Chapter 1

Statistical Image Modeling and Processing Using Wavelet Domain Hidden Markov Models

GUOLIANG FAN

School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, Oklahoma
Email: glfan@okstate.edu

XIANG-GEN XIA

Department of Electrical and Computer Engineering
University of Delaware
Newark, Delaware
Email: xxia@ece.udel.edu

1.1 Introduction

Digital image processing has a broad spectrum of applications, and it has rapidly evolving fields with many theoretical as well as technological breakthroughs during the last decades. Generally speaking, there have been mainly two trends in the image processing research community: one leads to the fundamental image representation and modeling techniques, and the other heads a variety of image processing applications. The two topics are so closely related and have greatly been stimulated by each other. Both of them benefit from the widespread use of computers with powerful computational capability, and the development of human life science, multimedia, and Internet technologies. This Chapter will cover both topics, i.e., image modeling and image processing. An image model attempts to capture the key characteristics of an image based on which image processing problems can be formulated and solved mathematically and systematically. As prerequisites, image models often play important roles in many image processing applications.

Developing mathematical tools is the key to image modeling. Roughly speaking, image modeling techniques can be divided into two groups, i.e deterministic modeling and statistical (stochastic) modeling. A deterministic model predicts a single outcome from a given set of circumstances. On the contrary, a statistical model regards an image as a realization of a certain probability model, and predicts a set of possible outcomes weighted by their likelihoods or probabilities. Both image models have different strengths for characterizing image features and apply to different image processing problems. In this work, we are particularly interested in statistical image modeling and processing, both of which are conducted in the wavelet-domain instead of the spatial domain where images exist. Why do we choose the wavelet transform? It is mainly because that the discrete wavelet transform (DWT) can give us a more favorable representation which helps us to develop efficient image modeling and processing techniques. For example, we show both the Lena image and its 3-scale DWT in Fig. 1.1.

Figure 1.1: Lena (256 × 256, 8bpp) image and its 3-scale wavelet transform where the gray level corresponds to the magnitudes of wavelet coefficients.
From Fig. 1.1, we can see that the DWT of an image can provide a compact, joint spatial-frequency, and multiscale representation, which applies a variety of image processing applications. Firstly, the compact property of DWT indicates that most image energy can be compacted onto few wavelet coefficients with large magnitudes, and at the same time, most coefficients are very small. This compact property allows us to capture the key characteristics of an image from those large wavelet coefficients, and it is found very useful in image compression, enhancement, and restoration, etc. Secondly, the joint spatial-frequency representation results in two evident properties of wavelet coefficient distribution, i.e., interscale persistence and intrascale clustering. The former one means that the wavelet coefficient distribution has some similarity across scales, and the latter one indicates that wavelet coefficients of similar magnitudes are very likely to cluster together. These two observations inspire many researchers to develop various statistical models for image restoration, compression, and classification, etc. Thirdly, the multiscale representation of DWT is also useful in many image processing applications, such as progressive image transmission, embedded image compression, multi-resolutions image analysis, etc. Wavelets are mathematical tools with powerful structure and many varieties. The multiresolution structure of wavelets allows one to zoom in local signal behavior to analyze signal details, or zoom out to get a global view of a signal. Although the idea of multiresolution analysis goes back to early years, it was formally developed in the 1980’s in [1], and the construction of compactly supported wavelets in [2] further attracts the attention of the larger scientific community and stimulates tremendous research activity, in particular in the areas of signal and image processing, and statistical modeling [3]. It was shown in [4] that wavelet basis is the optimal basis for data compression, noise reduction, and statistical estimation.

Another important mathematical tool of statistical modeling is the hidden Markov model (HMM) which was developed in the 1960s. As a finite state machine, underlying an HMM is a basic Markov chain which can be characterized by state transition probabilities. In an HMM, at each unit of time a single observation is generated from the current state according to a probability distribution depending only on the state. Thus, in contrast to a Markov chain, since the observation is a random function of the state, it is not in general possible to determine the current state by simply examining the current observation. In other words, the state of HMM is hidden to the observer. In Fig. 1.2, we illustrate a general structure of an HMM, where the output of the Markov chain is distorted by a certain noise source.
There are many real-world signals that are amenable to the finite state mechanism in an HMM, e.g., speech signals. Thus HMMs have earned their popularity in large part from their successful application to speech recognition. However, HMMs cannot be applied to image modeling in the spatial domain directly due to the large number of intensity levels of image pixels. Recently, a marriage of DWT and HMM produced a new efficient mathematical tool for statistical modeling, namely wavelet-domain HMMs which were originally proposed in [5] and further developed for image modeling and processing in [6–9]. Wavelet-domain HMMs are found useful for statistical modeling due to the fact that the decorrelation of DWT greatly reduces the number of states of wavelet coefficients. Using probabilistic graphs, wavelet-domain HMMs can effectively characterize the joint statistics of wavelet coefficients of a given signal or image. They have been applied to image denoising [7], image segmentation [8], Bayesian analysis [6], and image retrieval [9].

In this Chapter, we study wavelet-domain HMMs regarding both statistical image modeling and the applications to various image processing problems. In particular, we investigate how wavelet-domain HMMs are tailored and trained for different applications in order to obtain the accurate and robust signal/image characterization, as well as how to develop efficient image processing algorithms to take advantages of wavelet-domains in different applications. The remainder of this chapter is organized into five sections as follows. We firstly review wavelet-domain HMMs for statistical modeling, and we also show some preliminary studies. Secondly, image denoising is discussed where a new wavelet-domain HMM is proposed which can outperform traditional wavelet-domain HMMs and most state-of-the-art denoising methods. Thirdly, multiscale Bayesian segmentation is studied where wavelet-domain HMMs are used to obtain statistical image characterization. Fourthly, Texture analysis and synthesis are investigated by using an improved wavelet-domain HMM for texture modeling which provides more accurate and complete texture characterization than traditional wavelet-domain HMMs. Finally, conclusions are given.

1.2 Wavelet-Domain Hidden Markov Models

The theory of hidden Markov models (HMMs) was originally developed in the 1960s in [10]. HMMs have earned their popularity mainly form the successful application to speech recognition. A recent review of the HMMs theory can be found from [11]. An HMM model has a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are “hidden” to the outside; hence the name Hidden Markov Model, as shown in Fig. 1.2. HMMs with states in a finite discrete set and uncertain parameters have been widely applied in areas such as communication systems, speech processing, and biological signal processing.
On the other hand, it is hard to apply HMMs to image processing directly in the spatial domain, since there are too many states (intensity levels of image pixels) to be handled by HMMs. Recently, a new framework of statistical signal and image modeling, called wavelet-domain hidden Markov models, has been recently developed in [5] and applied to signal and image processing in [6–9]. Wavelet-domain hidden Markov models (HMMs) are finite state machines in the wavelet-domain, and the decorrelation of the wavelet transform can greatly reduce the number of hidden states, making HMMs useful and manipulable in the wavelet-domain. In the following, we briefly review wavelet-domain HMMs in the 1-dimensional (1-D) case. For more details, we refer the reader to [5].

Given a bandpass wavelet function $\psi(t)$ and a lowpass scaling function $\phi(t)$, the discrete wavelet transform (DWT) represents a signal $s(t)$ of size $N$ in terms of shifted versions of $\phi(t)$ and shifted and dilated versions of $\psi(t)$, as shown in Fig. 1.3(a). The atoms of DWT are $\psi_{j,i}(t) \equiv 2^{-j/2}\psi(2^{-j}t - i)$, $\phi_{j,i}(t) \equiv 2^{-j/2}\phi(2^{-j}t - i)$, $j, i \in \{1, ..., J\}$, and the wavelet representation can be written as [12],

$$s(t) = \sum_{i=0}^{N_j-1} u_{j,i} \phi_{j,i}(t) + \sum_{j=1}^{J} \sum_{i=0}^{N_j-1} w_{j,i} \psi_{j,i}(t), \quad (1.1)$$

where $J$ denotes the scale of analysis, and scale $J$ indicates the coarsest scale or lowest resolution of analysis. $N_j = N/2^j$ is the number of coefficients in scale $j$. $u_{j,i} = \int s(t) \phi_{j,i}(t)dt$ is the scaling coefficient, which measures the local mean around the time $2^j i$. $w_{j,i} = \int s(t) \psi_{j,i}(t)dt$ is the wavelet coefficient which characterizes the local variation around the time $2^j i$ and the frequency $2^j f_0$. Due to the multiscale binary-tree structure, given a wavelet coefficient $w_{j,i}$, its parent is $w_{j+1, [i/2]}$, where the operation $[x]$ takes the integer part of $x$, and its two children are $w_{j-1, 2i}$ and $w_{j-1, 2i+1}$, as shown in Fig. 1.3(a). In what follows, we use $\mathbf{w}$ to denote the vector of all wavelet coefficients.

For most real-world signals, the set of wavelet coefficients is sparse. This means that the majority of the coefficients are small and only a few coefficients contain the most of the signal energy. Thus, the probability density function (pdf), $f_W(w)$, of the wavelet coefficients $w$ can be described by a peak (centered at $w = 0$) and heavy-tailed non-Gaussian density, where $W$ stands for the random variable of $w$. It was presented in [13] that Gaussian mixture model (GMM) can well approximate this non-Gaussian density, as shown in Fig. 1.3(b). Therefore, we associate each wavelet coefficient $w$ with a set of discrete hidden states $S = 0, 1, ..., M - 1$, which have probability mass functions (pmf), $p_S(m)$. Given $S = m$, the pdf of the coefficient $w$ is Gaussian with mean $\mu_m$ and variance $\sigma_m^2$. We can parameterize a M-state GMM by $\pi = \{p_S(m), \mu_m, \sigma_m^2 | m = 0, 1, ..., M - 1\}$, and the overall pdf of $w$ is determined by

$$f_W(w) = \sum_{m=0}^{M-1} p_S(m) f_W|S(w|S = m), \quad (1.2)$$

where

$$f_W|S(w|S = m) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp \left(-\frac{(w - \mu_m)^2}{2\sigma_m^2}\right) = g(w; \mu_m, \sigma_m^2). \quad (1.3)$$
Figure 1.3: (a) Tiling of the time-frequency plane of DWT. The solid dot at the center corresponds to the scaling coefficients or wavelet coefficients. The tree structure is shown by the link of the dotted lines. (b) The histogram of the 1-scale DWT (Daubechies-4) of the “fruit” image where a 2-state zero-mean Gaussian Mixture Model can closely fit the real DWT data [5]. (c) Wavelet-domain hidden Markov tree (HMT) where the white node represents the state variable \( S \) and the black node denotes the wavelet coefficient \( W \).

Although \( w \) is conditionally Gaussian given its state \( S = m \), it is not Gaussian in general due to the randomness of the state variable \( S \).

Though the orthogonal DWT can decorrelate an image with almost uncorrelated wavelet coefficients, it is widely understood that there are considerable amount of high-order dependencies existing in \( w \). This can be observed from the characteristics of the wavelet coefficient distribution, such as *intrascalar clustering* and *interscale persistence*, as shown in Fig. 1.1. Therefore, in [5], a tree-structured hidden Markov Tree (HMT) model was developed by connecting state variables of wavelet coefficients vertically across the scale, as shown in Fig. 1.3(c), where we can see that HMT is able to capture the underlying interscale dependencies between parent and child state variables, which the second order statistics cannot provide. In HMT, each coefficient \( W_{j,i} \) is conditionally independent of all other random variables given its state \( S_{j,i} \). Thus an \( M \)-state HMT is parameterized by

- \( p_{S_j}(m) \): the pmf of the root node \( S_j \) with \( m = 0, 1, ..., M - 1 \);
- \( e_{j,j+1}^{m,n} = p_{S_j | S_{j+1}}(m | S_{j+1} [i/2] = n) \): the transition probability that \( S_{j+1} \) is in state \( m \) given that \( S_{j+1} [i/2] \) is in state \( n \), \( j = 1, ..., J - 1 \) and \( m, n = 0, 1, ..., M - 1 \);
- \( \mu_{j,m} \) and \( \gamma_{j,m}^2 \): the mean and variance, respectively, of \( W_{j,i} \) given that \( S_{j,i} \) is in state \( m \), \( j = 1, ..., J \) and \( m = 0, 1, ..., M - 1 \).

These parameters can be grouped into a model parameter vector \( \theta \) as

\[
\theta = \{ p_{S_j}(m), e_{j,j+1}^{m,n}, \mu_{j,m}, \gamma_{j,m}^2 | j = 1, ..., J; n, m = 0, ..., M - 1 \}. \tag{1.4}
\]
The accurate estimation of HMT model parameters is essential to its practical applications, which can be effectively approached by the iterative Expectation Maximization (EM) algorithm [14] that is well known to numerically approximate maximum likelihood estimates for mixture density problems. The EM algorithm has a basic structure and the implementation steps are problem dependent. The EM algorithm for HMT model training will be presented briefly here, and we refer the reader to [5] for more details. In the case of the HMT model training using the EM algorithm, we try to fit an $M$-state HMT model $\theta$ defined in (1.4) to the observed $J$-scale tree-structured DWT, i.e., $w$. The iterative structure is shown as follows:

- **Step 1. Initialization:** Set an initial model estimate $\theta^0$, and iteration counter $l = 0$.

- **Step 2. E step:** Calculate $p(S|w, \theta^l)$, which is the joint pmf for the hidden state variables and used in the maximization of $E_S[\ln f(w, S|\theta)|w, \theta^l]$.

- **Step 3. M step:** Set $\theta^{l+1} = \arg \max_\theta E_S[\ln f(w, S|\theta)|w, \theta^l]$.

- **Step 4. Iteration:** Set $l = l + 1$. If it converges, then stop; else, return to Step 2.

In [5, 15], the wavelet-domain HMMs have been applied to signal estimation, detection, and synthesis. Specifically, an “empirical” Bayesian approach was developed to denoise a signal corrupted by additive white Gaussian noise (AWGN). It was demonstrated that signal denoising using wavelet-domain HMT outperformed other traditional wavelet-based signal denoising methods with well-preserved detailed structures. Given a noisy signal of the noise power $\sigma^2$, the HMT model $\theta$ is firstly obtained via the EM training when we can also estimate the posterior hidden state probabilities $p(S_{j,i}|w, \theta)$ for each wavelet coefficient $W_{j,i}$. Then we can obtain the conditional mean estimate for $Y_{j,i}$ by the chain rule for the conditional expectation:

$$E[Y_{j,i}|w, \theta] = \sum_{m=0}^{M-1} p(S_{j,i} = m|w, \theta) \times \frac{\gamma_{j,m}^2 - \sigma^2}{\gamma_{j,m}^2} w_{j,i}. \quad (1.5)$$

The denoised signal is achieved by the inverse DWT (IDWT) of these estimates of wavelet coefficients.

In the following, we show two preliminary studies regarding the application of wavelet-domain HMMs to signal modeling and processing, i.e., training efficiency and modeling accuracy [16].

### 1.2.1 Initialization of EM Algorithm

It was mentioned that the “intelligent” initialization of the EM algorithm may provide the fast convergence of the HMT model training, but few comments on how to achieve an effective initial setting were given in [5]. We here propose an initialization scheme for the EM training algorithm which allows the efficient HMT model learning process. Given a $J$-scale DWT $w$, the two steps in the initialization are referred to as *horizontal scanning* and *vertical counting*. Specifically, the former step estimates the initial settings of the GMMs in different scales, and the latter step determines the initial transition probabilities $\gamma_{j,m}^{m,n}$.
**Horizontal Scanning:** It was assumed in [5] that the wavelet coefficients in the same scale have the same density. Therefore, the wavelet coefficients can be grouped into different categories according to their scales, and each group can be characterized by a 2-state GMM. The task of horizontal scanning is to fit a 2-state GMM parameterized by \( \pi_j = \{ p_{S_j}(m), \mu_{j,m} = 0, \gamma_{j,m}^2 | m = 0, 1 \} \) to \( w_j \), the wavelet coefficients in scale \( j \). We develop an EM algorithm [17] to implement the horizontal scanning. Given \( w_j \), we want to estimate the GMM \( \pi_j \) that maximizes the likelihood \( E[\ln f(w_j | \pi_j)] | w_j, \pi_j \) where \( f(w_j | \pi_j) \) is

\[
f(w_j | \pi_j) = \prod_{i=0}^{N_j-1} f(w_{j,i} | \pi_j), \tag{1.6}
\]

\( f(w_{j,i} | \pi_j) \) is given in (1.3). We start the horizontal scanning from a neutral initial setting, \( \pi_0^j \), which has equal probabilities of \( p_{S_j}(0) = p_{S_j}(1) = 0.5 \). \( w_j \) can be divided into two groups of the same number, \( N_j/2 \), according to their magnitudes. We use the variances of two groups as the initial values of the mixture variances, i.e., \( \gamma_{j,0}^2 \) and \( \gamma_{j,1}^2 \). After we determine \( \pi_0^j \), the horizontal scanning is performed as follows.

- **Step 1. Initialization:** Set \( \pi_0^j \) and the iteration counter \( c = 0 \).

- **Step 2. E step:** Calculate \( p(S_{j,i} | w_{j,i}, \pi_j) \) that is the conditional pmf of \( S_{j,i} \)

\[
p(S_{j,i} = m | w_{j,i}, \pi_j) = \frac{p_{S_j}(m) g(w_{j,i}; 0, \gamma_{j,m}^2)}{\sum_{n=0}^{N_j} p_{S_j}(n) g(w_{j,i}; 0, \gamma_{j,n}^2)}. \tag{1.7}
\]

- **Step 3. M step:** Set \( \pi_j^{c+1} = \arg \max_{\pi_j} E[\ln f(w_j, S_j | \pi_j) | w_j, \pi_j^c] \). We update the entries of \( \pi_j^{c+1} \) as

\[
p_{S_j}(m) = \frac{1}{N_j} \sum_{i=0}^{N_j-1} p(S_{j,i} = m | w_{j,i}, \pi_j) \tag{1.8}
\]

\[
\gamma_{j,m}^2 = \frac{\sum_{i=0}^{N_j-1} w_{j,i}^2 p(S_{j,i} = m | w_{j,i}, \pi_j)}{p_{S_j}(m) N_j} \tag{1.9}
\]

- **Step 4. Iteration:** Set \( c = c + 1 \). If it converges, then stop; else, return to Step 2.

The extra computational cost introduced by the horizontal scanning is not significant compared with the computational complexity of the HMT model training.

**Vertical Counting:** After estimating the initial GMM at each scale, the following vertical counting step is used to estimate the initial transition probabilities \( \pi_{j,i}^{m,n} \) between two neighboring scales. Given \( \pi_j \) and \( w_{j,i} \), we can determine the initial hidden state of \( w_{j,i}, S_{j,i} \), based on the maximum likelihood criteria as:

\[
S_{j,i} = \begin{cases} 
0 & \text{if } |w_{j,i}| < T_j \\
1 & \text{otherwise}
\end{cases} \tag{1.10}
\]

where \( T_j = \sqrt{\frac{\gamma_{j,0}^2 \gamma_{j,1}^2 \ln \gamma_{j,0}^2 - \ln \gamma_{j,1}^2}{\gamma_{j,1}^2 \gamma_{j,0}^2}} \). Given the initial states, we count state transition frequencies between every two neighboring scales and along the tree-structured wavelet coefficient set. Then we set the initial values of the transition probabilities \( \pi_{j,i}^{m,n} \) by the normalized transition frequencies. We average the
transition probabilities of different scales in order to achieve a robust estimation:

$$e_{j,j+1}^{m,n} = \frac{\sum_{j=1}^{J} \# \{ S_{j,i} = m \text{ and } S_{j+1,\lfloor i/2 \rfloor} = n \lfloor i = 0, ..., N_j - 1 \} \# \{ S_{j+1, \lfloor i/2 \rfloor} = n \lfloor i = 0, ..., N_j - 1 \} - 1}{J - 1},$$  \hspace{1cm} (1.11)$$

where $\#(A = B)$ denotes the number of the event $A = B$ occurring. From our numerous simulation results, the above scale independent initial values by averaging the probabilities across different scales outperforms the scale dependent ones without averaging. This averaging operation may not be necessary when we have enough data during the model training, such as in image processing. The above scheme can efficiently characterize the initial marginal and joint statistics of $w, \theta$, which allows an efficient EM model training.

### 1.2.2 Improved HMT-2 Model

In order to capture more cross-correlation of wavelet coefficients between two neighboring scales, a new HMM, HMT-2, is developed and shown in Fig. 1.4(a), where the state of the wavelet coefficient $w$ depends not only on the state of its parent node, but also on the state of the twin of its parent. This strategy is very popular in the graphical modeling and Bayesian network literatures [18]. Two reasons for this consideration are (i) the local stationary property of most signals and the correlation of the wavelet functions in two neighboring scales, and (ii) the wavelet filter bank decompositions, where the filter length may be long enough for the coefficients to have more interscale dependencies. Usually, the analysis and the training of more complicated HMMs become more difficult [19]. A simplified interpretation of HMT-2 is illustrated in Fig. 1.4(b) where two coefficients are integrated into one node. Actually, HMT-2 is operated in the same way as the HMT model in [5] except for the number of the hidden states associated with each node. If we assume two hidden states for each coefficient, i.e., 0 and 1, each node of HMT-2 will have four states: 00, 01, 10, and 11. We call our new model as HMT-2 instead of four-state HMT in order to distinguish it from the original $M$-state HMT when $M = 4$. We also develop the EM training algorithm for HMT-2 based on the one in [5]. It is worthwhile noting that the initialization of the HMT-2 model training is operated in the same way as the one of the 2-state HMT except for the state combination of two coefficients in one node after the horizontal scanning step.
1.2.3 Simulation Results

The proposed two-step initialization technique and the new HMT-2 model are examined here based on experiments on a set of test signals. For comparison, a neutral initial setting, where all probabilities are evenly distributed, is also studied. Our simulation is conducted on Donoho’s length-1024 test signals: Bumps, Blocks, Doppler, and Heavisine [20] and the detailed experiment setup can be found from Table 1.1. A comparison is made between two different initialization schemes, the neutral initial setting and the proposed method in terms of the EM convergence rate, as shown in Fig. 1.5. We can easily find that the proposed initialization scheme can accelerate the convergence rate for all test signals. The fast convergence rates of EM iterations show the effectiveness of the new initialization method, and the similar model likelihoods indicate the similar denoising performances, i.e., the mean square error (MSE).

Table 1.1: Signal denoising results using HMT and HMT-2, where the MSEs were averaged over 1000 trials, and the EM training is initialized by (a) the neutral setting and (b) the proposed two-step scheme.

<table>
<thead>
<tr>
<th></th>
<th>Doppler</th>
<th>Bumps</th>
<th>Blocks</th>
<th>Heavisine</th>
<th>Doppler</th>
<th>Bumps</th>
<th>Blocks</th>
<th>Heavisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMT (a)</td>
<td>0.139</td>
<td>0.313</td>
<td>0.298</td>
<td>0.098</td>
<td>0.270</td>
<td>0.560</td>
<td>0.663</td>
<td>0.161</td>
</tr>
<tr>
<td>HMT (b)</td>
<td>0.129</td>
<td>0.306</td>
<td>0.290</td>
<td>0.089</td>
<td>0.262</td>
<td>0.553</td>
<td>0.518</td>
<td>0.154</td>
</tr>
<tr>
<td>HMT-2(a)</td>
<td>0.175</td>
<td>0.470</td>
<td>0.463</td>
<td>0.112</td>
<td>0.331</td>
<td>0.858</td>
<td>0.834</td>
<td>0.164</td>
</tr>
<tr>
<td>HMT-2(b)</td>
<td>0.120</td>
<td>0.292</td>
<td>0.273</td>
<td>0.080</td>
<td>0.241</td>
<td>0.569</td>
<td>0.529</td>
<td>0.142</td>
</tr>
</tbody>
</table>

Since the proposed HMT-2 captures more interscale dependencies of wavelet coefficients and particularly applies to the DWT with long filters, we use the Symmlet-8 DWT for four signals. In Fig. 1.6 we show the HMT-2 model training results using two different initialization schemes, and we also compare the two models in terms of signal denoising with different noise powers in Table 1.1. From Fig. 1.6, we find that the EM convergence rate of HMT-2 is very fast (usually several steps) with both initialization schemes. This may be due to the fact that the complicated dependency structure involved in HMT-2 facilitates the wavelet-domain Bayesian inference with more efficient EM training process. From Table 1.1, we also see that the initialization is essential to the EM training of HMT-2, i.e., the last denoising performance. Compared with 2-state HMT, 4-state HMT-2 also improves denoising performance for most cases.

In this section, we have briefly introduced and improved wavelet-domain HMMs originally developed in [5]. This section serves the background materials of this Chapter. Meanwhile, these preliminary studies will lead to more powerful wavelet-domain HMMs as well as image processing algorithms afterwards. Particularly, we will study image denoising, image segmentation, and texture analysis and synthesis, where wavelet-domain HMM are discussed and tailored for different applications.
Figure 1.5: The plots of $\ln(f(w|\theta))$ with respect to the EM iteration number. Four noisy signals $\sigma^2_{\theta_i} = 1.0$ are tested, where the proposed initialization scheme (solid line: 1) and the neutral initial setting (dotted line: 2) are used, respectively. (a) Doppler (1:MSE=0.142, 2:MSE=0.148). (b) Bumps (1:MSE=0.261, 2:MSE=0.261). (b) Blocks (1:MSE=0.083, 2:MSE=0.083). (d) Heavisine (1:MSE=0.071, 2:MSE=0.088).

Figure 1.6: The plots of $\ln(f(w|\theta))$ with respect to the EM iteration number. Four noisy signals $\sigma^2_{\theta_i} = 1.0$ are tested, where the proposed initialization scheme (solid line: 1) and the neutral initial setting (dotted line: 2) are used, respectively. (a) Doppler (1:MSE=0.111, 2:MSE=0.163). (b) Bumps (1:MSE=0.306, 2:MSE=0.532). (b) Blocks (1:MSE=0.256, 2:MSE=0.468). (d) Heavisine (1:MSE=0.070, 2:MSE=0.095).
1.3 Image Denoising

Wavelet-domain statistical image modeling can be roughly categorized into three groups: the interscale models, e.g., [5, 7, 21], the intrascale models, e.g., [22, 23], and the hybrid inter and intrascale models, e.g., [15, 24–26]. These models allow more accurate image modeling and more effective image processing, e.g., denoising and estimation, than other methods which assume wavelet coefficients to be independent. Particularly, [5, 7] wavelet-domain HMT imposes a tree-structured Markov chains capture interscale dependencies of wavelet coefficients. In [15], both interscale and intrascale dependencies can be efficiently captured by a so-called contextual hidden Markov model (CHMM). However, the local statistics of the wavelet coefficients cannot be well characterized by HMT and CHMM. In other words, neither HMT nor CHMM has sufficient spatial adaptability that is found useful in the statistical modeling of image wavelet coefficients [22]. Secondly, though the tree structure involved in HMT facilitates statistical modeling by capturing the key characteristics of DWT along the hierarchical wavelet subtree, it also introduces denoising artifacts in the denoised image due to the artificial structure. Thirdly, the tree-structured EM training algorithm of HMT is computationally expensive. In this work, we propose a new wavelet-domain HMM by considering above three issues, i.e., spatial adaptability, reduced denoising artifacts, and fast model training [27].

1.3.1 Gaussian Mixture Field

Given the J-scale DWT of an N × N image, \( w_{j,k,i} \) denotes the \((k,i)\)th coefficient in scale \( j \) where we omit the subband notation, \( j = 1, ..., J \) and \( k, i = 0, 1, ..., N_j - 1 \) with \( N_j = N / 2^j \). \( W_{j,k,i} \) and \( S_{j,k,i} \) are the continuous random variable and the discrete state variable of \( w_{j,k,i} \), respectively. In [5, 7, 15], the GMM, \( \Pi_j = \{ p_{S_j}(m), \sigma^2_{j,m} | m = 0, 1 \} \), is assumed for the wavelet coefficients in scale \( j \), and \( S_j \) is the state variable associated with scale \( j \). Sufficient data in scale \( j \) allows the robust estimation of \( \Pi_j \) at the loss of spatial adaptability of statistical image modeling in the wavelet-domain. In this work, we propose Gaussian Mixture Field (GMF) which can be thought of as an extension of the GMM. GMF assumes that each wavelet coefficient, \( w_{j,k,i} \), follows a local GMM parameterized by \( \Pi_{j,k,i} = \{ p_{S_{j,k,i}}(m), \sigma^2_{j,k,i,m} | m = 0, 1 \} \). \( \Pi_{j,k,i} \) can be estimated by the neighborhood of \( w_{j,k,i} \), \( \Omega_{j,k,i} \), which is selected by a square window of \( 2C_j + 1 \) and centered at \( w_{j,k,i} \), as shown in Fig. 1.7(a), i.e., \( \Omega_{j,k,i} = \{ w_{j,x,y} | x = k-C_j, ..., k+C_j; y = i-C_j, ..., i+C_j \} \). GMF is a highly localized model to exploit the local statistics of the wavelet coefficients. In particular, it applies to images where the non-stationary properties are prominent.

1.3.2 Local Contextual Hidden Markov Model

In addition to GMF, we use a context model to capture intrascale dependencies of wavelet coefficient, as shown in Fig. 1.7(b). We define the random context variable of \( W_{j,k,i} \) by \( V_{j,k,i} \) whose value is \( v_{j,k,i} = 1 \) if
the similar idea in the previous section. Given the additive white Gaussian noise (AWGN) of variance 

$$\mathcal{N}_j$$

where

$$\mathcal{A}_l$$

contextual hidden Markov model (LCHMM) for

$$\mathcal{C}_n$$

part of two neighboring local models, and the black and white parts are the distinct parts.

The illustration of the fast implementation of the LCHMM training, where the gray part is the overlapped

$$\mathcal{D}_p$$

wavelet coefficient conditioning on its context value. The EM training algorithm can be developed from the

$$\mathcal{E}_q$$

/DB

/AR

/A2

may not be robust. In this work, we can solve this problem by providing a good initial setting of

$$\mathcal{F}_r$$

where

$$\mathcal{G}_t$$


The LCHMM training is performed as follows, where

$$\mathcal{H}_u$$

Step 1. Initialization:

(1.0) \( \Pi_j^0 \) = \( \{ \pi_j(0) = p_S(1) = 0.5, \sigma_{j,0}^2 = \sigma_j^2, \sigma_{j,1}^2 = 2\delta_j^2 - \sigma_j^2 \} \) and set \( p = 0 \).

(1.1) E step: Given \( \Pi_j^p \), calculate (Bayes rule)

$$p(S_{j,k,i} = m | w_{j,k,i}; \Pi_j^p) = \frac{p_S(m) g(w_{j,k,i}; 0, \sigma_j^2(m))}{\sum_{m=0}^{\mathcal{I}_j} p_S(m) g(w_{j,k,i}; 0, \sigma_j^2(m))}.$$  (1.14)
(1.2) M step: Compute the elements of \( \Pi_j^{p+1} \) by
\[
p_{S_j}(m) = \sum_{k=0}^{N_k-1} \sum_{i=0}^{N_j-1} p(S_{j,k,i} = m | w_{j,k,i}, \Pi_j^p), \tag{1.15}
\]
\[
\sigma_{j,m}^2 = \frac{\sum_{k=0}^{N_k-1} \sum_{i=0}^{N_j-1} w_{j,k,i}^2 p(S_{j,k,i} = m | w_{j,k,i}, \Pi_j^p)}{N_j^2 p_{S_j}(m)}. \tag{1.16}
\]

(1.3) Iteration: Set \( p = p + 1 \). If it converges (or \( p = N_p \)), then go to Step 1.4; else go to Step 1.1.

(1.4) Set \( c = 0 \) and set the elements in \( \Theta_{j,k,i}^0 \) by
\[
p_{S_{j,k,i}}(m) = \sum_x \sum_y p(S_{j,x,y} = m | w_{j,x,y}, \Pi_j), \tag{1.17}
\]
\[
\sigma_{j,k,i,m}^2 = \frac{\sum_x \sum_y w_{j,x,y}^2 p(S_{j,x,y} = m | w_{j,x,y}, \Pi_j)}{(2C_j + 1)^2 p_{S_{j,k,i}}(m)}. \tag{1.18}
\]
\[
p_{v_{j,k,i}|S_{j,k,i}}(v|m) = \frac{\sum_{m=0}^{N_k} p_{S_{j,k,i}}(m)p_{v_{j,k,i}|S_{j,k,i}}(v|m)g(w_{j,k,i}|0, \sigma_{j,k,i,m}^2)}{\sum_{m=0}^{N_k} p_{S_{j,k,i}}(m)p_{v_{j,k,i}|S_{j,k,i}}(v|m)g(w_{j,k,i}|0, \sigma_{j,k,i,m}^2)}. \tag{1.19}
\]

- **Step 2. E step:** Given \( \Theta_{j,k,i}^c, j, k = 0, 1, ..., N_j - 1 \), calculate (Bayes rule)
\[
p_{S_{j,k,i}|V_{j,k,i},W_{j,k,i}}(m|w_{j,k,i}, v_{j,k,i} = v) = \frac{p_{S_{j,k,i}}(m)p_{v_{j,k,i}|S_{j,k,i}}(v|m)g(w_{j,k,i}|0, \sigma_{j,k,i,m}^2)}{\sum_{m=0}^{N_k} p_{S_{j,k,i}}(m)p_{v_{j,k,i}|S_{j,k,i}}(v|m)g(w_{j,k,i}|0, \sigma_{j,k,i,m}^2)). \tag{1.20}
\]

- **Step 3. M step:** Compute the elements of \( \Theta_{j,k,i}^{c+1} \), \( j, k = 0, 1, ..., N_j - 1 \), by
\[
p_{S_{j,k,i}}(m) = \sum_x \sum_y p_{S_{j,x,y}|V_{j,x,y},W_{j,x,y}}(m|w_{j,x,y}, v_{j,x,y}), \tag{1.21}
\]
\[
\sigma_{j,k,i,m}^2 = \frac{\sum_x \sum_y w_{j,x,y}^2 p_{S_{j,x,y}|V_{j,x,y},W_{j,x,y}}(m|w_{j,x,y}, v_{j,x,y})}{(2C_j + 1)^2 p_{S_{j,k,i}}(m)}. \tag{1.22}
\]
\[
p_{v_{j,k,i}|S_{j,k,i}}(v|m) = \frac{\sum_{m=0}^{N_k} p_{S_{j,k,i}}(m)p_{v_{j,k,i}|S_{j,k,i}}(v|m)g(w_{j,k,i}|0, \sigma_{j,k,i,m}^2)}{p_{S_{j,k,i}}(m)}. \tag{1.23}
\]

- **Step 4. Iteration:** Set \( c = c + 1 \). If it converges (or \( c = N_c \)), then stop; else go to Step 2.

### 1.3.3 Fast EM Model Training

It seems that the LCHMM training is computationally expensive, since Step 3 (M step) is performed on each wavelet coefficient. As a matter of fact, it is easy to notice that there are many overlapped computations in Step 3, as shown in Fig. 1.7(c). The actual computational complexity of the LCHMM training is slightly higher than that of CHMM in [15] and lower than those of HMMs in [5, 7]. Given \( \Theta_{j,k,i} \), we can estimate the noise-free \( Y_{j,k,i} \) from \( W_{j,k,i} \) as the conditional mean as
\[
E[Y_{j,k,i}|w_{j,k,i}, v_{j,k,i}] = \frac{1}{m=0} p_{S_{j,k,i}|V_{j,k,i},W_{j,k,i}}(m|w_{j,k,i}, v_{j,k,i}) \frac{\sigma_{j,k,i,m}^2}{\sigma_{j,k,i,m}^2 + \sigma_{j,m}^2} w_{j,k,i} \tag{1.24}
\]

The denoised image is the IDWT of the above estimates of wavelet coefficients. We expect the proposed LCHMM has the spatial adaptability, reduced denoising artifacts, and fast model training process.
1.3.4 Simplified Shift-Invariant Denoising

The lack of shift-invariant property of the orthogonal DWT results in the visually disturbing artifacts in denoised images. The “Cycle-spinning” technique was proposed in [28] to solve this problem, where signal denoising is applied to all shifts of the noisy signal, and the denoised results are then averaged. It can be shown that shift-invariant image denoising is equivalent to image denoising based on redundant wavelet transforms, such as those used in [7, 24, 29]. In this work, we consider the 16 shifted versions which are obtained from shifting the noisy image by 1, 2, 3, and 4 pixels in each dimension, respectively. This simplification was found sufficient for most images in practice. We assume that the LCHMM parameters are the same as those of the 16 shifted versions. Therefore, the EM training is performed only once, and the LCHMM training results are applied to the 16 images for denoising.

1.3.5 Simulation Results

We apply LCHMM to image denoising for real images Barbara and Lena (8bpp, 512 × 512) with AWGN of known variance $\sigma^2_n$. The experimental setting is given as follows: (1) the window size of the local GMM in
Table 1.2: PSNR (dB) results from several recent denoising algorithms.

<table>
<thead>
<tr>
<th>Noisy Images</th>
<th>Lena</th>
<th>Barbara</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_0$</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Denoising Methods</td>
<td>Orthogonal DWT</td>
<td>Orthogonal DWT</td>
</tr>
<tr>
<td>Donoho’s HT (D8) [20]</td>
<td>31.6</td>
<td>29.8</td>
</tr>
<tr>
<td>Wiener (Matlab)</td>
<td>32.7</td>
<td>31.3</td>
</tr>
<tr>
<td>HMT (D8) [7]</td>
<td>33.9</td>
<td>31.8</td>
</tr>
<tr>
<td>SAWT (S8) [24]</td>
<td>-</td>
<td>31.8</td>
</tr>
<tr>
<td>LAWMAP (D8) [22]</td>
<td>34.3</td>
<td>32.4</td>
</tr>
<tr>
<td>AHMF (D8) [23]</td>
<td>34.5</td>
<td>32.5</td>
</tr>
<tr>
<td>SSM (D8) [25]</td>
<td>34.8</td>
<td>32.5</td>
</tr>
<tr>
<td>LCHMM (D8)</td>
<td>34.4</td>
<td>32.4</td>
</tr>
<tr>
<td>LCHMM (S8)</td>
<td>34.5</td>
<td>32.6</td>
</tr>
</tbody>
</table>

GMF decreases with the increase of the scale in order to adapt to the higher variations of wavelet coefficients in coarser scales, and in practice, $\{C_j = 6 - j | j = 1, 2, 3, 4, 5\}$ are found both effective and efficient; (2) for simplicity, we also fix the iteration numbers of the initialization step and the EM training step to be $N_p = 20$ and $N_c = 5$; (3) we use the 5-scale DWT where two wavelets, Daubechies-8 (D8) and Symmlet-8 (S8), are tested; (4) the DWT is used with two setups: the orthogonal DWT and the redundant DWT or shift-invariant (SI) techniques. The PSNR results are shown in Table 1.2 where several recent image denoising algorithms are compared. It is shown that LCHMM provides the excellent denoising performance for the two images, especially for the Barbara image where the non-stationarity property is prominent. LCHMM outperforms all the other methods in most cases. We also show the visual quality of image denoising (with D8 wavelet) in Fig. 1.8 where LCHMM and LCHMM-SI provide better visual quality with less artifacts than HMT.

1.3.6 Discussions of Image Denoising

In this section, we have proposed a new wavelet-domain HMM, called local contextual hidden Markov model (LCHMM) for statistical image modeling. The simulation results show that LCHMM can achieve the state-of-the-art image denoising performance with three major advantages, i.e., spatial adaptability, non-structured local region modeling, and fast model training. However, the main drawback of LCHMM is “over-fitting” in terms of number of model parameters which is even larger than the number of wavelet coefficients to be modeled. This drawback may prevent LCHMM from the wider applications. Nevertheless, here LCHMM demonstrates its evident advantages in image estimation and restoration applications.
1.4 Image Segmentation

Bayesian approaches to image segmentation have been proven efficient to integrate both image features and prior contextual properties, where maximum *a posteriori* (MAP) estimation is usually involved. In [30, 31], Markov random field (MRF) was developed to model contextual behavior of image data, and Bayesian segmentation becomes the MAP estimate of the unknown MRF from the observed data. Since the MRF model usually favors the formation of large uniformly classified regions, it may over-smooth the texture boundaries and wipe off small isolated areas. The non-causal dependence structure of MRF’s typically results in the high computational complexity. Recently, researchers proposed multiscale techniques which apply contextual behavior in the coarser scale to guide the decision in the finer scale and retain the underlying MRF model in each fixed scale, e.g., [32, 33]. In particular, in [33], Markovian dependencies are assumed across scales to capture interscale dependencies of multiscale class labels with a causal MRF structure, so that a non-iterative segmentation algorithm was developed where a sequential MAP (SMAP) estimator replaces the MAP estimator. In this section, we will develop a joint multi-context and multiscale (JMCMS) approach to Bayesian segmentation which can be formulated as a multi-objective optimization as the extension of the single objective optimization involved in [8, 33, 34]. To estimate the SMAP with respect to multiple context models in JMCMS, we use the heuristic multi-stage problem solving technique in [35]. The simulation results show that the proposed JMCMS algorithm improves the accuracy of texture classification, boundary localization and detection at the comparable computational cost [36].

1.4.1 Multiscale Bayesian Segmentation

We briefly review the multiscale segmentation approaches in [33]. Given a random field $Y$, we need to accurately estimate the pixel label in $X$ where each label specifies one of $N$ possible classes. Bayesian estimators attempt to minimize the average cost of an erroneous segmentation, as shown in the following:

$$\hat{x} = \arg\max_x E[C(X, x) | Y = y]$$

(1.25)

where $C(X, x)$ is the cost of estimating the true segmentation, $X$. The MAP estimate is the solution of (1.25), if we use the cost functional of $C_{MAP}(X, x) = 1$ whenever any pixel is incorrectly classified. It means that the MAP estimator aims at maximizing the probability that all pixels will be correctly classified. It is known that the MAP estimator is excessively conservative. Therefore, multiscale Bayesian segmentation was proposed in [33], where sequential MAP (SMAP) cost function, $C_{SMAP}(X, x)$, was introduced by proportionally summing up the segmentation errors from multiple scales together. The SMAP estimator aims at minimizing the spatial size of errors, resulting in more desirable segmentation results with lower computational complexity than the MAP estimator. The multiscale image model proposed in [33] is com-
posed of a series of random fields at multiple scales. Each scale has a random field of image feature vectors, $Y^{(n)}$, and a random field of class labels, $X^{(n)}$. We denote an individual sample at scale $n$ by $y_s^{(n)}$ and $x_s^{(n)}$, where $s$ is the position in a 2-D lattice $S^{(n)}$. Assuming Markovian dependencies across scales, the SMAP recursion can be computed in the fashion of coarse-to-fine as follows,

$$\hat{x}^{(n)} = \arg \max_{x^{(n)}} \{ \log p_{y^{(n)}}(y|x^{(n)}) + \log p_{x^{(n+1)}}(x^{(n+1)} | \hat{x}^{(n+1)}) \}.$$  \hspace{1cm} (1.26)

The two terms in (1.26) are the likelihood function of the image feature $y^{(n)}$ and the context-based prior knowledge from the next coarser scale, respectively. Specifically, the quadtree pyramid was developed in [33] to capture interscale dependencies of multiscale class labels regarding the latter part of (1.26). Thanks to the multiscale embedded structure, the quadtree model allows the efficient recursive computation of likelihood functions, but it also results in discontinuous texture boundaries due to the fact that spatially adjacent samples may not have common parent sample at the next coarser scale. Therefore, a more generalized pyramid graph model was introduced in [34] where each sample has more parent samples in the next coarser scale. However, this pyramid graph also complicates the computation of likelihood functions, and the fine-to-coarse recursion of (1.26) has to be solved approximately. A trainable context model for multiscale Bayesian segmentation was proposed in [34], where $x_s^{(n)}$ is assumed to be only dependent on $x_s^{(n)}$, a set of neighboring samples $(5 \times 5)$ at the coarser scale, and $\partial_s \subset S^{(n+1)}$ denotes a $5 \times 5$ window of samples at scale $n + 1$. The behavior of this simplified contextual structure can be trained off-line by providing sufficient training data including many images and their ground truth segmentations. Then the segmentation can be accomplished efficiently via a single fine-to-coarse-to-fine iteration through the pyramid.

### 1.4.2 HMTseg Algorithm

A distinct context-based Bayesian segmentation algorithm was proposed in [8] where the context model is characterized by a context vector $v^{(n)}$ derived from a set of neighboring samples $(3 \times 3)$ in the coarser scale. It is assumed that, given $y_s^{(n)}$, its context vector $v_s^{(n)} = \{x_s^{(n)}, x_{\ell_s}^{(n)}\}$ can provide supplementary information regarding $x_s^{(n)}$, where $x_s^{(n)}$ denotes the class label of the parent sample and $x_{\ell_s}^{(n)}$ the dominant class label of the $3 \times 3$ samples at the coarser scale. Both $\ell_{gs} \subset S^{(n+1)}$, the position of the parent sample, and $\ell_s \subset S^{(n+1)}$, a $3 \times 3$ window centered at $\ell_{gs}$, are at scale $n + 1$. So given $v_s^{(n)}, x_s^{(n)}$ is independent with all other class labels. In particular, the contextual prior $p_{x^{(n)}|v^{(n)}}(c|u)$ is involved in the SMAP estimation which can be estimated by maximizing the following context-based mixture model likelihood as,

$$f(y^{(n)}|v^{(n)} = u) = \prod_{s \in S^{(n)}} \sum_{c=1}^{N_c} p_{x^{(n)}|v^{(n)}}(c|v_s^{(n)} = u) f(y_s^{(n)}|x_s^{(n)} = c), \hspace{1cm} (1.27)$$
where the likelihood function $f(y^{[n]} | x^{[n]} = c)$ is computed by using the wavelet-domain HMT model. An iterative EM algorithm was developed in [8] to approach (1.27), and the SMAP estimate is obtained by

$$x^{[n]} = \arg \max_{x^{[n]}} p_{x^{[n]} | y^{(n)}}(x^{[n]} | y^{[n]}, y^{(n)})$$

(1.28)

where

$$p_{x^{(n)} | y^{(n)}}(x^{(n)} | y^{(n)}, y^{(n)}) = \frac{p_{x^{(n)}}(x^{[n]} | y^{(n)}) p_{x^{[n]} | y^{(n)}}(y^{[n]} | x^{[n]} = c) f(y^{[n]} | x^{[n]} = c)}{\sum_{c=1}^{N_c} p_{x^{(n)}}(c) p_{x^{[n]} | y^{(n)}}(y^{[n]} | x^{[n]} = c) f(y^{[n]} | x^{[n]} = c)}$$

(1.29)

Particularly, wavelet-domain HMT was used to obtain the statistical multiscale characterization regarding the likelihood function $f(y^{[n]} | x^{[n]} = c)$. Using the Haar DWT of the best spatial localizability, an image can be recursively divided into four sub-images of same size $J$ times and represented in a pyramid of $J$ scales, as shown in Fig. 1.9(a). We denote a dyadic block at scale $n$ as $y^{[n]}$. Given a set of Haar wavelet coefficients $w$ and a set of HMT model parameters $\theta$, the dyadic block $y^{[n]}$ is associated with three wavelet subtrees $\{T_{LH}^{[n]}, T_{HL}^{[n]}, T_{HH}^{[n]}\}$. The three wavelet subtrees are rooted in the tree wavelet coefficients from three wavelet subbands at scale $n$. Regarding the model likelihood in (1.29), the computation $f(y^{[n]} | \theta)$ is a realization of the HMT model $\theta$ and is obtained by

$$f(y^{[n]} | \theta) = f(T_{LH}^{[n]} | \theta_{LH}) f(T_{HL}^{[n]} | \theta_{HL}) f(T_{HH}^{[n]} | \theta_{HH}),$$

(1.30)

where it is assumed that three wavelet subbands are independent and each component in (1.30) can be computed based the closed formula in [5].

### 1.4.3 Joint Multi-context and Multiscale (JMCMS) Approach

The context-based Bayesian segmentation approaches [8, 33, 34] have been applied to multispectral SPOT images, document images, and aerial photos, etc. It was found that segmentation results in homogeneous regions are usually better than those around texture boundaries. This is mainly owing to the fact that the context models used in those approaches mainly capture interscale dependencies and encourage the formation of large uniformly classified regions with less consideration on texture boundaries. To improve the segmentation results in both homogeneous regions and texture boundaries simultaneously, we discuss two
Figure 1.10: Five context models between two neighboring scales where the coarser scale (top) and finer scale (bottom) are shown. \{V_1, ..., V_d\} \ (d = 1, 2, 3) is the context vector, and \(\Omega\) is defined in (1.31).

questions in this work. (i) What are the characteristics of context models of different structures in terms of their segmentation results? (ii) How can multiple context models of distinct advantages be integrated to implement the Bayesian segmentation? To answer the first question, we apply a set of numerical criteria to quantify the segmentation performance, and we conduct experiments on a set of synthetic mosaics to quantitatively analyze context models. We then propose a joint multi-context and multiscale (JMCMS) approach to Bayesian segmentation which is formulated as a multi-objective optimization problem. Particularly, We use the multi-stage problem solving technique in [35] to estimate SMAP of JMCMS.

Given a sample \(x_{s}^{[n]}\), its contextual information may come from some “neighbors” in the spatial and/or scale spaces. Then we naturally have three non-overlapped contextual sources as \(P = x_{s}^{[n]}\), \(NP = x_{\ell_{s}}^{[n]}\), and \(N = x_{\tilde{n}}^{[n]}\), where \(\ell_{s}\) is the 3 \times 3 window centered at \(\xi_{s}\) and excluding \(\xi_{s}\) at scale \(n + 1\), and \(\tilde{n}\) is the 3 \times 3 window centered at \(s\) and excluding \(s\) at scale \(n\). Specifically, \(P\) is the class label of \(\xi_{s}\), and \(PN\) and \(N\) are dominant class labels of \(s\) and \(\tilde{n}\), respectively. Other contextual sources could be possible, but we believe \(P\), \(NP\) and \(N\) are the most important ones, since they are the nearest to \(x_{s}^{[n]}\) in the pyramid representation, and high-order context models may introduce the context dilution problem [37]. Instead of using the majority voting scheme used in [8], which may have ambiguity when \(N_{c} > 2\), we determine the dominant class label, e.g., \(x_{\tilde{n}}^{[n]}\), over several samples, e.g., \(\tilde{n}\), by

\[
x_{\tilde{n}}^{[n]} = \arg \max_{c \in \{1, ..., N_{c}\}} \sum_{\ell \in \tilde{n}} p_{x_{\ell}^{[n]}|t(n), y(n)}(c|p_{c}^{[n]}, y_{\ell}^{[n]}),
\]

(1.31)

where we assume that each sample has the same textural contribution, that is measured by its posterior probability in (1.28), to the dominant class label over several samples. \(x_{\tilde{n}}^{[n]}\) can be obtained similarly. Based on \(P\), \(PN\), and \(N\), we develop five context models of different orders \(d\) as follows.

- \(d = 1\): Context-1 = \(\{P\}\) and Context-5 = \(\{N\}\);
- \(d = 3\): Context-3 = \(\{P, NP, N\}\).
Figure 1.11: The 10 synthetic mosaics (256 × 256, 8bpp) from Brodatz album [41]. From left to right, top to bottom: Mosaic1 (D9/D68), Mosaic2 (D16/D24), Mosaic3 (D15/D38), Mosaic4 (D16/D84/D24/D19), Mosaic5 (D24/D68/D16/D19), Mosaic6 (D9/D16/D19/D24/D28), Mosaic7 (D16/D24/D84), Mosaic8 (D38/D16/D15), Mosaic9 (D9/D16/D19), Mosaic10 (D24/D68/D16/D19).

Table 1.3: Segmentation results of the five contexts regarding $P_a$, $P_b$, and $P_c$.

<table>
<thead>
<tr>
<th></th>
<th>Context-1</th>
<th>Context-2</th>
<th>Context-3</th>
<th>Context-4</th>
<th>Context-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_a$</td>
<td>0.9365</td>
<td><strong>0.9728</strong></td>
<td>0.9260</td>
<td>0.9186</td>
<td>0.8327</td>
</tr>
<tr>
<td>$P_b$</td>
<td>0.2098</td>
<td>0.3567</td>
<td><strong>0.3716</strong></td>
<td>0.3000</td>
<td>0.1173</td>
</tr>
<tr>
<td>$P_c$</td>
<td>0.6389</td>
<td>0.5189</td>
<td>0.7071</td>
<td>0.7222</td>
<td><strong>0.7237</strong></td>
</tr>
</tbody>
</table>

The five context models are also as shown in Fig. 1.10, among which Context-1 and Context-2 are interscale context models which are similar to those used in [8, 34] to encourage the formation of large uniformly classified regions. Context-5 is an intrascale context model often used in the MRF literature to ensure the local homogeneous labeling with high sensitivity to boundaries. Context-3 and Context-4 are hybrid inter and intra scale context models which have the similar characteristics with those used in [38–40]. We anticipate that those context models have distinct effects on the segmentation results in terms of classification, boundary localization, and boundary detection. To study their characteristics, we use three numerical criteria to quantify the segmentation performance. Specifically, $P_a$ is the percentage of pixels which are correctly classified, showing accuracy, $P_b$ the percentage of boundaries that coincide with the true ones, showing boundary specificity, and $P_c$ the percentage of true boundaries that can be detected, showing boundary sensitivity. We conduct segmentation experiments on 10 mosaics, as shown in Fig. 1.11. For each context, we perform pixel-level segmentation on the 10 mosaics using the supervised context-based segmentation algorithm in [8]. $P_a$, $P_b$, and $P_c$ which are averaged over 10 trials are shown in Table 1.3.
A good segmentation requires high $P_a$, $P_b$, and $P_c$. Even though $P_a$ is usually the most important one, high $P_b$ and $P_c$ provide more desirable segmentation results with high accuracy of boundary localization and detection. On the other hand, boundary localization and detection in textured regions are usually regarded as difficult issues due to abundant edges and structures around textured boundaries [42, 43]. From Table 1.3, it is found that none of the five context models can work well singly in terms of the three criteria. For example, Context-2 has the best $P_a$ but the worst $P_c$. This fact experimentally verifies that the context models used in [8, 34] are good choices in terms of $P_a$. Context-5 is the strongest in $P_c$ but the weakest in $P_a$. Context-3 gives the highest $P_b$, but $P_a$ and $P_c$ suffer. These observations are almost completely consistent in each trial. Intuitively speaking, interscale context models, e.g., Context-1 and Context-2, favor $P_a$ by encouraging the formation of large uniformly classified regions across scales of the pyramid. The intrascale context model Context-5 helps $P_c$ by being sensitive to boundaries within a scale. As a hybrid inter and intra scale context model, Context-3 provides the best $P_b$ by appropriately balancing both interscale and intrascale dependencies into the SMAP Bayesian estimation. Hereby, a natural idea is to integrate multiple context models to achieve high $P_a$, $P_b$, and $P_c$ simultaneously.

Generally speaking, given $\mathbf{y} = \{\mathbf{y}^n|n = 1, 2, ..., L\}$ the collection of multiscale random fields of an image $\mathbf{Y}$, a context model $\mathbf{V}$ is used to simplify the characterization of the joint statistics of $\mathbf{y}$ with the local contextual modeling. Thus, given different context models, we can have different statistical characterizations of $\mathbf{y}$. Accordingly, we may have different Bayesian segmentation results. For example, the quadtree pyramid in [33] and the interscale context models in [8, 34] emphasize the homogeneity of the labeling across scales, and the segmentation results tend to be composed of large uniformly classified regions. However, those contexts cannot provide high accuracy of boundary localization and detection due to their limitations on boundary characterization. Similar to the multiscale image modeling in [38–40], intrascale or hybrid inter and intra scale context models can be used to achieve more accurate contextual modeling around boundaries, e.g., Context-3, Context-4, and Context-5. However, those contexts may be challenged in some homogeneous regions where the homogeneity is not very good in a certain scale. In this work, our goal is to apply multiple context models which have different advantages for image segmentation. Hence, $\mathbf{y}$ can be represented as multiple ($Z$) copies and each copy is characterized by a distinct context model, i.e., $\{\mathbf{y}_z|z = 1, 2, ..., Z\}$. Since different context models provide different multiscale modeling, leading to distinct results in terms of $P_a$, $P_b$, and $P_c$, we propose a joint multi-context and multiscale (JMCMS) approach to Bayesian segmentation, which reformulates (1.25) as a multi-objective optimization as,

\begin{equation}
\hat{x} = \arg\max_x E[C_{SMAP}(X, x)|Y_1 = \mathbf{y}_1],
\end{equation}

\begin{equation}
\vdots
\end{equation}

\begin{equation}
\hat{x} = \arg\max_x E[C_{SMAP}(X, x)|Y_Z = \mathbf{y}_Z].
\end{equation}
The multi-objective optimization in (1.32) is roughly analogous to the multiple criteria of \( P_a, P_b, \) and \( P_c \), and it can be regarded as a generalization of the single-optimization in (1.25). The problem of (1.32) can be approached by a heuristic algorithm, called the multi-stage problem solving technique in [35]. In other words, the problem in (1.32) can be broken into multiple stages, and the solution of a stage defines the constraints on the latter stage. Thus (1.32) can be solved based on multiple context models individually and sequentially. According to the multi-stage problem solving technique, the SMAP estimation of the posterior probabilities, as defined in (1.29), is conducted for all dyadic blocks with respect to three contexts individually and sequentially, and the SMAP decision is only made in the final step according to (1.25) or (1.28). The new JMCMS algorithm can be widely applied to different multiscale Bayesian segmentation methods using distinct texture models or texture features. Here we particularly adopt the supervised segmentation algorithm in [8] to implement context-based Bayesian segmentation where the wavelet-domain HMT is used to obtain multiscale texture characterization.

The implementation of the JMCMS approach to Bayesian segmentation is briefly listed as follows, where \( Z \) context models are used as \( \{ V_1, ..., V_Z \} \), and an \( L \)-scale image pyramid is involved, and \( n = 0 \) means the pixel-level representation. An important issue that should be addressed here is the determination of context vectors during the EM training process. Especially, the causal interscale context models, e.g., Context-1 and Context-2, have fixed context vectors during the EM training process. Meanwhile, the non-causal intrascale context models, e.g., Context-3, Context-4, and Context-5, require the real-time update of context vectors during each iteration based on the results of the previous step. The JMCMS segmentation algorithm is implemented as follows.

- **Step 1.** Set \( n = L - 1 \), starting from the next to the coarsest scale;

- **Step 2.** Set \( z = 1 \), starting from the first context model \( V_1 \) in the list;

- **Step 3.** Set \( p = 0 \), initializing \( \{ p_x^{(n)}(c), p_{\epsilon|\epsilon^{(n)}}(u|v) \} \) and \( v^{(n)} \);

- **Step 4.** Expectation (E) Step, as defined in (1.29);

- **Step 5.** If context model \( V_z \) is non-causal, update \( v^{(n)} \); else continue;

- **Step 6.** Maximization (M) Step, update contextual prior as

\[
p_{x|\epsilon^{(n)}}(c) = \sum_{x \in S^{(n)}} p_{x|\epsilon^{(n)}; \gamma^{(n)}}(x^{(n)} = c|\tilde{e}_s^{(n)}, \gamma_s^{(n)}) \tag{1.33}
\]

\[
p_{\epsilon|\epsilon^{(n)}}(u|c) = \frac{1}{p_{x|\epsilon^{(n)}}(c)} \sum_{e^{(n)} = u} p_{x|\epsilon^{(n)}; \gamma^{(n)}}(x^{(n)} = e|\tilde{e}_s^{(n)}, \gamma_s^{(n)}) \tag{1.34}
\]

- **Step 7.** Set \( p = p + 1 \). If converged (or \( p = N_p \)), then stop; else go to 4;

- **Step 8.** Set \( z = z + 1 \). If \( z > Z \), then stop; else use context \( X_z \) and go to 3;

- **Step 9.** Set \( n = n - 1 \). If \( n < 0 \), then stop; else go to 2;

- **Step 10.** \( \arg \max_c p_{x; \epsilon^{(0)}; \gamma^{(0)}}(c|\tilde{e}^{(0)}, \gamma^{(0)}) \) gives the pixel-level segmentation.
Table 1.4: The optimal JMCMS on the ten mosaics with \( Z = 1, 2, 3 \).

<table>
<thead>
<tr>
<th>JMCMS</th>
<th>Context-2(( Z = 1 ))</th>
<th>Context-2-5(( Z = 2 ))</th>
<th>Context-2-3-5 (( Z = 3 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_a )</td>
<td>0.9728</td>
<td>0.9893</td>
<td>0.9897</td>
</tr>
<tr>
<td>( P_b )</td>
<td>0.3567</td>
<td>0.6923</td>
<td>0.7259</td>
</tr>
<tr>
<td>( P_c )</td>
<td>0.5189</td>
<td>0.7314</td>
<td>0.7337</td>
</tr>
</tbody>
</table>

In Table 1.4, we list the optimal (numerically) settings of JMCMS on the ten synthetic mosaics when \( Z = 1, 2, 3 \). In practice, \( Z = 3 \) is found sufficient for the 10 mosaics in Fig. 1.11, and \( Z > 3 \) doesn’t help much in terms of \( P_a, P_b \), and \( P_c \). We also found that the JMCMS of Context-2-3-5, i.e., \( V_1=\text{Context-2}, V_2=\text{Context-3} \), and \( V_3=\text{Context-5} \), is the numerically best setting for the 10 mosaics regarding the three criteria, and it is almost completely consistent in each trial. It is interesting to notice that the context ordering of the optimal JMCMS algorithms given the order \( Z \) in Table 1.4 also somehow follows a coarse-to-fine way.

It is worth noting three facts about JMCMS: (1) The JMCMS of Context-2-3-5 may not be the universally optimal design, and we have the flexibility to design the tailored JMCMS for a specific application; (2) As an alternative, some sophisticated methods of selecting different contexts in a spatially adaptive fashion could be developed to improve the segmentation results. However, since the contextual prior is trained by the EM algorithm which needs sufficient data for the efficient training, the spatially adaptive context selection faces the difficulty of the robust prior estimation. The generally designed JMCMS here has good robustness and adaptability to various image data and texture boundaries; (3) The new JMCMS approach is neither pre-processing nor post-processing on the segmentation map, since the SMAP decision is only made in the final stage. The propose JMCMS is new approach to drive the Bayesian estimation of posterior probabilities toward a desired solution via multiple context models step by step.

### 1.4.4 Simulation Results

Here we test the proposed JMCMS approach of Context-2-3-5 (\( Z = 3 \)) on both synthetic mosaics and remotely sensed images. At the same time, we also study the segmentation algorithm in [8] where only Context-2 (\( Z = 1 \)) is used. For both cases, we use the wavelet-domain HMT model to obtain multiscale statistical texture characterization. We fix total iteration numbers of two methods to be the same e.g., \( N_p \times Z = 30 \). Thus they have the similar computational complexity, and the execution time is about 20-30 seconds for \( 256 \times 256 \) images (\( N_c = 2, 3, 4 \)) on a Pentium-II 400 computer. The average improvements on \( \tilde{P}_a, \tilde{P}_b \), and \( \tilde{P}_c \), are about 2\%, 32\%, and 18\%, respectively, over the ten mosaics given in Fig. 1.11. We also show the segmentation results of five mosaics in Fig. 1.12 where the improvements on \( P_a, P_b \), and \( P_c \) are also given.
Though Context-2 in [8] provides generally good segmentation results in homogeneous regions, the texture boundaries cannot be well localized and detected, i.e., low $R_b$ and $P_c$. This is the major shortcoming of most multiscale segmentation approaches where only interscale context models are used, e.g., [8, 33, 34]. It is shown that JMCMS can overcome this limitation. Firstly, accuracy of boundary localization and boundary detection are significantly improved with much smoother texture boundaries, as shown by $R_b$ and $P_c$. Secondly, the classification accuracy in homogeneous regions is also improved by reducing misclassified and isolated pixels, as shown by $P_a$. These improvements are owing to multiple context models used in JMCMS, where contextual information is propagated both across scales and via multiple context models to warrant good segmentation results in both homogeneous regions and boundaries.

One may argue that some simple processing methods, such as the morphological operation, can also provide smoother boundary localization. However, there are three limitations in the morphological operation for post-processing segmentation maps. Firstly, it cannot deal with errors of large size, as those appear in Fig. 1.12(d). Secondly, it may weaken the accuracy of boundary detection by producing over-smoothed boundaries. Thirdly, it may wipe off some small isolated targets which are important to some applications.

In the following, we conduct experiments on remotely sensed images, including an aerial photo and a SAR image, as shown in Fig. 1.13. Texture models are first trained on image samples which are manually extracted from original images ($512 \times 512$, 8bpp). We can see the improvements of JMCMS (Context-2-3-5) over HMTseg (Context-2). The accuracy of texture classification, in particular, boundary localization and detection are improved. Meanwhile, the small targets are kept in the segmentation map.
1.4.5 Discussions of Image Segmentation

In this section, a joint multi-context and multiscale (JMCMS) approach to Bayesian segmentation has been proposed. JMCMS is able to accumulate contextual behavior both across scales and via multiple context models, allowing more effective Bayesian estimation. JMCMS applies the wavelet-domain HMT to obtain multiscale texture characterization. JMCMS can be formulated as a multi-objective optimization which can be approached by the heuristic multi-stage problem solving technique. The proposed JMCMS algorithm has been applied to both synthetic mosaics and remotely sensed images. Simulation results show that JMCMS improves the accuracy of texture classification, and in particular, boundary localization and boundary detection over HMTseg. Meanwhile, small targets are kept well in the segmentation maps. We expect that the segmentation performance can be further improved by using more accurate texture models or features. JMCMS approach can be applied to other Bayesian segmentation algorithms using different texture models or features which are suitable for characterizing the texture information in remotely sensed images. Meanwhile, all Bayesian segmentation algorithms in [8, 33, 34] and the proposed JMCMS are supervised segmentation where the texture models are trained prior to the segmentation process. Unsupervised image segmentation using JMCMS and HMT was studied in [44] with promising results.
1.5 Texture Analysis and Synthesis

Textures play important roles in many computer vision and image processing applications, since images of real objects often do not exhibit regions of uniform and smooth intensities, but variations of intensities with certain repeated structures or patterns, referred to as visual texture. Most recent works on textures predominantly concentrate on two areas [45]. One is multi-channel filtering theory, which was inspired by the multi-channel filtering mechanism in neurophysiology [46] and was motivated by the evident advantages of multiscale texture analysis [32]. The other area is statistical modeling which characterizes textures as probability distributions from random fields, and statistical theories enable us to formulate and solve the problems of texture processing mathematically and systematically. Recently, texture characterization, based on the discrete wavelet transform (DWT) which integrates the above two aspects, has attracted much attention and was found useful for a variety of texture analysis and synthesis applications, including texture classification, texture segmentation, and texture synthesis. These approaches have been found more efficient than the traditional methods by considering the characteristics of human visual system on perceiving textures.

Wavelet-domain HMMs, e.g., HMT, have found powerful for statistical signal and image modeling and processing. When HMT was applied to image processing, it was usually assumed that the three DWT subbands, i.e., $HL$, $LH$, and $HH$, are independent. This assumption is found valid in modeling most real images, since real images usually carry a large amount of randomly distributed edges or structures, which weaken the cross-correlation between DWT subbands. However, we observed that, for natural textures, in particular structural textures, the regular spatial structures or patterns may result in certain dependencies across the three DWT subbands. It was also shown in [47] that the dependencies across subbands are useful for wavelet-based texture characterization. Specifically, in [9], a vector wavelet-domain HMT was proposed which incorporates multivariate Gaussian densities to capture statistical dependencies across DWT subbands. The vector HMT was applied to the redundant wavelet transform to obtain rotation invariant texture retrieval. It was demonstrated that the 2-state vector HMT in [9] has a moderate feature size but provides more accurate texture characterization than the 2-state scalar HMT in [5, 7].

In this section, we propose a new wavelet-domain HMM, HMT-3S, by integrating the three DWT subbands into one tree structure. In addition to the joint DWT statistics captured by HMT, the proposed HMT-3S can also exploit statistical dependencies across DWT subbands. Different from the vector HMM proposed in [9], we still impose the simple variable Gaussian mixture densities in the wavelet-domain. In HMT-3S, the state combination of three wavelet coefficients from the three DWT subbands results in the state number increased from 2 to 8, and the dependencies across DWT subbands can be characterized by the enlarged state transition matrices, i.e., $2 \times 2$ in HMT and $8 \times 8$ in HMT-3S. It is demonstrated that the more accurate texture characterization from HMT-3S improves the performance of texture analysis and synthesis [48].
1.5.1 Wavelet-Domain HMT-3S

As discussed before, the 2-state GMM is used to characterize the marginal statistics of wavelet coefficients. If we consider all wavelet coefficients are independent, we will obtain the so-called Independence Mixture Model (IMM) [5]. Wavelet-domain HMT was proposed to mainly capture interscale dependencies of wavelet coefficients across scales. When HMT was extended to 2-D case for image processing, the three wavelet subbands are usually considered as independent [7, 8]. In order to improve the accuracy of texture characterization by capturing dependencies across DWT subbands, we propose a new wavelet-domain HMM, HMT-3S, by grouping the three DWT subbands into one quad-tree structure. This grouping strategy is popular in the graphical modeling and Bayesian network literatures [18], and as discussed before, it was used to develop an improved HMT, HMT-2, by grouping every two neighboring coefficients into one node. It was shown that the more complete statistical characterization of DWT from HMT-2 improves the signal denoising performance, as well as the model training efficiency. We show the simplified characterization of HMT-3S in Fig. 1.14, where we see that HMT-3S has the same quad-tree structure as HMT, except for the number of coefficients in a node, i.e. the state number. If we assume the hidden state number to be 2 for each wavelet coefficient, there are 8 states in a node of HMT-3S. \(^1\) It is worth noting that 2-state GMMs are still used to characterize the DWT marginal statistics. Thus HMT-3S is parameterized by

\[
\theta_{HMT-3S} = \{p_J(u), c_{j,j-1}^u, \sigma_{B,j,b}^2 | B \in B, j = 1, ..., J, u, v = 0, ..., 7; b = 0, 1\}.
\]

The EM training algorithm in [5] can be straightforwardly extended to the 8-state HMT-3S. Similar to HMT, the HMT-3S model likelihood can be computed as follows.

\[
f(w|\theta_{HMT-3S}) = \sum_{k,i=0}^{N_j-1} \log \left( \frac{1}{7} \sum_{u=0}^{7} f_u(T_{j,k,i}|\theta_{HMT-3S}, u) \right),
\]

\(^1\)Since there are three coefficients in a node which follow three GMMs respectively, the three-dimensional Gaussian PDF is involved in HMT-3S. For simplicity, we assume the covariances of the multivariate Gaussian PDF are zeros here.
where $\mathcal{T}_{j,k,i}$ is the complex wavelet subtree rooted at $w_{j,k,i} = \{w_{j,k,i}^{LL}, w_{j,k,i}^{LH}, w_{j,k,i}^{HL}, w_{j,k,i}^{HH}\}$, as shown in Fig. 1.14, and $f(\mathcal{T}_{j,k,i}|\theta_{HMT-3S}, u)$ can be computed in a recursive fine-to-coarse fashion as follows,

$$f_u(\mathcal{T}_{j,k,i}|\theta_{HMT-3S}, u) = p_j(u)g(w_{j,k,i}|u)\left(\prod_{s=2k}^{2k+1} \prod_{t=2}^{t+1} \sum_{v=0}^{7} e_{j,s-1}^{u,v}f_u(\mathcal{T}_{j-1,s,t}|\theta_{HMT}, v)\right),$$

(1.36)

and in the finest scale, i.e. $j = 1$, we have

$$f_v(\mathcal{T}_{1,k,i}|\theta_{HMT-3S}, v) = p_1(v)g(w_{1,k,i}|v),$$

(1.37)

and

$$g(w_{j,k,i}|v) = \prod_{B \in \mathcal{B}} g(w_{j,k,i}^B|0, \sigma_{B,j,b}^2),$$

(1.38)

where $b = S_B \& v$, with $S_{HL} = 1$, $S_{LH} = 2$, and $S_{HH} = 4$.

Both HMT and HMT-3S have the similar recursion to compute model likelihood functions. Particularly, HMT-3S involves more parameters to characterize statistical dependencies across DWT subbands, i.e. the $2 \times 2$ state transition matrix in HMT, $e_{j-1}^{B,j}(m, n)$ with $m, n = 0, 1$, becomes the $8 \times 8$ one in HMT-3S, $e_{j-1}^{u,v}$ with $u, v = 0, 1, ..., 7$. We expect that the HMT-3S allows more accurate texture characterization by capturing more complete DWT cross-correlations, as shown in the following three texture processing applications, i.e. classification, segmentation, and synthesis.

### 1.5.2 Texture Classification

Texture classification is one of the most important applications of texture analysis, and it involves the identification of the texture class given a homogeneous textured region. Here, four wavelet-based texture classification methods are tested in the following, which adopt four different wavelet-domain features or models, i.e. the wavelet energy signature (WES), IMM, HMT, and HMT-3S. WES characterizes the energy distribution along the frequency axis over scales and subbands, and we adopt the Euclidean distance the measure the difference between two WESs. Moreover, we also study texture classification using the three wavelet-domain statistical models, i.e. IMM, HMT, and HMT-3S all of which can be adapted into the maximum likelihood classifier. Given the trained model $\theta$ and the observed DWT $w$, we have closed formulas to compute the model likelihood functions $f(w|\theta)$ that measure how well the models $\theta$ (IMM, HMT, and HMT-3S) describe the data $w$. For simplicity, we consider the case in which the prior probabilities to the texture classes are equal, and where the goal is to minimize the overall probability of classification error, the optimal decision becomes the maximum likelihood (ML) rule, which is to choose the class which makes the observed data most likely, i.e.

$$C_{ML} = \arg \max_{c \in \{1, ..., N_c\}} f(w|\theta^c).$$

(1.39)
The simulation of texture classification in this work is based on 55 Brodatz textures of a size of 640 by 640. There were two reasons for us to choose these 55 textures in this work. One was that the overall PCC of the WES along is less than 80% upon this set of textures, thus creating a hard problem in examining the IMM, HMT, and HMT-3S in terms of the accuracy of texture characterization. The other was that the classification experiment here is conducted on 64 × 64 texture samples, so we exclude the textures that are non-homogeneous in this small size, thus providing a reasonable test environment.

Figure 1.15: The sample images (64 × 64, 8bpp) of the 55 Brodatz textures used in the experiment.

Prior to the classification experiment, four sets of texture features or models, i.e. WES, IMM, HMT and HMT-3S, are obtained and stored for each texture. In order to ensure the robustness of the model estimation. All models are trained from the whole texture image (640 × 640). It was also found that the translation of the training image has almost no effect on the model estimation. On the other hand, in order to test and compare four methods with respect to their results for texture classification, we divide a texture image into 19 × 19 subimages of 64 by 64, so that horizontally and vertically neighboring subimages overlap each other with 32 columns (or rows). Here we use the 4-scale Daubechies-8 DWT. The selection of optimal wavelet filter basis for texture characterization was studied in [49], and we are particularly interested in wavelet-domain statistical modeling which may be further jointly considered with the results reported in [49]. The four texture classification methods are performed on 361 texture samples, and the PCC is recorded for each texture. Then we use the the distribution, mean, and standard deviation (stdev) of 55 PCCs over 55 trials to evaluate the overall classification performance, as shown in Table 1.5. From Table 1.5, we have the following three major results.

---

2 These textures are obtained from the Brodatz database at http://www.ux.his.no/~tranden/brodatz/.
Table 1.5: Texture classification performance of the four methods in terms of PCC (%). “(+)” and “(*)” signs denote a strong structural texture and a statistical texture, respectively. The feature sizes of WES, IMM, HMT, and HMT-3S are 12, 36, 45, and 199, respectively.

<table>
<thead>
<tr>
<th>Textures</th>
<th>WES</th>
<th>IMM</th>
<th>HMT</th>
<th>HMT-3S</th>
<th>Textures</th>
<th>WES</th>
<th>IMM</th>
<th>HMT</th>
<th>HMT-3S</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1(+)</td>
<td>92.2</td>
<td>94.2</td>
<td>92.5</td>
<td>97.8</td>
<td>D77</td>
<td>100</td>
<td>99.4</td>
<td>99.4</td>
<td>99.7</td>
</tr>
<tr>
<td>D3(*)</td>
<td>83.7</td>
<td>93.4</td>
<td>95.3</td>
<td>97.2</td>
<td>D78(*)</td>
<td>87.5</td>
<td>96.7</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>D4(*)</td>
<td>80.9</td>
<td>94.7</td>
<td>94.7</td>
<td>95.6</td>
<td>D79</td>
<td>95</td>
<td>97.8</td>
<td>99.2</td>
<td>100</td>
</tr>
<tr>
<td>D6(+)</td>
<td>93.9</td>
<td>99.4</td>
<td>100</td>
<td>100</td>
<td>D80(*)</td>
<td>70.9</td>
<td>85.6</td>
<td>93.1</td>
<td>95.6</td>
</tr>
<tr>
<td>D9(*)</td>
<td>47.6</td>
<td>98.1</td>
<td>97.8</td>
<td>97.5</td>
<td>D81(+)</td>
<td>72.6</td>
<td>89.8</td>
<td>84.8</td>
<td>92.8</td>
</tr>
<tr>
<td>D11(*)</td>
<td>72.3</td>
<td>98.3</td>
<td>96.7</td>
<td>100</td>
<td>D82(*)</td>
<td>99.7</td>
<td>100</td>
<td>99.7</td>
<td>100</td>
</tr>
<tr>
<td>D15(+)</td>
<td>74.8</td>
<td>83.9</td>
<td>83.4</td>
<td>91.1</td>
<td>D83(*)</td>
<td>95.3</td>
<td>99.2</td>
<td>98.1</td>
<td>98.6</td>
</tr>
<tr>
<td>D16(*)</td>
<td>99.7</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>D84(*)</td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>D17</td>
<td>99.2</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>D85(*)</td>
<td>93.4</td>
<td>97.8</td>
<td>97</td>
<td>99.4</td>
</tr>
<tr>
<td>D18(*)</td>
<td>40.2</td>
<td>90</td>
<td>82.5</td>
<td>74.8</td>
<td>D87(+)</td>
<td>83.1</td>
<td>93.9</td>
<td>85</td>
<td>97.2</td>
</tr>
<tr>
<td>D19</td>
<td>85.3</td>
<td>86.4</td>
<td>75.1</td>
<td>90.6</td>
<td>D92</td>
<td>87.3</td>
<td>97.8</td>
<td>98.3</td>
<td>99.2</td>
</tr>
<tr>
<td>D20(+)</td>
<td>63.4</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>D93</td>
<td>74</td>
<td>74</td>
<td>95.3</td>
<td>97.5</td>
</tr>
<tr>
<td>D21</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>D94(+)</td>
<td>74.5</td>
<td>80.9</td>
<td>93.6</td>
<td>95.8</td>
</tr>
<tr>
<td>D22</td>
<td>50.4</td>
<td>81.2</td>
<td>95.8</td>
<td>87.3</td>
<td>D95(+)</td>
<td>47.9</td>
<td>92</td>
<td>90.6</td>
<td>94.5</td>
</tr>
<tr>
<td>D24(*)</td>
<td>89.5</td>
<td>92.8</td>
<td>92.2</td>
<td>92.5</td>
<td>D98(+)</td>
<td>59.8</td>
<td>88.1</td>
<td>88.1</td>
<td>92</td>
</tr>
<tr>
<td>D28(*)</td>
<td>64</td>
<td>96.1</td>
<td>91.7</td>
<td>91.4</td>
<td>D101</td>
<td>33</td>
<td>95.6</td>
<td>96.4</td>
<td>94.2</td>
</tr>
<tr>
<td>D29(*)</td>
<td>90.3</td>
<td>98.9</td>
<td>99.7</td>
<td>99.7</td>
<td>D102(+)</td>
<td>37.7</td>
<td>42.1</td>
<td>49.9</td>
<td>77</td>
</tr>
<tr>
<td>D32(*)</td>
<td>99.7</td>
<td>99.2</td>
<td>99.7</td>
<td>99.7</td>
<td>D103</td>
<td>80.1</td>
<td>99.7</td>
<td>98.9</td>
<td>98.3</td>
</tr>
<tr>
<td>D33(*)</td>
<td>82.8</td>
<td>90.3</td>
<td>95</td>
<td>93.9</td>
<td>D104</td>
<td>50.1</td>
<td>71.2</td>
<td>74.5</td>
<td>75.3</td>
</tr>
<tr>
<td>D34(+)</td>
<td>91.4</td>
<td>99.7</td>
<td>100</td>
<td>100</td>
<td>D106</td>
<td>98.1</td>
<td>9.4</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>D35(+)</td>
<td>81.4</td>
<td>96.7</td>
<td>83.7</td>
<td>93.6</td>
<td>D109(*)</td>
<td>57.6</td>
<td>80.3</td>
<td>83.7</td>
<td>85.6</td>
</tr>
<tr>
<td>D36(+)</td>
<td>68.4</td>
<td>87</td>
<td>91.4</td>
<td>93.9</td>
<td>D110(*)</td>
<td>80.6</td>
<td>97</td>
<td>99.4</td>
<td>100</td>
</tr>
<tr>
<td>D49</td>
<td>100</td>
<td>1.4</td>
<td>100</td>
<td>100</td>
<td>D111(+)</td>
<td>70.6</td>
<td>94.2</td>
<td>95</td>
<td>98.6</td>
</tr>
<tr>
<td>D52(+)</td>
<td>48.8</td>
<td>67.3</td>
<td>84.8</td>
<td>92.5</td>
<td>(#PCC=100)</td>
<td>3</td>
<td>7</td>
<td>13</td>
<td>18</td>
</tr>
<tr>
<td>D53(+)</td>
<td>99.2</td>
<td>70.4</td>
<td>99.2</td>
<td>99.7</td>
<td>(#100&gt;PCC≥90)</td>
<td>16</td>
<td>29</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>D55(+)</td>
<td>93.6</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>(#90&gt;PCC≥80)</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>D57(*)</td>
<td>78.7</td>
<td>92.8</td>
<td>98.1</td>
<td>97.8</td>
<td>(#80&gt;PCC≥70)</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>D65(+)</td>
<td>67.3</td>
<td>78.4</td>
<td>100</td>
<td>100</td>
<td>(#PCC&lt;70)</td>
<td>16</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D66(+)</td>
<td>41.3</td>
<td>98.1</td>
<td>100</td>
<td>100</td>
<td>Mean of 20 PCCs (+)</td>
<td>72.5</td>
<td>86.0</td>
<td>90.4</td>
<td>95.0</td>
</tr>
<tr>
<td>D68</td>
<td>96.1</td>
<td>88.1</td>
<td>77</td>
<td>78.1</td>
<td>Mean of 20 PCCs (*)</td>
<td>80.5</td>
<td>95.1</td>
<td>95.7</td>
<td>96.0</td>
</tr>
<tr>
<td>D75(+)</td>
<td>67.9</td>
<td>52.4</td>
<td>74.8</td>
<td>76.5</td>
<td>Mean of 55 PCCs (all)</td>
<td>77.4</td>
<td>87.4</td>
<td>93.1</td>
<td>95.2</td>
</tr>
<tr>
<td>D76</td>
<td>70.6</td>
<td>94.7</td>
<td>99.4</td>
<td>100</td>
<td>Stdev of 55 PCCs (all)</td>
<td>18.99</td>
<td>20.04</td>
<td>9.52</td>
<td>6.97</td>
</tr>
</tbody>
</table>
Overall Performance: HMT-3S gives the highest overall PCC (above 95%) and the best numerical stability with the smallest Stdev of PCCs over 55 textures (below 7%). Specifically, HMT-3S can correctly identify 18 textures out of 55 ones with 100% PCCs, and it can accurately classify most textures with over 90% PCCs, (48 out of 55). There is no PCC< 70% for HMT-3S. For most textures, IMM, HMT, HMT-3S outperform WES by significantly improving PCCs, showing the exploration of the marginal statistics and higher-order dependencies are useful for texture characterization.

Structural Textures: The advantages of HMT-3S over WES, IMM, and HMT are evident for structural textures, where dependencies across DWT subbands are relatively strong. Particularly, for 20 strong structural textures that are marked with (+) in Table 1.5, HMT-3S can outperform WES, IMM, and HMT by 17.5%, 9%, and 4.5% PCC improvements, respectively. This fact is consistent with the major motivation of HMT-3S.

Statistical Textures: IMM, HMT, and HMT-3S have the similar performance, showing that the exploration of statistical dependencies across scales or across subbands cannot significantly improve texture characterization for statistical textures, where high-order dependencies are relatively weak. Particularly, for 20 statistical textures that are marked with (*) in Table 1.5, the PCCs from IMM, HMT, and HMT-3S are very close. This observation is also consistent with the prerequisite of HMT-3S and HMT for statistical modeling.

As the main drawback, the feature size of HMT-3S is the largest one. This is mainly due to the fact that 8 × 8 state transition matrices are involved in HMT-3S which characterize dependencies across both scales and subbands. Still the above experiment manifests that HMT-3S can improve the accuracy of statistical texture characterization, in particular for structural textures. This fact can be further verified by the application of texture segmentation as follows.

1.5.3 Texture Segmentation

Texture segmentation is closely related to texture classification. In particular, we want to study how the segmentation performance can be improved by using a more accurate texture model, i.e. HMT-3S instead of HMT used in (1.26). It was shown that both texture modeling and contextual modeling are important to the performance of multiscale Bayesian segmentation [50]. The JMCMS algorithm described before improves texture segmentation by using more sophisticated contextual modeling of multiscale class labels regarding the latter part of (1.26). We here attempt to improve texture segmentation by using more accurate texture models, i.e., HMT-3S instead of HMT, regarding the former part of (1.26). It is worth noting that the computation of model likelihood is the only step related to the feature size and is neglectable compared with
Figure 1.16: (a) Mosaic-A. (b) HMTseg (HMT), $P_a = 88.10\%$. (c) JMCMS (HMT), $P_a = 95.08\%$. (d) JMCMS (HMT-3S), $P_a = 96.55\%$. (e) Mosaic-B. (f) HMTseg (HMT), $P_a = 69.84.10\%$. (g) JMCMS (HMT), $P_a = 73.37\%$. (h) JMCMS (HMT-3S), $P_a = 87.13\%$.

the latter Bayesian estimation in JMCMS or HMTseg. Hereby we study both the HMTseg and the JMCMS algorithms by substituting HMT-3S for HMT to compute the model likelihood used in Bayesian estimation. In Fig. 1.16, we show the segmentation results of two synthetic texture mosaics using both HMTseg and the JMCMS algorithms where two texture models, HMT-3S and HMT, are tested. The computational complexity of three implementations are similar for multiscale Bayesian segmentation.

We see that both JMCMS and HMT-3S can improve segmentation results over the HMTseg algorithm in terms of classification accuracy by emphasizing the two terms in (1.26), respectively. Moreover, the combination of JMCMS and HMT-3S provides the best results with homogeneous regions and texture boundaries. Intuitively speaking, HMT-3S intends to generate homogeneous segmentation regions by providing robust and accurate texture characterization, and JMCMS attempts to clean misclassified pixels and to smooth texture boundaries by minimizing the spatial size of errors. Thought two efforts seem to be conflicting, the integration of HMT-3S and JMCMS under the Bayesian estimation framework can provide a well balanced segmentation result, as shown in Figure 1.16. Particularly, when the number of texture types, $N_e$, is large, e.g. $N_e = 9$ in Fig. 1.16(a)(e), the advantages of HMT-3S over HMT are significant for segmentation. Hereby the segmentation results further verify the strengths of HMT-3S over HMT for statistical texture characterization.
1.5.4 Texture Synthesis

As the counter-part issue of texture analysis, texture synthesis attempts to generate synthetic textures which are visually indistinguishable to real textures according to a certain parametric texture model or a set of texture features. Texture synthesis is often used in image compression and image rendering applications. We want to study how to apply wavelet-domain HMMs to texture synthesis. In particular, in (1.35), we have closed formulas to compute likelihood functions \( f(\mathbf{w} | \theta) \) given the model \( \theta \) (HMT or HMT-3S) and the observed data \( \mathbf{w} \), and \( f(\mathbf{w} | \theta) \) shows how well the model \( \theta \) fits the data \( \mathbf{w} \). For texture analysis, we use the EM algorithm to estimate \( \theta \) of HMT or HMT-3S by maximizing \( f(\mathbf{w} | \theta) \). On the contrary, when we apply HMT or HMT-3S to texture synthesis, we need to generate \( \mathbf{w} \) which can be best parameterized by a given \( \theta \) as

\[
\hat{\mathbf{w}} = \arg \max_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w} | \theta),
\]

subject to the mean and variance constraints of \( \mathbf{w} \) implied by \( \theta \). Thus the problem of texture synthesis using HMT can be formulated as a constrained optimization. Usually, to make it easier, we can change a constrained optimization into an unconstrained one by using the penalty function technique [51]. On the other hand, since it is hard to compute the overall \( \mathbf{w} \) by a single operation, we adopt the multiscale scheme in [47] to update \( \mathbf{w} \) in a coarse-to-fine fashion, as described in the following where the problem is discussed in the 1-D form and applies to the 2-D case.

Given a 1-D HMT \( \theta \), and \( \mathbf{w}_j \), the set of \( N_j \) wavelet coefficients in scale \( j \), the local model likelihood of two adjacent scales, scale \( j \) and scale \( j+1 \), \( f(\mathbf{w}_j, \mathbf{w}_{j+1} | \theta) \) can be computed as,

\[
f(\mathbf{w}_j, \mathbf{w}_{j+1} | \theta) = \log(\sum_{k} \alpha_{j+1,k}(m)\beta_{j+1,k}(m)),
\]

where

\[
\alpha_{j+1,k}(m) = p_{j+1}(m)g(w_{j+1,k};0,\sigma_{j+1,m}^2),
\]

\[
\beta_{j+1,k}(m) = \prod_{i=0}^{1} (\sum_{n=0}^{1} n_{j+1}(w_{j,2k+i};0,\sigma_{j,n}^2)).
\]

Since we use the coarse-to-fine scheme, we fix \( \mathbf{w}_{j+1} \) to find \( \mathbf{w}_j \) which can maximize \( f(\mathbf{w}_j, \mathbf{w}_{j+1} | \theta) \), i.e., \( f(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}) \), subject to the mean and variance constraints of expected \( \mathbf{w}_j \). As said before, we can define a new objective function \( h(\mathbf{w}_j | \theta) \) by introducing two penalty functions as follows,

\[
h(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}) = f(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}) - K_1(\frac{\sum_k w_{j,k}}{N_j} - \eta_j)^2 - K_2(\frac{\sum_k w_{j,k}^2}{N_j} - \delta_j^2)^2,
\]

\[
= f(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}) - K_1(\Delta e_1)^2 - K_2(\Delta e_2)^2,
\]

where \( \eta_j = 0 \) and \( \delta_j^2 = \sum_{m=0}^{1} p_{j}(m)\sigma_{j,m}^2 \) are the expected mean and variance implied by \( \theta \), respectively. \( \Delta e_1 \) and \( \Delta e_2 \) are the errors between the mean and variance of estimated \( \mathbf{w}_j \) and the expected ones, respectively. \( K_1 \) and \( K_2 \) are two positive constants whose values are set empirically to balance the effects of three
terms in \( h(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}) \) appropriately. Thus the constrained ML-based texture synthesis in (1.40) is changed into an unconstrained optimization as

\[
\mathbf{w}_j = \arg \max_{\mathbf{w}_j \in \mathbb{R}} h(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}). \tag{1.45}
\]

The solution to (1.45) can be iteratively obtained by the steepest ascent algorithm which updates \( \mathbf{w}_j \) in the direction of maximizing \( h(\mathbf{w}_j | \theta, \mathbf{w}_{j+1}) \) as,

\[
\mathbf{w}_j = \mathbf{w}_j + \lambda_j \nabla(h(\mathbf{w}_j | \theta, \mathbf{w}_{j+1})), \tag{1.46}
\]

where \( \lambda_j \) is the step size which is tuned to warrant the convergence of the algorithm, and

\[
\mathbf{w}_j = \begin{bmatrix}
w_{j,0} \\
w_{j,1} \\
\vdots \\
w_{j,L}
\end{bmatrix},
\quad \nabla(h(\mathbf{w}_j | \theta)) = \begin{bmatrix}
\frac{\partial h}{\partial w_{j,0}} \\
\frac{\partial h}{\partial w_{j,0}} \\
\vdots \\
\frac{\partial h}{\partial w_{j,L}}
\end{bmatrix},
\]

where \( L = N_j - 1 \), and

\[
\frac{\partial h}{\partial w_{j,k}} = \sum_{m=0}^{1} \frac{\Lambda_p}{\alpha_{j+1,l}(m)} \left( \frac{1}{\sum_{n=0}^{m} e_{j+1,l+1}(0, \sigma_{j,n})} \right) \left( \frac{1}{\sum_{n=0}^{m} e_{j+1,l+1}(w_{j,k}) \sigma_{j,n}^2} \right) \nabla_h \left( \frac{2K_0}{N_j} \Delta e_1 - \frac{4K_1}{N_j} \Delta e_2 w_{j,k}, \right) \tag{1.47}
\]

where \( k = 0, \ldots, L, l = [k/2], i = 2l (2l + 1) \) if \( k \) is odd (even), and \( g' \) is the derivative of the Gaussian function \( g \). Actually, \( w_{j,i} \) and \( w_{j,k} \) are two neighbors sharing the same parent \( w_{j+1,i} \) in the wavelet subtree. From (1.47), we see the update of \( w_{j,k} \) is dependent on four terms, i.e. \( \Lambda_p \) from its parent, \( \Lambda_n \) from its neighbor, \( \Lambda_o \) from its own, and \( \Lambda_e \) from the errors of mean and variance, so that \( w_{j,k} \) can be updated along the direction of \( \nabla(h(\mathbf{w}_j | \theta)) \) subject to those constraints. Particularly, the cross-correlation constraint from HMT is mainly represented by the \( e_{j,i} \) in \( \Lambda_n \) and \( \Lambda_o \). (1.41) to (1.47) can be straightforwardly extended to the 2-D HMT and HMT-3S, since 1-D HMT, 2-D HMT, and 2-D HMT-3S share the similar tree structure, and have the similar recursive computation of model likelihood defined in (1.35). The major changes to (1.47) will be on \( \Lambda_n \) and \( \Lambda_o \) when the 2-D HMT or HMT-3S is used. Thanks to the integrated characterization of the joint DWT statistics from wavelet-domain HMT and HMT-3S, the problem of texture synthesis can be formulated mathematically and solved systematically, as shown from (1.40) to (1.47). It is expected that HMT and HMT-3S allow us to impose statistical constraints efficiently for texture analysis.

Since HMT mainly capture cross-correlations of the DWT, auto-correlations are also needed here to represent the periodicity and global oriented structures in textures [47, 52]. The statistical constraints from
auto-correlations can be imposed by zero-phase 2-D linear filtering [47]. In addition, we adopt the histogram specification algorithm to impose the statistics of gray-level texture pixels [53]. Using the similar framework to the one in [47], we develop a new texture synthesis algorithm, as shown in Fig. 1.17, where only a 2-scale DWT is used for simplicity.

![Figure 1.17: The texture synthesis algorithm using HMT where a 2-scale DWT is adopted.](image)

The synthesis process begins with an image containing samples of Gaussian white noise. In the wavelet-domain, a recursive coarse-to-fine procedure imposes the statistical constraints through HMT and the autocorrelations, while simultaneously reconstructing a lowpass image, until obtain the synthesized image after the histogram specification in the spatial domain. The entire process is repeated until converge (visually). The same as the most synthesis algorithms, we cannot guarantee convergence. In practice, the proposed algorithm has a fast visual convergence rate with appropriately tuned constants, i.e., $\lambda$ in (1.46), and $K_1$ and $K_2$ in (1.44). The current implementation requires roughly 10 minutes to synthesis a $256 \times 256$ texture (10 iterations) on a 400MHz Pentium-II computer.

In this experiment, we adopt the 4-level Daubechies-8 wavelet for texture synthesis. We have tried other wavelets, such as Daubechies-4,6,10 and Symmlet-4,5,6, etc. It seems that the wavelet with longer filter banks may provide slightly better texture synthesis results than the one with shorter filter banks regarding the convergence rate and visual quality. This may be due to the reason that the longer filter bank can provide a more compact DWT representation that allows more accurate statistical texture modeling. Both HMT and HMT-3S are tested here. Even though we found that HMT-3S allows faster visual convergence of the synthesized textures than HMT does, they are quite close eventually. We guess that autocorrelation constraints of the reconstructed low-pass images can partially compensate the loss of the characterization of dependencies across DWT subbands. We show four synthesized textures with 5 iterations in Fig. 1.18. It is shown that HMT-3S provides more perceptually favorable results than HMT does, and spatial structures or patterns can be generally replicated with less artifacts and distortions, e.g., D17, D68, and D87. Two models work similarly well for the textures dominated by random structures which may weaken dependencies across subbands, e.g, D57. This fact is consistent with the texture classification results in Section 1.5.2. More texture synthesis results can be found in Fig. 1.19. However, we also noticed that both HMT and HMT-3S cannot effectively capture the spatial periodicity and regularity in textures, as shown in Fig. 1.20.
Figure 1.18: Texture synthesis using HMT and HMT-3S.
Figure 1.19: Texture synthesis results using HMT-3S.

Figure 1.20: Texture synthesis failures.
Generally speaking, there are three major observations from the above experiment regarding the performance of texture synthesis as follows:

- For structural textures with irregular patterns, HMT-3S outperforms HMT by providing smoother and shaper structures, such as D68, D87, and D76, etc. This shows that HMT-3S is able to reproduce the sharpness and smoothness of the irregular structures by characterizing statistical dependencies across DWT subbands.

- For structural textures with periodic or regular patterns, both HMT-3S and HMT fail to reproduce the regular or periodic structures, such as D20, D36, and D75, etc. This is due to the fact that both HMT and HMT-3S are statistical models which cannot handle the regular and periodic structures or patterns. Secondly, the orthogonal/non-redundant DWT has the poor shift-invariant property and cannot well preserve the regularity and/or periodicity in the wavelet-domain, as indicated in [49].

- For statistical texture, the texture synthesis results are very close for HMT and HMT-3S, such as D57, D33, and D19, etc. That is because the high-order dependencies across scales or subbands are relatively weak for those textures where HMT and HMT-3S offer the similar statistical characterization.

1.5.5 Discussions of Texture Analysis and Synthesis

In this section, we have studied texture analysis and synthesis using wavelet-domain HMMs, in particular the hidden Markov tree (HMT), which were originally proposed in [5]. More importantly, we have proposed a new HMT-3S model to capture statistical dependencies across DWT subbands by using the graphical grouping technique. The basic idea of the HMT-3S is that a more complete statistical characterization of DWT can be implemented by more sophisticated graph structures for Bayesian inference in the wavelet-domain. The proposed HMT-3S has been applied to texture analysis, including classification and segmentation, and texture synthesis as well. The simulation results show that the proposed HMT-3S outperforms HMT by improving the PCC of texture classification, the segmentation accuracy, and the visual similarity of synthetic textures. This work demonstrates the applications of wavelet-domain hidden Markov models to texture analysis and synthesis, and shows that wavelet-domain statistical image modeling plays an important role in texture characterization. One limitation of this work is the proposed HMT-3S is established in the non-redundant DWT that is inferior to the redundant DWT for statistical modeling. However, this work can be useful for the applications where the non-redundant DWT is preferred, e.g., compression-domain image segmentation and feature extraction, etc. Moreover, the proposed texture synthesis algorithm can also be applied to other HMM in the redundant DWT that has a reasonable feature size.
1.6 Conclusions

In this Chapter, we have studied wavelet-domain statistical image modeling and processing. In particular, we have investigated wavelet-domain hidden Markov models (HMMs) which were originally proposed in [5] for statistical signal/image processing. We first improved wavelet-domain HMMs in terms of their training efficiency and modeling accuracy by developing several new techniques which further inspire our studies toward four applications: image denoising, image segmentation, and texture analysis and synthesis where we can obtain state-of-the-art performance or promising results by developing new wavelet-domain HMMs as well as efficient image processing algorithms. We conclude this chapter as follows.

- We show that training efficiency and modeling accuracy of wavelet-domain HMMs are critical to their applications to practical signal and image processing problems. Specifically, an efficient EM initialization scheme can improve the training performance of HMMs, especially for those newly developed HMMs, i.e., HMT-2, LCHMM, and HMT-3S. Meanwhile, the graphical grouping and classification schemes have been found efficient to obtain more accurate statistical modeling.

- We suggest that the spatial adaptability and non-structured local regional modeling are essential to the image denoising applications, since we need to consider the non-stationary property of real images. The tree-structured HMM, e.g., HMT, results in obvious denoising artifacts in denoised images. Thus, a new local contextual hidden Markov model (LCHMM) was proposed which provides state-of-the-art denoising performance at the low computational complexity.

- We argue that the performance of multiscale Bayesian segmentation can be improved by strengthening two factors: contextual modeling and texture characterization. A new joint multi-context and multiscale (JMCMS) approach to Bayesian segmentation was developed to consider the first factor. In JMCMS, contextual behavior can be accumulated both across scales and via multiple context models. On the other hand, a new wavelet-domain HMM, HMT-3S, was proposed to emphasize the second factor by providing more accurate texture characterization than HMT does. It was shown that the combination of JMCMS and HMT-3S provides the best segmentation results among all tested methods, as measured by the three numerical criteria.

- We point out that wavelet-domain statistical modeling is essential to texture analysis and texture synthesis applications. Unlike that for image denoising, the hierarchical tree-structured HMMs are desired, such as HMT or HMT-3S, which regard the whole wavelet subtree as one instance of the statistical model. Meanwhile, efficient texture processing algorithms are also very important to the applications of wavelet-domain HMMs for texture-related processing, such as maximum likelihood (ML)-based texture classification and ML-based texture synthesis.
Bibliography


