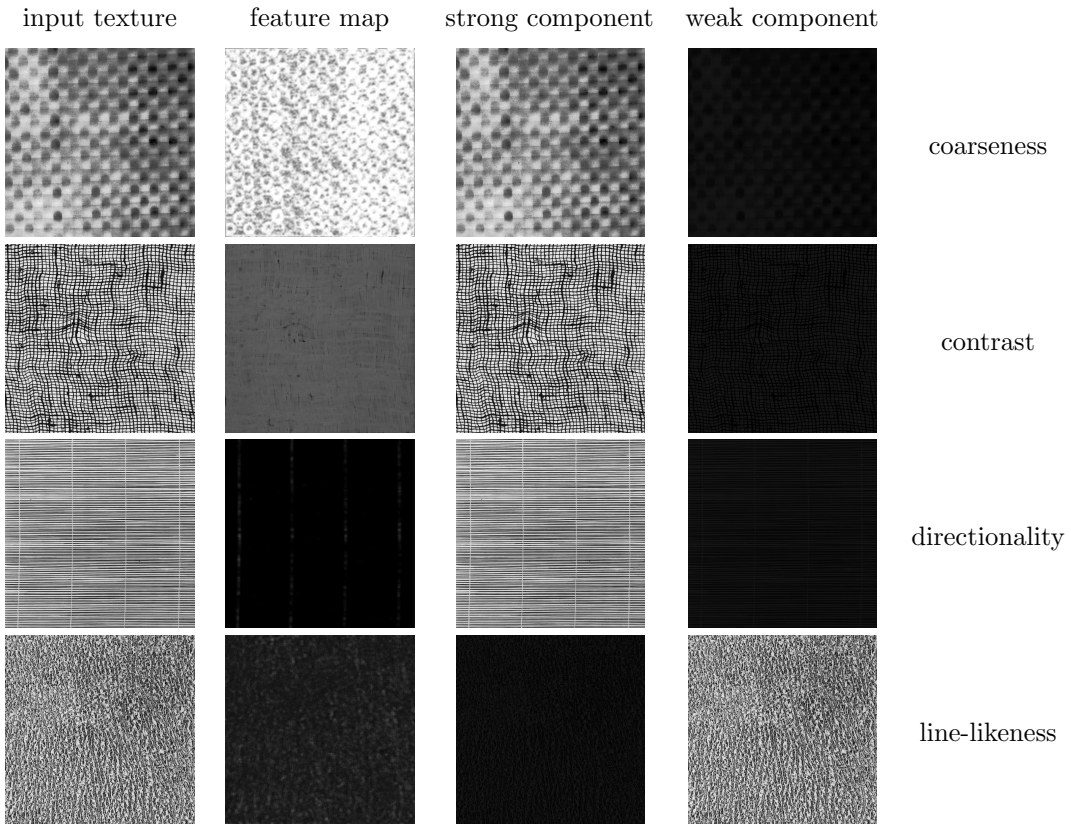


Figure 4.29: The decomposition of the pure texture exhibiting one consistent texture characteristic by the WT-TC-MCA. Rows (from top to bottom): pure textures exhibiting coarseness, contrast, directionality, line-likeness consistently. Columns (from left to right): input textures, feature maps of the corresponding texture characteristics, the decomposed "strong" components and the decomposed "weak" components.

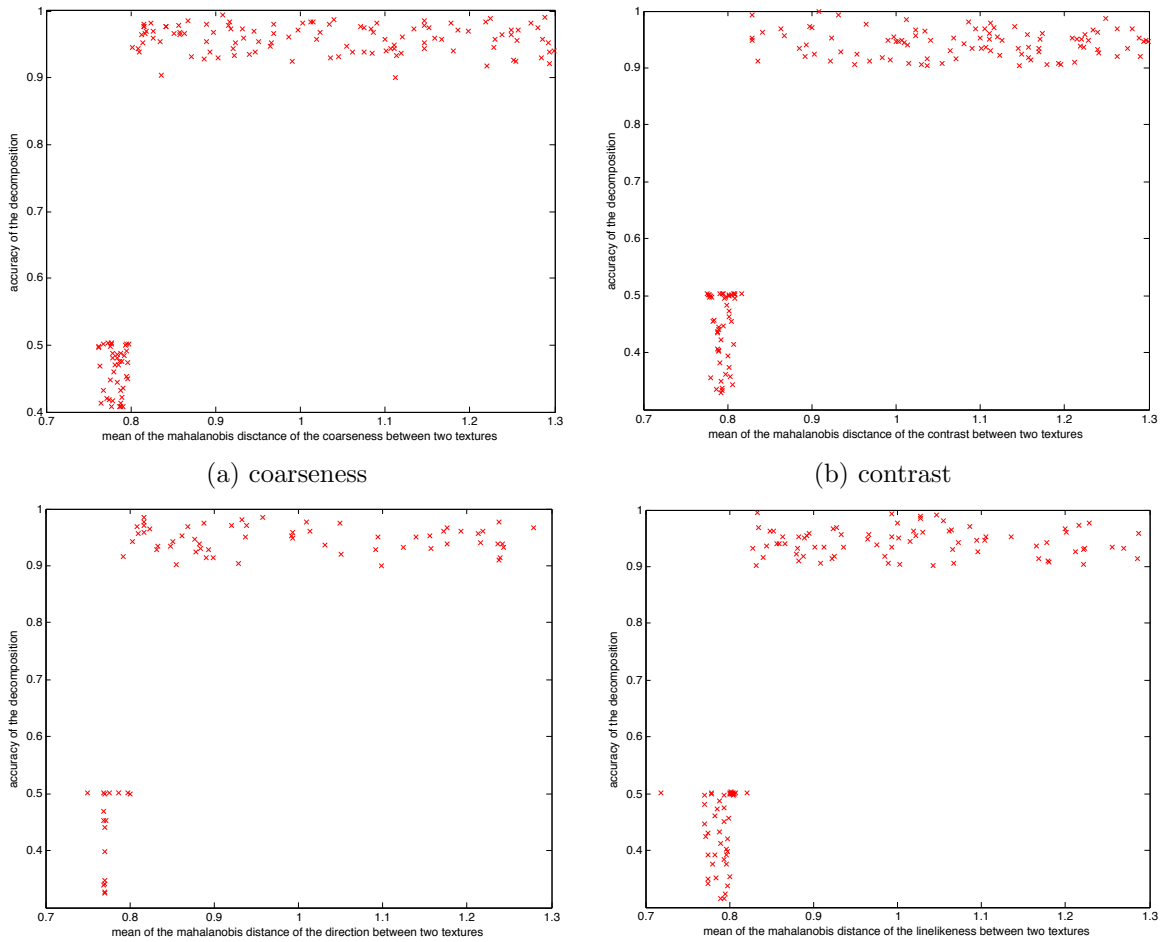
to the **coarseness**. And for the contrast, line-likeness and directionality, the thresholds are $[0.81, 0.83]$, $[0.80, 0.82]$ and $[0.81, 0.82]$ respectively. The "decomposable" threshold is quite small, which means most of the testing textural images can be decomposed by the proposed method. Table 4.5 shows the number of images which can be decomposed or not according to the 4 specific characteristics, as well as the average decomposition accuracies of the decomposable images and indecomposable images respectively. For coarseness, 400 images over the total 435 can be decomposed with an average accuracy of 98.85%, while the rest 35 images result in an average accuracy of 48.75%; for contrast, 395 images can be decomposed with an average accuracy of 98.43%, while the rest 40 images result in an average accuracy of 47.92%; for line-likeness, 389 images can be decomposed with an average accuracy of 97.95%, while the rest 46 images result in an average accuracy of 49.25%; and for directionality, 390 images can be decomposed with an average accuracy of 98.45%, while the rest 45 images result in an average accuracy of 47.12%. These results show that the proposed decomposition method can provide quite stable performance in decomposing the textural images according to different texture characteristics over most of the test mosaic images.

characteristic	number of successfully decomposed images (of the total 435 images)	average decomposition accuracy
coarseness	400	98.85%
contrast	395	98.43%
directionality	389	97.95%
line-likeness	390	98.45%
characteristic	number of unsuccessfully decomposed images (of the total 435 images)	average decomposition accuracy
coarseness	35	48.75%
contrast	40	47.92%
directionality	46	49.25%
line-likeness	45	47.12%

Table 4.5: The number of decomposable and indecomposable images of all the 435 testing images according to coarseness, contrast, directionality and line-likeness, as well as the corresponding average decomposition accuracies.

4.5.5 Discussion

To analyze the advantages of the WT-TC-MCA over the conventional image decomposition, we herein compare the feature maps of the components from the WT-TC-MCA and standard "cartoon and texture" morphological component analysis (MCA). In Fig. 4.31, an example mosaic texture image is decomposed by MCA and WT-TC-MCA (according to coarseness, contrast, directionality and line-likeness), then the texture characteristics of the separated components are measured and mapped. It shows that for each one of the



(a) coarseness

(b) contrast

(c) directionality

(d) line-likeness

Figure 4.30: The relationship between the decomposition accuracy and differences with respect to texture characteristics (coarseness, contrast, directionality and line-likeness from (a) to (d)). The decomposition accuracy is calculated by Eq. 4.39.

characteristics, the regions with strong aspect are concentrated in the strong component after WT-TC-MCA while those with weak aspect are mostly in the weak component. However, in the case of the MCA, the distribution of the features does not show any performance of clustering because the dictionaries MCA used cannot represent different texture characteristics.

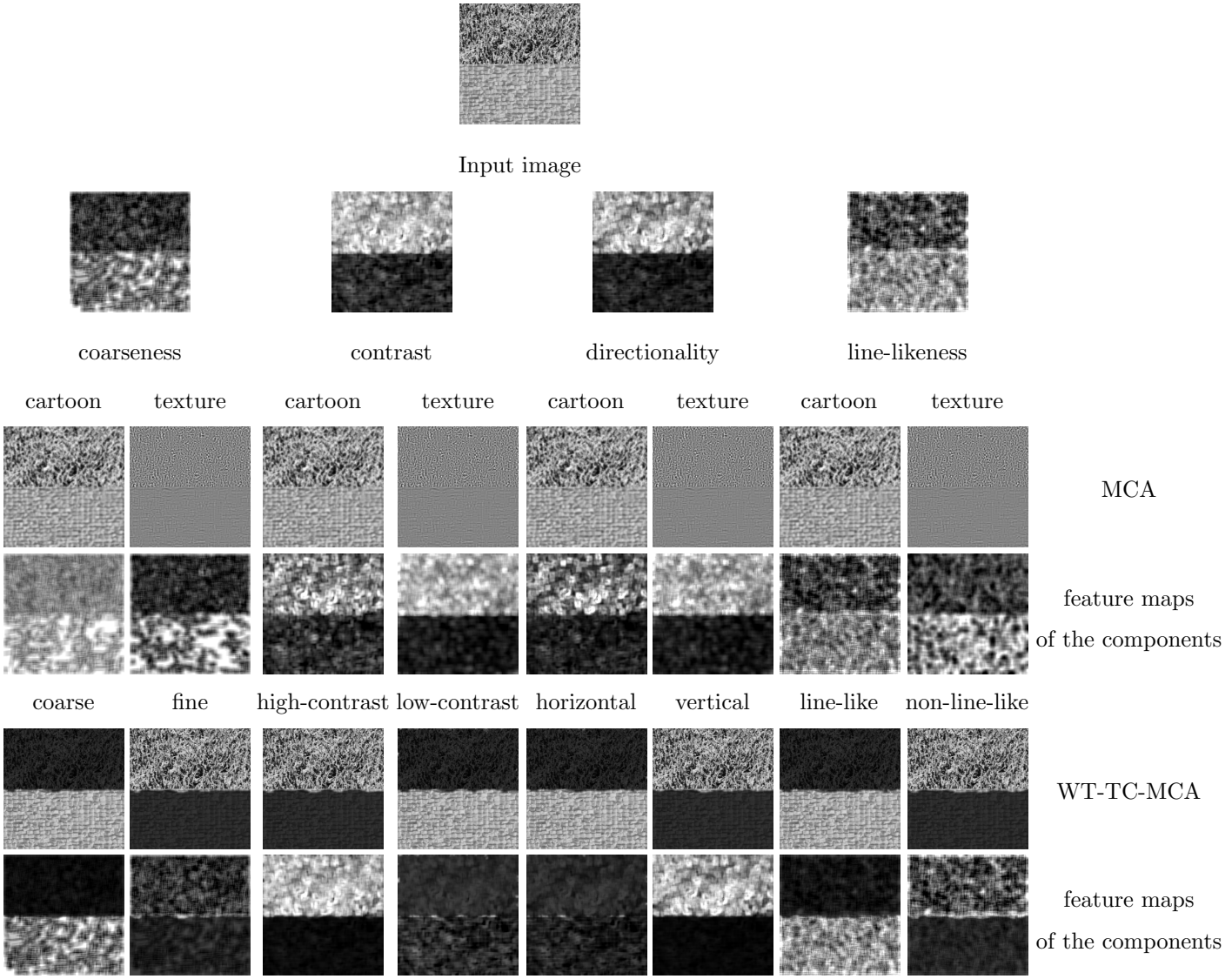


Figure 4.31: The feature maps of the decomposed components of the example mosaic texture image with two textures in it by different methods. Rows: decomposition results and the corresponding Tamura's feature maps by different methods: original image, traditional "cartoon+texture" MCA and WT-TC-MCA (from top to bottom). Columns: decomposition results and the feature maps with respect to different texture characteristics: coarseness, contrast, directionality and line-likeness (from left to right). The Tamura's features are locally calculated as described in Chapter 3.

Fig. 4.32 shows the comparison of feature separation on an example real-world image, which exhibits results similar to those for the mosaic image. Therefore, it demonstrates that the WT-TC-MCA, with the help of dictionaries representing texture characteristics, outperforms the conventional component analysis

methods in separating the different texture characteristics, leading to better performance in decomposing the image according to different texture characteristics.

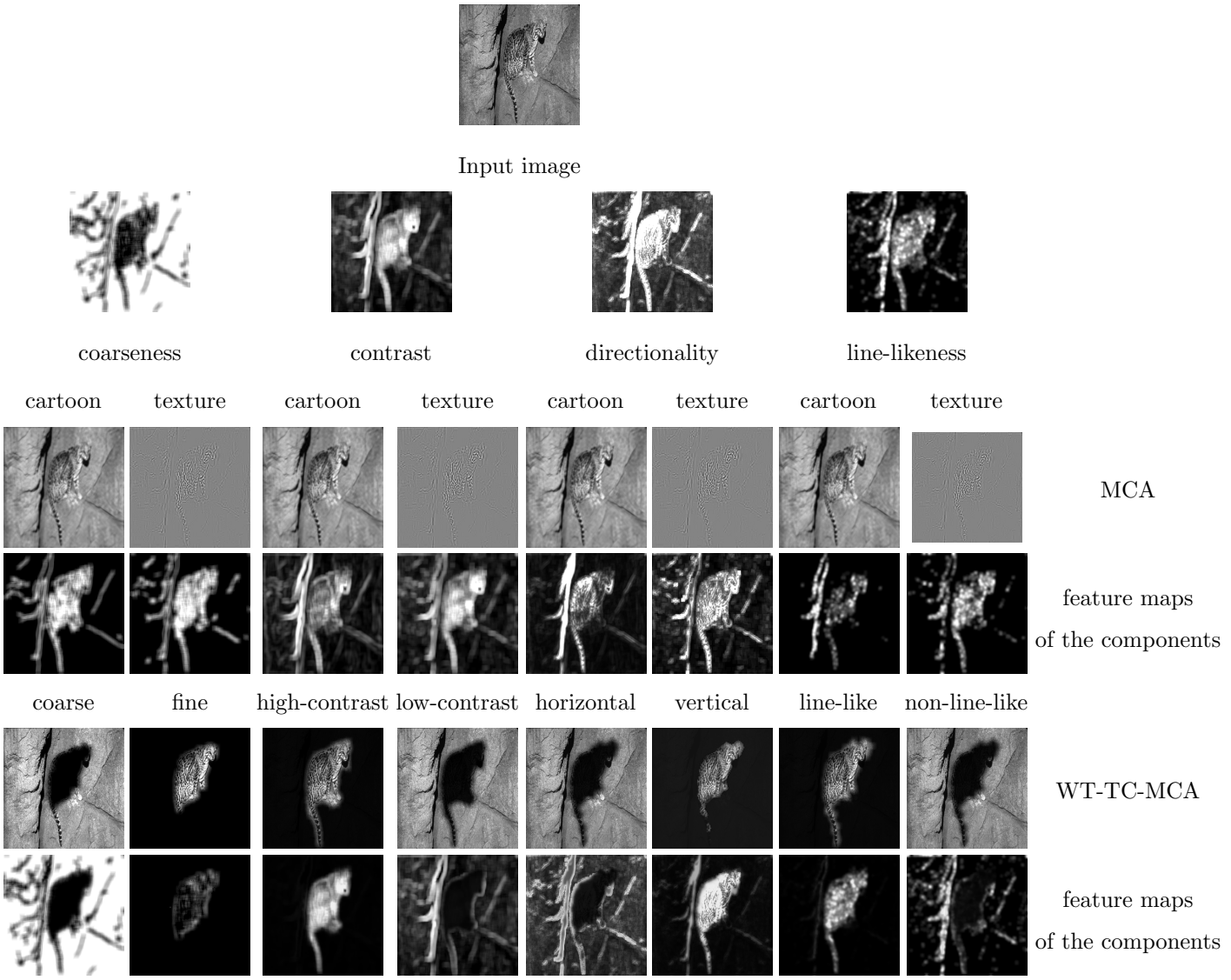


Figure 4.32: The feature maps of the decomposed components of the example real-world image with two types of contents in it by different methods. Rows: decomposition results and the corresponding Tamura's feature maps by different methods: original image, traditional "cartoon+texture" MCA and WT-TC-MCA (from top to bottom). Columns: decomposition results and the feature maps with respect to different texture characteristics: coarseness, contrast, directionality and line-likeness (from left to right). The Tamura's features are locally calculated as described in Chapter 3.

4.6 Conclusion

In this chapter, we proposed a texture characteristic based morphological component analysis using wavelet-based dictionaries. Image can be decomposed into pairwise components representing a given texture char-

acteristic strongly and weakly with the wavelet-based dictionaries that can sparsely highlight strong and weak texture characteristics, respectively. Experiments are implemented to compare the performance of WT-TC-MCA and other methods in decomposing both mosaic texture images and real-world images. The experimental results show that the WT-TC-MCA can successfully decompose the image into components strongly and weakly exhibiting the certain characteristics, even when the variation of a given characteristic over the image is relatively small. In the next chapter we will we will develop a texture enhancement method where the components resulting from the WT-TC-MCA decomposition can be be manipulated to enhance their own properties.

Chapter 5

Wavelet-based texture components enhancement

In the last chapter, the input image can be decomposed into pairwise components representing some certain texture characteristics strongly and weakly. To obtain the enhanced image with the texture characteristics more distinguishable, these components with the certain texture characteristics are enhanced to highlight their own properties and then recombined together. Therefore, in this chapter, we will discuss how to enhance the separated components with respect to the certain texture feature.

5.1 Introduction

Image enhancement methods can be used to enhance the components separated from the image. Based on the literature reviewed in Chapter 2.3, non-linear image manipulation methods are suitable for our purpose of enhancing the texture characteristics represented by different components. Moreover, the wavelet-based image enhancement methods have become more attractive recently. By manipulating the wavelet coefficients in different sub-bands with different functions, the wavelet-based enhancement methods can manipulate different types of image content in different ways at the same time, e.g., suppressing the noise and amplifying the edges, so that the image can be enhanced with more accentuated texture characteristics.

Most of the wavelet-based image enhancement methods follow the idea that a threshold T is determined, and then the coefficients larger and smaller than T are modified according to two different rules. The key points of this kind of method are to estimate the threshold T correctly and optimally design the rules for modifying coefficients. Based on the different selections of the threshold, the recent methods can be classified as hard thresholding and soft thresholding.

Hard thresholding functions, as first proposed in [66], retain all wavelet coefficients greater than the given threshold T and set the others to zero. The threshold T is chosen according to the signal energy and the noise variance σ^2. It was considered a simple but very useful method for processing images in the wavelet domain since it preserves the coefficients larger than T that are assumed to be attributed to the original signal and removes the coefficients smaller than T that are assumed to be due to noise. However, the method causes a lot of artifacts in the images when it fails to remove moderately strong coefficients caused by noise.

To overcome the disadvantages of hard thresholding, soft thresholding, also known as wavelet shrinkage, was proposed in [62]. In the wavelet shrinkage method, wavelet coefficients above the threshold are shrunk

by the absolute value of the threshold itself. Most of the wavelet shrinkage literature was based on methods for choosing the optimal threshold which can be adaptive or non-adaptive to the image. VISUShrink [64] is a non-adaptive universal threshold which depends only on the number of the data points. It has asymptotic equivalence and obtains best performance, in terms of mean square error (MSE), when the number of pixels reaches infinity. Another advantage of VISUShrink was that the threshold can be easily computed with knowledge of the noise variance and image size, so it can implement the de-noising efficiently. However, it generates overly-smoothed images because its threshold choice can be too large due to its dependence on the number of pixels in the image. To address the drawback of the VISUShrink, the SUREShrink was proposed as an adaptive thresholding method where the wavelet coefficients were treated in level-by-level fashion. In each level, when there is information that the wavelet representation of that level is not sparse, a threshold that minimizes Stein's unbiased risk estimate (SURE) is applied. The main advantage of SUREShrink is that it calculates a separate threshold value for each of the detail levels, which avoids over-smoothing to some extent. In [65], it is shown that SUREShrink performs better than VISUShrink in recovering the high frequency signal. However, SUREShrink performs poorly in situations of extreme sparsity of the wavelet coefficients, where the noise contributed to the SURE profile by the many coordinates where the signal is zero overwhelms the information contributed to the SURE profile by the few coordinates where the signal is nonzero. Moreover, the SUREShrink algorithm is not efficient because it calculates a threshold for every level, but only the thresholds from the first few decomposition levels are necessary for image processing. After the work of Donoho et al., BayesShrink [39] was proposed as another adaptive, data-driven thresholding method where the wavelet coefficients in a sub-band of the image can be represented effectively by a Generalized Gaussian Distribution (GCD) and the threshold is derived in a Bayesian framework. Since each high-frequency sub-band can have a different generalized Gaussian distribution, and therefore a different variance, a separate threshold is calculated for each detail sub-band using only the data from that sub-band and the noise variance, which is assumed to be constant across all sub-bands. However, for the noise whose variance is proportional to the measurements of image pixels, BayesShrink has the disadvantage that it is only effective for small-amplitude noise coefficients. Chen et al. [40] proposed the wavelet thresholding method by incorporating neighbouring coefficients, named NeighShrink. The method thresholded the wavelet coefficient according to the magnitude of the squared sum of all the wavelet coefficients, i.e., the local energy, within the neighbourhood window. In contrast to the conventional wavelet shrinkage method proposed by Donoho, the NeighShrink de-noising method avoids the main disadvantage of the former wavelet shrinkage method that too many wavelet coefficients that might contain useful image information (textures, edges or other details) are attenuated too greatly. Because the correlation among the wavelet coefficients in the neighbourhood is fully considered in NeighShrink, it can preserve the image details very well when removing noise. Sendur et al. [178] proposed BiShrink that determines the threshold by using a bivariate shrinkage function between a coefficient and its parent whose statistical dependency is modelled by a bivariate probability density function (PDF). It considers the relationship between the coefficients at different decomposition levels, not only those

at the same scale. However, the performance of BiShrink is highly dependent on the estimation of noise and signal variance of each wavelet coefficient, resulting in poor de-noising when the homogeneous regions are heavily corrupted by noise.

The methods introduced above have proven efficient for image de-noising. There are not yet, however, good methods for enhancing image details or contrast in the wavelet domain. It is difficult to differentiate the coefficients corresponding to details or high contrast areas from those corresponding to noise in wavelet domain so as to apply different modification to those coefficients. However, attempts have been proposed recently [95, 173, 206]. In [128], X. Liu et al. proposed a non-linear operator to modify the high-frequency sub-bands of wavelet decomposition. Low-frequency coefficients are modified for adjusting the grayscale range ofthe image, and high-frequency coefficients are modified to enhance local contrast and edges. Most of the popular non-linear filters can be used to modify these coefficients.

However, all of the wavelet-based image enhancement methods suffer from the same disadvantages as other non-linear enhancing methods, that is, few of them can enhance the texture characteristics other than contrast. Therefore, to enhance the components representing Tamura's texture characteristics, it is necessary to find wavelet-based enhancement methods for the coarseness, contrast, directionality and line-likeness components of an image, respectively, such that each method enhances exactly one characteristic while not influencing the others.

5.2 Concept of image enhancement based on wavelet transformation

Typically, the enhancement of a given image I by using wavelet transform is implemented in three steps. Firstly, the image is decomposed with wavelet transform into four sub-bands as follows:

$$\begin{aligned} W_\phi(j_0,m,n) &= \tfrac{1}{\sqrt{MN}}\textstyle\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)\phi_{j_0,m,n}(x,y) \\ W_\varphi^i(j_0,m,n) &= \tfrac{1}{\sqrt{MN}}\textstyle\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)\varphi_{j_0,m,n}^i(x,y) \end{aligned}, \tag{5.1}$$

where $f(x,y)$ represents the pixel in image I, $\phi(x,y)$ is the scaling function giving the low-frequency approximation of the image, $\varphi^i(x,y)$ are the wavelet functions measuring the variations along columns ($i=H$), rows ($i=V$) and diagonals ($i=D$), M and N are the rows and columns of the image respectively. Then the image is decomposed into $W_\phi(j_0,m,n)$ representing the low-frequency approximate image at the scale j_0 and $W_\varphi^i(j_0,m,n)$ representing the high-frequency approximate images of horizontal part ($i=H$), vertical part ($i=V$) and diagonal part ($i=D$) at the scale j_0.

Secondly, some adequate enhancement is made to the high-frequency sub-bands representing the edges or textures in the image:

$$\left[W_\varphi^i(j,m,n)\right]' = \begin{cases} \operatorname{sgn}\left(W_\varphi^i(j,m,n)\right)\cdot\left(K_1\left|W_\varphi^i(j,m,n)\right|\right) & \text{if } \left|W_\varphi^i(j,m,n)\right| \le T \\ \operatorname{sgn}\left(W_\varphi^i(j,m,n)\right)\cdot\left(K_2\left|W_\varphi^i(j,m,n)+K_1T\right|\right) & \text{if } \left|W_\varphi^i(j,m,n)\right| > T \end{cases}, \tag{5.2}$$

where K_1 and K_2 are the enhancing coefficients for the wavelet coefficient $W_{\varphi}^{i}(j,x,y)$, and T is the threshold. If $K_1 > K_2$, the coefficients with small magnitudes are amplified so that the weak edges of the image are enhanced. While if $K_1 < K_2$, the coefficients with large magnitudes are amplified more, thus the sharp edges are highlighted.

With the manipulated wavelet coefficients, the image is enhanced by the inverse-transform as the last step:

$$f^{'}(x,y) = \frac{1}{\sqrt{MN}} \sum_{m} \sum_{n} W_{\phi}(j_0,m,n)\phi_{j_0,m,n}(x,y) + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_{m} \sum_{n} \left[W_{\varphi}^{i}(j,m,n)\right]^{'} \varphi_{j_0,m,n}^{i}(x,y). \tag{5.3}$$

Other functions have also been proposed to manipulate the wavelet coefficients so that the image can be enhanced in local contrast and edges [110]. However, most of the recent enhancement methods only manipulate the magnitudes of the wavelet coefficients in the high-frequency sub-bands, so only the texture characteristics based on the magnitudes of the edges, such as local contrast, directionality, can be enhanced. In this thesis, we manipulate the magnitudes and the distribution of the wavelet coefficients in both low-frequency and high-frequency sub-bands, so that the image can be enhanced with respect to other texture characteristics.

5.3 Texture characteristic enhancement based on wavelet transformation

The proposed texture characteristics enhancement methods apply the same scheme of the wavelet-based enhancement methods as discussed in Chapter 5.2. The image components, which are separated from the image using the WT-TC-MCA method from Chapter 4, are firstly decomposed by a given wavelet transform:

$$\begin{aligned} \alpha_{s,i} &= [a_{s,i}, h_{s,i}, v_{s,i}, d_{s,i}] = \mathrm{dwt}\left(s_{s,i}, j\right) \\ \alpha_{w,i} &= [a_{w,i}, h_{w,i}, v_{w,i}, d_{w,i}] = \mathrm{dwt}\left(s_{w,i}, j\right) \end{aligned}, \tag{5.4}$$

where $s_{s,i}$ and $s_{w,i}$ are the image components representing the coarseness ($i = 1$), contrast ($i = 2$), directionality ($i = 3$) and line-likeness ($i = 4$) strongly and weakly, $\alpha_{s,i}$ and $\alpha_{w,i}$ are the wavelet coefficients of these image components via the wavelet transform $dwt(\cdot)$. Then the wavelet coefficients are manipulated by the functions designed for enhancing each texture characteristic respectively:

$$\begin{aligned} \alpha_{s,i}^{'} &= f_{s,i}\left(\alpha_{s,i}\right) \\ \alpha_{w,i}^{'} &= f_{w,i}\left(\alpha_{w,i}\right) \end{aligned}, \tag{5.5}$$

where $\alpha_{s,i}^{'}$ and $\alpha_{w,i}^{'}$ are the wavelet coefficients of the components manipulated by the functions $f_{s,i}(\cdot)$ and $f_{w,i}(\cdot)$ that accentuate the characteristics represented by the wavelet coefficients $\alpha_{s,i}$ and $\alpha_{w,i}$ of the image components. The enhanced components $s_{s,i}^{'}$ and $s_{w,i}^{'}$ are then computed from the inverse wavelet transform

of the certain manipulated coefficients:

$$\begin{matrix} s'_{s,i} = \text{idwt}(\alpha'_{s,i}) \\ s'_{w,i} = \text{idwt}(\alpha'_{w,i}) \end{matrix}. \tag{5.6}$$

The proposed wavelet-based image enhancement method differs from the conventional ones in the functions for manipulating the wavelet coefficients. We don't simply apply the thresholding method on coefficients in high-frequency sub-bands, but manipulate the coefficients in all sub-bands by the spatial-like non-linear filters to accentuate the relations between wavelet coefficients in the neighbourhood. The filters or functions are different due to different texture characteristics to be highlighted, so we will first discuss the details of designing the enhancement functions in the following part.

5.3.1 Discipline of texture characteristics enhancement

As discussed in Chapter 4, different texture characteristics can be represented by wavelet coefficients in different sub-bands. Given the texture characteristics: coarseness, contrast, directionality and line-likeness, the wavelet coefficients are used as follows to reflect them:

1. the high-frequency sub-bands of DWT contains the information of the distribution of edges of the image so as to reflect the coarseness of the image;

2. the low-frequency sub-band of DWT contains the information of the distribution of pixel intensities of the image and reflects the contrast of the image;

3. the high-frequency sub-bands of DWT preserves the directions and positions of the edges in the image, and the mean value of the distribution of the high frequency coefficients reflects the directionality;

4. the variance of the distribution of the coefficients in high frequency sub-bands with different directions reflects the line-likeness of the image.

Given how the wavelet coefficients reflect the texture characteristics, it can be summarized how to highlight the texture characteristics by modifying the corresponding wavelet coefficients. However, because the coefficients may relate to more than one texture characteristic, it should be also required that the modification of the coefficients will not affect the texture characteristics other than that which we are attempting to enhance. Therefore, the requirements of the enhancement scheme are shown in Table 5.1.

According to the "rules" for enhancement of the wavelet coefficients, different functions $f_{s,i}$ and $f_{w,i}$ in Eq. 5.5 for manipulating the wavelet coefficients for specific texture characteristics are described in the following parts.

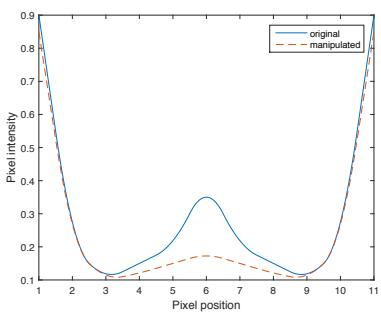
characteristics	operations need to do	operation need to avoid
coarseness	modification of the number of edges in the neighbourhood	1. modification of the gray levels in the neighbourhood; 2. modification of the mean and the variance of the edge directions in the neighbourhood.
contrast	modification of the variation of the gray levels in the neighbourhood	1. modification of the edge occurrences in the neighbourhood; 2. modification of the edge directions in the neighbourhood.
directionality	modification of the number of edge pixels with the certain direction	1. modification of the distribution of the edge directions in the neighbourhood; 2. modification of the distribution of the gray levels in the neighbourhood.
line-likeness	modification of the variance and the kurtosis of the edge directions	1. modification of the mean value of the directions in the neighbourhood; 2. modification of the number of edges in the neighbourhood.

Table 5.1: The requirements of the scheme of the methods in enhancing texture characteristics reflected by wavelet coefficients.

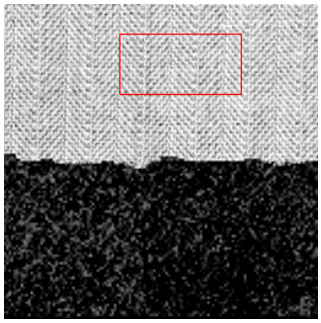
5.3.2 Manipulations for the coarseness-decomposed components

Coarse (strong coarseness) component enhancement

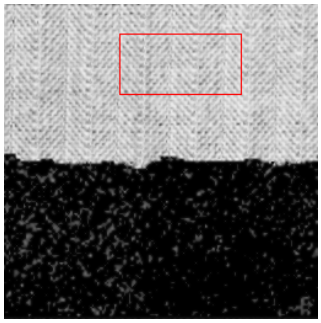
To enhance the coarse component's property, that is, make the component coarser, can be considered intuitively as the process of removing the small-size, granular texels from the component. It requires removing the "weak" edges where the gradients are relatively small, and enhance the "strong" edges showing the main structures image. Fig. 5.1(a) shows the enhancement method by a one-dimensional diagram. There are two "texels" in the diagram: one is from 1 to 6 and the other is from 6 to 11. By removing the "weak" edges at position 6 while preserving other pixels as much as possible, the two small texels with the width of 5 merge to one larger texel with the width of 10, leading to the coarseness magnified twice than that before enhancement.



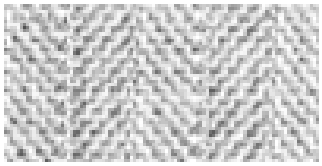
(a) Simulation of manipulating a coarse texture, shown as a one-dimensional curve



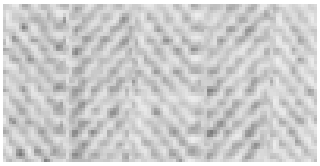
(b) Original coarse component



(c) Manipulated coarse component



(d) Details of the image inset in (b)



(e) Details of the image inset in (c)

Figure 5.1: The diagram and example of the coarse component enhancement. (a) The diagram of making the given one-dimension curve coarser; (b) the coarse component separated from the example image; (c) the manipulated coarse component where the textures are coarser; (d) details of the image inset in (b), marked by the red rectangle; (e) details of the image inset in (c), marked by the red rectangle.

In the wavelet domain, the edges are represented by the coefficients with high magnitudes in high-frequency sub-bands, and the gradients can be calculated by the variances of the magnitudes of the coefficients in approximate sub-band. Therefore, the following function is proposed to enhance the coarse component $s_{s,1}$:

$$\alpha'_{s,1} = \left(g\left(|\alpha_{s,1}|\right) + k \cdot d\right) \cdot \operatorname{sign}\left(\alpha_{s,1}\right) \tag{5.7}$$

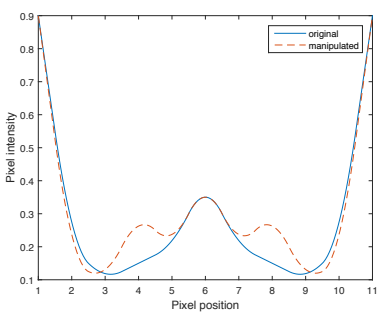
$$d = |\alpha_{s,1}| - g\left(|\alpha_{s,1}|\right)$$

$$k = \frac{e^{d_{max}-d} - 1}{e^{d_{max}-d_{min}} - 1}$$

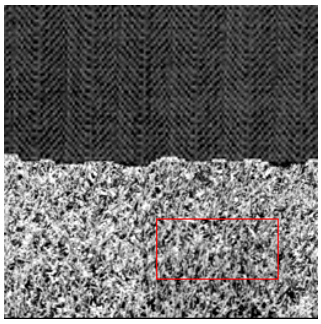
where $|\cdot|$ denotes the magnitude of the coefficients, $g(\cdot)$ is a 7×7 Gaussian filter with $\sigma = 1$, k is the weight of the difference d between the magnitude of wavelet coefficient and that of the Gaussian filtered coefficient. The strong edges which have a larger d will be enhanced while the weak edges or smooth regions which have smaller d will be further suppressed, enhancing the texture coarseness, as shown in Fig. 5.1(b) and Fig. 5.1(c).

Fine (weak coarseness) component enhancement

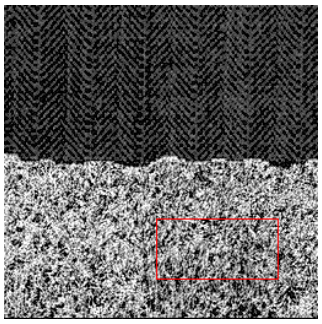
The enhancement of the fine component, which is to make the component finer, follows the opposite idea of enhancing the coarse component, that is, increasing the number of texels in the local neighbourhood. It is the process of adding granules in the texture regions, or cutting the complete texels into small pieces, which requires adding pixels with singular intensities to the neighbouring pixels so that the size of texels will be smaller. Fig. 5.2(a) is the diagram of enhancing "fineness" in one dimension. Originally there are two texels which are represented by the points from 1 to 6 and those from 6 to 11 respectively. By adding or changing the pixel intensities at position 4 and 8 to intensities more different than their surroundings, the two small texels with the width of 5 are split to four smaller texels with width 2 or 3, resulting in a finer texture.



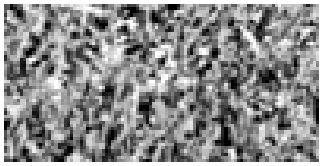
(a) Simulation of manipulating a fine texture, shown as a one-dimensional curve



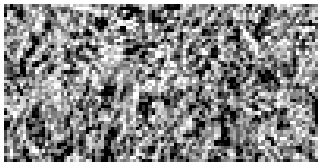
(b) Original fine component



(c) Manipulated fine component



(d) Details of the image inset in (b)



(e) Details of the image inset in (c)

Figure 5.2: The diagram and example of the fine component enhancement. (a) The diagram of making the given one-dimension curve finer; (b) the fine component separated from the example image; (c) the manipulated fine component where the textures are finer; (d) details of the image inset in (b), marked by the red rectangle; (e) details of the image inset in (c), marked by the red rectangle.

In the wavelet domain, we increase the texture primitives in the local neighbourhood by adding multi-

plicative noise dependent on the magnitudes of wavelet coefficients:

$$\alpha'_{w,1} = \left(g\left(|\alpha_{w,1}|\right) + k \cdot n \cdot |\alpha_{w,1}|\right) \cdot \text{sign}\left(\alpha_{w,1}\right) \tag{5.8}$$
$$k = \frac{e^d - 1}{e - 1}$$

where $|\cdot|$ denotes the magnitudes of the coefficients, n is the uniformly distributed random noise, k is the weight of the noise, calculated based on d, the local standard deviation of the magnitudes of wavelet coefficients. By adding these random noise to the fine regions where the magnitudes of wavelet coefficients change rapidly, the textures in Fig. 5.2(b) will be "shattered" locally, leading to an increasing of fineness shown in Fig. 5.2(c).

5.3.3 Manipulation for the contrast-decomposed components

High-contrast component enhancement

Enhancement of the high-contrast component is achieved by making the contrast of the local neighbourhood higher. The contrast measures the change of the pixel intensities in the neighbourhood, so enhancement of the contrast means increasing the the number of gray levels and the difference between the maximum and minimum intensity in the neighbourhood, but still preserving the positions of the edges . A diagram is shown as Fig. 5.3(a) demonstrating the method of contrast enhancement. By changing the pixel intensities of the pixels so that they have higher gradients to the neighbours, the smooth regions with gradual intensity changes are modified to the regions with rapid intensity changes.

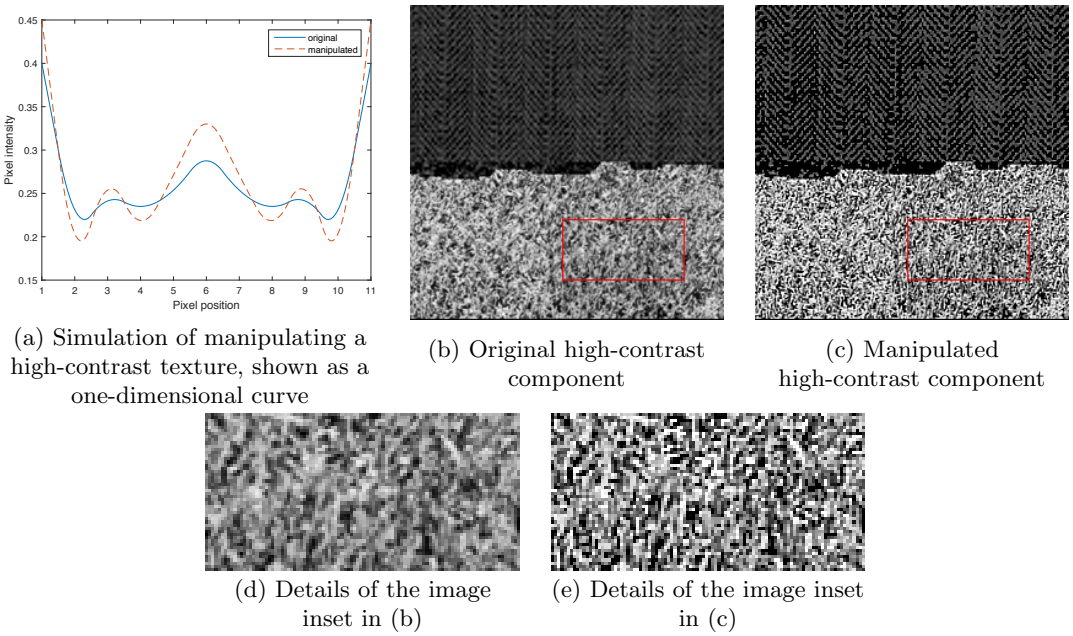
In the wavelet domain, the magnitudes of the coefficients in the approximate sub-band represent the intensity distribution of the image, and the magnitudes of the coefficients in the high frequency sub-bands represent the gradients of pixels in the image, therefore we use a weighted Laplacian filtering to increase the local contrast:

$$\alpha'_{s,2} = \left(|\alpha_{s,2}| + k \cdot \nabla^2\left(|\alpha_{s,2}|\right)\right) \cdot \text{sign}\left(\alpha_{s,2}\right) \tag{5.9}$$
$$k = e^{\sqrt{|h_{s,2}|^2 + |v_{s,2}|^2}}$$

where $|\cdot|$ denotes the magnitudes of the coefficients, $|\nabla^2(\cdot)|$ is a 5×5 Laplacian filter, and k is the weight based on the magnitude distribution of wavelet coefficients in horizontal and vertical sub-bands. The Laplacian of wavelet coefficients has high magnitudes in the high-contrast regions. As shown in Fig. 5.3(c), the magnitudes of the Laplacian are adjusted by k, which is positively related to the magnitudes of high-frequency wavelet coefficients, and added to the input wavelet coefficients, resulting in higher contrast of the original regions.

Low-contrast component enhancement

Contrary to the enhancement of the high-contrast component, the enhancement of the low-contrast component needs to compress the range the range of gray levels in the local neighbourhood. The common way to



(a) Simulation of manipulating a high-contrast texture, shown as a one-dimensional curve

(b) Original high-contrast component

(c) Manipulated high-contrast component

(d) Details of the image inset in (b)

(e) Details of the image inset in (c)

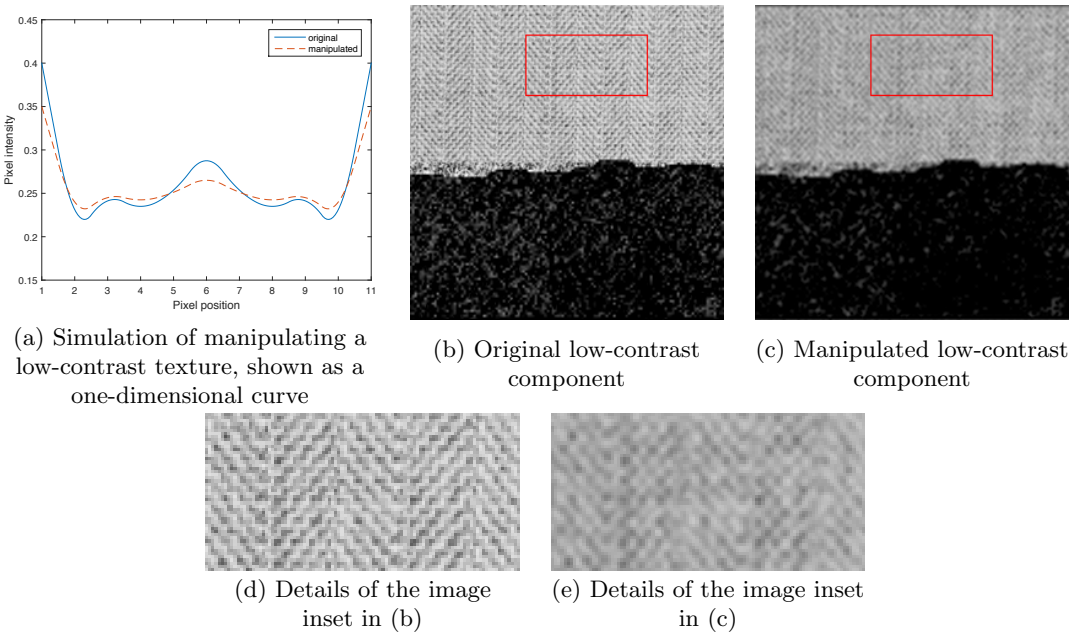
Figure 5.3: The diagram and example of the high-contrast component enhancement. (a) The diagram of making the given one-dimension curve higher contrast; (b) the high-contrast component separated from the example image; (c) the manipulated high-contrast component where the textures are with higher contrast; (d) details of the image inset in (b), marked by the red rectangle; (e) details of the image inset in (c), marked by the red rectangle.

achieve this target is to lower the differences between the centre pixels and the neighbours according to the intensity variations themselves, therefore the histogram of the gray levels will be compressed to the mean value of the neighbourhood and the texture structures are preserved. Fig. 5.4(a) shows how the low-contrast enhancement is implemented. By modifying the intensities at position 1,3,5,7,9, the histogram of the gray levels is changes from a distribution with wide range (from 0.3 to 0.4) to a concentrated distribution (most are around 0.25).

While in the wavelet domain, a weighted Gaussian filtering where the weights are calculated from the coefficients in the horizontal and vertical sub-bands is applied to the coefficients in approximate sub-band:

$$\begin{aligned} \alpha_{w,2}^{'} &= \left(g\left(\left|\alpha_{w,2}\right|\right) / \left(1+k\right)\right) \cdot \operatorname{sign}\left(\alpha_{w,2}\right) \\ k &= \sqrt{\left|h_{w,2}\right|^2 + \left|v_{w,2}\right|^2} \end{aligned} \tag{5.10}$$

where $g\left(\cdot\right)$ is a 7×7 Gaussian filtering with $\sigma = 1$, k is the weighting coefficients calculated from the distribution of magnitudes of wavelet coefficients in the horizontal and vertical wavelet sub-bands. The Gaussian filtering compresses the magnitudes of the wavelet coefficients, and the weighting coefficient k further increases compression on the smooth region where the $h_{w,2}$ and $v_{w,2}$ are low, as shown in Fig. 5.4(b) and Fig. 5.4(c).



(a) Simulation of manipulating a low-contrast texture, shown as a one-dimensional curve

(b) Original low-contrast component

(c) Manipulated low-contrast component

(d) Details of the image inset in (b)

(e) Details of the image inset in (c)

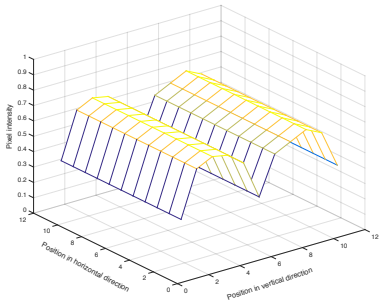
Figure 5.4: The diagram and example of the low-contrast component enhancement. (a) The diagram of making the given one-dimension curve lower contrast; (b) the low-contrast component separated from the example image; (c) the manipulated low-contrast component where the textures are with lower contrast; (d) details of the image inset in (b), marked by the red rectangle; (e) details of the image inset in (c), marked by the red rectangle.

5.3.4 Manipulation for the directionality-decomposed components

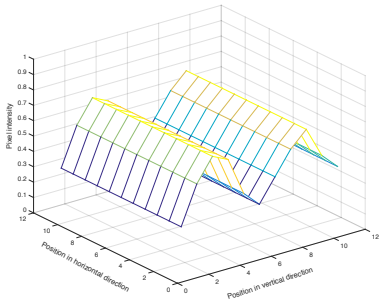
Horizontal (strong directionality) component enhancement

The enhancement of horizontal component requires that we make the component more "horizontal-like". The directions of edges in the horizontal components are close to horizontal, so the enhancement means making these edges more obvious than before. It requires increasing the pixel intensities along the horizontal direction and decreasing the pixel intensities along the other directions. Fig. 5.5(a) and Fig. 5.5(b) show diagrams of the horizontal-enhancement method. By increasing the pixel intensities with $x = 3$ and $x = 9$ and decreasing the pixel intensities with $x = 2$, $x = 4$, $x = 8$ and $x = 10$, this horizontal edge becomes more obvious compared with others in the image, leading to the image exhibits more horizontal.

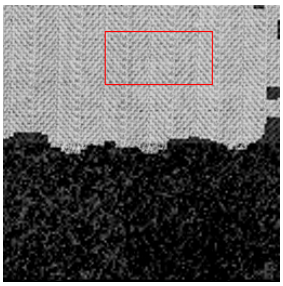
The position of horizontal edges are mostly represented by the wavelet coefficients in the horizontal sub-band and the magnitudes of the edges are reflected by the coefficients in approximate sub-band. To enhance the horizontal components, the wavelet coefficients in the approximate and horizontal sub-bands are manipulated to highlight those representing horizontal edges, while the wavelet coefficients in the vertical and diagonal sub-bands are removed so that the horizontal coefficients are more dominant in the output image.



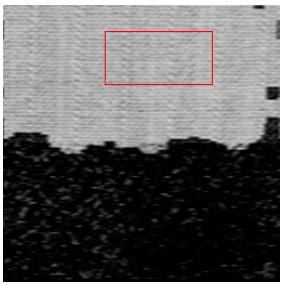
(a) Simulation of a horizontal texture



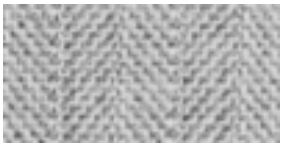
(b) Simulation of the manipulated horizontal texture



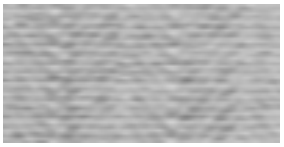
(c) Original horizontal component



(d) Manipulated horizontal component



(e) Details of the image inset in (c)



(f) Details of the image inset in (d)

Figure 5.5: The diagram and example of the horizontal component enhancement. (a) The diagram of example texture with horizontal edges; (b) the diagram of accentuating the horizontal edges in the example texture; (c) the horizontal component separated from the example image; (d) the manipulated horizontal component where the horizontal edges are more obvious; (e) details of the image inset in (c), marked by the red rectangle; (f) details of the image inset in (d), marked by the red rectangle.

The coefficients of horizontal component $\alpha_{s,3}$ is made more horizontal by:

$$\begin{aligned} a_{s,3}^{'} &= f_h\left(|a_{s,3}|\right) \cdot \operatorname{sign}\left(a_{s,3}\right) \\ h_{s,3}^{'} &= f_h\left(|h_{s,3}|\right) \cdot \operatorname{sign}\left(h_{s,3}\right) \\ v_{s,3}^{'} &= 0 \\ d_{s,3}^{'} &= 0 \end{aligned} \tag{5.11}$$

where $f_h(\cdot)$ is an averaging filter with the length of 5 and diffusing in horizontal direction: $f_h = \left[\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right]$. The wavelet coefficients in approximate and horizontal sub-bands are "moving" horizontally and the coefficients with other directions are removed, therefore the image in Fig. 5.5(c) is made more horizontal in Fig. 5.5(d).

Vertical (weak directionality) component enhancement

Since the vertical component can be considered as the rotation of the horizontal component, the enhancement of the vertical component can be implemented using the same method as that for horizontal component enhancement. By magnifying the gradients along the vertical edges and degrading the gradients along the other directions, the component is made more vertical. Fig. 5.6(a) to Fig. 5.6(b) show the process of the enhancement: increasing the intensities of pixels with $y = 6$ and $y = 9$, and decreasing the intensities of pixels with $y = 5$ and $y = 7$, the vertical edges are enhanced.

In the wavelet domain, the vertical component $\alpha_{w,3}$ is manipulated by manipulating the coefficients in approximate sub-band (representing the magnitudes of the edges) and vertical sub-band (representing the positions of the edges):

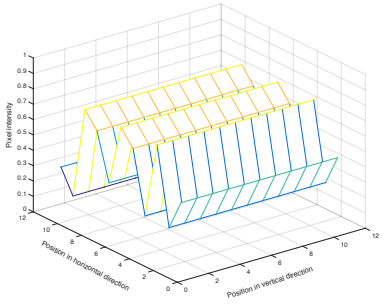
$$\begin{aligned} a_{w,3}^{'} &= f_v\left(|a_{w,3}|\right) \cdot \operatorname{sign}\left(a_{w,3}\right) \\ h_{w,3}^{'} &= 0 \\ v_{w,3}^{'} &= f_v\left(|v_{w,3}|\right) \cdot \operatorname{sign}\left(v_{w,3}\right) \\ d_{w,3}^{'} &= 0 \end{aligned} \tag{5.12}$$

where $f_v(\cdot)$ is the averaging filter with the length of 5 and diffusing in vertical direction: $f_v = \left[\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right]^T$. Similarly, in Fig. 5.6(d), the vertical components are made more vertical from the original image in Fig. 5.6(c) by "moving" approximate and vertical coefficients vertically and removing coefficients with other directions.

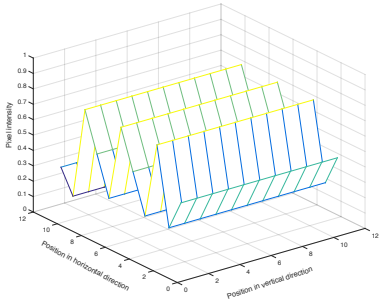
5.3.5 Manipulation for the line-likeness-decomposed components

Line-like (strong line-likeness) component enhancement

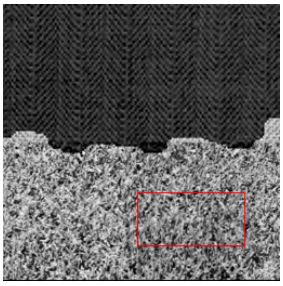
As discussed in Chapter 3, the line-likeness measures the similarity of the directions of the edges in the neighbourhood. Therefore, enhancing the line-like component means making the edges in the neighbourhood exhibit similar direction as much as possible. The basic idea of line-like enhancement is to remove the edges



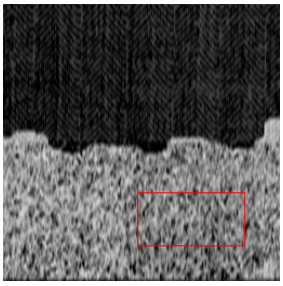
(a) Simulation of a vertical texture



(b) Simulation of the manipulated vertical texture



(c) Original vertical component



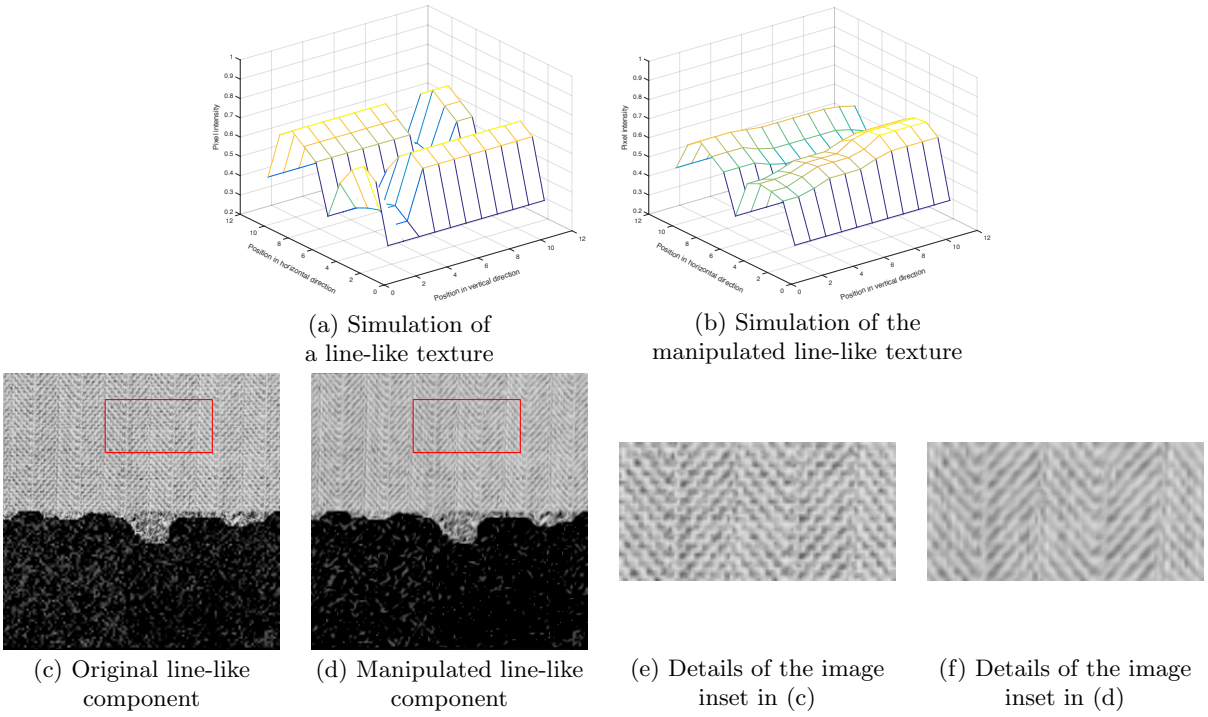
(d) Manipulated vertical component

(e) Details of the image inset in (c)

(f) Details of the image inset in (d)

Figure 5.6: The diagram and example of the vertical component enhancement. (a) The diagram of example texture with vertical edges; (b) the diagram of accentuating the vertical edges in the example texture; (c) the vertical component separated from the example image; (d) the manipulated vertical component where the vertical edges are more obvious; (e) details of the image inset in (c), marked by the red rectangle; (f) details of the image inset in (d), marked by the red rectangle.

with directions different from the major direction in the neighbourhood. Fig. 5.7(a) and Fig. 5.7(b) show the process of modification of edges. By increasing the edges along position $x = 2, x = 3$ and $x = 8, x = 9$, most of the edges with gradient magnitudes larger than threshold 0.5 are parallel to x-axis, therefore the distribution of edge directions can be more concentrated and the edges would be more line-like.



(a) Simulation of a line-like texture

(b) Simulation of the manipulated line-like texture

(c) Original line-like component

(d) Manipulated line-like component

(e) Details of the image inset in (c)

(f) Details of the image inset in (d)

Figure 5.7: The diagram and example of the line-like component enhancement. (a) The diagram of example texture where the edges exhibit different directions ; (b) the diagram of making the directions of the edges similar in the example texture; (c) the line-like component separated from the example image; (d) the manipulated line-like component where most of the edges are close to horizontal; (e) details of the image inset in (c), marked by the red rectangle; (f) details of the image inset in (d), marked by the red rectangle.

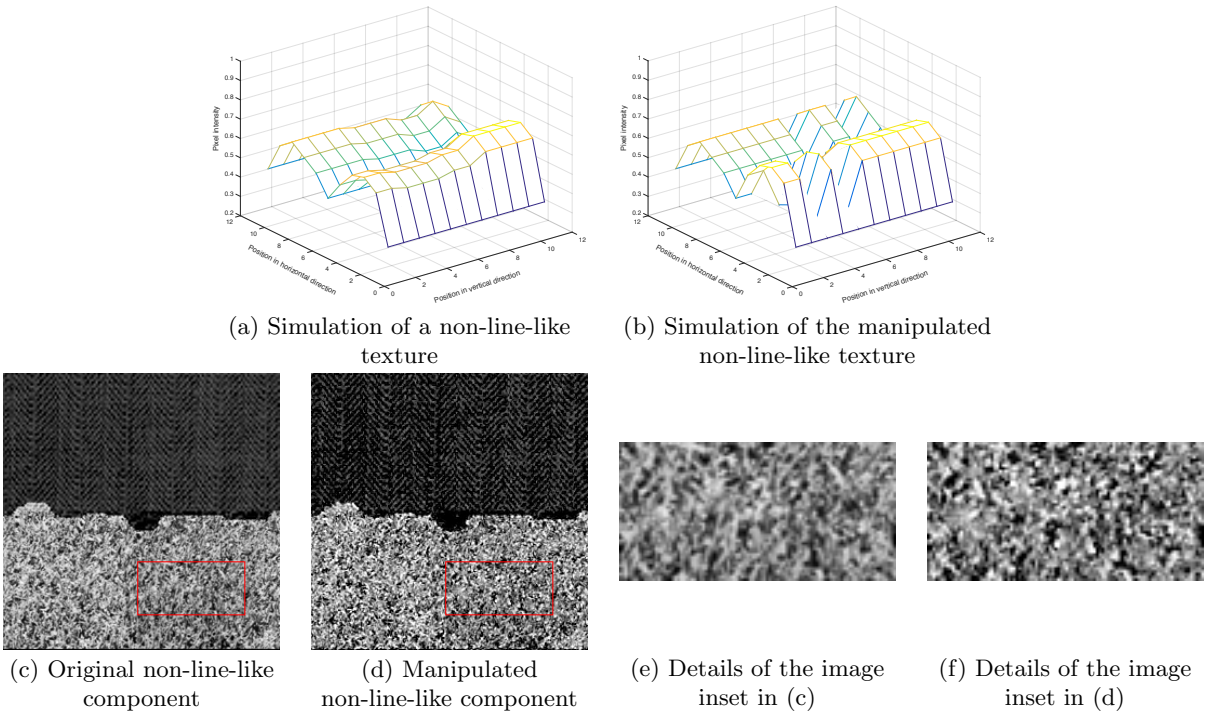
Inspired by the enhancement of components for directionality, we use a local motion filter to enhance the wavelet coefficients in every sub-band:

$$\alpha_{s,4}^{'} = \left(\left|\alpha_{s,4}\right| \otimes m_L\left(\theta, w\right)\right) \cdot \operatorname{sign}\left(\alpha_{s,4}\right), \tag{5.13}$$

where $m_L\left(\theta, w\right)$ is the local averaging filter adaptive to the neighbourhood L, with the length of w that is the size of the neighbourhood, and diffusing in the direction θ that is the edge direction (quantized into 4 directions 0, $\frac{\pi}{2}$, $\frac{\pi}{4}$, $\frac{3\pi}{4}$) occurring the most frequently in L. This filter "pulls" the coefficients into the same direction locally and the line-likeness will be enhanced since the line-likeness is defined as the frequency of the occurrence of the same direction in the local neighbourhood, as shown in Fig. 5.7(d).

Non-line-like component enhancement

The non-line-like component exhibits the opposite feature to that of the line-like component: the edges in the neighbourhood have very different directions to each other. So the more non-line-like the component is, the directions of the edges should vary more in the local window. In the extreme case, we add "branches" to the original edges along the perpendicular directions to the original ones. Fig. 5.8 and Fig. 5.8(b) show the above assumption. By increasing the gradients at the pixels with $x = 3$, $x = 5$, $x = 7$ and $x = 10$, which is perpendicular to the edges at $y = 3$ and $y = 9$, the distribution of the directionality is changed from the concentration at horizontal to equally distribution to horizontal and vertical.



(a) Simulation of a non-line-like texture

(b) Simulation of the manipulated non-line-like texture

(c) Original non-line-like component

(d) Manipulated non-line-like component

(e) Details of the image inset in (c)

(f) Details of the image inset in (d)

Figure 5.8: The diagram and example of the non-line-like component enhancement. (a) The diagram of example texture where the edges exhibit similar directions ; (b) the diagram of making the horizontal and vertical edges equally distributed in the example texture; (c) the non-line-like component separated from the example image; (d) the manipulated non-line-like component where the directions of the edges are more different; (e) details of the image inset in (c), marked by the red rectangle; (f) details of the image inset in (d), marked by the red rectangle.

In wavelet domain, we propose to enhance it by adding multiplicative noise to the coefficients representing

edges in the images:

$$
\begin{aligned}
a^{'}_{w,4} &= (|a_{w,4}| + k_a \cdot n \cdot |a_{w,4}|) \cdot \text{sign}\,(a_{w,4}) \\
h^{'}_{w,4} &= (|h_{w,4}| + k_h \cdot n \cdot |h_{w,4}|) \cdot \text{sign}\,(h_{w,4}) \\
v^{'}_{w,4} &= (|v_{w,4}| + k_v \cdot n \cdot |v_{w,4}|) \cdot \text{sign}\,(v_{w,4}) \\
d^{'}_{w,4} &= (|d_{w,4}| + k_d \cdot n \cdot |d_{w,4}|) \cdot \text{sign}\,(d_{w,4}) \\
k &= \frac{e^{d_{max}-d} - 1}{e^{d_{max}-d_{min}} - 1}
\end{aligned}
\tag{5.14}
$$

where n is a uniformly distributed random noise with mean 0 and variance 0.16, k_a, k_h, k_v and k_d are again the weighting coefficients depending on the changes of the edge directions d of the wavelet coefficients magnitudes. d_{max} and d_{min} are the maximum and minimum value of d. The added noises make the directions of the edges more random, which enhance the non-line-likeness of the texture, as shown in Fig. 5.8(d).

5.4 Analysis of the enhancement methods

Based on the discussion in Chapter 5.3.1, the enhancement methods should not only accentuate the corresponding characteristics, but also not affect the other characteristics that the components to be enhanced don't represent, or at least not affect as much as the enhancement methods for those certain characteristics. Therefore, in this chapter, we will discuss the insensitivity of these enhancement methods in affecting other texture characteristics.

5.4.1 Insensitivity of the coarseness-enhancement methods

Coarse-enhancement method

According to the function for enhancing the coarse component, as shown in Eq. 5.7, the weighting coefficient k is close to 1 if the difference d is small and k is close to 0 if d is large. Since the d measures the difference between the magnitude of the wavelet coefficient and its Gaussian filtered value, d is small when the centre magnitude is larger than its neighbourhood, representing the strong edges in the image, and d is large when the magnitudes in the neighbourhood are close to each other, representing the smooth or the weak-edge regions. Therefore, after the enhancement, the strong edges where the gradients are large will be preserved and the smooth or weak-edge-filling regions will be smoothened more. The change of the coarseness feature map is shown as Fig. 5.9(3b).

The change of contrast feature map is shown in Fig. 5.9(3c). The contrast, which measures the difference between the largest intensity and the smallest intensity, will not be changed too much, because the maximum is more likely to occur in the strong-edge regions, which are preserved by the enhancement method, and the minimum is more likely to occur in the weak-edge regions, which are degraded, but the change won't be large because of its original low value.

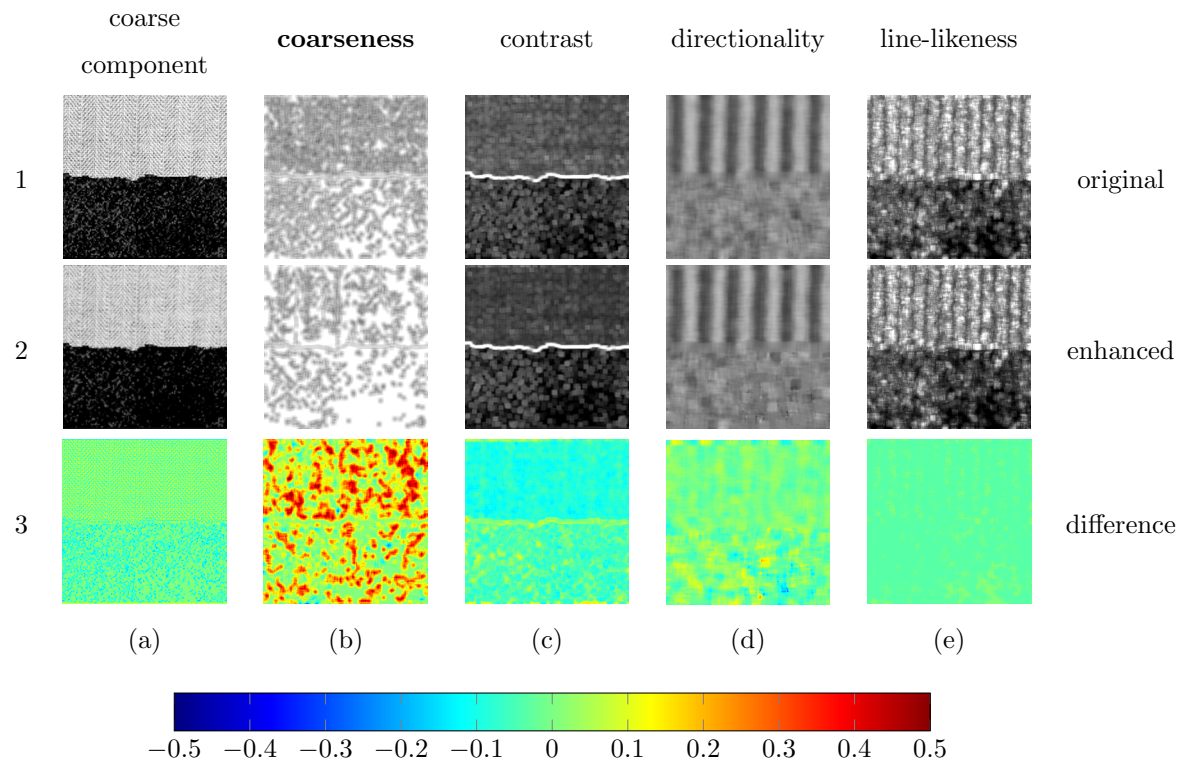


Figure 5.9: The changes from the original coarse component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component and its features, enhanced component and its features, the differences between the original component and the enhanced one displayed by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean"large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method increases a lot the coarseness (red in the coarseness difference map) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

The changes of the directionality and line-likeness feature maps are shown in Fig. 5.9(3d) and Fig. 5.9(3e) respectively. The directionality and the line-likeness, which measures the mean value and the variance of the distribution of the edge directions, will not be changed much because the enhancement functions are applied to every wavelet sub-bands, without the bias in any certain directional sub-band.

Fine-enhancement method

The fine component enhancement is described in Eq. 5.8, which adds weighted random noise to the regions where the magnitudes of wavelet coefficients change rapidly. The weighting coefficient k is a function of the local standard deviation d of the coefficient magnitudes, so k is close to 0 if d is small and k is close to 1 if d is large. Since d measures the speed of the change of the magnitude, it is small at the smooth region where the magnitudes change gradually and it is large around the edges where the magnitudes change quickly. The edges are cut into "pieces", leading to an increasing number of edges in the local region therefore the fineness is enhanced, as shown in Fig. 5.10(3b).

The change of the local contrast of the enhanced fine component is shown in Fig. 5.10(3c). The contrast is not changed too much because the noise is uniformly, randomly distributed in the local window without increasing or decreasing the maximum, minimum or the gray levels too much.

The changes of the directionality and line-likeness of the enhanced component are shown in Fig. 5.10(3d) and Fig. 5.10(3e) respectively. With the same reason for the enhancement of the coarse component that the modifications occur in every wavelet sub-bands, these two features remained almost the same in the enhanced fine component.

5.4.2 Insensitivity of the contrast-enhanced methods

high-contrast-enhancement method

The high-contrast component is enhanced as Eq. 5.9, where the magnitudes of the coefficients are first processed by a weighted Laplacian filter and then added to the original magnitudes. The weights k are positively correlated to the magnitudes of the coefficients in wavelet high-frequency sub-bands, representing the occurrence of the pixels with high gradients to the surroundings. Therefore, the contrast of the component is increased as shown in Fig. 5.11(3c).

The change of the feature map for coarseness from the original to the enhanced high-contrast component is shown in Fig. 5.11(3b). The enhancement of the contrast increases the magnitudes of the existing edges in the neighbourhood, which strengthens some weak edges in the local regions, However, most of these additive edges occur at the regions where there are subtle gradients between the pixels, and their magnitudes are not large enough to be considered as "true" edges for the calculation of the coarseness, so the coarseness will not be affected as much as the contrast.

Fig. 5.11(3d) shows the change of the feature map for directionality. Since the enhancement of the pixels

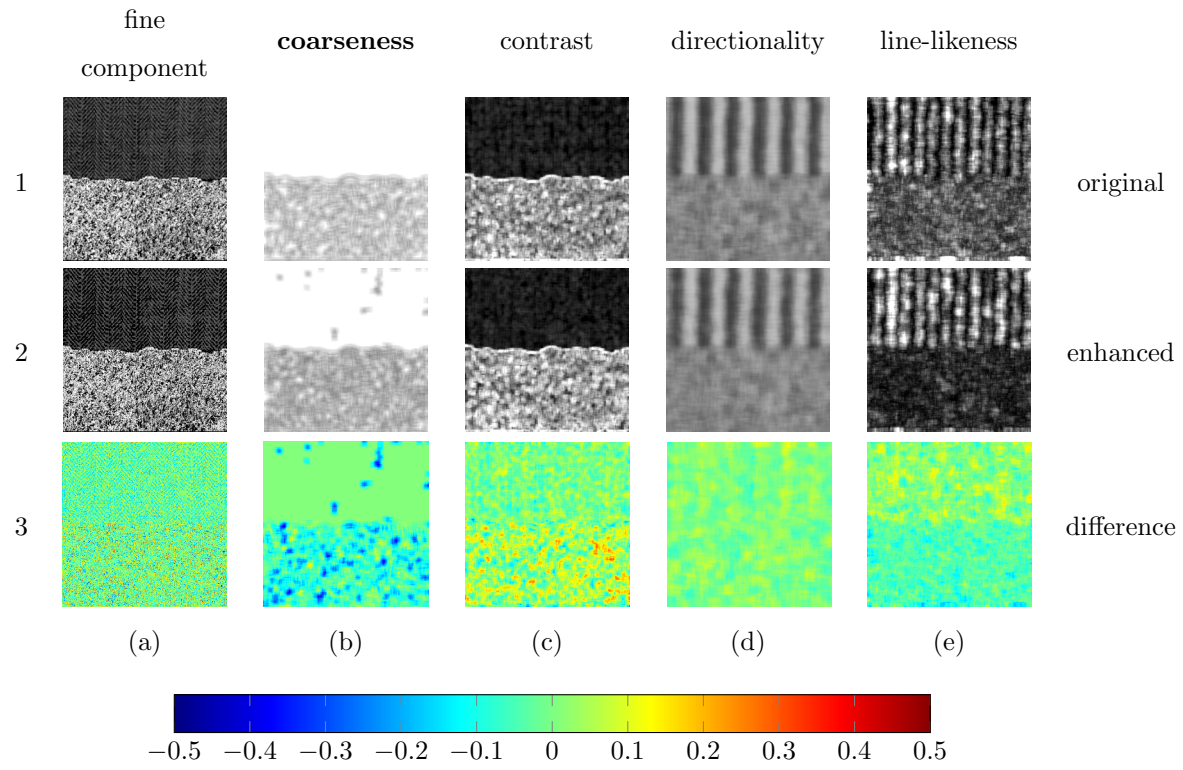


Figure 5.10: The changes from the original fine component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method decreases the coarseness a lot (blue in the coarseness difference map) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

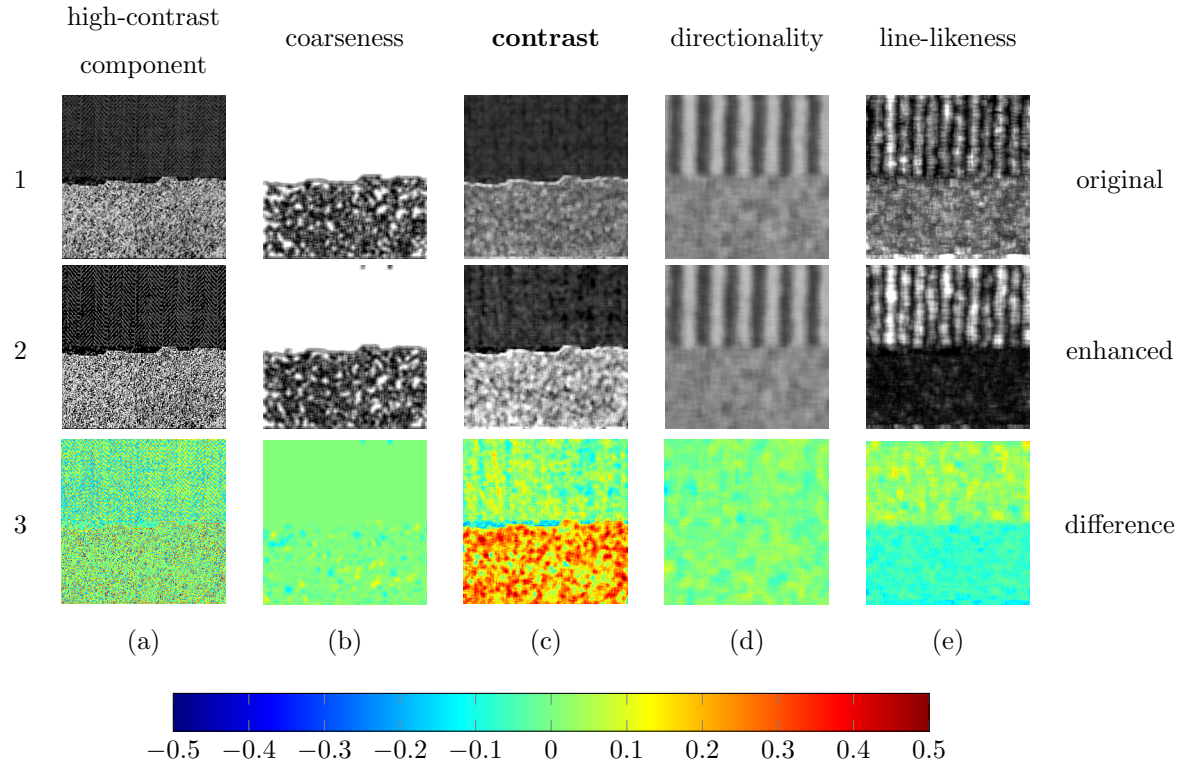


Figure 5.11: The changes from the original high-contrast component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method increases the contrast a lot (red in the contrast difference map) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

are positively correlated to the magnitudes of the gradients, the added edges with different directions but small magnitudes will not change much of the directionality that measures the weighted mean of the local edge directions.

The change of the feature map for line-likeness is shown in Fig. 5.11(3e). The increment of the magnitudes of the weak edges will increase the line-likeness if they have similar direction to the surrounding edges while decrease the line-likeness otherwise. However, the magnitudes of these additive edges are still quite small so the change of the line-likeness is not as much as that of the contrast.

low-contrast-enhancement method

The function for enhancing the low-contrast component is applied to the wavelet coefficients in approximate sub-band as shown in Eq. 5.10. The coefficients are first smoothed by the Gaussian filter and then divided by the weighting coefficient k representing the magnitudes of the image gradients. k gets larger when the magnitudes of the coefficients in horizontal and vertical sub-bands increase, so the pixels on the edges represented by these coefficients are smoothed more than the pixels in the smooth regions. Therefore, as shown in Fig. 5.12(3c), the contrast of the image is decreased by reducing the gradients between every pair of pixels.

The change of the coarseness map is shown in Fig. 5.12(3b), smoothening the pixels on the edges and removing the weak, isolated edges will increase the coarseness at the same time. However, the existing strong edges or the regions with high gradients are preserved so that the structure of the component is not changed much, i.e., the number of the edges, and therefore the coarseness is not be changed much.

The coefficients in the high-frequency sub-bands are not modified much so the existing edges are not changed much in structures, therefore the directionality and line-likeness, which measure the mean value and the variance of the local edge directions respectively. As shown in Fig. 5.12 (3d) and (3e), the changes of the directionality and line-likeness are not as much as that of the contrast.

5.4.3 Insensitivity of the directionality-enhancement methods

We discuss the insensitivity of the two directionality-enhancement methods together because they actually apply the same functions to enhance the image, just with a rotation of the filter and a different use in the horizontal and vertical wavelet sub-bands. As shown in Eq. 5.11 and Eq. 5.12, the enhancement applies the motion filter with one certain direction to the coefficients in the approximate sub-band and a certain directional sub-band. Since the filter will enhance the edges with the certain direction and remove the edges with directions other than that which the filter responds to, the horizontal or vertical edges distribute more uniformly in the image, causing any directionality to become more obvious, as shown in Fig. 5.13(3d) and Fig. 5.14(3d).

Take the horizontal component enhancement as example, though the vertical edges are removed, some weak horizontal edges are enhanced at the same time. Therefore, the total number of edges in the neighbour-

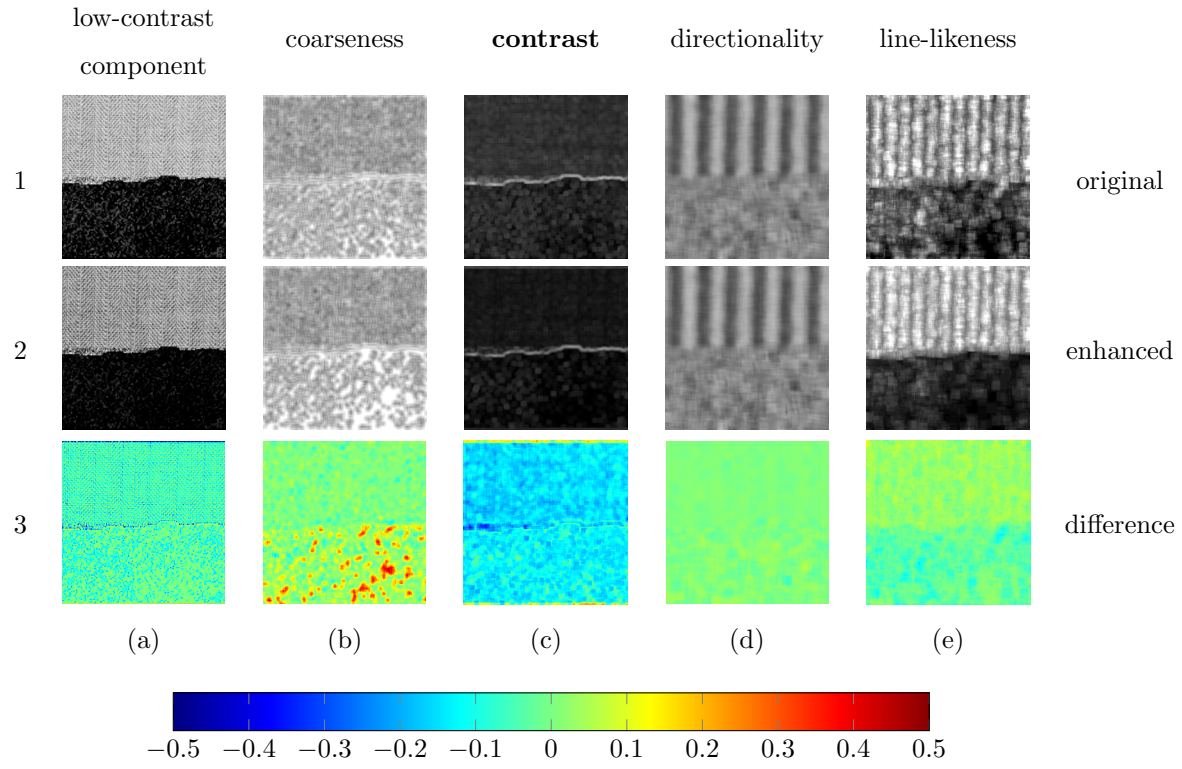


Figure 5.12: The changes from the original low-contrast component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method decreases the contrast a lot (blue in the contrast difference map) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

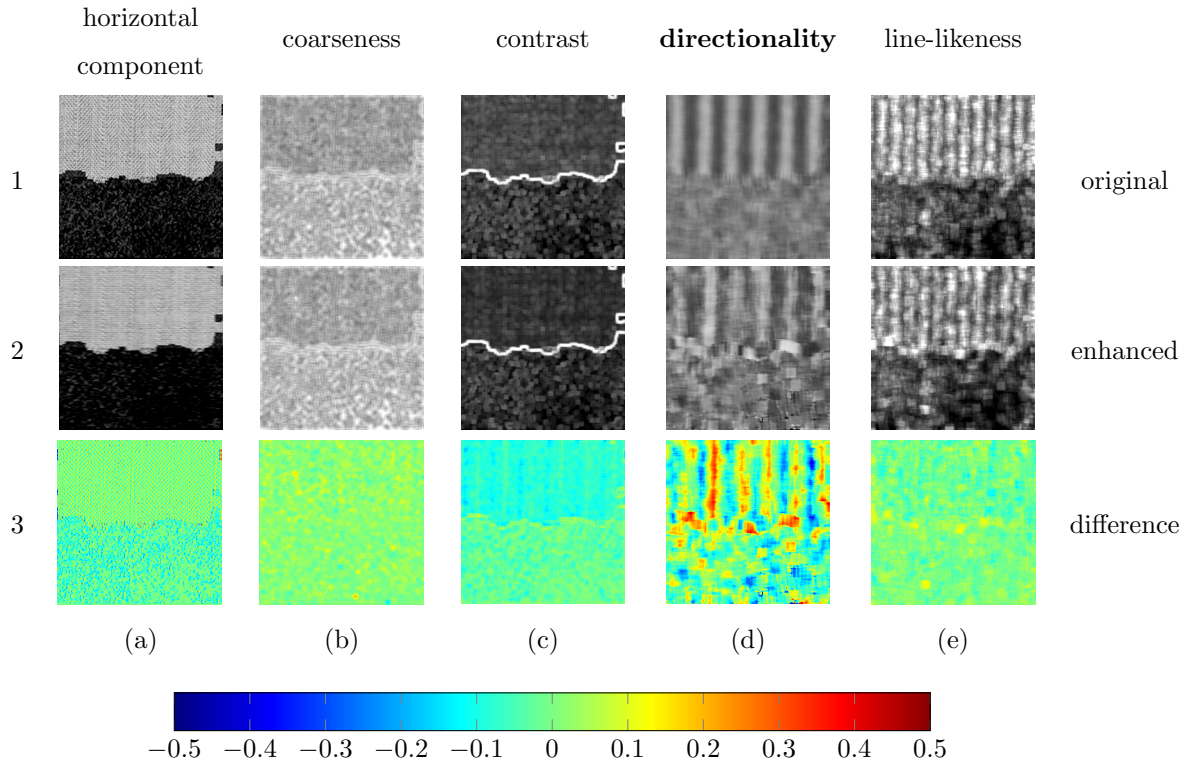


Figure 5.13: The changes from the original horizontal component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method changes the directionality much closer to horizontal ($F_{dir} = 0$ or $F_{dir} = 1$) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

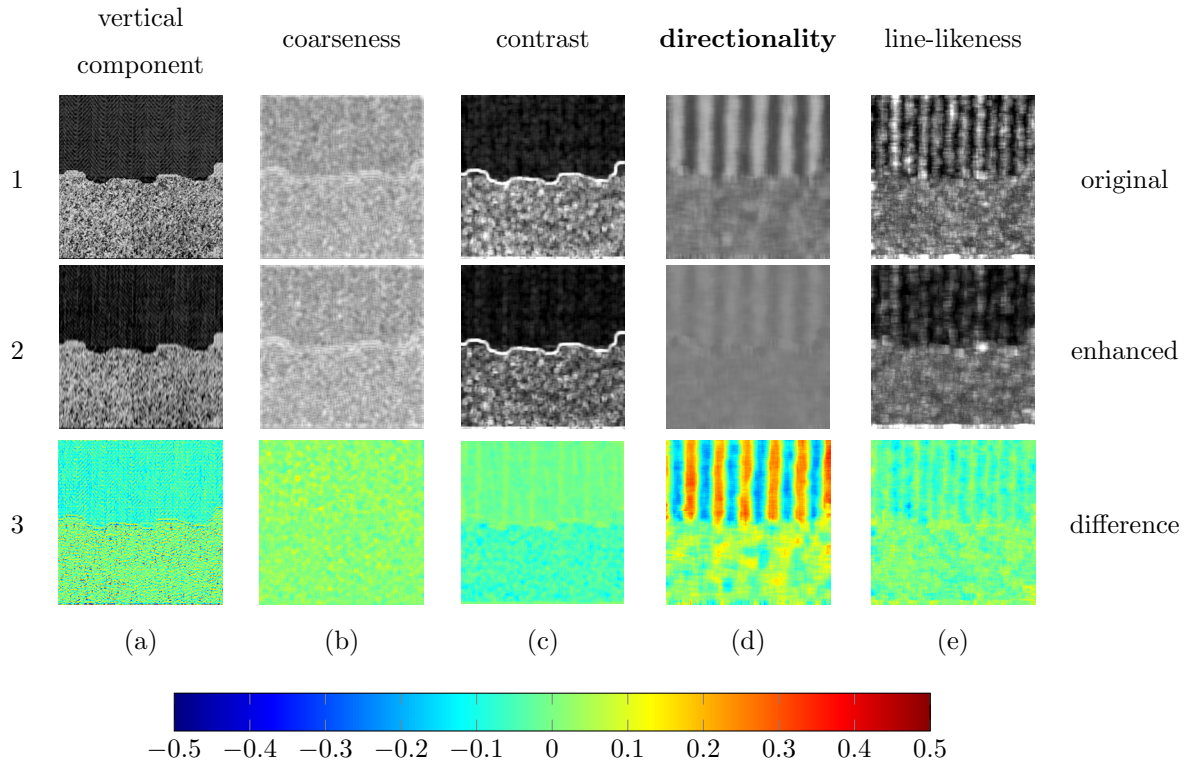


Figure 5.14: The changes from the original vertical component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method changes the directionality much closer to vertical ($F_{dir} = 0.5$) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

hood won't be changed much, leading to that the coarseness doesn't change much, as shown in Fig. 5.13(3b). The edges with large magnitudes in horizontal direction are preserved so that the local maximum won't be changed, while the edges with the other direction are suppressed but the local minimum won't be lowered too much because they are already with low magnitudes, therefore the local contrast of the enhanced component is not changed much as shown in Fig. 5.13(3c). Because the edges with horizontal direction are preserved as much as possible, the line-likeness, which measures the co-occurrence of the similar edge directions, will not be affected much as shown in Fig. 5.13(3e). The analysis for the vertical component enhancement method is the same as the above analysis for the horizontal component enhancement method.

5.4.4 Insensitivity of the line-likeness-enhancement methods

line-like-enhancement method

The enhancement of line-like component is implemented as Eq. 5.13, where local motion filters are applied to the neighbourhoods, making the edges close to similar direction locally. It means that in each neighbourhood, the image is processed as the enhancement for the directionality so that the edges in the local window have similar directions to the main direction of the neighbourhood. The change of the line-likeness of the image is shown as Fig. 5.15(3e), where the line-likeness is increased in most of the line-like regions.

As shown in Fig. 5.15(3b), to enhance the line-likeness, the edge structures are changed to have similar directions and some edges with very different directions to the surrounding ones are smoothed, therefore, the coarseness is increased because the number of edges in the neighbourhood is decreased. However, most of the edges in the line-like component have similar directions so the increment of the coarseness is not much.

The change of the contrast of line-like-enhanced component is shown in Fig. 5.15(3c). The contrast remains almost the same because the line-like component exhibits obvious directionality locally, then each local region is processed by the motion filter, which we discussed in Chapter 5.3.3 such that it won't affect the contrast of the component with obvious directionality much.

Since the edges in each local window have the similar direction and the local motion filter is adaptive to the certain direction for each neighbourhood, the directionality of the enhanced component won't be changed as much as the line-likeness, as shown in Fig. 5.15(3e).

non-line-like-enhancement method

The enhancement of non-line-like component is shown in Eq. 5.14. Weighted multiplicative noise is added to the wavelet coefficients representing edges in the image. The weights k for the noise are a function of the changes of the edge directions. When the change d is small, k is close to 1, so the regions with edges in similar directions are laced with strong multiplicative noise to reduce the homogeneity of the edge direction. When the change d gets large, k is close to 0, which prevents over-enhancement of the already non-line-like regions. Therefore, the component is getting more non-line-like but not corrupted with noise everywhere.

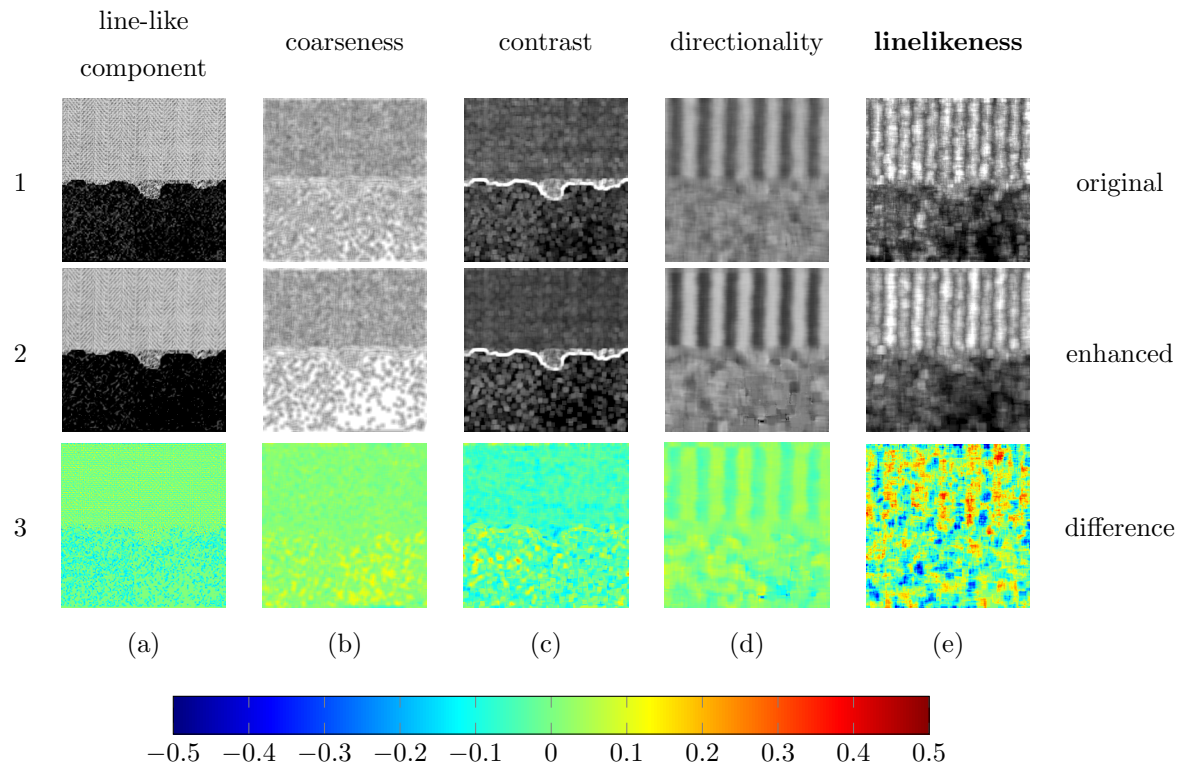


Figure 5.15: The changes from the original line-like component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method increases the line-likeness a lot (red in the line-likeness difference map) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

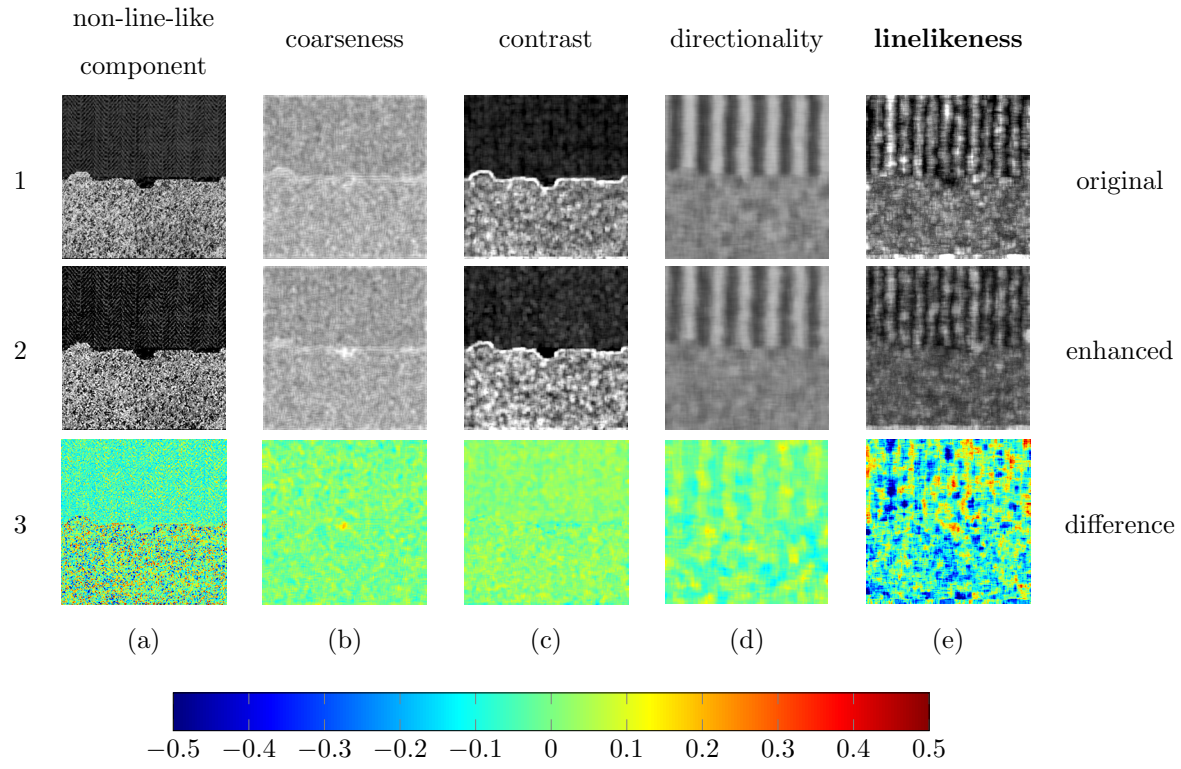


Figure 5.16: The changes from the original non-line-like component to the enhanced one with respect to image intensity, coarseness map, contrast map, directionality map and line-likeness map. Rows (from top to bottom): original component with its features, enhanced component with its features, the differences between the original component and the enhanced one displaying by colour maps, the colour bar showing the colour scale of the above colour maps (red and blue mean "large change", cyan and yellow mean "small change"). Columns (from left to right): the original and enhanced image intensities, feature maps with respect to coarseness, contrast, directionality and line-likeness, respectively. As the expectation, the proposed method decreases the line-likeness a lot (blue in the line-likeness difference map) and doesn't change other characteristics as much (cyan or yellow in the other difference maps).

As shown in Fig. 5.16(3b), the noise added to the image is multiplicative noise, which is correlated to the original texture structures, therefore the enhancement won't add extra edges or remove existing edges too much from the input component, leading to the coarseness not much affected.

As shown in Fig. 5.16(3c), the contrast of the enhanced component is increased by adding noise that increases the gray levels in the neighbourhood. However, the noise is uniformly distributed noise with zero-mean, so the gray levels of the enhanced component will increase and decrease randomly resulting in the contrast is not changed as much as the line-likeness.

For the directionality, since the edges in non-line-like component are with very different directions distributed evenly in the neighbourhood, the circular mean value of the directions in the local window is close to 0. After enhancement, the edges in the same neighbourhood will have more different directions and more randomly distributed, making the mean value of the directions still close to 0. Therefore, we conclude that the enhancement of non-line-like component won't change much of the directionality, as shown in Fig. 5.16(3d).

5.5 Experiments and analysis

In this part, each of the wavelet-based texture characteristic enhancement methods are evaluated for its performance in accentuating the corresponding texture characteristic in the image.

5.5.1 Experimental images, comparators and methods

Experimental images

To evaluate the enhancement of the texture characteristics by the proposed method, we use the texture images from the Brodatz texture database [24] and the SIPI database [208]. There is only one type of texture contents in a given texture image, so the mean value of the texture characteristics calculated from all pixels in the image is sufficient to describe the texture. Then the changes of the certain texture characteristics over all the texture images in the database can be conveniently calculated because only one value is assigned to one image with respect to one texture characteristic. Fig. 5.17 shows some examples of the testing images.

Comparators and experimental methods

Several state-of-the-art image enhancement methods are used here to compare with the proposed wavelet-based texture characteristic enhancement method:

1. Unsharp masking filter (UM) [158], which enhances texture by emphasizing its high frequency contents.

2. Wavelet VISUShrink method (VISU) [47], which enhances texture by removing noise via shrinking wavelet coefficients in high-frequency sub-bands not exceeding certain thresholds. It is used as an example of all the other similar wavelet soft thresholding methods;

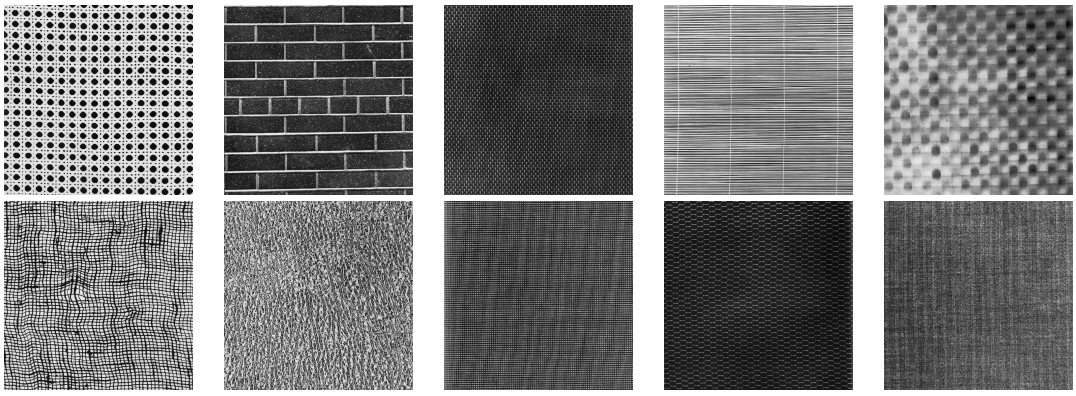


Figure 5.17: Examples of images used in the experiments. The texture images are from the Brodatz texture dataset [24] and the SIPI texture database [208]

3. Coherence-enhancing diffusion filter (CDF) [210], which preserves strong discontinuities at edges while removing artifacts from smooth regions, so that image textures are enhanced.

4. Shock diffusion filter (SHK) [209], which smooths along the coherent texture flow orientations and reduces diffusivity at non-coherent structures so as to enhance the texture details.

These comparator methods are selected to represent at least one of the major types of texture enhancement methods discussed in Chapter 2. Then the experiments are implemented as follows:

1. apply the proposed wavelet-based enhancement method and the comparators to the testing images;

2. compare the feature maps of the images and the images enhanced by different methods with respect to the 4 Tamura's texture characteristics: coarseness, contrast, directionality and line-likeness;

3. compute the features of the original images and those of the enhanced images by different enhancement methods. For each texture feature, the changes of mean values from the original images and the enhanced images are calculated, and the hypothesis testing is implemented both in each group of the feature values and between groups of the feature values before and after enhancement, so that it is determined if the observed differences are real effects.

5.5.2 Results of enhancing pure textures

Fig. 5.18 to Fig. 5.25 show the results of the enhancement of the coarse textures, fine textures, high-contrast textures, low-contrast textures, horizontal textures, vertical textures, line-like textures and non-line-like textures by the 8 wavelet-based enhanced methods respectively, compared with the other texture enhancement methods. The changes of the feature maps with respect to each characteristics before and after the enhancement are also shown in the corresponding figure. The unsharp masking (UM) filter firstly calculates the high-frequency contents by a high-pass filter, then adds the high-frequency contents back to get a sharpening image. Therefore, it works well in increasing the local contrast and decreasing the local

coarseness. But it doesn't affect other texture characteristics because it does nothing with edge direction. Wavelet VISUShrink or other wavelet shrinkage methods can reduce the high frequency content or noise in the image, therefore, the coarseness will be enhanced and the contrast will be decreased. However, the texture edges are not modified by the enhancement method, so that the directionality and line-likeness cannot be enhanced, either the strong or the weak aspect. The coherence-enhancing diffusion filter enhances the directionality and line-likeness by removing artifacts and connecting the continuous edges. But it cannot enhance the coarseness and contrast because the number of edges or gray levels in the local window is not modified. The shock filter (SHK) enhances the texture by producing a sharp discontinuity (shock) at the borderline or edge between two influence zones. The contrast will be increased and the line-likeness will be decreased because of the added sharp shocks. But since the shocks are added along the texture edges, the coarseness and the directionality that measure the number of local edges and the mean value of the local edge directions respectively are not modified much. Moreover, all these enhancement methods can only modify either the strong or weak aspect (not both) of certain texture characteristics. Our wavelet-based enhancement methods can enhance the image by applying different, independent modifications of the wavelet coefficients. Therefore, both strong and weak aspects of all the texture characteristics can be enhanced while affecting others less than the certain texture characteristics supposed to be enhanced. Table 5.2 shows the changes of mean values of the four texture features from the original images to the images enhanced by different methods. The certain feature values from the original images and enhanced images were subjected to a Lilliefors test ($P < 0.05$) [81]. The Lilliefor's test is a statistical hypothesis test that tests the null hypothesis that a group of samples is normally distributed against the alternative hypothesis that they are not. A Lilliefor's test for each group of the certain texture features failed to reject the null hypothesis in all cases. Therefore, we performed a student's T-test ($P < 0.05$) [81] on the groups of the feature values calculated from the original images and the enhanced ones. The student's T-test (for paired samples) tests the null hypothesis that the differences between paired samples are symmetrically distributed around zero, against the null hypothesis that they are not. Each of the wavelet-based texture characteristic enhancement methods can accentuate the corresponding texture characteristics statistically significantly ($P2 < 0.05$) while having no statistically significant effect on the other texture characteristics ($P2 > 0.05$).

		coarseness	contrast	directionality	line-likeness
original	mean	0.7476	0.2698	0.4846	0.4110
	P1	0.1372	0.2815	0.4808	0.0595
	P2	-	-	-	-
UM	mean	+0.0167	+0.4316	+0.0196	-0.0174
	P1	0.1907	0.1743	0.0740	0.2001
	P2	**0.0078**	**0.0114**	0.2337	0.3786
VISU	mean	+0.0394	-0.1553	+0.0026	-0.1374
	P1	0.1400	0.3109	0.1008	0.2320
	P2	**0.0059**	**0.0146**	0.7034	0.1036
CDF	mean	+0.0130	-0.0278	+0.0112	+0.0035
	P1	0.0837	0.4261	0.3219	0.3668
	P2	**0.0073**	**0.0157**	0.4732	0.4113
SHK	mean	-0.0075	+0.0751	-0.0063	-0.0126

	P1	0.2855	0.4990	0.2479	0.1781
	P2	**0.0176**	**0.0013**	0.5358	0.4883
	mean	+0.0837	-0.0136	-0.0098	-0.0073
WT-COR-S	P1	0.4945	0.1140	0.4637	0.2433
	P2	**0.0015**	0.0675	0.3565	0.4686
	mean	-0.0720	+0.0180	+0.0055	-0.0022
WT-COR-W	P1	0.3456	0.2947	0.2535	0.1661
	P2	**0.0343**	0.0579	0.4456	0.5886
	mean	-0.0017	+0.2346	+0.0005	-0.0120
WT-CON-S	P1	0.3635	0.2343	0.2764	0.3229
	P2	0.6284	**0.0069**	0.8626	0.2188
	mean	+0.0048	-0.1721	-0.0011	-0.0120
WT-CON-W	P1	0.2307	0.0505	0.3829	0.1849
	P2	0.8987	**0.0167**	0.5888	0.2224
	mean	+0.0031	+0.0175	+0.0900	+0.0045
WT-DIR-S	P1	0.1456	0.4057	0.2169	0.1097
	P2	0.0936	0.0787	**0.0106**	0.2248
	mean	+0.0051	-0.0124	-0.0507	-0.0035
WT-DIR-W	P1	0.4958	0.1425	0.2642	0.2457
	P2	0.0877	0.0679	**0.0177**	0.6204
	mean	-0.0033	-0.0156	0.0048	+0.0850
WT-LIN-S	P1	0.1228	0.0998	0.2280	0.3721
	P2	0.3864	0.0656	0.7498	**0.0226**
	mean	+0.0071	+0.0194	+0.0061	-0.0750
WT-LIN-W	P1	0.3808	0.2625	0.3488	0.2321
	P2	0.4127	0.0688	0.2146	**0.0224**

Table 5.2: The changes of the mean values of the texture characteristics of the 112 Brodatz textures before and after different enhancement methods: UM[158], VISU[47], SHK[210], CDF[209] and the proposed eight wavelet-based texture characteristics enhancement methods. For each enhancement method, P1 is the P-value of doing Lilliefors test ($P < 0.05$) [81] for each group of the certain texture features, and P2 is the P-value of doing the student's T-test ($P < 0.05$) [81] for the groups of the certain texture features calculated from the original images and the enhanced images, since the texture features in each group are normally distributed ($P1 > 0.05$).

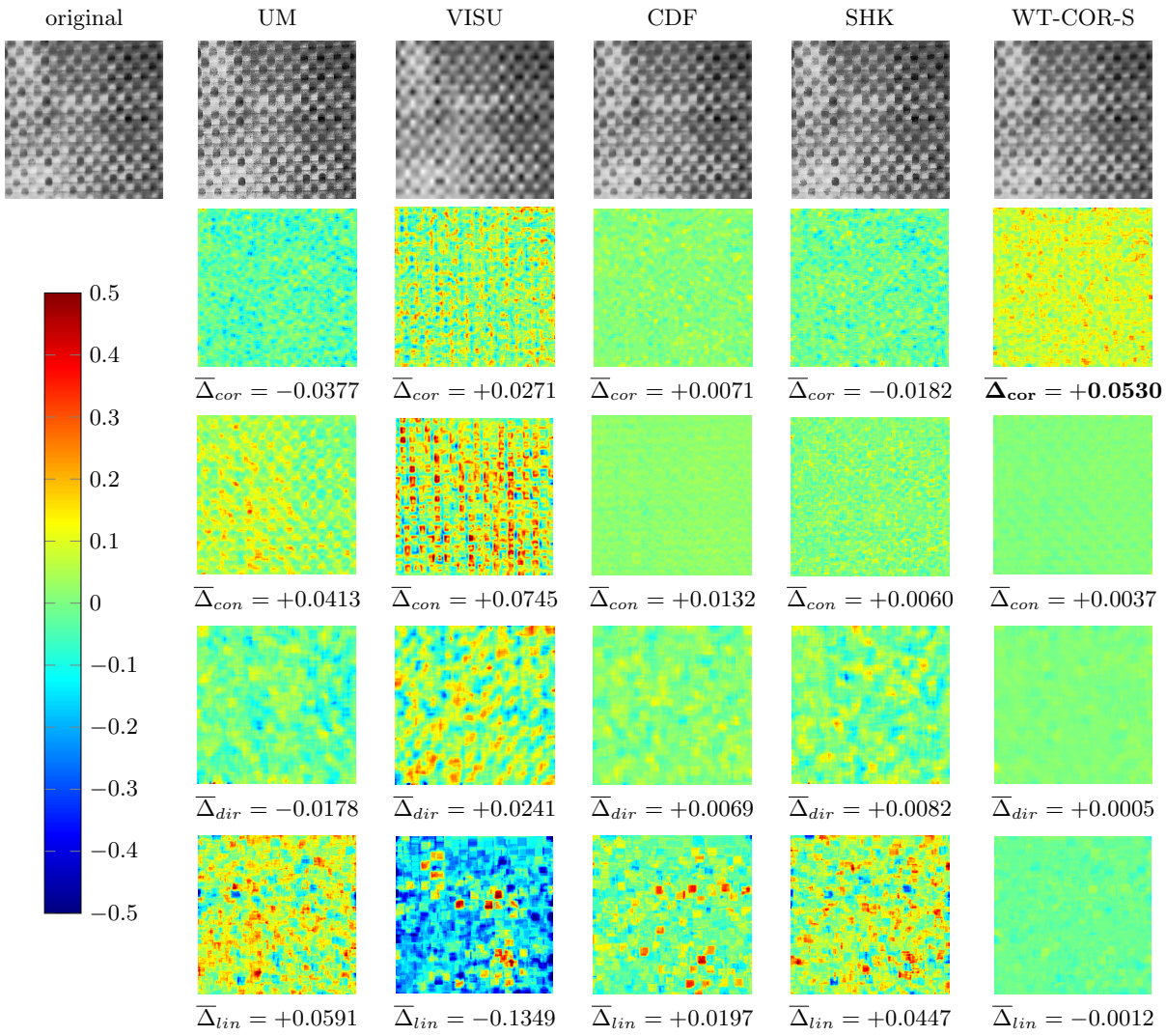


Figure 5.18: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **coarse** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based coarse-enhancement method (WT-COR-S). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of coarse enhancement is to increase the coarseness of the image (represented by the red colour in coarseness difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The coarseness of the texture manipulated by the wavelet-based coarse-enhancement method obviously increases, meaning the texture is manipulated **coarser**, while the contrast, directionality and line-likeness of the manipulated texture are not changed as much as the coarseness, proving the insensitivity of the coarse-enhancement method. The comparing enhancement methods have little effect in increasing the coarseness, or change other characteristics much while increasing the coarseness of the texture: UM, CDF and SHK cannot increase the coarseness, VISU can increase the coarseness but affects contrast and line-likeness too much

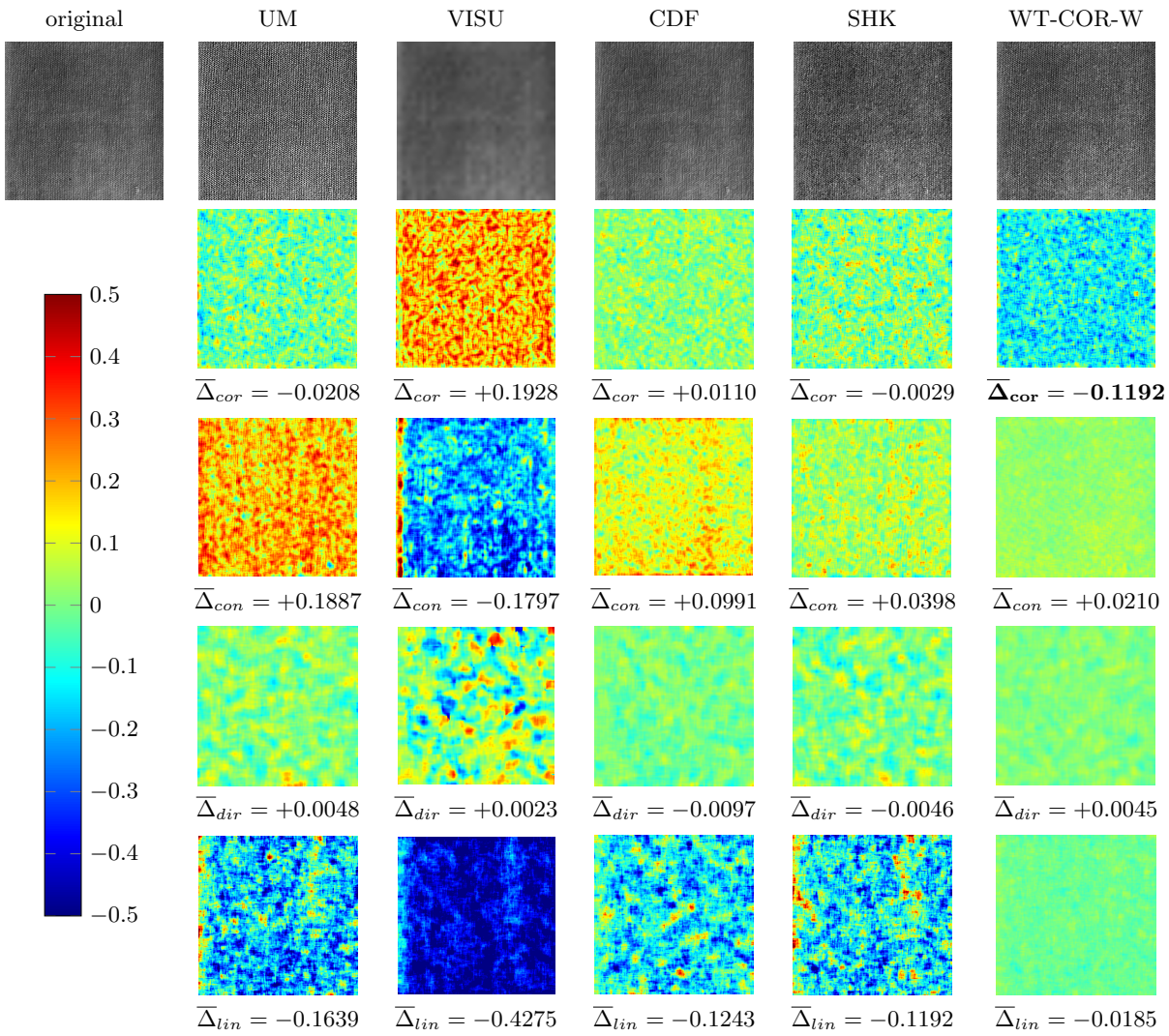


Figure 5.19: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **fine** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based fine-enhancement method (WT-COR-W). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of fine enhancement is to decrease the coarseness of the image (represented by the blue colour in coarseness difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The coarseness of the texture manipulated by the wavelet-based fine-enhancement method obviously decreases, meaning the texture is manipulated **finer**, while the contrast, directionality and line-likeness of the manipulated texture are not changed as much as the coarseness, proving the insensitivity of the coarse-enhancement method. The comparing enhancement methods have little effect in decreasing the coarseness, or change other characteristics much while increasing the coarseness of the texture: UM and SHK can decrease the coarseness but affect the line-likeness and contrast too much, VISU increases the coarseness and decreases the contrast and line-likeness, CDF cannot change the coarseness much.

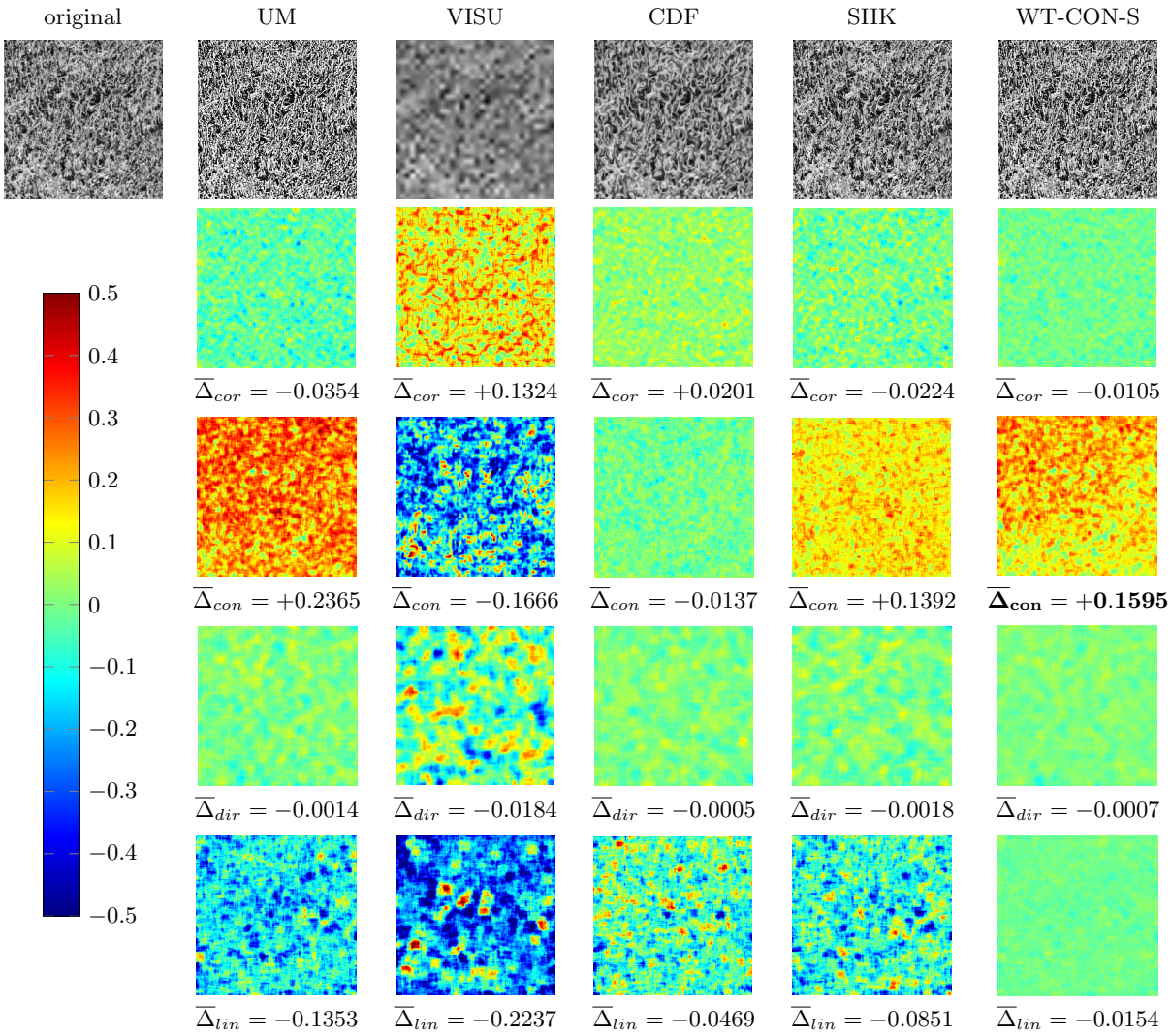


Figure 5.20: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **high-contrast** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based high-contrast-enhancement method (WT-CON-S). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of high-contrast enhancement is to increase the contrast of the image (represented by the red colour in contrast difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The contrast of the texture manipulated by the wavelet-based high-contrast-enhancement method obviously increases, meaning the texture is manipulated with **higher contrast**, while the coarseness, directionality and line-likeness of the manipulated texture are not changed as much as the contrast, proving the insensitivity of the high-contrast-enhancement method. The comparing enhancement methods have little effect in increasing the contrast, or change other characteristics much while increasing the contrast of the texture: UM and SHK can increase the contrast but decrease the line-likeness too much, VISU decreases the contrast, CDF cannot change the contrast much.

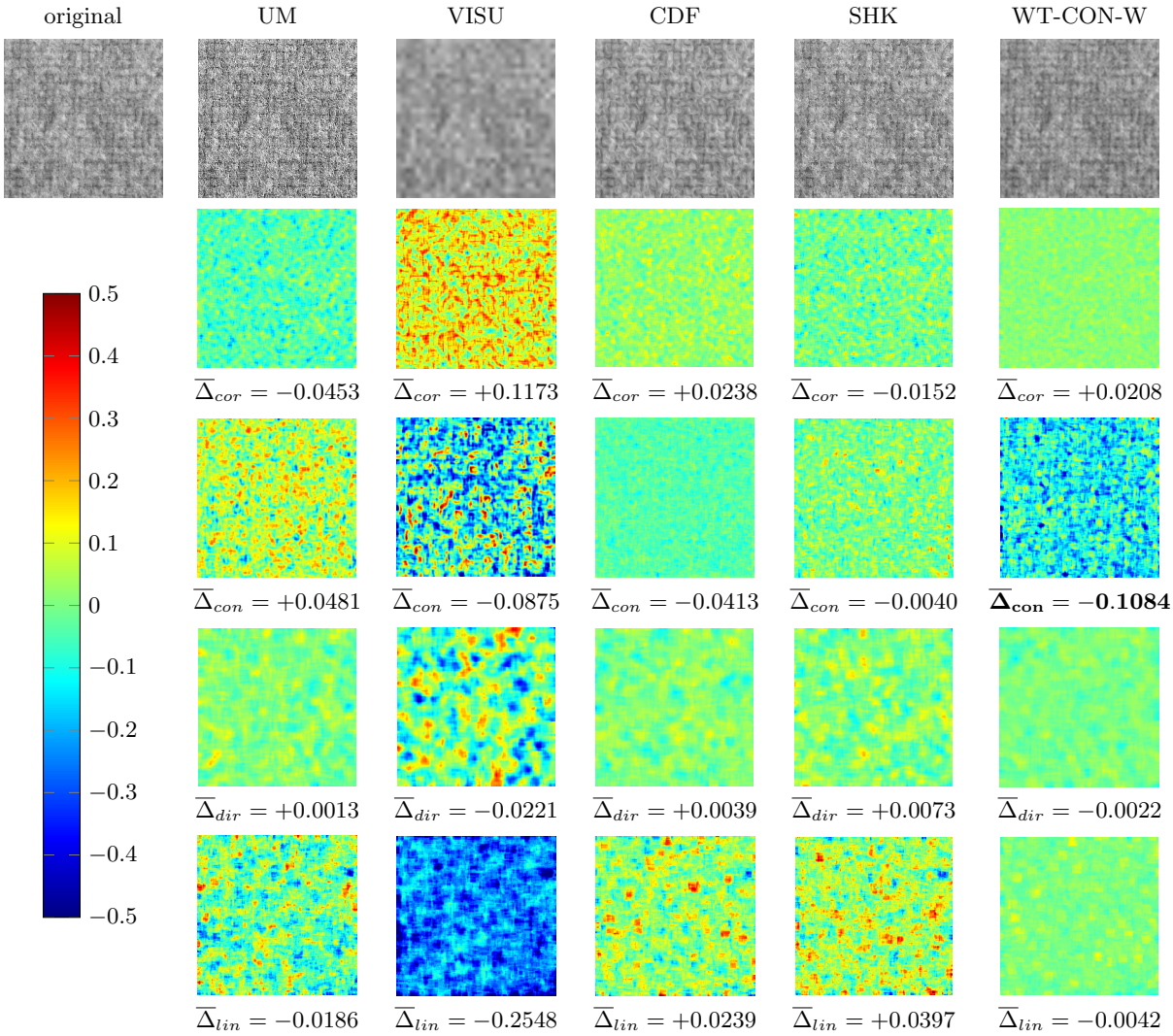


Figure 5.21: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **low-contrast** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based low-contrast-enhancement method (WT-CON-W). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of high-contrast enhancement is to decrease the contrast of the image (represented by the blue colour in contrast difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The contrast of the texture manipulated by the wavelet-based low-contrast-enhancement method method decreases a lot, meaning the texture is manipulated to **lower contrast**, while the coarseness, directionality and line-likeness of the manipulated texture are not changed as much as the contrast, proving the insensitivity of the low-contrast-enhancement method. The comparing enhancement methods have little effect in decreasing the contrast, or change other characteristics much while decreasing the contrast of the texture: VISU can decrease the contrast but decreases the line-likeness too much at the same time, UM and SHK increases the contrast, CDF cannot change the contrast much.

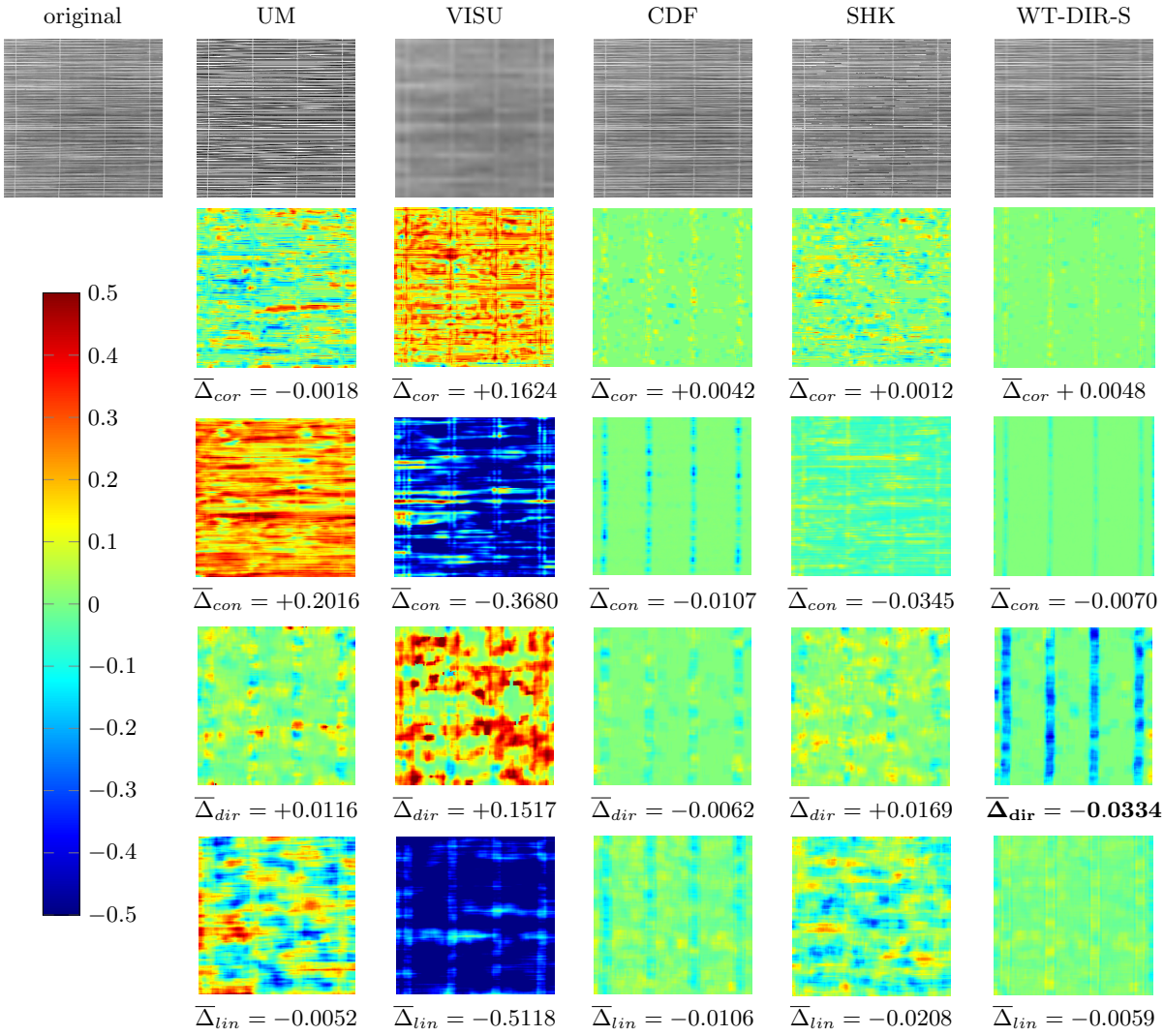


Figure 5.22: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **horizontal** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based horizontal-enhancement method (WT-DIR-S). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of horizontal enhancement is to decrease the directionality of the image (represented by the blue colour in directionality difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The wavelet-based horizontal-enhancement method decreases the directionality in the original vertical-like regions from 0.5 to around 0, therefore the directionality of the whole image is closer to 0, meaning the texture is **more horizontal**, while the coarseness, contrast and line-likeness of the manipulated texture are not changed as much as the directionality, proving the insensitivity of the horizontal-enhancement method. The comparing enhancement methods have little effect in manipulating the directionality, or change other characteristics much while changing the directionality: VISU can increase the directionality but affects other characteristics too much, UM, CDF and SHK cannot change the directionality much.

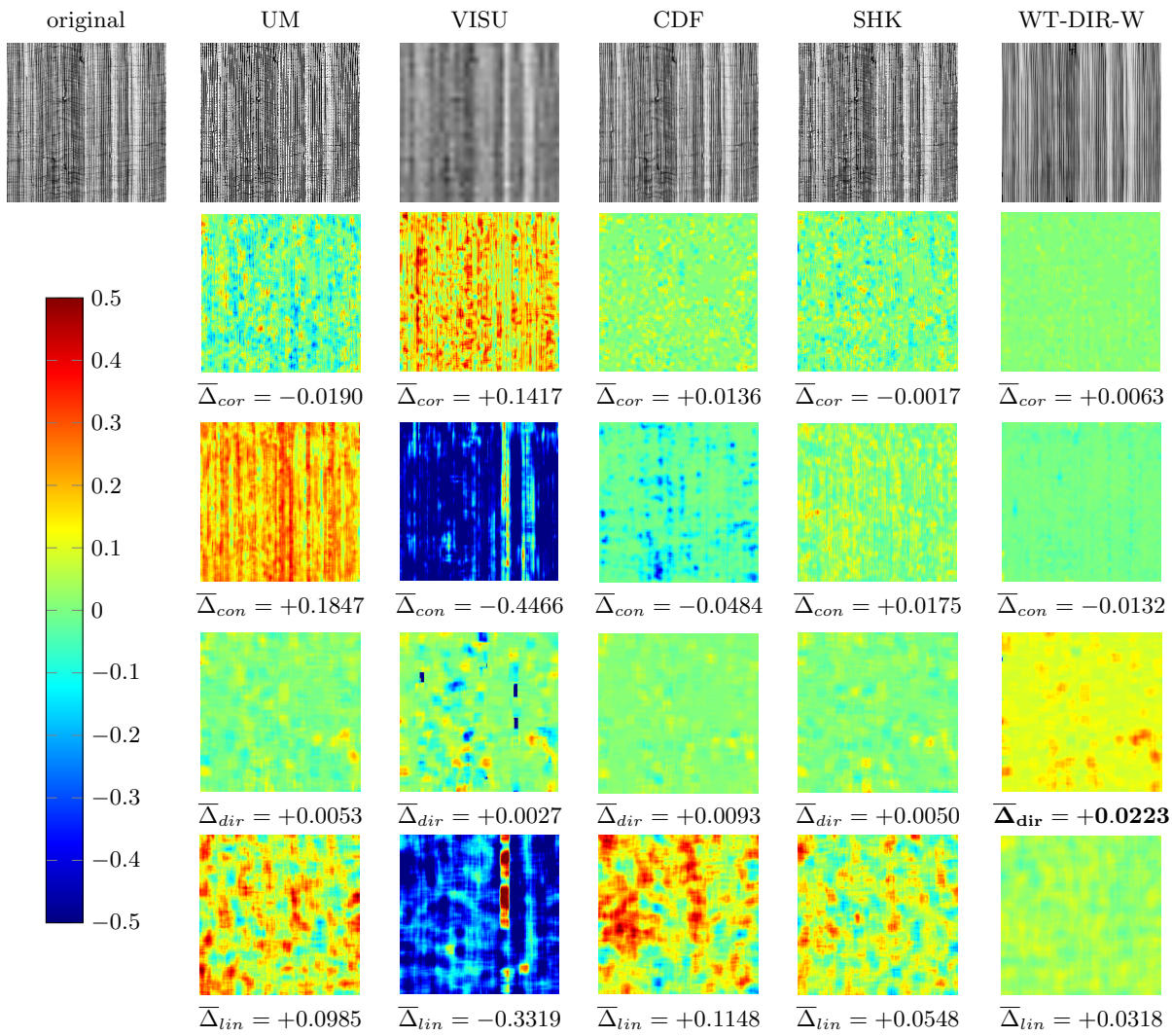


Figure 5.23: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **vertical** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based vertical-enhancement method (WT-DIR-W). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of vertical enhancement is to change the directionality of the image to 0.5 (represented by the dark yellow or light blue colour in directionality difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The directionality of the texture manipulated by the wavelet-based coarse-enhancement method increases closer to 0.5, meaning the texture is **more vertical**, while the coarseness, contrast and line-likeness of the manipulated texture are not changed as much as the directionality, proving the insensitivity of the vertical-enhancement method. The comparing enhancement methods have little effect in manipulating the directionality, or change other characteristics much while manipulating the directionality: VISU can decrease the directionality somehow but affects other characteristics too much, UM, CDF and SHK cannot change the directionality much.

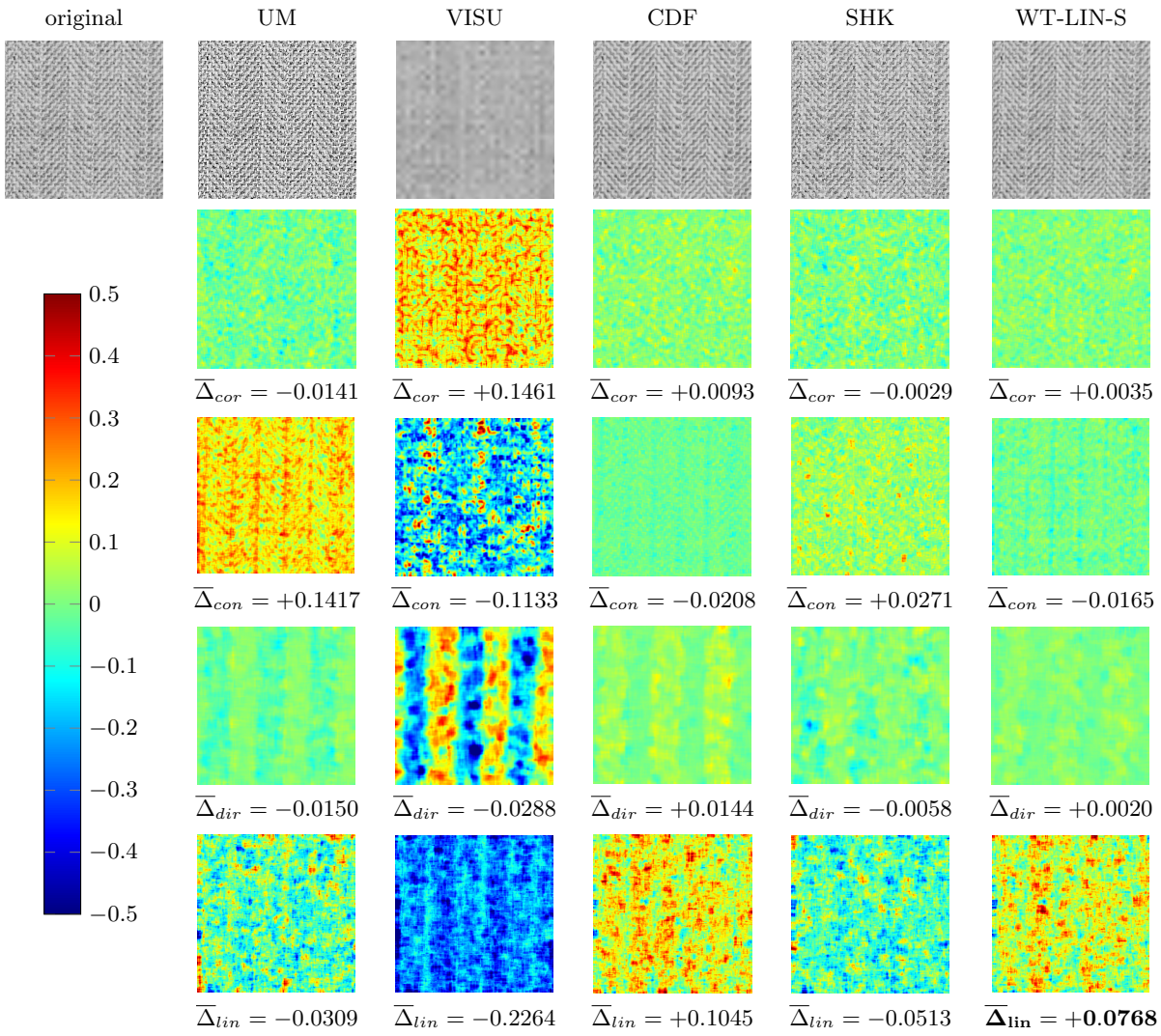


Figure 5.24: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **line-like** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based line-like-enhancement method (WT-LIN-S). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of line-like enhancement is to increase the line-likeness of the image (represented by the red colour in line-likeness difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The line-likeness of the texture manipulated by the wavelet-based line-like-enhancement method increases much more than the coarseness, contrast and directionality, meaning the texture is manipulated **more line-like** while the coarseness, contrast and directionality of the manipulated texture not changed as much as the line-likeness due to the insensitivity of the line-like-enhancement method. The comparing enhancement methods have little effect in increasing the line-likeness, or change other characteristics much while changing the line-likeness of the texture: CDF can increase the line-likeness but affect directionality much, UM, VISU and SHK all decrease the line-likeness.

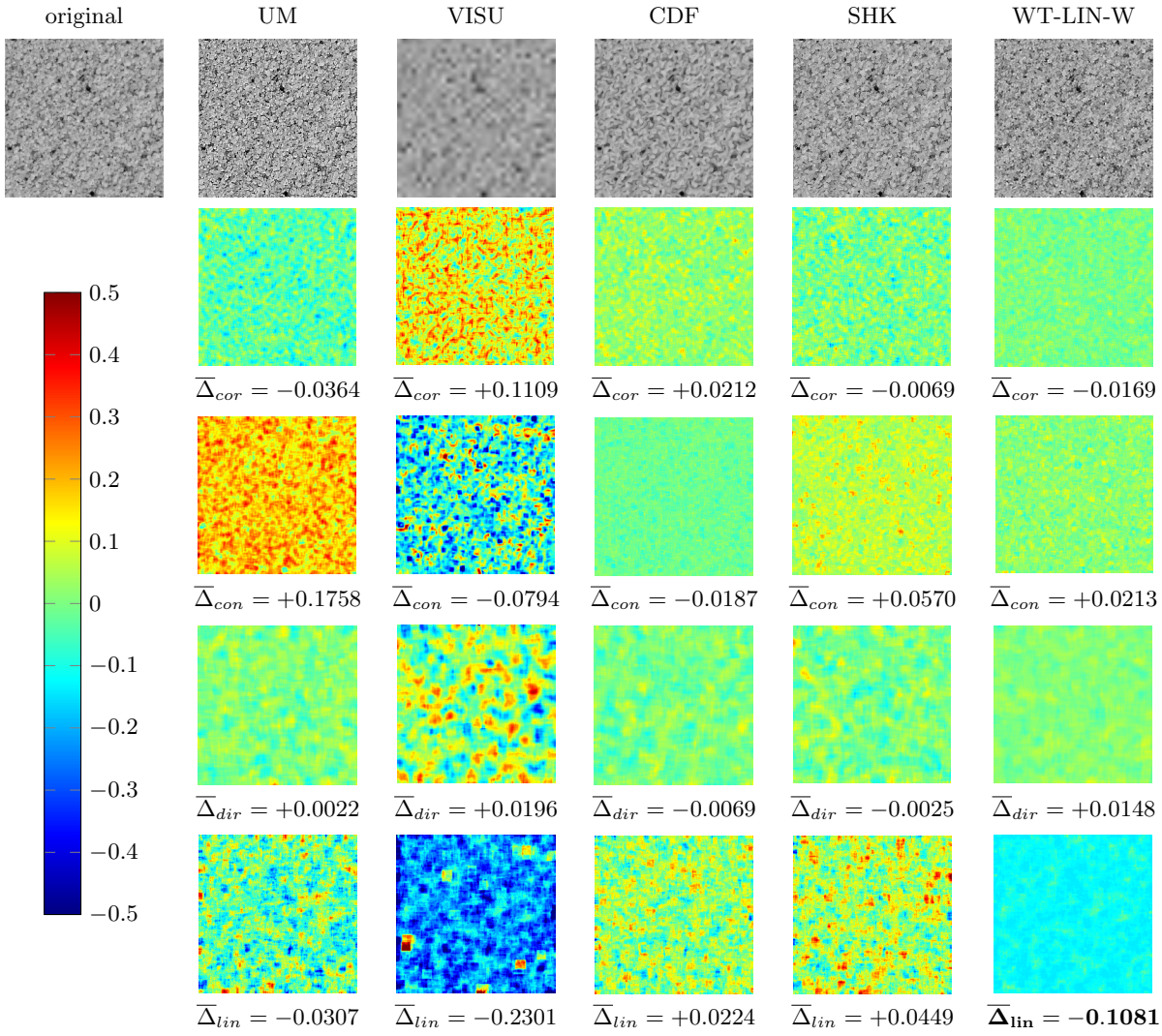


Figure 5.25: Comparison of the performances of different image enhancement methods in manipulating different texture characteristics of the example **non-line-like** texture image. Row 1: the original image and the images enhanced by UM[158], VISU[47], SHK[210], CDF[209] and the proposed wavelet-based non-line-like-enhancement method (WT-LIN-W). Row 2 to Row 5: the differences between the feature maps of the original image and those of each enhanced image with respect to coarseness, contrast, directionality and line-likeness displayed by colour maps, with the mean values of the corresponding textural differences $\overline{\Delta}_{cor}$, $\overline{\Delta}_{con}$, $\overline{\Delta}_{dir}$ and $\overline{\Delta}_{lin}$. The colour bar in first column shows the colour scale of the colour maps displaying the textural differences (red and blue mean "large change", cyan and yellow mean "small change"). The texture characteristics are selected and calculated as the definition in Chapter 3. The expected result of non-line-like enhancement is to decrease the line-likeness of the image (represented by the blue colour in line-likeness difference map) while not changing other texture characteristics that much (represented by the cyan or yellow colour in other characteristic difference maps). The line-likeness of the texture manipulated by the wavelet-based line-like-enhancement method decreases much more than the coarseness, contrast and directionality, meaning the texture is manipulated **more non-line-like** while the coarseness, contrast and directionality of the manipulated texture not changed as much as the line-likeness due to the insensitivity of the non-line-like-enhancement method. The comparing enhancement methods have little effect in decreasing the line-likeness, or change other characteristics much while changing the line-likeness of the texture: UM and VISU can decrease the line-likeness but affect contrast and coarseness too much, CDF and SHK both increase the line-likeness.

5.6 Conclusion

In this chapter, we proposed a wavelet-based texture characteristic manipulation method consisting of a series of functions that modify the wavelet coefficients of the given image. The image can be enhanced to exhibit any arbitrary texture characteristic more accentuated without affecting other characteristics of the image by using the corresponding wavelet coefficients modification function. Experiments are implemented to compare the performance of the proposed method and other methods in enhancing both texture images and natural images. The experimental results show that the proposed method can manipulate the image where each texture characteristic can be accentuated independently without interrupting other characteristics. With the manipulation, the components of the image representing different texture characteristics can be enhanced with the corresponding characteristics more noticeably.

Chapter 6

Wavelet-based textural difference enhancement based on morphological component analysis and the applications

The main problems in developing an image enhancement method that enlarges the textural differences in the image have been solved in the previous three chapters respectively and the proposed methods are summarized as follows:

1. in Chapter 3, the textural differences in a given image are locally described by the novel definitions of Tamura's texture characteristics, which are selected by the PCA-based feature selection method;

2. in Chapter 4, the given image is considered as the sum of components representing these texture characteristics, then these components are decomposed from the image based on the morphological component analysis using wavelet-based thresholding methods as dictionaries;

3. in Chapter 5, different texture characteristics are enhanced by wavelet-based enhancement methods independently without affecting other characteristics.

In the following parts, these methods are combined together to enhance the image so that different textures are more different, and this image enhancement method is shown to be effective as pre-processing step for various of image processing algorithms.

6.1 Textural difference enhancement method based on morphological component analysis

Fig. 6.1 shows the procedure of the WT-TC-MCA in magnifying the textural differences in the image. Intuitively, each step functions as follows:

1. the image is decomposed to several pairs of components where each pair consists of a component that strongly exhibits a particular texture characteristic and a component that weakly exhibits it, or exhibits opposite characteristics, e.g. a "coarse" component and a "fine" component;

2. the components are manipulated to enhance the texture characteristics they are meant to capture, e.g. a high-coarseness (coarse) component is manipulated so that so it becomes coarser, a low-coarseness (fine) component is manipulated so that it becomes finer;

3. the manipulated components are recombined to obtain an image in which textures are more different from each other than in the original image with respect to the chosen texture characteristics.

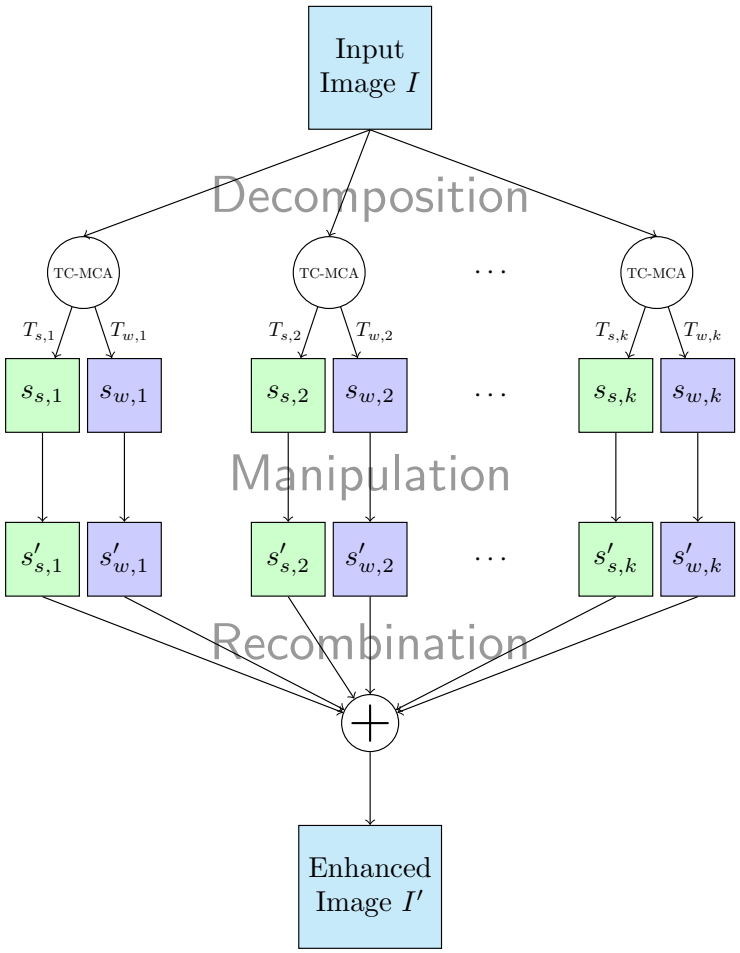


Figure 6.1: The process of enhancing image by wavelet-based texture characteristic morphological component analysis (WT-TC-MCA). The input image is decomposed to components representing strong and weak aspects of the selected characteristics. Components are then modified to highlight their representative characteristics. The enhanced image is the combination of the modified components so that the textural differences are more obvious in terms of the certain characteristics.

6.1.1 Decomposition of the image

For a given texture image I consisting of components representing different texture characteristics, WT-TC-MCA decomposes the image to pairwise components following the function described as Eq. 4.11 in Chapter

4, wherein each pair is composed of components showing the strong and weak aspects of a certain texture characteristic of the image respectively. For example, the input image as shown in Fig. 6.2 is decomposed to "coarse" and "fine" components according to coarseness, "high-contrast" and "low-contrast" components according to contrast, "horizontal" and "vertical" components according to directionality, "line-like" and "non-line-like" components according to line-likeness, respectively.

6.1.2 Manipulation of the image components

The components are then modified to singularize the texture characteristics they exhibit. Because the image components are still images, the manipulation of the components are implemented as the image texture characteristic enhancement method proposed in Chapter 5.3. Each image component exhibits one certain texture characteristic separately, so by applying the functions for enhancing the corresponding texture characteristics, the image components can be enhanced with the certain texture characteristics more apparent while preserving the other characteristics. The image components, shown in the second row of Fig. 6.2, are manipulated to obtain the enhanced components, shown in the third row of Fig. 6.2, where the corresponding texture characteristics are made more extreme.

6.1.3 Re-combination of the manipulated components

The manipulated components $s^{'}_{s,i}$ and $s^{'}_{w,i}$, which represent the strong and weak aspect of the i-th characteristic ($i = 1, 2, \ldots, k$) more distinctly, are re-combined to form the enhanced image $I^{'}$ as follows:

$$I' = \frac{1}{k} \sum_{i=1}^{k} \left(s^{'}_{s,i} + s^{'}_{w,i} \right). \tag{6.1}$$

The enhancement of an example image is shown in Fig. 6.2. The feature maps of the input image and the enhanced image in Fig. 6.2 with respect to coarseness, contrast, directionality and line-likeness are shown in Fig. 6.3. Consider the coarseness maps: the coarse regions with high values of coarseness are manipulated to regions with higher values in the feature map, while the fine regions with low values of coarseness are manipulated to regions with lower values, leading to magnified differences between two regions with respect to coarseness. The same enhancement happens in the feature maps of other three characteristics, therefore different textures in the manipulated image are more different than those in the original image.

6.1.4 Experiments on textural differences enlargement by the WT-TC-MCA

In this part, we'll evaluate the performance of the WT-TC-MCA for the task of textural difference enhancement, compared with other image enhancement methods. The experiment in Chapter 5 proves the ability of WT-TC-MCA to accentuate each certain texture characteristic, while the experiment here is concerned with evaluating the ability of WT-TC-MCA to magnify the textural differences between different texture regions in a given image with respect to the texture descriptor proposed in Chapter 3.

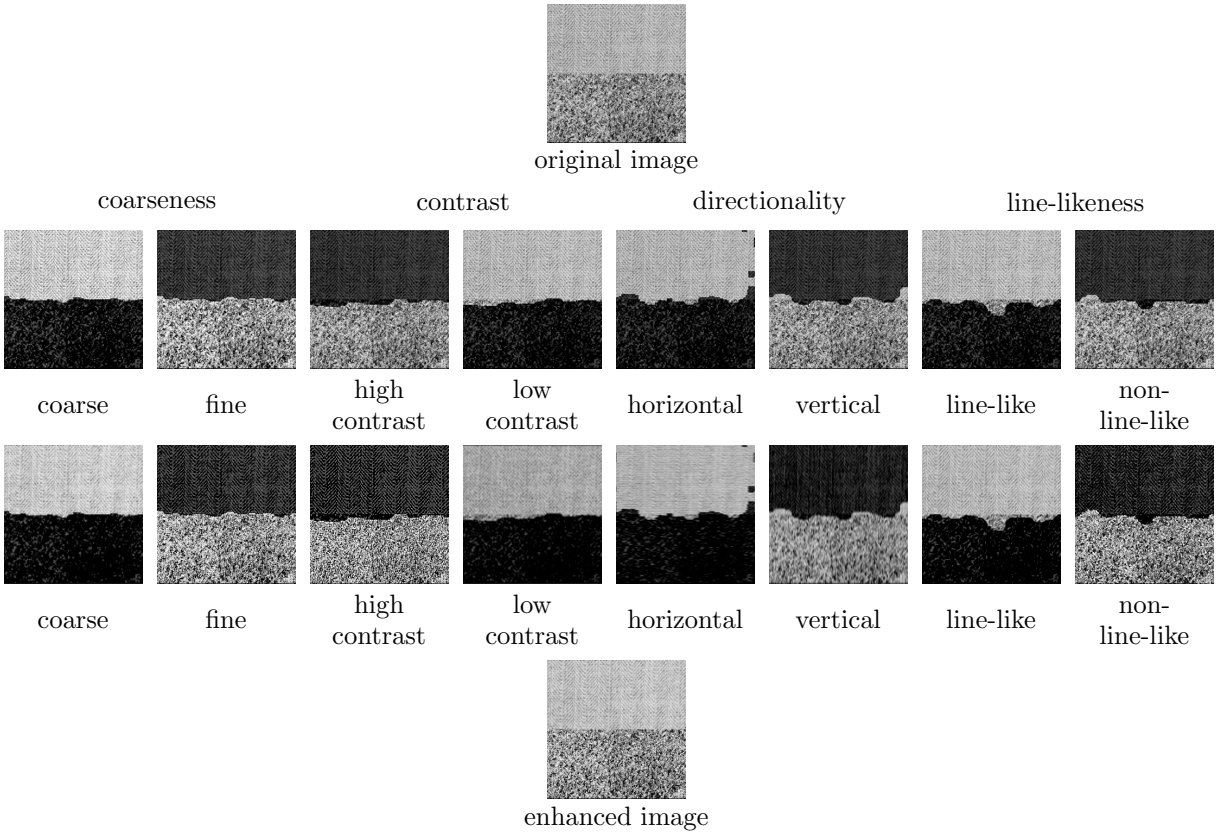


Figure 6.2: The enhancement of the texture image where there are multiple textures. The original image is first decomposed into components exhibiting a certain texture characteristic strongly and weakly. Then the components corresponding to strong and weak characteristics are manipulated with accentuated texture characteristics. The enhanced image is the re-combination of all the enhanced components equally.

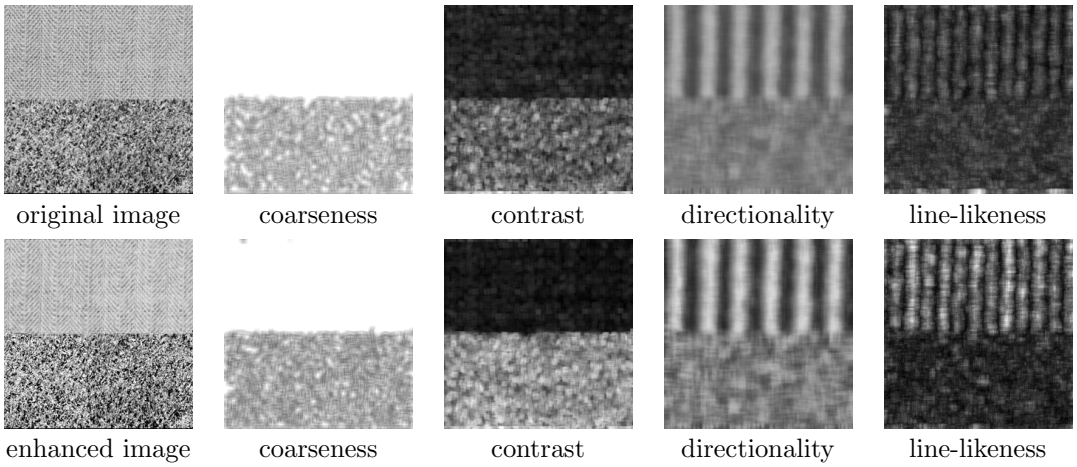
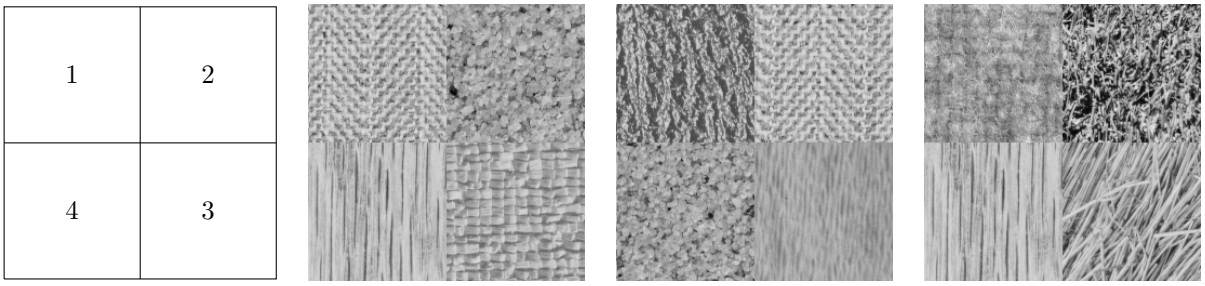


Figure 6.3: The coarseness, contrast, directionality and line-likeness of the original image and the image enhanced by the WT-TC-MCA. Two textures in the original image are made more distinct with respect to these texture characteristics.

Testing images, comparators and benchmarks

Similar to the experiments in Chapter 4, we use the synthesized mosaic texture images as a starting step to test the performance of different enhancement methods in magnifying the textural differences. These mosaic texture images consist of 4 textures randomly selected from the Brodatz [24] and SIPI [208] textures and combined following the diagram in Fig. 6.4(a). Examples of the mosaic texture images are shown as Fig. 6.4(b), (c) and (d). The textural differences in these images are relatively easy to calculate because texture regions are exactly pre-set and texture descriptions are homogeneous in each of the region with one single texture.



(a) Texture map for mosaic images (b) Mosaic texture image #1 (c) Mosaic texture image #2 (d) Mosaic texture image #3

Figure 6.4: Examples of mosaic texture images used in the experiments. (a) is the diagram to combine the Brodatz textures and SIPI textures into the mosaic images. (b), (c) and (d) are three examples of mosaic texture images synthesized using 4 textures randomly selected from the database and combined as the diagram in (a).

As discussed in Sec. 4.5.1, only these synthesized mosaic texture images are not sufficient for evaluating the performance of different texture enhancement methods because the images in actual enhancement problems are more complex with texture regions not as homogeneous as the pure textures. Therefore, we use the natural images from the BSDS500 dataset [135] as another category of testing images in our experiments for a more comprehensive evaluation of different enhancement methods in the textural differences enhancement. Fig. 6.5 shows some examples of these experimental natural images.

Figure 6.5: Examples of natural images used in the experiments. The images are from the the BSDS500 dataset [135].

As introduced in Chapter 5.5, the unsharp masking filter (UM) [158], wavelet VISUShrink method (VISU)

[47], standard MCA filter (MCA-CT), coherence-enhancing diffusion filter (CDF) [209] and shock diffusion filter (SHK) [210] are again used here as the comparators of the WT-TC-MCA to evaluate the performance of the textural differences enlargement because of their wide use in texture enhancement.

The textural differences between different textures in the image are measured by the Mahalanobis distance [132] between the clusters of four-dimensional vectors ([coarseness, contrast, directionality, line-likeness]) of different textures in the image. As shown in Fig. 6.6, for a given image with k different textures in it, there are k clusters of four-dimensional vectors for the original image ("before" clusters) and k clusters for the processed image ("after" clusters). We then computed the Mahalanobis distance between each "before" cluster mean and each other "before" cluster, resulting in A_k^2 Mahalanobis distances $d^i_{\text{before}}, i = 1, \ldots, A_k^2$. These A_k^2 Mahalanobis distances are illustrated on the right side of Fig. 6.6. The same was done for the "after" clusters, resulting in A_k^2 Mahalanobis distances $d^i_{\text{after}}, i = 1, \ldots, A_k^2$, with the mean-cluster pairs indexed by i in the same order as for the "before" images. Finally, we computed the difference between the "before" and "after" Mahalanobis distances for each mean-cluster pairing:

$$D_i = d^i_{after} - d^i_{before}, \quad i = 1, 2, \ldots, A_k^2. \tag{6.2}$$

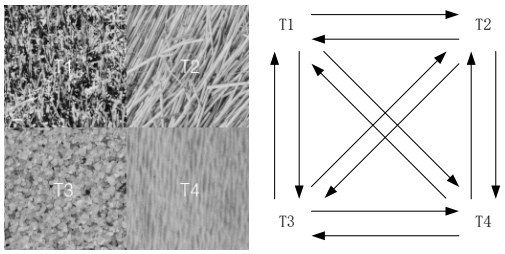


Figure 6.6: An example mosaic image with 4 textures. $T1$, $T2$, $T3$, $T4$ are 4 different textures, resulting in $A_4^2 = 12$ ordered-pairwise Mahalanobis distances. Textures become mutually more different due to a particular enhancement method if the 12 Mahalanobis distances are increased after processing.

Mahalanobis distance [132] measures the distance between a point and a cluster of points normalized with respect to the spread of the distribution of points in the cluster (intuitively, distances along each dimensional axis can be thought of as being in units of *standard deviations*). If all of the "before" and "after" Mahalanobis distances between mean-cluster pairs increase after processing, it means that the clusters have mutually moved away from each other in the multidimensional space and become more separated, which means that the textures have become mutually more different with respect to their descriptions by the four texture characteristics.

Our experiment was conducted as follows:

1. the testing images were enhanced by WT-TC-MCA and each of the comparator methods;

2. the local Tamura's features, consisting of the four specific characteristics: coarseness, contrast, directionality and line-likeness, were extracted from the 15×15 neighbourhood of every pixel of each original testing image and each enhanced testing image by each enhancement method to create feature maps;

3. the differences between the "before" and "after" Mahalanobis distances D_i were computed as Eq. 6.2 for each pair of textures in the original image and the image enhanced by different methods;

4. the changes of the Mahalanobis distances between the textures from the original images to the images enhanced by different methods are compared to evaluate how well the different texture enhancement methods were able to enlarge textural differences.

Experimental results on mosaic texture images

The results of enhancing an example mosaic texture image by different image enhancement methods are shown in Fig. 6.7. Fig. 6.8 shows all twelve D_i (Mahalanobis distance differences) for the 4 textures in Fig. 6.7 after processing by different methods. The UM filter increases the local contrast, decreases the local coarseness, but the directionality and line-likeness are remained the same. The VISU and MCA-CT filters increase the coarseness, decrease the contrast, but preserve the directionality and line-likeness because the structures of the texture are not changed. Coherence-enhancing diffusion filter (CDF) connects the edges with similar directions so that the directionality is changed and line-likeness is enhanced, however the coarseness and contrast are not changed because the number of edges and gray levels are not modified. Shock filter (SHK) adds sharp shocks between different influence zones so that the contrast is enhanced and the line-likeness is decreased, but the coarseness and the directionality that measure the number and the mean direction of local edges are not modified much. Moreover, since all these methods process the textures as a single "texture " component, all the different textures in the image are enhanced to the same extent. However, our WT-TC-MCA method can enhance different textures to different extents with respect to their own properties because it separates the textures into components representing different visual characteristics and modifies these components to accentuate the corresponding texture characteristics separately. As a result, the proposed texture enhancement method enlarges the textural differences between different textures in the enhanced image. As shown in Fig. 6.8, the proposed method is the only one for which the Mahalanobis distances between clusters is consistently increased. The comparator methods all exhibit pairs of clusters that become less separated, and even when clusters become more separated, the magnitude of the increased separation is generally less than that for the proposed method.

Experimental results on natural images

Fig. 6.9 shows images enhanced by the same methods and the feature maps of different texture characteristics in the corresponding images. Fig. 6.10 shows the Mahalanobis distance differences D_i for different textural regions in Fig. 6.9 before and after enhanced by different methods. Note that in this experiment, the natural image is considered as the composition of two textures: foreground textures and background textures. Therefore, the number of the Mahalanobis distance differences in the image is 2 ($A_2^1 = 2$). Like the results on mosaic texture images, the proposed method is the only enhancement method that enlarges the textural differences between the two regions with respect to our features corresponding to Tamura's texture

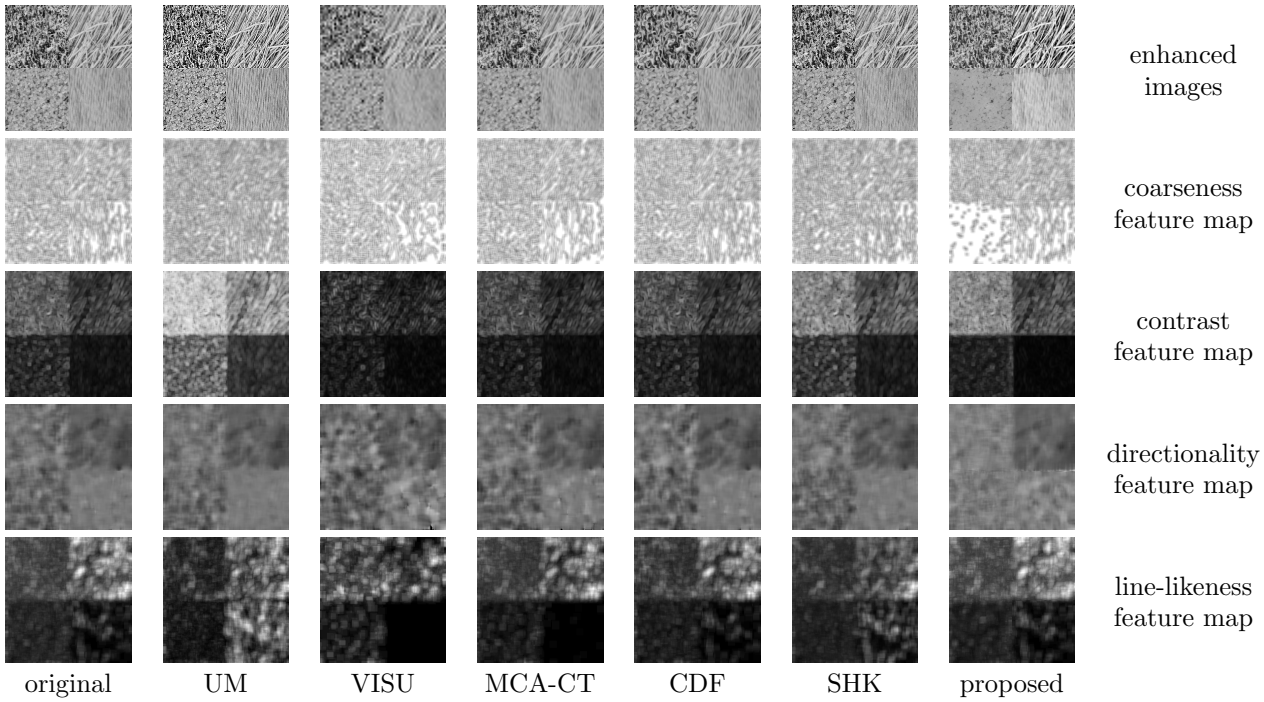


Figure 6.7: Comparison of the performances of different image enhancement methods in making textures more distinct with respect to the texture characteristics. Columns: the original image and images enhanced by UM[158], VISU[47], traditional MCA[189], SHK[210], CDF[209] and the proposed WT-TC-MCA. Rows: the maps of coarseness, contrast, directionality and line-likeness of the original and enhanced images.

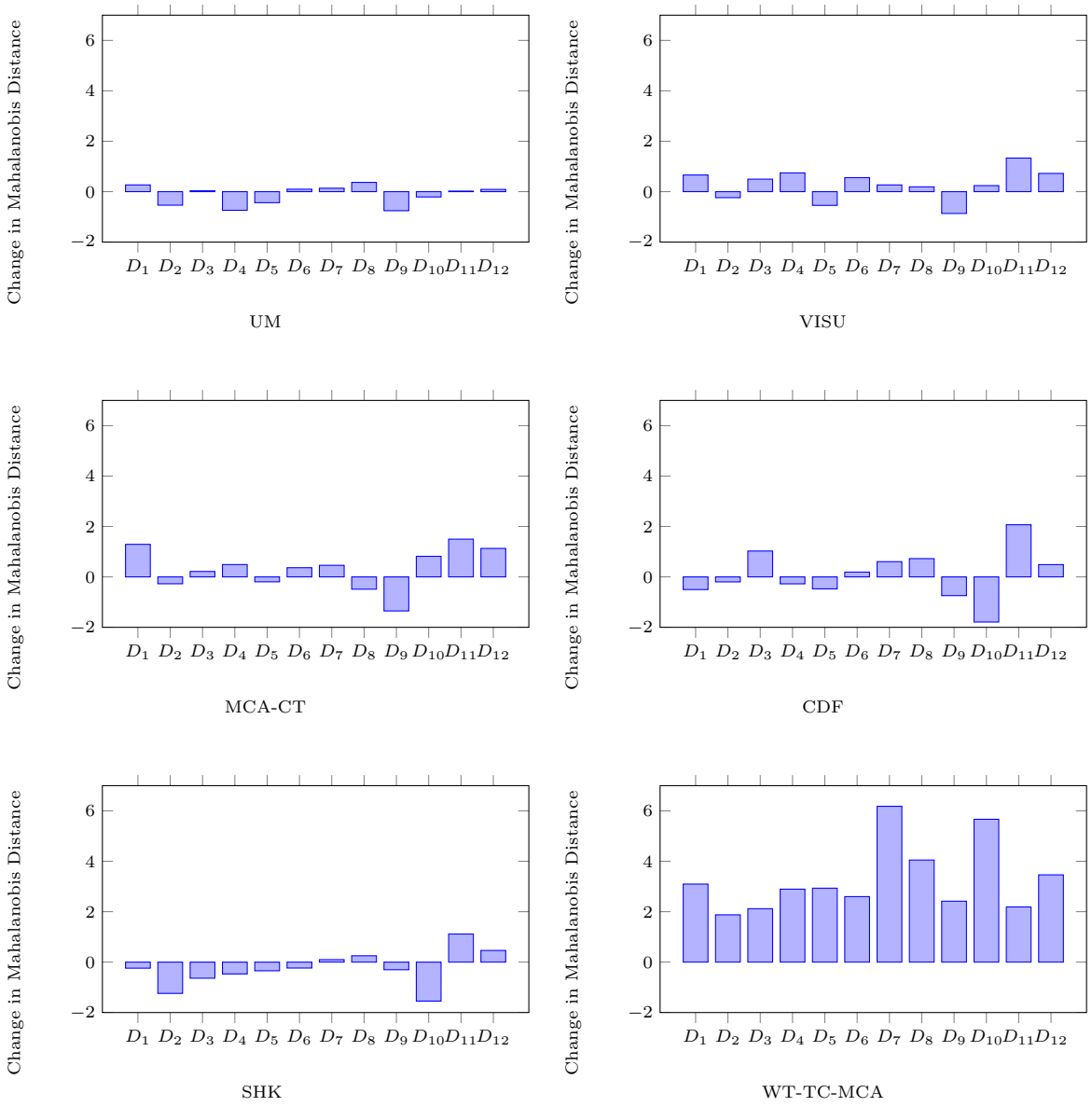


Figure 6.8: The changes in the twelve Mahalanobis distances between clusters of local texture characteristic vectors in the image in Figure 6.6 after processing with unsharp masking (UM), VISUShrink (VISU), MCA-CT filtering (MCA-CT), coherence enhancing diffusion (CDF), shock filtering (SHK) and "texture characteristic" MCA filtering with proposed manipulation method (WT-TC-MCA). Only the proposed method increases the Mahalanobis distance between all clusters of textures descriptions which is the main mechanism behind the textural differences enlargement proposed in Chapter 6.1. Each D_i is computed using Eq. 6.2.

characteristics: the differences of coarseness, contrast and line-likeness are all magnified, while the differences of directionality are not changed much since the original differences between two regions are too small to discriminate. The Mahalanobis distance differences between two textural regions enhanced by the proposed method are increased more than the comparators, demonstrating its ability in enhancing the textural differences in the image.

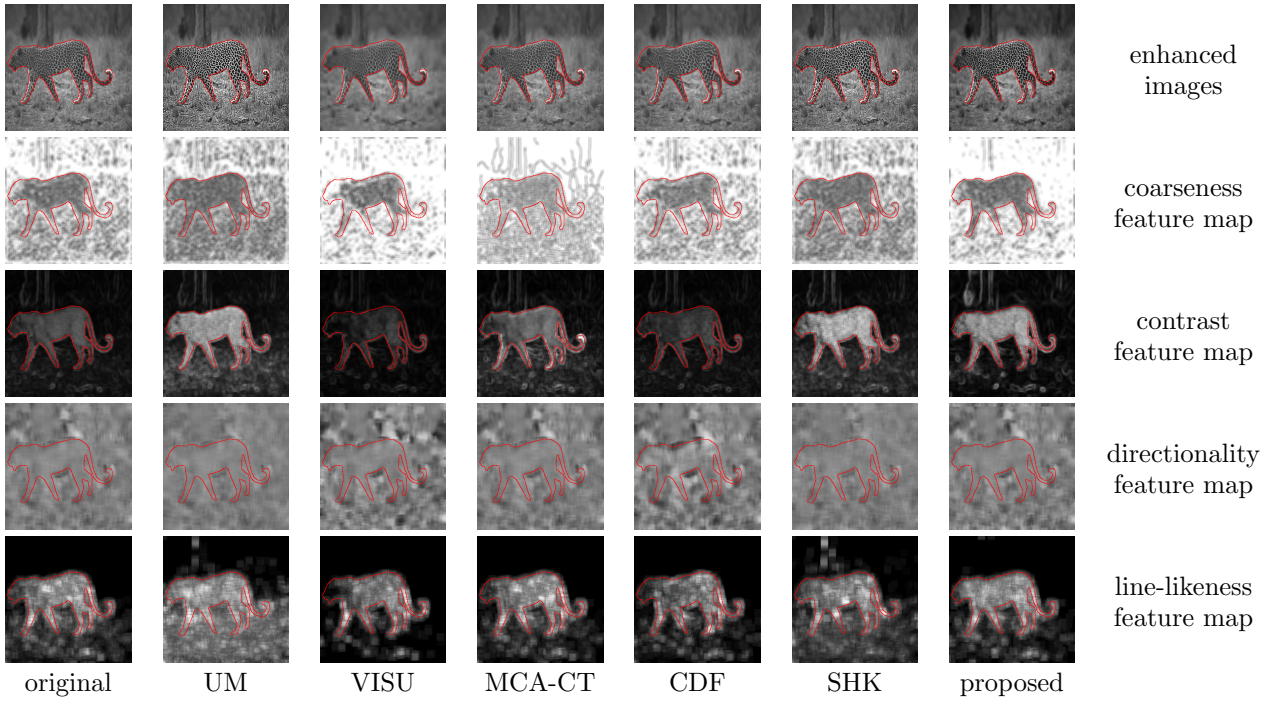


Figure 6.9: Comparison of the performances of different image enhancement methods in magnifying the textural differences in natural image more distinct with respect to the texture characteristics. Columns: the original image and images enhanced by UM[158], VISU[47], traditional MCA[189], SHK[210], CDF[209] and the proposed WT-TC-MCA. Rows: the maps of coarseness, contrast, directionality and line-likeness of the original and enhanced images. The red boundary is the ground truth boundary between the foreground object and the background.

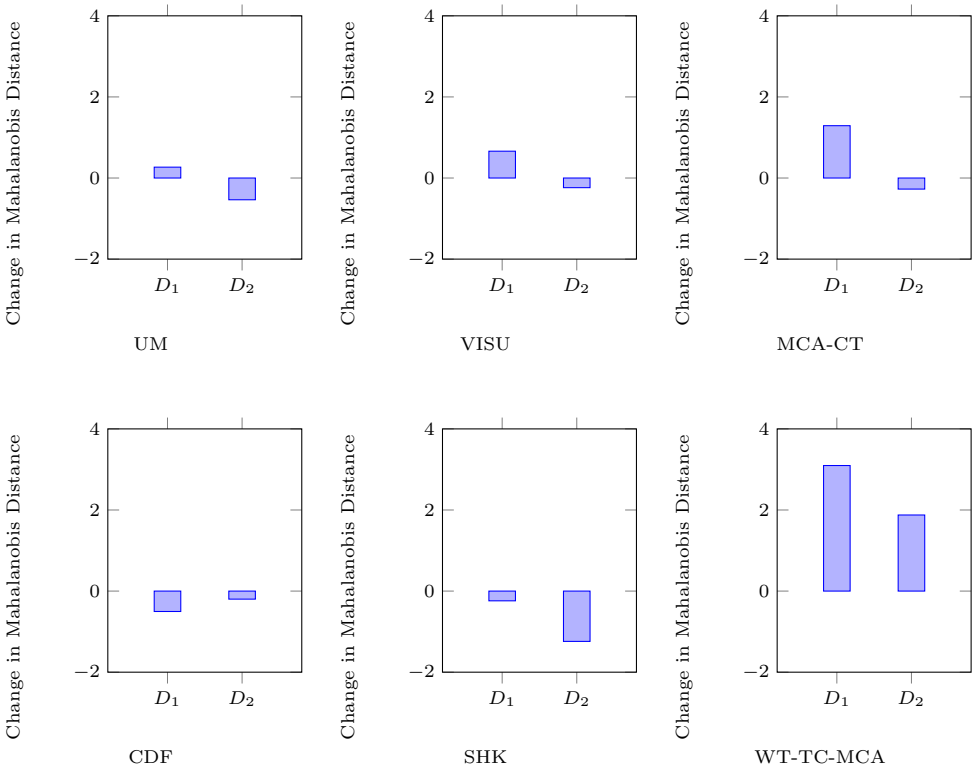


Figure 6.10: The changes in the twelve Mahalanobis distances between clusters of local texture characteristic vectors in the image in Figure 6.9 after processing with unsharp masking (UM), VISUShrink (VISU), MCA-CT filtering (MCA-CT), coherence enhancing diffusion (CDF), shock filtering (SHK) and "texture characteristic" MCA filtering with proposed manipulation method (WT-TC-MCA). Only the proposed method increases the Mahalanobis distance between all clusters of textures descriptions which is the main mechanism behind the textural differences enlargement proposed in Chapter 6.1. Each D_i is computed using Eq. 6.2.

.

6.2 WT-TC-MCA as pre-processing method prior to grayscale image segmentation

Since the WT-TC-MCA can enhance the textural differences in the image, the performance of texture-based image segmentation, which is to partition the image into regions exhibiting similar texture features within them and different texture features between regions, can be enhanced by using the image enhanced by WT-TC-MCA method to preprocess images to be segmented.

6.2.1 The algorithm of the WT-TC-MCA image enhancement as the pre-processing of image segmentation

Fig. 6.11 shows the schematic of applying WT-TC-MCA to enhance the gray scale image. Each step functions as follows:

1. the local Tamura's features are extracted from the 7×7 neighbourhood of every pixel in the input image I;
2. the PCA-based feature selection method in Chapter 3 is applied to the feature vectors of all the pixels to choose the key characteristics to describe the differences between different textures in the image;
3. based on the selected texture feature, the WT-TC-MCA algorithm is applied to the image to enhance the textural differences with respect to the corresponding texture characteristics: 1) the image is decomposed to components representing the Tamura texture characteristics, 2) each components is manipulated to enhance its own characteristic by the corresponding wavelet-based manipulation functions, and 3) the components are re-combined to get the enhanced image where the textural differences are magnified with respect to the selected texture characteristics.

To be more intuitive, an example of enhancing a gray scale image is shown in Fig. 6.12. The coarseness, contrast, directionality and line-likeness are calculated locally for the input image. After the use of PCA-based feature selection, coarseness and contrast are chosen as the key characteristics in describing the image. Row 2 of Fig. 6.12 show the two pairs of components representing coarse, fine, high-contrast and low-contrast respectively. Then the components are manipulated as shown in the Row 3 of Fig. 6.12. By combining these enhanced components, the image is enhanced where the differences between different regions with respect to coarseness and contrast are getting larger. The maps of coarseness and contrast of the input image and output image are shown in Fig. 6.13.

6.2.2 Experiments and analysis

The performance of the proposed grayscale image enhancement method in improving grayscale image segmentation algorithms is tested and evaluated in this chapter.

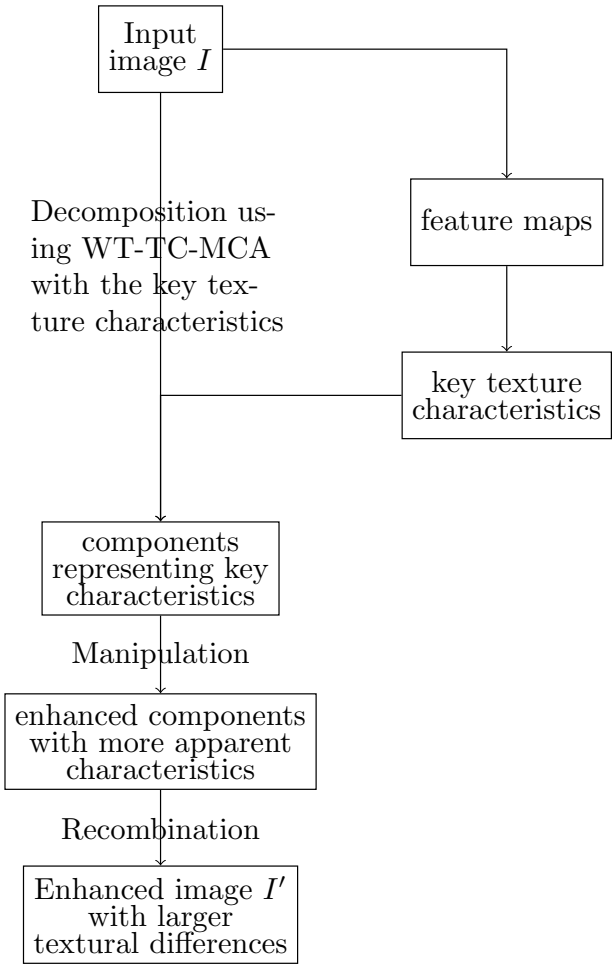


Figure 6.11: The process of pre-processing the image for improving the image segmentation. The most distinctive texture characteristics are selected by the PCA-based feature selection method. The image is enhanced by the WT-TC-MCA to enlarge the textural differences in the image with respect to these certain texture characteristics.

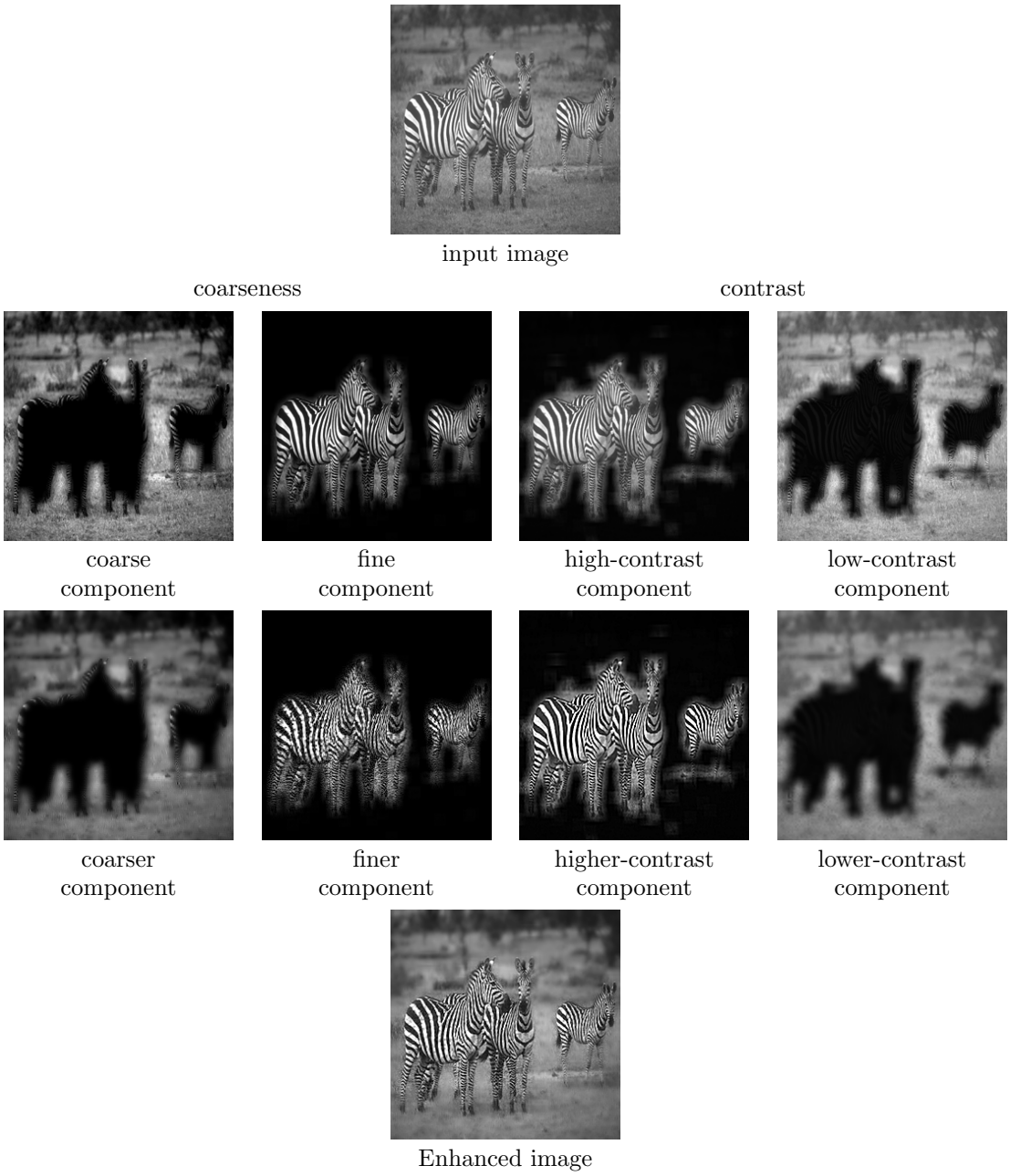


Figure 6.12: The enhancement of an example natural image as the pre-processing step prior to the segmentation of the image. The image is decomposed corresponding to coarseness and contrast, which are selected by the PCA feature selection method. Then the components representing coarseness and contrast strongly and weakly are manipulated so that the corresponding characteristics are more accentuated. The enhanced image, as the re-combination of the enhanced components, have the textures more distinguishable so that the texture-based segmentation could be improved.

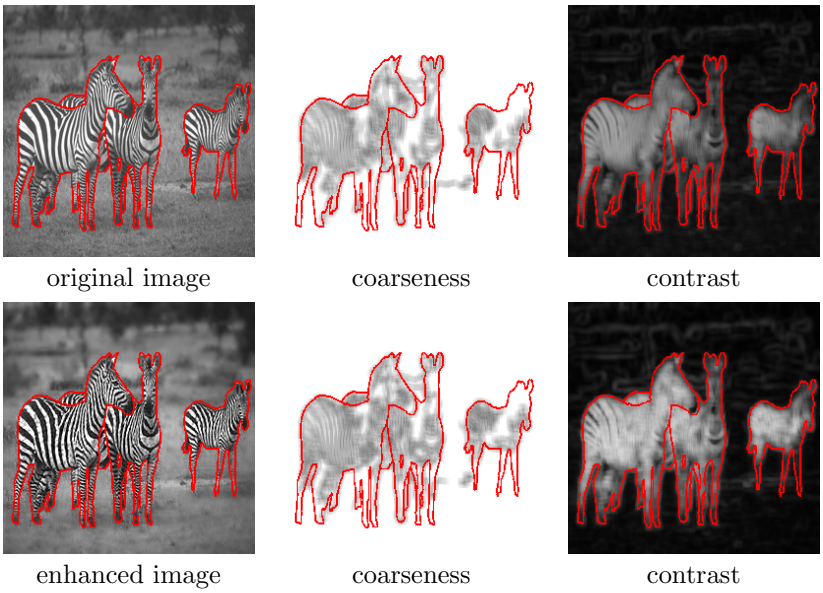


Figure 6.13: The coarseness and contrast of the original natural image and the image enhanced by the WT-TC-MCA method. Two texture regions (inside and outside the red boundary) are enhanced with larger differences with respect to both of the two texture characteristics. The red boundary is the ground truth boundary between the foreground object and the background.

The test images include the synthetic texture images that are synthesized by combining textures from the Brodatz texture database [24] and SIPI texture database [208], and the natural images that are from the BSDS500 dataset [135]. The same texture enhancement methods in Chapter 6.6.1, including unsharp masking (UM) [158], wavelet VISUShrink (VISU) [47], coherence-enhancing diffusion filter (CDF) [209], shock filter (SHK) [210], standard MCA filter (MCA-CT), are used as comparators in the experiments. Then the image segmentation tests are carried out as follows:

1. the test images are enhanced as described in Chapter 6.2.1 as well as with each of the comparator methods;

2. the mosaic texture images are segmented by Tamura-feature-based segmentation [201], LBP-based segmentation [137] and Factorization-based segmentation [220]; and the natural images are segmented by the active contour segmentation [38], the diffusion-based feature-based segmentation [167] and the texture and boundary compression segmentation [142] methods;

3. the segmentation effects of the images before and after enhancement are compared in terms of the precision, recall and accuracy as defined in [224]:

$$\begin{array}{c} \mathrm{Pre} = (\sum_{i=1}^{n} |S_i \cap R_{imax}| \,/\, \sum_{i=1}^{n} |S_i|) \times 100\% \\ \mathrm{Rec} = (\sum_{i=1}^{m} |R_i \cap S_{imax}| \,/\, \sum_{i=1}^{m} |R_i|) \times 100\%, \\ \mathrm{Acc} = (\mathrm{Pre} + \mathrm{Rec})/2 \end{array} \tag{6.3}$$

where Pre, Rec and Acc are the precision, recall and accuracy respectively, $|\cdot|$ denotes the area represented by the number of pixels in a region, R_{imax} is the matched reference object for each segment S_i

while S_{imax} is the matched segment for each reference object R_i;

4. the average precision and recall of segmenting all the enhanced images by a certain algorithm with each given parameter are computed as:

$$\begin{aligned} \overline{\mathrm{Pre}_\delta} &= \sum_{j=1}^{N} \mathrm{Pre}_{j,\delta}/N \\ \overline{\mathrm{Rec}_\delta} &= \sum_{j=1}^{N} \mathrm{Rec}_{j,\delta}/N \end{aligned}, \tag{6.4}$$

where δ denotes the given parameter for the certain segmentation algorithm, j is the index of the image in the database and N is the total number of the images in the database. With changing the parameters, the precision-recall curves are drawn to evaluate the general performance of different image enhancement methods in improving the given segmentation algorithm.

Evaluation of improving the segmentation of mosaic texture images

The results of enhancing a mosaic texture image by different enhancement methods and segmenting the enhanced images by different texture-based segmentation algorithms are shown in Fig. 6.14.

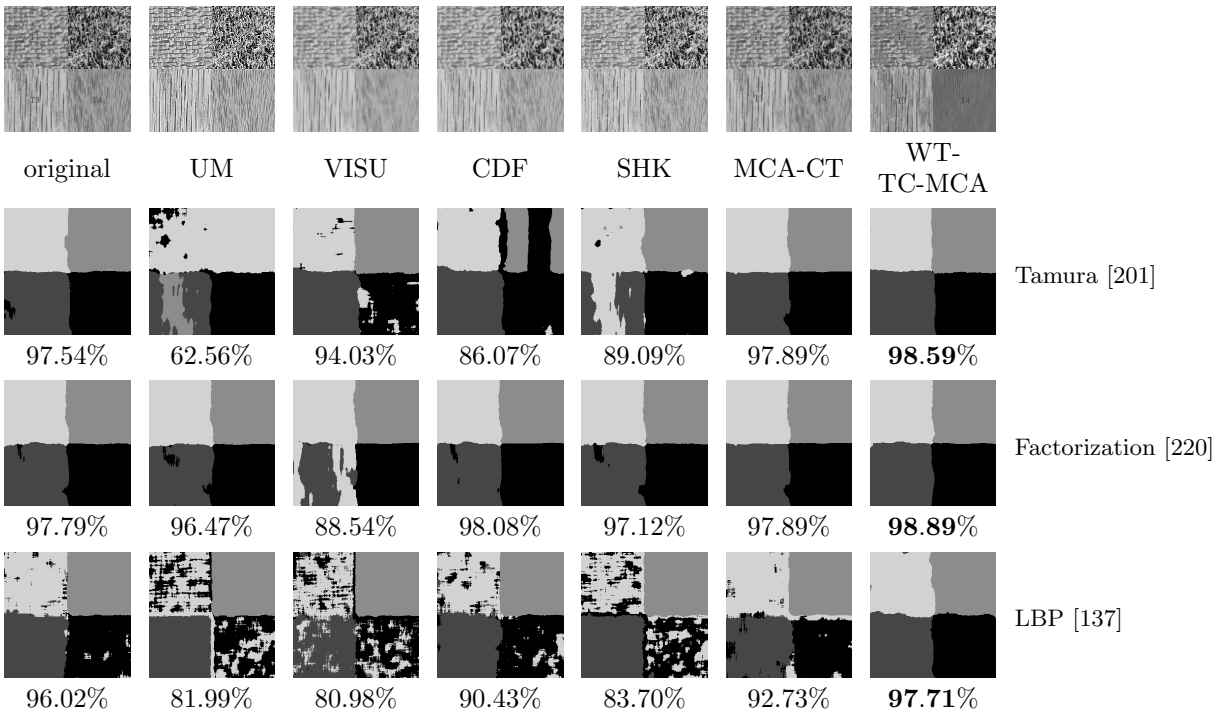


Figure 6.14: Comparison of the performances of different image enhancement methods in improving the effect of segmenting the example mosaic texture image by various texture-based segmentation algorithms. Columns: the original image and images enhanced by UM[158], VISU[47], SHK[210], CDF[209], "cartoon+texture" MCA filtering (MCA-CT)[189] and the proposed WT-TC-MCA. Rows: the effects of segmenting the images by Tamura-feature-based segmentation [201], LBP-based segmentation [137] and Factorization-based segmentation [220], together with the segmentation accuracies defined in Eq. 6.3.

As discussed in Chapter 6.1.4, the WT-TC-MCA can enlarge the textural differences in the image with respect to the certain texture characteristics more than other conventional texture enhancement methods

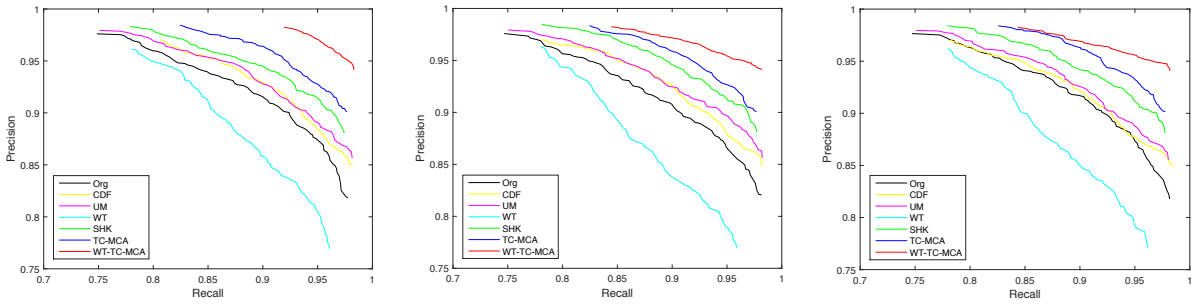
because of its ability in extracting the components representing different texture characteristics and manipulating these components to accentuate the corresponding texture characteristics independently. The texture-based segmentation algorithms are therefore improved by using the image enhanced by WT-TC-MCA more than using the image enhanced by the other methods since these segmentation algorithms are equal to label the regions with different texture features. The mean accuracies and standard deviations of segmenting 200 mosaic images and the enhanced ones by different segmentation methods are shown in Table 6.1. The accuracies from segmenting the images with each segmentation algorithm and enhancement method (including no enhancement) were subjected to a Lilliefors test. For all algorithms we were unable to reject the null hypothesis that the segmentation accuracies were normally distributed ($p > 0.05$). For each segmentation algorithm and enhancement method we performed a student's T-test [81] on the accuracies obtained for the original images and on the accuracies obtained for the enhanced images. In all case the null hypothesis was rejected ($p < 10^{-4}$), and the observed differences in mean accuracy before and after enhancement are statistically significant. Fig. 6.15 shows the precision-recall curves for different segmentation methods over the mosaic images enhanced by the comparator methods, WT-TC-MCA provides greater precision for almost all values of recall.

	Tamura			Factorization			LBP		
	Acc (%)	P1	P2	Acc (%)	P1	P2	Acc (%)	P1	P2
original	92.82 ± 1.11	0.2694	-	92.37 ± 1.27	0.1492	-	84.67 ± 1.21	0.0989	-
UM	92.67 ± 1.46	0.1504	$5.08e^{-24}$	93.79 ± 1.38	0.2389	$5.69e^{-30}$	87.20 ± 1.40	0.0940	$5.91e^{-16}$
VISU	92.07 ± 0.30	0.4322	$2.57e^{-34}$	93.07 ± 1.08	0.0681	$1.92e^{-40}$	84.61 ± 0.39	0.1662	$6.60e^{-44}$
CDF	91.42 ± 1.08	0.3415	$1.24e^{-5}$	94.54 ± 1.27	0.1080	$9.80e^{-11}$	85.81 ± 1.19	0.2597	$1.29e^{-5}$
SHK	94.02 ± 1.44	0.1292	$7.19e^{-28}$	95.81 ± 1.04	0.2757	$1.03e^{-29}$	87.30 ± 1.39	0.3266	$2.14e^{-23}$
MCA-CT	94.53 ± 1.03	0.4769	$9.13e^{-27}$	94.75 ± 1.07	0.3323	$1.35e^{-28}$	90.79 ± 1.08	0.1288	$6.06e^{-25}$
WT-TC-MCA	$\mathbf{95.53 \pm 0.24}$	0.2978	$3.14e^{-30}$	$\mathbf{97.36 \pm 0.95}$	0.4312	$2.08e^{-25}$	$\mathbf{92.12 \pm 0.96}$	0.2553	$9.36e^{-23}$

Table 6.1: Mean accuracy and standard deviation of segmenting 200 mosaic images and the enhanced ones by different texture-based segmentation methods. For each enhancement method, P1 is the P-value of a Lilliefors test ($P < 0.05$) [81] for each group of segmentation accuracies, and P2 is the P-value of doing Student's t-test ($P < 0.05$) [81] for the groups of the segmentation accuracies of the original images and the enhanced images. The differences in the means, though, in some cases, small, are statistically significant ($P2 < 10^{-4}$).

Evaluation of improving the segmentation of natural images

Fig. 6.16 shows the enhancement and segmentation of the natural image after texture enhancement by the comparison methods. Table 6.2 shows the mean accuracies and standard deviations of segmenting 200 natural images and the enhanced ones by different segmentation algorithms, together with the p-values from the Student's t-test ($P < 0.05$) between the accuracies of the original images and the enhanced images. Similar as the results of mosaic images, the WT-TC-MCA provides larger, statistically significant ($P2 < 0.05$) improvement of accuracy of segmenting the enhanced images by all the segmentation algorithms. With the changes of parameters, the WT-TC-MCA stably improves the general performance of each segmentation



(a) Tamura (b) Factorization (c) LBP

Figure 6.15: The precision-recall curves of segmenting synthetic texture images enhanced by different methods using different segmentation algorithms. (a) Tamura-feature-based K-means segmentation, with the parameter window size $w = [5 \rightarrow 20]$. (b) Factorization based texture segmentation, with the parameters spatial frequency band-width $b = [0.4 \rightarrow 2.5]$ and spatial aspect ratio $\gamma = [0.2 \rightarrow 1]$. (c) LBP based texture segmentation method, with the parameter window size $w = [5 \rightarrow 20]$. WT-TC-MCA method, represented by the red curve, results in higher precision for most values of recall.

algorithm as the precision-recall curves shown in Fig. 6.17.

Similarly to the results for mosaic texture images, the WT-TC-MCA provides larger improvement of accuracy in segmenting the enhanced natural images by all the segmentation algorithms, and better performance of segmentation over all the natural images in the database in terms of precision and recall. The average accuracies of segmenting 200 natural images and the enhancing ones by different segmentation methods are shown in Table. 6.2.

	AC			Diffusion-based feature space			TBES		
	Acc (%)	P1	P2	Acc (%)	P1	P2	Acc (%)	P1	P2
original	87.83 ± 3.21	0.2517	-	76.99 ± 2.65	0.1561	-	87.16 ± 0.64	0.0884	-
UM	87.54 ± 4.36	0.3664	$9.75e^{-8}$	79.05 ± 3.58	0.1788	$6.79e^{-11}$	87.82 ± 1.23	0.1811	$2.58e^{-16}$
VISU	87.47 ± 2.51	0.3953	$1.27e^{-6}$	78.93 ± 2.15	0.3688	$1.08e^{-20}$	87.10 ± 1.09	0.0758	$5.30e^{-8}$
CDF	87.83 ± 3.09	0.3490	$2.00e^{-13}$	78.58 ± 2.45	0.2096	$6.57e^{-16}$	87.16 ± 0.75	0.1044	$5.15e^{-16}$
SHK	88.35 ± 4.03	0.2267	$3.15e^{-6}$	79.53 ± 4.04	0.2569	$5.26e^{-5}$	87.40 ± 1.24	0.2584	$4.38e^{-7}$
MCA-CT	89.01 ± 4.27	0.4373	$3.05e^{-16}$	82.16 ± 4.13	0.2767	$9.17e^{-14}$	88.63 ± 1.66	0.3601	$3.35e^{-25}$
WT-TC-MCA	$\mathbf{90.55 \pm 3.36}$	0.4305	$2.95e^{-19}$	$\mathbf{87.68 \pm 3.08}$	0.2493	$1.09e^{-14}$	$\mathbf{91.87 \pm 1.67}$	0.4022	$1.02e^{-26}$

Table 6.2: Mean accuracy and standard deviation of segmenting 200 natural images and the enhanced ones by different texture-based segmentation methods. For each enhancement method, P1 is the P-value of a Lilliefors test ($P < 0.05$) [81] for each group of segmentation accuracies, and P2 is the P-value for the Student's t-test ($P < 0.05$) [81] for the groups of the segmentation accuracies of the original images and the enhanced images. WT-TC-MCA outperforms other enhancement methods in increasing the accuracy for every testing segmentation method. The differences in means are all statistically significant ($P2 < 10^{-4}$).

According to the above experimental results, we conclude that use of the WT-TC-MCA method as a preprocessing step prior to the grayscale image segmentation can improve the performance of these segmentation algorithms, and likely others as well, because the WT-TC-MCA can enhance the textural differences in the image with respect to the key texture characteristics differentiating the different texture regions.

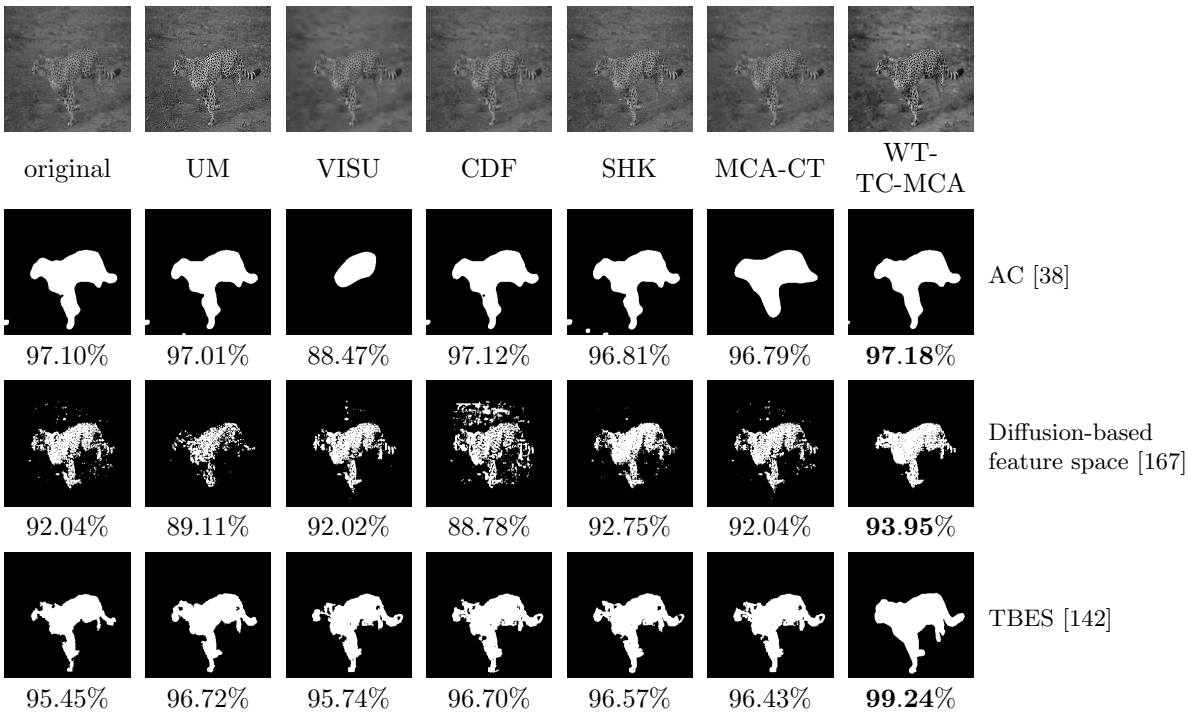
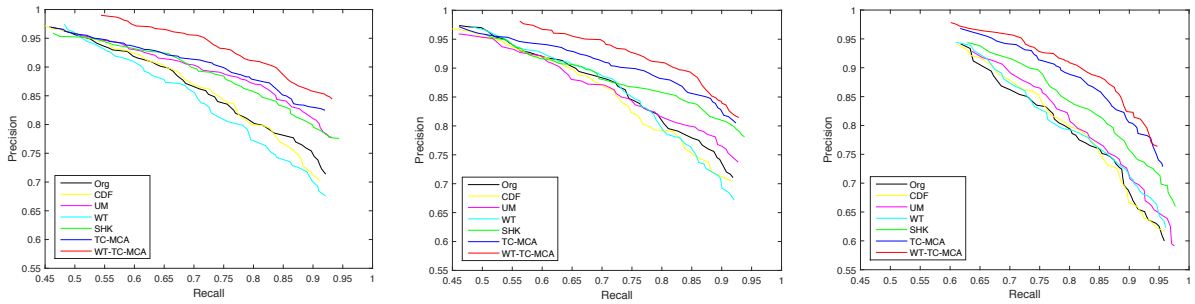


Figure 6.16: Comparison of the performances of different image enhancement methods in improving the effect of segmenting the example natural image by various texture-based segmentation algorithms. Columns: the original image and the enhanced ones by UM[158], VISU[47], SHK[210], CDF[209], "cartoon+texture" MCA filtering (MCA-CT)[189] and the proposed WT-TC-MCA. Rows: the effects of segmenting the images by the active contour segmentation [38], the diffusion based feature based segmentation [167] and the texture and boundary compression segmentation [142], together with the segmentation accuracies defined in Eq. 6.3.



(a) AC (b) Diffusion feature space (c) TBES

Figure 6.17: The precision-recall curves of segmenting natural texture images enhanced by different methods using different segmentation algorithms. (a) Active contour segmentation, with the parameters: $\mu = [1e1, 1e2, 1e3, 1e4]$ and patch size $w = [5 \rightarrow 15]$. (b) Diffusion feature space based segmentation, with the parameters: regularization weight $v = [0.1 \rightarrow 1]$ and parzen window size $\sigma = [1 \rightarrow 10]$. (c) Texture and boundary compression segmentation, with the parameters: distortion level $\epsilon = [25 \rightarrow 400]$ and max window size $w_M = [5 \rightarrow 20]$. WT-TC-MCA method, represented by the red curve, results in higher precision for most values of recall.

6.3 WT-TC-MCA as pre-processing method prior to colour image segmentation

Since the WT-TC-MCA can enhance the textural differences in grayscale images, it is available to enhance the colour image as an extension of grayscale image enhancement. In this chapter, the colour image enhancement used WT-TC-MCA is proposed as the pre-precessing of the colour image segmentation algorithms.

6.3.1 The algorithm of the WT-TC-MCA colour image enhancement

The WT-TC-MCA texture enhancement method is applied to the luminance component from the CIELab colour space. The CIELab colour space transforms the RGB colour space into a luminance channel L and two opponent chrominance channels a and b [14]. By enhancing the L component of the colour image with WT-TC-MCA, the textures are differentiated more whereas chrominance values remain unchanged. As shown in Fig. 6.18, the WT-TC-MCA based colour image enhancement is implemented as follows:

1. transform the input image I from RGB colour space to CIELab colour space;
2. enhance the L component by the WT-TC-MCA method so that texture differences in the L channel are modified to be mutually more different to obtain the enhanced component L';
3. replace the L component with $L^{'}$, then transform the colour image back to the RGB colour space, yielding the texture-enhanced colour image $I^{'}$.

6.3.2 Experimental results and analysis

The colour image enhancement method discussed above is evaluated by analyzing the performance improvement of colour image segmentation algorithms, because the main purpose of the image enhancement is to differentiate the texture details in the colour image so as to enhance the effect of colour image segmentation algorithms.

Experimental materials

The colour images utilized in this part are from the Berkerly segmentation dataset [135]. The performance of the proposed method is compared with the state-of-the-art colour image enhancement methods as follows:

1. adaptive neighbourhood histogram equalization (ANHE) [27], which equalizes the intensities in a variable-shaped neighbourhood containing pixels similar to the seed pixels and updates the seed pixels' intensities with the equalized ones;
2. J. Han, S. Yang and B. Lee's work (HP-ILP) [88], which defines the histogram of a colour image whose cumulative distribution function (cdf) is the accumulation of probability distribution functions (pdfs)

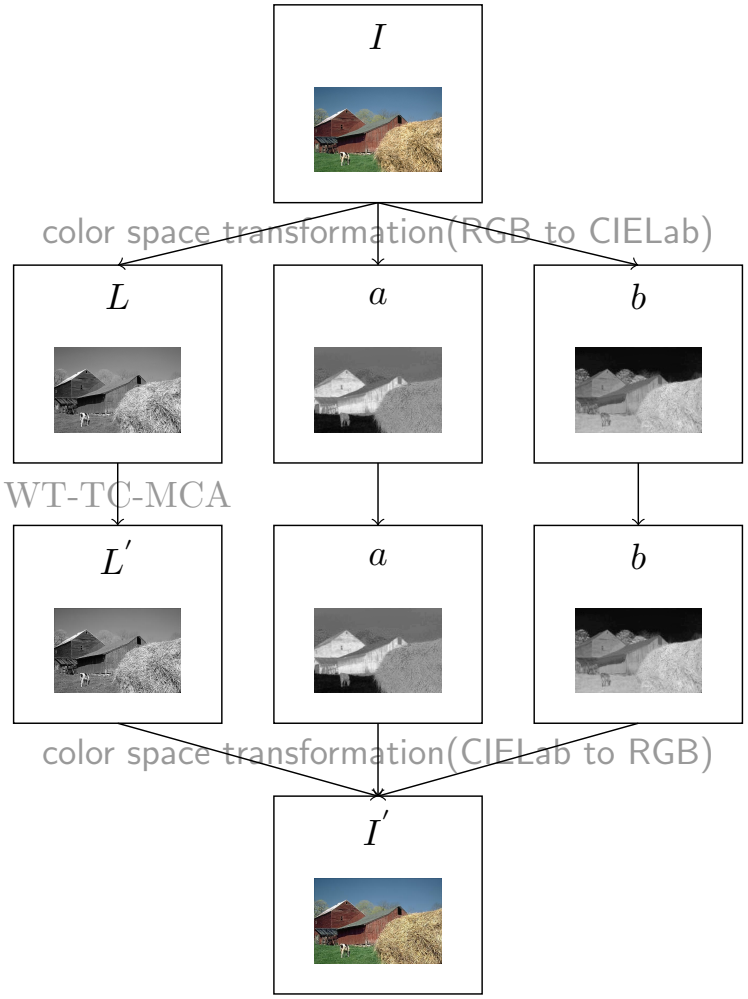


Figure 6.18: The WT-TC-MCA based colour image enhancement process. The input image I is transformed to luminance component L and chrominance components a and b. The luminance component L is enhanced by WT-TC-MCA method to $L^{'}$ while the chrominance components a and b are kept unchanged. By transforming the colour image back to RGB colour space, the image $I^{'}$ is enhanced with more differentiable textures.

within the box of size $R \times G \times B$ in 3D colour space. Then the image is enhanced by equalizing this 3D colour histogram with the method proposed in [146] preserving both the hue and the gamut constraints;

3. multi-scale retinex with colour restoration (MSRCR) [89], which is based on the retinex image enhancement concept that enhances the image by processing the incident light and the reflectance of the object separately;

4. spatial entropy-based contrast enhancement using discrete cosine transform (SECE-DCT) [36], where DCT transform domain coefficients of the image globally enhanced by SECE is further weighted to obtain both globally and locally enhanced image;

5. luminance and contrast masking of human visual system based image enhancement (HVS-SWT-LCM) [147], which combines a multi-resolution transform with luminance masking and contrast masking based on human visual system. It is capable of adjusting the brightness level of the image and providing both dynamic range compression and contrast enhancement;

6. histogram equalization and unsharp masking (HE-UM) [42], which applies both histogram equalization and unsharp masking filter to the luminance channel of the image;

7. coherence enhancing diffusion filter (CDF) [209], which applies the CDF texture enhancement filter to enhance the luminance of the colour image;

8. shock filtering (SHK) [210], which enhances the luminance channel of the colour image as the grayscale texture image.

With the above comparator methods, the segmentation tests are carried out as follows:

1. the test images are enhanced as described in Chapter 3, and with the comparator methods listed above;
2. the original images and the images enhanced by the proposed method were segmented using several segmentation algorithms: gPb-owt-ucm [9], UCM [8], Mean Shift [51], N-cuts [54], region merging [73] and Canny [32], and evaluated using the BSDS500 benchmark. All the BSDS500 benchmarks are calculated as the average values of comparing the segmentation results to the ground truths drawn by different human subjects;
3. the original images and the images enhanced by different image enhancement methods were segmented with the hierarchical segmentation algorithm gPb-owt-ucm [9] and evaluated using the BSDS500 benchmark.

Performance of enhanced colour image segmentation

Fig. 6.19 shows an example image enhanced by different colour image enhancement methods, together with the gPb-owt-ucm segmentation results, which performs the best over all images in the dataset. The AHNE

method stretches the brightness too much and distorts the colour, leading over-enhancement of weak edges. The HP-ILP method focuses on local textures but produces artifacts in smooth regions. MSRCR and HVS-SWT-LCM methods both make the images too bright and distort the colour in the dark areas, adding a lot of unwanted textures. SECE-DCT degrades the textures in bright regions. HE-UM enhances the edges globally however degrades the textures with similar local intensities. CDF and SHK both change the shapes of textures in the images, which cannot highlight textures in the regions of interest either. The proposed method leads to better segmentation results because WT-TC-MCA can enhance textures to different extents with respect to their own properties because it separates the textures into components representing different visual characteristics and modifies these components in different ways.

Fig. 6.20 shows the evaluation of segmentation algorithms on the BSDS500 images and those enhanced by the proposed method. Table 6.3 shows the F-measures when choosing an optimal scale for the entire dataset (ODS) or per image (OIS), as well as the average precision (AP). Use of the proposed method prior to segmentation improves the performance of every segmentation method.

	original			WT-TC-MCA		
	ODS	OIS	AP	ODS	OIS	AP
gPb-owt-ucm[9]	0.69	0.72	0.70	**0.73**	**0.75**	**0.76**
UCM[8]	0.66	0.68	0.65	**0.68**	**0.69**	**0.68**
Mean Shift[51]	0.62	0.64	0.58	**0.66**	**0.68**	**0.63**
N-cuts[54]	0.60	0.64	0.54	**0.63**	**0.68**	**0.58**
region merging[73]	0.56	0.59	0.48	**0.60**	**0.62**	**0.56**
Canny[32]	0.54	0.57	0.43	**0.56**	**0.57**	**0.51**

Table 6.3: The F-measure of segmenting BSDS500 images and images enhanced by the proposed method with different segmentation methods.

Then we selected the gPb-owt-ucm segmentation method, which has the best performance in segmenting colour images in the dataset. Fig. 6.21 shows the evaluation of gPb-owt-ucm in segmenting images enhanced by different enhancing methods. Table 6.4 shows the F-measure of different enhanced images in the testing dataset. Use of the proposed method prior to segmentation leads to a better segmentation effect than other enhancement methods.

	F-measure				F-measure		
	ODS	OIS	AP		ODS	OIS	AP
original	0.69	0.72	0.71	HVS-SWT-LCM[147]	0.71	0.74	0.75
ANHE[27]	0.69	0.71	0.69	HE-UM[42]	0.68	0.71	0.71
HP-ILP[88]	0.71	0.74	0.74	CDF[209]	0.72	0.74	0.75
MSRCR[89]	0.70	0.72	0.71	SHK[210]	0.71	0.72	0.73
SECE-DCT[36]	0.71	0.74	0.75	WT-TC-MCA	**0.73**	**0.75**	**0.76**

Table 6.4: The F-measure of segmenting BSDS500 images and the images enhanced by different methods using the gPb-owt-ucm segmentation method.

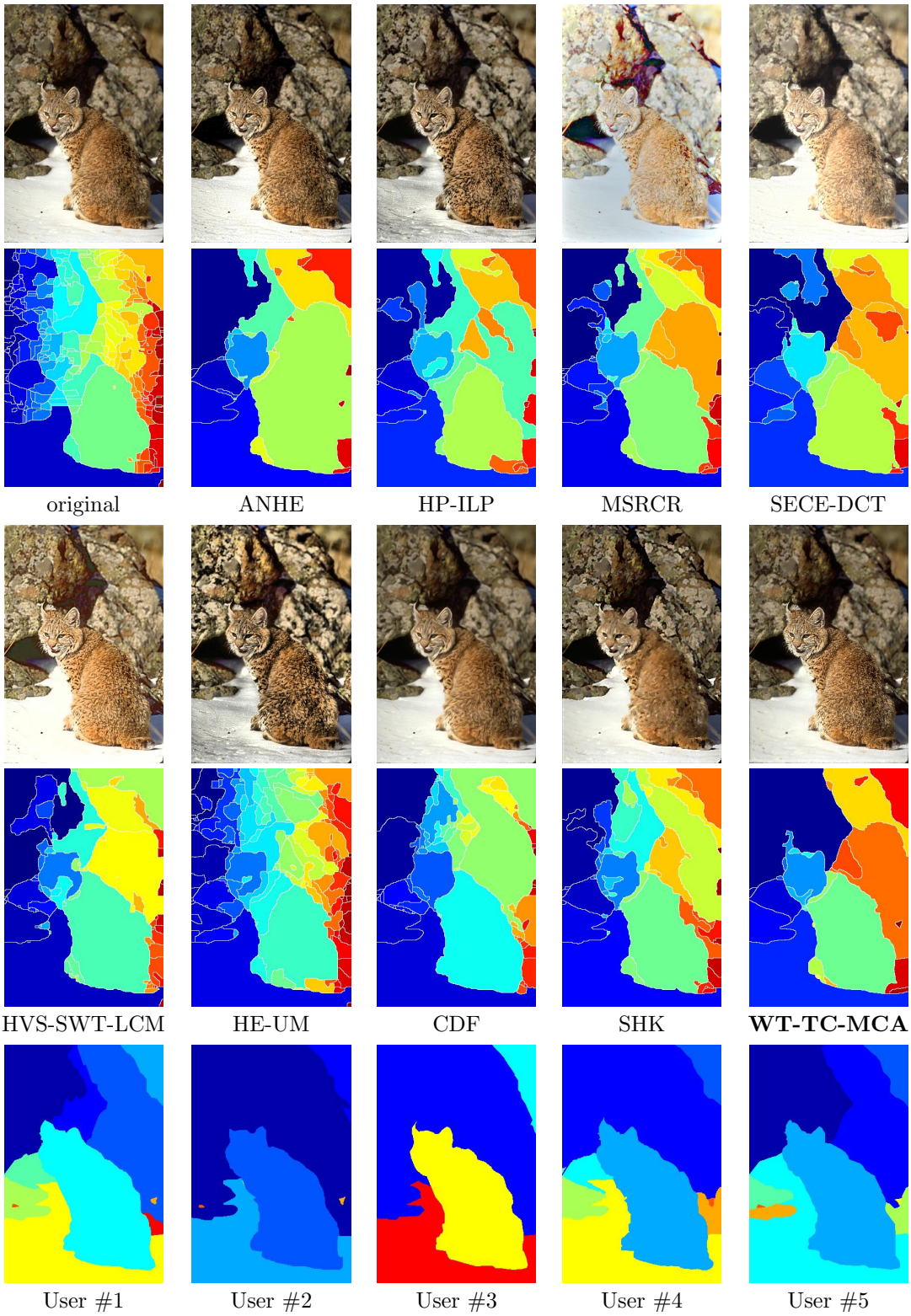
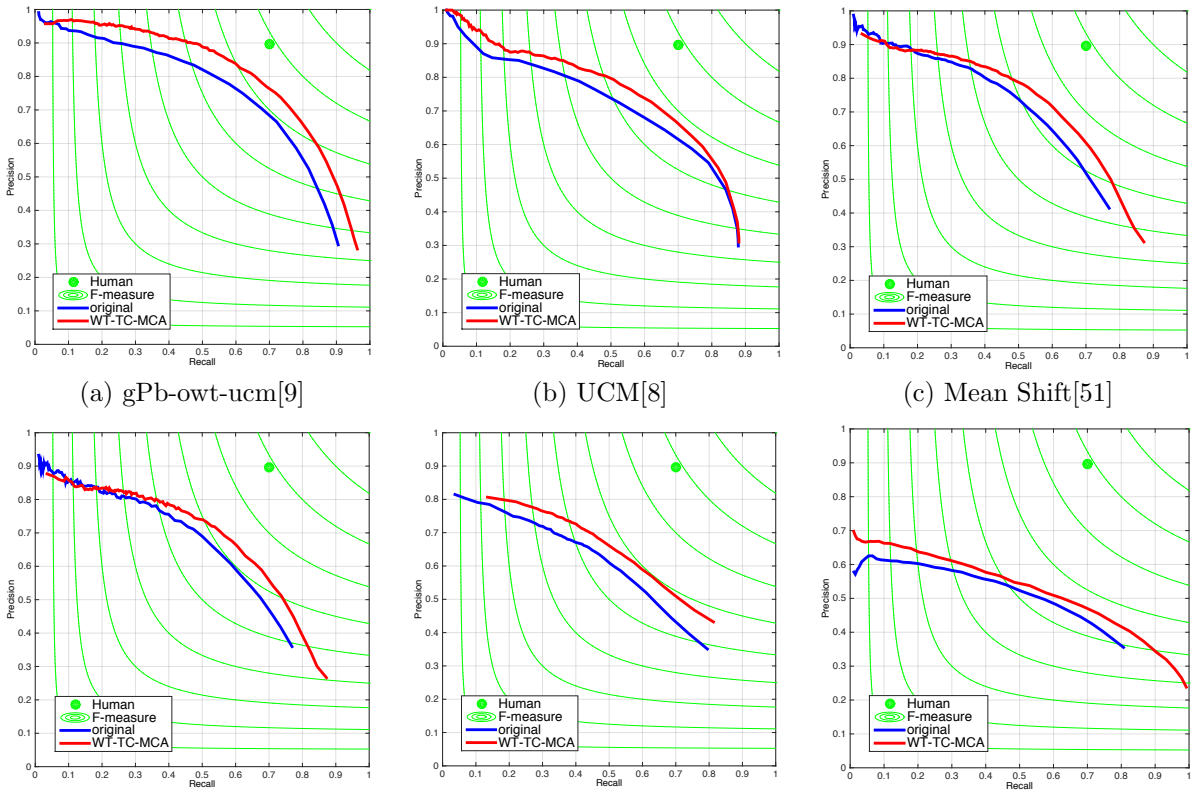


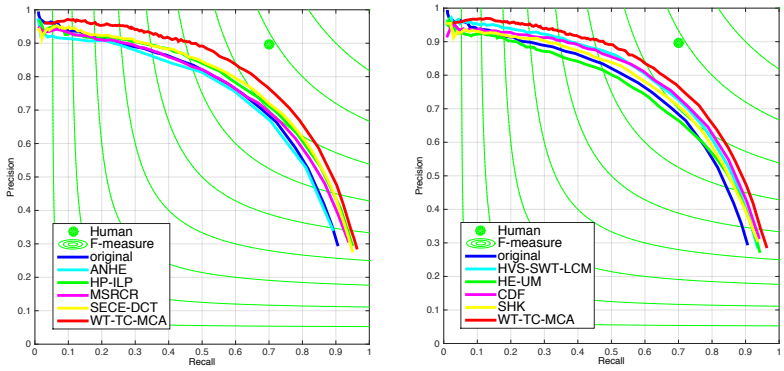
Figure 6.19: Comparison of the performances of different colour image enhancement methods in improving the effect of segmenting the example colour image by gPb-owt-ucm. Row 1 and Row 3: colour image enhanced by ANHE [27], HP-ILP [88], MSRCR [89], SECE-DCT [36], HVS-SWT-LCM [147], HE-UM [42], CDF [209], SHK [210] and WT-TC-MCA. Row 2 and Row 4: the segmentation of the enhanced images by gPb-owt-ucm. Row 5: the ground truths of segmenting the colour image by 5 human users.



(a) gPb-owt-ucm[9] (b) UCM[8] (c) Mean Shift[51]

(d) N-cuts[54] (e) region merging[73] (f) Canny[32]

Figure 6.20: Evaluation of different segmentation algorithms on the BSDS500 images and those enhanced by the proposed method. Blue curves are the precision-recall curves of segmenting the original images, red curves are the precision-recall curves of segmenting the enhanced images.



(a) (b)

Figure 6.21: Evaluation of the gPb-owt-ucm segmentation method on the BSD500 images and those enhanced by different image enhancment methods. (a) The comparison among the original images (blue), ANHE (cyan), HP-ILP (green), MSRCR (magenta), SECE-DCT (yellow) and the proposed method (red); (b) the comparison among the original images (blue), HVS-SWT-LCM (cyan), HE-UM (green), CDF (magenta), SHK (yellow) and the proposed method (red).

6.4 Conclusion

In this part, several applications based on the wavelet-based texture characteristics morphological component analysis (WT-TC-MCA) are proposed and discussed, including the textural difference enhancement method based on morphological component analysis, WT-TC-MCA grayscale image enhancement and WT-TC-MCA colour image enhancement method. The experimental results demonstrate that the textural difference enhancement method based on morphological component analysis can successfully enlarge the textural differences in the image because of the different texture characteristics are made more apparent, use of the WT-TC-MCA grayscale image enhancement and the WT-TC-MCA colour image enhancement as pre-processing step to image segmentation algorithms can effectively improve the performance of those segmentation algorithms by making the textures more differentiable.

Chapter 7

Conclusions

This thesis analyzed the method and application of enhancing textural differences in the image. To enlarge the textural differences, three main problems need to be solved:

1. Which characteristics should be used to describe the textural differences?
2. Can the image be considered as the composition of components representing the certain characteristics?
3. How can a certain component separated from the image be enhanced to accentuate the corresponding texture characteristic while not affecting other characteristics represented by other components?

Through this thesis, these three problems are solved by three different methods: local Tamura's human visual perceptual texture description (Chapter 3), wavelet-based texture characteristics morphological component analysis (Chapter 4), wavelet-based texture characteristics enhancement (Chapter 5). These separate modules are combined as a whole framework and the framework is applied as the image enhancement prior to grayscale and colour image segmentation algorithms (Chapter 6). The main contributions of this thesis are listed below.

7.1 Contributions

Reviews on image description, image decomposition and image manipulation methods. Statistical image description, image decomposition methods and image manipulation techniques have been exhaustively reviewed in Chapter 2. Mainstream image descriptors have been introduced and analyzed with their advantages and disadvantages. Human visual perceptual texture descriptors are highlighted as a method based on the relation between the texture and human observation. For the analysis of image decomposition methods, a novel classification of the image decomposition approaches is proposed. Image decomposition techniques can be classified by different assumptions of the input images, different assumptions of output results or different algorithmic approaches. Thus morphological component analysis is considered as the best method to find components representing texture characteristics in the image. While for image manipulation, both linear and non-linear, both de-noising and enhancing modification methods are introduced. As a result, non-linear image manipulation methods are used as the foundation for this thesis work.

Human visual perceptual texture description and PCA-based feature selection method. In Chapter 3, Tamura's texture features, including coarseness, contrast, directionality, line-likeness, regularity

and roughness, are novelly calculated to locally describe the neighbourhood centred at each pixel in the image. Then the principal component analysis concept is applied to find the key features that are necessary and sufficient enough to differentiate the textures or images in the dataset, or the various regions in the given textural image.

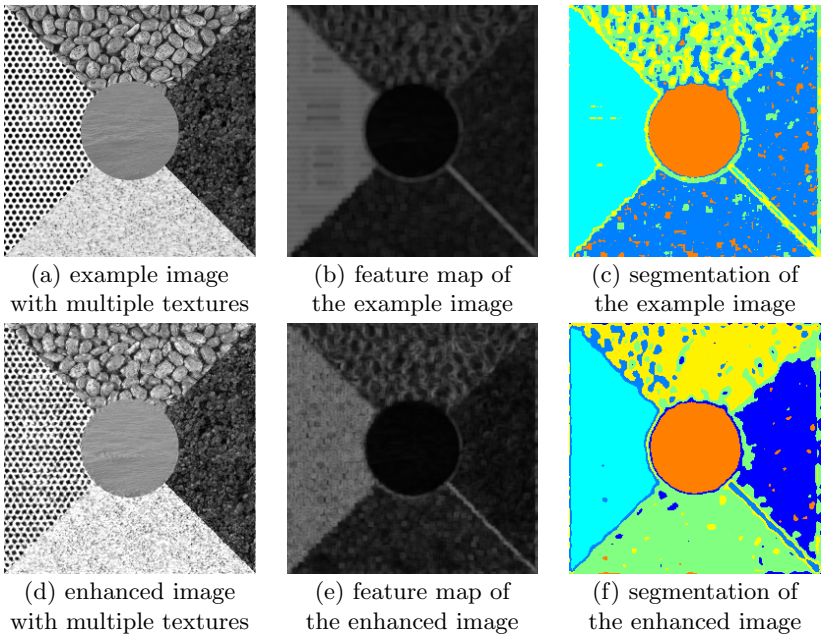
Texture characteristics morphological component analysis using wavelet-based dictionaries. In Chapter 4, a series of wavelet-based soft thresholding methods are proposed as the dictionaries to represent the certain texture characteristics. The thresholding methods are proposed pairwisely so that each of them exhibit one certain texture characteristics either strongly or weakly. The sensitivity and sparsity of each thresholding method as the dictionary to represent the certain texture characteristic are also demonstrated in this chapter.

Wavelet-based texture characteristics enhancement methods. In Chapter 5, various wavelet coefficients manipulation methods are proposed to accentuate the texture characteristics independently. Each non-linear wavelet coefficient manipulation function can enhance one texture characteristic, including coarse, fine, high-contrast, low-contrast, horizontal, vertical, line-like and non-line-like, while not affecting other texture characteristics of the image.

Textural differences enhancement by wavelet-based texture characteristics morphological component analysis. In Chapter 6, the three methods proposed in Chapter 3,4 and 5 are combined to one uniform framework as a novel method to enlarge the textural differences in the image. The key texture characteristics that differentiate different textures in the image are found first, then the image is decomposed by the morphological component analysis using wavelet-based dictionaries into components representing the key texture characteristics, then the components are enhanced by the wavelet coefficient manipulation functions to make the components exhibit the corresponding texture characteristics more apparently, after re-combining the manipulated components, the image is enhanced with the textures more differentiable with respect to the certain texture characteristics.

Applications of the textural differences enhancement method. In Chapter 6, the proposed textural differences enhancement method is used as the pre-processing steps of the grayscale and colour image segmentation algorithms respectively, so that the performance of these texture-based segmentation algorithms are improved more effectively than pre-processed by other image enhancement methods. When we look back at the segmentation problem mentioned in Chapter 1, Fig. 7.1 shows an example of how the image is enhanced by the WT-TC-MCA method and the segmentation is therefore improved. The textural differences enhancement method is adaptive in finding the key discriminant texture characteristics, decomposing the image into components exhibiting these characteristics strongly and weakly and accentuating the texture characteristics in the corresponding components, making no assumptions about the descriptors used for differentiating different textures. Therefore, the proposed textural differences enhancement method can be used as a general pre-processing step prior to any texture-based segmentation method. In theory, it should be able to improve the performance of any algorithm that relies on distinguishing regions of the image

based on a texture description.



(a) example image with multiple textures | (b) feature map of the example image | (c) segmentation of the example image

(d) enhanced image with multiple textures | (e) feature map of the enhanced image | (f) segmentation of the enhanced image

Figure 7.1: An example of texture-based segmentation improvement. (a) is the example image with five different Brodatz [24] textures in it, (b) is the map of local contrast description [184] of the example image, (c) is the segmentation of the example texture image by applying the k-means clustering [10] to the feature map, (d) is the image enhanced by the WT-TC-MCA method, (e) is the map of local contrast description of the enhanced image, (f) is the segmentation of the enhanced texture image by applying the k-means clustering to the feature map.

7.2 Future work

The design of the enhancement of textural differences in the image system involves the consideration of a wide range of questions. For the three steps proposed to compose the enhancement algorithm, in addition to the solutions which have been proposed and described in this thesis, a lot of ideas have the possibility to solve the description, decomposition and manipulation steps. On the other hand, the problem of textural differences enhancement itself is still a quite open problem, where many questions remain as undeveloped ideas that need to be further analyzed. We suggest all these undeveloped, possible ideas as our future work.

In order to conveniently organize these ideas for the future, we have divided them into two basic parts. The first one is composed of some possible improvements that can be analyzed in order to refine the proposed image enhancement method. The second part suggests some novel strategies that can be explored to enlarge the textural differences in the image.

7.2.1 Further work on the proposed enhancement method

Image description based on human visual perception. As has been denoted, there are different image

analysis techniques which describe the textural differences in images, especially differences with respect to human visual perception. The results of comparing these types of high-level texture features have been inconclusive and it is difficult to assess which is the best method. Therefore, other human visual perceptual features should be considered. Furthermore, the relationship between the texture features and human visual perception should be analyzed so that the differences with respect to these high-level texture features can reflect how a human observes different textures more precisely.

Dictionaries representing texture characteristics. As discussed in this thesis, the core problem of decomposition is to find dictionaries suitable for representing the certain types of image contents. The proposed method using wavelet-based dictionaries that apply different thresholding methods to remove the components not exhibiting the selected characteristics is considered as an indirect way to represent the corresponding components. Therefore, sparse transformations other than the wavelet transform should be considered as a more direct way to separate components with different texture characteristics from the image.

Non-linear texture characteristics enhancement methods. In this thesis, a series of functions are proposed to manipulate the wavelet coefficients to enhance certain texture characteristics of the image while not affecting other characteristics. As discussed in the literature review, wavelet-based image manipulation methods have some disadvantages in losing image information after transformation, or increasing noise or artifacts in the image. Therefore, other non-linear image enhancement methods should be studied so that texture characteristics can be enhanced by non-linear spatial filters independently. Moreover, dictionaries that can represent characteristics other than coarseness, contrast, directionality and line-likeness should be proposed to model the image more completely.

Applications using textural differences enhancement method. The proposed textural differences enhancement method has been used prior to grayscale and colour image segmentation algorithms. However, as one of the pre-processing steps, the ability of the proposed method in improving the performance of the image processing methods other than the image segmentation is not demonstrated. Thus, the enhancement should be applied to enhance the image and used to widen the range of image processing methods, including image classification, image retrieval, object detection or some others.

7.2.2 Further work on the textural differences enhancement

Methods to model the image. The human visual perceptual characteristics are not the only way to model different types of image content. It is possible to model the image as the linear or non-linear combination of some specific components representing the content features that can differentiate different textures in the image. For example, different shapes of textures can be used as the criteria so that the image can be considered as the combination of these components with different shapes of textures. Then the image can be enhanced by an enhancing schematic similar to that proposed in this thesis, but with different components.

Methods to enhance the image directly. The proposed image enhancement method manipulates the image quite indirectly, where it first finds the components of the image, then enhances the components and

finally re-combines the enhanced components to get the enhanced image. In other words, the proposed method considers the image as a function of components, then it manipulates the function output by manipulating the components. If there exists a method for considering the image as the input variable, the image could be manipulated with a function where the variables are the image data and the coefficients are calculated as the function of the variables themselves. Therefore, different image regions with different features would be manipulated differently, leading to magnified differences between different regions of the image. For example, the coherence-enhancing diffusion filter [209] and shock filter [210] are able to manipulate textures by the PDE function of the textures themselves.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] M. Abdulrahman, T. R. Gwadabe, F. J. Abdu, and A. Eleyan. Gabor wavelet transform based facial expression recognition using pca and lbp. In *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, pages 2265–2268, April 2014.

[3] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[4] S.S. Agaian, B. Silver, and K.A. Panetta. Transform Coefficient Histogram-Based Image Enhancement Algorithms Using Contrast Entropy. *Image Processing, IEEE Transactions on*, 16(3):741–758, 2007.

[5] Douglas G Altman and J Martin Bland. Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal*, 308(6943):1552, 1994.

[6] M. Amadasun and R. King. Textural features corresponding to textural properties. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):1264–1274, Sep 1989.

[7] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image Coding Using Wavelet Transform. *Image Processing, IEEE Transactions on*, 1(2):205–220, 1992.

[8] P. Arbelaez. Boundary Extraction in Natural Images Using Ultrametric Contour Maps. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 182–182, June 2006.

[9] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, May 2011.

[10] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[11] Jean-François Aujol, Gilles Aubert, Laure Blanc-Féraud, and Antonin Chambolle. Image Decomposition into A Bounded Variation Component and An Oscillating Component. *Journal of Mathematical Imaging and Vision*, 22(1):71–88, 2005.

[12] Jean-François Aujol and Antonin Chambolle. Dual Norms and Image Decomposition Models. *International Journal of Computer Vision*, 63(1):85–104, 2005.

[13] Jean-François Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. Structure-Texture Image Decomposition—Modeling, Algorithms, and Parameter Selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.

[14] Patel Janakkumar Baldevbhai and RS Anand. Color image segmentation for medical images using L* a* b* color space. *IOSR Journal of Electronics and Communication Engineering (IOSRJECE)*, 1(2):24–45, 2012.

[15] R.H. Bamberger and M. J T Smith. A Filter Bank for the Directional Decomposition of Images: Theory and Design. *Signal Processing, IEEE Transactions on*, 40(4):882–893, 1992.

[16] M.S. Bartlett, Javier R. Movellan, and T.J. Sejnowski. Face Recognition by Independent Component Analysis. *Neural Networks, IEEE Transactions on*, 13(6):1450–1464, 2002.

[17] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013.

[18] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous Structure and Texture Image Inpainting. *Image Processing, IEEE Transactions on*, 12(8):882–889, 2003.

[19] J. Bigun, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(8):775–790, Aug 1991.

[20] J. Bobin, Y. Moudden, and J.-L. Starck. Enhanced source separation by morphological component analysis. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V, May 2006.

[21] J. Bobin, Y. Moudden, J.-L. Starck, and M. Elad. Morphological diversity and source separation. *Signal Processing Letters, IEEE*, 13(7):409–412, July 2006.

[22] J Bobin, Y Moudden, JL Starck, and M Elad. Multichannel Morphological Component Analysis. *Proceedings of Spars, 2005*, pages 103–106, 2005.

[23] J. Bobin, J-L Starck, J.M. Fadili, Y. Moudden, and D.L. Donoho. Morphological Component Analysis: An Adaptive Thresholding Strategy. *Image Processing, IEEE Transactions on*, 16(11):2675–2681, 2007.

[24] P Brodatz. *A Photographic Album for Arts and Design*. Dover Publishing Co., Toronto, Canada, 1966.

[25] A. Buades, B. Coll, and J.-M. Morel. A Non-Local Algorithm for Image Denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65, June 2005.

[26] A. Buades, T.M. Le, J. M Morel, and L.A. Vese. Fast Cartoon+Texture Image Filters. *Image Processing, IEEE Transactions on*, 19(8):1978–1986, 2010.

[27] Vasile Buzuloiu, Mihai Ciuc, Rangaraj M. Rangayyan, and Constantin Vertan. Adaptive-neighborhood histogram equalization of color images. *Journal of Electronic Imaging*, 10(2):445–459, 2001.

[28] Emmanuel Candès, Laurent Demanet, David Donoho, and Lexing Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899, 2006.

[29] Emmanuel J. Candès. Harmonic analysis of neural networks. *Applied and Computational Harmonic Analysis*, 6(2):197–218, March 1999.

[30] Emmanuel J. Candès and David L. Donoho. Ridgelets: a key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 357(1760):2495–2509, 1999.

[31] Emmanuel J. Candes and David L. Donoho. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report, DTIC Document, 2000.

[32] John Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov 1986.

[33] Peng Cao, Ting Rui, Jin-lin Zhang, and You Zhou. An improved SIFT Matching Algorithm based on Locality Preserving Projection LPP. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, ICIMCS '12, pages 192–195, New York, NY, USA, 2012. ACM.

[34] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2707–2714, June 2010.

[35] Luigia Carlucci. A Formal System for Texture Languages. *Pattern Recognition*, 4(1):53–72, 1972.

[36] T. Celik. Spatial Entropy-Based Global and Local Image Contrast Enhancement. *Image Processing, IEEE Transactions on*, 23(12):5298–5308, Dec 2014.

[37] A. Chambolle. Partial differential equations and image processing. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 16–20, Nov 1994.

[38] T.F. Chan and L.A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, Feb 2001.

[39] S.G. Chang, Bin Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *Image Processing, IEEE Transactions on*, 9(9):1532–1546, 2000.

[40] G.Y. Chen, T.D. Bui, and A. Krzyzak. Image denoising using neighbouring wavelet coefficients. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 2, pages ii–917–20, 2004.

[41] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[42] Xiaoming Chen and Lili Lv. A Compositive Contrast Enhancement Algorithm of IR Image. In *Information Technology and Applications (ITA), 2013 International Conference on*, pages 58–62, Nov 2013.

[43] Yunmei Chen, Stacey Levine, and Murali Rao. Variable Exponent, Linear Growth Functionals in Image Restoration. *SIAM Journal on Applied Mathematics*, 66(4):1383–1406, 2006.

[44] J. Chi and M. Eramian. Enhancement of textural differences based on morphological component analysis. *IEEE Transactions on Image Processing*, 24(9):2671–2684, Sept 2015.

[45] Jianning Chi and Mark Eramian. Enhancing textural differences using wavelet-based texture characteristics morphological component analysis: A preprocessing method for improving image segmentation. accepted, April 2016.

[46] Jianning Chi and Mark Eramian. Wavelet-based texture-characteristic morphological component analysis for colour image enhancement. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 4097–4101. IEEE, September 2016.

[47] Hugh A. Chipman, Eric D. Kolaczyk, and Robert E. McCulloch. Adaptive Bayesian Wavelet Shrinkage. *Journal of the American Statistical Association*, 92(440):1413–1421, 1997.

[48] Raymond Y Cho, Vicky Yang, and Peter E Hallett. Reliability and dimensionality of judgments of visually textured materials. *Perception & psychophysics*, 62(4):735–752, 2000.

[49] Andrzej Cichocki, Shun-ichi Amari, et al. *Adaptive Blind Signal and Image Processing*. John Wiley Chichester, 2002.

[50] Ira Cohen, Ira Cohen, Qi Tian Xiang, Sean Zhou, Xiang Sean, Zhou Thomas, and Thomas S." Huang. Feature Selection Using Principal Feature Analysis. page 02, 2002.

[51] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, May 2002.

[52] Pierre Comon. Independent Component Analysis. *Higher-Order Statistics*, pages 29–38, 1992.

[53] G-H Cottet and L Germain. Image processing through reaction combined with nonlinear diffusion. *Mathematics of Computation*, pages 659–673, 1993.

[54] T. Cour, F. Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131 vol. 2, June 2005.

[55] George R. Cross and Anil K. Jain. Markov Random Field Texture Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5(1):25–39, Jan. 1983.

[56] R.N. Czerwinski, D.L. Jones, and Jr. O'Brien, W.D. Line and boundary detection in speckle images. *Image Processing, IEEE Transactions on*, 7(12):1700–1714, 1998.

[57] Rajeshwar Dass, Rajeshwar Dass, and Swapna" Devi. Image Segmentation Techniques. *International Journal of Electronics and Communication Technology*, 3(1):66–67, 2012.

[58] K. Deguchi and I. Morishita. Texture characterization and texture-based image partitioning using two-dimensional linear estimation techniques. *Computers, IEEE Transactions on*, C-27(8):739–745, 1978.

[59] Minh N. Do and Martin Vetterli. Contourlets: A Directional Multiresolution Image Representation. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–357–I–360, 2002.

[60] M.N. Do and M. Vetterli. Pyramidal directional filter banks and curvelets. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 158–161, 2001.

[61] M.N. Do and M. Vetterli. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *Image Processing, IEEE Transactions on*, 14(12):2091–2106, 2005.

[62] D. L. Donoho. Denoising by soft-thresholding. *IEEE Trans. Inf. Theory*, 41(3):613–627, March 1995.

[63] David L. Donoho and Mark R. Duncan. Digital curvelet transform: strategy, implementation, and experiments. *Wavelet Applications Vii*, 4056:12–30, 2000.

[64] David L. Donoho and Iain M. Johnstone. Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika*, 81(3):425–455, 1994.

[65] David L Donoho and Jain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[66] D.L. Donoho and I.M. Johnstone. Threshold selection for wavelet shrinkage of noisy data. In *Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE*, volume 1, pages A24–A25, 1994.

[67] O. Egger and W. Li. Subband Coding of Images Using Asymmetrical Filter Banks. *Image Processing, IEEE Transactions on*, 4(4):478–485, 1995.

[68] Moumen El-Melegy, Ennumeri A Zanaty, Walaa M Abd-Elhafiez, and Aly Farag. On cluster validity indexes in fuzzy and hard clustering algorithms for image segmentation. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–5. IEEE, 2007.

[69] M. Elad, J.-L. Starck, P. Querre, and D.L. Donoho. Simultaneous Cartoon and Texture Image Inpainting Using Morphological Component Analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.

[70] Mark G Eramian, Gregg P Adams, and Roger A Pierson. Enhancing ultrasound texture differences for developing an in vivo'virtual histology'approach to bovine ovarian imaging. *Reproduction, Fertility and Development*, 19(8):910–924, 2007.

[71] Jalal M Fadili and Jean-Luc Starck. Sparse representations and bayesian image inpainting. In *Proceedings of International Conferences SPARS'05*, pages 4–pp, 2005.

[72] M.J. Fadili, J-L Starck, J. Bobin, and Y. Moudden. Image Decomposition and Separation Using Sparse Representations: An Overview. *Proceedings of the IEEE*, 98(6):983–994, 2010.

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

[74] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281–305, 1987.

[75] J.E. Fowler. The Redundant Discrete Wavelet Transform and Additive Noise. *Signal Processing Letters, IEEE*, 12(9):629–632, 2005.

[76] K. Fujii and T. Arikawa. Urban object reconstruction using airborne laser elevation image and aerial image. *Geoscience and Remote Sensing, IEEE Transactions on*, 40(10):2234–2240, Oct 2002.

[77] Vishwadeep Garg and Kulbir Singh. An improved Grunwald-Letnikov fractional differential mask for image texture enhancement. *International Journal of Advanced Computer Science and Application (IJACSA)*, 3(3), 2012.

[78] Qi Ge, Liang Xiao, Jun Zhang, and Zhi Hui Wei. An improved region-based model with local statistical features for image segmentation. *Pattern Recognition*, 45(4):1578–1590, 2012.

[79] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 456–463. IEEE, 2003.

[80] Ali Ghodsi. Dimensionality reduction a short tutorial. Technical report, Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada, 2006.

[81] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric statistical inference*. Springer, 5 edition, 2011.

[82] Jérôme Gilles. Noisy Image Decomposition: A New Structure, Texture and Noise Model based on Local Adaptivity. *Journal of Mathematical Imaging and Vision*, 28(3):285–295, 2007.

[83] Rafael C Gonzalez and E Richard. *Digital Image Processing*. Prentice Hall Press, 2002.

[84] R. Gribonval and M. Nielsen. Sparse Representations in Unions of Bases. *Information Theory, IEEE Transactions on*, 49(12):3320–3325, 2003.

[85] Rémi Gribonval and Sylvain Lesage. A survey of Sparse Component Analysis for blind source separation: principles, perspectives, and new challenges. In *ESANN'06 proceedings - 14th European Symposium on Artificial Neural Networks*, pages 323–330, Bruges, Belgium, April 2006. d-side publi.

[86] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski. Non-rigid Dense Correspondence with Applications for Image Enhancement. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 70:1–70:10, New York, NY, USA, 2011. ACM.

[87] Yoav Hacohen, Eli Shechtman, and Dani Lischinski. Deblurring by example using dense correspondence. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2384–2391, December 2013.

[88] Ji-Hee Han, Sejung Yang, and Byung-Uk Lee. A Novel 3-D Color Histogram Equalization Method With Uniform 1-D Gray Scale Histogram. *Image Processing, IEEE Transactions on*, 20(2):506–512, Feb 2011.

[89] M.C. Hanumantharaju, M. Ravishankar, D.R. Rameshbabu, and S. Ramachandran. Color Image Enhancement Using Multiscale Retinex with Modified Color Restoration Technique. In *Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on*, pages 93–97, Feb 2011.

[90] R.M. Haralick, K. Shanmugam, and Its"Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621, Nov. 1973.

[91] Robert M Haralock and Linda G Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[92] F. J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, Jan 1978.

[93] Ronald R Hocking. Developments in Linear Regression Methodology: 1959-l982. *Technometrics*, 25(3):219–230, 1983.

[94] L. Hong, Yifei Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):777–789, 1998.

[95] Ching-Tang Hsieh, Eugene Lai, and You-Chuang Wang. An effective algorithm for fingerprint image enhancement based on wavelet transform. *Pattern Recognition*, 36(2):303–312, 2003.

[96] Fuyuan Hu, Shaohui Si, Hau San Wong, Baochuan Fu, MaoXin Si, and Heng Luo. An adaptive approach for texture enhancement based on a fractional differential operator with non-integer step and order. *Neurocomputing*, 158:295–306, 2015.

[97] QH Huang, Shuai Wang, and Zhao Liu. Improved Algorithm of Image Feature Extraction based on Independent Component Analysis. *Opto-Electronic Engineering*, 1:121–125, 2007.

[98] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(1):13–18, 1979.

[99] Zhi-Chun Huang, P.P.K. Chan, W.W.Y. Ng, and D.S. Yeung. Content-based Image Retrieval using Color Moment and Gabor Texture Feature. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, volume 2, pages 719–724, July.

[100] A. Hyvarinen, R. Cristescu, and E. Oja. A Fast Algorithm for Estimating Overcomplete ICA Bases for Image Windows. In *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, volume 2, pages 894–899, 1999.

[101] Aapo Hyvarinen. Survey on Independent Component Analysis. *Neural computing surveys*, 2(4):94–128, 1999.

[102] Aapo Hyvärinen and Erkki Oja. Independent Component Analysis: Algorithms and Applications. *Neural networks*, 13(4):411–430, 2000.

[103] M. M. Islam, Dengsheng Zhang, and Guojun Lu. A geometric method to compute directionality features for texture images. In *2008 IEEE International Conference on Multimedia and Expo*, pages 1521–1524, June 2008.

[104] Anil K Jain. *Fundamentals of digital image processing*, volume 3. Prentice-Hall Englewood Cliffs, 1989.

[105] Hamid A Jalab and Rabha W Ibrahim. Texture enhancement based on the savitzky-golay fractional differential operator. *Mathematical Problems in Engineering*, 2013, 2013.

[106] Robert Jenssen and Torbjørn Eltoft. ICA Filter Bank for Segmentation of Textured Images. In *The 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan*, pages 827–832, 2003.

[107] Hongliang Jin, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. Face Detection using Improved LBP under Bayesian Framework. In *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, pages 306–309, dec. 2004.

[108] Hydin John and J Anitha. A Study of Image Segmentation Approaches. *International Journal of Advanced Research in Electronics and Communication Engineering*, 1(4):62–68, 2012.

[109] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[110] C. R. Jung and J. Scharcanski. Sharpening dermatological color images in the wavelet domain. *IEEE Journal of Selected Topics in Signal Processing*, 3(1):4–13, Feb 2009.

[111] Timor Kadir and Michael Brady. Saliency, Scale and Image Description. *International Journal of Computer Vision*, 45:83–105, 2001.

[112] Wen-Xiong Kang, Qing-Qiang Yang, and Run-Peng Liang. The comparative research on image segmentation algorithms. In *2009 First International Workshop on Education Technology and Computer Science*, pages 703–707. IEEE, 2009.

[113] Yan Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 506–513, June 2004.

[114] WJ Krzanowski. Selection of variables to preserve multivariate data structure, using principal components. *Applied Statistics*, pages 22–33, 1987.

[115] WJ Krzanowski. A stopping rule for structure-preserving variable selection. *Statistics and Computing*, 6(1):51–56, 1996.

[116] Kenneth I Laws. Texture energy measures. In *Proc. Image Understanding Workshop*, pages 47–51. University of Southern California Los Angles Image Processing Inst, 1979.

[117] Kenneth I Laws. Rapid Texture Identification. In *Proc. SPIE 0238, Image Processing for Missile Guidance, 376*, volume 238, pages 376–381, 1980.

[118] S. Lazebnik, C. Schmid, and J. Ponce. A Sparse Texture Representation using Affine-Invariant Regions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 319–324, june 2003.

[119] A. S. Lewis and G. Knowles. Image Compression Using the 2-D Wavelet Transform. *Image Processing, IEEE Transactions on*, 1(2):244–250, 1992.

[120] Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. *IEEE Transactions on Neural Networks*, 17(1):157–165, Jan 2006.

[121] Kaiyang Liao and Guizhong Liu. An Improved Local Feature Descriptor based on SIFT. In *Proceedings of the Second International Conference on Internet Multimedia Computing and Service*, ICIMCS '10, pages 131–134, New York, NY, USA, 2010. ACM.

[122] S. Liao, M.W.K. Law, and A.C.S. Chung. Dominant Local Binary Patterns for Texture Classification. *Image Processing, IEEE Transactions on*, 18(5):1107–1118, May 2009.

[123] Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.

[124] Alan Wee-Chung Liew and Hong Yan. An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation. *Medical Imaging, IEEE Transactions on*, 22(9):1063–1075, 2003.

[125] Tai-Shan Lin and Jack Meador. Statistical feature extraction and selection for IC test pattern analysis. In *Circuits and Systems, 1992. ISCAS'92. Proceedings., 1992 IEEE International Symposium on*, volume 1, pages 391–394. IEEE, 1992.

[126] Paulo JG Lisboa and AR Mehri-Dehnavi. Sensitivity methods for variable selection using the MLP. In *Neural Networks for Identification, Control, Robotics, and Signal/Image Processing, 1996. Proceedings., International Workshop on*, pages 330–338. IEEE, 1996.

[127] Shigang Liu and Yali Peng. A local region-based chan–vese model for image segmentation. *Pattern Recognition*, 45(7):2769–2779, 2012.

[128] Xingmiao Liu, Jing Zhao, and Shicheng Wang. Nonlinear Algorithm of Image Enhancement Based on Wavelet Transform. In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, pages 1–4, 2009.

[129] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[130] D.G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157, 1999.

[131] Yuan Luo, Cai ming Wu, and Yi Zhang. Facial expression recognition based on fusion feature of PCA and LBP with SVM. *Optik - International Journal for Light and Electron Optics*, 124(17):2767–2770, 2013.

[132] PC Mahalanobis. Mahalanobis distance. In *Proceedings National Institute of Science of India*, volume 49, pages 234–256, 1936.

[133] Raman Maini and Himanshu Aggarwal. A Comprehensive Review of Image Enhancement Techniques. March 2010.

[134] Tomáš Majtner and David Svoboda. Extension of tamura texture features for 3d fluorescence microscopy. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 301–307. IEEE, 2012.

[135] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423, 2001.

[136] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, Feb 2001.

[137] Matti Pietikäinen. Image analysis with local binary patterns. In Heikki Kalviainen, Jussi Parkkinen, and Arto Kaarna, editors, *Image Analysis*, volume 3540, pages 115–118. Springer Berlin Heidelberg, 2005.

[138] George P McCabe. Principal variables. *Technometrics*, 26(2):137–144, 1984.

[139] Yves Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: the Fifteenth Dean Jacqueline B. Lewis memorial lectures*, volume 22. Amer Mathematical Society, 2001.

[140] Tomer Michaeli and Michal Irani. *Blind Deblurring Using Internal Patch Recurrence*, pages 783–798. Springer International Publishing, Cham, 2014.

[141] K. Mikolajczyk and C. Schmid. Indexing based on Scale Invariant Interest Points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525–531, 2001.

[142] Hossein Mobahi, ShankarR. Rao, AllenY. Yang, ShankarS. Sastry, and Yi Ma. Segmentation of Natural Images by Texture and Boundary Compression. *International Journal of Computer Vision*, 95(1):86–98, 2011.

[143] A. Mojsilovic and B. Rogowitz. Capturing image semantics with low-level descriptors. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 18–21, 2001.

[144] Plinio Moreno, Alexandre Bernardino, and JosÃ© Santos-Victor. Improving the SIFT Descriptor with Smooth Derivative Filters. *Pattern Recognition Letters*, 30(1):18–26, 2009.

[145] Kevin P Murphy. Naive bayes classifiers. *University of British Columbia*, 2006.

[146] S.K. Naik and C.A. Murthy. Hue-preserving color image enhancement without gamut problem. *Image Processing, IEEE Transactions on*, 12(12):1591–1598, Dec 2003.

[147] S.C. Nercessian, K.A. Panetta, and S.S. Agaian. Non-Linear Direct Multi-Scale Image Enhancement Based on the Luminance and Contrast Masking Characteristics of the Human Visual System. *Image Processing, IEEE Transactions on*, 22(9):3549–3561, Sept 2013.

[148] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE, 2006.

[149] Mila Nikolova. A Variational Approach to Remove Outliers and Impulse Noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120, 2004.

[150] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, July 2002.

[151] S. Osher and L. Rudin. Feature-Oriented Image Enhancement Using Shock Filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

[152] Chandrajit Pal, Amlan Chakrabarti, and Ranjan Ghosh. A brief survey of recent edge-preserving smoothing algorithms on digital images. *CoRR*, abs/1503.07297, 2015.

[153] S. Park, M. J T Smith, and Jun Jae Lee. Fingerprint Enhancement based on the Directional Filter Bank. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 3, pages 793–796, 2000.

[154] Dinesh D Patil and Sonal G Deore. Medical image segmentation: a review. *International Journal of Computer Science and Mobile Computing*, 2(1):22–27, 2013.

[155] Soo-Chang Pei and Chao-Nan Lin. Image normalization for pattern recognition. *Image and Vision Computing*, 13(10):711–723, 1995.

[156] Alex P. Pentland. Fractal-Based Description of Natural Scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):661–674, 1984.

[157] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987.

[158] Andrea Polesel, Giovanni Ramponi, and V John Mathews. Image enhancement via adaptive unsharp masking. *IEEE transactions on image processing*, 9(3):505–510, 2000.

[159] Y. F. Pu, J. L. Zhou, and X. Yuan. Fractional Differential Mask: A Fractional Differential-Based Approach for Multiscale Texture Enhancement. *IEEE Transactions on Image Processing*, 19(2):491–511, Feb 2010.

[160] Pavel Pudil and Jana Novovičová. Novel methods for feature subset selection with respect to problem knowledge. In *Feature Extraction, Construction and Selection*, pages 101–116. Springer, 1998.

[161] S. Qian and D. Chen. Discrete Gabor transform. *IEEE Transactions on Signal Processing*, 41(7):2429–2438, Jul 1993.

[162] A Ravishankar Rao. *Computing oriented texture fields*. Springer, 1990.

[163] A.R. Rao and G.L. Lohse. Towards a texture naming system: Identifying relevant dimensions of texture. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pages 220–227, Oct 1993.

[164] K. R. Rao, Patrick Yip, and Vladimir Britanak. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Inc., Orlando, FL, USA, 2007.

[165] Andrea Rocco and Bruce J. West. Fractional calculus and the evolution of fractal phenomena. *Physica A: Statistical Mechanics and its Applications*, 265(3â“4):535 – 546, 1999.

[166] Azriel Rosenfeld. The Max Roberts Operator is a Hueckel-Type Edge Detector. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-3(1):101–103, 1981.

[167] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 699–704, June 2003.

[168] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[169] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear Total Variation based Noise Removal Algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[170] Sachin D. Ruikar and Dharmpal D. Doye. Wavelet based image denoising technique. *International Journal of Advanced Computer Science and Applications*, 2(3), 2011.

[171] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250, Jun 1996.

[172] Sujata Saini and Komal Arora. A Study Analysis on the Different Image Segmentation Techniques. *International Journal of Information & Computation Technology*, 4(14):1445–1452, 2014.

[173] P Sakellaropoulos, L Costaridou, and G Panayiotakis. A wavelet-based spatially adaptive method for mammographic contrast enhancement. *Physics in Medicine and Biology*, 48(6):787, 2003.

[174] F. Schaffalitzky and A. Zisserman. Automated Scene Matching in Movies. In MichaelS. Lew, Nicu Sebe, and JohnP. Eakins, editors, *Image and Video Retrieval*, volume 2383 of *Lecture Notes in Computer Science*, pages 186–197. Springer Berlin Heidelberg, 2002.

[175] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5):530–535, May 1997.

[176] Thomas Schoenemann, Fredrik Kahl, Simon Masnou, and Daniel Cremers. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *International Journal of Computer Vision*, 99(1):53–68, 2012.

[177] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-Dimensional Sift Descriptor and its Application to Action Recognition. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 357–360, New York, NY, USA, 2007. ACM.

[178] Levent Sendur and I.W. Selesnick. A bivariate shrinkage function for wavelet-based denoising. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–1261–II–1264, 2002.

[179] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, Dec 1993.

[180] Arnold Roy Shulman. Optical data processing. 1970.

[181] Heung-Yeung Shum, Katsushi Ikeuchi, and Raj Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(9):854–867, 1995.

[182] E.P. Simoncelli. Design of Multi-dimensional Derivative Filters. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 790–794, 1994.

[183] Irwin Sobel. History and definition of the sobel operator. *Retrieved from the World Wide Web*, 2014.

[184] T. Song, H. Li, F. Meng, Q. Wu, B. Luo, B. Zeng, and M. Gabbouj. Noise-Robust Texture Description Using Local Contrast Patterns via Global Measures. *IEEE Signal Processing Letters*, 21(1):93–96, Jan 2014.

[185] GE Sotak and Kim L Boyer. The Laplacian-of-Gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Computer Vision, Graphics, and Image Processing*, 48(2):147–189, 1989.

[186] GN Srinivasan and G Shobha. Statistical Texture Analysis. In *Proceedings of world academy of science, engineering and technology*, volume 36, pages 1264–1269, 2008.

[187] J-L Starck, E.J. Candes, and D.L. Donoho. The Curvelet Transform for Image Denoising. *Image Processing, IEEE Transactions on*, 11(6):670–684, 2002.

[188] J-L Starck, M. Elad, and D.L. Donoho. Image Decomposition via the Combination of Sparse Representations and a Variational Approach. *Image Processing, IEEE Transactions on*, 14(10):1570–1582, 2005.

[189] J.-L. Starck, Y. Moudden, J. Bobin, M. Elad, and D. L. Donoho. Morphological Component Analysis. *International Society for Optics and Photonics, 2005*, pages 59140Q–59140Q–15, 2005.

[190] J-L Starck and Fionn Murtagh. *Astronomical image and data analysis*. Springer Science & Business Media, 2007.

[191] Jean-Luc Starck, Michael Elad, and David Donoho. Redundant multiscale transforms and their application for morphological component separation. *Advances in Imaging and Electron Physics*, 132:287–348, 2004.

[192] Jean-Luc Starck, Mikael Elad, and David L. Donoho. Image Decomposition: Separation of Texture from Piecewise Smooth Content. pages 571–582, 2003.

[193] Jean-Luc Starck, Fionn D Murtagh, and Albert Bijaoui. *Image processing and data analysis: the multiscale approach*. Cambridge University Press, 1998.

[194] Boaz J. Super and Alan C. Bovik. Shape from texture using local spectral moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):333–343, 1995.

[195] R.N. Sutton and Ernest L. Hall. Texture measures for automatic classification of pulmonary disease. *Computers, IEEE Transactions on*, C-21(7):667–676, July 1972.

[196] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural Features Corresponding to Visual Perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, 1978.

[197] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[198] A. Toet. A Morphological Pyramidal Image Decomposition. *Pattern Recognition Letters*, 9(4):255–261, 1989.

[199] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, 1998.

[200] Fumiaki Tomita and Saburo Tsuji. *Computer Analysis of Visual Textures*. Kluwer Academic Publishers, 1990.

[201] Meng-Hsiun Tsai, Yung-Kuan Chan, An-Mei Hsu, Chia-Yi Chuang, Chuin-Mu Wang, and Po-Whei Huang. Feature-Based Image Segmentation. *Journal of Imaging Science and Technology*, 57(1):10505–1–10505–12, 2013.

[202] Michael Unser and Murray Eden. Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):717–728, 1989.

[203] L. Vincent. Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *Image Processing, IEEE Transactions on*, 2(2):176–201, 1993.

[204] Dennis Wackerly, William Mendenhall, and Richard L Scheaffer. *Mathematical statistics with applications*. Nelson Education, 2007.

[205] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[206] Yi Wan and Dongbin Shi. Joint Exact Histogram Specification and Image Enhancement Through the Wavelet Transform. *Image Processing, IEEE Transactions on*, 16(9):2245–2250, 2007.

[207] Xin Wang. Laplacian Operator-Based Edge Detectors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(5):886–890, 2007.

[208] G Weber. Usc-sipi image database: Version 4. 1993.

[209] Joachim Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2-3):111–127, 1999.

[210] Joachim Weickert. Coherence-enhancing shock filters. In Bernd Michaelis and Gerald Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 1–8. Springer Berlin Heidelberg, 2003.

[211] Joachim Weickert and Hanno Scharr. A Scheme for Coherence-Enhancing Diffusion Filtering with Optimized Rotation Invariance. *Journal of Visual Communication and Image Representation*, 13(12):103 – 118, 2002.

[212] S. Wesolkowski, M.E. Jernigan, and R.D. Dony. Comparison of Color Image Edge Detectors in Multiple Color Spaces. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 2, pages 796–799, 2000.

[213] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[214] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[215] Shuicheng Yan, Huan Wang, Xiaoou Tang, and T. Huang. Exploring Feature Descritors for Face Recognition. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages 629–632, Apr. 2007.

[216] Ching-Chung Yang. Image Enhancement by Modified Contrast-stretching Manipulation. *Optics and Laser Technology*, 38(3):196–201, 2006.

[217] Hong Yang and Yiding Wang. An LBP-based Face Recognition Method with Hamming Distance Constraint. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 645–649, Aug. 2007.

[218] Pu Yi-Fei. Application of fractional differential approach to digital image processing. *Journal of Sichuan University (Engineering Science Edition)*, 3:22, 2007.

[219] Wotao Yin, Donald Goldfarb, and Stanley Osher. Image Cartoon-Texture Decomposition and Feature Selection Using the Total Variation Regularized L1 Functional. *Variational, Geometric, and Level Set Methods in Computer Vision*, 3752:73–84, 2005.

[220] Jiangye Yuan, DeLiang Wang, and A.M. Cheriyadat. Factorization-Based Texture Segmentation. *Image Processing, IEEE Transactions on*, 24(11):3488–3497, Nov 2015.

[221] C. Zeng and H. Ma. Robust Head-Shoulder Detection by PCA-Based Multilevel HOG-LBP Detector for People Counting. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2069–2072, Aug 2010.

[222] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, 73:213–238, 2007.

[223] Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, and StanZ. Li. Face Detection Based on Multi-Block LBP Representation. In Seong-Whan Lee and StanZ. Li, editors, *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 11–18. Springer Berlin Heidelberg, 2007.

[224] Xueliang Zhang, Xuezhi Feng, Pengfeng Xiao, Guangjun He, and Liujun Zhu. Segmentation quality evaluation using region-based precision and recall measures for remote sensing images. *Journal of Photogrammetry and Remote Sensing*, 102:73 – 84, 2015.

[225] Yang Zhao. Theories and applications of lbp: A survey. In De-Shuang Huang, Yong Gan, Phalguni Gupta, and M.Michael Gromiha, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, volume 6839 of *Lecture Notes in Computer Science*, pages 112–120. Springer Berlin Heidelberg, 2012.

[226] Xiang S Zhou and Thomas S Huang. CBIR: from low-level features to high-level semantics. In *Electronic Imaging*, pages 426–431. International Society for Optics and Photonics, 2000.

[227] Michael Zibulevsky, Pavel Kisilev, Yehoshua Y Zeevi, and Barak A Pearlmutter. Blind Source Separation via Multinode Sparse Representation. *Advances in neural information processing systems*, 14:1049–1056, 2002.

[228] Michael Zibulevsky and Barak A Pearlmutter. Blind Source Separation by Sparse Decomposition in a Signal Dictionary. *Neural Computation*, 13(4):863–882, 2001.

[229] J.B. Zimmerman, S.M. Pizer, E.V. Staab, J.R. Perry, W. McCartney, and B.C. Brenton. An evaluation of the effectiveness of adaptive histogram equalization for contrast enhancement. *Medical Imaging, IEEE Transactions on*, 7(4):304–312, 1988.

[230] Steven W Zucker. Toward a model of texture. *Computer Graphics and Image Processing*, 5(2):190–202, 1976.