# Multiscale Texture Segmentation using Wavelet-Domain Hidden Markov Models

*Hyeokho Choi and Richard Baraniuk* *

Rice University, Department of Electrical and Computer Engineering, Houston, Texas 77005

## Abstract

*Wavelet-domain Hidden Markov Tree (HMT) models are powerful tools for modeling the statistical properties of wavelet transforms. By characterizing the joint statistics of the wavelet coefficients, HMTs efficiently capture the characteristics of a large class of real-world signals and images. In this paper, we apply this multiscale statistical description to the texture segmentation problem. Using the inherent tree structure of the HMT, we classify textures at various scales and then fuse these decisions into a reliable pixel-by-pixel segmentation.*

## 1 Introduction

The goal of an image segmentation algorithm is to assign a class label to each pixel of an image based on the properties of the pixels and their relationships with their neighbors. The segmentation process is a joint detection and estimation of the class labels and shapes of regions with homogeneous behavior.

For proper segmentation of images, both the large and small scale behaviors should be utilized to segment both large, homogeneous regions and detailed boundary regions. Thus, it is natural to approach the segmentation problem using multiscale analysis. Efforts have been exerted to model this multiscale behavior with autoregressive models [1, 2] and multiscale random fields [3]. In this paper, we propose a multiscale texture segmentation algorithm based on the wavelet transform.

Recently, the *wavelet-domain Hidden Markov Tree (HMT) model* was proposed to model the statistical properties of wavelet transforms [4, 5]. By modeling each wavelet coefficient as a Gaussian mixture density and by capturing the dependencies between wavelet coefficients as hidden state transitions, HMTs provide a natural setting for exploiting the structure inherent in real-world signals and images for signal detection and classification.

In this paper, we apply the tree structure of the HMT model to multiscale signal classification. By computing the likelihoods of dyadic subblocks of the image at different scales, we obtain several "raw" segmentations. Coarse scale segmentations are more reliable for large, homogeneous regions, while fine scale segmentations are more appropriate around boundaries between different textures. By combining raw segmentations from different scales, we obtain a robust and accurate overall result. We accomplish this *interscale fusion* by encoding the segmentation results from different scales using a special tree-structured graph. Before we develop these new algorithms, we sketch some background on wavelets and wavelet-domain HMT models.

## 2 Background

### 2.1 The wavelet transform

The discrete wavelet transform (DWT) represents a 1-d signal $z(t)$ in terms of shifted versions of a lowpass scaling function $\phi(t)$ and shifted and dilated versions of a prototype bandpass wavelet function $\psi(t)$ [6]. For special choices of $\phi(t)$ and $\psi(t)$, the functions $\psi_{j,k}(t) \equiv 2^{j/2}\psi\left(2^j t - k\right)$, $\phi_{j,k}(t) \equiv 2^{j/2}\phi\left(2^j t - k\right)$, with $j, k \in \mathbb{Z}$ form an orthonormal basis, and we have the representation [6]

$$z = \sum_k u_{j_0,k}\,\phi_{j_0,k} + \sum_{j=j_0}^{\infty}\sum_k w_{j,k}\,\psi_{j,k}, \qquad (1)$$

with $u_{j,k} \equiv \int z(t)\,\phi^*_{j,k}(t)\,dt$ and $w_{j,k} \equiv \int z(t)\,\psi^*_{j,k}(t)\,dt$.

The *wavelet coefficient* $w_{j,k}$ measures the signal content around time $2^{-j}k$ and frequency $2^j f_0$. The scaling coefficient $u_{j,k}$ measures the local mean around time $2^{-j}k$. The DWT (1) employs scaling coefficients only at scale $j_0$; wavelet coefficients at scales $j > j_0$ represent higher resolution approximation to the signal. Any filter bank DWT implementation produces all of the scaling coefficients $u_{j,k}$, $j > j_0$ as a natural byproduct [6].

*This work was supported by NSF, grant MIP–9457438, DARPA/AFOSR, grant F49620–97–1–0513, Texas Instruments, and the Rice Consortium for Computational Seismic Interpretation.

Email: choi@ece.rice.edu, richb@rice.edu

Internet: www.dsp.rice.edu

To keep the notation manageable in the sequel, we will adopt an abstract index scheme for the DWT coefficients: $u_{j,k} \to u_i$, $w_{j,k} \to w_i$.

We can easily construct 2-d wavelets from the 1-d $\psi$ and $\phi$ by setting $\mathbf{x} \equiv (x, y) \in \mathbb{R}^2$ and $\psi^{\mathrm{HL}}(\mathbf{x}) = \psi(x)\phi(y)$, $\psi^{\mathrm{LH}}(\mathbf{x}) = \phi(x)\psi(y)$, and $\psi^{\mathrm{HH}}(\mathbf{x}) = \psi(x)\psi(y)$. If we let $\Psi \equiv \{\psi^{\mathrm{HL}}, \psi^{\mathrm{LH}}, \psi^{\mathrm{HH}}\}$, then the set of functions $\{\psi_{j,\mathbf{k}} \equiv 2^j\psi(2^j\mathbf{x} - \mathbf{k})\}_{\psi\in\Psi, j\in\mathbb{Z}, \mathbf{k}\in\mathbb{Z}^2}$ forms an orthonormal basis for $L_2(\mathbb{R}^2)$; that is, for every $z \in L_2(\mathbb{R}^2)$, we have

$$z = \sum_{j>j_0, \mathbf{k}\in\mathbb{Z}^2, \psi\in\Psi} w_{j,\mathbf{k},\psi}\, \psi_{j,\mathbf{k}} + \sum_{\mathbf{k}\in\mathbb{Z}^2} u_{j_0,\mathbf{k}}\, \phi_{j_0,\mathbf{k}}, \qquad (2)$$

with $u_{j_0,\mathbf{k}} \equiv \int_{\mathbb{R}^2} z(\mathbf{x})\phi_{j_0,\mathbf{k}}(\mathbf{x})d\mathbf{x}$ and $w_{j,\mathbf{k},\psi} \equiv \int_{\mathbb{R}^2} z(\mathbf{x})\psi_{j,\mathbf{k}}(\mathbf{x})d\mathbf{x}$.

### 2.2 Hidden Markov tree model

The *compression property* of the wavelet transform states that the transform of many real-world signals consists of a small number of large coefficients and a large number of small coefficients. We can consider the collection of small wavelet coefficients as outcomes of a probability density function (pdf) with small variance. Similarly, the collection of large coefficients can be considered as outcomes of a pdf with large variance. Hence, the pdf $f_{W_i}(w_i)$ of each wavelet coefficient is well approximated by *Gaussian mixture model.* To each wavelet coefficient $W_i$, we associate a discrete hidden state $S_i$ that takes on values $m = 1, \ldots, M$ with probability mass function (pmf) $p_{S_i}(m)$. Conditioned on $S_i = m$, $W_i$ is Gaussian with mean $\mu_{i,m}$ and variance $\sigma^2_{i,m}$. Thus, its overall pdf is given by

$$f_{W_i}(w_i) \;=\; \sum_{m=1}^{M} p_{S_i}(m)\, f_{W_i|S_i}(w_i|S_i = m). \qquad (3)$$

We consider only the case of $M = 2$ in this paper; however, the Gaussian mixture model can provide an arbitrarily close fit to the actual $f_W(w)$ as $M > 2$.

To generate a realization of $W$ using the mixture model, we first randomly select a state variable $S$ according to $p_S(s)$ and then draw an observation $w$ according to $f_{W|S}(w|S = s)$. Although each wavelet coefficient $W$ is conditionally Gaussian given its state variable $S$, the wavelet coefficient has an overall non-Gaussian density due to the randomness of $S$.

HMT models are multidimensional mixture models in which the hidden states have a Markov dependency structure [4]. Once we model the marginal density of each wavelet coefficient as a Gaussian mixture model, the correlation between wavelet coefficients can be captured by specifying the joint pmf of the hidden states. To capture the "persistence" of large/small values of wavelet coefficients across scales, we can model the correlations between wavelet coefficients as a binary tree where each branch indicates the dependency between the connected coefficients. Although coefficients that are not connected by the binary tree model are also correlated, we ignore these dependencies to simplify the model.

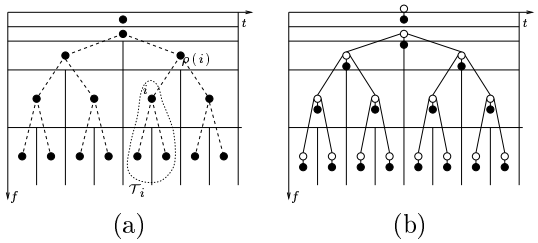



**Figure 1.** (a) Tiling of time-frequency plane by the atoms of the wavelet transform. Each box depicts the idealized support of a scaling atom (top row) or a wavelet atom (other rows) in time-frequency. The figure also illustrates our tree notation for indexing coefficients. (b) 1-d wavelet-domain Hidden Markov Tree (HMT) model. We model each coefficient as a Gaussian mixture with a hidden state variable. Black nodes represent wavelet coefficients; white nodes represent hidden mixture state variables. Connecting the states vertically across scale yields the HMT model.

Figure 1(a) shows the tiling of the time-frequency plane by a wavelet transform and the binary tree that models the dependencies between wavelet coefficients across scale. In order to describe the relationships between wavelet coefficients, we will use the notation $\rho(i)$ for the parent of node $i$. We also define $\mathcal{T}_i$ as the subtree of wavelet coefficients with root at node $i$, so that the subtree $\mathcal{T}_i$ contains coefficient $w_i$ and all of its descendants.

The HMT model is specified via the Gaussian mixture parameters $\mu_{i,m}, \sigma^2_{i,m}$, the transition probabilities $\epsilon^{mn}_{i,\rho(i)} = p_{S_i|S_{\rho(i)}}(m|S_{\rho(i)} = n)$, and the pmf $p_{S_1}(m)$ for the root node $S_1$. These parameters can be grouped into a model parameter vector $\mathbf{\Theta}$. We train the HMT to capture the wavelet-domain characteristics of the signals of interest using the iterative Expectation Maximization (EM) algorithm [4]. For a given set of training signals, the trained model $\mathbf{\Theta}$ approximates the joint pdf $f(\mathbf{w})$ of all wavelet coefficients.

In the HMT model, each wavelet coefficient $W_i$ is conditionally independent of all other random variables given its state $S_i$. Furthermore, given the parent state $S_{\rho(i)}$, the nodes $\{S_i, W_i\}$ are independent of the entire tree except for $S_i$'s descendants. The Markov structure of the model is on the states of the wavelet coefficients, not on the coefficients themselves (see Figure 1(b)).

The wavelet HMT model easily generalizes to 2-d us-

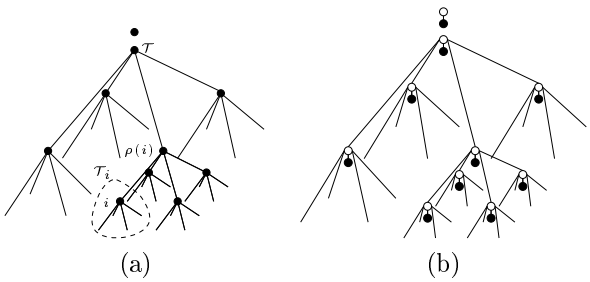



**Figure 2.** (a) Quadtree of 2-d wavelet coefficients for each subband, (b) 2-d wavelet-domain HMT model. Black nodes represent wavelet coefficients; white nodes represent hidden state variables.

ing a quadtree model to capture the dependencies between the wavelet coefficients, with each wavelet state connected to the four "child" wavelet states below it (see Figure 2). The EM algorithm for the 1-d HMT model in [4] can be used without modification if we interpret the parent-child relations between nodes appropriately for quadtrees.

# 3 Multiscale Segmentation using HMT

## 3.1 Multiscale classification

The key step in our segmentation algorithm is to classify dyadic blocks of the image at different scales based on trained HMT models. For classification, we use the principle of maximum likelihood detection.[1]

Given an image of $2^J \times 2^J$ pixels, we define the *dyadic squares* at scale $j$ to be the squares obtained by dividing the image into $2^j \times 2^j$ square regions of size $2^{J-j} \times 2^{J-j}$ pixels each for $j = 0, \ldots, J$. Denote each dyadic square as $D_i$, where $i$ is an abstract index, with $J(i)$ the scale of $D_i$. In the sequel we will use the Haar wavelet transform. Because each Haar wavelet coefficient is computed from the pixel values in a dyadic square, we have a one-to-one correspondence between the wavelet coefficients and the dyadic squares.

Although the dyadic squares at a certain scale are correlated, we assume that they are independent for the multiscale classification step. We capture the dependencies later by combining classification results from different scales.

Texture classification using the HMT is simple: First, we obtain wavelet-domain HMT models for the candidate textures by training HMTs on hand-segmented training images. Then, to classify a dyadic block, we compute the conditional likelihood of the corresponding subtree for each candidate texture model; the texture model maximizing the likelihood is chosen as the texture of the block. This likelihood computation is easily implemented using the HMT EM algorithm [4].

In a 2-d HMT model $\mathbf{\Theta}$, we have three quadtrees corresponding to three different subbands. Denote the three quadtree models as $\mathbf{\Theta}^{\mathrm{HH}}$, $\mathbf{\Theta}^{\mathrm{HL}}$ and $\mathbf{\Theta}^{\mathrm{LH}}$, respectively. For subtree $\mathcal{T}_i^{\mathrm{HH}}$ in subband $\mathbf{\Theta}^{\mathrm{HH}}$ corresponding to dyadic square $D_i$, we compute the conditional likelihood $\beta_i(m) = f(\mathcal{T}_i^{\mathrm{HH}}|S_i = m, \mathbf{\Theta}^{\mathrm{HH}})$ and conditional probability $p(S_i = m|\mathbf{w}, \mathbf{\Theta}^{\mathrm{HH}})$, where $\mathbf{w} = \{w_i\}$ is the collection of all wavelet coefficients in the subband HH. Then, the conditional likelihood $f(\mathcal{T}_i^{\mathrm{HH}}|\mathbf{\Theta}^{\mathrm{HH}})$ can be computed as

$$f(\mathcal{T}_i^{\mathrm{HH}}|\mathbf{\Theta}^{\mathrm{HH}}) = \sum_{m=1}^{M} \beta_i(m) p(S_i = m|\mathbf{w}, \mathbf{\Theta}^{\mathrm{HH}}). \quad (4)$$

For the HL and LH subbands, we can similarly compute the likelihoods $f(\mathcal{T}_i^{\mathrm{HL}}|\mathbf{\Theta}^{\mathrm{HL}})$ and $f(\mathcal{T}_i^{\mathrm{LH}}|\mathbf{\Theta}^{\mathrm{LH}})$, respectively.

There are several ways in which we can combine the likelihoods of the wavelet coefficients from different subbands. The simplest and most effective method we have found assumes that all three subbands are independent. Then, we simply multiply the three likelihood functions together to obtain the total likelihood $f(D_i|\mathbf{\Theta})$ of the dyadic square $D_i$. Another possibility ties the three root nodes of the quadtrees together and compute likelihoods for the combined tree.

For each node $i$, the dyadic square $D_i$ can be classified by finding the texture model $\mathbf{\Theta}$ for which the likelihood $f(D_i|\mathbf{\Theta})$ is maximized, producing a raw segmentation of the image down to $2 \times 2$ pixels image blocks. We call this segmentation a "raw" segmentation, because we do not use any relationship between segmentations across different scales. We expect this "raw" segmentation to be more reliable at coarse scales, because classification of large image blocks is more reliable due to their richness in statistical information. However, at coarse scales, the boundaries between different textures will not be captured accurately. At fine scales, the segmentation is less reliable, but boundaries are better captured.

When we compute the likelihoods of wavelet coefficient subtrees, we ignore the scaling coefficients. Thus, we do not take advantage of the information in the local brightness of the image. This is why we can classify only down to $2 \times 2$ blocks. We can obtain a pixel-level classification using the histogram of pixel brightness for each texture.

However, because we ignore the scaling coefficients, the local brightness levels of the texture regions do not affect the performance down to $2 \times 2$ scale. This is an advantage of the use of HMT model for texture segmentation. The HMT model captures only joint statistics

[1] We can also use the maximum a posteriori probability detection if we have prior pmfs of texture classes.

between pixels, and thus it is robust to the change of local average brightness. This is a desirable feature because the local brightness of an image often varies at different regions. We now propose a way to combine the classification results at different scales to obtain a final, reliable segmentation.

### 3.2 Context-based Bayesian interscale fusion

Because we obtain the multiscale classifications without incorporating the dependencies between different dyadic squares in a scale, the raw segmentation results often leave much to be desired. In order to obtain an accurate segmentation result, we need to capture the dependencies between dyadic squares.

To model the dependencies between dyadic squares, we consider the *labeling tree* (LT) that is a tree-structured graph where the nodes correspond to dyadic squares of the given image (see Figure 3(a)). Because the square under inspection is highly correlated with its parents and neighbors, the decisions (class labels) of these neighboring squares should influence the decisions. For example, if the parent square was classified as a certain class, this can provide prior information on the class of its children, since parent and children squares are likely to be of same class. The same intuition holds for neighboring squares.

Although the dependencies between squares can be modeled using a general probabilistic graph [4], the complexity of the algorithms to handle the graph becomes prohibitively high when the number of incorporated neighborhood nodes is large. In this paper, we propose a context-based method [5] to capture the dependencies that is easy to manipulate with minimal computational expense. To further simplify the algorithm, we specify the contexts causally in scale, where the contexts of nodes in a certain scale are determined based on the decisions in the previous coarse scale, and the final segmentation is obtained by a simple descent of the LT.

We define the context for dyadic square $D_i$ as the length-$P$ vector $\mathbf{V}_i \equiv [V_{i,1}, V_{i,2}, \ldots V_{i,P}]$ formed as a function of the likelihoods of other parent/neighboring dyadic squares (see Figure 3(a)). We condition the likelihood of $D_i$ on $\mathbf{V}_i$ to determine its class. The idea is for $\mathbf{V}_i$ to provide supplementary information to the labeling tree, so that given the context, we can treat each node of the labeling tree as independent.

Assume we have $L$ candidate classes of texture $c_i = 1, \ldots, L$. By conditioning the likelihood of dyadic square $D_i$ of labeling tree on $\mathbf{V}_i$, we have the context based mixture model for $D_i$:

$$f(D_i^j|\mathbf{V}_i, \mathbf{\Theta}_l) = \sum_{l=1}^{L} p_{C_i|\mathbf{V}_i}(c_i = l|\mathbf{v}_i) f(D_i|\mathbf{\Theta}_l), \quad (5)$$

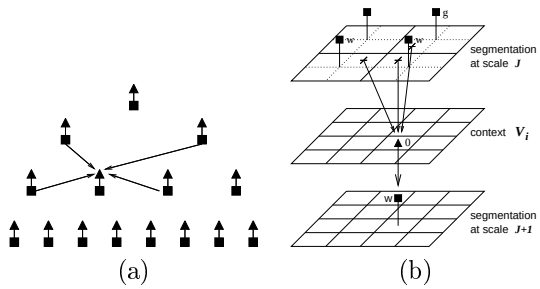



**Figure 3.** (a) Context-based labeling tree (LT) for segmentation. To each dyadic square (dark square), we augment a (triangle) context node. The context is a function of the other dyadic squares, (b) Context model used in simulations. For each dyadic square, we consider its parent node and two adjacent blocks of the parent. Depending on the classification of these 3 squares, the context $v_i$ is assigned such that $v_i = 1$ if all three are classified as wood (ground, for aerial photo), $v_i = -1$ if all are classified as grass (sea, for aerial photo), and $v_i = 0$ otherwise.

where the likelihoods $f(D_i|\mathbf{\Theta}_l) = f_{D_i|C_i}(D_i|c_i)$ are computed using HMT models for texture $l$, $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$ is the probability that node $i$ is of class $l$ given the context $\mathbf{V}_i$, and the summation is over all candidate textures for $c_i$. We assume that the pmfs of texture class $p_{C_i|\mathbf{V}_i}$ are constant at each scale (i.e., independent of $i$ if $J(i) = k$).

Denote the entire image as $D_1$. Let $k$ be the scale under consideration, and let $\mathbf{V}^k$ denote the collection of all contexts at scale $k$. The contextual Bayes classification starts by estimating the probabilities $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$ to maximize the likelihood of the entire image computed in scale $k$, assuming all subtrees with root node at scale $k$ are independent, given as

$$f(D_1|\mathbf{v}^k) = \prod_{i \text{ s.t. } J(i)=k} \sum_{l=1}^{L} p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i) f(D_i|\mathbf{\Theta}_l). \quad (6)$$

The values of $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$ are obtained by averaging over all the nodes in that scale. In practice, we do not specify $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$ directly, but rather specify $p_{\mathbf{V}_i|C_i}(\mathbf{v}_i|l)$ and apply Bayes rule. Thus, the actual probabilities that should be computed are $\epsilon_{i,n} \equiv p_{C_i}(n)$ and $\alpha_{i,\mathbf{v},n} \equiv p_{\mathbf{V}_i|C_i}(\mathbf{v}_i|n)$. The set of probabilities $\mathbf{P} = \{\epsilon_{i,n}, \alpha_{i,\mathbf{v},n}\}$ are computed using the EM algorithm below.

**EM Algorithm for Contextual Labeling Tree**

**Initialize:** Choose $\mathbf{P}^0$ and set $I = 0$.

The natural choice of $\mathbf{P}^0$ is the set of parameters obtained in the previous coarse scale.

**Expectation (E):** Given $\mathbf{P}^I$, calculate (Bayes rule)

$$p_{C_i|\mathbf{V}_i,D_i}(n|\mathbf{v}_i,D_i) = \frac{\epsilon_{i,n}\alpha_{i,v,n}f_{D_i|C_i}(D_i|n)}{\sum_{l=1}^{L}\epsilon_{i,l}\alpha_{i,\mathbf{v},l}f_{D_i|C_i}(D_i|l)}. \quad (7)$$

**Maximization (M):** Compute the elements of $\mathbf{P}^{I+1}$

$$\epsilon_{i,n} = \frac{1}{2^{J(i)}} \sum_{k \text{ s.t. } J(k)=J(i)} p_{C_k|\mathbf{V}_k,D_k}(n|\mathbf{v}_k,D_k), \quad (8)$$

$$\alpha_{i,\mathbf{v},n} = \frac{1}{\epsilon_{i,n}} \sum_{k \text{ s.t. } J(k)=J(i),\mathbf{v}_k=\mathbf{v}_i} p_{C_k|\mathbf{V}_k,D_k}(n|\mathbf{v}_k,D_k). \quad (9)$$

**Iterate:** Increment $I \to I+1$. Apply E and M until converged.

Once $\epsilon_{i,n}$ and $\alpha_{i,\mathbf{v},n}$ are computed, the context-based Bayes classification is performed by finding the class label to maximize $p_{C_i|\mathbf{V}_i,D_i}(n|\mathbf{v}_i,D_i)$.

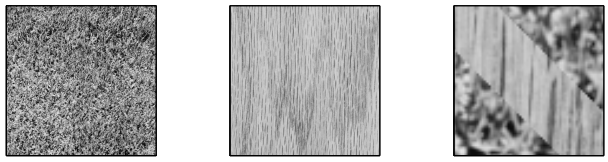
A pixel level segmentation requires a model for the pdf of a single pixel for each of the textures. The histogram of pixel intensities for each texture provides one simple model. For textures, it can be well approximated as a Gaussian mixture model. For the pixels of the training images, we fit a 2-state Gaussian mixture model to the pixel values, and we compute the likelihood of each pixel. Then, using the same algorithm to go down to pixel level segmentation using the results at $2 \times 2$ block scale, we obtain the final-pixel level segmentation.

## 4 Examples

In the first set of simulations, we segment a simple synthesized texture image consisting of a combination of "grass" and "wood" texture images from the USC Image Database.[2] To keep things simple, we considered only $64 \times 64$ images in the simulation. First we randomly selected five $64 \times 64$ blocks from the original "grass" and "wood" images to train respective HMT models. The training was performed with intra-scale tying [4] to avoid over-fitting. The test image was created from randomly chosen $64 \times 64$ grass and wood images. The mosaic test image is shown in Fig. 4(c); the $(i,j)$ elements with $|i-j| \leq 18$ correspond to wood texture and the remaining region is grass texture.

Inspired by the success of hybrid tree model in [3], we use a very simple context structure: the context of each dyadic square depends on the class labels of three squares in the previous coarser scale (see Fig. 3(b)). The dyadic square to be classified corresponds to a quadrant of its parent square. We use the class labels of the parent square and its two neighbors bordering on the quadrant in order to specify the context. The context $v_i$ used in the examples is a scalar ($P = 1$) taking three different values according to the class labels

[2] http://sipi.usc.edu/services.html



(a) (b) (c)

**Figure 4.** (a) Grass and (b) wood texture images from the USC image data base. (c) $64 \times 64$ mosaic test image to be segmented.

(a) (b) (c)

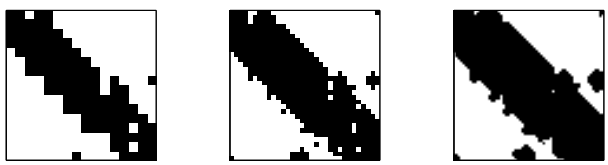
**Figure 5.** Raw segmentation of the grass-wood test image: (a) $4 \times 4$-block scale, (b) $2 \times 2$-block scale, (c) pixel-level segmentation. Classification accuracy increases with block size (coarser scale) because more statistical information is available.

of the three squares (parent and its two neighbors) as shown in Fig. 3(b).

Figure 5 illustrates the raw segmentation results before interscale fusion processing. At coarse scales, the classification is reliable, but the details at the texture boundaries are not well represented due to the large block size. The boundaries are better classified at finer scales, but we make many classification errors because of the paucity of statistical information in each small block. The pixel-level raw segmentation was obtained using a 2-state Gaussian mixture pixel brightness model.

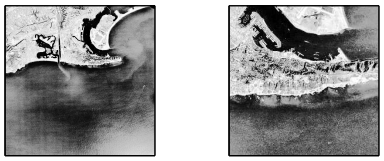
Figure 6 shows the segmentation results resulting from interscale fusion of the raw segmentation results. Comparing Figures 5 and 6, we observe an enormous increase in the segmentation accuracy.

In the next set of simulations, we segment a real aerial photograph from the USC Image Database. The original image is a $1024 \times 1024$ image with "ground" and "sea" regions as shown in Figure 7(a). Figure 7(b)



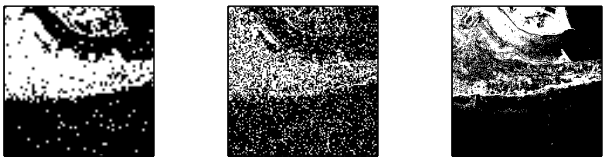
(a) (b) (c)

**Figure 6.** Segmentation results for grass-wood image using context-based Bayesian processing: (a) $4 \times 4$-block scale, (b) $2 \times 2$-block scale, (c) pixel-level segmentation.



(a) (b)

**Figure 7.** (a) Original aerial photo and (b) test subimage.



(a) (b) (c)

**Figure 8.** Raw segmentation of aerial photo: (a) 4 × 4-block scale, (b) 2 × 2-block scale, (c) pixel-level segmentation.

shows a 256 × 256 subblock of the full image that is to be segmented using our algorithm. First, the homogeneous "ground" and "sea" regions were obtained by hand-segmenting the original 1024 × 1024 image. These homogeneous regions were used to train two HMT models. Although the size of the images we segment is 256 × 256, the HMT models were trained using only 64 × 64 images. For the raw segmentation, we divide the 256 × 256 image into sixteen 64 × 64 blocks, and each block is classified using the HMT model. We used the same context model as in the synthetic grass/wood segmentation problem.

Figure 8 shows the raw segmentation results of the aerial photo. Again, the pixel-level raw segmentation was obtained using a 2-state Gaussian mixture model for pixel brightness of ground and sea textures. The pixel brightness model is clearly not appropriate in this case, because the overall brightness level in different portions of the full image varies considerably. As a result, the pixel level raw segmentation is not desirable.

Figure 9 illustrates the segmentation results after utilizing the coarse-to-fine interscale fusion. We observe very good segmentation results down to 2 × 2 block scale. The pixel level segmentation is again undesirable because the raw segmentation data at pixel level affects the final segmentation adversely.

## 5 Conclusions

In this paper, we have developed a new framework for texture segmentation based on wavelet-domain hidden Markov models. By concisely modeling the statistical behavior of textures at multiple scales and by combining segmentations at multiple scales, the algorithm produces a robust and accurate segmentation of

(a) (b) (c)

**Figure 9.** Segmentation of aerial photo using context-based Bayesian processing: (a) 4 × 4-block scale, (b) 2 × 2-block scale, (c) pixel-level segmentation.

texture images.

We believe the proposed segmentation algorithm can be applied to many different types of images, including radar/sonar images, medical images (CT/ultrasound), and document images (text/picture segmentation). To incorporate different characteristics of various images, the use of different wavelet bases and more complicated context models should be investigated. In addition, the analysis of detection errors for wavelet-domain HMT based classifiers remains future research topic.

## References


[1] M. Basseville et al., "Modeling and Estimation of Multiresolution Stochastic Processes," *IEEE Trans. Inf. Theory,* vol. 38, no. 2, pp. 766-784, Mar. 1992.

[2] C. H. Fosgate et al., "Multiscale Segmentation and Anomaly Enhancement of SAR Imagery," *IEEE Trans. Image Proc.,* vol. 6, no. 1, pp. 7-20, Jan. 1997.

[3] C. A. Bouman and M. Shapiro, "A Multiscale Random Field Model for Bayesian Image Segmentation," *IEEE Trans. Image Proc.,* vol. 3, no. 2, pp. 162-177, Mar. 1994

[4] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-Based Statistical Signal Processing Using Hidden Markov Models," *IEEE Trans. Signal. Proc.,* vol. 46, no. 4, pp. 886-902, Apr. 1998.

[5] M. S. Crouse and R. G. Baraniuk, "Contextual Hidden Markov Models for Wavelet-domain Signal Processing," in *Proc. 31st Asilomar Conference,* vol. 1, pp. 95-100, Nov. 1997.

[6] I. Daubechies, *Ten Lectures on Wavelets.* New York: SIAM, 1992.