**USC-SIPI REPORT #198**

# Texture Analysis and Classification with Tree-Structured Wavelet Transform

**by**

**Tianhorng Chang and C.-C. Jay Kuo**

**February 1992**

Signal and Image Processing Institute
**UNIVERSITY OF SOUTHERN CALIFORNIA**
Department of Electrical Engineering-Systems
Electrical Engineering Building
University Park/MC-2564
Los Angeles, CA 90089 U.S.A.

Submitted to IEEE Transactions on Image Processing

# Texture Analysis and Classification with Tree-Structured Wavelet Transform *

Tianhorng Chang † and C.-C. Jay Kuo †

February 21, 1992

**Abstract**

Although textures have been studied for more than thirty years, research on texture analysis is still active. The main difficulty of the problem is due to the lack of an adequate tool which characterizes different scales of textures effectively. Traditional methods based on the second-order statistics or the Gaussian Markov Random Field (GMRF) model share one common weakness. That is, they primarily focus on the coupling of pixels in a single scale. Recent developments in multiresolution analysis such as the Gabor and wavelet transforms help to overcome this difficulty. In this research, we propose a multiresolution approach based on a tree-structured wavelet transform for texture classification. The development of tree-structured wavelet transform is motivated by the observation that textures are quasi-periodic signals whose dominant frequencies are located in the middle frequency channels. With the transform, we are able to zoom into desired frequency channels and perform further decomposition. In contrast, the conventional wavelet transform only decomposes subsignals in low frequency channels. We also develop a progressive texture classification algorithm which is not only computationally attractive but also has excellent performance.

## 1 Introduction

Textures provide important characteristics for surface and object identification from aerial or satellite photographs and biomedical images. Their analysis is fundamental to many applications such as industrial monitoring of product quality, remote sensing of earth resources, and medical diagnosis with computer tomography. Numerous research work has been done on texture analysis, classification and segmentation for last three decades. Despite the effort, texture analysis is still considered an interesting but difficult problem in image processing.

*This work was supported by a National Science Foundation Research Initiation Award (ASC-9009323).

†The authors are with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, California 90089-2564. E-mail: tianhorn@sipi.usc.edu and cckuo@sipi.usc.edu.

Early research work [11], [23], [29], [30], [41] such as SGLDM [24] and correlation [6], [19] was based on the second-order statistics of textures. In the 80's, researchers developed the Gaussian Markov Random Field (GMRF) [5], [8], [12], [18], [31], [32], [47], [48] and Gibbs distribution [16], [17], [22], [34] texture models, where the gray levels between nearest neighboring pixels are characterized by a certain stochastic relationship. All the above methods share one common weakness. That is, they primarily focus on the coupling between image pixels on a single scale. Laws [35] proposed a simple scheme which used local linear transformations and energy computation to extract texture features. The simple scheme often gives reasonably good performance, and has been studied and improved by many researchers [10], [26], [27], [43], [44]. More recently, methods based on multichannel or multiresolution analysis have received a lot of attention [1], [3], [45]. They often outperform traditional methods based on the second-order statistics or the GMRF model. Laws method turns out to be a multichannel method, which explains its good performance. This suggests that the main difficulty of traditional texture analysis is due to the lack of an adequate tool which characterizes different scales of textures effectively. Recent developments in time/frequency analysis such as Gabor and wavelet transforms provide good multiresolution analytical tools and should help to overcome this difficulty.

Textures can be modeled as a quasi-periodic pattern and detected by highly concentrated spatial frequencies and orientations. Recent study on the human vision system indicates that the spatial/frequency representation [7], [9], [39], [46] which preserves both global and local information is adequate for quasi-periodic signals. This observation has motivated researchers to develop multiresolution texture models. New algorithms such as methods with the Gabor transform have been proposed, and successful results have been reported [2], [3], [15], [20], [38]. However, the use of the Gabor transform as a texture discriminant tool requires parameters such as spatial frequencies and orientations and the shape of the Gaussian envelope function. These parameters are often determined by applying the Fourier transform to textures and examining the peaks of computed power spectra. In this research, we use the wavelet transform for texture analysis and classification. One clear advantage of the wavelet transform over the Gabor transform is that spatial frequencies and orientations can be determined automatically in the transform process so that no additional procedure is required for parameter estimation.

The wavelet transform is a new spatial/frequency analytical tool which has been under intensive study during the last five years [13], [14], [25], [36], [37], [40], [42]. We propose

a multiresolution texture model based on a modified wavelet transform called the *tree-structured wavelet transform*. The traditional pyramid-type wavelet transform decomposes subsignals in the low frequency channels. However, since the most significant information of a texture often appears in the middle frequency channels, the conventional wavelet transform does not apply properly. To modify the transform, we use a certain energy function to characterize the strength of a subsignal whose components are concentrated in a frequency channel, and perform a further decomposition for the channel containing a significant amount of energy.

To achieve the texture classification task, we adopt a two-phase (i.e. learning and classification) procedure. In the learning phase, the tree-structured wavelet transform is performed on various samples of known textures. Although different samples of the same texture may result in slightly different tree-structured wavelet decompositions, we associate each decomposition with an energy function defined on the transform domain called the *energy map* and find a unique texture representation by averaging the energy maps of all samples. Then, two schemes are considered in the classification phase. One scheme is to select $J$ channels containing largest $J$ energy values of unknown textures as features for the basis of comparison. The other scheme uses the channel with the largest energy value as a single feature to eliminates the unlikely choices. If there are still more than one possible choices remaining, we add the second largest energy channel as the second feature. The same procedure is repeated until there is no more ambiguity. The resulting method is called the progressive classification algorithm, which is not only computationally attractive but also has excellent performance. It provides a systematic way to classify textures without clear dominant frequencies. Note that this kind of texture is very difficult to discriminate for existing methods using the Gabor transform or multichannel filter-bank decomposition.

This paper is organized as follows. In Section 2, we briefly review the conventional wavelet transform and introduce the new tree-structured wavelet transform. We use an example to demonstrate the difference between conventional and tree-structured wavelet transforms, and explain why the new transform is appropriate for texture representation. Two texture classification algorithms are described in Section 3. The first algorithm uses a fixed number of features whereas the second algorithm uses a minimum number of features sufficient for classification. Experimental results of texture classification are presented in Section 4. We compare our method and methods based on the Gabor transform in Section 5. Concluding remarks are given in Section 6.

# 2 Tree-Structured Wavelet Transform for Textured Images

## 2.1 Review of Wavelet Transform

By the wavelet transform, we decompose a signal with a family of real orthonormal bases obtained through translation and dilation of a kernel function $\psi(x)$ known as the *mother wavelet*. The basis function of a discrete wavelet transform can be expressed as

$$\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n), \tag{2.1}$$

where $m$ and $n$ are integers. Due to the orthonormal property, the wavelet coefficients of a signal $f(x)$ can be easily computed via

$$c_{m,n} = \int_{-\infty}^{+\infty} f(x)\psi_{m,n}(x)dx,$$

and the synthesis formula

$$f(x) = \sum_{m,n} c_{m,n}\psi_{m,n}(x)$$

can be used to recover $f(x)$ from its wavelet coefficients.

To construct the mother wavelet $\psi(x)$, we may first determine a *scaling function* $\phi(x)$, which satisfies the two-scale difference equation [13], [42]

$$\phi(x) = \sqrt{2}\sum_{k} h(k)\phi(2x - k). \tag{2.2}$$

Then, the wavelet kernel $\psi(x)$ is related to the scaling function via

$$\psi(x) = \sqrt{2}\sum_{k} g(k)\phi(2x - k), \tag{2.3}$$

where

$$g(k) = (-1)^k h(1 - k). \tag{2.4}$$

The coefficients $h(k)$ in (2.2) have to meet several conditions for the set of basis wavelet functions in (2.1) to be uniqueness, orthonormal and with a certain degree of regularity [42]. Several different sets of coefficients $h(k)$ satisfying the above conditions can be found in the wavelet literature [13], [14], [36], [37] (see also Table 1).

The coefficients $h(k)$ and $g(k)$ play a very crucial role in a given discrete wavelet transform. In fact, to perform the wavelet transform does not require the explicit forms of $\phi(x)$

and $\psi(x)$ but only depends only on $h(k)$ and $g(k)$. To illustrate this, let us consider a function

$$f_0(x) = \sum_k c_{0,k}\phi_{0,k}(x), \tag{2.5}$$

where the coefficients $c_{0,k}$ are given. We want to decompose $f(x)$ into two components: a lower resolution component $f_1(x)$ and the wavelet component at scale $m = 1$. This can be achieved by projecting $f(x)$ on $\phi_{1,n}(x)$ and $\psi_{1,k}(x)$ or, equivalently, by integrating (2.5) with $\phi_{1,n}(x)$ and $\psi_{1,k}(x)$. Let the results be denoted by $c_{1,n}$ and $d_{1,n}$, respectively. It is straightforward to verify that

$$\begin{aligned} f_0(x) &= \sum_k c_{0,k}\phi_{0,k}(x), \\ &= \sum_k [c_{1,k}\phi_{1,k}(x) + d_{1,k}\psi_{1,k}(x)], \end{aligned}$$

where

$$c_{1,n} = \sum_k c_{0,k}h(k-2n), \qquad d_{1,n} = \sum_k c_{0,k}g(k-2n).$$

In fact, the same procedure can be applied to any scale $m = j+1$ for $j \geq 0$. That is, we have

$$\begin{aligned} f_j(x) &= \sum_k c_{j,k}\phi_{j,k}(x) \\ &= \sum_k [c_{j+1,k}\phi_{j+1,k}(x) + d_{j+1,k}\psi_{j+1,k}(x)], \end{aligned}$$

where the coefficients $c_{j+1,n}$ and $d_{j+1,n}$ at scale $j+1$ are related to the coefficients $c_{j,k}$ at scale $j$ via

$$c_{j+1,n} = \sum_k c_{j,k}h(k-2n), \qquad d_{j+1,n} = \sum_k c_{j,k}g(k-2n) \quad \text{for} \quad j \geq 0. \tag{2.6}$$

Thus, (2.6) provides a recursive algorithm for wavelet decomposition through $h(k)$ and $g(k)$. After $J$-level decomposition, the final outputs include a set of $J$-level wavelet coefficients $d_{j,n}$, $1 \leq j \leq J$, and the coefficient $c_{J,n}$ for a low resolution component $\phi_{J,k}(x)$. By using a similar approach, we can derive a recursive algorithm for function synthesis based on its wavelet coefficients $d_{j,n}$, $1 \leq j \leq J$, and $c_{J,n}$

$$c_{j,k} = \sum_n c_{j+1,n}h(k-2n) + \sum_n d_{j+1,n}g(k-2n). \tag{2.7}$$

It is convenient to view the decomposition (2.6) as passing a signal $c_{j,k}$ through a pair of filters $H$ and $G$ with impulse responses $\tilde{h}(n)$ and $\tilde{g}(n)$ and downsampling the filtered signals by two (dropping every other sample), where $\tilde{h}(n)$ and $\tilde{g}(n)$ are defined as

$$\tilde{h}(n) = h(-n), \qquad \tilde{g}(n) = g(-n).$$

The pair of filters $H$ and $G$ correspond to the halfband lowpass and highpass filters and are called the *quadrature mirror filters* in the signal processing literature, respectively. The reconstruction procedure is implemented by upsampling the subsignals $c_{j+1}$ and $d_{j+1}$ (inserting a zero between neighboring samples) and filtering with $h(n)$ and $g(n)$, respectively, and adding these two filtered signals together. Usually the signal decomposition scheme is performed recursively to the output of the lowpass filter $\tilde{h}$. It leads to the conventional wavelet transform or the so-called pyramid wavelet decomposition. Thus, the wavelet transform provides a multiresolution filter-bank decomposition of a signal with a set of orthonormal bases.

The coefficients $h(k)$, $0 \leq k \leq 15$, of the Battle-Lemarié and the 16-tap Daubechies wavelet transforms are listed in Table 1. The coefficients $h_b(k)$ of the Battle-Lemarié wavelet are symmetric, i.e. $h_b(k) = h_b(-k)$. Although $h_b(k)$ is an infinite duration sequence, it decays exponentially for large $k$. In contrast, the coefficients $h_d(k)$ of the 16-tap Daubechies wavelet has a compact support of length 16, i.e. $h_d(k) = 0$ if $k < 0$ or $k > 15$.

The 2-D wavelet transform can be formed by the tensor product of two 1-D wavelet transforms along the horizontal and vertical directions. The corresponding 2-D sequences can be written as

$$h_{LL}(k,l) = h(k)h(l) \qquad h_{LH}(k,l) = h(k)g(l)$$
$$h_{HL}(k,l) = g(k)h(l) \qquad h_{HH}(k,l) = g(k)g(l),$$

where the first and second subscripts denote the lowpass or highpass filtering characteristics of the corresponding filters in the $x$- and $y$-directions, respectively. The conventional pyramid wavelet transform performs further wavelet decomposition recursively to the output of the lowpass filter $H_{LL}$ with impulse response $\tilde{h}_{LL}(k,l) = \tilde{h}(k)\tilde{h}(l)$.

## 2.2 Tree-structured Wavelet Transform

The pyramid-structured wavelet transform decomposes a signal into a set of frequency channels which have narrower bandwidths in the lower frequency region. The transform is

| | Battle-Lemarié | 16-tap Daubechies |
|---|---|---|
| $h(0)$ | 0.7661300537597422 | 0.054415842243 |
| $h(1)$ | 0.4339226335893024 | 0.312871590914 |
| $h(2)$ | -0.0502017246714322 | 0.675630736297 |
| $h(3)$ | -0.1100370183880987 | 0.585354683654 |
| $h(4)$ | 0.0320808974701767 | -0.015829105256 |
| $h(5)$ | 0.0420683514407039 | -0.284015542962 |
| $h(6)$ | -0.0171763154919797 | 0.000472484574 |
| $h(7)$ | -0.0179823209809603 | 0.128747426620 |
| $h(8)$ | 0.0086852948130698 | -0.017369301002 |
| $h(9)$ | 0.0082014772059938 | -0.044088253931 |
| $h(10)$ | -0.0043538394577629 | 0.013981027917 |
| $h(11)$ | -0.0038824252655926 | 0.008746094047 |
| $h(12)$ | 0.0021867123701413 | -0.004870352993 |
| $h(13)$ | 0.0018821335238871 | -0.000391740373 |
| $h(14)$ | -0.0011037398203844 | 0.000675449409 |
| $h(15)$ | -0.0009271987314557 | -0.000117476784 |

Table 1: Wavelet Transform Filter Coefficients

suitable for signals consisting primarily of smooth components so that their information is concentrated in the low frequency regions. However, it is not suitable for textured images, since textures are quasi-periodic signals whose dominant frequency channels are located in the middle frequency region. This concept can be illustrated in Figure 1 where the pyramid-structured wavelet transform is applied to two different kinds of images. We use the Lena image as a representative for an ordinary image. The image and its pyramid-structured wavelet transform are shown in Figure 1 (a). The textured image French Canvas (D21) and its pyramid-structured wavelet transform are shown in Figure 1 (b) for comparison. By examining the wavelet transformed image in Figure 1 (a), we recognize the Lena image clearly from its low frequency channel (the upper left corner). In contrast, we are not able to recognize a similar texture pattern in the same low frequency channel for Figure 1 (b). Instead, we observe some horizontal and vertical line patterns clearly in the middle frequency region. The simple experiment implies that the low frequency region of textures may not necessarily contain significant information. Thus, an appropriate way to perform the wavelet transform for textures is to detect the significant frequency channels and then to decompose them further.

The above idea leads naturally to a new type wavelet transform which is called the tree-structured wavelet transform. The key difference between this algorithm and the traditional pyramid algorithm is that the decomposition is no longer simply applied to the low frequency subsignals recursively. Instead, it can be applied to the output of any filter $H_{LL}$, $H_{LH}$, $H_{HL}$ or $H_{HH}$. The block diagram of the tree-structured wavelet decomposition is shown in Figure 2. Note that it is usually unnecessary and expensive to decompose all subsignals in each scale to achieve a full decomposition. To avoid a full decomposition, we may consider a criterion to decide whether a decomposition is needed for a particular output. We use the averaged $l_1$-norm

$$e(\mathbf{x}) = \frac{1}{N}||\mathbf{x}||_1 = \frac{1}{N}\sum_{i=1}^{N}|x_i|, \tag{2.8}$$

where $\mathbf{x} = (x_1, \cdots, x_N)$, as the energy function to locate dominant frequency channels. Although there exist other norms such as the $l_2$-norm for the energy function, we find that the $l_1$ and $l_2$ norms make little difference in the final results. The energy function (2.8) is chosen due to its simplicity. The tree-structured wavelet transform is given below.

**Algorithm 1 : Tree-Structured Wavelet Transform**

1. Decompose a given textured image with 2D two-scale wavelet transform into 4 subimages, which can be viewed as the parent and children nodes in a tree.

2. Calculate the energy of each decomposed image (children node). That is, if the decomposed image is $x(m,n)$, with $1 \leq m \leq M$ and $1 \leq n \leq N$, the energy is

$$e = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|x(m,n)|.$$

3. If the energy of a subimage is significantly smaller than others, we stop the decomposition in this region since it contains less information. This step can be achieved by comparing the energy with the largest energy value in the same scale. That is, if $e < Ce_{\max}$, stop decomposing this region where $C$ is a constant less than 1.

4. If the energy of a subimage is significantly larger, we apply the above decomposition procedure to the subimage.

Four textures and their tree-structured wavelet decompositions with $C = 0.3$ are shown in Figure 3. With respect to each tree-structured wavelet transform, we calculate the

normalized energy at its leaves, and obtain a normalized energy function defined on the spatial/frequency domain called the *energy map*, which will be used for texture classification. Figure 4 shows the energy map of texture D53. Note that the energy map is in fact representable by the well known *quadtree* structure [28].

To investigate the robustness of the tree-structured wavelet decomposition, we apply the transform to 100 samples of French Canvas (D21) and Oriental Straw Cloth (D53) with $C = 0.15$. It turns out that all samples of texture D53 have exactly the same 4-level tree structure as given in Figure 3 (b). In contrast, there exist two 3-level tree structures occurring 84 and 16 times, respectively, for texture D21. With a further decomposition, we observe sixteen 4-level tree structures. Actually, the number of different tree structures highly depends on the constant $C$. For example, by decreasing the value of $C$, the tree-structure of texture D53 will not be the same for all samples. This observation suggests that the structure of the tree may not serve as a good candidate for texture representation.

However, if we focus on dominant frequency channels having large energy value, the representation becomes much more robust. Table 2 shows different kinds of energy maps with the first 5 dominant channels for 100 samples of texture D53 with threshold $C = 0.15$. The meaning of the frequency channel ABBB is described in Figure 5. We see from Table 2 that the first 3 dominant channels are the same for all samples. Although the 4th and 5th dominant frequency channels may differ, only three channels (AABB, AACC and AABC) appear.

| tree type | number of occurrence | dominant frequency channels | | | | |
|---|---|---|---|---|---|---|
| | | first | second | third | fourth | fifth |
| 1 | 49 | ABBB | ABAB | BBBB | AABB | AACC |
| 2 | 29 | ABBB | ABAB | BBBB | AABB | AABC |
| 3 | 14 | ABBB | ABAB | BBBB | AACC | AABB |
| 4 | 5 | ABBB | ABAB | BBBB | AABC | AABB |
| 5 | 3 | ABBB | ABAB | BBBB | AACC | AABC |

Table 2: The first 5 dominant channels for texture D53.

For texture D21 with $C = 0.15$, we can classify them into 8 different types according to the first 5 dominant channels as shown in Table 3. Note that there are only six different channels appearing in the table. The first two dominant channels are either $ADDD$ or $ADAD$, and the 3rd and 4th dominant channels are either $BABA$ or $BBBA$.

| tree type | number of occurrence | dominant frequency channels | | | | |
|---|---|---|---|---|---|---|
| | | first | second | third | fourth | fifth |
| 1 | 38 | ADAD | ADDD | BABA | BBBA | ABAB |
| 2 | 20 | ADAD | ADDD | BABA | BBBA | ABBB |
| 3 | 17 | ADDD | ADAD | BABA | BBBA | ABAB |
| 4 | 8 | ADDD | ADAD | BABA | BBBA | ABBB |
| 5 | 7 | ADAD | ADDD | BBBA | BABA | ABAB |
| 6 | 5 | ADDD | ADAD | BBBA | BABA | ABAB |
| 7 | 4 | ADAD | ADDD | BBBA | BABA | ABBB |
| 8 | 1 | ADDD | ADAD | BBBA | BABA | ABBB |

Table 3: The first 5 dominant channels for texture D21.

The tree-structured wavelet transform generates a multiresolution/multichannel texture representation with complete basis functions which have a good spatial/frequency localization property. It is interesting to compare the tree-structured wavelet transform with the local linear transform method from the multichannel decomposition viewpoint. The local linear transform method uses a fixed numbers of filter masks with predetermined frequencies and bandwidths. The set of filter masks are usually determined by extensive experiments and may vary for different textures. In contrast, the tree-structured wavelet transform determine important channels dynamically according to a certain energy calculation and can be viewed as an adaptive multichannel method.

# 3 Texture Classification Algorithms

## 3.1 Classification with Fixed Number of Features

A simple texture classification algorithm follows directly from Algorithm 1. The process is detailed as follows.

**Algorithm 2 : Classification algorithm with $J$ features.**

- Learning phase.

  1. Given $m$ samples obtained from the same texture, decompose each sample with the tree-structured wavelet transform and calculate the normalized energy at its leaves which defines an energy function on the spatial/frequency domain known as the energy map.

2. Generate a representative energy map for each texture by averaging the energy maps over all $m$ samples.
3. Repeat the process for all textures.

- Classification phase

1. Decompose an unknown texture with the tree-structured wavelet transform and construct its energy map.
2. Pick up the first $J$ dominant channels which are the leaf nodes in the energy map with the largest energy values as features. Denote this feature set by $\mathbf{x} = (x_1, \ldots, x_J)$.
3. For texture $i$ in the database, pick up the energy values in the same channels and denote the energy value by $\mathbf{m}_i = (m_{i,1}, \ldots, m_{i,J})$. Note that if the associated leaf node does not exist in the energy map, the normalized energy value of its parent node can be used. This often means that the energy value of this channel is small so that the texture is not similar to the unknown texture and we discard texture $i$ from the candidate list and test the next texture in the database.
4. Calculate the discrimination function for textures in the candidate list by

$$D_i = \text{distance}(\mathbf{x}, \mathbf{m}_i). \tag{3.1}$$

5. Assign the unknown texture to texture $i$ if $D_i < D_j$ for all $j \neq i$.

The above algorithm uses the energy values at the $J$ most dominant channels as features for classification. Several distance functions can be used in (3.1). In particular, we consider four such functions and list them in Table 4, where $C_i$ is the covariance matrix of the feature set for texture $i$.

The Mahalanobis distance is a useful measure of similarity if some statistical properties of textures are known. In particular, if $C_i$ is a diagonal matrix or, equivalently, features are independent of each other, the Mahalanobis distance reduces to the form

$$D_{4,i} = \sum_{j=1}^{J} \frac{(x_j - m_{i,j})^2}{c_{i,j}}, \tag{3.2}$$

where $c_{i,j}$ is the variance of feature $j$ and class $i$. For the case that $C_i$ is not diagonal but diagonally dominant, formula (3.2) is still a good choice for distance measure. One special

| | |
|---|---|
| Euclidean Distance: | $D_{1,i} = \sum_{j=1}^{J}(x_j - m_{i,j})^2$ |
| Bayes Distance: | $D_{2,i} = (x - m_i)^T C_i^{-1}(x - m_i) + \ln \lvert C_i \rvert$ |
| Mahalanobis Distance: | $D_{3,i} = (x - m_i)^T C_i^{-1}(x - m_i)$ |
| Simplified Mahalanobis Distance: | $D_{4,i} = \sum_{j=1}^{J} \frac{(x_j - m_{i,j})^2}{c_{i,j}}$ |

Table 4: Four distance functions for classification.

feature of (3.2) is that it can be computed recursively. That is, the distance $D_{4,i}^{(J+1)}$ using $J+1$ features can be expressed in terms of the partial sum $D_{4,i}^{(J)}$ and the new feature $x_{J+1}$ as

$$\begin{aligned} D_{4,i}^{(J+1)} &= \sum_{j=1}^{J+1} \frac{(x_j - m_{i,j})^2}{c_{i,j}} \\ &= D_{4,i}^{(J)} + \frac{(x_{J+1} - m_{i,J+1})^2}{c_{i,J+1}}. \end{aligned} \tag{3.3}$$

Furthermore, if the joint density function of features is available, more sophisticated distance measures such as the one based on the Bayes decision rule can be used. By the law of large number, we may assume that the density function of features is Gaussian so that the Bayes decision function assumes the form

$$D_2(x, m_i, C_i) = (x - m_i)^T C_i^{-1}(x - m_i) + \ln |C_i|.$$

Note that the Bayes distance is similar to the Mahalanobis distance for features with Gaussian distribution except the addition of the second term known as the covariance difference.

## 3.2 Progressive Classification Algorithm

The above classification algorithm is effective for classifying textures with special dominant frequency channels. The experiment shows that textures of this type can be classified using only one or two features (see Section 4). However, for the situation where textures who have no significant dominant frequency channels (random textures) or where multiple textures who have similar dominant frequency channels, we need more features for classification. It is difficult to determine the number of features required a priori. For example, if its first $J$ dominant channels are close to several textures in the database, more than $J$ features may be needed for its classification. This problem can be solved by a progressive classification

algorithm where we start with one feature which eliminates very unlikely candidates and then another feature is added for further elimination. The procedure is repeated until only there is only one texture left in the candidate list. The algorithm is detailed as follows.

**Algorithm 3 : Progressive classification algorithm**

1. Use the tree-structured wavelet transform to decompose an unknown texture and construct its energy map.

2. Use the channels with large energy values $x_j$, $j = 1, 2, \cdots$, as features and arrange them in order, i.e. $x_1 > x_2 > \cdots > x_{j-1} > x_j > x_{j+1} > \cdots$.

3. Order textures in the database into a candidate list, and perform the following iteration from the first feature (i.e. $x_k, k = 1$).

   (i) Remove textures from the candidate list if they do not have the same leaf node as the $k$th dominant channel of the unknown texture.

   (ii) For the remaining textures, denote the energy value and the variance in this channel by $m_{i,j}$ and $c_{i,j}$ for texture $i$ and feature $j$, where $j \leq k$, and calculate the simplified Mahalanobis distance

   $$D_{4,i} = \sum_{j=1}^{J} \frac{(x_j - m_{i,j})^2}{c_{i,j}}.$$

   Let $D_{\min} = \min D_{4,i}$. If $D_{4,i} > K D_{\min}$, where $K$ is a constant greater than one, remove texture $i$ from the list.

   (iii) If there is only one texture left, assign the unknown texture to this texture. Otherwise, perform the next iteration by increasing the value of $k$ by one.

In step 3-(i), we discard the very unlikely textures because of poor similarity. The algorithm usually removes most irrelevant textures from the list in only one or two iterations, and all remaining textures are visually similar to the unknown texture. Although the number of features used for classifying a particular texture may be large, the computational complexity is in fact much lower than the one using a fixed number of features. The constant $K$ in step 2-(ii) is a controllable parameter which serves as a threshold for eliminating irrelevant textures. Empirically, $K = 10$ gives a very satisfactory result. We can always increase the reliability of the algorithm by increasing the value of $K$ with higher computational cost.

# 4 Experimental Results

We use 30 textures obtained from the Brodatz's texture album [4] and list them in Table 5. Each texture is scanned with 150 dpi resolution, and each image is of size 512 x 512 pixels with 256 gray levels. The mean of each image is removed before the processing. One hundred sample images of size 256 x 256 are randomly chosen from the original image and used in both training and classification phases. The mean and the covariance of the samples are calculated with the leave-one-out algorithm [21], [33] in classification.

| Texture ID | Texture Description | Texture ID | Texture Description |
|---|---|---|---|
| D3 | Reptile skin | D57 | Handmade paper |
| D4 | Pressed cork | D65 | Handwoven oriental rattan |
| D6 | Woven aluminum wire | D68 | Wood grain |
| D9 | Grass lawn | D74 | Coffee beans |
| D11 | Homoespun woolen cloth | D77 | Cotton canvas |
| D16 | Herringbone | D78 | Oriental straw cloth 3 |
| D19 | Woolen cloth with soft tufts | D79 | Oriental grass fiber cloth |
| D21 | French canvas | D82 | Oriental straw cloth 4 |
| D24 | Pressed calf leather | D83 | Woven matting |
| D29 | Beach sand | D84 | Raffia looped to a high pile |
| D34 | Netting | D92 | Pigskin |
| D36 | Lizard skin | D95 | Brick wall |
| D52 | Oriental straw cloth 1 | D102 | Cane |
| D53 | Oriental straw cloth 2 | D103 | Loose burlap |
| D55 | Straw matting | D105 | Cheesecloth |

Table 5: Textures used in the experiments.

## 4.1 Classification with Fixed Number of Features

We use the largest 5 dominant channels of the unknown texture as features, and classify the texture with 4 distance functions summarized in Table 4. Experimental results with Battle-Lemarié cubic spline wavelet basis functions are given in Table 6. The average of classification rates of the 30 tested textures is called the overall correct classification rate and listed in the last row of the table.

Figure 6 shows the overall correct classification rate as a function of the number of features for different distance measures. It is clear from the figure that more than 3 fea-

| Texture ID | Correct Classification Rate | | | |
|---|---|---|---|---|
| | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
| D3 | 88% | 100% | 100% | 100% |
| D4 | 58% | 96% | 97% | 95% |
| D6 | 98% | 100% | 100% | 100% |
| D9 | 96% | 97% | 96% | 91% |
| D11 | 100% | 100% | 100% | 100% |
| D16 | 100% | 100% | 100% | 100% |
| D19 | 100% | 100% | 100% | 100% |
| D21 | 100% | 100% | 100% | 100% |
| D24 | 100% | 100% | 100% | 99% |
| D29 | 92% | 100% | 100% | 99% |
| D34 | 76% | 100% | 100% | 100% |
| D36 | 100% | 100% | 100% | 100% |
| D52 | 96% | 100% | 100% | 100% |
| D53 | 100% | 100% | 100% | 100% |
| D55 | 99% | 100% | 100% | 100% |
| D57 | 98% | 100% | 100% | 100% |
| D65 | 100% | 100% | 100% | 97% |
| D68 | 97% | 99% | 99% | 98% |
| D74 | 100% | 100% | 89% | 96% |
| D77 | 100% | 100% | 100% | 100% |
| D78 | 82% | 98% | 100% | 93% |
| D79 | 100% | 100% | 100% | 100% |
| D82 | 100% | 100% | 100% | 100% |
| D83 | 86% | 99% | 100% | 98% |
| D84 | 100% | 100% | 100% | 100% |
| D92 | 100% | 100% | 100% | 100% |
| D95 | 98% | 100% | 100% | 100% |
| D102 | 96% | 100% | 100% | 100% |
| D103 | 100% | 100% | 100% | 100% |
| D105 | 100% | 100% | 100% | 100% |
| Overall | 95.3% | 99.6% | 99.4% | 98.9% |

Table 6: Classification results with 4 distance functions.

tures are needed for the overall rate to be greater than 90 percent. Distance $D_2$ has the best performance whereas distance $D_1$ has the worst performance. Note also that $D_2$ is particularly attractive for a small number of features and $D_1$ should be avoided for a larger number of features. Distances $D_2$, $D_3$ and $D_4$ give a similar performance, when the number of features is more than 3. This result implies that the correct classification rate will be higher by incorporating the statistical information of texture features.

We show the overall correct classification rate with different Daubechies wavelet bases in Figure 7. The performance is better for filters with a larger size especially when four or less features are used. Since the transform filter $\tilde{H}$ is in fact a lowpass filter, it will be more close to an ideal lowpass filter if more coefficients are used. Besides, we observe a very similar performance for the 16-tap Daubechies wavelet and the truncated Battle-Lemarié cubic spline wavelet with coefficients $h_b(k)$, $-8 \leq k \leq 8$. This indicates that the classification algorithm is insensitive to different wavelet bases in use.

## 4.2 Progressive Classification Algorithm

Although the covariance matrix of features is only diagonally dominant, the simplified Mahalanobis distance $D_4$ performs reasonably well according to Figure 6 and Figure 7. Since it can be computed recursively from the partial sum in the previous iteration as indicated in (3.3), the Mahalanobis distance $D_4$ with independent feature assumption is used in the progressive classification algorithm. We list the average number of features in the progressive classification algorithm in Table 7 with the Battle-Lemarié cubic spline wavelet basis. We see from the table that 100% correct classification rate can be achieved with only 4 features on the average.

Textures which cannot be discriminated easily using 5 features in the previous experiment require more features in the progressive algorithm as well, since they are similar to other textures whose dominant frequencies are in the low frequency region. Despite more features are needed, the computational complexity is not as high as in the non-progressive algorithm. After two or three iterations, only a few textures which are similar to the unknown texture in the dominant frequencies remain in the candidate list. Thus, the overall computational complexity is still low. For textures significantly different from the others, only one or two features are needed.

We expect that the average number of features in the progressive algorithm will increase if we have more texture samples in the database, since the chance of several textures with

| Texture ID | Average numbers of Features |
|---|---|
| D3 | 2.78 |
| D4 | 7.78 |
| D6 | 3.58 |
| D9 | 8.96 |
| D11 | 2.27 |
| D16 | 1.00 |
| D19 | 5.82 |
| D21 | 1.00 |
| D24 | 6.37 |
| D29 | 6.68 |
| D34 | 4.08 |
| D36 | 1.19 |
| D52 | 2.65 |
| D53 | 1.00 |
| D55 | 2.87 |
| D57 | 4.08 |
| D65 | 5.98 |
| D68 | 8.46 |
| D74 | 9.68 |
| D77 | 2.00 |
| D78 | 5.20 |
| D79 | 3.14 |
| D82 | 3.99 |
| D83 | 5.68 |
| D84 | 1.88 |
| D92 | 5.50 |
| D95 | 4.16 |
| D102 | 1.59 |
| D103 | 1.05 |
| D105 | 1.31 |
| Overall | 4.0577 |

Table 7: Average features used in progressive classification.

similar dominant frequencies becomes higher. To discriminate textures with similar dominant frequencies is in general a difficult task. The error probability of all other multichannel or multiresolution methods will increase. The problem is however more manageable by systematically increasing the number of features with the progressive algorithm. This is an important feature for the progressive algorithm, which makes the tree-structured wavelet decomposition more attractive than other known multichannel or multiresolution methods.

# 5 Comparison of Methods Based on Wavelet and Gabor Transforms

Both wavelet and Gabor Transforms provide good analytic tools for nonstationary signals such as textures. For the Gabor transform, we project a signal on a set of basic functions of the form

$$g_{m,n}(x) = g(x - n\alpha)\exp(j2\pi m\beta(x - n\alpha)),$$

where $m$, $n$ are integers and the Gaussian function is often used as the envelope function, i.e.

$$g(x) = 2^{1/4}\exp(-\pi x^2).$$

Motivated by results from vision research, Porat [38] suggested to adopt a nonuniform sampling in the frequency axis with the octave scale. The idea leads to an expansion known as the Gaborian pyramid algorithm which is very close to the pyramid-structured wavelet transform except that the basis functions are not orthogonal.

It is worthwhile to point out several distinct features of our method in comparison with methods using the Gabor transform. First, the wavelet transform provides a complete representation in the sense that the original signal can be exactly reconstructed from wavelet coefficients. In contrast, oversampling is needed to completely represent a signal with the Gabor transform, Second, the filters $h(n)$ and $g(n)$ used in the wavelet transform remain the same between two consecutive scales whereas the Gabor transform requires different filters. Third, textures with no strong peaks in their Fourier power spectra cannot be easily handled by the Gabor transform, since a few dominant frequency channels have to be decided a priori as features. The Fourier power spectra of textures D21 and D53 are shown in Figure 8. Since they have strong peaks in their spectra, both Gabor and wavelet transforms are applicable. However, some textures may not have strong periodicities and their energy spreads in the low frequency region. One such example is illustrated in Figures 9 (a) and (b), where

texture pressed cork (D4) and its Fourier power spectrum are given. It is very difficult to locate the peaks since they are clustered in the low frequency region. Beside, their positions vary from samples to samples. The tree-structured wavelet transform applied to the same texture is shown in Figures 9 (c) and (d), where low and high energy thresholds ($C = 0.3$ and 0.6) are used for terminating further decomposition, respectively. We find that channel AAAA contains the largest amount of energy, and the feature is quite robust with respect to different samples of pressed cork in the tree-structured wavelet representation. Fourth, the tree-structured wavelet transform determines the feature channels automatically in the transform process whereas the Gabor transform needs an additional procedure to estimate feature parameters such as spatial frequencies and orientations. Last, the wavelet transform is computationally efficient. Since the image is downsampled throughout tree-structured wavelet decomposition, the number of operations is proportional to $O(N/4^i)$ and the total complexity for a $M$-scale decomposition can be roughly estimated by

$$\sum_{i=0}^{M} \frac{N}{4^i} < \frac{4N}{3}.$$

Note that the above estimate is independent of the number of features and the depth of the tree but it does depend on the number of branches in the tree. For comparison, the Gabor transform requires $O(JN)$ operations where $J$ is the number of features used.

# 6 Conclusion

It is shown in this research that the tree-structured wavelet transform provides a good analytic tool for texture analysis. Although the conventional pyramid-structured wavelet transform is suitable for images with energy concentrated in the low frequency region, the tree-structured wavelet transform is more natural and effective for textures which have dominant middle frequency channels. The use of Gabor transform as a texture discriminant requires several parameters such as spatial frequencies and orientations, and the shape of the Gaussian envelope function, whereas the tree-structured wavelet transform provides a systematic procedure to zoom into dominant frequency channels. The progressive texture classification algorithm can be used to classify textures with similar dominant frequencies. It is not only computationally attractive but also has excellent performance. Our current work has so far focused on algorithmic development and experimental justification. More thorough theoretical analysis is expected in the future. The application of tree-structured

wavelet transform to texture segmentation is under our current investigation. The tree-structured wavelet transform may also be applicable to speech processing and other interesting problems.

## References


[1] C. Bounman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 13, pp. 99–113, Feb. 1991.

[2] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. on Signal Processing*, Vol. 39, pp. 2025–2043, Sept. 91.

[3] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 12, Jan. 1990.

[4] P. Brodatz, *Textures: A Photographic Album for Artists & Designers*, Dover Publications, Inc., New York, 1966.

[5] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," *Pattern Recognition*, Vol. 2, pp. 79–112, 1985.

[6] P. C. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 5, pp. 64–69, Jan. 1983.

[7] H.-I. Choi and W. J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, Vol. 37, pp. 862–871, June 1989.

[8] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 9, pp. 195–219, Mar. 1987.

[9] L. Cohen, "Time-frequency distributions – a review," *Proceedings of the IEEE*, Vol. 77, pp. 941–981, July 1989.

[10] P. Cohen, C. T. Ledinh, and V. Lacasse, "Classification of natural textures by means of two-dimensional orthogonal masks," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, Vol. 37, pp. 125–128, Jan. 1989.

[11] R. W. Conners and C. A. Harlow, "A theoretical comparison of texture algorithm," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 2, pp. 204–222, May 1980.

[12] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 5, pp. 25–39, Jan. 1983.

[13] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, Vol. 41, pp. 909–996, Nov. 1988.

[14] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Information Theory*, Vol. 36, pp. 961–1005, Sept. 1990.

[15] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for images analysis and compression," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, Vol. 36, pp. 1169–1179, July 1988.

[16] H. Derin, "Segmentation of textured images using Gibbs random fields," *Computer, Vision, Graphics, and Image Processing*, Vol. 35, pp. 72–98, 1986.

[17] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 9, pp. 39–55, Jan. 1987.

[18] H. Derin, H. Elliott, R. Cristi, and D. Geman, "Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 6, pp. 707–720, Nov. 1984.

[19] O. D. Faugeras and W. K. Pratt, "Decorrelation methods of texture feature extraction," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 2, pp. 323–332, July 1980.

[20] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological Cybernetics*, Vol. 61, pp. 103–113, 1989.

[21] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Boston, 1990.

[22] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 6, pp. 721–741, Nov. 1984.

[23] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, Vol. 67, pp. 786–804, May 1979.

[24] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification," *IEEE Trans. on System, Man, and Cybernetic*, Vol. 8, pp. 610–621, Nov. 1973.

[25] C. E. Heil and D. F. Walnut, "Continuous and discrete wavelet transforms," *SIAM Review*, Vol. 31, pp. 628–666, Dec. 1989.

[26] J. Y. Hsiao and A. A. Sawchuk, "Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 11, pp. 1279–1292, Dec. 1989.

[27] J. Y. Hsiao and A. A. Sawchuk, "Unsupervised textured image segmentation using feature smoothing and probabilistic relaxation techniques," *Computer, Vision, Graphics, and Image Processing*, Vol. 48, pp. 1–21, 1989.

[28] A. K. Jain, *Fundamentals of Digital Image Processinng*, Prenntice Hall, Inc., New Jersey, 1989.

[29] B. Julesz, "Visual pattern discrimination," *IRE Transactions on Information Theory*, Vol. 8, pp. 84–97, Feb. 1962.

[30] B. Julesz, "Experiments in the visual perception of texture," *Scientific American*, Vol. 232, pp. 765–771, June 1977.

[31] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial-interaction models of images," *IEEE Trans. on Information Theory*, Vol. 29, pp. 60–72, Jan. 1983.

[32] R. L. Kashyap and R. Chellappa, "Texture synthesis using 2-D noncausal autoregressive models," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, Vol. 33, pp. 194–203, Jan. 1985.

[33] P. A. Lachenbruch, "Estimation of error rates in discriminant analysis," ph.d. dissertation, Univ. of California, Los Angeles, 1965.

[34] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 11, pp. 799–813, Aug. 1989.

[35] K. I. Laws, "Texture Image Segmentation," Ph.D. dissertation 940, Image Processing Inst., Univ. of Southern California, Jan. 1980.

[36] S. G. Mallat, "Multiresolution approximation and wavelet orthonormal bases of L2," *Trans. Amer. Math. Soc.*, June 1989.

[37] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 11, pp. 674–693, July 1989.

[38] M. Porat and Y. Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 10, pp. 452–467, July 1988.

[39] T. R. Reed and H. Wechsler, "Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 12, pp. 1–12, Jan. 1990.

[40] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE SP Magazine*, pp. 14–38, Oct. 1991.

[41] A. Rosenfeld and L. S. Davis, "Image segmentation and image models," *Proceedings of the IEEE*, Vol. 67, pp. 764–773, May 1979.

[42] G. Strang, "Wavelets and dilation equations: a brief introduction," *SIAM Review*, Vol. 31, pp. 614–627, Dec. 1989.

[43] M. Unser, "Local linear transforms for texture measurements," *Signal Processing*, Vol. 11, pp. 61–79, 1986.

[44] M. Unser, "Sum and difference histograms for texture classification," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 8, pp. 118–125, Jan. 1986.

[45] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 11, pp. 717–728, July 1989.

[46] R. Wilson and G. H. Granlund, "The uncertainty principle in image processing," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 6, pp. 758–767, Nov. 1984.

[47] J. W. Woods, S. Dravida, and R. Mediavillā, "Image estimation using doubly stochastic Gaussian random field models," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 9, pp. 245–253, Mar. 1987.

[48] R. Yokoyama and R. M. Haralick, "Texture pattern image generation by regular Markov chain," *Pattern Recognition*, Vol. 11, pp. 225–234, 1979.

# Figure Captions

Figure 1: Pyramid-structured wavelet transforms of (a) Lena and (b) French Canvas.

Figure 2: Block diagram of tree-structured wavelet decomposition.

Figure 3: Tree-structured wavelet transforms of (a) French Canvas (D21), (b) Oriental Straw Cloth (D53), (c) Wool (D19) and (d) Raffia (D84).

Figure 4: Energy map for Oriental Straw Cloth (D53).

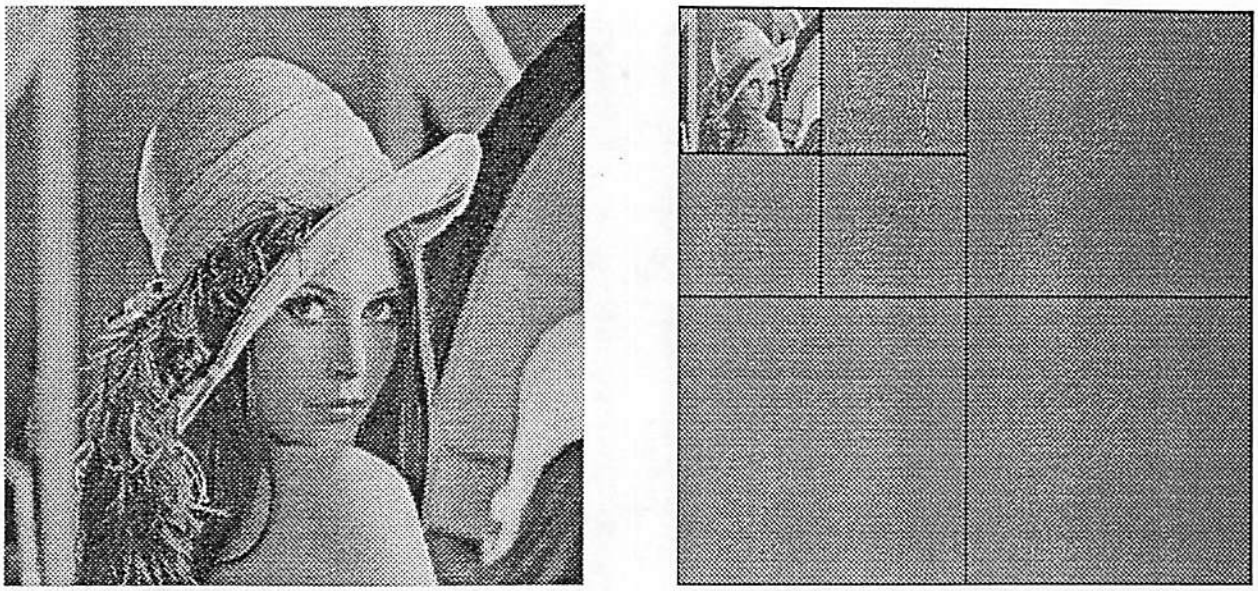
Figure 5: The meaning of Channel ABBB in (a) tree-structured wavelet transform domain and (b) quadtree representation.

Figure 6: Correct classification rate for Battle-Lemarié cubic spline wavelet basis.

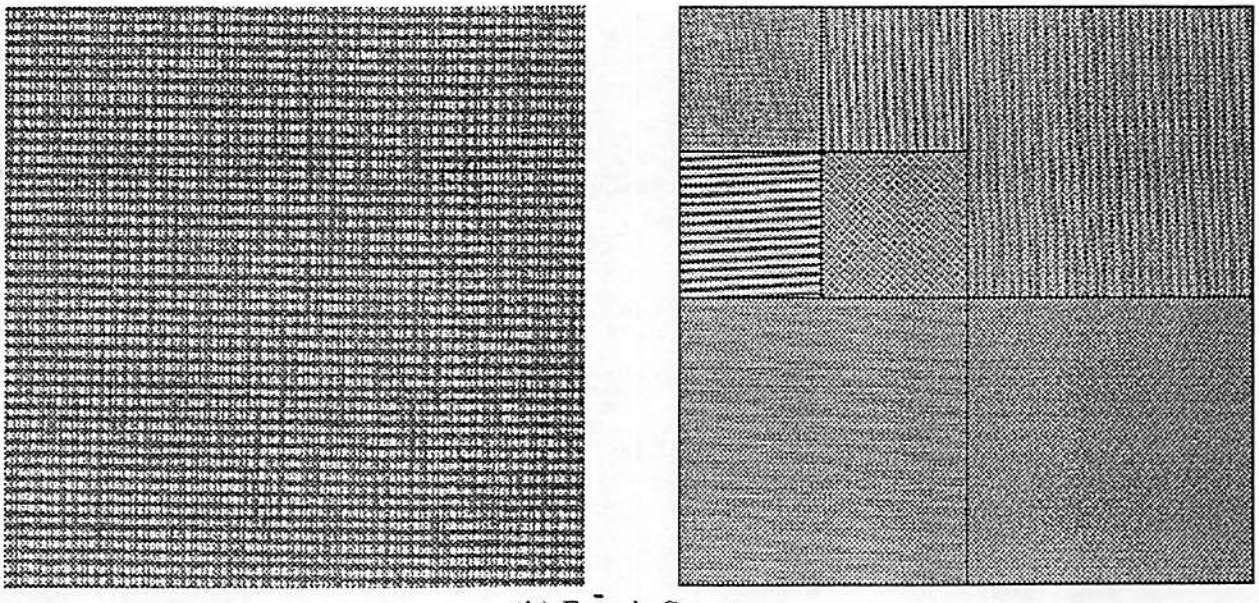
Figure 7: Correct classification rates for different Daubechies wavelet bases.

Figure 8: Fourier power spectra for (a) French Canvas (D21) and (b) Oriental Straw Cloth (D53).

Figure 9: Pressed Cork (D4) and its Fourier power spectrum and tree-structured wavelet transform maps.



(a) Lena



(b) French Canvas

Figure 1: Pyramid-structured wavelet transforms of (a) Lena and (b) French Canvas.

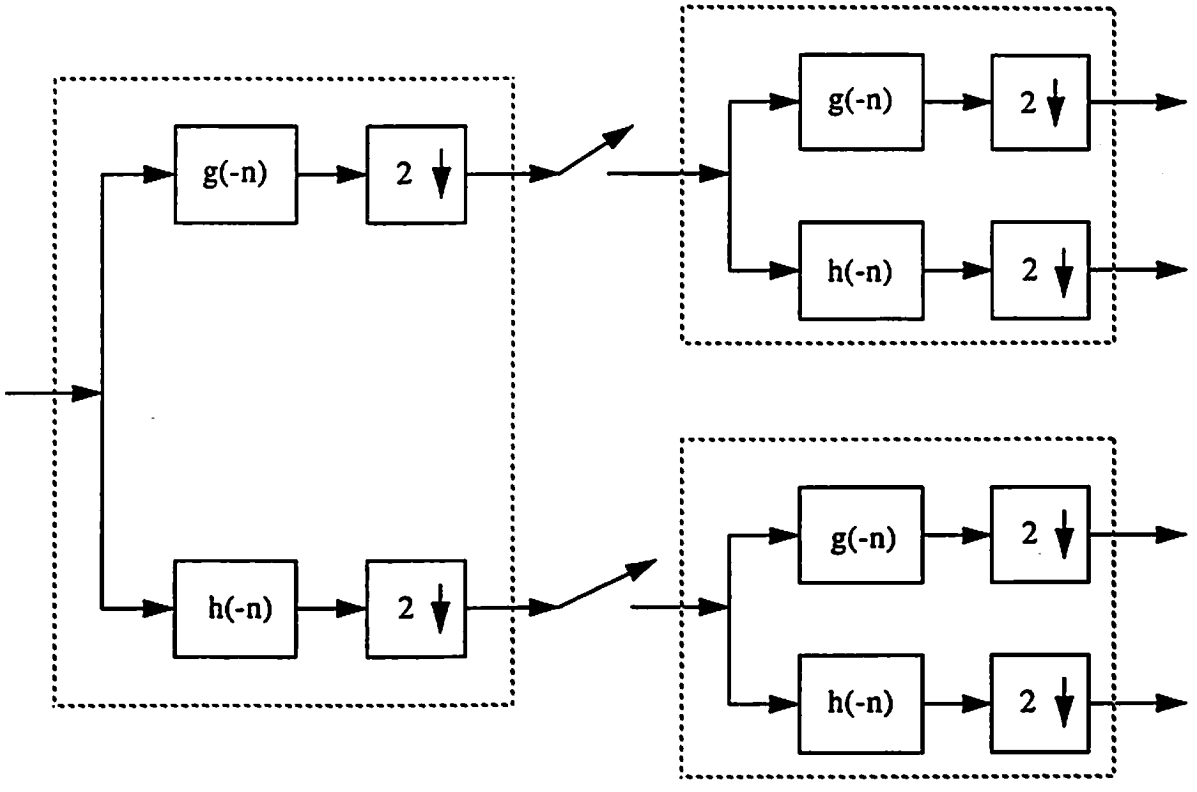
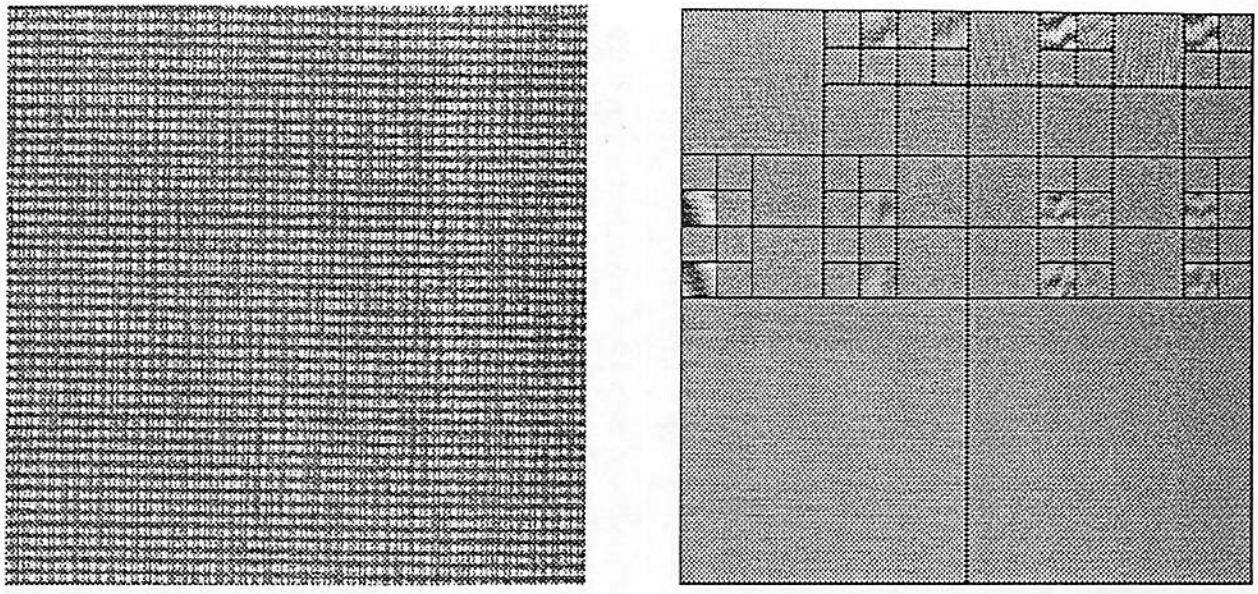
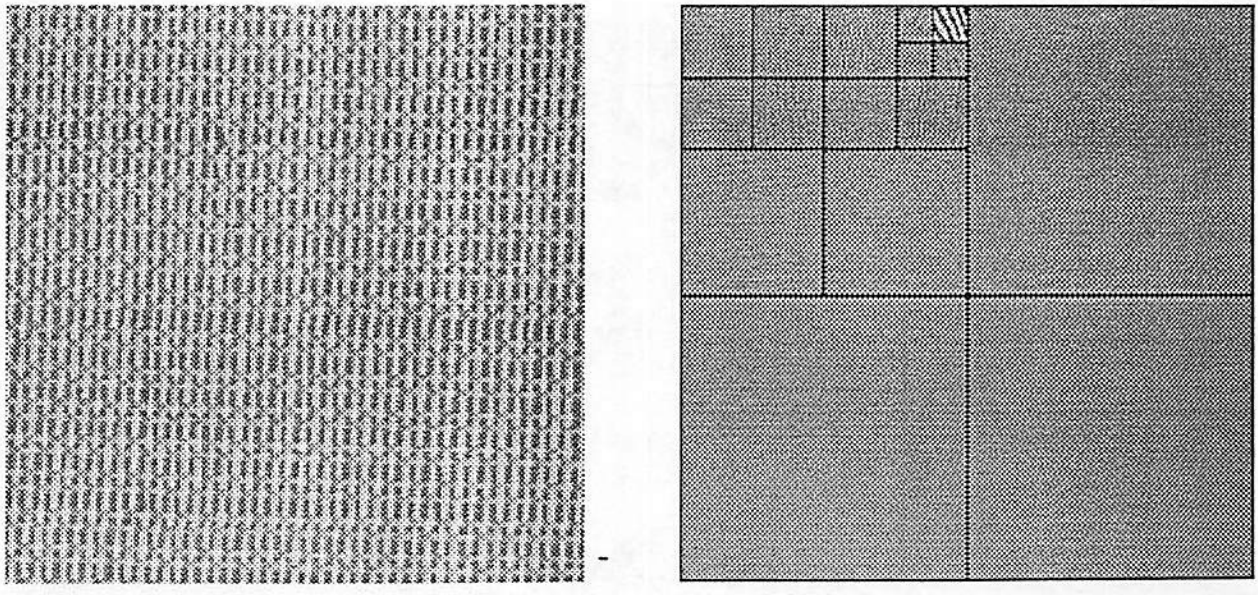


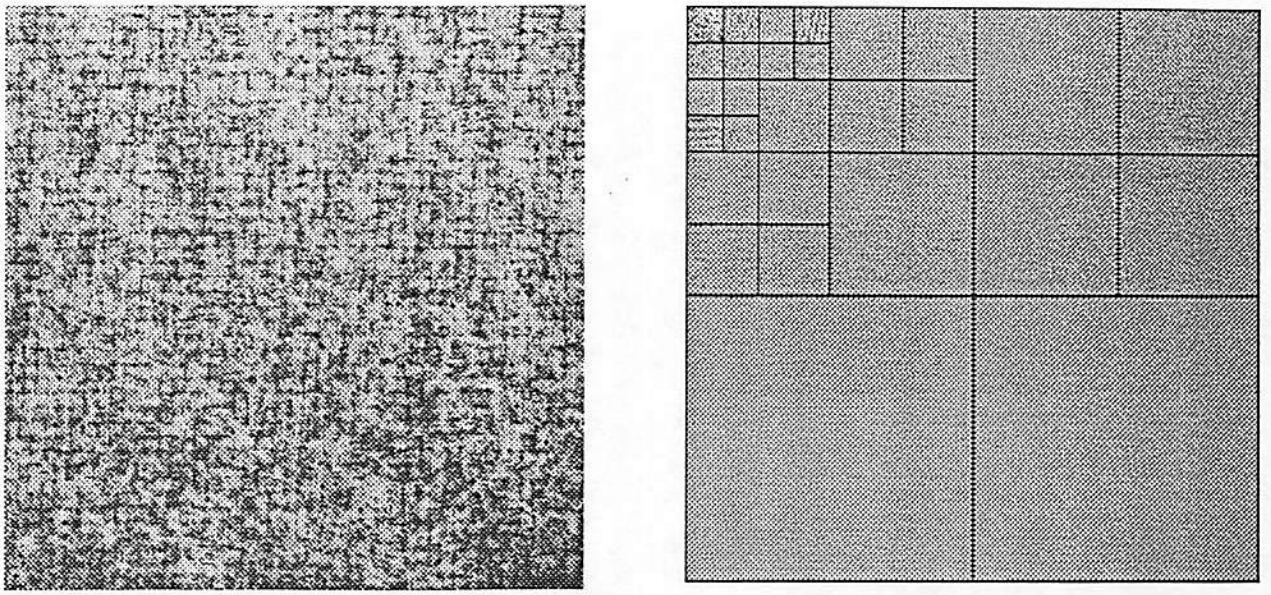

Figure 2: Block diagram of tree-structured wavelet decomposition.



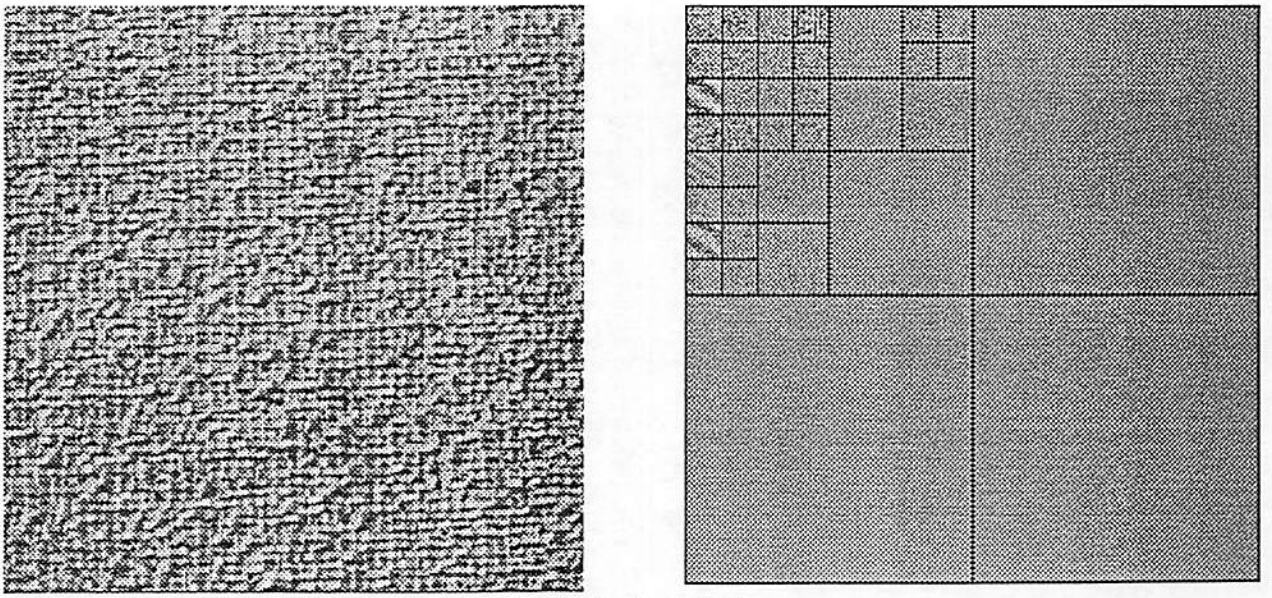
(a) French Canvas (D21)



(b) Oriental Straw Cloth (D53)

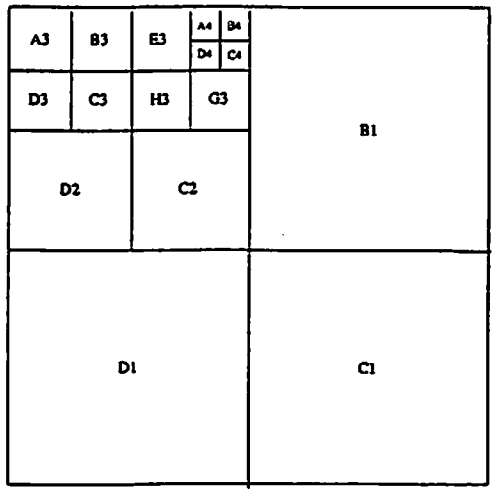


(c) Wool (D19)

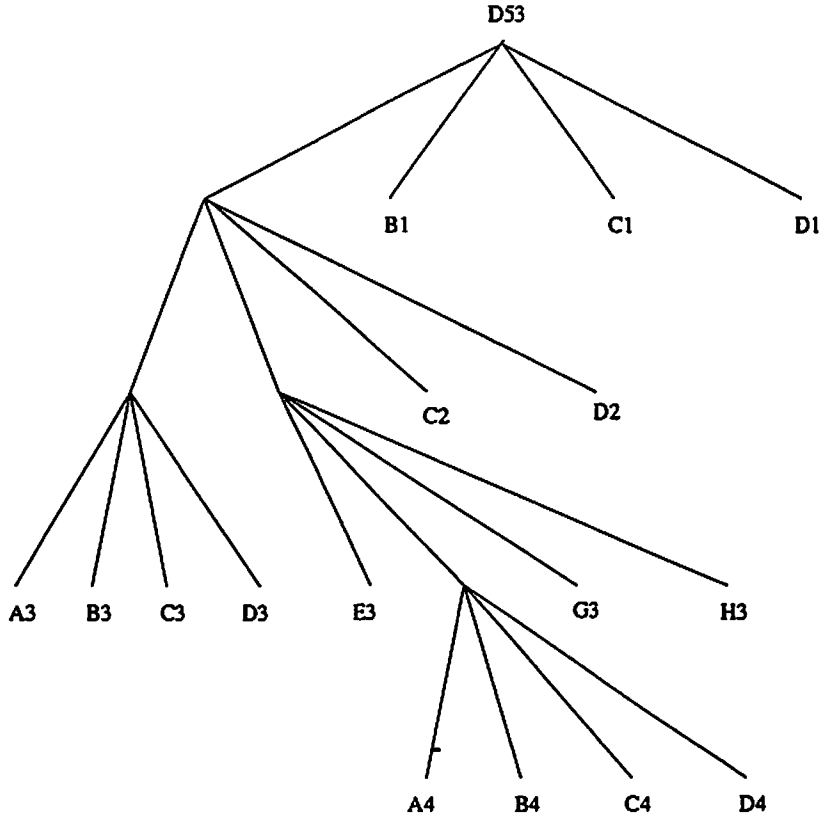


(d) Raffia (D84)

Figure 3: Tree-structured wavelet transforms of (a) French Canvas (D21), (b) Oriental Straw Cloth (D53), (c) Wool (D19) and (d) Raffia (D84).

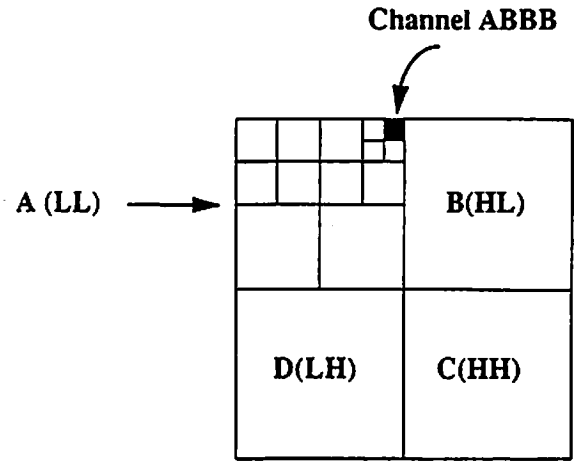



(a) channel decomposition via tree-structured wavelet transform

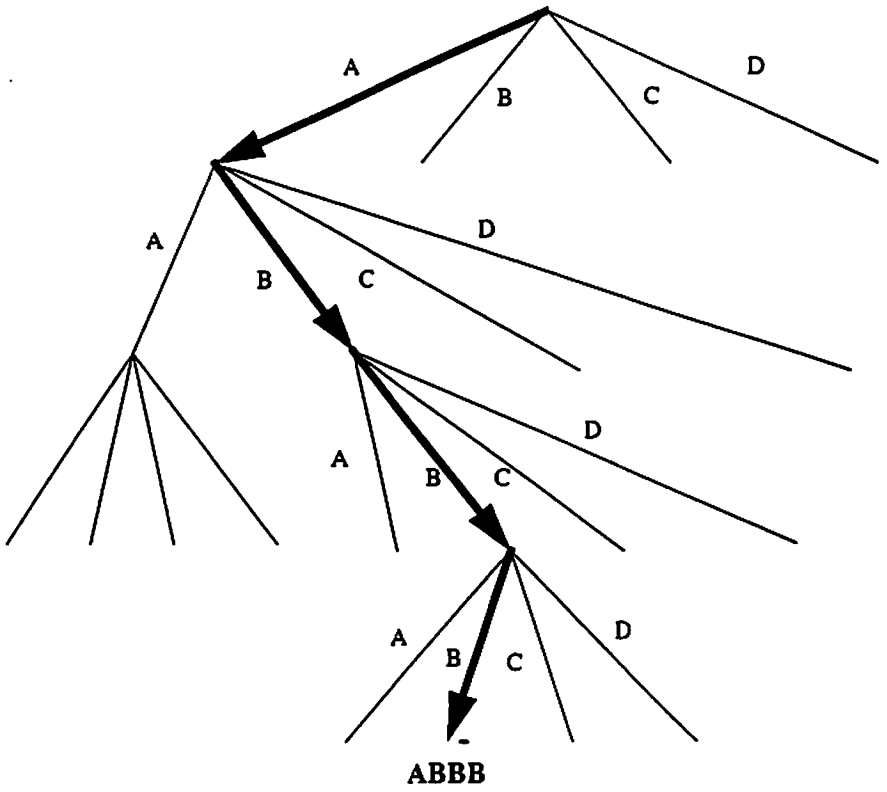



(b) quadtree representation

Figure 4: Energy map for Oriental Straw Cloth (D53).




(a) tree-structured wavelet transform domain




(b) quadtree representation

Figure 5: The meaning of Channel ABBB in (a) tree-structured wavelet transform domain and (b) quadtree representation.

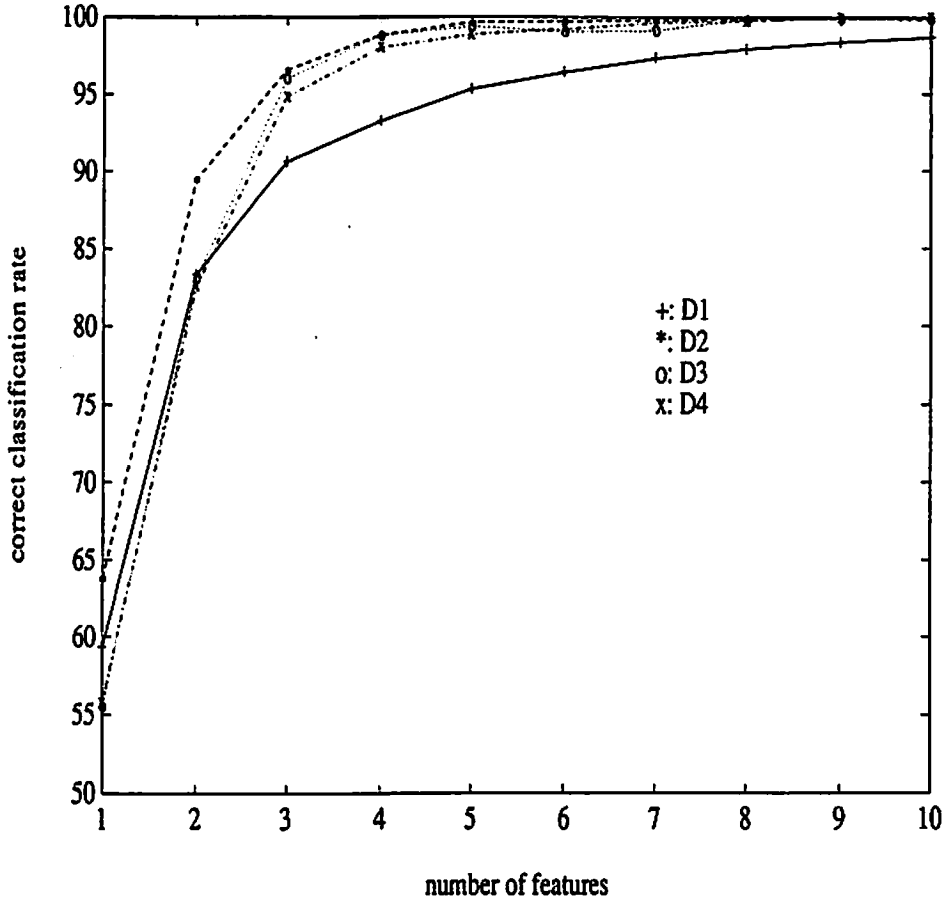
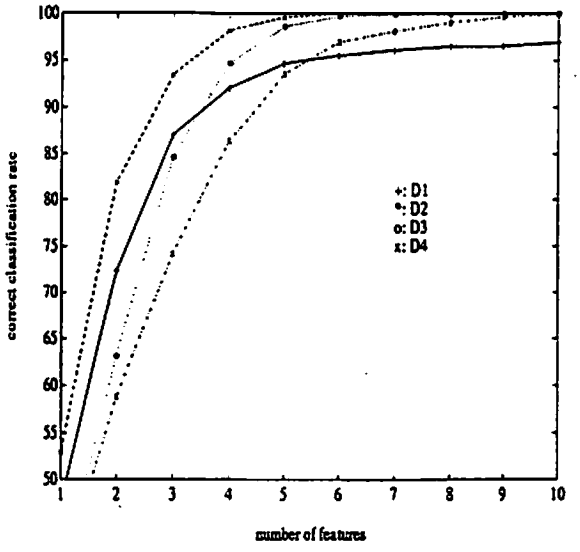
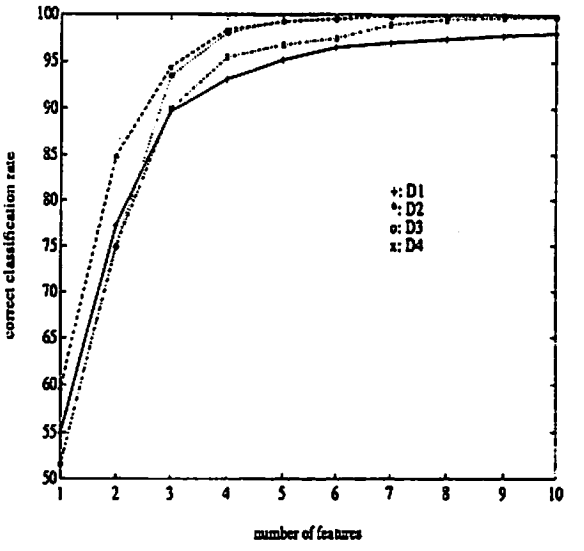


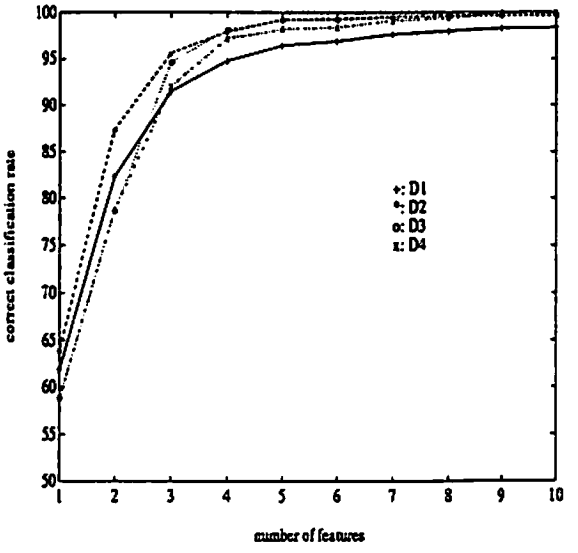

Figure 6: Correct classification rate for Battle-Lemarié cubic spline wavelet basis



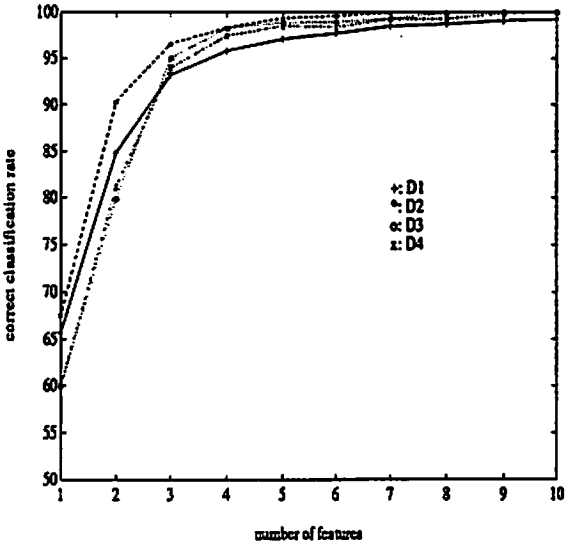

(a) Daubechies 2-tap




(b) Daubechies 4-tap

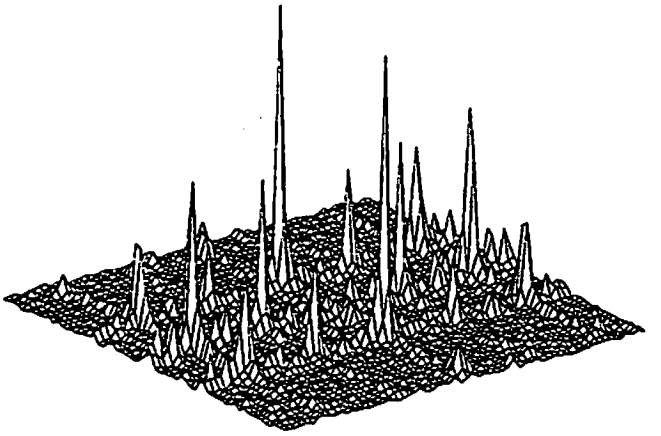



(c) Daubechies 8-tap

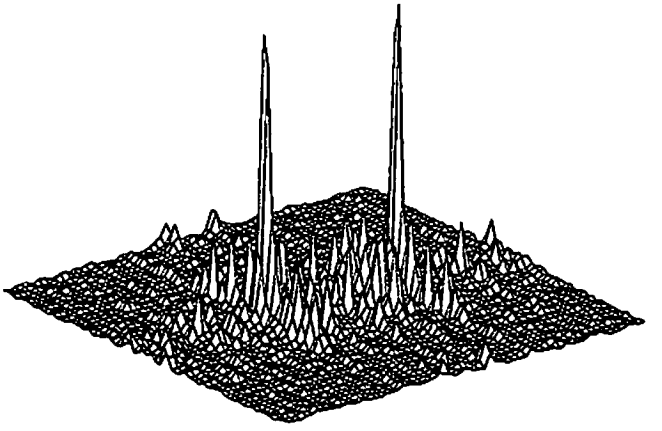



(d) Daubechies 16-tap

Figure 7: Correct classification rate for different Daubechies wavelet bases.



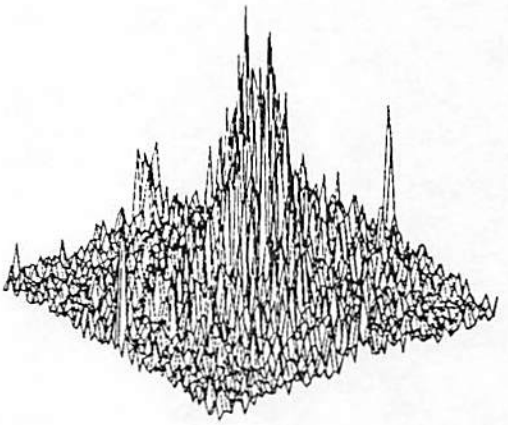
(a) French Canvas (D21)



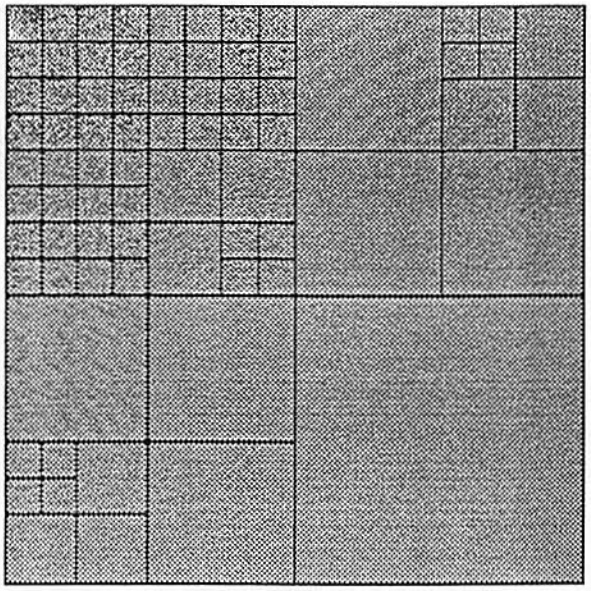
(b) Oriental Straw Cloth (D53)

Figure 8: Fourier power spectra for (a) French Canvas (D21) and (b) Oriental Straw Cloth (D53).

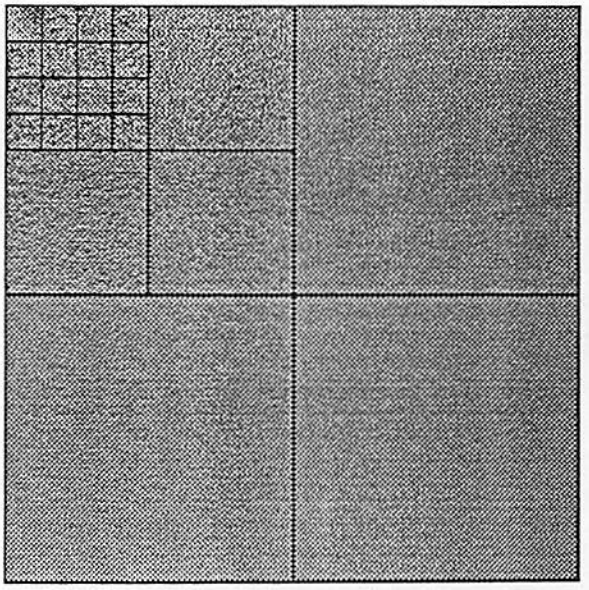
(a) Pressed Pork (D4)



(b) Fourier power spectrum



(c) Tree wavelet map with $C = 0.3$



(d) Tree wavelet map with $C = 0.6$

Figure 9: Pressed Cork (D4) and its Fourier power spectrum and tree-structured wavelet transform maps.