The Hong Kong Polytechnic University

New Wavelet Model for Texture-Based Feature Extraction and Retrieval

By

Chan Wai Yip Patrick

A Thesis submitted for the Degree of Master of Philosophy in the Department of Electronic and Information Engineering of The Hong Kong Polytechnic University

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University

2002

Abstract

As a picture can describe a thousand of words, it is necessary to develop a computer management system that can efficiently handle image/video data. Subsequently, a good image/video database system should provide a fast and accurate method that can help us handle the matching problem of visual features. Texture can be recognized as an important feature for content-based image/video feature similarity matching. One of the objectives of this research is to investigate a fast and accurate texture feature retrieval method which can be efficiently used on an image/video database for similarity searching. We suggest using the over-complete wavelet-based model for texture feature analysis. The method matches the viewpoint of human vision on multi-channels, frequencies and orientations of texture properties. The computation time for this texture analysis is small as compared to other well-known texture analysis methods, such as the Gabor wavelets. A new texture analysis approach from the Laplacian of Gaussian(LOG)-based over-complete wavelets is derived. It is proven to be suitable for texture analysis from both theoretical and experimental studies. Also, a new texture representation that can significantly improve the retrieval accuracy of texture features for both over-complete and sub-sampling wavelet schemes is given. We have compared different texture-based feature retrieval methods, and experimental results indicate that our proposed method achieve the highest retrieval rate on the entire Brodatz texture database with a low feature analysing time. It is thus suitable for real-time content-based retrieval applications. We have also analysed the retrieval performance of various texture-based feature retrieval methods under different Gaussian noise levels. Results of the study indicate that the over-complete wavelet scheme is robust against noise, due to its translation invariant property.

The traditional over-complete wavelet implementation contains redundancy between filtering structures. Also, it causes boundary artifact in reconstructed images. Therefore, based on LOG-base over-complete wavelet transform derived for texture analysis, we have investigated a fast implementation method which aims to reduce the complexity and boundary artifact of wavelet transforms. Results of our theoretical and experimental studies show that the computational complexity can be reduced significantly by using the spatial approach as compared to the filtering approach. Furthermore, it can effectively eliminate the boundary artifacts, during multi-resolution wavelet transforms effectively, and thus perfect reconstruction is always obtained.

Acknowledgments

I am very grateful to Professor Wan-Chi Siu for his excellent guidance, encouragement and support during the course of this research. He spent much of his invaluable time with me discussing my research, reviewing and correcting my articles and drafts of this thesis. His profound knowledge and experience in digital signal processing, his rigorous approach to research, many interesting and stimulating lectures that he has given, and his hard working style and devotion to science research, have inspired me during my work on the thesis. I learn much knowledge from him that becomes valuable assets for my future study and career.

I would also like to express my appreciation to Professor David Feng for his support and encouragement during my research. He gave me much of his profound knowledge and experience supporting of my research.

I would also like to express my appreciation to all colleagues in the Center of Multimedia Signal Processing for their friendship, support and encouragement. Especially, I would like thanks to Dr. Bonnie Law, Dr. F. H. Long, Dr. Q. Wong, Dr. Y. L. Chan, Mr. C. P. Chow, Mr. H. K. Cheung, Mr. W. H Wong, Mr. K. C. Hui, Mr. K. T. Fung, Miss K. L. Lau, Mr. W. L. Hui and all the other members in the Center of Multimedia Signal Processing. They aided me immensely during my project and shared their useful ideas during discussions.

Finally, but not last, I would like to express my gratitude for studentship support from the Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University.

Table of Contents

List of Figures

List of Tables

Statements of Originality

The following points are claimed to be original in this work.

1. Based on the Canny-based over-complete wavelet proposed by Mallat and Zhong, a new Laplacian of Gaussian(LOG)-based over-complete wavelet has been derived. The new wavelet kernel corresponds to the second deviative of the Spline family and is equivalent to the Laplacian of Gaussian filter. The wavelet kernel is proved to allow perfect reconstruction, i.e., a stable inverse transform can always be obtained. Details of 1D LOG-based over-complete wavelet transforms can be found in Section 3.1.2. And details of 2D LOG-based over-complete wavelet transform can be found in Section 3.2.2.

2. This thesis suggests using the LOG-based over-complete wavelet for texture feature extraction. Similar to other spatial approaches for texture analysis, such as Gabor wavelets and sub-sampling wavelets, the over-complete wavelet transform can characterize a texture image in multi-frequencies and orientations. This is similar to the human vision system on texture recognition. Even though the over-complete wavelet scheme requires longer computational time than the sub-sampling wavelet scheme for texture analysis, it does not have the translation invariant problem. Compared to the Canny-based over-complete wavelet, the LOG-based over-compete wavelet has a better distinguishing capability on the thickness of a line in texture image. Therefore, it is more suitable for describing the texture features. Details about using the LOG-based over-complete wavelet for texture feature extraction can be found in Section 3.2.2.

3. A new texture representation which can significantly increase the retrieval rate of the entire Brodatz texture database is proposed. The new representation can effectively characterize the texture feature of an image which is extracted from over-complete wavelet transforms and sub-sampling wavelet transforms. Different from the common statistical method for texture feature representation which calculates the variance of filter responses, the new representation calculates the mean and standard deviation of bandpass filter responses at positive and negative sides separately. As a consequence, it produces a 52 dimensional texture feature vector. Retrieval results on the entire Brodatz database show that the new representation can increase significantly the retrieval rate of both sub-sampling and over-complete wavelets scheme for texture extraction compared to using the variance representation. More details of the new feature representation are in Section 3.2. And details of comparative retrieval results using the new feature representation and other feature representations are in Section 3.3.2.

4. Comparative experiments are conducted to compare different methods of similarity retrieval on the entire Brodatz texture database. The comparison of texture feature extraction methods includes LOG-based over-complete wavelet, Canny-based over-complete wavelet, Gabor wavelets and various sub-sampling wavelets. Experimental results show that the LOG-based over-complete wavelet can achieve the highest retrieval rate on the entire Brodatz texture database as compared to others. Furthermore, these methods have been tested on the Brodatz texture image with different levels of Gaussian noise. The over-complete wavelets, both Canny- and LOG-based wavelet kernels, have the smallest reduction retrieval rate as compared to other retrieval methods. It is demonstrated

that the over-complete wavelet scheme can achieve a more robust performance under different noise levels. Details of the comparison experiments are in Section 3.3.3 and Section 3.3.4.

5. This research proposes a feature reduction method which reduces the redundancy information in the new feature representation. As a consequence, it reduces the length of texture features from 52 to 41 dimensions. Using the reduced feature representation, it can efficiently decrease 25% of the retrieval time as compared to the original feature representation. From the retrieval results on the entire Brodatz texture database using LOG-based over-complete wavelet, the reduce feature representation causes only a 0.25% decrease in retrieval rate as compared to the original representation. Details of the reduced feature representation and its retrieval results are in Section 3.3.6.

6. A new spatial implementation for the LOG-based over-complete wavelet transform is proposed. The wavelet transform and conventional filtering approaches ignore the redundancy between the lowpass and the bandpass filters and may cause boundary artifacts. By deriving some general expressions for the computational cost using the conventional filtering implementation, it reveals that the inverse transform is significantly more costly than the forward transform. It is undesirable for real-time applications. By exploiting the redundancy between filters, it is possibly to use an alternative implementation method to reduce the redundancy. A spatial implementation of the 1D LOG-based over-complete wavelet transform is proposed. Results of both theoretical and experimental studies show that this spatial implementation of the LOG-based over-complete wavelet can highly decrease the computational complexity, especially the

complexity of inverse transforms. It can eliminate the boundary artifact of reconstructed signals completely without the use of any compensation method. Details of 1D spatial implementation of LOG-based over-complete wavelet and its experimental results can be found in Section 4.2 to Section 4.3.

7. Based on the spatial implementation of LOG-based over-complete wavelet for 1D signals, we have extended the approach to the 2D wavelet transform. Similar to the problems in 1D, the inverse transform is significantly more costly as compared to the forward transform in the 2D case. It also causes boundary artifact of reconstructed images. A 2D spatial implementation is proposed which can efficiently reduce the computational time of the transform, especially the inverse transform, as compared to the filtering approach. It also eliminates the boundary artifact of the reconstructed image completely without any compensation. Details of 2D spatial implementation of the LOG-based over-complete wavelet and its experimental results are in Section 5.2 to Section 5.4.

Author's Publications

(List of Publications of the Author on which this thesis is based)

International journal papers

Recent submissions

1. Wai-Yip Chan, Wan-Chi Siu and Ngai-Fong Law and Dagan Feng, "A New Texture Model For Content-Based Image Extraction and Retrieval", Submitted to IEEE Transactions on Image Processing.

Paper in preparation

1. Wai-Yip Chan, Wan-Chi Siu and Ngai-Fong Law, "Fast Algorithm for 1D Laplacian of Gaussian-based Over-Complete Wavelets ", paper in preparation.

2. "A Spatial Approach of Laplacian of Gaussian-Based Over-Complete Wavelets with Content-based Feature Extraction and Retrieval", paper in preparation.

International conference papers

Paper published

1. Fuhui Long, Wai-Yip Chan, David Feng and Wan-Chi Siu, "Region Mosaic Based Wold Decomposition with Its Applications in Image Retrieval Systems", Proceeding International Symposium on Signal Processing and Intelligent System, pp. 628-632, GuangZhou, China, Nov. 1999.

2. Fuhui Long, Wai-Yip Chan, David Feng and Wan-Chi Siu, "Video Retrieval Based on Revised Wold Decomposition and Gabor Filtering for Texture Analysis", Proceedings, 2000 International Workshop on Multimedia Data Store, Retrieval, Integration and Application, pp. 264-268, Hong Kong, Jan. 2000.

CHAPTER 1

1. Introduction

The growth of digital image and video information in computer technology demands an efficient tool for management. Conventional database management system may help to solve the problems of data storage, indexing and retrieval. However, visual features, such as image and video, can not easily be described by their contents or by words. As a result, the conventional word searching system may not sufficiently satisfy the requirement for indexing. It directly affects the accuracy of similarity searching. In order to provide a system with image and video search capabilities, content-based visual query techniques are being developed, to index the visual features of images[1, 2, 3, 4, 5, 6] and videos[7, 8]. The principle of content-based retrieval system is based on pictorial similarity for searching.

Among different content-based features which have been proposed, texture feature is recognized as an important candidate for image/video data searching. However, retrieval of texture images is a very difficult task. Texture images contain a wide variety of patterns, e.g. bricks, sand and grass patches, etc. To develop a practical system for texture similarity searching, an appropriate model is needed, and it should make use of human vision on texture for its evaluation. Furthermore, the time for texture analysis is another essential consideration to develop applications. Computer users may not be patient enough for long analysing time.

1.1. Texture

1.1.1. What is the definition of texture?

It is very difficult to define texture even we recognize its importance. Textures can be described as a spatial arrangement of pixels which contain some homogeneous patterns. This involves different feature properties, including statistical[**11**] and structural features[**12**], or both[**13**]. Texture may contain structured or random elements but may also contain no fundamental sub-units. Examples of texture images from the Brodatz Album[**14**], as shown in Figure 1.1, illustrate a wide variety of textures. Texture may contain structural (e.g.D1, D3, D49 and D69) or random patterns (e.g. D2 and D5). We can further classify structural patterns into periodic patterns (e.g.D1, D3) and directionality patterns (E.g. D49 and D69).

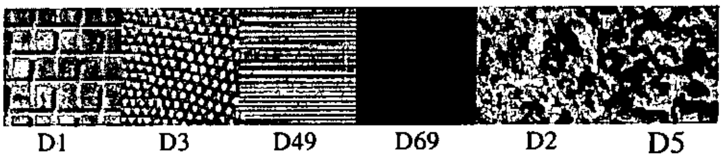


Figure 1.1, Examples of texture image from the Brodatz Album. From left to right, D1(Woven Aluminum wire), D3(Reptile skin), D49(Straw screening), D69(Wood grain), D2(Field stone) and D5(Expanded mica).

Much research effort[**15, 16, 17, 18, 19, 20, 21**] have tried to define texture and hence use it to form practical machine vision systems. However, there is no generally an agreed definition of texture. Some definitions of texture features are perceptually motivated and others are drived completely by the application in which the definition is used.

1.1.2. Related research work on texture feature retrieval

Techniques for texture modeling have been investigated for a long time in different research areas such as image segmentation, classification and similarity retrieval.

Texture extraction is concerned with finding a suitable model to describe the feature of natural texture images. Texture retrieval is to match characteristics of extracted textures with the reference. Examples of excellent texture retrieval tools include 2D Wold-based feature modeling[22], Gabor wavelets[23, 24], and wavelet packets[25, 26] etc. Liu and Picard[22] compared Wold-based features with other statistical texture approaches and concluded that our human vision system should contain some degree of structural and statistical structures. Wold-based analysis can achieve outstanding retrieval rate on the entire Brodatz database retrieval test[28]. Ma and Manjunath[23] proposed to use Gabor wavelets for texture feature retrieval. Their comparison of the Gabor wavelets with various texture analysing methods show that the Gabor wavelets achieved the highest retrieval results on the entire Brodatz texture database. They concluded that the Gabor wavelet is the best texture analysing method for multi-orientation and multi-frequency cases which involve human vision on textures. Mojsilovic *et al*[25] analysed more than 30 wavelet kernels for a sub-sampling wavelet scheme with 30 Brodatz Albums to test their performance on classifications. They found that the bi-orthogonal wavelet kernel can achieve better classification results. Texture feature retrieval is an on-going topic and the objective of which is to find a better texture analysing method that could achieve better retrieval results.

1.1.3. Related work on texture-based feature image retrieval

Several content-based image retrieval systems utilize texture feature sets to aid image retrieval. Some well known content-based image retrieval systems include the IBM QBIC[29], Virage system[30], Photobook[31], Netra[32] etc. All provide texture feature searching. The IBM QBIC system is the first that provides low-level texture similarity search. Its texture features were based upon the one which was proposed by

Tamura et al[29]. The texture properties including coarseness, directionality and contrast. It indices texture features of the whole image and are indexed manually divided into regions. The Virage system provides image query by texture. In the Virage system, it indexes the global texture feature of an image, rather than regions of texture within the images. The MIT's Photobook[31] uses the Wold-based feature for texture analysis. It defines texture into three components, periodicity, directionality and randomness. In the Photobook system, it provides local region search using textures. The system provides scene retrieval within a picture such as building, cloud, tree, etc. Manjunath and Ma[32] used Gabor wavelets-based representations to extract texture information in their Netra system. It supports local region searching. Manjunath *et al*[33, 34] applied the method to retrieve images based upon the detection of various features of the earth's terrain for satellite image retrieval in the Alexander project[33, 34].

1.2. Overview of the texture feature retrieval system

A texture-based feature extraction and retrieval system basically includes three modules: texture extraction, feature representation and similarity measurement. In texture extraction, a model is used to describe the characteristics of the texture images. Feature representation is concerned with the extraction of important features from the model for texture description. Similarity measurement, which is based on the features found from the feature representation, is then used for texture retrieval. In this section, we will provide an overview of each component in the system.

1.2.1. Texture extraction

Techniques for texture analysis have been investigated for a long time in areas such as image segmentation and retrieval. Texture extraction is concerned with finding a model which can describe natural texture images. The model should confirm with the human viewpoint on most of the texture patterns. There are basically three approaches to texture description, namely the statistical, structural and spectral approaches. More detailed discussion will be given in Section 1.3.

1.2.2. Feature representation

A good feature representation is needed to increase retrieval accuracy. In feature representation, the purpose is to use a low dimensional feature vector to describe important details of a pattern. In [35], it was found that two texture patterns of randomly oriented lines are perceived as similar, if the mean and the standard derivation of the line orientations are the same regardless of the actual distribution. E.g. for analysis using wavelets or Gabor features, the frequently used feature representation parameters are simply the mean and the standard derivation. Furthermore, this simple representation helps solve the translation variant problem that might occur in similarity measurement.

1.2.3. Similarity measurement

The objective of the similarity measurement is to compare the difference between features in two patterns and find out how similar the two patterns are. In our retrieval model, we used a distance measure for similarity measurement. This allows high dimensional vectors to be combined to give a scalar which is used in database image query. A common similarity calculation is the distance measure. Let f^a and f^b be the M-dimensional feature vectors of texture patterns a and b respectively. The distance between a and b is defined as,

$$dis(a,b)=\sum_{m=1}^{M}\left|\frac{f_m^a-f_m^b}{\alpha(f_m)}\right| \qquad (1.2\text{-}1)$$

where f_m^x denotes the m-th element in the feature vector x and $\alpha(f_m)$ is the standard derivation of the m-th element. The standard derivation $\alpha(f_m)$ is used to normalize the individual feature components over all query images. A small distance value $dis(a,b)$ indicates that the two patterns a and b are similar.

1.3. Texture-based feature extraction

Texture-based feature extraction is concerned with finding a suitable model which can precisely describe various natural textures. The model is defined based on human vision or from psychophysiology conclusions. In this section, we summarize different texture analysis methods and classify them into three approaches, namely the statistical, structural/statistical and spectral.

1.3.1. Statistical approach

The statistical approach characterizes texture in terms of local statistical measures, either by the statistics of pixel values or by a stochastic model. It indicates the early human viewpoint on texture pattern. Examples of this approach include auto-correlation[35], run length[36], and simultaneous auto-regressive models(SAR)[37]. Many papers gave good discussions on the area of texture modeling using Markov-type random field models[38] during 1980's. Statistical approach is suitable for analysis on random looking textures. In general, these methods are good at modeling random patterns such as sand and cloud but not so suitable for modeling highly structured patterns such as brick walls and nettings. To satisfy the human vision for similarity searching, a texture model should contain some degree of structures on texture.

1.3.2. Structural/Statistical approach

The structural approach assumes that textures are composed of texture primitives. The texture is produced by the placement of these primitives according to certain placement rules. Example of texture methods using structural analysis can be found in **[39, 40, 41]**. A pure structural analysis is incapable of capturing or generating randomness. Researchers combined both statistical and structural approaches to model textures. Examples of the texture properties observed by Rao and Lohse**[42]** conclude that a texture usually contains three independent patterns of repetitivity, directionality and granularity. Analogy models have been made to combine structural/statistical characterization of texture models in low-level human vision. Liu and Picard[22] extended the 2-D Wold decomposition, proposed by Francos *et al*[**43, 44**], to develop a 2D Wold-based feature model. The Wold-based feature model describes texture patterns in terms of "periodicity, directionality and randomness" which possess properties such as "repetition, directionality and granularity" as defined by Rao and Lohese. It is proven that the model can describe human vision on a statistical/structural approach. In general, the statistical/structural approach provides a systematical viewpoint of natural texture image. However, the main deficiency of this approach is the definition of the texture pattern which may not easily cover the nature of texture. Moreover, formulating the model into machine version is non-trivial. For example, the Wold-based model uses complicated classification and jointing procedures respectively to distinguish properties of various texture features or combine them.

1.3.3. Spectral approach

Studies in psychophysiology have suggested that the human is able to analyse texture images by making use of multi-channels, frequency components and orientations.

Campbell and Robson[45], who performed psychophysiology experiments using various gating patterns, concluded that the human visual system decomposes an image into filtered images of various frequencies and orientations. De Valsois *et al.*[46] have also arrived at a similar conclusion for the human vision on texture. These psychophysiology studies have motivated recently texture analysis using spectral approaches.

Spectral approach models the texture features into localized spatial-frequency contents. The commonly used spectral approach for texture analysis is wavelet-based filtering. In general, wavelet-based filtering attains a joint resolution in spatial and spectral components that are bounded by the uncertainty principle[47, 48, 49]. By spacing the filters at octave-band distances, the wavelet filter bank provides a trade-off between space and frequency resolutions. In particular, the Gabor filter is able to model the receptive fields sufficiently for texture discrimination experiments. The Gabor filter allows selection of directional and spectral features from texture patterns. Some excellent work on applying the Gabor-based filtering for texture analysis was done by Manjunath and Ma[24]. The Gabor wavelets, which place the set of Gabor filtering bank in octave band distance, extract texture features in multi-orientations and multi-resolutions from image patterns. High retrieval rate on Brodatz texture database[28] was reported in [19], indicating that the Gabor wavelet is able to analyse a texture image in multi-channels, frequencies and orientations. The main deficiency of the Gabor filter is its long analysing time, which hinders real-time applications. Another commonly used spectral approach is the sub-sampling wavelets. The sub-sampling wavelets also allows multi-channels and multi-orientation decompositions and is suitable for texture analysis[25, 26, 27] due to its half sampling structure and separable wavelet transforms. Therefore, it can achieve high speed processing for texture analysis as compared to Gabor wavelets. However, this scheme is not translation invariant and this affects its performance in texture analysis.

1.4. Focus: The over-complete wavelet scheme

In this dissertation, we present a new model for texture feature indexing and retrieval. The objective of this dissertation is to introduce a new texture analysis approach in spatial domain. We suggest using the over-complete wavelet scheme proposed by Mallat and Zhong[9], due to its low analysing time and translation invariant property. The method agrees with the studies of psychophysiology for approaches using multi-channels, frequencies and orientations analysis of the visual image. However, instead of using the wavelet kernel in [9], we have derived a new wavelet kernel which corresponds to the second deviative of the Spline family and is equivalent to the Laplacian of Gaussian filter[50]. We will prove that the new wavelet kernel is more suitable for texture analysis as compared to the original wavelet kernel. The proposed wavelet kernel allows perfect reconstruction, i.e., a stable inverse transform can always be obtained. We also present a new representation model for wavelet-based texture feature, which can significantly improve the accuracy of the retrieval rate. A comprehensive set of experiments on the Brodatz texture database is carried out to compare the retrieval accuracy of our proposed wavelet representation, the sub-sampling wavelet scheme and the Gabor wavelet feature[24]. Lastly, the retrieval results of different features under various Gaussian noise levels are examined. In particular, it is demonstrated that the over-complete wavelet scheme can achieve a more robust performance than the sub-sampling wavelet scheme under different noise levels. We summarize the contribution of this dissertation as follows,

- A new texture analysing feature using the over-complete wavelets is proposed. A new wavelet kernel, which is proven to be suitable for texture analysis, is derived.
- A new texture representation which can significantly improve the retrieval accuracy of texture features for both the over-complete and the sub-sampling wavelet schemes is proposed.
- Comparisons on different texture-based feature retrieval methods are given. Experimental results indicate that our proposed method can achieve excellent retrieval rate with a low feature analysing time. It is thus suitable for real-time content-based retrieval applications.
- An analysis on the retrieval performance of various texture-based feature retrieval methods under different Gaussian noise levels has been performed. Results indicate that the over-complete wavelet scheme is robust to noise, due to its translation invariant property.

Furthermore, we have also analysed the wavelet transforms[52-55] and proposed some fast algorithms for the LOG-based over-complete wavelet. By exploiting the redundancy between the bandpass and lowpass FIR filters in the wavelet transforms, we have found that it is possibly to use an alternative implementation method to reduce the redundancy. Therefore, we propose a spatial implementation for this LOG-based wavelet transform. Compared to the conventional filtering approach, the spatial implementation can significantly improve the performance of wavelet transforms. The following is a list of contributions about these new spatial approaches.

- A new spatial implementation for both 1D and 2D LOG-based over-complete wavelet transform is proposed;

- the computational complexity is reduced significantly using the spatial approach as compared to the filtering approach; and
- the boundary artifacts are eliminated completely during multi-resolution wavelet transforms without extra processing. Perfect reconstruction is thus always obtained easily, even at the boundary.

1.5. Thesis organization

The rest of this dissertation is organized as follows. Chapter 2 reviews different wavelet algorithms for texture feature extraction. These include the sub-sampling wavelet schemes[51], and Gabor wavelets[24]. A discussion on the advantages and disadvantages of two wavelet schemes for texture analysis is given. Also, we discuss the motivation of using over-complete wavelets for texture analysis. Chapter 3 presents a new texture feature extraction and retrieval scheme using Laplacian of Gaussian-based over-complete wavelet for feature extraction. A comparative study with other texture analysing features, including the Gabor wavelet feature and various sub-sampling wavelets, is also presented. A fast spatial implementation for both 1D and 2D Laplacian of Gaussian based wavelet will be discussed in Chapter 4. Conclusion and future directions will be discussed in Chapter 5.

CHAPTER 2

2. Texture feature extraction using wavelet – based approach

In this chapter, we give an overview of two commonly used spatial approaches for multi-frequency, orientation and channel texture pattern analysis. They include the Gabor wavelets (Gabor filtering bank) and the sub-sampling wavelet scheme. We will review the methods of wavelet transforms and discuss their advantages and disadvantages. Having made the analysis on the problems available in these two wavelet schemes, we give our motivation of this study propose use the over-complete wavelet scheme for texture analysis. We will preview the structure of over-compete wavelets and discuss how an over-complete wavelet scheme can overcome the deficiencies in Gabor wavelets and sub-sampling wavelet schemes for texture analysis and retrieval.

2.1 Two dimensional Gabor filter

Two-dimensional Gabor filters were proposed by Daugman[57] for modeling receptive field of simple cells in the visual cortex of some mammals. Gabor filters can be considered as a space-frequency technique that can be used to characterize textures. There are sufficient psychophysical data to support that a Gabor filter is efficient for texture discrimination for different frequencies and orientations. References**[58-60]** give some key ideas for using the Gabor filter for various texture analysis. Eqn.2.1-1 denotes the general 2D Gabor function in the space domain. The 2D Gabor function consists of a sinusoidal plane wave of a certain frequency and orientation modulated by Gaussian envelope as follows,

$$G(x,y,\sigma,\omega,\theta,\phi)=f(x,y,\sigma,\omega,\theta,\phi)\exp\left(\frac{(x-x_0)^2+(y-y_0)^2}{-2\sigma^2}\right) \quad (2.1\text{-}1)$$

where G represents the Gabor function. The parameters of the Gabor function include the center of filter(x_0,y_0), frequency(θ), bandwidth(ω), orientation(σ) and phase(ϕ). Center of filter(x_0,y_0) is used to locate the center of the Gaussian envelope in space domain. Orientation(σ) specifies the standard deviation along both x- and y-axes. The frequency(θ) of the sinusoidal plane wave is determined by the bandwidth(ω), and orientation(σ) is the angle of orientation, and ϕ is the phase of the plane wave. Function f is a sinusoidal wave function is given by,

$$f(x,y,\sigma,\omega,\theta,\phi)=\sin(\omega[x\cos\theta-y\sin\theta]+\phi) \quad (2.1\text{-}2)$$

Gabor filter is a frequency and orientation selective filter. E.g. when the phase $\phi=0$, the Fourier transform of the Gabor function, $F(u,v)$, is given by,

$$F(u,v)=A\left\{\exp\left(-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right)+\exp\left(-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right)\right\}$$

where σ_u and σ_v specify the standard deviation along u- and v-axes in the spatial domain, respectively. Note that standard deviations are related to σ_x and σ_y by $\sigma_u = {}^{1}\!/_{2\pi\sigma_x}$ and $\sigma_v = {}^{1}\!/_{2\pi\sigma_y}$. σ_x and σ_y are the standard deviations of horizontal and vertical directions in spatial domain. A is the magnitude of the Gabor function which is equal to $A = 2\pi\sigma_x\sigma_y$. This function gives two lobes in the spatial domain, one centered at u_0 and other at $-u_0$.

2.1.1. Gabor filter bank design for texture analysis

Due to the frequency and the orientation selectivity properties of the Gabor filter, much work was proposed to using a bank of Gabor filters to extract the multi-frequencies, orientations and channels information in texture analysis[**23-24, 58-63**]. Manjunath and Ma[**24**] proposed a design strategy of Gabor filter bank for texture feature retrieval. They designed a set of Gabor filters in the spatial domain which aimed at minimizing the redundant information between individual filters. We call the design of Gabor filtering bank as Gabor wavelets in the rest of the dissertation. Eqn.2.1-6 to Eqn.2.1-8 denote the general equations to design Gabor wavelets,

$$a = \left(U_h / U_l \right)^{-\frac{1}{S-1}} \qquad (2.1\text{-}6)$$

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}} \qquad (2.1\text{-}7)$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln\left(\frac{\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}} \qquad (2.1\text{-}8)$$

where U_h and U_l denote the lower and upper center frequencies of interest, respectively, in the spatial domain. Assume that k and s are the number of orientations and the number of scales in the multi-resolution decompositions respectively. Parameters σ_u and σ_v are the standard derivations of the filter along the abscissa and ordinate in frequency domain respectively. These two parameters give the shape of the filter and minimize the redundancy between filters. They ensure that half-peak magnitude support of the filter response in frequency domain with least overlapping to each other. Figure 2.1 depicts the Gabor filtering banks design in spatial domain. Experiments[24] have indicated that four levels and six orientations of the Gabor wavelet transform can achieve the best retrieval results. For texture feature representation, the absolute mean and standard derivation of each channel of the image should be used. As a consequence, 48 dimensional feature vectors are obtained from a texture image. Distance measure is used, as depicted in eqn.1.2-1, to measure the similarity between two images.

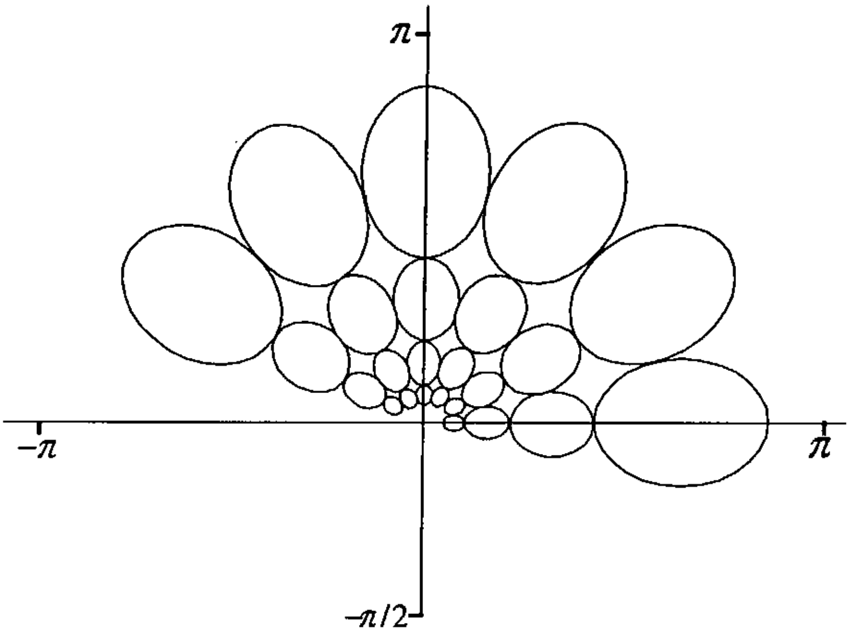


Figure 2.1, The Gabor filter banks design in spatial domain. It shows the design strategic for the four levels and six orientations of Gabor filter bank.

2.1.2. Discussion of Gabor wavelets

Gabor wavelets allow multi-frequencies, orientations, and channels for texture analysis which confirm with the human vision system. It achieves very good retrieval results on the entire Brodatz texture database[28] as reported in [24], compared to other texture models such as statistical and spatial approaches. The retrieval rate using Gabor Wavelets is about 74.13% on the entire Brodatz database. However, the main deficiency of Gabor wavelets is its lengthy analysing time. It takes about 9.5 seconds to analyse a grey image with 128x128 pixels[1]. The reason for long analysing time is because of its non-separable properties and thus it has to analyse the pattern in the frequency domain. To implement the Gabor wavelets for texture analysis, it is necessary to transform the whole image pattern into the frequency domain. To extract the texture feature, the filtered image has to be transformed back to the spatial domain. As a result, the analysing time is greatly increased. The long waiting time of texture analysis makes it unsuitable for real-time applications.

2.2. Sub-sampling wavelet schemes for texture analysis

Another common spatial approach for texture analysis[25-27, 64-65] is dyadic sub-sampling wavelet [47, 51]. Compared to Gabor wavelets, the advantages of sub-sampling wavelet scheme is its sub-sampling properties during wavelet transforms and separable properties which can analyse local space and frequency relation in spatial domain. Rather than analysing texture patterns in the spatial domain, the texture features are extracted from spatial domain to convolute with FIR filters, which can greatly reduce the computational complexity. In the following section, we discuss the dyadic sub-sampling wavelet transform and show how the method is used for multi-frequencies, orientations and channels texture analysis and retrieval.

2.2.1. Sub-sampling wavelet schemes for texture analysis

We start from the one dimensional dyadic sub-sampling wavelet transform. Dyadic sub-sampling wavelet transform performs the decomposition of a signal onto the family of functions by the mother wavelet,

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n) \tag{2.2-1}$$

[1] The timing experiment reported in [24] are carried on a SUN SPAR20 machine with one processor. The Gabor wavelets were implemented by MATLAB.

where m and n are the scale and the translation parameters respectively. And t denotes the time parameter. The mother wavelets $\psi_{m,n}(t)$ generated from a prototype mother wavelet by dilation and translation has to satisfy the condition that $\int \psi(t)dt = 0$. This implies at least some oscillations. The mother wavelet is constructed from the two scale equations,

$$\phi(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} h_0(k)\phi(2t-k) \tag{2.2-2}$$

$$\psi(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} h_1(k)\phi(2t-k) \tag{2.2-3}$$

where $h_0(k)$ and $h_1(k)$ are coefficients of the lowpass and bandpass filters respectively. In the case of the dyadic discrete wavelet transform(DWT), filters, $h_0(k)$ and $h_1(k)$ can be used for DWT computation instead of the explicit forms, $\phi(t)$ and $\psi(t)$, respectively. For the *j-th* level of dyadic wavelet transform, the signal dyadic discrete approximation lowpass details, $f_{2^j}(x)$, and dyadic bandpass details, $d_{2^j}(x)$, can be computed by,

$$f_{2^{j+1}}(x) = \sum_{k=-\infty}^{\infty}\left[f_{2^j}(k)h_0(x-2k) + d_{2^j}(k)h_1(x-2k)\right] \tag{2.2-4}$$

It is well known in the subband filtering community that linear phase, such as symmetry and orthogonality, is not possible for wavelet transform filters. However, the symmetry of decomposition filters can be achieved, by relaxing the orthogonality condition and introducing the biorthogonal bases, where the scaling function and the wavelet used for decomposition are different from those used for synthesis. In such a scheme, instead of two filters h_0 and h_1, we have four filters h_0, h_1, g_0 and g_1. Hence, Eqn.2.3-2 and Eqn.2.3-3 have to be written as,

$$f_{2^j}(x)=\sum_{k=-\infty}^{\infty} f_{2^{j+1}}(k)h_0(k-2x) \qquad (2.2\text{-}5)$$

$$d_{2^j}(x)=\sum_{k=-\infty}^{\infty} f_{2^{j+1}}(k)g_1(k-2x) \qquad (2.2\text{-}6)$$

$$f_{2^{j+1}}(x)=\sum_{k=-\infty}^{\infty}\left[f_{2^j}(k)g_0(x-2k)+d_{2^j}(k)h_1(x-2k)\right] \qquad (2.2\text{-}7)$$

where $h_1(n)=(-1)^n h_0(1-n)$ and $g_1(n)=(-1)^n g_0(1-n)$. Figure 2.2 and Figure 2.3 depict the forward and the inverse dyadic sub-sampling transforms respectively.

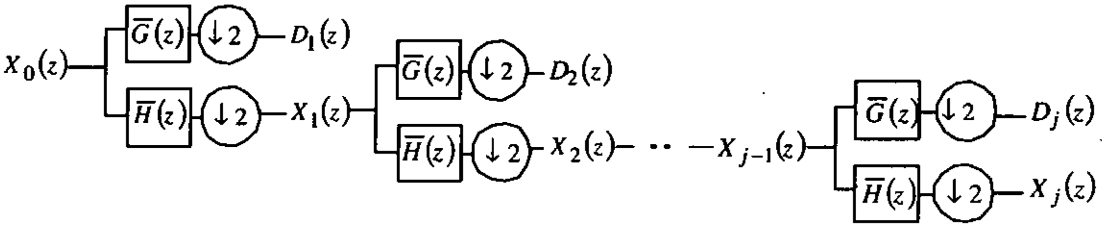


Figure 2.2, *j-th* level of the forward transform of dyadic sub-sampling wavelets.

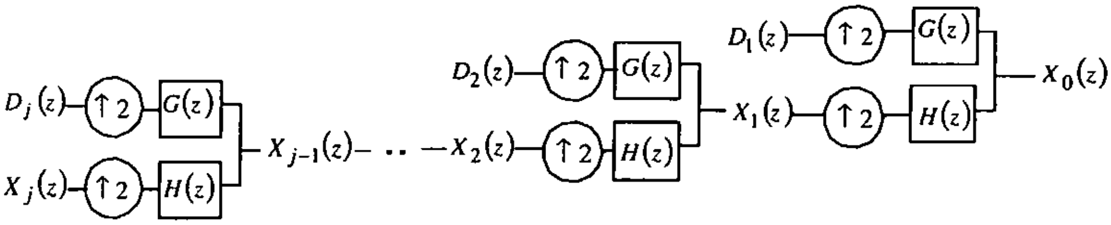


Figure 2.3, *j-th* level of the inverse transform of dyadic sub-sampling wavelets.

The simplest way to compute two-dimensional dyadic wavelet transform is to apply one-dimensional transforms over image rows and columns separately. Thus, the image decomposition is obtained by using separable filters along the abscissa and ordinate and the same algorithm as in 1D case. The transform is characterized by 2D scaling function,

$$\phi(t_x,t_y)=\phi(t_x)\phi(t_y) \tag{2.2-8}$$

and by three wavelets which are given as follows,

$$\psi_1(t_x,t_y)=\phi(t_x)\psi(t_y) \tag{2.2-9}$$

$$\psi_2(t_x,t_y)=\phi(t_x)\psi(t_y) \tag{2.2-10}$$

$$\psi_3(t_x,t_y)=\psi(t_x)\psi(t_y) \tag{2.2-11}$$

Eqn.2.2-9, 2.2-10 and 2.2-11 represent the bandpass transform of the x-axis, y-axis and the diagonal directional transform respectively. The wavelet transform decomposes an image with an overall scale factor of four and produces one low-resolution sub-image and three wavelet coefficient sub-images in each level. The dyadic sub-sampling wavelet transform can be used for multi-frequency and orientation in texture analysis. A two-dimension frequency spectrum of the forward decomposition is depicted in Figure 2.4.

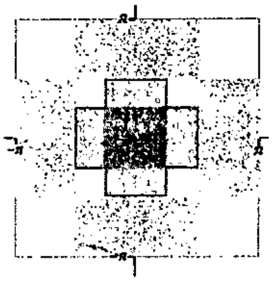


Figure 2.4, 2D dyadic sub-sampling wavelet transform in spatial domain.
It shows the 2 levels of sub-sampling wavelet transforms.

2.2.2. Texture retrieval using sub-sampling wavelet schemes

Texture feature is extracted from each filter channel in the wavelet transforms. Many researchers proposed using various feature representation and similarity measurement methods for texture retrieval. The most common approach is to characterize texture feature using statistical representation to calculate each of the channel energy, such as the mean[65], standard deviation[25], etc. For similarity retrieval, many measures were suggested, such as Euclidean distance[26], Mahalanobis distance[27], etc. Many experiments show that sub-sampling wavelets can achieve a high rate on texture classification[25-27, 65].

2.2.3. Discussion of the sub-sampling wavelet schemes for texture extraction

Sub-sampling wavelet scheme can also extract texture properties in multi-frequencies, orientations, and channels which agree with the human vision system on texture. Compared to Gabor wavelets, the sub-sampling wavelet scheme can greatly reduce the analysing time. It is because of the separable properties of wavelet transform and sub-sampling operations in each level of the transform. For each level of decomposition, three quarters of sampling filtered image are ignored. It directly decreases the processing time in each level of transform. On the other hand, sub-sampling operation also causes the translation variant problem. Due to the sub-sampling operation, many samples in pattern are deleted during each of transform level, and pattern details are lost in each channel as well. Thus it reduces the precision needed to obtain the texture feature for representation. Consequently, it affects directly the retrieval results.

2.3. Texture feature extraction and retrieval using the over-complete wavelet transform

One of the objectives of this research is to find a texture retrieval method which can achieve low analysing time but high texture retrieval accuracy. Due to the deficiencies in the previous two wavelet-based transforms, the motivation of this research is to find an appropriate spatial approach which can achieve low analysing time and permits multi-frequencies, orientations and channels for texture analysis. Furthermore, the wavelet transform must be translation invariant without details lost during each level of transform. Therefore, we suggest using the over-complete wavelets, which were proposed by Mallat and Zhong[9], for texture analysis. The advantages of the over-complete wavelet scheme are translation invariant during multi-resolution transforms and low processing time. Furthermore, the wavelet transform has compact support and separable properties which allow processing data in the spatial domain.

2.3.1. The over-complete wavelet transform and its use for texture analysis

The over-complete wavelet scheme used by Mallat & Zhong[9] is obtained by assuming that the wavelet kernel is derived from the deviative of a Spline function. For one dimensional wavelet transform, let $\theta(x)$ denote the Gaussian smoothing function, whose integral is equal to 1. It converges to 0 at infinity and is k times differentiable. Then,

$$\psi(x)=\frac{d^k\theta(x)}{dx^k} \quad k\in Z \tag{2.3-1}$$

where $\psi(x)$ defines the wavelet kernel whose integral is always zero. The dyadic wavelet representation is defined as

$$\psi_{2^i}(x)=\frac{1}{2^i}\psi\left(\frac{x}{2^i}\right) \quad i\in Z \tag{2.3-2}$$

where $\psi_{2^i}(x)$ is the wavelet function dilated by a dyadic scaling factor 2^i. Suppose a signal $f(x)$ is wavelet transformed, the lowpass $S_{2^i}f(x)$ and the bandpass $W_{2^i}f(x)$ filter responses are given respectively as

$$S_{2^i}f(x)=f*\phi_{2^i}(x) \text{ and } W_{2^i}f(x)=f*\psi_{2^i}(x) \tag{2.3-3}$$

where $*$ denotes convolution and

$$\phi_{2^i}(x)=\frac{1}{2^i}\phi\left(\frac{x}{2^i}\right) \tag{2.3-4}$$

Similar to the sub-sampling wavelet scheme, the 2D over-complete wavelet transform is also separable. Let us denote the wavelet kernel in x and y directions respectively at scale s as

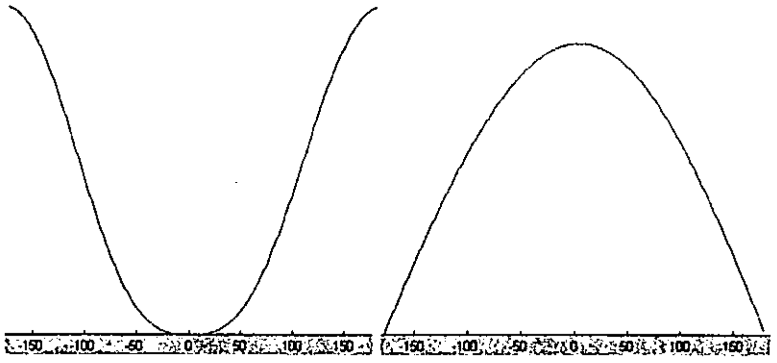
$$\psi_s^1(x,y) \text{ and } \psi_s^2(x,y) \tag{2.3-5}$$

For a 2D image $f(x,y)$, the wavelet transform is defined as,

$$S_{2^j} f(x,y) = f * \phi_{2^j}(x,y) \tag{2.3-6}$$

$$W_{2^j}^1 f(x,y) = f * \psi_{2^j}^1(x,y) \text{ and } W_{2^j}^2 f(x,y) = f * \psi_{2^j}^2(x,y) \tag{2.3-7}$$

where $S_{2^j} f(x,y)$ is the lowpass filter response, $W_{2^j}^1 f(x,y)$ and $W_{2^j}^2 f(x,y)$ are the vertical and the horizontal filter responses respectively. Figure 2.5 shows the frequency spectra of the bandpass and the lowpass filters.



(a) **(b)**

Figure 2.5, 1D frequency spectrum of the (a) bandpass (b) lowpass of the quadratic over-complete Canny based kernel.

The original wavelet transform contains both vertical and horizontal filter responses. It is because these two directions have already received sufficient information to reconstruct an image. However, since we wish to apply the wavelet scheme for texture analysis, we have to include more information on its directions for texture pattern analysis. In order to provide a better analysis on line orientation in texture analysis, we include a third directional transform, $W_{2^j}^3 f(x,y)$ which is the response by applying the bandpass filter in both x and y directions, i.e.,

$$W_{2^j}^3 f(x,y) = f * \psi_{2^j}^1(x,y) * \psi_{2^j}^2(x,y) \tag{2.3-8}$$

In summary for texture analysis, a single level wavelet transform using the over-complete wavelets produces four outputs: $S_{2^j} f(x)$, $W_{2^j}^1 f(x,y)$, $W_{2^j}^2 f(x,y)$ and $W_{2^j}^3 f(x,y)$.

2.3.2. The Canny-based wavelet kernel

Mallat and Zhong have proposed a Canny-based edge operator for the wavelet kernel in the over-complete wavelet transform. It is found that the resultant modulus maxima correspond to the multi-scale edges in a signal. As points of sharp variations are one of the most important features for analysing signal properties, the over-complete representation using this kernel is able to provide a meaningful signal characterization.

In this section, we will briefly summarize this Canny-based wavelet kernel. The Canny-based wavelet kernel basically corresponds to a first deviative edge detector, i.e., k is set to one in eqn.2.4-1. As the Spline function can be written as

$$\theta(\omega) = \left(\frac{\sin \omega/4}{\omega/4} \right)^{2n+2} \tag{2.3-9}$$

The wavelet kernel and the scaling function[9] can be shown to be

$$\psi(\omega) = -\omega^2 \left(\frac{\sin \omega/4}{\omega/4} \right)^{2n+1} \tag{2.3-10}$$

$$\phi(\omega)=\left(\frac{\sin{\omega/2}}{\omega/2}\right)^{2n+2} \tag{2.3-11}$$

where n is the order. If n is 2, it corresponds to the commonly used quadratic Spline wavelet. Figure 2.5 shows the frequency spectrum of the bandpass and lowpass filters. For implementation, the over-complete wavelet transform can be computed by performing purely discrete convolutions with the input signal. All are FIR structures. The lowpass and the bandpass filters are given respectively as,

$$H(\omega)=e^{i\omega/2}\left(\cos\frac{\omega}{2}\right)^{2n+1} \tag{2.3-12}$$

and

$$G(\omega)=-4ie^{i\omega/2}\left(\sin\frac{\omega}{2}\right) \tag{2.3-13}$$

The expressions for the reconstruction filters are given as follows,

$$K(\omega)=\frac{1-|H(\omega)|^2}{G(\omega)} \tag{2.3-14}$$

and

$$L(\omega)=\frac{1+|H(\omega)|^2}{2} \tag{2.3-15}$$

Figure 2.6 and Figure 2.7 show the 1D forward and inverse wavelet transforms. Figure 2.8 and 2.9 show the 2D forward and inverse wavelet transform. Table 2.1 shows the filter coefficients of Spline order for $n=0,\ 1,\ 2$ in Eqn.2.3-12 of the Canny-based over-complete wavelet transform.

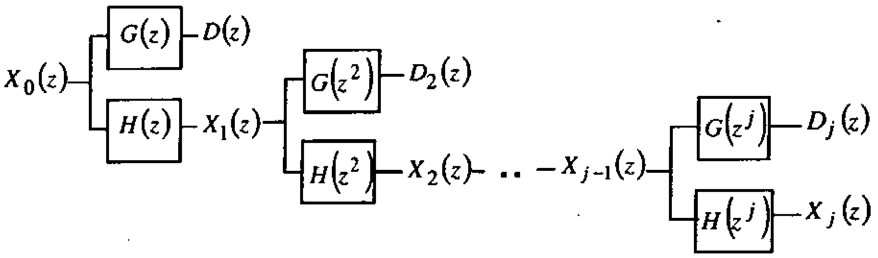


Figure 2.6, *j*-th level one dimension forward over-complete wavelet transform.

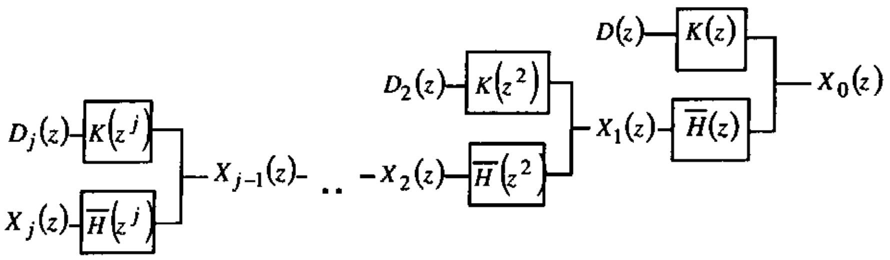


Figure 2.7, *j*-th level one dimension inverse over-complete wavelet transform.

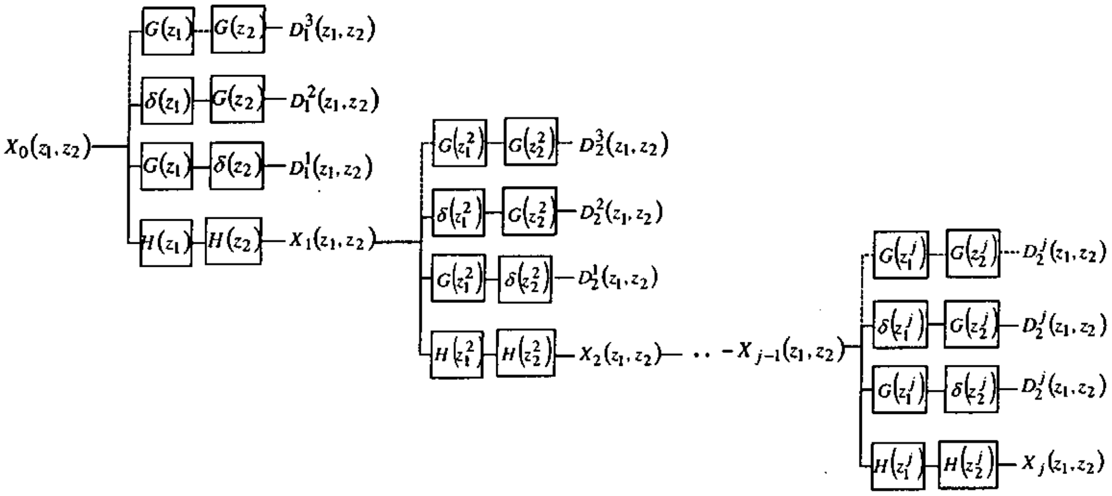


Figure 2.8, *j*-th level two dimension forward over-complete wavelet transform. Dotted lines indicate the third directional transform which is suggest for texture analysis.

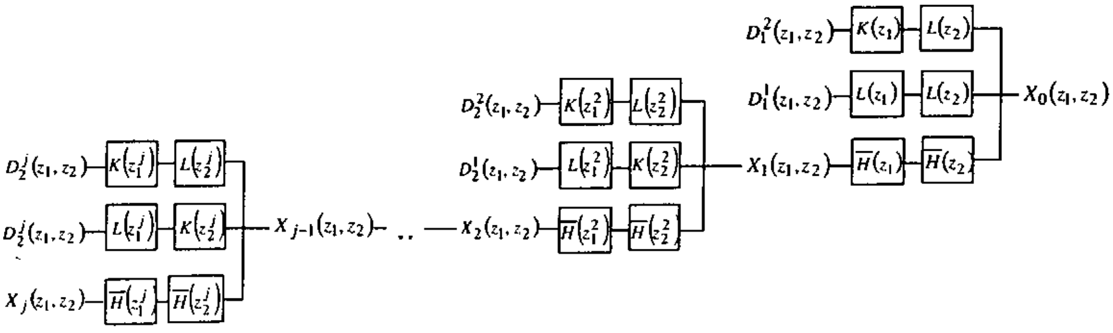


Figure 2.9, *j*-th two-dimensional inverse over-complete two dimensional wavelet transform.

Spline Wavelet	n=1				n=2				n=3			
k	G(k)	H(k)	K(k)	L(k)	G(k)	H(k)	K(k)	L(k)	G(k)	H(k)	K(k)	L(k)
-6												
-5											1/2048	1/2048
-4											11/2048	10/2048
-3							1/128	1/128			7/256	45/2048
-2							7/128	6/128		1/32	11/128	120/2048
-1			-1/8	1/8		1/8	11/64	15/128		5/32	193/1024	210/2048
0	-2	1/2	-1/8	6/8	-2	3/8	-11/64	84/128	-2	5/16	-193/1024	1276/2048
1	2	1/2		1/8	2	3/8	-7/128	15/128	2	5/16	-11/128	210/2048
2						1/8	-1/128	6/128		5/32	-7/256	120/2048
3								1/128		1/32	-11/2048	45/2048
4											-1/2048	10/2048
5												1/2048
6												

Table 2.1, Filter coefficients of the Canny-based wavelets kernel with Linear, quadratic and Cubic Spline (n=1,2 and 3).

2.3.3. Discussion on the over-complete wavelet for texture feature extraction

Similar to the previous two wavelet schemes, the over-complete wavelet can be used for multi-frequencies, orientations and channels in texture analysis. The benefit of using over-complete wavelet transform is its separable properties and thus it can perform the transform in the spatial domain. It overcomes the problems in frequency domain as seen in the Gabor wavelets. Moreover, it is translation invariant because there is no sub-sampling. Feature details can be retained in filtered images and thus help to keep the information for texture feature extraction. Chapter 3 will contains more details for texture representation.

2.4. Summary

In this chapter, we have discussed different spatial approach using multi-frequencies, orientations and channels for texture analysis. These schemes include Gabor wavelets and sub-sampling wavelet schemes. Both analysing methods confirm with the human vision system on texture. However, each scheme has its deficiencies for texture analysis. For examples, the Gabor wavelets need high computational complexity and sub-sampling has the translation variant problems. Therefore, we should investigate a new texture analysis method that uses low computational complexity and can solve the translation variant problem. The new method should confirm with the human vision system on multi-frequencies, orientation and channels for texture analysis. We have suggested the over-complete wavelet scheme for texture analysis. The wavelets scheme is separable in the spatial domain and thus the computational complexity is low. Also, it is translation invariant because there is no down-sampling during wavelet decomposition. We have discussed the advantages of the over-complete wavelets, compared them to the Gabor wavelet and the sub-sampling wavelet scheme for texture analysis. Furthermore, rather than using Canny-based over-complete transform[9], we propose to use the Laplacian of Gaussian(LOG) based over-complete wavelet which is more efficient for texture analysis and retrieval. The LOG-based over-complete wavelet is derived from a Spline order which has similar structure as the Canny-based over-complete wavelet. This wavelet kernel is more efficient for analysing the line information in texture pattern. In Chapter 3, we will present texture feature extraction and retrieval using the over-complete wavelet technique. Also, the details of LOG-based wavelet kernel its use for texture retrieval will be given. Comparative texture retrieval experiments using various sub-sampling wavelets and Gabor wavelets will also be presented.

CHAPTER 3

3. Texture-based feature extraction and retrieval using the Laplacian of Gaussian based over-complete wavelet transforms

In this chapter, we introduce a new texture analysis method which requires low feature analysing time but possesses high retrieval accuracy. We suggest using the over-complete wavelet scheme proposed by Mallat and Zhong[9], due to its low analysing time and translation invariant property. However, instead of using the wavelet kernel in [9], we derive a new wavelet kernel which corresponds to the second deviative of the Spline family and is equivalent to the Laplacian of Gaussian (LOG) filter. We prove that the new wavelet kernel is more suitable for texture analysis as compared to the original wavelet kernel. The proposed wavelet kernel allows perfect reconstruction, i.e., a stable inverse transform can always be obtained. We also present a new representation model for wavelet-based texture feature which can significantly improve the accuracy of the retrieval rate. A comprehensive experiment on the Brodatz texture database is carried out to compare the retrieval accuracy of our proposed wavelet representation, the sub-sampling wavelet scheme and the Gabor wavelet feature[24]. Lastly, the retrieval results of different features under various Gaussian noise levels are examined. In particular, it is demonstrated that the over-complete wavelet scheme can achieve a more robust performance than the sub-sampling wavelet scheme under various noise levels.

3.1. The proposed second deviative-based (Laplacian of Gaussian-based) wavelet kernel

3.1.1. Deficiency in first deviative-based wavelet kernel

Mallat and Zhong[9] chose the wavelet kernel to be the first deviative of a smoothing function as discussed in Chapter 2.3. It was shown to be equal to the Canny operator for edge detection[67]. They pointed out that the Canny detector provides a large variance response for local extrema in the signal and a low variance response on other parts of the signal. On the other hand, Marr-Hidreth described a zero-crossing-based edge detection method which is derived from the Laplacian of Gaussian filter[68]. The filter detects the local extrema from the zero-crossing point between the maxima and the minima. We have examined both filter properties with different input as shown in Table 3.1.

As observed from Table 3.1, the input signal, $G_2(x)$, has one data sample 'thicker' than $G_1(x)$. Using only mean and standard deviation for signal representation, it can be seen that ∇G_1 and ∇G_2 from the Canny filter have exactly the same representation. On the other hand, the variance of $\nabla^2 G_2$ is different from that of $\nabla^2 G_1$ on using the Laplacian of Gaussian filter. Different from the requirement in edge detection, texture analysis needs to emphasize line information. From this example, it is clear that the Laplacian of Gaussian filter is more sensitive to the thickness of line than the first deviative-based Canny filter. It should have better performance in distinguishing the thickness of lines in texture pattern.

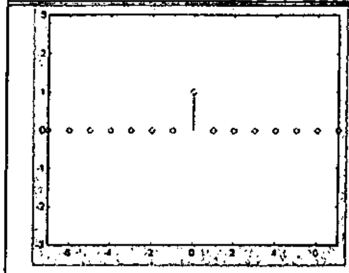
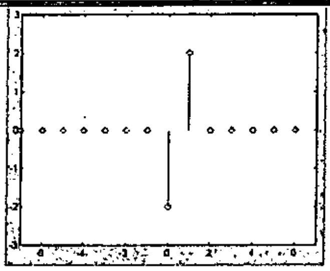
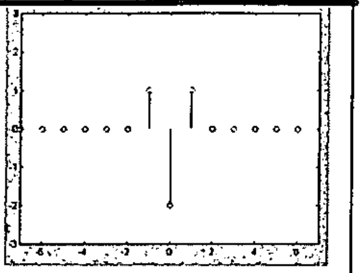
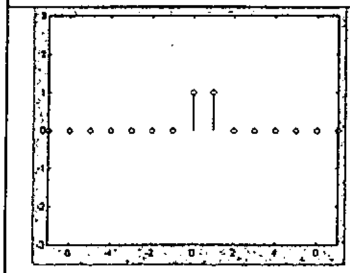
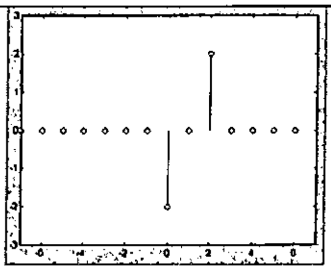
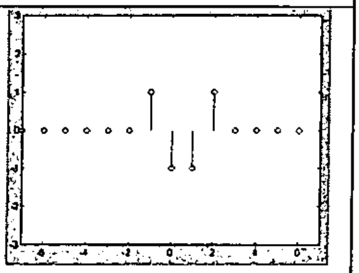
Test examples	Canny	Laplacian of Gaussian
 $G_1(x)=\delta(0)$	 $\nabla G_1 = \frac{dG_1(x)}{dx}$	 $\nabla^2 G_1 = \frac{d^2G_1(x)}{dx^2}$
 $G_2(x)=\delta(0)+\delta(1)$	 $\nabla G_2 = \frac{dG_2(x)}{dx}$	 $\nabla^2 G_2 = \frac{d^2G_2(x)}{dx^2}$

Table 3.1, Examples of input signals filtered by Canny detector and Laplacian of Gaussian.

3.1.2. The proposed 1D second deviative-based wavelet kernel

A new wavelet kernel using the Laplacian of Gaussian filter is derived in this section. In our derivation, we also consider the smoothing function $\theta(\omega)$ which is derived from the Spline wavelet family. The general expression of the Spline function is given by,

$$\theta(\omega)=e^{-i\frac{\xi\omega}{2}}\left(\frac{\sin{\omega/2}}{\omega/2}\right)^{m+1} \quad where\ \xi\begin{cases}=1 & if\ \ m\ \ is\ \ even\\=0 & if\ \ m\ \ is\ \ odd\end{cases} \qquad (3.1\text{-}1)$$

Similar to [9], we impose $\phi(\omega)$ to have the form,

$$\phi(\omega)=\prod_{p=1}^{\infty}H\left(2^{-p}\omega\right) \qquad (3.1\text{-}2)$$

where $H(\omega)$ is a 2π periodic differentiable function. The lowpass kernel $\phi(\omega)$ can be obtained from multiplication of lowpass filters H in the forward transforms as shown in Figure 2.6. The lowpass filter $H(\omega)$ satisfies the following conditions,

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 \le 1 \quad and \quad |H(0)| = 1 \tag{3.1-3}$$

The proof of eqn.3.1-3 can be found in **[69]** which are sufficient so that eqn.3.1-2 defines a smoothing function. To obtain a Laplacian of Gaussian-typed wavelet function, k is chosen to be two in eqn.2.3-1. From eqn.3.1-2, we have,

$$\phi(2\omega) = H(\omega)\phi(\omega) \tag{3.1-4}$$

To obtain the band pass signal, let us define the wavelet kernel $\psi(2\omega)$ as,

$$\psi(2\omega) = G(\omega)\phi(\omega) \tag{3.1-5}$$

where $G(\omega)$ is a 2π periodic function. To eliminate the aliasing effect of biorthogonal wavelet design during reconstruction, it needs to choose the reconstruction kernel $\chi(\omega)$ to compensate the aliasing caused by the band pass kernel. For a dyadic transformation, the kernels, including $\psi(\omega)$, $\phi(\omega)$ and $\chi(\omega)$, must satisfy the following constraint,

$$\psi(2\omega)\chi(2\omega) = |\phi(\omega)|^2 - |\phi(2\omega)|^2 \tag{3.1-6}$$

Figure 3.1 shows the diagram of idea signal reconstruction using the inverse overcomplete wavelet transform.

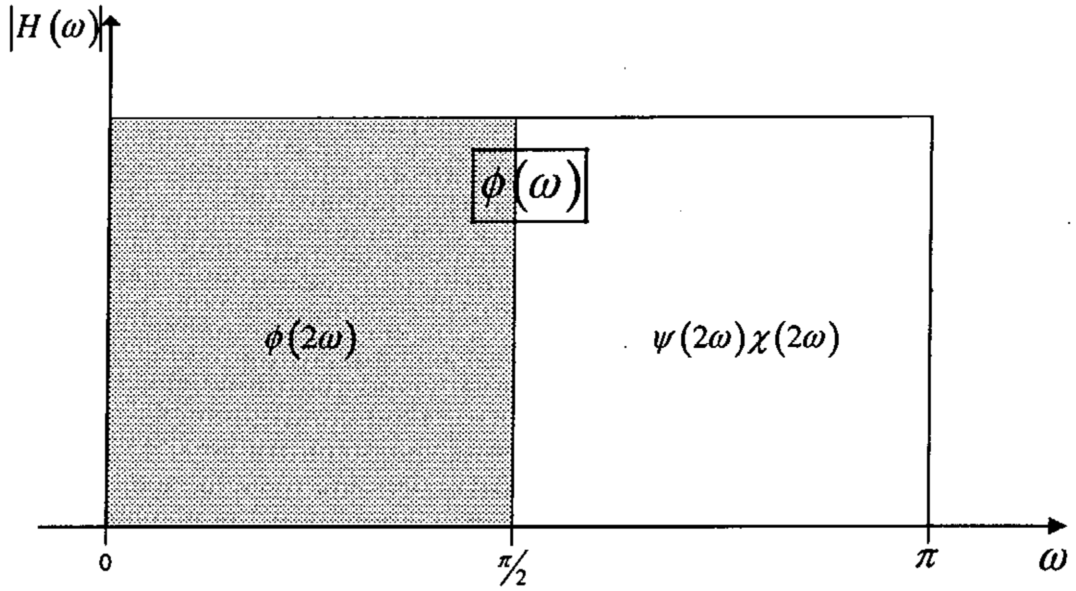


Figure 3.1, Ideal signal reconstruction using inverse overcomplete wavelet transform.

The reconstruction wavelet kernel $\chi(\omega)$ is defined as,

$$\chi(2\omega)=K(\omega)\phi(\omega) \tag{3.1-7}$$

$K(\omega)$ is a 2π periodic function for reconstruction. We want the wavelet kernel $\psi(\omega)$ to be equal to the second deviative of the smoothing function $\theta(\omega)$. This implies that $\psi(\omega)$ should have a zero of order 2 at $\omega=0$. To satisfy this requirement, we define,

$$H(\omega)=\left(\cos\frac{\omega}{2}\right)^{2n} \tag{3.1-8}$$

$$G(\omega)=-8\left(\sin\frac{\omega}{2}\right)^{2} \tag{3.1-9}$$

Hence, the corresponding reconstruction filter is defined as,

$$K(\omega)=\frac{1-|H(\omega)|^{2}}{G(\omega)} \tag{3.1-10}$$

Substituting eqn.3.1-8 and eqn.3.1-9 into eqn.3.1-10, it becomes,

$$K(\omega) = -\frac{1}{16}\sum_{k=0}^{2n-1}\left(\cos\frac{\omega}{2}\right)^{2k} \tag{3.1-11}$$

From eqn.3.1-4, eqn.3.1-5, eqn.3.1-8 and eqn.3.1-9, it can be shown that

$$\psi(\omega) = -\omega^2\left(\frac{\sin{\omega/4}}{\omega/4}\right)^{2n+2} \tag{3.1-12}$$

$$\phi(\omega) = \left(\frac{\sin{\omega/2}}{\omega/2}\right)^{2n} \tag{3.1-13}$$

Therefore, by using eqn.2.3-1 and eqn.3.1-12, we prove that the smoothing function $\theta(\omega)$ is equivalent to the Spline function defined in eqn.3.1-1, i.e.,

$$\theta(\omega) = \left(\frac{\sin{\omega/4}}{\omega/4}\right)^{2n+2} \tag{3.1-14}$$

Using the expressions in eqn.3.1-8 to 3.1-11, the filter coefficients can be obtained as shown in Table 3.2,

N	H(n)	G(n)	K(n)	L(n)
-4				0.001953125
-3			-0.0009765625	0.015625
-2	0.0625		-0.009765625	0.0546875
-1	0.25	4	-0.0478515625	0.109375
0	0.375	-8	-0.1484375	0.6367187
1	0.25	4	-0.0478515625	0.109375
2	0.0625		-0.009765625	0.0546875
3			-0.0009765625	0.015625
4				0.001953125

Table 3.2, Finite impulse response of the filters H, G, K and L that correspond to the second deviative Spline Wavelet

Note the fast wavelet algorithm discussed in [9] (c.f., Figure 2.6 to Figure 2.7 for 1D case and Figure 2.8 to Figure 2.9 for 2D case which will discussed in next section) is also suitable for the implementation of our proposed wavelet kernel. For a p-level decomposition of the dyadic wavelet transform, $2^p - 1$ zeros are inserted between each of the coefficients of the filters H, $\overline{H}$, G and K for both forward and inverse transforms. To solve the boundary problem, we can use the symmetric extension to extend the signal data at the boundary.

3.1.3. The proposed 2D second deviative-based wavelet kernel

As discussed in Chapter 2.3, the 2D transforms are simply extended from 1D transforms due to its separable nature. We defined the kernels $\psi^1(x,y)$ and $\psi^2(x,y)$ in eqn.2.3-5, for the vertical and the horizontal directions, as follows,

$$\psi^1(x,y)=\psi(x)2\phi(2y) \quad \textit{and} \quad \psi^2(x,y)=\psi(y)2\phi(2x) \tag{3.1-15}$$

Since we consider a second deviative-based operator, we define the wavelet kernels to be the partial deviative of $\theta(x,y)$, i.e.,

$$\psi^1(x,y)=\frac{\partial^2\theta^1(x,y)}{\partial x^2} \quad \textit{and} \quad \psi^2(x,y)=\frac{\partial^2\theta^2(x,y)}{\partial y^2} \tag{3.1-16}$$

where

$$\theta^1(x,y)=\theta(x)2\phi(2y) \quad \textit{and} \quad \theta^2(x,y)=\theta(y)2\phi(2x) \tag{3.1-17}$$

For the dyadic wavelet transformation, we define the 2D bandpass functions as follows,

$$\psi^1(2\omega_x,2\omega_y)=G(\omega_x)\phi(\omega_x)\phi(\omega_y) \tag{3.1-18}$$

$$\psi^2(2\omega_x,2\omega_y)=G(\omega_y)\phi(\omega_x)\phi_0(\omega_y) \tag{3.1-19}$$

The reconstruction kernels, $\chi^1(\omega_x,\omega_x)$ and $\chi^2(\omega_x,\omega_x)$ can be defined as,

$$\chi^1(2\omega_x,2\omega_x)=K(\omega_x)L(\omega_y)\phi(\omega_x)\phi(\omega_y) \qquad (3.1\text{-}20)$$

$$\chi^2(2\omega_x,2\omega_x)=K(\omega_y)L(\omega_x)\phi(\omega_x)\phi(\omega_y) \qquad (3.1\text{-}21)$$

where $L(\omega)$ is 2π periodic and satisfies the following conditions,

$$G(\omega)K(\omega)+|H(\omega)|^2=1 \qquad (3.1\text{-}22)$$

$$L(\omega)=\frac{1+|H(\omega)|^2}{2} \qquad (3.1\text{-}23)$$

The derivation of $G(\omega), K(\omega)$ and $H(\omega)$ have been given in Section 3.1.2. Expanding eqn.3.1-23, the filter coefficient function $L(n)$ is obtained and is shown in Table 3.2. Note that we also include the third bandpass direction in the texture analysis model. It can simply be obtained by convolving the bandpass filters $G(\omega_x)$ and $G(\omega_y)$, i.e.,

$$\psi^3(2\omega_x,2\omega_y)=G(\omega_x)G(\omega_y)\phi(\omega_x)\phi(\omega_y)$$

The forward and the backward of LOG-based wavelet transforms had shown in Figure 2.7 and Figure 2.8. It has the same structure as Canny-based over-complete wavelet. Figure 3.2 shows an example of wavelet transform using the proposed wavelet kernel, and Figure 3.3 shows the reconstructed image. It can be seen that the proposed wavelet kernel can achieve a perfect reconstruction.

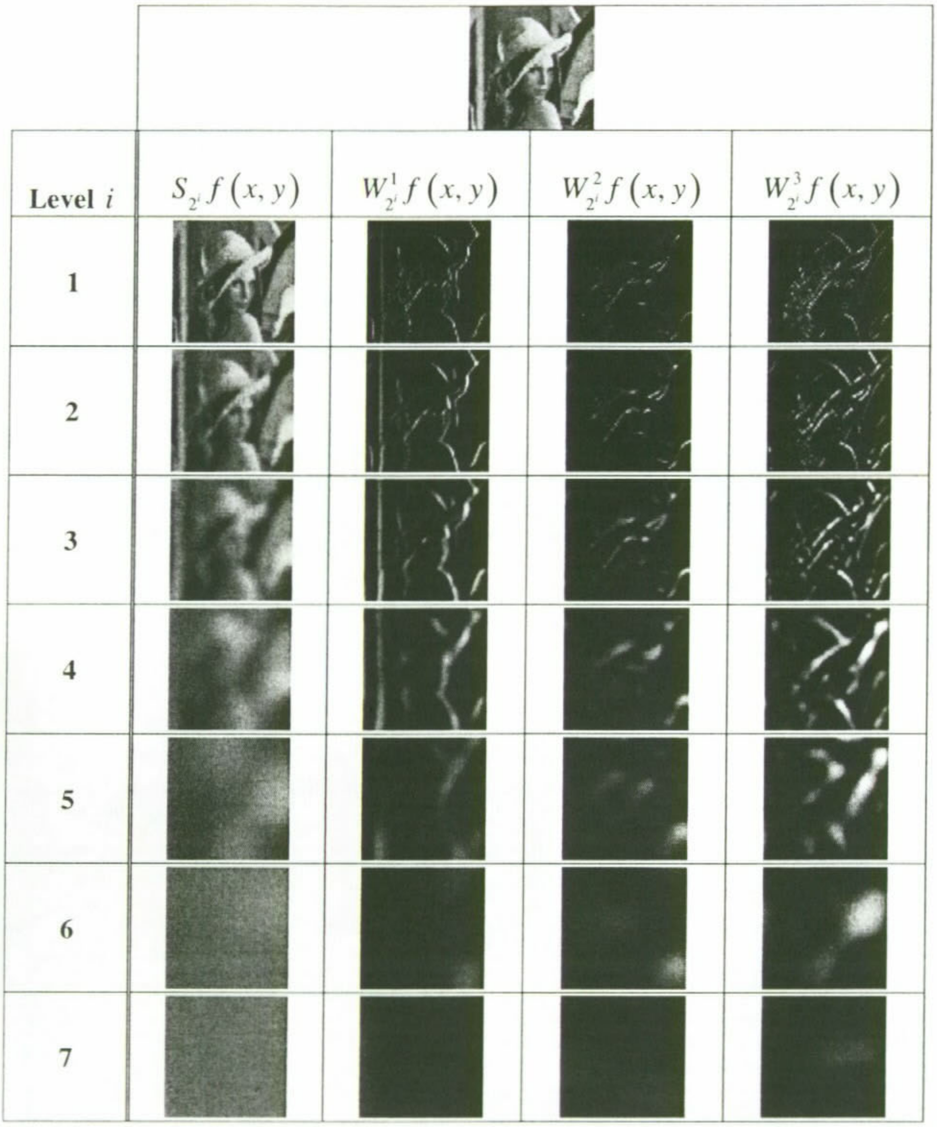
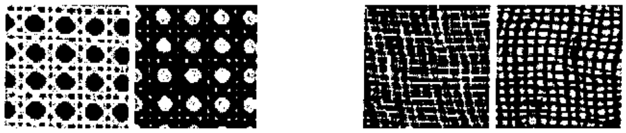


Figure 3.2, Example of the dyadic wavelet transform – Top: The original image. From left to right, 1st column: lowpass response, 2nd column: vertical bandpass response, 3rd column: horizontal bandpass response, 4thcolumn: the bandpass response of both horizontal and vertical filters.

Figure 3.3, Example of image reconstruction: (**Left**) Original image (**Right**) Reconstructed image

3.2. New feature representation model

Many wavelet feature representations use simple statistical method such as mean[64] or variance[27] for texture feature representation. Different statistical values are efficient to represent specific features, such as the mean values can represent the intensity and the variance can represent the variation of image pattern. However, consider two texture images with the same pattern but opposite intensity, e.g., the D101 and D102 or D103 and D104 in Brodatz Album[13] as shown in Figure 3.4. As feature representation used in [27], the similarity measure based on the variance only may not distinguish between intensities of patterns.



D101 and **D102** (Cane) **D103** and **D104** (Loose burlap)

Figure 3.4, The examples of Brodatz Album that contain similar texture patterns but with inverse intensities.

In [70], it was found that two texture patterns of randomly oriented lines are perceived as similar, if the mean and the standard derivation of the line orientations are the same, regardless of the actual distribution. Manjunath and Ma[24] proposed a texture feature representation method which calculates both mean and standard deviation from the Gabor wavelet results. The method can significantly increase the retrieval rate from the Brodatz database test[28]. We extent their feature representation method and apply in separable wavelet schemes. We found that the new representation is more suitable for representing wavelet features. After one level wavelet transform, we can obtain a lowpass signal $S_{2^j} f(x,y)$ and three bandpass signals $W_{2^j}^b f(x,y)$ for $b=1,2,3$ as explained in Sections 2.3.1 and 3.1.3. For the lowpass signals, we have still to calculate the mean and the standard deviation. However, for all bandpass signals, we calculate the mean μ and the standard deviation σ at positive and negative sides separately. Mathematically, at $s = 2^i$ level, the mean and the standard deviation are calculated respectively as,

$$\mu\left(W_s^b f(x,y)\right)\begin{cases} \mu^+\left(W_s^b f(x,y)\right) = \dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1} W_s^b f(n,m)}{L_1} \\ \mu^-\left(W_s^b f(x,y)\right) = \dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1} W_s^b f(n,m)}{L_2} \end{cases} \quad where \quad W_s^b f(n,m) \begin{cases} >0 \\ <0 \end{cases} \tag{3.2-1}$$

$$\sigma\left(W_s^b f(x,y)\right)\begin{cases} \sigma^+\left(W_s^b f(x,y)\right) = \sqrt{\dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1}\left[\mu^+\left(W_s^b f\right) - \left(W_s^b f(n,m)\right)\right]^2}{L_1}} \\ \sigma^-\left(W_s^b f(x,y)\right) = \sqrt{\dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1}\left[\mu^-\left(W_s^b f\right) - \left(W_s^b f(n,m)\right)\right]^2}{L_2}} \end{cases} \quad where \quad W_s^b f(n,m) \begin{cases} >0 \\ <0 \end{cases} \tag{3.2-2}$$

where L_1 and L_2 are the number of samples greater than and less than zero in the bandpass signal respectively. By using this new representation, the "inverse" matching problems can be solved. In fact, this new representation can be used for other schemes such as the sub-sampling wavelets. As shown in the experimental results, the new representation can significantly increase the retrieval rate, for both the over-complete and the sub-sampling wavelet features as compared to the original representation.

For each level of decomposition, we obtain three bandpass and one lowpass filter responses. For each bandpass, we calculate the mean and the standard deviation values using eqn.3.2-1 and eqn.3.2-2 respectively. As for the lowpass filter, we simply take the mean and standard deviation values because the value is always positive and approaches the mean value of the image. Therefore, for each level of decomposition, we obtain 7 pairs of mean and standard deviation values. For four decomposition levels, the feature vector contains 28 pairs of mean and standard deviation which gives a total of 56 features, i.e.,

$$f = \left(u_{00}\sigma_{00}u_{01}^{+}\sigma_{01}^{+}u_{01}^{-}\sigma_{01}^{-}u_{02}^{+}\sigma_{02}^{+}u_{02}^{-}\sigma_{02}^{-}u_{03}^{+}\sigma_{03}^{+}u_{03}^{-}\sigma_{03}^{-}u_{10}\sigma_{10}\ldots\ldots u_{33}^{+}\sigma_{33}^{+}u_{33}^{-}\sigma_{33}^{-}\right) \qquad (3.2\text{-}3)$$

It is possible to further reduce the number of features in eqn.3.2-3 without adversely affecting the retrieval accuracy. Section 3.3.6 will describe a way to do this and give some experimental results. It is shown that features can be reduced to 41 with a slight decrease in retrieval accuracy.

3.3. Retrieval Performance

In this section, we present retrieval results on the entire Brodatz database for both the original and Gaussian noise corrupted images. A comparative study of the proposed over-complete wavelets with other texture retrieval methods is also carried out. These methods include the Gabor wavelet feature[24], the Canny-based over-complete wavelets[9] and various sub-sampling wavelets[51, 71-73].

3.3.1. Brodatz texture database and retrieval performance measure

To test the accuracy of various texture retrieval approaches, the full Brodatz texture database is examined. The Brodatz texture database is obtained from the entire Brodatz Album. In our experiment, we obtain 111 texture images collection[2] from the Brodatz Album. Figure 3.5 shows the entire Brodatz Album images. Each album image is an 8 bit gray level image and the size is 512x512. We extract nine 128x128 non-overlapping sub-images at the center of each texture image. Therefore, the database contains a total of 999 sub-images. Every sub-image extracted from the same texture image is considered to be from one texture class.

A request query image is obtained from one of the sub-images in the database. A query is then performed on the whole database and the results are ranked in the order

[2] The entire Brodatz Album contains 112 texture images. However, the texture image D14 is missing in our Brodatz Album collection.

depending upon the distance measurement of feature vectors as discussed in Section 1.2.3. The accuracy is calculated by averaging the overall retrieval rate of the same class images within a retrieval size. Excluding the query sub-image, a 100% collection is achieved if the other 8 sub-images from the same texture class are retrieved at the top 8 positions. More details on this retrieval analysis technique can be found in **[28]**.

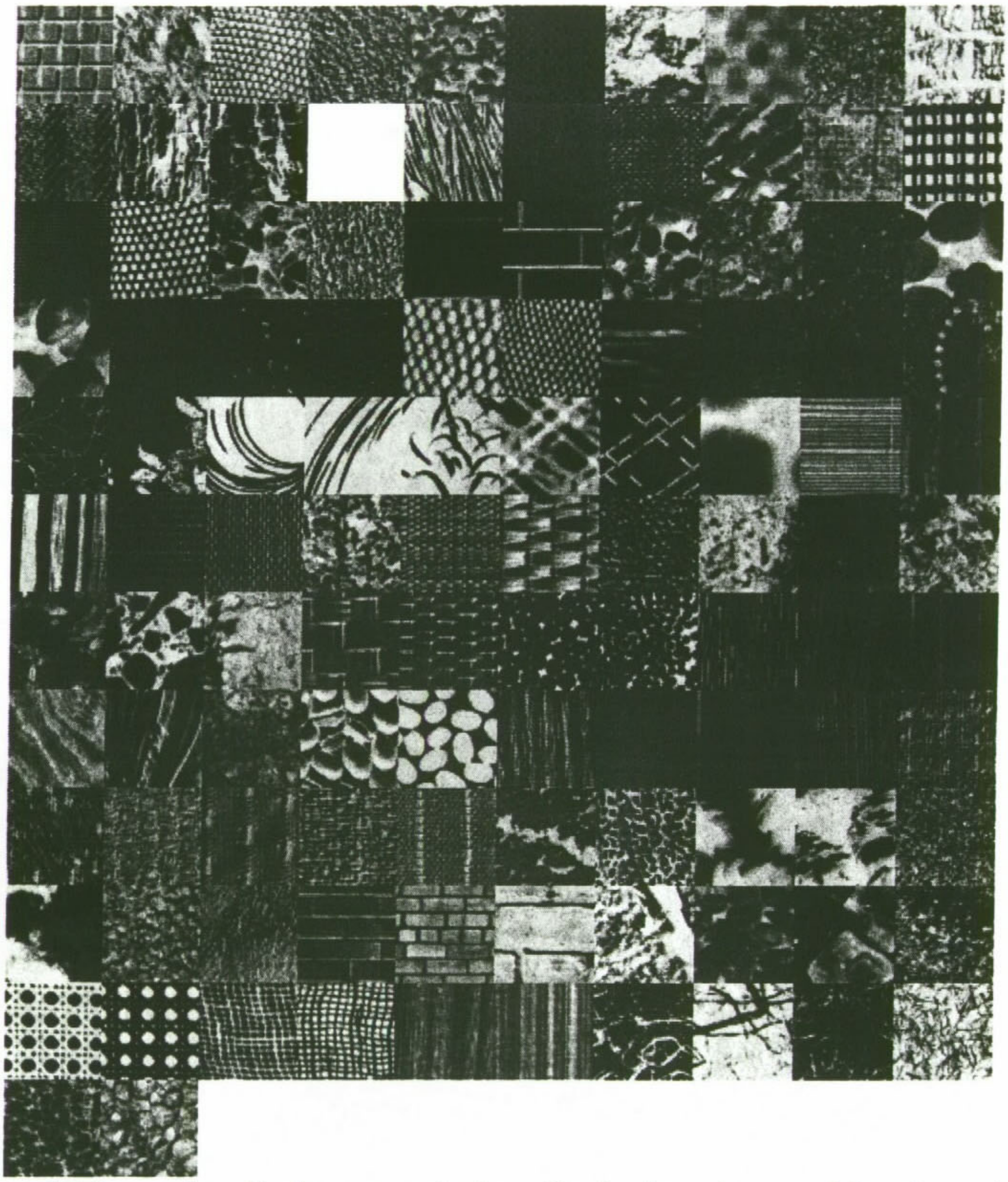


Figure 3.5, Images form Brodatz texture database. Starting from the upper left and scanning in raster order gives D1 to D112 images. Note that D14 is missing in our database and occupies blank position.

3.3.2. Retrieval rate comparison for different feature representation

Section 3.2 discussed the proposed feature representation while eqn.3.2-1 and eqn.3.2-2 show the methods to calculate the mean and standard deviation of the texture features. In this section, the experimental results of the retrieval rate using our proposed method and the variance only approach are shown. The tested features include the over-complete wavelets with both the Laplacian of Gaussian-based and Canny-based kernels, and several sub-sampling wavelets with different wavelet kernels. Similar to the feature representation in [27], we calculate the standard deviation in the bandpass and the lowpass outputs of both the over-complete and the sub-sampling wavelet transforms. The feature length is 13 for four level decompositions. Table 3.3 shows the retrieval rate in the Brodatz database texture as discussed in Section 3.3.1. Figure 3.6 shows the retrieval results of the two over-complete wavelets and one sub-sampling wavelets.

The experimental results show that the proposed feature representation can increase the retrieval rate, by around 11-14%, compared to the case of using variance only for both the over-complete and the sub-sampling wavelets scheme.

Texture Feature	(O) Laplacian of Guassian	(O) Canny	(S) Villa1810	(S) Antonini	(S) Odegard	(S) Daub8	(S) Haar
Retrieval rate using New feature representation(%)	79.65	76.93	76.46	76.26	76.38	74.74	74.82
Retrieval rate using Variance only (%)	65.98	62.07	66.19	65.04	65.03	63.56	61.34

(O) – Over-complete wavelets, (S) – sub-sampling wavelets,

Table 3.3, Comparison of Retrieval performance of different features using new feature representation and variance only.

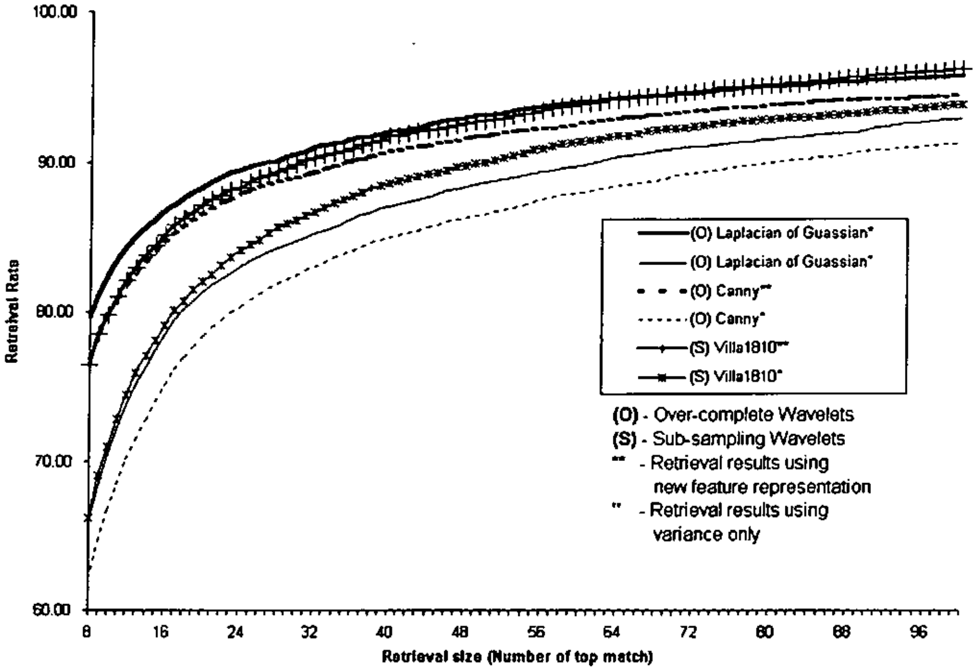


Figure 3.6, Retrieval performance of different texture features, using the new feature representation and variance, on Brodatz texture database. Note that for clarity, we show two over-complete wavelets and one sub-sampling wavelet transform only.

3.3.3. Comparative study of different texture features

In this section, a comparative study of the three texture retrieval methods is presented. Except for the retrieval method using the Gabor wavelet feature that used 48 dimensions feature representation[24], all the other retrieval methods follow the new approach as discussed in Section 3.2. In our analysis, four decomposition levels in the wavelet transform are found to perform the best on the entire Brodatz texture database test. Table 3.4 shows the average retrieval rate of each Brodatz Album. Examples of the retrieval results are shown in Section 3.3.5. Due to space limitations, we present only 4 retrieval results in our study, they are the over-complete wavelet using the proposed Laplacian of Gaussian-based structure and the Canny-based structure, the Gabor wavelet features and the sub-sampling wavelets using Villa1810. The overall retrieval performance was obtained by averaging the retrieval rate of each album. Figure 3.7 and Table 3.5 show the overall retrieval results of the Brodatz texture database test. The timing information including the average analysis times and the retrieval times of the three retrieval methods are shown in Table 3.6.

Average retrieval rate in percentage of each Brodatz Album (%)

(O) – Over-complete scheme, (S) – Sub-sampling scheme, (G) – Gabor wavelet scheme

	(O) Laplacian of Gaussian	(O) Canny	(S) Villa 1810	(G) Gabor feature		(O) Laplacian of Gaussian	(O) Canny	(S) Villa 1810	(G) Gabor feature		(O) Laplacian of Gaussian	(O) Canny	(S) Villa 1810	(G) Gabor feature
D1	100.00	100.00	97.22	100.00	**D40**	55.56	50.00	58.33	41.67	**D78**	66.67	59.72	66.67	72.22
D2	80.56	75.00	73.61	98.61	**D41**	87.50	75.00	86.11	76.39	**D79**	98.61	95.83	100.00	100.00
D3	100.00	100.00	100.00	94.44	**D42**	52.78	41.67	45.83	33.33	**D80**	86.11	83.33	77.78	98.61
D4	80.56	72.22	80.56	98.61	**D43**	12.50	9.72	11.11	2.78	**D81**	87.50	86.11	66.67	100.00
D5	76.39	70.83	63.89	73.61	**D44**	30.56	30.56	34.72	11.11	**D82**	100.00	100.00	100.00	100.00
D6	100.00	100.00	100.00	100.00	**D45**	40.28	37.50	36.11	9.72	**D83**	100.00	100.00	100.00	100.00
D7	48.61	48.61	45.83	52.78	**D46**	100.00	100.00	100.00	100.00	**D84**	100.00	100.00	100.00	100.00
D8	100.00	100.00	86.11	100.00	**D47**	100.00	100.00	100.00	100.00	**D85**	98.61	98.61	90.28	100.00
D9	93.06	87.50	97.22	95.83	**D48**	98.61	98.61	100.00	88.89	**D86**	77.78	66.67	69.44	80.56
D10	98.61	98.61	95.83	100.00	**D49**	100.00	100.00	100.00	100.00	**D87**	100.00	100.00	98.61	98.61
D11	100.00	100.00	100.00	100.00	**D50**	72.22	68.06	77.78	79.17	**D88**	68.06	76.39	58.33	20.83
D12	79.17	79.17	81.94	100.00	**D51**	94.44	94.44	90.28	93.06	**D89**	36.11	30.56	38.89	19.44
D13	58.33	50.00	43.06	70.83	**D52**	72.22	73.61	69.44	90.28	**D90**	45.83	43.06	34.72	62.50
D15	80.56	73.61	84.72	77.78	**D53**	100.00	100.00	100.00	100.00	**D91**	19.44	16.67	18.06	25.00
D16	100.00	100.00	100.00	100.00	**D54**	63.89	66.67	58.33	63.89	**D92**	100.00	100.00	100.00	94.44
D17	100.00	100.00	100.00	100.00	**D55**	100.00	100.00	100.00	100.00	**D93**	93.06	81.94	88.89	98.61
D18	84.72	75.00	88.89	98.61	**D56**	100.00	100.00	95.83	100.00	**D94**	98.61	94.44	97.22	100.00
D19	100.00	100.00	97.22	100.00	**D57**	100.00	100.00	100.00	100.00	**D95**	100.00	100.00	100.00	100.00
D20	100.00	100.00	100.00	100.00	**D58**	12.50	13.89	11.11	16.67	**D96**	100.00	95.83	90.28	93.06
D21	100.00	100.00	100.00	100.00	**D59**	27.78	23.61	37.50	26.39	**D97**	45.83	36.11	47.22	34.72
D22	70.83	59.72	69.44	75.00	**D60**	47.22	36.11	45.83	54.17	**D98**	77.78	75.00	66.67	55.56
D23	50.00	48.61	41.67	63.89	**D61**	56.94	58.33	47.22	38.89	**D99**	34.72	36.11	50.00	29.17
D24	95.83	94.44	91.67	94.44	**D62**	58.33	58.33	55.56	52.78	**D100**	62.50	61.11	29.17	47.22
D25	100.00	100.00	100.00	100.00	**D63**	59.72	56.94	55.56	34.72	**D101**	100.00	100.00	87.50	56.94
D26	100.00	95.83	58.33	100.00	**D64**	95.83	94.44	97.22	88.89	**D102**	100.00	100.00	94.44	54.17
D27	55.56	45.83	45.83	54.17	**D65**	100.00	100.00	100.00	100.00	**D103**	100.00	70.83	87.50	43.06
D28	100.00	97.22	94.44	98.61	**D66**	95.83	83.33	100.00	86.11	**D104**	100.00	80.56	87.50	52.78
D29	100.00	100.00	94.44	98.61	**D67**	70.83	58.33	56.94	86.11	**D105**	65.28	55.56	81.94	48.61
D30	47.22	43.06	36.11	33.33	**D68**	100.00	100.00	100.00	100.00	**D106**	87.50	83.33	81.94	75.00
D31	36.11	50.00	31.94	25.00	**D69**	41.67	37.50	41.67	26.39	**D107**	77.78	73.61	63.89	62.50
D32	100.00	100.00	100.00	100.00	**D70**	50.00	50.00	50.00	47.22	**D108**	54.17	48.61	55.56	40.28
D33	97.22	95.83	94.44	98.61	**D71**	91.67	84.72	91.67	83.33	**D109**	86.11	77.78	83.33	72.22
D34	100.00	98.61	100.00	100.00	**D72**	50.00	52.78	50.00	34.72	**D110**	100.00	100.00	100.00	84.72
D35	100.00	97.22	90.28	90.28	**D73**	38.89	41.67	40.28	45.83	**D111**	93.06	91.67	87.50	88.89
D36	80.56	79.17	84.72	86.11	**D74**	100.00	100.00	90.28	88.89	**D112**	54.17	54.17	54.17	69.44
D37	100.00	98.61	100.00	100.00	**D75**	100.00	100.00	100.00	97.22	**Avg.**	**79.65**	**76.93**	**76.46**	**75.70**
D38	80.56	80.56	69.44	95.83	**D76**	100.00	100.00	100.00	100.00					
D39	54.17	54.17	59.72	50.00	**D77**	100.00	100.00	100.00	100.00					

Table 3.4, Average of retrieval rate of each Brodatz Album. For clarity, only 4 retrieval results are shown.

Retrieval Method	Over-complete scheme		Gabor feature wavelet	Sub-sampling scheme				
Wavelet Kernel	Laplacian of Gaussian	Canny	Gabor	Villa1810	Antonini	Odegard	Daub8	Haar
Retrieval size = 8	79.65	76.93	75.70	76.46	76.26	76.38	74.74	74.82

Table 3.5, Performance of different texture features at a retrieval size of 8.

	Over-complete wavelets scheme	Gabor feature wavelets scheme	Sub-sampling wavelet scheme
Feature extraction time (sec)	~1	~7.5	~0.2-0.3
Feature vector length	56	48	56
Query time (sec)	~1.1	~1	~1.1

Table 3.6, The timing information for different texture retrieval methods on an Intel Pentium II-300 MHz PC using a C-language implementation.

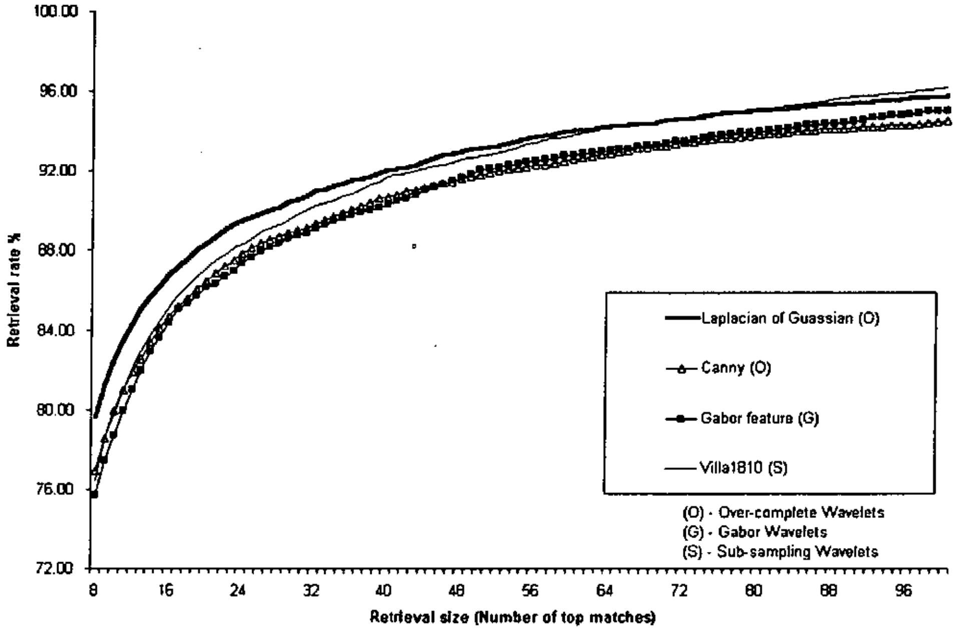


Figure 3.7, Retrieval performance of different texture features on Brodatz texture database. Note for clarity, only 4 retrieval methods are shown.

As it can be seen in Table 3.5, all retrieval methods can achieve above 70% retrieval rate. In particular, our proposed Laplacian of Gaussian based over-complete wavelets achieve above 79% from this test which is the highest retrieval rate among the tests.

Compared to sub-sampling and the Gabor Wavelet features, it has around 3-5% higher retrieval rate.

Comparing the timing information in Table 3.6, the sub-sampling methods have the shortest analysing times due to its sub-sampling nature. The two over-complete wavelets features take about one second. The Gabor wavelet feature takes the longest, around 7.5 sec. However, all three methods have nearly the same query time due to the short length of the feature vector.

3.3.4. Retrieve performance under different noise levels

Real world signals usually contain noise signals from the ideal signal. These are not part of the ideal signal. A noise signal may be caused by a wide range of sources, such as variations in the environmental variations or quantization errors. Independent noise can be described by an additive noise model, where the recorded image $x'(z_1, z_2)$ is the sum of the true image $x(z_1, z_2)$ and the noise $n(z_1, z_2)$

$$x'(z_1, z_2) = x(z_1, z_2) + n(z_1, z_2)$$

The noise is often zero-mean and described by its variance σ_n^2. The impact of the noise on the image is often described by the signal to noise ratio (SNR), which is given by

$$SNR = \frac{\sigma_x}{\sigma_n}$$

where σ_x is the variances of the true image. In many cases, additive noise is evenly distributed over the frequency domain (i.e. white noise), e.g. Gaussian noise is one of the white noise signal which distributes over the frequency domain, whereas an image contains mostly low frequency information.

Translation invariant is an important consideration in multi-resolution pattern analysis. It could seriously affect the texture retrieval results since any mistake in information characterization will directly affect the accuracy of the retrieval results. Theoretically, the over-complete wavelets should perform better than the sub-sampling wavelet scheme under different noise levels due to its translation invariant properly. However, it is necessary to assess and confirm this in an experimental setting. In this section, we present a comparative study of the three retrieval methods with white noise corrupted data. In the experiments, different levels of Gaussian noise are added to the Brodatz Album with the peak-signal-to-noise ratio that ranges from OdB to 25dB. We assume that the peak value of each texture image in database is 255, and we take every 5dB step for examination.

Figure 3.8 shows a plot of the percentage drop of the retrieval rate against signal-to-noise ratios at the first eight queried images(retrieval size of 8), and Table 3.7 shows the overall retrieval rate at the retrieval size of 8.

From the experimental results, the retrieval rates of both Laplacian of Gaussian-based and Canny-based over-complete schemes are least affected as the signal-to-noise-ratio decreases. The sub-sampling wavelet scheme using various kernels has a large percentage drop in retrieval rate as the signal-to-noise ratio decreases. The Gabor wavelet scheme remains to have a high retrieval rate at low noise level but the retrieval accuracy decreases dramatically as the noise level increases. In summary, the over-complete wavelet scheme has a robust performance even under severe noise condition.

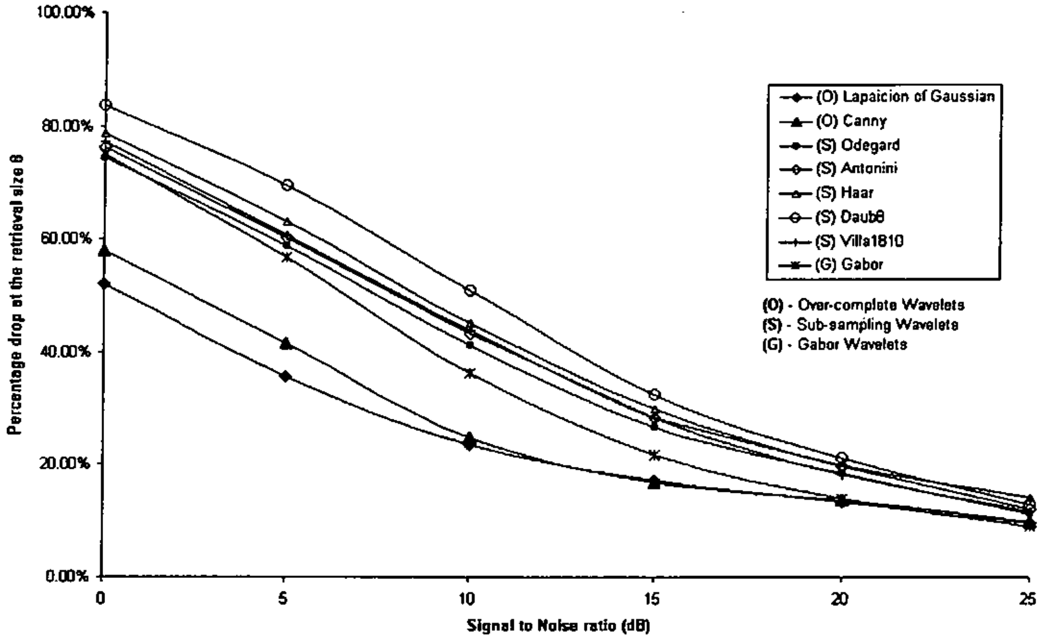


Figure 3.8, Percentages drop of the retrieval rate versus the signal-to-noise ratio at the retrieval size of 8.

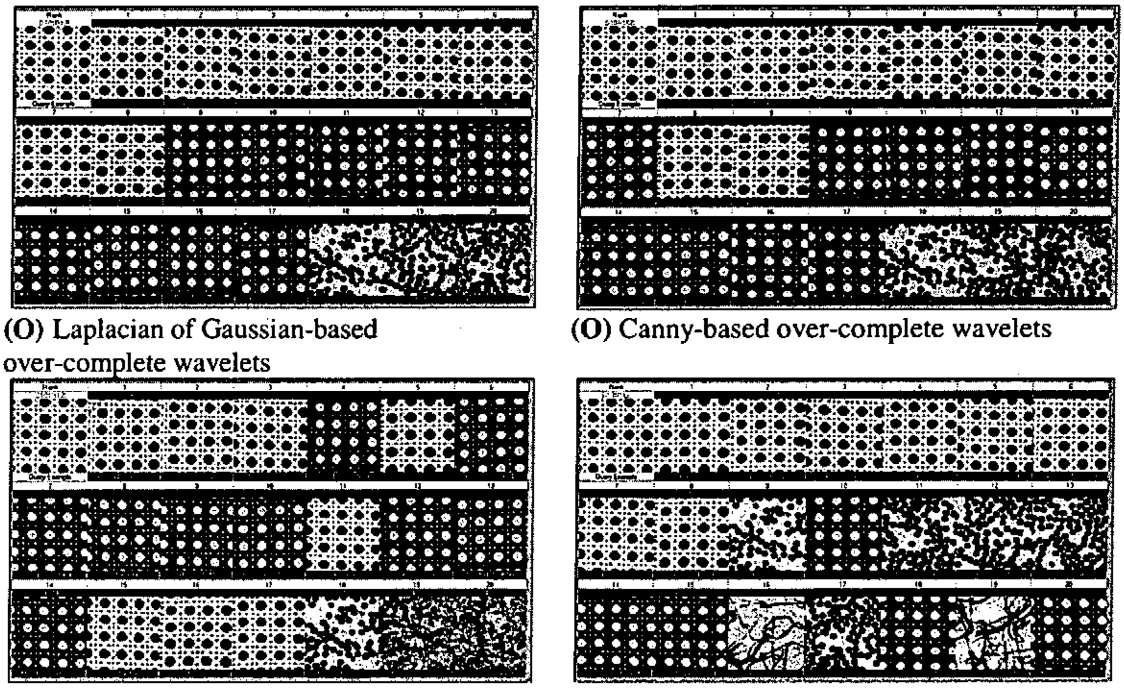
The average retrieval rates with different Signal-to-Noise Ratio													
Texture feature	Noise free	0 (dB)		5 (dB)		10 (dB)		15 (dB)		20 (dB)		25 (dB)	
	Retrieval rate %	Retrieval rate %	% drop	Retrieval rate %	% drop	Retrieval rate %	% drop	Retrieval rate %	% drop	Retrieval rate %	% drop	Retrieval rate %	% drop
(O) Laplacian of Gaussian	79.65	38.26	51.96%	51.31	35.58%	60.94	23.50%	65.90	17.26%	69.06	13.31%	72.02	9.58%
(O) Canny	76.93	32.38	57.91%	44.94	41.57%	57.78	24.89%	64.03	16.77%	66.42	13.66%	69.41	9.78%
(S) Odegard	76.38	19.61	74.33%	31.52	58.73%	44.88	41.24%	56.04	26.62%	62.35	18.37%	67.81	11.22%
(S) Antonini	76.26	18.09	76.28%	30.27	60.31%	43.33	43.18%	54.69	28.29%	61.22	19.72%	67.02	12.12%
(S) Haar	74.82	16.02	78.60%	27.63	63.08%	41.17	44.98%	52.38	30.00%	59.92	19.92%	64.25	14.13%
(S) Daub8	74.74	12.22	83.64%	22.87	69.40%	36.82	50.73%	50.59	32.31%	59.03	21.01%	65.08	12.92%
(S) Villa1810	76.46	17.37	77.29%	30.04	60.71%	42.91	43.89%	54.84	28.28%	62.58	18.16%	67.57	11.63%
(G) Gabor	75.70	18.97	74.94%	32.80	56.68%	48.26	36.25%	59.30	21.67%	65.14	13.95%	68.81	9.11%

(O) – Over-complete wavelets, **(S)** – Sub-sampling wavelets, **(G)** – Gabor wavelets

Table 3.7, Retrieval results at the retrieval size of 8 for different noise levels

3.3.5. Retrieval examples

Figures 3-9, 3-10 and 3-11 show the top 20 retrieval images of different retrieval schemes using different query image. These schemes include the two over-complete wavelet schemes (the proposed Laplacian of Gaussian-based kernel and the Canny-based kernel), the Multi-resolution Gabor wavelet and the sub-sampling wavelet using Villa 1810 kernel.



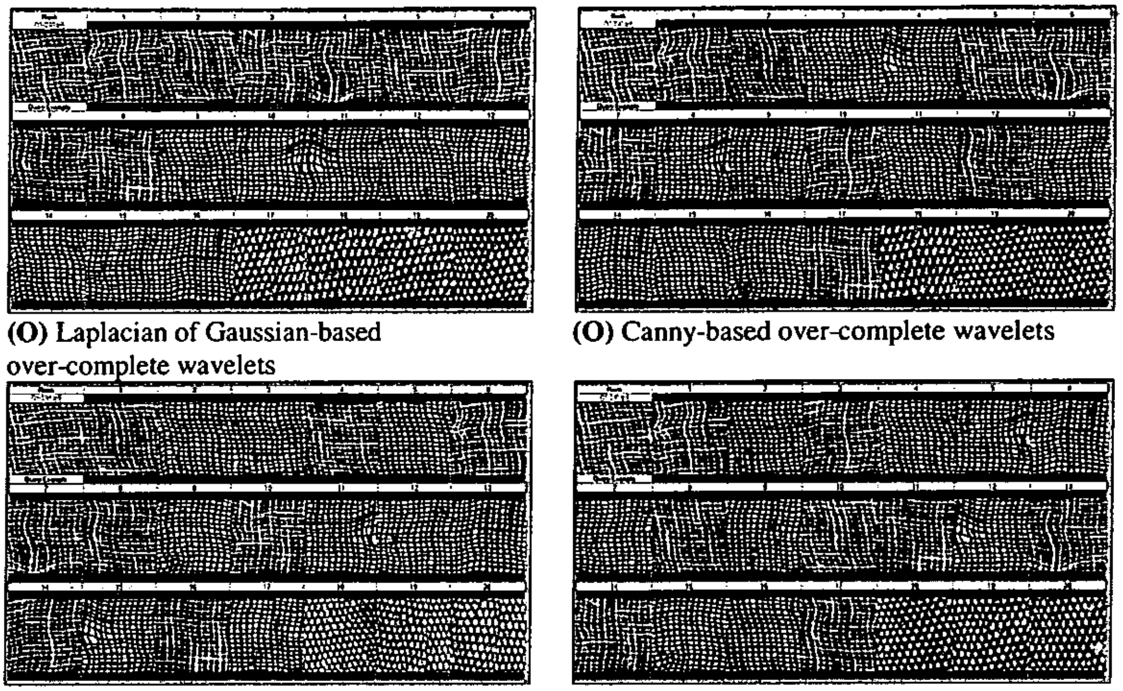
(O) Laplacian of Gaussian-based over-complete wavelets

(O) Canny-based over-complete wavelets

(G) Gabor Feature

(S) Sub-sampling wavelet using Villa 1810

Figure 3.9, Image retrieval results of the Cane (D102 in Album texture). In each result, the top left is the query example, and the images are raster-scan ordered by their similarities to the query image



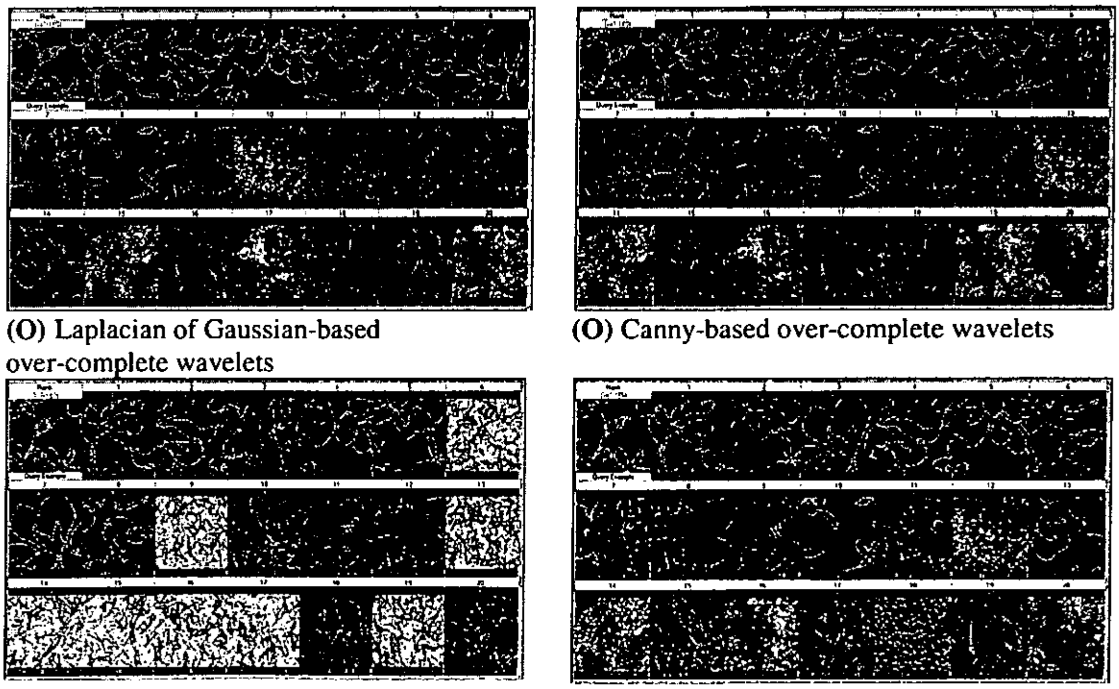
(O) Laplacian of Gaussian-based over-complete wavelets

(O) Canny-based over-complete wavelets

(G) Gabor Feature

(S) Sub-sampling wavelet using Villa 1810

Figure 3.10, Image retrieval results of the Loose burlap (D104 in Album texture). In each result the top left is the query example, and the images are raster-scan ordered by their similarities to the query image.



(O) Laplacian of Gaussian-based over-complete wavelets

(O) Canny-based over-complete wavelets

(G) Gabor Feature

(S) Sub-sampling wavelet using Villa 1810

Figure 3.11 Image retrieval result of the Lace (D41 in Album texture). In each result the top left is the query example, and the images are raster-scan ordered by their similarities to the query image.

3.3.6. Feature vector reduction

It can easily be seen that some features in eqn.3.2-3 are redundant and could be eliminated in order to speed up the query time. Since the mean value of the bandpass response is always zero, the magnitude of the mean value at both positive and negative sides should be the same. We can simply combine these two features by taking their averages, i.e.

$$u^{'}_{mn} = \frac{|u^{+}_{mn}| + |u^{-}_{mn}|}{2} \quad \textit{where} \;\; m = 0,1,2,3,4.... \;\; \textit{and} \;\; n = 1,2,3$$

where m is the level of decomposition and n is the corresponding bandpass response. On the other hand, the mean values of the lowpass responses at different decomposition levels are approximately equal to the mean of the texture pattern. Therefore, we can ignore all the mean values from the low pass response. As a result, the feature length of a four level wavelet transform is decreased from 56 to 41. The modified feature representation can be written as:

$$f = \left(u_{00}\sigma_{00}u^{'}_{01}\sigma^{+}_{01}\sigma^{-}_{01}u^{'}_{02}\sigma^{+}_{02}\sigma^{-}_{02}u^{'}_{03}\sigma^{+}_{03}\sigma^{-}_{03}\sigma_{10}.......u^{'}_{32}\sigma^{+}_{32}\sigma^{-}_{32}u^{'}_{33}\sigma^{+}_{33}\sigma^{-}_{33}\right) \quad (3.3\text{-}1)$$

The length of the feature vector is decreased by 25%. The comparison of retrieval results using the original and reduced features are shown in Figure 3.12 The overall retrieval rate using this new representation becomes 79.4%, a decrease of about 0.25% from the original proposed feature representation as shown in Table 3.5. However, the retrieval rate is still higher than other texture features as can be seen in Section 3.3.3. Therefore, this feature reduction does not seriously affect retrieval accuracy while decreasing the query time.

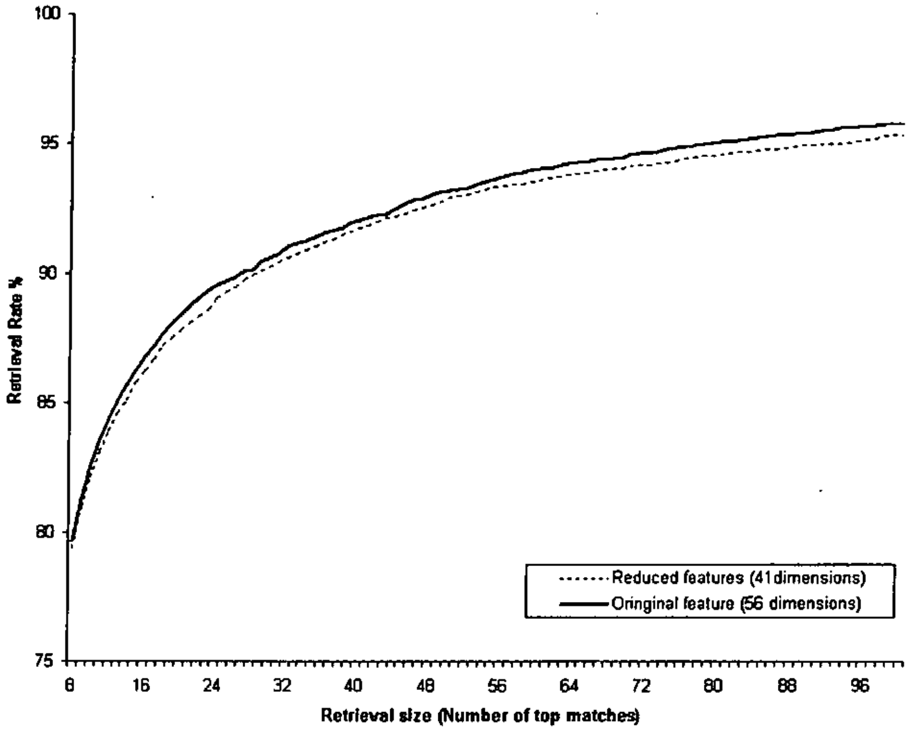


Figure 3.12 Comparison of retrieval rate between the original and reduced feature representation

3.3.7. Summary of retrieval results

Some important points from the retrieval results are summarized as follows:

- The over-complete wavelets using either the Laplacian of Gaussian-based or the Canny-based kernels can achieve a higher retrieval rate than other schemes.

- Comparing the Canny-based wavelet kernel in [9] and the proposed Laplacian of Gaussian-based wavelet kernel in the over-complete scheme, our proposed wavelet kernel has a higher retrieval rate in the test. It proves that the proposed wavelet kernel is more suitable for texture analysis due to its line thickness detection capability as discussed in Section 3.1.1.

- The proposed feature representation, i.e. calculating the mean and standard deviation separately on the positive and the negative filter responses, can be applied to both the over-complete and the sub-sampling wavelet schemes. The retrieval accuracy is significantly increased as compared to the representation that uses a single variance only.

- Analysis time is greatly reduced by using the over-complete wavelets as compared to the Gabor wavelet feature. It indicates that the new texture module is suitable for real-time texture feature extraction.
- From the retrieval experiment with white noise corrupted data, the over-complete wavelet scheme is the least affected as compared to other schemes. It proves that this scheme has a robust performance even under severe noise levels.

3.4. Summary

This chapter introduces a new texture feature extraction scheme using the over-complete wavelets that have the advantages of low analysing time and translation invariant property. However, instead of using the first deviative-based wavelet kernel proposed by Mallat and Zhong[9] discussed in Chapter 2, we have derived a new kernel that is based on the Laplacian of Gaussian kernel and proved that it is more suitable for texture feature characterization. We also present a new feature representation method that calculates the mean and the standard deviation separately on the positive and negative sides. It is found that the new feature representation can significantly improve the accuracy of texture retrieval for both the over-complete wavelets and the sub-sampling wavelets. From the experimental results using the Brodatz texture database, our proposed over-complete wavelet scheme can achieve a low analysing time, the highest retrieval accuracy and robust performance even under severe noise levels. It indicates that this scheme is suitable for real-time texture analysis for content-based image/video retrieval applications.

CHAPTER 4

4. A Fast algorithm for 1D Laplacian of Gaussian-based over-complete wavelet

As we mentioned in Chapter 1, wavelet transform is a powerful technique that has wide application in signal processing. The technique is especially useful for signal compression, image enhancement, content-based feature analysis, etc, which involve very heavy computation. Fast algorithms for wavelet transforms become essential to reducing the processing time. Most of the fast algorithms are designed for specific processors[74-77]. However, due to the cost of hardware, it is not economical for some applications. Softwarè approach using fast algorithms for wavelet transform is a good alternative. However, not much research on fast algorithms using the software approach is available. Recent studies of fast algorithms, such as lifting scheme for sub-sampling wavelets[52] and spatial implementation for Canny-based over-complete wavelets[53, 54], can greatly reduce the computation time for wavelet transforms.

The Laplacian of Gaussian(LOG)-based wavelet kernel has shown superior performance in texture feature analysis as discussed in Chapter 3. The kernel gives a complete wavelet transform and thus it allows perfect reconstruction of the signal. However, implementation using conventional filtering is inefficient and causes artifacts. By deriving a general expression for the computational cost using the conventional filtering implementation, it reveals that the inverse transform is significantly more costly in computation than the forward transform. In order to reduce the computational complexity, a spatial implementation is proposed. Both theoretical and experimental studies show that the new approach can greatly decrease the cost of transforms. Furthermore, the boundary artifact effect can be eliminated during multi-level wavelet transforms using the new approach. The description can also be more accurate for different image processing applications, such as the description of texture contents for similarity retrieval.

In this chapter, we propose the fast algorithm for LOG-based over-complete wavelet transforms. We start from the 1D case for the wavelet transforms in Section 4.1. We first discuss the computational complexity of the transform using the conventional filtering approach. And then we propose the spatial implementation method to reduce the computational complexity. Later, we extend the fast algorithm of the 1D wavelet transform to the 2D wavelet transforms in Section 4.2. In order to simplify the discussion, we use z-transform to explain the mathematical formulation in the rest of chapter.

4.1. 1D dyadic LOG-based over-complete wavelet representation

4.1.1. Conventional filtering approach for LOG-based over-complete wavelet representation 1D transforms

Assume the lowpass $X(z)$ and bandpass $D(z)$ filter responses are obtained through filtering as follows,

$$X_{j+1}(z)=H(z_j)X_j(z) \tag{4.1-1}$$

$$D_{j+1}(z)=G(z_j)X_j(z) \tag{4.1-2}$$

where $z_j = z^{2^{j-1}}$. $H(z)$ and $G(z)$ are the lowpass and bandpass FIR filters respectively, while index j indicates the level of the wavelet transform. The LOG-based wavelet kernel with an arbitrary order n can be written as

$$H(z_j) = \frac{z_j^n}{2^{2n}}\left(1+z_j^{-1}\right)^{2n} \tag{4.1-3}$$

$$G(z_j) = 4\left(z_j^{-1} - 2 + z_j\right) \tag{4.1-4}$$

In reconstruction,

$$X_j(z) = K(z_j)D_{j+1}(z) + \overline{H}(z_j)X_{j+1}(z) \tag{4.1-5}$$

where $\overline{H}(z_j)$ is the time inverse of $H(z_j)$. The reconstruction filter $K(z_j)$ is written as,

$$K(z_j) = \frac{1-\left|H(z_j)\right|^2}{G(z_j)} \tag{4.1-6}$$

The structure of the one-dimensional LOG-based over-complete wavelet transforms are the same as the Canny-based over-complete wavelet transforms. Figure 2.6 and Figure 2.7 also give the LOG-based over-complete forward and inverse transforms, respectively. Table 4-1 shows the first level *(j=0)* filter coefficients of *G, H* and *K,* for *n=1,2 and 3* in eqn.4.1-4, 4.1-5 and 4.1-6. To analyse the computational complexity of the forward and the inverse wavelet transforms, the numbers of additions and multiplications are calculated. Using the Binomial theorem, $H(z_j)$ can easily be expanded as follows,

$$H(z_j) = \frac{1}{2^{2n}}\left(\sum_{k=0}^{n-1} {}^{2n}C_k\left[z_j^{n-k} + z_j^{-(n-k)}\right] + {}^{2n}C_n\right) \tag{4.1-7}$$

where C is the binomial expression. Substituting eqn.4.1-3 and eqn.4.1-4 to eqn.4.1-6, $K\left(z_j\right)$can be rewritten as,

$$K\left(z_j\right)=B_0+\sum_{m=1}^{2n-1}B_m\left[z_j^m+z_j^{-m}\right] \qquad (4.1\text{-}8)$$

where

$$B_m=\left(-\frac{1}{16}\right)\left(\sum_{k=m}^{2n-1}\frac{1}{2^{2k}}\,{}^{2k}C_{k-m}\right)$$

By analysing eqn.4.1-4, eqn.4.1-7 and eqn.4.1-8, the computational costs of both forward and inverse filters can be expressed as follows,

$$Cost\left(G\right)=\left(2\right)Cost_{add}+\left(2\right)Cost_{multiply}$$

$$Cost\left(H\right)=Cost\left(\overline{H}\right)=\left(2n\right)Cost_{add}+\left(n+1\right)Cost_{multiply}$$

$$Cost\left(K\right)=\left(4n-2\right)Cost_{add}+\left(2n\right)Cost_{multiply}$$

where $Cost_{add}$ and $Cost_{multiply}$ are the computational costs of an addition and a multiplication operations respectively.

Note 4.1: The computational complexities of the forward and the inverse wavelet transforms at level j, using the filtering approach, can be written as,

$$Cost_F\left(Forward\right)=\left(2n+2\right)Cost_{add}+\left(n+3\right)Cost_{multiply}$$

$$Cost_F\left(Inverse\right)=\left(6n-2\right)Cost_{add}+\left(3n+1\right)Cost_{multiply}$$

Proof: The forward wavelet transform consists of filtering the input signal with G and H. Thus the computational complexity of the forward wavelet transform is calculated by summing up $Cost(G)$ and $Cost(H)$. Similarly, for the inverse wavelet transform, it equals the sum of $Cost(\overline{H})$ and $Cost(K)$.□

Note that these complexities are independent of the decomposition level. As shown in Note 4.1, the inverse transform requires approximately three times the cost of the forward transform. By exploiting the redundancy between filters, an alternative implementation is proposed to reduce complexity.

4.2. Proposed spatial approach for 1D LOG-based wavelet transform

Similar to other over-complete schemes, the LOG-based kernel provides a redundant signal representation. We can exploit the correlation between the lowpass and the bandpass outputs to simplify some of the computations. From eqn.4.1-2,

$$z_j^{n-1}\cdot G(z_j)=4\left(z_j^{n-2}-2\cdot z_j^{n-1}+z_j^{n}\right) \tag{4.1-9}$$

$$z_j^{-(n-1)}\cdot G(z_j)=4\left(z_j^{-(n-2)}-2\cdot z_j^{-(n-1)}+z_j^{-n}\right) \tag{4.1-10}$$

Rearranging eqn.4.1-9, eqn.4.1-10 and using mathematical induction, it can be proved that for an arbitrary *m*,

$$z_j^{m}+z_j^{-m}=\frac{1}{4}\left\{\left(\sum_{k=1}^{m-1}(m-k)\left[z_j^{k}+z_j^{-k}\right]\right)\cdot G(z_j)+m\cdot G(z_j)+8\right\} \tag{4.1-11}$$

By substituting eqn.4.1-11 into eqn.4.1-7, and using the fact that,

$$2^{2n}=\sum_{k=0}^{n-1}2\left({}^{2n}C_k\right)+{}^{2n}C_n$$

we obtain the following equation,

$$X_{j+1}(z)=X_j(z)+D_j(z)\left\{E_n+\sum_{l=1}^{n-1}E_{n-l}\left(z_j^{-l}+z_j^{l}\right)\right\} \tag{4.1-12}$$

where

$$E_m=\frac{1}{2^{2n+2}}\sum_{k=1}^{m}k\left({}^{2n}C_{m-k}\right) \tag{4.1-13}$$

Define $F\left[D_j(z)\right]$ as the predictor at the j level wavelet transform, then,

$$F\left[D_j(z)\right]=D_j(z)\left\{E_n+\sum_{l=1}^{n-1}E_{n-l}\left(z_j^{-l}+z_j^{l}\right)\right\} \tag{4.1-14}$$

$$X_{j+1}(z)=X_j(z)+F\left[D_j(z)\right] \tag{4.1-15}$$

Eqn.4.1-15 shows that $X_{j+1}(z)$ can be obtained simply by adding the predictor to $X_j(z)$. This pure spatial relationship also provides an alternative way for calculating the inverse. We can replace the addition with subtraction to reconstruct the signal. Figure 4-2 shows the new structure for implementing wavelet transforms from eqn.4.1-15.

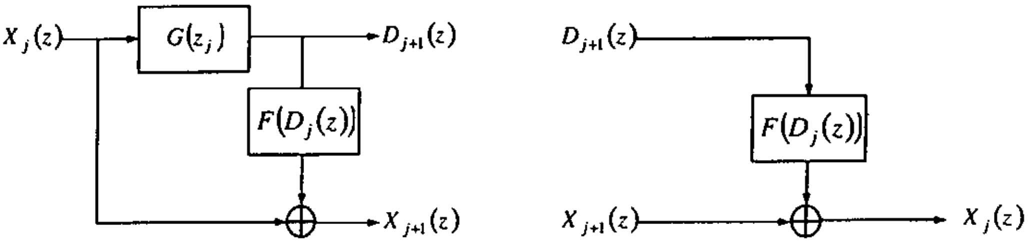


Figure 4.1, Fast algorithm for LOG-based wavelets. **(a)** Forward transform, **(b)** Inverse transform

For the complexity of the new structure, note that,

$$Cost\left(F\left[D_j(z)\right]\right)=(2n-2)Cost_{add}+(n)Cost_{multiply}$$

Note 4.2: The computational complexities of the 1D forward and inverse wavelet transforms of the proposed spatial algorithm are,

$$Cost_S(Forward) = (2n+1)Cost_{add} + (n+2)Cost_{multiply}$$

$$Cost_S(Inverse) = (2n-1)Cost_{add} + (n)Cost_{multiply}$$

Proof: The forward complexity equals the sum of the costs of $G(z)$, $F[D(z)]$, and one addition operation as shown in eqn.4.1-15. The inverse complexity equals the sum of the cost of $F[D(z)]$ and one addition.□

4.3. Analysis of the spatial approach

The spatial approach can efficiently reduce the computational complexity of both forward and inverse wavelet transforms. Comparing Note 4.1 and Note 4.2, the complexity of the forward wavelet transform is slightly decreased, while for the inverse wavelet transform it is greatly reduced (nearly 3 times) due to the use of a simple spatial approach for reconstruction.

For instance, the Spline wavelet (*n*=*2*) achieves a saving of 1 addition and 1 multiplication in the forward wavelet transform. In the inverse transform, the number of additions is reduced from 10 to 3 and multiplications from 7 to 2.

A comparative study was also carried out experimentally. The average computation times using both filtering and spatial approaches are shown in Figure 4.2. Consistent with the theoretical studies, the spatial approach is more efficient than the filtering approach.

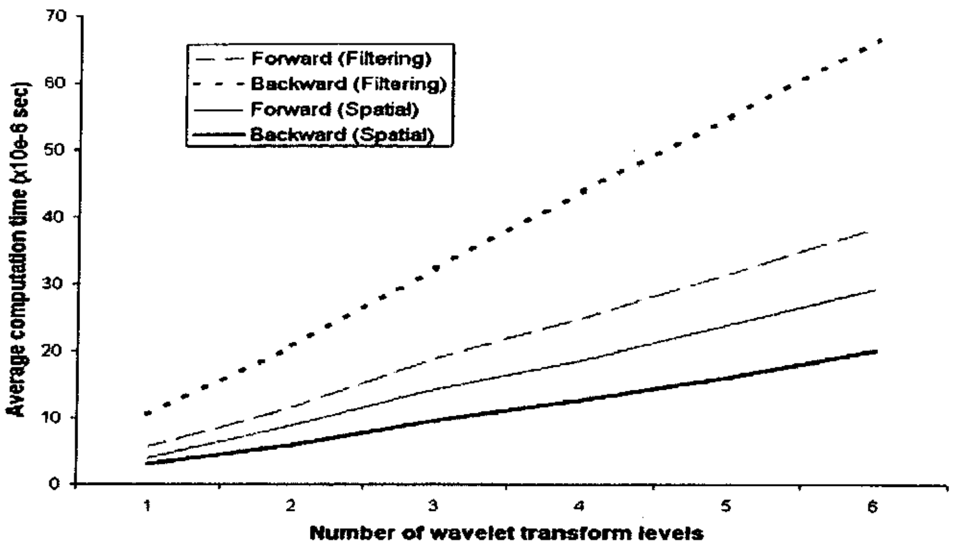


Figure 4.2, Comparison of computation time for the Spline wavelet (n=2), using filtering approach and spatial approach, versus different levels of wavelet transform.

Since $F\left[D(z)\right]$ remains the same in both forward and inverse wavelet transforms, the spatial approach does not require boundary corrections after reconstruction as in the filtering approach. Any boundary extension scheme can be used while maintaining the ease of the inverse transform.

Our proposed approach is also able to achieve a lossless transform. The denominator $\left(2^{2n}\right)$ in eqn.3 can be ignored to give integer filter coefficients. As the coefficients are integer and no floating point error is introduced in the proposed spatial approach, the wavelet transform becomes lossless.

4.4. Summary

In summary, our approach reduces the computational complexity, handles boundary extension in a flexible way and also allows a lossless wavelet transform.

CHAPTER 5

5. Fast algorithm for 2D Laplacian of Gaussian-based over-complete wavelet

The Laplacian of Gaussian (LOG)-based wavelet kernel has recently been proposed for texture feature analysis and retrieval as discussed in Chapter 3. It belongs to the class of the over-complete wavelet scheme which was proposed by Mallat and Zhong[9]. There are several advantages of using this kernel. First, it has been shown that this kernel can achieve the highest retrieval rate as compared to kernels from various sub-sampling wavelet schemes[70-73] and the Gabor wavelet[24]. Second, it allows a simple implementation structure due to its separable nature. Because of the translational invariant property, a robust performance under different levels of noises has also been reported in section 3.3.4.

Despite these advantages, a major issue in a real-time texture analysis and retrieval system is the computational complexity. Implementation using conventional filtering approach is inefficient and can cause boundary artifacts unless proper compensation is done either before or after the transform. By deriving some general expressions for the computational cost using the conventional filtering implementation, it reveals that the inverse transform is significantly more costly than the forward transform. A class of recent techniques[52-55], commonly know as the lifting scheme, shows that a pure spatial implementation can greatly reduce the computation time in the wavelet transform and help remove the artifacts. This approach is based on finding the common computations between the lowpass and the bandpass filters in the wavelet kernel to reduce the complexity.

In this Chapter, we identify common computations within the 2 dimensional LOG-based wavelet kernel in the over-complete wavelet scheme. Similar to the lifting scheme in the sub-sampling wavelet scheme, it results in an efficient pure spatial implementation. Both theoretical and experimental studies show that the new implementation can reduce the computational cost of the wavelet transform. In particular, the cost associated with the inverse transform is reduced significantly. Furthermore, the new implementation can completely eliminate the boundary artifacts without any extra boundary compensation.

This Chapter is organized as follows. Section 5.1 provides an overview of the LOG-based over-complete wavelet representation. In particular, the computational costs for the forward and the inverse transforms using the conventional filtering approach are given. Section 5.2 introduces the new spatial approach of the wavelet kernel and analyzes its computational cost. The boundary artifact that may occur in the filtering approach and how it can be eliminated by the new approach are discussed. Section 5.3 discusses the texture analysis using the new approach. In Section 5.4, comparative experiments on the complexity of the forward and the inverse transforms between the filtering approach and the proposed spatial approach are given. A comparative study, using the LOG-based wavelet kernel and other texture analysis methods, of retrieval accuracy on using the entire Brodatz texture database is also presented. Finally, Section 5.5 concludes this Chapter.

5.1. The over-complete wavelet representation using the LOG-based wavelet kernel

5.1.1. The 2D dyadic LOG-based over-complete wavelet representation

The over-complete wavelet transform of a pattern such as an image is obtained by applying filters separately to both the horizontal and the vertical directions. For the 2D dyadic LOG-based over-complete wavelet representation, three filtering responses are obtained at each level of decomposition. They are the lowpass approximation of the original image and two bandpass outputs - one bandpass output shows the horizontal feature while the other shows the vertical feature of an image. Mathematically, the lowpass output at the j^{th} dyadic level is given by

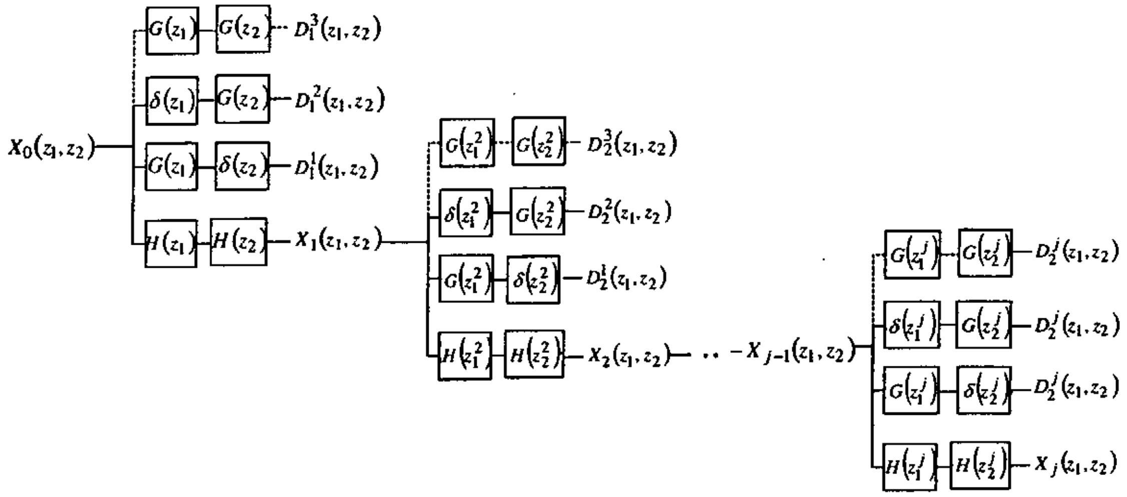
$$X_{j+1}(z_1,z_2)=H(z_{1,j})H(z_{2,j})X_j(z_1,z_2) \qquad (5.1\text{-}1)$$

where $z_{k,j}=z_k^{2^{j-1}}$ and k denotes the direction of the filter, $z_{1,j}$ and $z_{2,j}$ denote the horizontal and the vertical direction samples at the j^{th} dyadic level of decomposition respectively. $X_j(z_1,z_2)$ is the j^{th} level lowpass approximation of the original image. The two bandpass outputs can be written as

$$D_{j+1}^1(z_1,z_2)=G(z_{1,j})\delta(z_{2,j})X_j(z_1,z_2) \qquad (5.1\text{-}2)$$

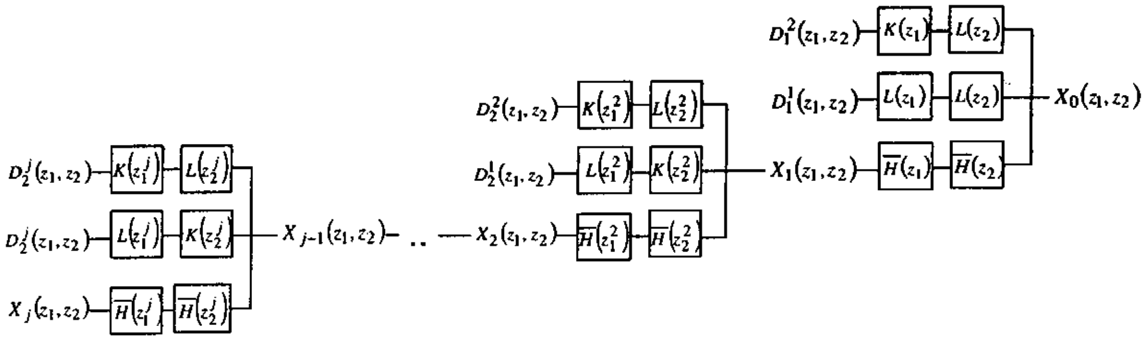
$$D_{j+1}^2(z_1,z_2)=\delta(z_{1,j})G(z_{2,j})X_j(z_1,z_2) \qquad (5.1\text{-}3)$$

where δ, G and H are delta function, the 1D bandpass filter and the 1D lowpass filter, respectively. $D_{j+1}^1(z_1,z_2)$ and $D_{j+1}^2(z_1,z_2)$ are the responses of the bandpass filters in the horizontal and the vertical directions at the j^{th} level of decomposition, respectively. Figure 5.1 shows the general structure of different levels of the wavelet transform. For the inverse over-complete wavelet transform, the lowpass approximation at the j^{th} level can be reconstructed as shown in eqn.5.1-4,



X0(z1,z2)
G(z1) G(z2) D1^3(z1,z2)
δ(z1) G(z2) D1^2(z1,z2)
G(z1) δ(z2) D1^1(z1,z2)
H(z1) H(z2) X1(z1,z2)
G(z1^2) G(z2^2) D2^3(z1,z2)
δ(z1^2) G(z2^2) D2^2(z1,z2)
G(z1^2) δ(z2^2) D2^1(z1,z2)
H(z1^2) H(z2^2) X2(z1,z2) .. X_{j-1}(z1,z2)
G(z1^j) G(z2^j) Dj^j(z1,z2)
δ(z1^j) G(z2^j) D2^j(z1,z2)
G(z1^j) δ(z2^j) D2^j(z1,z2)
H(z1^j) H(z2^j) X_j(z1,z2)

Figure 5.1, Forward over-complete wavelet transform of the jth level. The solid line shows the traditional filtering transform, while the dotted line shows the third direction for texture feature analysis [9].



D1^2(z1,z2) K(z1) L(z2)
D1^1(z1,z2) L(z1) L(z2)
X0(z1,z2)
X1(z1,z2) H̄(z1) H̄(z2)
D2^2(z1,z2) K(z1^2) L(z2^2)
D2^1(z1,z2) L(z1^2) K(z2^2)
X2(z1,z2) H̄(z1^2) H̄(z2^2)
D2^j(z1,z2) K(z1^j) L(z2^j)
D2^j(z1,z2) L(z1^j) K(z2^j)
X_j(z1,z2) H̄(z1^j) H̄(z2^j)
X_{j-1}(z1,z2) ..

Figure 5.2, Inverse over-complete wavelet transform of the jth level.

$$X_j(z_1,z_2)=K(z_{1,j+1})L(z_{2,j+1})D^1_{j+1}(z_1,z_2) + L(z_{1,j+1})K(z_{2,j+1})D^2_{j+1}(z_1,z_2)+\overline{H}(z_{1,j+1})\overline{H}(z_{2,j+1})X_{j+1}(z_1,z_2) \qquad (5.1\text{-}4)$$

where $\overline{H}$ is the time inverse of H, K and L are the 1D bandpass reconstruction filters. The reconstruction framework is shown in Figure 5.2. Similar to the forward transform, the inverse filter is applied separately to the horizontal and the vertical directions. By substituting eqn.5.1-1, eqn.5.1-2 and eqn.5.1-3 to eqn.5.1-4, the perfect reconstruction constraint can be stated as follows,

$$K\left(z_{1,j+1}\right)L\left(z_{2,j+1}\right)G\left(z_{1,j+1}\right)+L\left(z_{1,j+1}\right)K\left(z_{2,j+1}\right)G\left(z_{2,j+1}\right)$$
$$+\overline{H}\left(z_{1,j+1}\right)\overline{H}\left(z_{2,j+1}\right)H\left(z_{1,j+1}\right)H\left(z_{2,j+1}\right)=1 \qquad (5.1\text{-}5)$$

Eqn.5.1-5 shows the condition for perfect reconstruction of the original image. One can see that there is considerable freedom in choosing these four filters in an over-complete wavelet scheme. Since points of sharp variations, such as edges, are usually one of the most important features for analysing and characterizing patterns, a way of choosing these filters is to construct these filters so that the resultant wavelet transform could characterize those sharp variations in an image. In [9], the 1D lowpass and bandpass filters, at the j^{th} level transform, for the order n are written respectively as,

$$H(z_{k,j})=\frac{z_{k,j}^{n}}{2^{2n}}\left(1+z_{k,j}^{-1}\right)^{2n} \qquad (5.1\text{-}6)$$

and

$$G(z_{k,j})=4z_{k,j}^{-1}-8+4z_{k,j} \qquad (5.1\text{-}7)$$

The wavelet function corresponds to the second derivative of a smoothing function. Thus this set of kernels corresponds to the Laplacian of Gaussian filter and the zero-crossing point shows the sharp variations such as edges in a pattern. Following the analysis from the 1D framework, the reconstruction filter, $K\left(z_{k,j}\right)$, can be expressed as

$$K\left(z_{k,j}\right)=\frac{1-\left|H\left(z_{k,j}\right)\right|^{2}}{G\left(z_{k,j}\right)}$$

Substituting eqn.5.1-6 and eqn.5.1-7 to the above expression, $K\left(z_{k,j}\right)$ can be rewritten as

$$K\left(z_{k,j}\right)=\left(-\frac{1}{16}\right)\left[\sum_{m=0}^{2n-1}\left(\frac{z_{k,j}^{-1/2}+z_{k,j}^{1/2}}{2}\right)^{2m}\right] \tag{5.1-8}$$

To achieve perfect reconstruction, it is required that,

$$L\left(z_{k,j}\right)=\frac{1+\left|H\left(z_{k,j}\right)\right|^{2}}{2} \tag{5.1-9}$$

Substituting eqn.5.1-6 into eqn.5.1-9, $L\left(z_{k,j}\right)$ can be expanded as,

$$L\left(z_{k,j}\right)=\frac{1}{2}\left[1+\left(\frac{z_{k,j}^{-1/2}+z_{k,j}^{1/2}}{2}\right)^{4n}\right] \tag{5.1-10}$$

For the texture feature analysis system proposed in [9], a third bandpass response, denoted as $D_j^3\left(z_1,z_2\right)$, is included to characterize the diagonal feature in a pattern. It is obtained by applying the 1D bandpass filter to the responses of both the vertical and the horizontal directions, i.e.

$$D_{j+1}^3\left(z_1,z_2\right)=G\left(z_{1,j}\right)G\left(z_{2,j}\right)X_j\left(z_1,z_2\right) \tag{5.1-11}$$

5.1.2. Filter complexity

To measure the computational costs required in the LOG-based wavelet kernel, we need to expand the filter expressions and find out the numbers of additions and multiplications involved. The bandpass filter G has a fixed number of coefficients and it requires 2 additions and 1 multiplication only. The computation cost is written as,

$$Cost(G)=\left(2\right)Cost_{add}+\left(2\right)Cost_{multiply} \tag{5.1-12}$$

where $Cost_{add}$ and $Cost_{multiply}$ are the computational costs of an addition and a multiplication operations respectively. For analysing the cost of the lowpass filter H, eqn.5.1-6 is expanded by using the Binomial theorem as follows,

$$H\left(z_{k,j}\right)=\frac{1}{2^{2n}}\left(\sum_{m=0}^{n-1}{}^{2n}C_m\left[z_{k,j}^{n-m}+z_{k,j}^{-(n-m)}\right]+{}^{2n}C_n\right) \tag{5.1-13}$$

where

$${}^{n}C_m=\frac{n!}{(n-m)!m!} \tag{5.1-14}$$

Eqn.5.1-13 shows that the cost of the lowpass filter H is,

$$Cost(H)=(2n)Cost_{add}+(n+1)Cost_{multiply} \tag{5.1-15}$$

In reconstruction, we need to find the costs of $\overline{H}$, K and L. Since $\overline{H}$ is the time inverse of H, the filter expression can simply be obtained by replacing z with z^{-1} in eqn.5.1-13. The cost of $\overline{H}$ is thus the same as H, i.e.

$$Cost(\overline{H})=(2n)Cost_{add}+(n+1)Cost_{multiply} \tag{5.1-16}$$

In calculating the complexity of K, we expand the summation in eqn.5.1-8 and use Binomial theorem to obtain the following expression,

$$K\left(z_{k,j}\right)=B_0+\sum_{m=1}^{2n-1}B_m\left[z_{k,j}^{-m}+z_{k,j}^{m}\right] \tag{5.1-17}$$

where

$$B_m=\left(-\frac{1}{16}\right)\left(\sum_{k=m}^{2n-1}\frac{1}{2^{2k}}{}^{2k}C_{k-m}\right) \tag{5.1-18}$$

The computation cost of K in eqn.5.1-17 can be written as,

$$Cost(K) = (4n-2)Cost_{add} + (2n)Cost_{multiply} \tag{5.1-19}$$

Again, by employing the Binomial theorem, an expression for L can be obtained by expanding eqn.5.1-10 as follows,

$$L\left(z_{k,j}\right) = \frac{1}{2} + \frac{1}{2^{4n+1}}\left\{\sum_{m=0}^{2n-1} {}^{4n}C_k\left[z_{k,j}^{-(2n-m)} + z_{k,j}^{2n-m}\right] + {}^{4n}C_{2n}\right\} \tag{5.1-20}$$

Rearrange eqn.5.1-20, it can be rewritten as,

$$L\left(z_{k,j}\right) = D + \sum_{m=0}^{2n-1} E_m\left[z_{k,j}^{-(2n-m)} + z_{k,j}^{2n-m}\right] \tag{5.1-21}$$

where

$$D = \frac{1}{2} + \frac{{}^{4n}C_{2n}}{2^{4n+1}} \quad \text{and} \quad E_m = \frac{{}^{4n}C_m}{2^{4n+1}}$$

The computation cost of L in eqn.5.1-21 means that,

$$Cost(L) = (4n)Cost_{add} + (2n+1)Cost_{multiply} \tag{5.1-22}$$

5.1.3. Computational cost

In each level of forward transform, both filters in the horizontal and the vertical directions are involved as shown in Figure 5.1. Substituting the filter expression for $H\left(z_{k,j}\right)$ in eqn.5.1-13 and eqn.5.1-1, the lowpass output $X_{j+1}(z_1, z_2)$, can be rewritten as follows,

$$\begin{aligned} X_{j+1}(z_1, z_2) = \frac{1}{2^{4n}} &\left(\sum_{k_1=0}^{n-1} {}^{2n}C_{k_1}\left[z_{1,j}^{-(n-k_1)} + z_{1,j}^{n-k_1}\right] + {}^{2n}C_n\right) \\ &\times\left(\sum_{k_2=0}^{n-1} {}^{2n}C_{k_2}\left[z_{2,j}^{-(n-k_2)} + z_{2,j}^{n-k_2}\right] + {}^{2n}C_n\right) X_j(z_1, z_2) \end{aligned} \tag{5.1-23}$$

Since the scalar factors in both $H(z_1)$ and $H(z_2)$ can be merged as in eqn.5.1-23, one multiplication is saved. Consequently, the complexity of obtaining $X_{j+1}(z_1, z_2)$ equals to two times the complexity of H and one less multiplication, i.e.,

$$Cost\left(X_{j+1}(z_1, z_2)\right) = (4n)Cost_{add} + (2n+1)Cost_{multiply} \tag{5.1-24}$$

The bandpass outputs, $D_{j+1}^1(z_1, z_2)$ and $D_{j+1}^2(z_1, z_2)$, can be found by substituting eqn.5.1-7 to eqn.5.1-2 and eqn.5.1-3 as follows,

$$D_{j+1}^1(z_1, z_2) = 4\left(z_{1,j}^{-1} - 2 + z_{1,j}\right)X_j(z) \tag{5.1-25}$$

$$D_{j+1}^2(z_1, z_2) = 4\left(z_{2,j}^{-1} - 2 + z_{2,j}\right)X_j(z) \tag{5.1-26}$$

Thus, the complexity of $D_{j+1}^1(z_1, z_2)$ and $D_{j+1}^2(z_1, z_2)$ can be written as,

$$Cost\left(D_{j+1}^1(z_1, z_2)\right) = Cost\left(D_{j+1}^2(z_1, z_2)\right) = (2)Cost_{add} + (2)Cost_{multiply} \tag{5.1-27}$$

To obtain the complexity of the inverse transform, we consider the complexities of $K(z_{1,j})L(z_{2,j})D_j^1(z_1, z_2)$, $L(z_{1,j})K(z_{2,j})D_j^2(z_1, z_2)$ and $\overline{H}(z_{1,j})\overline{H}(z_{2,j})X_j(z_1, z_2)$ as in eqn.5.1-4. The structure of multi-level inverse transform is shown in Figure 5.2. Using the filter expressions in eqn.5.1-17 and eqn.5.1-20, $K(z_{1,j})L(z_{2,j})D_j^1(z_1, z_2)$ can be expanded as,

$$\begin{aligned} &K(z_{1,j})L(z_{2,j})D_j^1(z_1, z_2) = \\ &\quad \left\{B_0 + \sum_{m=1}^{2n-1} B_m\left[z_{1,j}^{-m} + z_{1,j}^{m}\right]\right\}\left\{D + \sum_{m=0}^{2n-1} E_m\left[z_{2,j}^{-(2n-m)} + z_{2,j}^{2n-m}\right]\right\} D_j^1(z_1, z_2) \end{aligned} \tag{5.1-28}$$

The complexity of filters K and L are,

$$Cost(K) = (4n-2)Cost_{add} + (2n)Cost_{multiply} \tag{5.1-29}$$

$$Cost(L) = (4n)Cost_{add} + (2n+1)Cost_{multiply} \tag{5.1-30}$$

Thus, the complexity of $K(z_{1,j})L(z_{2,j})D_j^1(z_1,z_2)$ is equal to the sum of the complexities in K and L i.e.

$$Cost\left(K(z_{1,j})L(z_{2,j})D_j^1(z_1,z_2)\right) = (8n-2)Cost_{add} + (4n+1)Cost_{multiply} \tag{5.1-31}$$

The complexity of $L(z_{1,j})K(z_{2,j})D_j^2(z_1,z_2)$ is also equal to the sum of the complexity in K and L, i.e.,

$$Cost\left(L(z_{1,j})K(z_{2,j})D_j^2(z_1,z_2)\right) = (8n-2)Cost_{add} + (4n+1)Cost_{multiply} \tag{5.1-32}$$

Let $\overline{H}(z_{1,j})\overline{H}(z_{2,j})$ be the time conjugate of $H(z_{1,j})H(z_{2,j})$. Therefore, its complexity is equal to the complexity of $H(z_{1,j})H(z_{2,j})$, i.e.,

$$Cost\left(\overline{H}(z_{1,j})\overline{H}(z_{2,j})\right) = (4n)Cost_{add} + (2n+1)Cost_{multiply} \tag{5.1-33}$$

Based on the complexities shown in eqn.5.1-24, eqn.5.1-27, eqn.5.1-31, eqn.5.1-32 and eqn.5.1-33, we can obtain the computation costs of the forward transform and the inverse transform. It is summarized as in theorem 5.1.

Theorem 5.1: The computational cost of a single level 2D LOG-based over-complete forward wavelet transform using the filtering approach is written as,

$$Cost_F(Forward) = (4n+4)Cost_{add} + (2n+5)Cost_{multiply} \tag{5.1-34}$$

The inverse transform using the filtering approach is written as,

$$Cost_F(Inverse) = (20n-4)Cost_{add} + (10n+3)Cost_{multiply} \tag{5.1-35}$$

Proof: At the j^{th} level forward wavelet transform, there are three outputs: $X_j(z_1,z_2)$, $D_j^1(z_1,z_2)$ and $D_j^2(z_1,z_2)$. The complexity of the forward transform can thus be obtained by summing their complexities as shown in eqn.5.1-24 and eqn.5.1-27. The operations in the inverse transform involve three parts; namely $K(z_{1,j})L(z_{2,j})D_j^1(z_1,z_2)$, $L(z_{1,j})K(z_{2,j})D_j^2(z_1,z_2)$ and $\overline{H}(z_{1,j})\overline{H}(z_{2,j})X_j(z_1,z_2)$. The inverse complexity is thus obtained by summing their complexities shown in eqn.5.1-31, eqn.5.1-32 and eqn.5.1-33.□

From Theorem 5.1, the inverse transform has approximately 5 times more computational cost than the forward transform. It is due to the fact that the inverse filters always have a longer coefficient length than the forward filters. It is undesirable in image analysis to have a higher computational complexity for reconstruction.

For texture feature analysis, the third bandpass output, $D_j^3(z_1,z_2)$, is included in the forward transform. It involves bandpass filtering in both horizontal and vertical directions, i.e., $G(z_{1,j})$ and $G(z_{2,j})$. In actual implementation, $D_{j+1}^3(z_1,z_2)$ can be calculated from either convolving $D_j^1(z_1,z_2)$ with $G(z_{2,j})$ or convolving $D_j^2(z_1,z_2)$ with $G(z_{1,j})$ as shown in Figure 5.1. The complexity of the forward transform for texture analysis thus requires only one more cost in filter G. The computation costs $Cost_F^T$ for the texture can thus be written as,

$$Cost_F^T(Forward) = (4n+6)Cost_{add} + (2n+7)Cost_{multiply} \qquad (5.1\text{-}36)$$

5.2. Proposed spatial implementation for the LOG-based wavelet kernel

The filtering technique in the sub-sampling wavelet scheme exploits the common computations between the lowpass and the bandpass filters. This results in a saving of nearly 50% in computations. In fact this idea can be extended to the over-complete wavelet scheme because of its redundant nature. A major objective of our study is to investigate the common computation in filtering the lowpass and the bandpass outputs to simplify some of the computations, especially in the inverse wavelet transform. Furthermore, it is found that the boundary artifacts do not appear in the new implementation structure.

5.2.1. Single level wavelet transform using the LOG-based wavelets

Let us consider a single level wavelet transform. Before illustrating the new approach, Lemma 1 is requested to be defined first.

Lemma 1: The following expression

$$Y(z)=\frac{1}{2^{2n}}\left(\sum_{k=0}^{n-1}{}^{2n}C_k\left[z^{-(n-k)}+z^{n-k}\right]+{}^{2n}C_n\right)X_0(z) \qquad (5.2\text{-}1)$$

can be written as,

$$Y(z)=X_0(z)+D_1(z)\left\{E_n+\sum_{m=1}^{n-1}E_{n-m}\cdot\left(z^{-m}+z^{m}\right)\right\} \qquad (5.2\text{-}2)$$

where

$$D_1(z)=G(z)X_0(z)=4\left(z^{-1}+2+z\right)X_0(z) \qquad (5.2\text{-}3)$$

and

$$E_m=\frac{1}{2^{2n+2}}\sum_{k=1}^{m}k\cdot{}^{2n}C_{m-k} \qquad (5.2\text{-}4)$$

Proof: Using Theorem A.1 which is shown in Appendix I, for an arbitrary m, we can write,

$$z^{-m}+z^{m}=\frac{1}{4}\left\{\left(\sum_{k=1}^{m-1}(m-k)\left[z^{-k}+z^{k}\right]\right)\cdot G(z)+m\cdot G(z)+8\right\} \qquad (5.2\text{-}5)$$

Substituting eqn.5.2-5 into eqn.5.2-1, we obtain,

$$Y(z)=X_0(z)+\frac{D_1(z)}{2^{2n+2}}\left\{\sum_{k=1}^{n}k\cdot {}^{2n}C_{n-k}+\sum_{m=1}^{n-1}\left(z^{-m}+z^{m}\right)\sum_{l=1}^{m}k\cdot {}^{2n}C_{m-l}\right\} \qquad (5.2\text{-}6)$$

Using eqn.5.2-4, eqn.5.2-6 becomes,

$$Y(z)=X_0(z)+D_1(z)\left\{E_n+\sum_{m=1}^{n-1}E_{n-m}\cdot\left(z^{-m}+z^{m}\right)\right\} \qquad (5.2\text{-}7)$$

which completes the proof.□

Using Lemma 1, the wavelet transform, along z_2 (the column) for every row of the image, can be written as,

$$H(z_2)X_0(z_1,z_2)=X_0(z_1,z_2)+D_1^2(z_1,z_2)\left\{E_n+\sum_{m=1}^{n-1}E_{m-l}\cdot\left(z_2^{-m}+z_2^{m}\right)\right\} (5.2\text{-}8)$$

Using the expression in eqn.5.2-8, we can arrive at Theorem 5.2. This theorem provides an alternative method for the implementation of the LOG-based wavelet kernel.

Theorem 5.2: The first level lowpass output of the over-complete wavelet transform using the LOG-based wavelet kernel with an arbitrary order n can be rewritten as,

$$X_1(z_1,z_2)=X_0(z_1,z_2)+F_1\left\{D_1^2(z_1,z_2)\right\}+F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\} \qquad (5.2\text{-}9)$$

where

$$F_1\left\{D_1^2(z_1,z_2)\right\}=Y_1\left[D_1^2(z_1,z_2)\right]=\left\{E_n+\sum_{l_2=1}^{n-1}E_{n-l_2}\left[z_2^{-l_2}+z_2^{l_2}\right]\right\}D_1^2(z_1,z_2) \qquad (5.2\text{-}10)$$

and

$$\begin{aligned}&F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\}\\ &\qquad=\left\{E_n+\sum_{l_1=1}^{n-1}E_{n-l_1}\left[z_1^{-l_1}+z_1^{l_1}\right]\right\}\left\{D_1^1(z_1,z_2)+G(z_1)Y_1\left[D_1^2(z_1,z_2)\right]\right\}\end{aligned} \qquad (5.2\text{-}11)$$

Proof: We start with substituting eqn.5.2-10 to eqn.5.1-1

$$X_1(z_1,z_2)=H(z_1)X_0(z_1,z_2)+H(z_1)\left\{E_n+\sum_{l_2=1}^{n-1}E_{n-l_2}\left[z_2{}^{-l_2}+z_2{}^{l_2}\right]\cdot\right\}D_1^2(z_1,z_2)$$

$$=\left\{\begin{array}{l}X_0(z_1,z_2)+\left\{E_n+\sum\limits_{l_1=1}^{n-1}E_{n-l_1}\left[z_1{}^{-l_1}+z_1{}^{l_1}\right]\cdot\right\}D_1^1(z_1,z_2)\\ +\dfrac{D_1^2(z_1,z_2)}{2^{2n+2}}\left\{\sum\limits_{k_1=0}^{n-1}{}^{2n}C_{k_1}\left[z_1^{-k_1}+z_1^{k_1}\right]+{}^{2n}C_n\right\}\left\{E_n+\sum\limits_{l_2=1}^{n-1}E_{n-l_2}\left[z_2{}^{-l_2}+z_2{}^{l_2}\right]\cdot\right\}\end{array}\right\}$$

(5.2-12)

The last term in eqn.5.2-12 can be rewritten as,

$$\frac{D_1^2(z_1,z_2)}{2^{2n+2}}\left\{\sum_{k_1=0}^{n-1}{}^{2n}C_{k_1}\left[z_1{}^{-(n-k_1)}+z_1^{n-k_1}\right]+{}^{2n}C_n\right\}\left\{E_n+\sum_{l_2=1}^{n-1}E_{n-l_2}\left[z_2{}^{-l_2}+z_2{}^{l_2}\right]\cdot\right\}$$

$$=\left\{\begin{array}{l}\left\{D_1^2(z_1,z_2)+\dfrac{1}{2^{2n+2}}\left[G(z_1)D_1^2(z_1,z_2)\right]\cdot\left\{E_n+\sum\limits_{l_3=1}^{n-1}E_{n-l_3}\left[z_1{}^{-l_3}+z_1{}^{l_3}\right]\right\}\right\}\\ \times\left\{E_n+\sum\limits_{l_2=1}^{n-1}E_{n-l_2}\left[z_2{}^{-l_2}+z_2{}^{l_2}\right]\cdot\right\}\end{array}\right.$$

$$=Y_1\left[D_1^2(z_1,z_2)\right]+\frac{1}{2^{2n+2}}\left\{E_n+\sum_{l_3=0}^{n-1}E_{n-l_3}\left[z_1{}^{-l_3}+z_1{}^{l_3}\right]\right\}G(z_1)Y_1\left[D_1^2(z_1,z_2)\right]$$

(5.2-13)

Substituting eqn.5.2-13 into eqn.5.2-12, the first level lowpass output can be expressed as,

$$X_1(z_1,z_2)=\left\{\begin{array}{l} X_0(z_1,z_2)+\left\{E_n+\sum_{l_1=1}^{n-1}E_{n-l_1}\left[z_1^{-l_1}+z_1^{l_1}\right]\cdot\right\}D_1^1(z_1,z_2) \\ +Y_1\left[D_1^2(z_1,z_2)\right]+\frac{1}{2^{2n+2}}\left\{E_n+\sum_{l_3=1}^{n-1}E_{n-l_3}\left[z_1^{-l_3}+z_1^{l_3}\right]\right\}G(z_1)Y_1\left[D_1^2(z_1,z_2)\right] \end{array}\right\}$$

$$=\left\{\begin{array}{l} X_0(z_1,z_2)+Y_1\left[D_1^2(z_1,z_2)\right] \\ +\frac{1}{2^{2n+2}}\left\{E_n+\sum_{l_3=1}^{n-1}E_{n-l_3}\left[z_1^{-l_3}+z_1^{l_3}\right]\cdot\right\}\left\{D_1^1(z_1,z_2)+G(z_1)Y_1\left[D_1^2(z_1,z_2)\right]\right\} \end{array}\right.$$

(5.2-14)

which completes the proof.□

Theorem 5.2 not only provides an alternative implementation scheme for the forward transform, but also simplifies the computation for the inverse transform. In particular, the inverse transform can be easily calculated as follows,

$$X_0(z_1,z_2)=X_1(z_1,z_2)-F_1\left\{D_1^2(z_1,z_2)\right\}-F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\} \quad (5.2\text{-}15)$$

The proposed implementation structures for the forward and the inverse transforms according to Theorem 5.2 and eqn.5.2-15 are shown in Figure 5.3 and Figure 5.4 respectively. It can be seen that a simple spatial implementation is used for image reconstruction. It greatly simplifies the computation involved in the inverse transform.

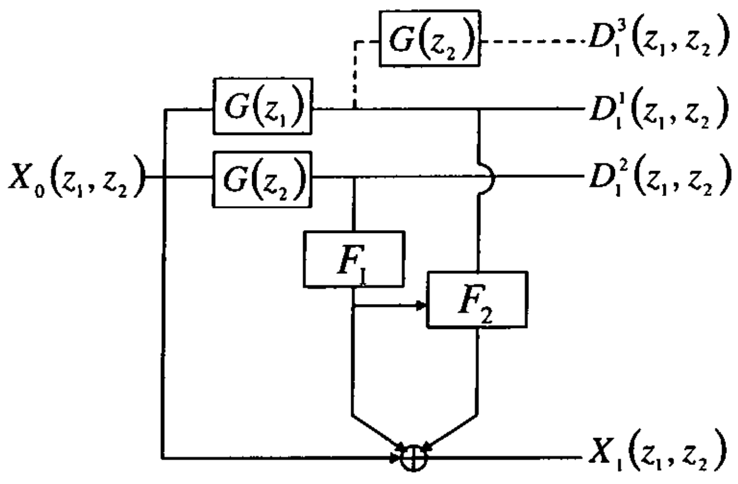


Figure 5.3, Proposed spatial implementation structure for the forward transform using the LOG-based wavelet kernel. Solid lines show the general structure of the wavelet transform. The Dotted line shows the third direction in texture analysis.

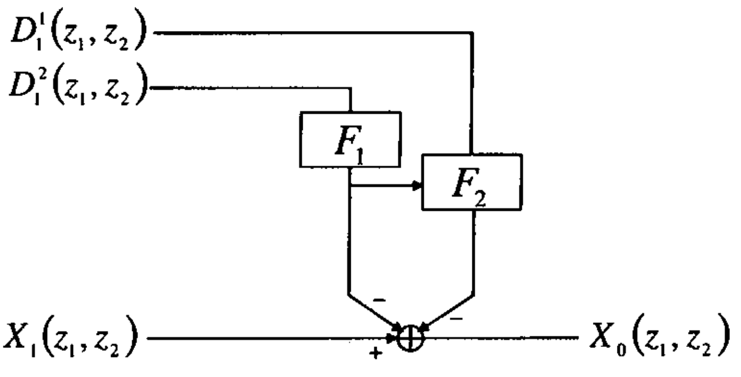


Figure 5.4, The proposed spatial implementation structure for the inverse transform using the LOG-based wavelet kernel.

5.2.2. Computational complexity of the new approach in the first stage transform

In analysing the computational complexity associated with the proposed scheme, the complexities in $F_1\{D_1^2(z_1,z_2)\}$ and $F_2\{D_1^1(z_1,z_2),Y_1[D_1^2(z_1,z_2)]\}$ given in eqn.5.2-12 and eqn.5.2-13 can be written as follows,

$$Cost_F\left(F_1\left\{D_1^1(z_1,z_2)\right\}\right)=(2n-2)Cost_{add}+(n)Cost_{multiply} \qquad (5.2\text{-}16)$$

$$Cost_F\left(F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\}\right)=(2n+1)Cost_{add}+(n+1)Cost_{multiply} \qquad (5.2\text{-}17)$$

Using both eqn.5.2-16 and eqn.5.2-17, the complexities for the forward and the inverse wavelet transforms can be defined as in Theorem 5.3.

Theorem 5.3: The LOG-based over-complete forward wavelet transform using the spatial implementation scheme described in Theorem 5.2 has a complexity of,

$$Cost_S(Forward)=(4n+5)Cost_{add}+(2n+5)Cost_{multiply} \qquad (5.2\text{-}18)$$

and the inverse has a complexity of,

$$Cost_S(Inverse)=(4n+1)Cost_{add}+(2n+1)Cost_{multiply} \qquad (5.2\text{-}19)$$

Proof: The forward transform has three outputs: two bandpass outputs, $D_1^1(z_1,z_2)$ and $D_1^2(z_1,z_2)$, and the lowpass output, $X_1(z_1,z_2)$. The costs for obtaining the two bandpass outputs are given in eqn.5.1-27. According to the new implementation scheme in Theorem 5.2, the lowpass output involves the calculation of $F_1\left\{D_1^1(z_1,z_2)\right\}$ and $F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\}$, its cost would include two additions, the complexities in obtaining $F_1\left\{D_1^1(z_1,z_2)\right\}$ and $F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\}$. By summing up these complexities, the forward complexity can be determined. In reconstruction, there is no need to calculate $D_1^1(z_1,z_2)$ and $D_1^2(z_1,z_2)$. Thus the inverse complexity would only involve two additions, plus the complexities in obtaining $F_1\left\{D_1^1(z_1,z_2)\right\}$ and $F_2\left\{D_1^1(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\}$.□

In the filtering approach, the cost of the forward transform is much smaller than that of the inverse transform as shown in Theorem 5.1. In contrast, the complexity of the forward transform is slightly higher than that of the inverse transform in the proposed implementation scheme as shown in Theorem 5.3. In comparing the filtering approach and the proposed scheme, we can see that the complexity of the forward transform of the proposed scheme has only one addition more than the filtering approach. This does not affect significantly the computational time in actual implementation. On the other hand, the complexity of the inverse transform of the proposed scheme is much lower than that of the filtering approach. In the proposed scheme, no filtering is required for the reconstruction of the original signal. Rather a simple spatial implementation is used for the reconstruction and thus the computational complexity of the inverse transform is greatly reduced. It can be seen that the inverse transform using the proposed scheme requires three multiplications and two additions less than the forward transform in the filtering approach.

Similar to the filtering approach for the texture analysis, the third direction $D_1^3(z_1, z_2)$ can be calculated from the convolution between $D_1^1(z_1, z_2)$ and $G_2(z_1, z_2)$ as shown in Figure 5.3. The cost can be calculated simply by adding the cost of $G(z)$ in eqn.5.1-12 to that of eqn.5.2-18, i.e.,

$$Cost_S^T(Forward) = (4n+7)Cost_{add} + (2n+7)Cost_{multiply} \qquad (5.2\text{-}20)$$

5.2.3. Multi-level wavelet decomposition using the proposed spatial approach

In the previous section, we explored the common operations between the lowpass and the bandpass outputs for the first level over-complete wavelet transform and provided a new spatial implementation structure for the wavelet transforms. In this section, we extend the proposed scheme to multiple level wavelet decomposition. As shown in Figure 5.1, the second level wavelet transform involves filters $G\left(z^2\right)$, $H\left(z^2\right)$, $K\left(z^2\right)$ and $L\left(z^2\right)$. Similar to the first level of wavelet transform, the two bandpass outputs and the lowpass approximation at the second level of decomposition can be written as,

$$D_2^1(z_1,z_2)=G\left(z_1^2\right)\delta\left(z_2^2\right)X_1(z_1,z_2) \tag{5.2-21}$$

$$D_2^2(z_1,z_2)=\delta\left(z_1^2\right)G\left(z_2^2\right)X_1(z_1,z_2) \tag{5.2-22}$$

$$X_2(z_1,z_2)=H\left(z_1^2\right)H\left(z_2^2\right)X_1(z_1,z_2) \tag{5.2-23}$$

Let us first analyse the computational complexity involved in calculating the second level output, $D_2^1(z_1,z_2)$, in comparing to the first level output, $D_1^1(z_1,z_2)$, eqn.5.2-21 can be rewritten as,

$$D_2^1(z_1,z_2)=G\left(z_1^2\right)X_{1,even}\left(z_1^2,z_2\right)+G\left(z_1^2\right)X_{1,odd}\left(z_1^2,z_2\right) \tag{5.2-24}$$

where $X_{1,even}(z_1,z_2)$ and $X_{1,odd}(z_1,z_2)$ are respectively the even and the odd parts of $X_1(z_1,z_2)$. As the length of either $X_{1,even}\left(z_1^2,z_2\right)$ or $X_{1,odd}\left(z_1^2,z_2\right)$ is only half of that of $X_1\left(z_1^2,z_2\right)$, the total number of operations involved in obtaining $D_2^1(z_1,z_2)$ is the same as that of $D_1^1(z_1,z_2)$. In fact, for any dyadic decomposition level, $j=2^{i-1}$,

$$\begin{aligned} D_j^1(z_1,z_2) &= G\left(z_1^{2^{i-1}}\right)X_{j-1}(z_1,z_2) \\ &= \sum_{k=0}^{2^{i-1}} G\left(z_1^{2^{i-1}}\right)X_{i-1,k}\left(z_1^{2^{i-1}},z_2\right) \end{aligned} \tag{5.2-25}$$

where

$$x'_{i,k}\left(n_1,n_2\right)=x_{i,k}\left(2^i n_1+k,n_2\right) \tag{5.2-26}$$

An analysis of eqn.5.2-26 shows that the number of operations involved in obtaining $D_j^1\left(z_1,z_2\right)$ remains the same as that of $D_1^1\left(z_1,z_2\right)$. Using eqn.5.2-22, a similar conclusion can be arrived for $D_j^2\left(z_1,z_2\right)$. Therefore, the computational complexities of the two bandpass outputs remain unchanged, regardless of the number of decomposition levels. Substituting eqn.5.1-23 to eqn.5.2-24, we obtain the second level lowpass response as shown below,

$$\begin{aligned}X_2\left(z_1,z_2\right)=\frac{1}{2^{4n}}\cdot&\left(\sum_{k_1=0}^{n-1}{}^{2n}C_{k_1}\left[z_1^{2(n-k_1)}+z_1^{-2(n-k_1)}\right]+{}^{2n}C_n\right)\\&\times\left(\sum_{k_2=0}^{n-1}{}^{2n}C_{k_2}\left[z_2^{2(n-k_1)}+z_2^{-2(n-k_1)}\right]+{}^{2n}C_n\right)X_1\left(z_1,z_2\right)\end{aligned} \tag{5.2-27}$$

Upon comparing eqn.5.2-25 and eqn.5.2-28, it shows that both expressions look very similar if z is replaced by z^2 in eqn.5.2-28. The analysis in Theorem 5.2 can thus be extended to obtain $X_2\left(z_1,z_2\right)$ by replacing $D_1^1\left(z_1,z_2\right)$ and $D_2^1\left(z_1,z_2\right)$ with $D_1^2\left(z_1,z_2\right)$ and $D_2^2\left(z_1,z_2\right)$ respectively, and z with z^2. This takes into account the filter interpolation in the subsequent decomposition levels. Mathematically, the second level lowpass output can be obtained as follows,

$$X_2\left(z_1,z_2\right)=X_1\left(z_1,z_2\right)+F_1\left\{D_2^2\left(z_1,z_2\right)\right\}+F_2\left\{D_1^1\left(z_1,z_2\right),Y_1\left[D_1^2\left(z_1,z_2\right)\right]\right\} \tag{5.2-28}$$

where

$$F_1\left\{D_2^2\left(z_1,z_2\right)\right\}=Y_2\left[D_2^2\left(z_1,z_2\right)\right]=\left\{E_n+\sum_{l_2=1}^{n-1}E_{n-l_2}\left[z_2^{2\cdot(-l_2)}+z_2^{2\cdot(l_2)}\right]\right\}D_2^2\left(z_1,z_2\right) \tag{5.2-29}$$

and

$$F_2\left\{D_2^1(z_1,z_2),Y_2\left[D_2^2(z_1,z_2)\right]\right\}$$
$$=\left\{E_n+\sum_{l_1=1}^{n-1}E_{n-l_1}\left[z_1^{2\cdot(-l_1)}+z_1^{2\cdot(l_1)}\right]\right\}\left\{D_2^1(z_1,z_2)+G\left(z_1^2\right)\cdot Y_2\left[D_2^2(z_1,z_2)\right]\right\}$$

(5.2-30)

Using similar argument for an arbitrary decomposition level, we arrive at the following theorem.

Theorem 5.4: For a LOG-based wavelet kernel, the j^{th} level lowpass output can be rewritten as,

$$X_{j+1}(z_1,z_2)=X_j(z_1,z_2)+F_1\left\{D_j^2(z_1,z_2)\right\}+F_2\left\{D_j^1(z_1,z_2),Y_j\left[D_j^2(z_1,z_2)\right]\right\} \quad (5.2\text{-}31)$$

where

$$F_1\left\{D_j^2(z_1,z_2)\right\}=Y_j\left[D_j^2(z_1,z_2)\right]=\left\{E_n+\sum_{l_2=1}^{n-1}E_{n-l_2}\left[z_{2,j}^{-l_2}+z_{2,j}^{l_2}\right]\right\}D_j^2(z_1,z_2) \quad (5.2\text{-}32)$$

and

$$F_2\left\{D_j^1(z_1,z_2),Y_j\left[D_j^2(z_1,z_2)\right]\right\}$$
$$=\left\{E_n+\sum_{l_1=1}^{n-1}E_{n-l_1}\left[z_{1,j}^{-l_1}+z_{1,j}^{l_1}\right]\right\}\left[D_j^1(z_1,z_2)+G\left(z_1^j\right)\cdot Y_j\left[D_j^2(z_1,z_2)\right]\right]$$

(5.2-33)

Proof: Using Lemma1 and eqn.5.1-1, we can obtain the lowpass response from the j^{th} decomposition level as,

$$X_{j+1}(z_1,z_2)=H\left(z_{1,j}\right)X_j(z_1,z_2)+H\left(z_{1,j}\right)\left\{E_n+\sum_{l_2=1}^{n-1}E_{n-l_2}\left[z_{2,j}^{-l_2}+z_{2,j}^{l_2}\right]\cdot\right\}D_j^2(z_1,z_2)$$

$$= \left\{ \begin{array}{l} X_j(z_1,z_2) + \left\{ E_n + \sum_{l_1=1}^{n-1} E_{n-l_1}\left[z_{1,j}^{-l_1} + z_{1,j}^{l_1} \right] \cdot \right\} D_j^1(z_1,z_2) \\ + \frac{D_j^2(z_1,z_2)}{2^{2n+2}} \left\{ \sum_{k_1=0}^{n-1} {}^{2n}C_{k_1}\left[z_{1,j}^{-(n-k_1)} + z_{1,j}^{n-k_1} \right] + {}^{2n}C_n \right\} \left\{ E_n + \sum_{l_2=1}^{n-1} E_{n-l_2}\left[z_{2,j}^{-l_2} + z_{2,j}^{l_2} \right] \cdot \right\} \end{array} \right\}$$

(5.2-34)

Following the analysis in eqn.5.2-14 and eqn.5.2-15, we will arrive at the prediction terms as shown in eqn.5.2-32 and eqn.5.2-33.□

Theorem 5.4 provides a general expression for the j^{th} level lowpass output. By comparing Theorem 5.4 and Theorem 5.2, it shows that the number of operations remains unchanged for all j. Therefore, the computational complexity of the lowpass output is independent of the decomposition level. In summary, the computational complexity of the proposed implementation remains the same as in Theorem 5.3 for an arbitrary level of decomposition.

5.2.4. Design examples

We provide some low order LOG-based wavelet examples using the new approach in this section. Table 5.1 shows the coefficients for kernels with different orders using the filtering approach. Substituting $n=1$ into eqn.5.2-9, a new implementation for the lowpass output becomes,

$$X_1(z_1,z_2) = X_0(z_1,z_2) + F_1\left\{D_1^2(z_1,z_2)\right\} + F_2\left\{D_1^1(z_1,z_2), Y_1\left[D_1^2(z_1,z_2)\right]\right\} \quad (5.2\text{-}35)$$

where

$$F_1\left\{D_1^2(z_1,z_2)\right\} = \left(\frac{1}{16}\right) D_1^2(z_1,z_2) \text{ and}$$

$$F_2\left\{D_1^1(z_1,z_2), Y_1\left[D_1^2(z_1,z_2)\right]\right\} = \left(\frac{1}{16}\right)\left\{D_1^1(z_1,z_2) + G(z_1)Y_1\left[D_1^1(z_1,z_2)\right]\right\}$$

For $n=2$, the lowpass output becomes,

$$X_1(z_1,z_2)=X_0(z_1,z_2)+F_1\{D_1^2(z_1,z_2)\}+F_2\{D_1^1(z_1,z_2),Y_1[D_1^2(z_1,z_2)]\} \quad (5.2\text{-}36)$$

where

$$F_1\{D_1^2(z_1,z_2)\}=Y_1[D_1^2(z_1,z_2)]=\left(\frac{1}{64}\right)[z_2^{-1}+6+z_2]D_1^2(z_1,z_2)$$

and

$$F_2\{D_1^1(z_1,z_2),Y_1[D_1^2(z_1,z_2)]\}=\left(\frac{1}{64}\right)[z_1^{-1}+6+z_1]\cdot\{D_1^1(z_1,z_2)+G(z_1)Y_1[D_1^2(z_1,z_2)]\}$$

Similarly, for $n=3$, we obtain the following lowpass output,

$$X_1(z_1,z_2)=X_0(z_1,z_2)+F_1\{D_1^2(z_1,z_2)\}+F_2\{D_1^1(z_1,z_2),Y_1[D_1^2(z_1,z_2)]\} \quad (5.2\text{-}37)$$

where

$$F_1\{D_1^2(z_1,z_2)\}=Y_1[D_1^2(z_1,z_2)]=\left(\frac{1}{256}\right)[30+8(z_2^2+z_2^{-2})+(z_2+z_2^{-1})]\cdot D_1^2(z_1,z_2)$$

and

$$F_2\{D_1^1(z_1,z_2),Y_1[D_1^2(z_1,z_2)]\}$$
$$=\left(\frac{1}{256}\right)[30+8(z_1^2+z_1^{-2})+(z_1+z_1^{-1})]\cdot\{D_1^1(z_1,z_2)+G(z_1)Y_1[D_1^2(z_1,z_2)]\}$$

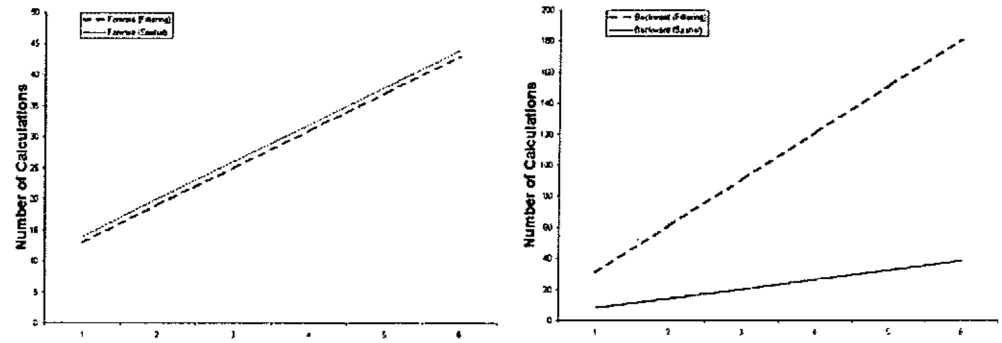
The computational costs of the wavelet kernels for $n=1$, 2 and 3 are shown in Table 5.2. It can be seen that the saving in computation of the inverse transform is significant. For $n=2$, the number of additions is reduced from 39 to 9, and the number of multiplications is reduced from 23 to 5. This corresponds to a saving of 76% and 70% of the costs of addition and multiplication respectively. For $n=3$, then number of additions is reduced from 58 to 13, and the number of multiplications is reduced from 33 to 7. This corresponds to a saving of 77% and 78% of the costs of addition and multiplication respectively. Figure 5.5 shows a comparison of costs of the forward and the inverse transforms using the filtering and the spatial approaches. The new approach is about five times faster than the conventional filtering approach of the inverse transforms. When the order of the wavelet kernel increases, the saving is also increased.

Spline Wavelet	n=1				n=2				n=3			
k	G(k)	H(k)	K(k)	L(k)	G(k)	H(k)	K(k)	L(k)	G(k)	H(k)	K(k)	L(k)
-6												1/8192
-5											-1/16384	12/8192
-4								1/512			-5/16384	66/8192
-3							-1/1024	8/512		1/64	-93/16384	220/8192
-2				1/32		1/16	-10/1024	28/512		6/64	-392/16384	495/8192
-1	4	1/4	-1/64	4/32	4	4/16	-49/1024	56/512	4	15/64	-1186/16384	792/8192
0	-8	2/4	-6/64	22/32	-8	6/16	-152/1024	326/512	-8	20/64	-2952/16384	5020/8192
1	4	1/4	-1/64	4/32	4	4/16	-49/1024	56/512	4	15/64	-1186/16384	792/8192
2				1/32		1/16	-10/1024	28/512		6/64	-392/16384	495/8192
3							-1/1024	8/512		1/64	-93/16384	220/8192
4								1/512			-5/16384	66/8192
5											-1/16384	12/8192
6												1/8192

Table 5.1, Filter coefficients of the LOG-based wavelets kernel with n=1,2 and 3.

	Filtering Approach		**Spatial Approach**	
n=1	*Forward*	*Inverse*	*Forward*	*Inverse*
Additions	8	18	9	5
Multiplications	7	13	7	3
n=2	Forward	*Inverse*	*Forward*	*Inverse*
Additions	12	38	13	9
Multiplications	9	23	9	5
n=3	*Forward*	*Inverse*	*Forward*	*Inverse*
Additions	16	58	17	13
Multiplications	11	33	11	7

Table 5.2, Computational complexity for n=1, 2 and 3 of LOG-based wavelets for the two-dimensional wavelet transform.



(a) **(b)**

Figure 5.5, Plots of **(a)** the forward transform **(b)** the backward transform for different order of the LOG-based wavelet kernel for a two-dimensional wavelet transform. Dotted lines indicate filtering approach and solid lines indicate spatial implementation.

5.2.5. Boundary artifacts

Comparing the structure of the proposed spatial implementation and the filtering approach, the proposed spatial implementation handles the boundary in a better way than the filtering approach. Under the filtering approach, there are two common ways to deal with the boundary extension problem: the image is extended before filtering or the boundary pixels are corrected after the inverse transform. The former would increase the computational time especially for a large image, while the latter involves the design of a set of filter dependent and often non-trivial boundary correction rules. However using the spatial implementation, any arbitrary boundary extension scheme can be used. The prediction terms, $F_1\left\{D_1^1(z_1,z_2)\right\}$ and $F_2\left\{D_1^2(z_1,z_2),Y_1\left[D_1^2(z_1,z_2)\right]\right\}$, remain the same in both forward and inverse transforms as shown in Figure 5.3 and Figure 5.4 respectively. Thus there is no need for any boundary correction after reconstruction; the image can always be reconstructed perfectly.

5.3. Texture-based feature extraction and retrieval analysis using the LOG-based wavelet kernel

Recent research on texture-based feature retrieval using the LOG-based wavelet kernel has shown a superior performance[9] compared to other well-known retrieval algorithm such as the Gabor filtering[24] and various sub-sampling wavelet schemes [70-73]. From the retrieval experiments[28] on the entire Brodatz texture database[13], it is shown that the LOG-based wavelet kernel can achieve the highest retrieval rate while being low in computational complexity. In order to make this thesis self-explanatory, we briefly review the structure of the texture retrieval model. Full details can be found in [9].

5.3.1. Texture extraction retrieval system

A texture-based feature extraction and retrieval system basically includes three modules: texture extraction, feature representation and similarity measurement. In texture extraction, a model is used to describe the characteristics of the texture images. Feature representation is concerned with the extraction of the important features from the model for texture description. Similarity measurement, which is based on the features found from the feature representation, is then used for texture retrieval.

Using the LOG-based wavelet kernel, a texture pattern is decomposed in a multiple resolution fashion for further analysis. From an arbitrary decomposition level *j*, we extract one lowpass signal $X_{j+1}(z_1, z_2)$ and two bandpass signals, $D^1_{j+1}(z_1, z_2)$ and $D^1_{j+1}(z_1, z_2)$, as shown in eqn.5.1-1, eqn.5.1-2 and eqn.5.1-3 respectively. In addition, for a good analysis of the orientation information in the texture pattern, the third direction of bandpass signal $D^3_{j+1}(z_1, z_2)$ in eqn.5.1-4 is included. Examining the retrieval accuracy in different levels of decomposition, the fourth level decomposition is found to achieve the highest retrieval rate[28] on the entire Brodatz texture database.

5.3.2. Feature representation

A good feature representation is needed to increase the retrieval accuracy. In feature representation, the purpose is to use a low dimensional feature vector to describe important details about the texture pattern. The statistical representation is widely used in different texture feature representation due to its simplify. After one level of decomposition, we obtained a lowpass signal $X_{j+1}(z_1, z_2)$ and three bandpass signals $D^b_{j+1}(z_1, z_2)$ for *b=1,2 or 3* as explained in eqn.5.1-2, eqn.5.1-3 and eqn.5.1-11. For the lowpass signal, we calculate the mean and the standard deviation. However, for all bandpass signals, we calculate the mean μ and the standard deviation σ at positive and negative sides separately. Mathematically, at $j = 2^i$, the mean and the standard deviation are calculated respectively as follows,

$$\mu\left(D_j^b(z_1,z_2)\right)\begin{cases}\mu^+\left(D_j^b(z_1,z_2)\right)=\dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1}D_j^b(z_1,z_2)}{L_1}\\ \mu^-\left(D_j^b(z_1,z_2)\right)=\dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1}D_j^b(z_1,z_2)}{L_2}\end{cases} \quad where \quad D_j^b(z_1,z_2)\begin{cases}>0\\ \\ <0\end{cases} \qquad (5.3\text{-}1)$$

$$\sigma\left(D_j^b(z_1,z_2)\right)\begin{cases}\sigma^+\left(D_j^b(z_1,z_2)\right)=\sqrt{\dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1}\left[\mu^+\left(D_j^b(z_1,z_2)\right)-\left(D_j^b(z_1,z_2)\right)\right]^2}{L_1}}\\ \sigma^-\left(D_j^b(z_1,z_2)\right)=\sqrt{\dfrac{\sum\limits_{n=0}^{N-1}\sum\limits_{m=0}^{M-1}\left[\mu^-\left(D_j^b(z_1,z_2)\right)-\left(D_j^b(z_1,z_2)\right)\right]^2}{L_2}}\end{cases} \quad where \quad D_j^b(z_1,z_2)\begin{cases}>0\\ \\ <0\end{cases} \qquad (5.3\text{-}2)$$

where L_1 and L_2 are the number of samples greater than and less than zero in the bandpass signal respectively, N and M denote the number of pixels in the horizontal and the vertical directions of image respectively.

For each level of decomposition, we obtain three bandpass and one lowpass filter responses. For each bandpass, we calculate the mean and the standard deviation values using eqn.5.3-1 and eqn.5.3-2 respectively. As for the lowpass filter, we simply take the mean and standard deviation values because the value is always positive and approaches to the mean value of the image. Therefore, for each level of decomposition, we obtain 7 pairs of mean and variance values. For four decomposition levels, the feature vector contains 28 pairs of mean and standard deviation, which gives a total of 56 features, i.e.,

$$f = \left(u_{00}\sigma_{00}u_{01}^+\sigma_{01}^+u_{01}^-\sigma_{01}^-u_{02}^+\sigma_{02}^+u_{02}^-\sigma_{02}^-u_{03}^+\sigma_{03}^+u_{03}^-\sigma_{03}^-u_{10}\sigma_{10}\ldots\ldots.u_{33}^+\sigma_{33}^+u_{33}^-\sigma_{33}^-\right) \qquad (5.3\text{-}3)$$

5.3.3. Similarity measurement

The objective of the similarity measurement is to compare the difference between features in two patterns and find out how similar the two patterns is. In our retrieval model, we used a distance measure for similarity measurement. This allows a high dimensional vector to be combined to give a scalar, which is used in database image query.

Let f^a and f^b be the M-dimensional feature vectors of texture patterns a and b respectively. The distance between a and b is defined as,

$$dis(a,b) = \sum_{m=1}^{M} \left| \frac{f_m^a - f_m^b}{\alpha(f_m)} \right| \qquad (5.3\text{-}4)$$

f_m^x denotes the m-th element in the feature vector x and $\alpha(f_m)$ is the standard derivation of the m-th element. The standard derivation $\alpha(f_m)$ is used to normalize the individual feature components over all query images. A small distance value $dis(a,b)$ indicates that the two patterns a and b are similar.

5.4. Experiment

In this section, we compare the performance of the spatial implementation and the filtering approach using the LOG-based wavelet kernel. This includes the computational complexity and the boundary artifact error from the image reconstruction. A comparison of the texture retrieval results on the entire Brodatz texture database between the LOG-based wavelet kernel using the new approach with other texture analysing methods is also presented.

5.4.1. Comparative study on the computational complexity

In order to confirm the theoretical analysis of both the filtering and the spatial approaches for the LOG-based wavelet kernel, some experiments on the computational complexity have been carried out. Programs are written in Borland C++ Builder program and run on an AMD Athlon 1.2GHz PC. The experiment is to calculate the average time per pixel in the forward and the inverse transforms. The size of image used in the experiment is 128x128 pixels. For each of the forward and the inverse transforms, we repeat the calculation one thousand times for the same image pattern and measure the average time per pixel,

$$T_{avg} = \frac{1}{1000}\sum_{k=1}^{1000}\frac{Time_k}{128x128} \qquad (5.4\text{-}1)$$

where T_{avg} and $Time_k$ denote the average time of each pixel and the time at the k^{th} wavelet transform respectively. The experiments were carried out for the order, $n=1$, 2 and 3 in eqn.5.1-2 and up to four decomposition levels. Table 5.3 shows the average computational times for different decomposition levels in the forward and the inverse transforms using the filtering and the spatial approaches. For $n=1$ in a single level of decomposition, the inverse transform using the spatial approach can save approximately 70% of computational time as compared to that for filtering approach. For $n=2$, the inverse transform using the spatial approach can save approximately 73% of computational time as compared that for filtering approach. For $n=3$, the inverse transform using the spatial approach can save approximately 78% of computational time as compared to that of filtering approach. These results are plotted in Figures 5.6, 5.7 and 5.8 for n is equal to 1, 2 and 3 respectively. It can be clearly seen that for the multi-level analysis with different orders n, the decomposition times in the inverse and the forward transforms are increased linearly as the decomposition level increases. The experimental results agree with the theoretic analysis of computational complexity in Theorem 5.1 and Theorem 5.3 for the filtering approach and the spatial approach. Although the computational complexity of the forward transform in the spatial approach is slightly higher than the filtering approach, it becomes insignificant for large *n* as shown in Figures 5.6-5.8. On the other hand, the computational time is greatly increased using the spatial approach rather than the filtering approach.

Unit = 10^{-6} Sec	**Filtering Approach**		**Spatial Approach**	
n=1	*Forward*	*Inverse*	*Forward*	*Inverse*
Level = 1	2.96592	5.22254	3.03821	1.52534
Level = 2	6.04775	10.7411	6.385	3.14142
Level = 3	9.4131	16.423	9.75127	5.01318
Level = 4	12.9788	23.0723	13.2775	6.49185
n=2	Forward	*Inverse*	*Forward*	*Inverse*
Level = 1	3.3117	9.88854	3.9606	2.76335
Level = 2	7.4275	20.4204	7.76258	5.3252
Level = 3	10.8451	29.4325	11.6847	7.9871
Level = 4	14.9327	40.8981	16.607	10.829
n=3	*Forward*	*Inverse*	*Forward*	*Inverse*
Level = 1	4.60306	14.1872	4.81069	2.98347
Level = 2	9.99194	28.0534	9.73156	5.97725
Level = 3	15.2403	42.5447	15.3373	8.92351
Level = 4	20.3415	56.799	20.7573	12.3251

Table 5.3, Average computation times for n=1,2 and 3 of LOG-based wavelets. All the experiment results were obtained from using an AMD Athlon 1.2GHz processor.

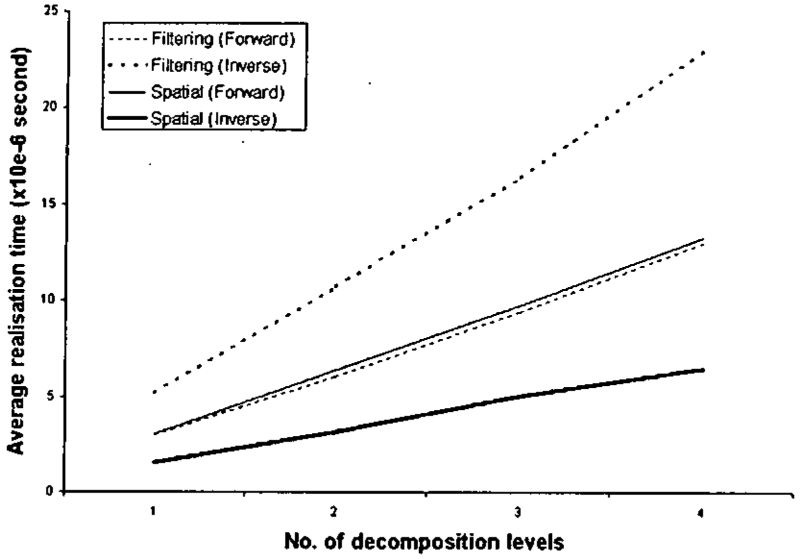


Figure 5.6, Realisation times for an image of size 128x128 pixels with different decomposition levels in the forward and the inverse transform using the LOG-based wavelet kernel with order n=1.

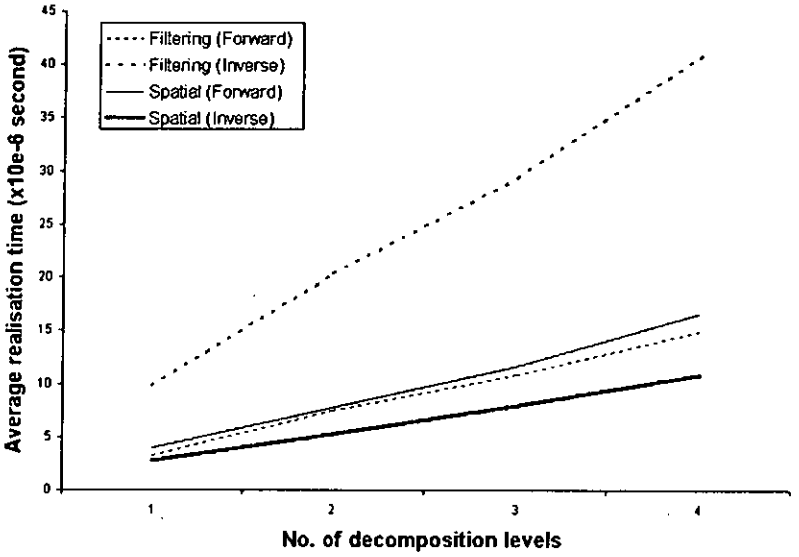


Figure 5.7, Realisation times for an image of size 128x128 pixels with different decomposition levels in the forward and the inverse transform using the LOG-based wavelet kernel with order n=2.

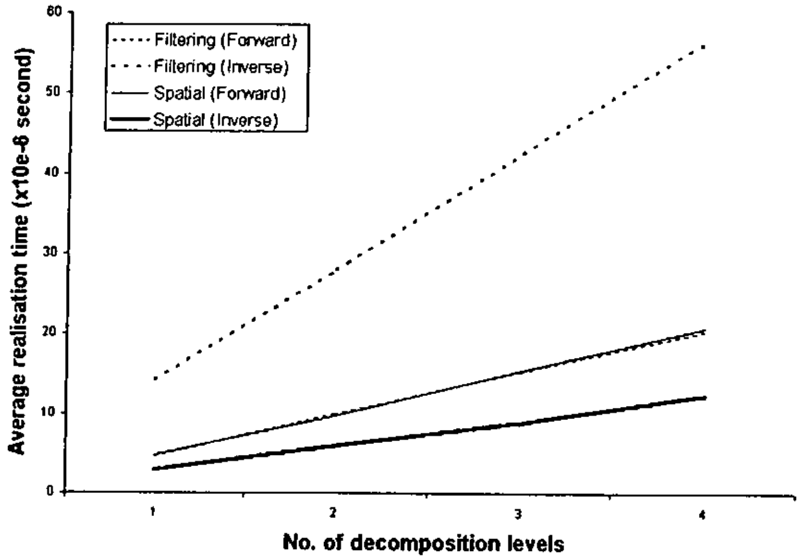


Figure 5.8, Realisation times for an image of size 128x128 pixels with different decomposition levels in the forward and the inverse transform using the LOG-based wavelet kernel with order n=3.

5.4.2. Boundary artifact on the reconstructed image

Experiments were carried out to investigate the boundary artifact error in different orders using the filtering approach and the spatial implementation. Different kinds of boundary extension methods, such as symmetric extension, periodic extension and zero padding, are compared. Figure 5.9 shows an example of the reconstructed image and the boundary artifact of the 128x128 image using the filtering approach. Table 5.4 lists the mean square errors in the reconstructed image for different level of decompositions. The MSE in the reconstructed image using the spatial approach is approximately zero on all kind of extension methods for all orders. The small errors are due to the floating-point implementation in the program. Note that all the filter coefficients for LOG-based wavelet kernel can be expressed in terms of an integer divided by 2^a, where a is an integer. Thus, one can avoid this floating-point error simply by multiplying each coefficient with 2^a to achieve prefect reconstruction. However, using the filtering approach, the boundary extension method determines the severity of the boundary artifact. For example, the zero padding causes relatively large MSE errors as compared to other two extension methods. Extra processings are required to remove this type of artifacts. We can thus conclude that the spatial approach can completely eliminate the boundary artifacts due to the prediction structure in the implementation. It is consistent with the analysis of the theoretic studies discussed in Section 5.3.5.

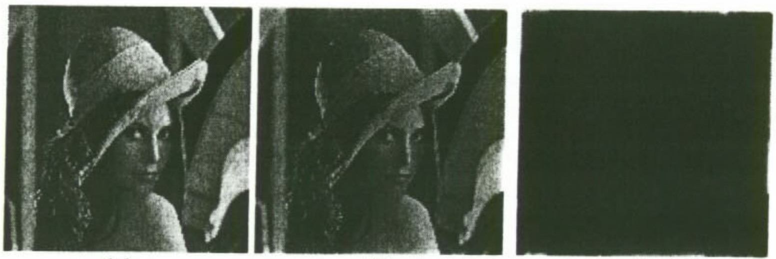

(a) **(b)** **(c)**

Figure 5.9, Example of the residual error using the filtering approach. **(a)** Original image; **(b)** reconstructed image; and **(c)** the difference between the original and the reconstructed images.

		MSE of the reconstructed image at different decomposition levels						
Boundary Extension	***n*** **Order**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Symmetric (*Filtering*)	n=1	0	0	0	2.88E-06	4.74E-06	5.9E-06	6.4E-06
	n=2	357.1433	535.5415	617.8259	656.7239	674.5634	685.7483	686.0295
	n=3	632.0102	673.5237	668.0302	663.2393	656.4475	661.1504	689.7277
Symmetric (*Spatial*)	n=1	0	0	0	0	1.51E-05	4.12E-05	4.87E-05
	n=2	0	0	4.62E-05	6.66E-05	7.94E-05	8.97E-05	9E-05
	n=3	0	5.45E-05	9.1E-05	0.000102	0.000112	0.000113	0.000113
Periodic (*Filtering*)	n=1	0	0	0	2.66E-06	4.92E-06	4.45E-06	4.5E-06
	n=2	354.4464	533.7122	616.3153	655.3734	673.4763	685.0622	685.3458
	n=3	627.4098	669.2703	663.6874	658.7151	651.8494	656.1674	684.4078
Periodic (*Spatial*)	n=1	0	0	0	0	5.39E-06	4.12E-05	4.15E-05
	n=2	0	0	4.56E-05	6.51E-05	7.62E-05	8.29E-05	9.76E-05
	n=3	0	5.46E-05	9.17E-05	0.000103	0.000117	0.000117	0.000149
Zero Padding (*Filtering*)	n=1	16.84547	29.47319	40.24727	49.10952	56.68426	62.49113	72.40338
	n=2	408.3870	618.464	728.7636	794.535	839.9896	873.4829	951.5075
	n=3	695.1384	785.7806	816.5108	841.3199	869.1735	904.3027	983.8851
Zero Padding (*Spatial*)	n=1	0	0	0	0	4.86E-06	2.87E-05	3.08E-05
	n=2	0	0	4.28E-05	5.78E-05	6.7E-05	7.05E-05	7.06E-05
	n=3	0	5.27E-05	8.79E-05	9.64E-05	0.000101	0.000102	0.000102

Table 5.4, A comparison of MSE of the reconstructed image using the filtering approach and the spatial implementation. The boundary extension methods include Symmetric, Periodic and Zero-padding and the decomposition levels are from 1 to 7.

5.4.3. Texture retrieval on the entire Brodatz texture database

A comparative study is carried out for the LOG-based wavelet kernel and other methods including the Gabor filtering[24], the sub-sampling wavelets[70-73] and the Canny-based wavelet kernel[9]. Figure 5.10 shows the retrieval results. The LOG-based wavelet kernel using the new approach can achieve a 79.22% retrieval rate on the entire Brodatz texture database[13], which is the highest retrieval rate among different feature extraction algorithms. It has around 4-7% higher retrieval rate as compared to other methods. Furthermore, due to the separable nature of the wavelet transform, the computation time can be greatly reduced as compared to Gabor filtering.

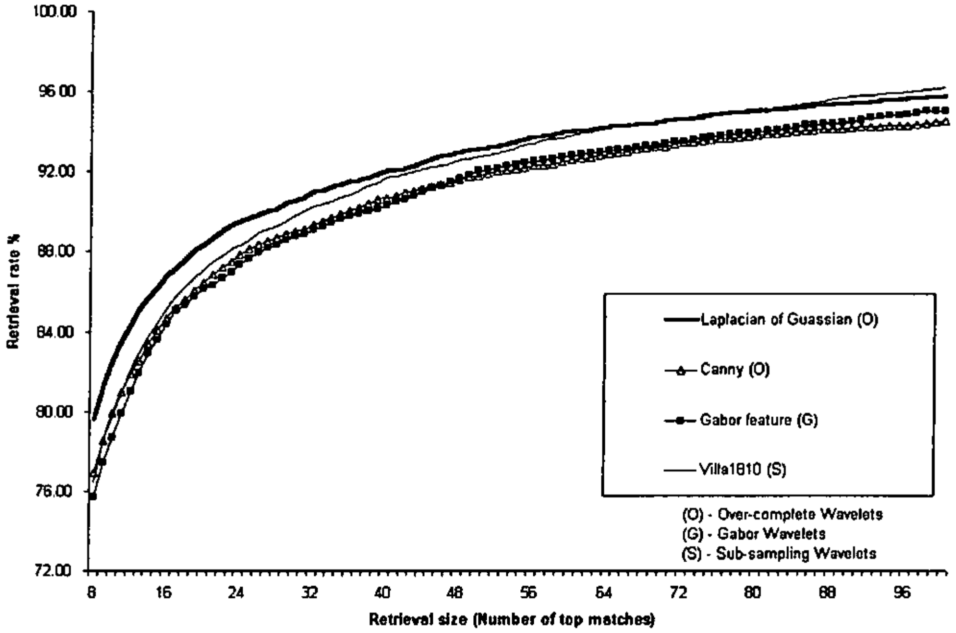


Figure 5.10, Comparison of the LOG-based wavelet kernel using the spatial approach with other texture feature analysing methods on the Brodatz texture database.

5.5. Summary

A new spatial implementation of the LOG-based wavelet kernel has been proposed. The new approach can greatly reduce the computational complexity. In particular, the inverse wavelet transform using the new approach can achieve about five times less computational cost than the inverse transform using the filtering approach. In addition, the new approach has a simple structure that the same prediction terms are used in both forward and inverse transforms. This helps eliminate the boundary artifact completely; no matter what kind of boundary extension methods are used. The approach is attractive that it is always not necessary to used boundary correction after reconstruction. Images can always be reconstructed perfectly. Experiments results show that the new approach can greatly decrease the computational cost of the inverse transform and eliminate the boundary artifact of the reconstructed image for different boundary extension methods. For the applications such as texture feature retrieval, it achieves the highest retrieval rate as compared to other texture analysing methods.

CHAPTER 6

6. Conclusion and future directions

6.1. Conclusion

This dissertation describes a new texture-based image feature retrieval technique which uses the over-complete wavelet scheme for texture feature analysis. The advantage of using the wavelet scheme is its ability to achieve low computational complexity and translation invariance for the multi-resolution wavelet transform. Compared to other texture-based feature retrieval methods such as Gabor wavelets and sub-sampling wavelet schemes, the over-complete wavelet schemes can achieve the highest retrieval rate on the entire Brodatz Texture database test. Also, it is robust even under severe noise levels. Instead of using the Canny-based wavelet kernel, we have derived a new kernel that is based on the Laplacian of Gaussian(LOG)-based wavelet kernel. It has been proven that the LOG-based wavelet kernel is more suitable for the characterization of the texture feature. It is especially suitable for analysing line information of texture pattern. We have proposed a new texture presentation representation. The feature is calculated from the variance separately on the positive and the negative filter responses. It can be applied to both the over-complete wavelet schemes and the sub-sampling wavelet schemes. The retrieval accuracy is significantly increased as compared to other feature representations.

We have further investigated the fast algorithm for the LOG-based over-complete wavelet transform in this dissertation. The kernel is a completed wavelet transform and thus it allows perfect reconstruction of the signal. However, implementation using conventional filtering approach is inefficient and causes artifacts. By deriving some general expressions for the computational cost using the conventional filtering implementation, it is found that the inverse transform is significantly more costly in computation than the forward transform. In order to reduce the computation complexity, a spatial implementation is proposed. Results of our theoretical and experimental studies show that the new approach can greatly decrease the cost of transforms. Furthermore, affect boundary artifacts can be eliminated by multi-level wavelet transforms using the new approach. It is accurate for uses in different image processing applications.

6.2. Future directions

6.2.1. Improve the distance measurement using appropriate weighting

In chapter 1, we have used a simple normalize distance measurement[24] to join two different vector spaces (mean and standard deviation) for calculating the similarity of images. Actually, to combine different vector spaces may not simple. As shown in section 3.3.6, we have done the experiment on the reduced feature representation which indicated a reduction in mean vector causes a drop in retrieval rate. It gives some hint that if appropriating weighting is added to the reduced feature, it may have decrease the retrieval results. Therefore, for more appropriate representation of the distance measure, we can rewrite the eqn.1.2-1 as follow,

$$dis(a,b)=\sum_{m=1}^{56}\left(w_m\cdot\left|\frac{f_m^a-f_m^b}{\alpha(f_m)}\right|\right)$$

where w is the weighting and m is index of the feature representation in eqn3.2-3. Appropriate weighting factors are more meaningful to different vector spaces and can increase the retrieval accuracy.

However, to estimate the weighting factor is non-trivial. It involves some prior knowledge, such as classification, relevant feedback or neural network, to estimate these weighting factors. This approach is related to the supervising texture retrieval [27, 64, 70]. And results from the literature that a significant improvement in retrieval can be achieved. Hence, our research work is only at the initial stage. Further researches are necessary to advantage the similarity measurement method.

6.2.2. Improving the directional pattern feature extraction using the over-complete wavelet transform

The over-complete wavelet scheme is a separable wavelet transform. It decomposes an image pattern into three directions, including horizontal, vertical and diagonal(45 degrees) components. However, the main deficiency of the transform for texture analysis is its limitation on the degree of direction information extracted from image pattern. Compared to non-separable wavelet scheme, such as Gabor wavelet, which is selective on the decomposition direction and thus it extracts better directional information for the texture pattern. As observed from the texture images in Brodatz Album, some image classes mainly consist of diagonal lines, such as D18, D38 and D52, in the Brodatz Album. Let us Compare the retrieval results of texture images between the Gabor wavelet and LOG-based over-complete wavelet as shown in Table 3.4 in Chapter 3. The retrieval performance of Gabor wavelet is around 18% which gives a better retrieval performance as compared to the LOG-based over-complete wavelet transform. It is the weakest point of the over-complete wavelet for the retrieval diagonal pattern. In order to improve the retrieval performance of using the over-complete wavelet scheme for texture images which consists mainly of diagonal lines, it should improve the extraction method of directional components. Wavelet packet[26] is a good possibility to improve the spatial-frequency relationship. It decomposes "smaller" frequency spectrum of feature and thus allows extraction of better directional information. However, the main deficiency of using wavelet packet is that the length of features should be increased. As a result, it directly increases the complexity on comparison. However the complexity problem can be resolved by using classification methods[26] which reduces the number of features used for texture retrieval.

6.2.3. Content-based feature image/video retrieval

The main objective of this research is to develop a texture-based feature retrieval method which is suitable for image/video applications for content-based feature searching and indexing purposes. In Chapter 3, we proved that the LOG-based wavelet transform can achieve superior retrieval performance on Brodatz texture database. The short analysis time using the over-complete wavelet-based implies a reduction in the waiting time on feature analysis. This method should be suitable for content-based image/video indexing and retrieval system.

Most of the content-based image/video retrieval systems support local region searching[2, 6, 32]. In order to apply the LOG-based over complete wavelet for real image/video data searching, further investigations are needed to resolve problems such as the problems of region-based searching and scale invariant. Further investigation should also be given to storage problem. In order to reduce feature contents stored in database and to achieve faster retrieval time, we should use fewer dimensions in texture representation. Many classification and training techniques[26, 78-80] can help to solve this problem.

6.2.4. Texture-based Image segmentation using the over-complete wavelet-based transform

Image segmentation is an important research area for image processing. Basically, image segmentation partitions an image into smaller homogeneous regions. Many texture-based image segmentation techniques were proposed, which were based on the similarity of local texture characteristic to distinguish regions. Gabor filters were commonly used for image segmentation[25, 81-83]. It permits extraction of multi-frequencies, orientations and channels of texture characteristics in image patterns and makes use of clustering similar properties of texture region. Although it has very good performance on segmentation, these segmentation methods always require a large amount of computation time. We can consider the LOG-based wavelet scheme since it can extract effectively multi-frequencies, orientations and channels of texture characteristics. Furthermore, it takes less computation time as compared to Gabor filters. It is suitable to apply it for image segmentation.

6.2.5. Shape-based feature image feature extraction and retrieval

Shape is recognized as another important content-based image retrieval feature. Over the past few decades, many shape analysis methods were proposed which aim to characterize shape features for similarity matching. An important class of shape analysis algorithms is based on the representation of the outer boundaries of objects. Fourier descriptor is a well-known shape analysing technique[84-86], which describes the global contour in frequency spectrum. For shape retrieval using the Fourier descriptor, it compares the feature extracted from frequency spectrum. The idea was extended to using the wavelet approach to describe the shape features to analysis of spatial-frequency. Examples of using dyadic sub-sampling wavelet approach to characterize the multi-resolution shape feature can be found in [87]. However, similar to the texture analysis discussed in Chapter 2, the sub-sampling wavelet scheme suffers from the translation variant problem. However, the over-complete wavelet transform gives no such problem. Some early results of using the Canny-based over-complete wavelet transform for the characterization of the corner detection of shape contour for multi-resolution have been reported[88]. However, as discussed in Chapter 3, the LOG-based operator seems to have a better analysis in "thickness" of line during feature extraction rather than Canny operator. Therefore, it is a possible method that can achieve better retrieval performance using shape features.

6.2.6. Other image processing applications

Wavelet is a powerful tool for image processing and it has been widely used on various image processing researches. This dissertation has introduced a new LOG-based over-complete wavelet which achieves low transform time and translation invariance during multi-levels wavelet transforms. Also, since the LOG-based over complete wavelet is a complete wavelet transform, it gives perfect reconstruction of the signal. As we have reported, the spatial implementation of the wavelet can greatly reduce the computation time and can eliminate the boundary artifacts. This wavelet approach can be applied to larger image processing areas such as image enhancement, noise elimination, image compression, texture synthesis etc. We believe that this wavelet approach will be widely used in other image processing researches in the near future.

References

[1] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. “"The QBIC project: Querying images by content using color, texture, and shape.” SPIE Proceedings on Storage and Retrieval for Image and Video Databases, Vol.1908, 1993.

[2] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu. “Virage image search engine: an open framework for image management”, SPIE Proceedings on Storage & Retrieval for Image and Video Databases IV, Vol.2670, pp.76 - 87, 1996.

[3] Pentland, R. W. Picard, and S. Sclaroff. “Photobook: Tools for content-based manipulation of image databases”, SPIE Proceedings on Storage and Retrieval Image and Video Databases II, Feb 1994.

[4] V. E. Ogle and M. Stonebraker. “Chabot: Retrieval from a relational database of images”, IEEE Computer, Vol.9, No.28, pp.40-48, 1995.

[5] R. K. Srihari, “Automatic indexing and content-based retrieval of captioned images”, IEEE Computer, Vol.9, No.28, pp.49 - 56, 1995.

[6] J. R. Smith and S.-F. Chang. “VisualSEEk: a fully automated content-based image query system.” Proceeding of ACM International Conference of Multimedia, pp.87-98, Boston, MA, 1996.

[7] S. W. Smoliar and H. Zhang “Content-based video indexing and retrieval”, IEEE Multimedia Magazine, 1994.

[8] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner,D. Lee, D. Petkovic, D. Steele, and P. Yanker. “Query by image and video content: The QBIC system" IEEE Computer”, Vol.28, pp.23 - 32, Sep 1995.

[9] Mallat. S. and Zhong. S., “Characterization of signals from multiscale edges”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, pp.710–732, Jul 1992.

[10] R. M. Haralick, “Statistical Image Texture Analysis”, HPRIP, pp.245-279, 1986.

[11] F. M. Vilnrotter, R. Nevatia, and K. E. Price, "Structural analysis of natural textures", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.8, pp.76-89, 1986.

[12] R. M. Haralick, "Statistical and structural approaches to texture", Proceeding of IEEE, Vol.67, pp.786-804, No.5, 1979.

[13] P. Brodatz, "Textures: A photographic album for artists & designers", New York, Dover, 1996.

[14] D. H. Ballard and C. M. Brown, "Computer Vision", Prentice-Hall, Inc, Englewood Cliffs, NJ, 1982.

[15] H. Tamura, S. Mori, and Y. Yamawaki, "Textural Features Corresponding to Visual Perception", IEEE Transactions on Systems Man, and Cybernetics, Vol.8, pp. 460-473, 1978.

[16] Sklansky, J., "Image Segmentation and Feature Extraction", IEEE Transactions on Systems, Man, and Cybernetics, SMC-8, pp. 237-247, 1978.

[17] Haralick, R.M., "Statistical and Structural Approaches to Texture", Proceedings of the IEEE, Vol.67, pp. 786-804, 1979.

[18] Richards, W. and A. Polit, "Texture matching", Kybernetic, Vol. 16, pp. 155-162, 1974.

[19] Zucker, S. W. and K. Kant, "Multiple-level Representations for Texture Discrimination", IEEE Proceedings on Pattern Recognition and Image Processing, pp. 609-614, 1981.

[20] Hawkins, J. K., "Textural Properties for Pattern Recognition", Picture Processing and Psychopictorics, Academic Press, 1969.

[21] Coggins, J. M., "A Framework for Texture Analysis Based on Spatial Filtering", Ph.D. Thesis, Computer Science Department, Michigan State University, East Lansing, Michigan, 1982.

[22] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, No. 7, pp.722-733, July 1996.

[23] W. Y. Ma and B. S. Manjunath. "Texture-based pattern retrieval from image databases", Multimedia Tools and Applications, Vol.2, No.1, pp.35 - 51, 1996.

[24] Manjunath. B. S. and Ma. W. Y., "Texture features for browsing and retrieval of image data", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, pp.837 –842, Aug 1996.

[25] C. Bovick, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.1, No.12, 1990.

[26] Laine and J.Fan, "Texture Classification by Wavelet Packet Signatures", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.11, No.15, 1993.

[27] Mojsilovic. A., Popovic. M. V. and Rackov. D. M., "On the selection of an optimal wavelet basis for texture characterization", IEEE Transactions on Image Processing, Vol. 9, pp. 2043 –2050, Dec 2000.

[28] R. W. Picard, T. Kabir, and F. Liu, "Real-time recognition with the entire Brodatz texture database", Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, pp.638-639, Jun 1993, New York.

[29] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, andC. Faloutsos. "The QBIC project: Querying images by content using color, texture, and shape", SPIE on Storage and Retrieval for Image and Video Databases, pp.173-185, 1993.

[30] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu. "Virage image search engine: an open framework for image management", SPIE on Storage & Retrieval for Image and VideoDatabases IV, Vol.2670, pp.76-87. 1996.

[31] Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases", SPIE on Storage and Retrieval of Image and Video Databases II, pp.73-75, 1994.

[32] W. Y. Ma and B. S. Manjunath, "NETRA: A Toolbox for navigating large image databases", IEEE Proceedings on Image Processing, 1997.

[33] W. Y. Ma and B. S. Manjunath, "A texture thesaurus for browsing large aerial photographs", Journal of the American Society for Information Science, Vol.49, No.7, pp.633-48, 1998.

[34] S. Manjunath, "Image Browsing in the Alexandria Digital Library Project", D-Lib Magazine, August 1995.

[35] Julesz and J. R. Bergen. Textons, "The fundamental elements in preattentive vision and perception of textures", The Bell System Technical Journal, Vol: 6, No.62, pp.1619-1645, July-August 1983.

[36] M. Galloway. "Texture analysis using gray level run lengths", Computer Graphics Image Processing, 4, pp.172-179, 1975.

[37] R. M. Haralick, K. Shanmugam, and I. Dinstein. "Textural features for image classification", IEEE Transaction on Systems, Man and Cybernetics, Vol.6. pp.610-621, 1973.

[38] J. W. Woods, "Two-dimensional discrete Markovian fields", IEEE Transaction on Information Theory, vol.2, pp.232-240, 1972.

[39] T. Matsuyama, S. Miura, and M. Nagao, "Structural analysis of natural textures by Fourier transformation", Computer Vision, Graph and Image Processing, Vol. 24, pp.347-362, 1983.

[40] F. M. Vilnrotter, R. Nevatia, and K. E. Price, "Structural analysis of natural textures", IEEE Transaction of Pattern Analysis and Machine Intelligence. Vol.9, No. 1, pp.76-89, 1986.

[41] P. Volet and M. Kunt, "Synthesis of natural structured textures", Signal Processing III: Theories and Applications, pp.913-916. North-Holland, 1986.

[42] R. Rao and G. L. Lohse, "Towards a texture Naming System: Identifying Relevant Dimensions of Texture", IEEE Processing on Visualization, pp.220-227, 1993.

[43] J. M. Francos, "Orthogonal Decompositions of 2D Random Fields and Their Applications in 2D Spectral Estimation", Signal Processing and Its Application, pp.207-227, 1993.

[44] J. M. Francos, A. Z. Meiri and B. Porat, "A Unified Texture Model Based on a 2D Wold Like Decomposition", IEEE Transactions on Signal Processing, pp.2665-2678, 1993.

[45] F. W. Campbell and F. G. Robson, "Application of Fourier Analysis to the Visbility of Gratings", Journal of Physiology, Vol.197, pp.551-566, 1968.

[46] P. Dewaele, P. V. Cool and A. Oosterlinck, "Texture Inspection with Self-Adaptive Convolution Filters", Proceedings of the 9th International Conference on Pattern Recognition, pp.56-60, 1988.

[47] K. Chui, "An Introduction to Wavelet", Academic Press, 1992.

[48] J. G. Daugman., "Entropy reduction and decorrelation in visual coding by oriented neural receptive fields", IEEE Transactions on Biomedical Engineering, Vol.36, Issue 1, pp.107-114 1989.

[49] R. Wilson and G. H. Granlund. "The uncertainty principle in image processing", IEEE Transaction on Transactions on Pattern Analysis and Machine Intelligence, Vol.6, 1984.

[50] Marr, "Vision", San Francisco: W.H. Freeman, 1982.

[51] Daubechies, "Ten lectures on wavelets", CBMS-NSF Series Appl. Math., SIAM, 1991

[52] W. Sweldens, "The Lifting Scheme: A Construction of Second Generation Wavelets", SIAM Journal on Mathematical Analysis, Vol. 29, No. 2, pp.343 – 379, 1998.

[53] N. F. Law and W. C. Siu, "Fast algorithm for Over-Complete Wavelet", Electronic letters, Vol.37, No.4, pp. 239-261, Feb, 2001, U.K.

[54] N. F. Law and W. C. Siu, "An Efficient Computational Scheme for the 2D Over-Complete Wavelet Transform", Submitted to IEEE on Signal Processing.

[55] W. Y. Chan, W. C. Siu and N. F. Law, "Fast Algorithm for 1D Laplacian of Gaussian-based Over-complete Wavelets", Paper in Preparation.

[56] M. E. Jernigan and F. D'Astous., "Entropy-based texture analysis in the spatial frequency domain", IEEE Transaction on Transactions on Pattern Analysis and Machine Intelligence, Vol.2, pp.237-274, 1984.

[57] Daugman. J. G., "Two Dimensional Spectral Analysis of Cortical Receptive Field Profiles", Vision Research, Vol.20, pp.847-856, 1980.

[58] Daugman. J. G., "Complete discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Compression", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.36, pp.1169-1179, 1988

[59] Turner. M. R., "Texture Discrimination by Gabor Functions", Biological Cybernetics, 55, pp.71-82, 1986.

[60] Clark. M. and A. C. Bovik, "Texture segmentation using Gabor modulation and demodulation", Pattern Recognition Letters, Vol.6, pp.261-267, 1987.

[61] M. Turceryan and A. Jain, "Texture analysis", Handbook of Pattern Recognition and Computer Vision, pages 235–276, World Scientific, Publishing Company, River Edge, NJ, 1993.

[62] A.C. Bovik, M/ Clark and W.S. Geisler, "Multichannel texture analysis using localized spatial", IEEE transons on PAMI, Vol.12, No.1 pp.55-73, 1990.

[63] Dunn, W. E. Hiffins and J. Wakeley, "Texture segmentation using 2D Gabor elemetary functions", IEEE transaction on PAMI, Vol.16, No.2, pp.130-149, 1994.

[64] T. Chang, C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform", IEEE Transaction on Image Processing, Vol.2, No.4, pp.429-441, 1993.

[65] Kundu and J.-L. Chen, "Texture classification using QMF bank-based subband decomposition", Computer Vision, Graphics and Image Processing (CVGIP): Graphical Models and Image Processing, Vol.54, No.5, 1992.

[66] P. P. Ohanian and R. C. Dubes, "Performance evaluation for four classes of textural features", Pattern Recognition, Vol.25, No.8, 1992.

[67] J. Canny. "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.8, pp.679-698, 1986.

[68] Marr, "Vision". San Francisco: W.H. Freeman, 1982.

[69] S. Mallat, "Multifrequency channel decompositions of image and wavelet models", IEEE Transactions of Acoustic Speech Signal Processing, vol.37, no.12, pp.2091-2110, Dec 1989.

[70] S. S. Wolfson and M. S. Landy, "Examining Edge- and Region-based Texture Analysis Mechanisms", Vision Research. Vol. 38, No. 3, pp.439-446, 1998.

[71] Odegard. J. E. anad Burrus. C.S., "Smooth biorthogonal wavelets for applications in image compression", IEEE on Digital Signal Processing Workshop Proceedings, pp.73 –76, 1996.

[72] Antonini. M., Barlaud. M, Mathieu. P and Daubechies. I, "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, Vol. 1, pp. 205 – 220, Apr 1992.

[73] Villasenor. J. D., Belzer. B. and Liao. J, "Wavelet filter evaluation for image compression", IEEE Transactions on Image Processing, Vol. 4, pp.1053 –1060, Aug 1995.

[74] Chakrabarti, C. and Vishwanath, M., "Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to mappings on SIMD array computers", IEEE Transactions on Signal Processing, Vol.43, Issue 3, pp.759-771, 1995.

[75] Hamdi, M. "Parallel architectures for wavelet transforms", IEEE Proceeding on Computer Architectures for Machine Perception, pp.376–384, 1993.

[76] M. Holstrom, "Parallelizing the fast wavelet transform", Parallel Computing, Vol.11, No.22, pp.1837-1848, 1995.

[77] J. Lu, "Parallelizing Mallat Algorithm for 2D Wavelet Transform", Information Processing Letters, Vol.45, No.5, pp.255-259, 1993.

[78] J. Rosiles and M. Smith, "Texture classification with a biorthogonal directional filter bank", IEEE Proceedings on Acoustics, Speech, and Signal Processing, Vol.3, pp.1549-1552, 2001. 2001.

[79] M. Unser, "Texture classification and segmentation using wavelet frames", IEEE Transactions on Image Processing, Vol. 4, Issue.11, pp.1549-1560, 1995.

[80] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform", IEEE Transactions on Image Processing, Vol.2, No.4, pp.429-441, 1993.

[81] Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation using 2-D Gabor elementary functions", IEEE Transaction on Pattern analysis and Machine Intelligence, Vol.16, No.2, 1994.

[82] K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters", Pattern Recognition, Vol.24, No.12, 1991.

[83] D. P. Mital, "Texture segmentation using Gabor filters", IEEE Proceedings on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Vol.1, pp.109-112, 2000.

[84] C.T. Zahn and R.Z. Roskies, "Fourier descriptors for plane close curves", IEEE Transaction on Computers, Vol.21, pp. 269-281, 1972.

[85] G.H. Granlund, "Fourier Preprocessing for hand print character recognition", IEEE Transaction on Computers, Vol.21, pp.195-201, 1972.

[86] O. Betrand, R. Queval, H. Maître, "Shape Interpolation by Fourier Descriptors with Application to Animation Graphics", Signal Processing, Vol.4, pp.53-58, 1981.

[87] Micheli-Tzanakou and I. Marsic, "Visual shape analysis by wavelets", IEEE Symposium on Neuroinformatics and Neurocomputers, Vol.1, pp.289-298, 1992.

[88] Cheikh, A. Quddus, and M. Gabbouj, "Multi-level shape recognition based on wavelet-transform modulus maxima", IEEE Proceedings on Image Analysis and Interpretation, The 4th IEEE Southwest Symposium. pp.8-12, 2000.

Appendices

Appendix 1. Realisation of bandpass filter in the Laplcian of Gaussian Based Wavelets

For the bandpass filter $G(z)$, we have

$$G(z)=4\left(z^{-1}-2+z\right) \tag{A-1}$$

Multiply eqn.A-1 by z^{n-1} at both sides, where *n* is an integer, gives

$$z^{n-1}\cdot G(z)=4\left(z^{n-2}-2\cdot z^{n-1}+z^{n}\right) \tag{A-2}$$

Similarly, multiply eqn.A-1 by $z^{-(n-1)}$ at both sides, gives

$$z^{-(n-1)}\cdot G(z)=4\left(z^{-(n-2)}-2\cdot z^{-(n-1)}+z^{-n}\right) \tag{A-3}$$

Adding eqn.A-2 and eqn.A-3, gives

$$z^{n-1}G(z)+z^{-(n-1)}G(z)=4\left(z^{n-2}-2\cdot z^{n-1}+z^{n}+z^{-(n-2)}-2\cdot z^{-(n-1)}+z^{-n}\right) \tag{A-4}$$

By rearranging eqn.A-4, we obtain,

$$z^{n}+z^{-n}=\left[\frac{z^{n-1}G(z)}{4}+\frac{z^{-(n-1)}G(z)}{4}-z^{n-2}+2\cdot z^{n-1}+2\cdot z^{-(n-1)}-z^{-(n-2)}\right] \tag{A-5}$$

Proof: Substitute *m=n* into eqn.A-5, we obtain the general expression as eqn.A-6. we use the mathematical inductions to prove the eqn.A-6 is true for all integer,.

$$z^{m}+z^{-m}=\frac{1}{4}\left\{\left(\sum_{k=1}^{m-1}(m-k)\left[z^{k}+z^{-k}\right]\right)\cdot G(z)+m\cdot G(z)+8\right\} \tag{A-6}$$

By substituting $n=1$, we can easily prove that eqn.A-5 are equal at both side, i.e.,

$$L.H.S \quad = z + z^{-1}$$

$$R.H.S \quad = \frac{1}{4}G(z) + \frac{1}{4}G(z) - z^{-1} + 2 + 2 - z$$

$$= z^{-1} - 2 + z + z^{-1} - 2 + z - z^{-1} + 2 + 2 - z$$

$$= \left(z^{-1} + z\right)$$

Assume that eqn.A-6 is true for $m=n$. Let us consider the case when $m=n+1$,

$$L.H.S \quad = z^{-(n+1)} + z^{n+1}$$

$$R.H.S \quad = \left(\frac{1}{4}\right)\left[\left(\sum_{k=1}^{n}(n+1-k)\left[z^{k} + z^{-k}\right]\right)\cdot G(z) + (n+1)\cdot G(z) + 8\right]$$

$$= \left\{\left(\frac{1}{4}\right)\left[\begin{array}{l}\left(z^{n} + 2z^{n-1} + \ldots + n\cdot z + n\cdot z^{-1} + \ldots + 2z^{-(n-1)} + z^{-n}\right)\left(4z - 8 + 4z^{-1}\right) \\ +(n+1)\left(4z - 8 + 4z^{-1}\right) + 8\end{array}\right]\right.$$

$$= \left\{\begin{array}{l}\left(z^{n} + 2z^{n-1} + \ldots + n\cdot z + n\cdot z^{-1} + \ldots + 2z^{-(n-1)} + z^{-n}\right)\left(z - 2 + 2z^{-1}\right) \\ +(n+1)\left(z - 2 + z^{-1}\right) + 2\end{array}\right.$$

$$= \left(z - 2 + z^{-1}\right)\left(z^{n} + 2z^{n-1} + \ldots + n\cdot z + n + 1 + n\cdot z^{-1} + \ldots + 2z^{-(n-1)} + z^{-n}\right) + 2$$

$$= \left\{\begin{array}{l}\left(z + z^{-1}\right)\left(z^{k} + 2z^{k-1} + \ldots + n\cdot z + n + 1 + n\cdot z^{-1} + \ldots + 2z^{-(k-1)} + z^{-k}\right) \\ -2\left(z^{k} + 2z^{k-1} + \ldots + n\cdot z + n + 1 + n\cdot z^{-1} + \ldots + 2z^{-(k-1)} + z^{-k}\right) + 2\end{array}\right. \qquad \text{(A-7)}$$

The first term in eqn.A-7 can be written as

$$\left(z^{-1} + z\right)\left(z^{n} + 2z^{n-1} + \ldots + n\cdot z + n + 1 + n\cdot z^{-1} + \ldots + 2z^{-(n-1)} + z^{-n}\right)$$

$$= \begin{cases} \begin{pmatrix} z^{n+1} + 2z^{n} + 3z^{n-1} + \ldots + (n)z^{2} + (n+1)z \\ +n + (n-1)z^{-1} + (n-2)z^{-2} + \ldots + 3z^{-(n-3)} + 2z^{-(n-2)} + z^{-(n-1)} \end{pmatrix} \\ + \begin{pmatrix} z^{n-1} + 2z^{n-2} + 3z^{n-3} + \ldots + (n-2)z^{2} + (n-1)z \\ +n + (n+1)z^{-1} + (n)z^{-2} + \ldots + 3z^{-(n-1)} + 2z^{-n} + z^{-(n+1)} \end{pmatrix} \end{cases}$$

$$= \begin{cases} z^{n+1} + 2z^{n} + 4z^{n-1} + \ldots + (2n)z + (2n) \\ +(2n)z^{-1} + \ldots + 4z^{-(n-1)} + 2z^{-n} + z^{-(n+1)} \end{cases} \tag{A-8}$$

The second term in eqn.A-7 can be written as

$$-2 \cdot \left(z^{n} + 2z^{n-1} + \ldots + (n+1) + \ldots + 2z^{-(n-1)} + z^{-n}\right)$$

$$= -\left(2z^{n} + 4z^{n-1} + \ldots + (2n)z + 2n + 2 + (2n)z^{-1} + \ldots + 4z^{-(n-1)} + z^{-n}\right) \tag{A-9}$$

Submitting eqn.A-8 and eqn.A-9 into eqn.A-7, it becomes,

$$R.H.S = z^{-(n+1)} + z^{n+1} = L.H.S$$

Therefore, the statement is true for *m=n+1*. By using the mathematical induction, the statement is true for all integer *n*, which completes the proof.□