Image Segmentation using Wavelet-domain Classification

Hyeokho Choi and Richard Baraniuk

Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, USA

ABSTRACT

We introduce a new image texture segmentation algorithm, HMTseg, based on wavelet-domain hidden Markov tree (HMT) models. The HMT model is a tree-structured probabilistic graph that captures the statistical properties of wavelet coefficients. Since the HMT is particularly well suited to images containing singularities (edges and ridges), it provides provides a good classifier for textures. Utilizing the inherent tree structure of the wavelet HMT and its fast training and likelihood computation algorithms, we perform multiscale texture classification at various scales. We then fuse these multiscale classifications using a Bayesian probabilistic graph to obtain a reliable final segmentation. Since HMTseg works on the wavelet transform of the image, it can directly segment wavelet-compressed images, without the need for decompression. We demonstrate the performance of HMTseg with synthetic, aerial photo, and document image segmentations.

1. INTRODUCTION

1.1. Image Segmentation

The image segmentation problem is to assign a *class label* to each pixel of an image based on the properties of the pixel and its relationship with its neighborhoods. The segmentation process is a joint detection and estimation of the class labels and the shapes of the regions with homogeneous statistical properties. A "good" segmentation should separate the image into simple regions with homogeneous properties. We call the homogeneous statistical property "texture." Recently, many authors have used Bayesian estimation techniques as a tool for estimating the region shapes and deciding their classes.[1,2] The segmentation is computed by estimating the best label for each pixel through Bayesian estimation of the shape and the class labels.

A number of Bayesian techniques have used region models to describe the joint statistics of the pixels in homogeneous regions. The Markov random field (MRF) model[3,4] has been extensively used for modeling the class labels for the pixels. MRFs model the local interactions between pixels and enable spatially local processing. The image is then segmented by (approximately) computing the maximum *a posteriori* (MAP) or maximum likelihood (ML) estimate of the pixel labels.

However, models capturing only local interactions have a limited ability to describe large-scale behavior. The model can be improved by incorporating more neighboring pixels, but this rapidly increases the complexity of the segmentation algorithm.

To capture the properties of each region of the image to be segmented, both the large- and small-scale behaviors should be utilized to properly segment both large, homogeneous regions and detailed boundary regions. Thus, it is natural to approach the segmentation problem using multiscale analysis. In this paper, we propose a multiscale image segmentation algorithm based on the *wavelet transform* of the image.

1.2. Statistical Models and Wavelets

Texture models play a fundamental rôle in image classification and segmentation. In *statistical modeling*, we regard the image $\boldsymbol{X}$ as a random realization from a family or distribution of images with joint probability density function (pdf) $f(\boldsymbol{x})$. Given $f(\boldsymbol{x})$, the standard statistical signal processing tasks, such as estimation, detection, and classification, are straightforward. However, the complete pdf is typically unavailable in practice or overly complicated, and so models are typically designed to focus on just the key image properties.

Transform-domain models are based on the idea that often a linear, invertible transform will "restructure" the image, leaving transform coefficients whose structure is "simpler" to model. Most real-world images, especially

---

Email: choi@ece.rice.edu, richb@rice.edu. Web: www.dsp.rice.edu. This work was supported by the National Science Foundation, grant MIP-9457438, DARPA/AFOSR, grant F49620-97-1-0513, Texas Instruments, and the Rice Consortium for Computational Seismic Interpretation.

texture images, are well characterized by their *singularity* (edge and ridge) structure. The wavelet transform provides a powerful transform domain for modeling real-world images containing singularities.[5]

The wavelet transform can be interpreted as a multiscale edge detector representing the singularity content of the image at multiple scales at three different orientations. Each singularity in the image is represented by a cascade of large wavelet coefficients across scale.[5] If the singularity is within the support of a wavelet basis function, the corresponding wavelet coefficient is large. Hence the wavelet coefficients at the singularity location tend to be large. Likewise, a smooth image region is represented by a cascade of small wavelet coefficients across scale. The wavelet transform of an $N \times N$ discrete image can be efficiently computed using a filter bank or lifting algorithm in $O(N^2)$ computations.[6,7]

The following salient features of wavelet transforms make the wavelet-domain statistical image processing attractive[5,6]:

W1. Locality: Each wavelet coefficient represents the image content localized in spatial location and frequency.

W2. Multiresolution: The wavelet transform analyzes the image at a nested set of scales.

W3. Energy Compaction: The wavelet transforms of real-world images tend to be sparse. A wavelet coefficient is large only if singularities are present within the support of the wavelet.

W4. Decorrelation: The wavelet coefficients of real-world images tend to be approximately decorrelated.

The Locality and Multiresolution properties (**W1**, **W2**) enable the wavelet transform to efficiently represent the local edge content of real-world images with large coefficients, resulting in the Compaction property (**W3**), because only a very small portion of a typical image contains edges. The Compaction and Decorrelation properties (**W3**, **W4**) simplify the statistical modeling of real-world images in the wavelet domain as compared with a direct spatial-domain modeling. Because most of the wavelet coefficients tend to be small, we need model only a small number of coefficients accurately. The Compaction (**W3**) of signal energy in wavelet domain can be restated in terms of a distinctive marginal distribution of the wavelet coefficients:

W5. NonGaussianity: The wavelet coefficients have peaky, heavy-tailed, nonGaussian marginal statistics.

The Decorrelation property (**W4**) inspires simple spatially localized modeling of the wavelet coefficients. There have been several approaches to model each wavelet coefficient independently with a nonGaussian marginal distribution.[8,9]

Although independent models are simple and easy to handle, modeling the residual dependencies between wavelet coefficients improves the model of real-world images considerably.[10,11] The relation between the singularities and the behavior of wavelet coefficients across scale indicates the following strong dependencies between the wavelet coefficients:

W6. Persistence and Clustering: Large/small values of wavelet coefficients tend to propagate across scales.[12]

This property tells us that the wavelet transforms of real-world images have a local dependency structure that should not be ignored.

Wavelet-domain statistical image models $f(\boldsymbol{w})$ have several distinct advances over pixel-domain models $f(\boldsymbol{x})$. Because the wavelet coefficients are approximately decorrelated, a simple dependency structure between the coefficients is sufficient to capture most of the joint statistics of the pixel values. In addition, the natural multiresolution structure of wavelet transform and its fast computational algorithm enable us to process the underlying image at a number of different scales with minimal expense. This is in stark contrast to other modeling frameworks, such as MRFs.

In this paper, we will focus on the *wavelet-domain hidden Markov tree (HMT)*.[10] The wavelet HMT models each nonGaussian wavelet marginal pdf as a 2-density Gaussian mixture with the hidden state signifying large/small magnitude of the coefficient. The persistence of wavelet coefficient magnitude across scale is modeled by linking the hidden states across scale in a Markov-1 tree. The parameters of the HMT model can be trained to match

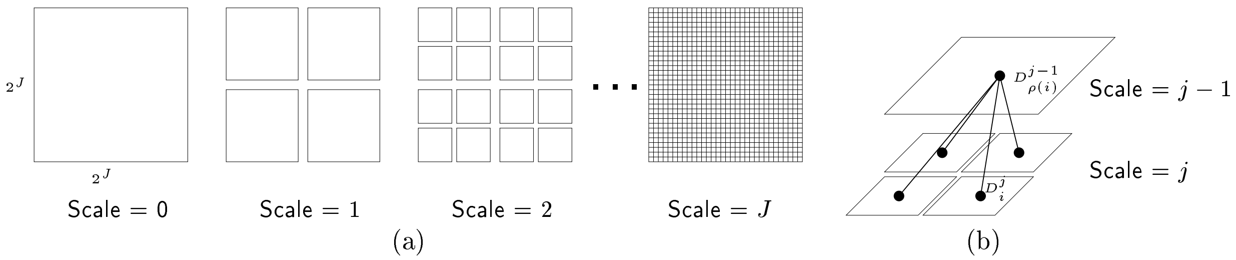

Figure 1. *(a) Image divided into dyadic squares at different scales. Each dyadic square can be associated with a subtree of Haar wavelet coefficients. (b) Quadtree structure of dyadic squares. The dyadic square $D^{j-1}_{\rho(i)}$ splits into four child squares at scale j.*

a set of training data using the Expectation-Maximization (EM) algorithm. The trained HMT then provides a good approximation of the joint pdf $f(\boldsymbol{w})$ of the wavelet coefficients and thus $f(\boldsymbol{x})$. Once trained, the likelihood computation $f(\boldsymbol{w})$ for a set of test data $\boldsymbol{x}$ is fast ($O(N^2)$) and straightforward.

In this paper, we will utilize wavelet HMT models for multiresolution image classification — the key step in our proposed multiscale segmentation algorithm.

1.3. Multiscale Segmentation Framework

To assign a class label to a given pixel, we classify a block of pixels that contains the pixel. Ideally, if the block contains only pixels of a single class and the classification of the block is correct, then the decision and assignment of labels will be in agreement. In general, however, the block will contain pixels from multiple classes. This makes the classification less reliable, even if the classification of each block is perfect. For instance, the assignment of class label is not reliable if the given pixel is at the boundary between two regions belonging to different classes.

Many previously proposed segmentation algorithms employ a *classification window* of some size with the hope that all the pixels in the window have the same class. A typical segmentation consists of the classification of each window of pixels followed by some post-processing.

Clearly the size of the classification window is crucial. A large window usually enhances the classification reliability (because many pixels provide rich statistical information) but simultaneously risks having pixels of different classes inside the window. Thus, using a large window produces accurate segmentations in large, homogeneous regions but poor segmentations along the boundaries between different classes. A small window reduces the possibility of having multiple classes in the window, but sacrifices classification reliability. Thus, a small window is more appropriate around the boundaries between different classes.

In multiscale segmentation, we use many classification windows of different sizes and then combine the results in the hope that we acquire the desirable features of both large and small classification windows.

In this paper, we use the *dyadic squares* of the image to implement classification windows of many different sizes. Given an image of size $2^J \times 2^J$, we define the dyadic squares at scale j to be the square pixel blocks 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 $j = 0, \ldots, J$ (see Figure 1(a)).

Because of the dyadic pyramid structure of the wavelet transform,[6] the wavelet coefficients have an inherent dyadic sampling structure. Using this dyadic structure of wavelet transforms and the wavelet HMT model, we can classify all dyadic squares at all scales, thus implementing many different classification window sizes. The resulting segmentation algorithm, which we christen *HMTseg*, is illustrated in Figure 2 and the algorithm below:

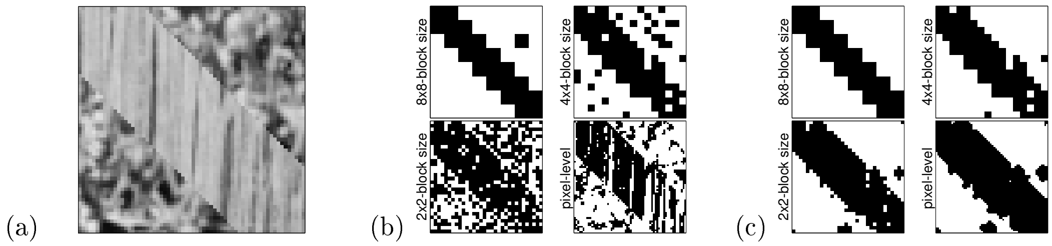

Figure 2. *(a) 64 × 64 grass/wood mosaic test image to be segmented. (b) Raw HMT-based multiscale classifications. (c) Segmentations using context-based interscale fusion. After interscale fusion, we obtain accurate segmentation down to pixel-level detail.*

HMTseg Algorithm

1. **Train wavelet-domain HMT models** for each texture using homogeneous training images (one set for each class).

2. **Compute likelihoods and perform raw classification.** Using the likelihood computation algorithm for the HMT models,[10] we compute the likelihoods of each dyadic square of the image at each different scale. A likelihood ratio test at each scale yields a set of ML *raw classifications* at a nested set of scales.

3. **Fuse raw multiscale classifications using a context-based labeling graph.** The labeling graph models the dependencies between different dyadic squares that are not utilized in the classification of each dyadic square. The Bayesian interscale fusion uses coarse-scale decisions to guide the finer-scale decisions.

HMTseg has several distinct advantages over existing techniques. To the best of our knowledge, this algorithm is the first attempt to use the *wavelet-domain* statistical modeling for multiscale image segmentation. This is in contrast to the multiscale segmentation approaches of Willsky et al.[13] and Bouman et al.,[14] which employ either multiscale models of only the scaling coefficients or the image class labels (for a more detailed comparison, see Choi et al.[15]).

2. MULTISCALE IMAGE SEGMENTATION — MOTIVATION

Each step of HMTseg is naturally motivated by a successive simplification of the ideal maximum likelihood segmentation method. In this section, we explain how we exploit the dyadic structure of the wavelet transform and HMT model to simplify the segmentation procedure.

2.1. Maximum likelihood image segmentation

In the areas of an image containing no sharp edges over large regions, the scene can often be characterized as exhibiting a consistent texture. The consistent structure of a single-textured region can be modeled by the joint statistics of the pixels in that region. The measurement of this consistency can be used to segment the image and classify its regions.

Consider the random field $\boldsymbol{X}$, the image to be segmented into regions of distinct statistical behavior. Individual pixels in $\boldsymbol{X}$ are denoted by X_s, where s is a member of a 2-D lattice of points S. The problem of ML image segmentation is to find the unknown label field $\boldsymbol{Y}$, where the labeled regions of $\boldsymbol{Y}$ correspond to the different regions of $\boldsymbol{X}$ with distinct stochastic properties, so that the likelihood of the image being generated by the model is maximized. Thus, ML segmentation requires the pdf for the image pixels of all possible shapes of regions for each texture.

The difficulty of the image segmentation arises because there are too many possibilities for the shape of a region, and it is impossible to specify the joint pixel pdf for all possible choices of partitioning the image into homogeneous texture regions. Moreover, even if the joint density could be specified for all possible shapes, the cost of computing the ML optimal segmentation would be prohibitive without a structured search algorithm.

In order to relieve the burden of specifying the joint statistics of all possible regions, we restrict the potential shapes of the regions for which we model the joint statistics. Here we will model the joint pdfs of the dyadic squares at different scales using wavelet HMT models. One advantage of considering the dyadic squares is that they are closely related to the Haar wavelet transform. Furthermore, the quadtree structure of the wavelet HMT enables an efficient computation of the likelihoods for ML classification.

Denote the dyadic squares at scale j by D_i^j, where i is an abstract index for the square. When we do not need to specify the scale of the dyadic square, we will drop the j. A square D_i^j is obtained by dividing a square at scale $j-1$ (the "parent" square) into four quadrants (the "child" squares). We denote the parent square by $D_{\rho(i)}^{j-1}$. The parent-child relations between squares at adjacent scales can be represented as a quadtree as shown in Figure 1(b). There is a one-to-one correspondence between the nodes of the quadtree and the dyadic squares at different scales.

2.2. Multiscale Classification and Fusion using a Contextual Labeling Graph

Given a tool for modeling the joint statistics of the dyadic squares and computing the likelihoods (such as the wavelet HMT), a classification at scale j can be found by deciding the class label for each square. The main information required for the classification of D_i^j is the likelihood $f(D_i^j|c_i)$ for each candidate class c_i. However, the likelihood itself may not be very reliable for the classification, especially for dyadic squares of small size, due to the paucity of the statistical information contained in the dyadic block. For reliable decisions, the dependencies between different dyadic squares at scale j should be taken into account, because neighboring dyadic squares are highly correlated with each other.

Modeling the dependencies between different dyadic squares requires extra effort. In general, we can model the dependencies between the dyadic squares at scale j directly as a *probabilistic graph*. However, the complexity of the graph model increases very rapidly as we consider more dependencies.

By modeling the dependencies "across" scale, we can model the dependencies between dyadic squares at a certain scale indirectly through the coarser scale squares. Because each dyadic square at a coarse scale contains many fine-scale descendants, the dependencies between different squares at the fine scale are modeled indirectly through their common ancestors at coarser scales. This naturally leads to a simple multiscale tree-structured dependency graph. We call this graph of dependencies the *labeling graph*. It motivates the multiscale likelihood computation and the fusion of the multiscale results to obtain the final segmentation. The fusion of the information at different scales leads to the statistical modeling of the dependencies between the dyadic squares at different scales.

A good approximation to the multiscale dependency structure is a tree-structured *Markov* probabilistic graph, where the dyadic squares at scale j depend only on the squares at the previous coarse scale $j-1$. Markov modeling of the dependencies leads to a scale-recursive decision of the class labels of the dyadic squares, where we classify the dyadic square D_i^j based on the its likelihood along with and information provided by the decisions in the previous coarser scale $j-1$.

If the dyadic square $D_{\rho(i)}^{j-1}$ was classified as a certain class, then it is very likely that its four children squares at scale j are of the same class, especially when j is large (at fine scales). Hence, we guide the decisions on the children squares based on the decision made on their parent square. This will tend to make the class labels of the 4 children the same unless the likelihood values for each of them strongly indicate otherwise. This reduces possible misclassifications due to slight perturbations of the likelihood values for the children squares. As well as the parent square, the neighbors of the parent can also guide the decision. The multiscale dependencies between the dyadic squares correspond to the fact that a typical real world image consists of several large, homogeneous regions.

Combining HMT-based classification and multiscale fusion results in a robust and accurate segmentation algorithm. Before we explain the details of the HMTseg, we first review its main components: the 2-D wavelet transform and wavelet HMT models.

3. THE 2-D WAVELET TRANSFORM

The wavelet transform represents the singularity content of a signal at multiple scales. In the context of the multiscale image classification/segmentation, the clearest interpretation of the wavelet transform results from the pyramidal multiscale construction a discrete image.[6]

We will focus on the simplest wavelet transform, that of Haar, in the development of the HMTseg algorithm. The construction of Haar wavelet coefficients of an image can be explained using the four 2-D wavelet filters

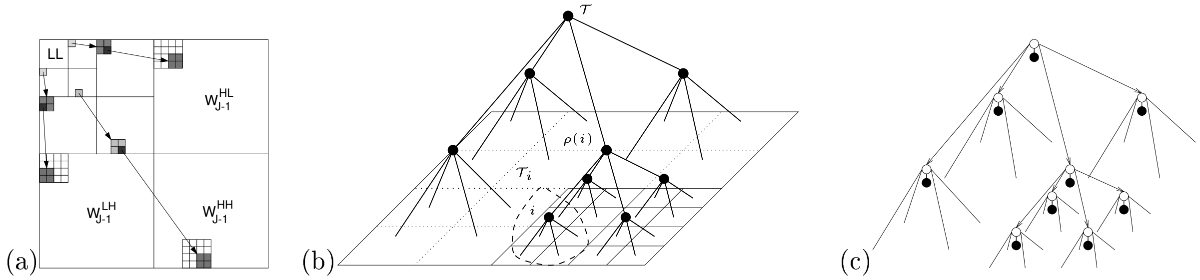

Figure 3. *(a) Parent-child dependencies of 2-D wavelet transform subbands. Each arrow points from the parent to its four children. Each parent has four children coefficients at the next finer scale in the same subband. (b) More detailed view of the quadtree structure in each subband. Each black node corresponds to a wavelet coefficient. The figure also illustrates our tree notation for indexing coefficients. (c) 2-D wavelet-domain HMT model. We model each coefficient as a Gaussian mixture with a hidden state variable (white node). Connecting the states vertically across scale yields the Hidden Markov Tree (HMT) model.*

$f_{\mathrm{LL}} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ (local smoother), $f_{\mathrm{LH}} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$ (vertical edge detector), $f_{\mathrm{HL}} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$ (horizontal edge detector), $f_{\mathrm{HH}} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ (diagonal edge detector). Let $u_J[k,l]$ denote an image of size $2^J \times 2^J$, with $1 \leq k,l \leq 2^J$. The pass-bands of the filters f_{LL}, f_{LH}, f_{HL}, and f_{HH} define the LL, LH, HL and HH subbands of the given image, respectively. The *scaling coefficients* $u_{J-1}[k,l]$, $1 \leq k,l \leq 2^{J-1}$ at scale $J-1$ in the LL subband are determined by placing the f_{LL} filter on the subimage $\{u_J[2k-1,2l-1], u_J[2k,2l-1], u_J[2k-1,2l], u_J[2k,2l]\}$ and computing the sum of the elements weighted by the filter coefficients. The resulting coefficients form a $2^{J-1} \times 2^{J-1}$ matrix. Through the same procedure but using the f_{LH}, f_{HL} and f_{HH} filters, we obtain the *wavelet coefficient* matrices $w^{\mathrm{LH}}_{J-1}[k,l]$, $w^{\mathrm{HL}}_{J-1}[k,l]$, and $w^{\mathrm{HH}}_{J-1}[k,l]$ at scale $J-1$. Each is of size $2^{J-1} \times 2^{J-1}$. The wavelet coefficient matrices at scale $J-1$ can be compactly stored as the $2^J \times 2^J$ matrix $\begin{bmatrix} u_{J-1} & w^{\mathrm{HL}}_{J-1} \\ w^{\mathrm{LH}}_{J-1} & w^{\mathrm{HH}}_{J-1} \end{bmatrix}$, which has the same size as the original image (see Figure 3(a)).

As is easily seen from the filter f_{LL}, the LL wavelet coefficient matrix $u_{J-1}[k,l]$ is a smoothed version of the original image that has been decimated by a factor of 2 in each direction. The filters f_{LH}, f_{HL}, and f_{HH} compute the highpass content of the image in the vertical, horizontal, and diagonal orientations, respectively, acting as *edge detectors* at the corresponding orientations. For example, a wavelet coefficient $w^{\mathrm{LH}}_{J-1}[k,l]$ is large if the 2×2 image block where the filter was placed contains a vertical edge and is small if the block does not contain an edge. Similarly, the magnitudes of the wavelet coefficients in the HL and HH indicate the presence of horizontal and diagonal edges at each spatial location.

This construction can be iterated on the coarse-scale image $u_{J-1}[k,l]$, producing the coefficients at the scale $J-2$. For a $2^J \times 2^J$ image, this procedure can be iterated up to J times, producing the coefficients from scales $0 \leq j \leq J$, as shown in Figure 3(a).

This iterative computation of each Haar wavelet coefficient from a 2×2 block in a finer-scale image leads naturally to a quadtree structure of the wavelet coefficients in each subband, as illustrated in Figure 3(a) and (b).[16] The root node of the quadtree corresponds to the coarsest-scale wavelet coefficient; as we move down the tree, the fine scale coefficients become more and more spatially localized.

To keep the notation manageable in the sequel, we will adopt an abstract index system for the DWT coefficients: Associate $w_j[k,l] \to w_i$, with $J(i)$ the scale of the coefficient i. We will also use $\mathbf{w}$ to denote the vector of all wavelet coefficients. In order to describe the relationships between the wavelet coefficients, we will use notation $\rho(i)$ for the parent and define $\mathcal{T}_i$ to be the subtree of wavelet coefficients with root node i, that is $\mathcal{T}_i$ contains the coefficient w_i and all of its descendants (see Figure 3(b)).

With the 2-D Haar wavelet transform, we can build a one-to-one correspondence between each wavelet coefficient in each subband at scale j and the dyadic squares of size $2^j \times 2^j$. For each subtree $\mathcal{T}_i$ with the coefficient w_i at the root node, we can find the dyadic square of image pixels D_i^j so that the subtree of wavelet coefficients is determined by the pixel values belonging to D_i^j.

Although longer wavelet filters are more appropriate for representing smooth signals, the Haar wavelet is more appropriate for our purpose of classifying dyadic blocks, due to its clear connection with the dyadic squares. We will see that it is more than sufficient to represent the singularity contents at multiple scales for the purpose of classification and segmentation of textures in the HMTseg algorithm.

4. WAVELET-DOMAIN HIDDEN MARKOV TREE MODEL

4.1. Modeling the nonGaussian marginal distribution (W5)

The Compaction property **W3** of the wavelet transform indicates that the transforms of many real-world signals consist of a small number of large coefficients and a large number of small coefficients. We can consider the collection of large wavelet coefficients as the outcomes of a probability density with large variance. Similarly, the collection of small coefficients can be considered as the outcomes of a density with small variance. Hence, the pdf $f_{W_i}(w_i)$ of each wavelet coefficient is well approximated by a 2-density *Gaussian mixture model.** To each wavelet coefficient W_i, we associate a discrete hidden state S_i that takes on values $m = \mathsf{S},\mathsf{L}$, signifying the small and large variance, 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

$$f_{W_i}(w_i) \;=\; \sum_{m=\mathsf{S},\mathsf{L}} p_{S_i}(m)\, f_{W_i|S_i}(w_i|S_i = m), \tag{1}$$

where $f_{W_i|S_i}(w_i|S_i = m) \sim N(\mu_{i,m}, \sigma^2_{i,m})$.

4.2. Modeling the key dependencies (W6)

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 probability mass function of the hidden states.

Although a local dependency modeling in wavelet domain is accurate thanks to the approximate decorrelation of wavelet coefficients, further simplification of the dependency structure is required for the tractability of the model. In the 2-D wavelet *hidden Markov tree* (HMT) model, we make the following two simplifications:

S1. The three subbands of the 2-D wavelet transform are independent.

S2. The dependency structure of wavelet coefficients in each subband has a quadtree structure as shown in Figure 3(b).

HMT models are directed probabilistic graphs of hidden mixture states in which the hidden states have a Markov dependency structure.[10] The quadtree dependency structure (**S2**) is appropriate for capturing the persistence of large/small wavelet coefficients, the key dependencies between wavelet coefficients of singularity-rich data. For each parent-child pair of hidden states $\{S_{\rho(i)}, S_i\}$, the state transition probabilities $\epsilon^{mn}_{i,\rho(i)}$ for $m, n = \mathsf{S},\mathsf{L}$ represent the probability that w_i is small/large given the parent coefficient $w_{\rho(i)}$ is small/large. For each i, we thus have the state transition probability matrix $\begin{bmatrix} \epsilon^{\mathsf{SS}}_{i,\rho(i)} & \epsilon^{\mathsf{LS}}_{i,\rho(i)} \\ \epsilon^{\mathsf{SL}}_{i,\rho(i)} & \epsilon^{\mathsf{LL}}_{i,\rho(i)} \end{bmatrix} = \begin{bmatrix} \epsilon^{\mathsf{SS}}_{i,\rho(i)} & 1-\epsilon^{\mathsf{LL}}_{i,\rho(i)} \\ 1-\epsilon^{\mathsf{SS}}_{i,\rho(i)} & \epsilon^{\mathsf{LL}}_{i,\rho(i)} \end{bmatrix}$.

The quantity $\epsilon^{\mathsf{SS}}_{i,\rho(i)}$ represents the probability that small coefficients persist from $W_{\rho(i)}$ to W_i. For a typical real-world image, we expect that $\epsilon^{\mathsf{LL}}_{i,\rho(i)}$ and $\epsilon^{\mathsf{SS}}_{i,\rho(i)}$ to be large due to the Persistence property (**W5**) of large/small wavelet coefficients across scale. The quadtree dependency structure also indirectly captures some clustering property of the coefficients within scale. Although coefficients that are not connected by the quadtree model are also correlated, we ignore those dependencies to simplify the training of the model.[10]

Figure 3(b) shows the quadtree structure of hidden states that models the dependencies between wavelet coefficients across scale. The above simplifying assumptions (**S1**,**S2**) yield a quadtree Markovian structure in the HMT.

It is important to emphasize that the Markov structure of the HMT is on the states of the wavelet coefficients and not on the coefficients themselves. In the HMT model, each wavelet coefficient W_i is conditionally independent

*We can use more than 2 mixture densities to provide a fit to the actual $f_W(w)$ to any desired fidelity.

of all other random variables given its state S_i. Furthermore, given the parent state $S_{\rho(i)}$, the pair of nodes $\{S_i, W_i\}$ are independent of the entire tree except for S_i's descendants.

In addition to modeling the wavelet coefficients, we can separately model the scaling coefficient using a mixture density.[10] However, for our image segmentation purpose, we intentionally ignore the scaling coefficients. Because the scaling coefficients correspond to the local averages of the pixel values, we thus build the statistical models for the image textures that are independent of the local brightness level. This is a desirable feature for many image segmentation applications, because even in a region having homogeneous statistical properties, the local brightness level often varies in different parts of the region.

In the HMT, there are six parameters for each wavelet coefficient: the means and variances of the mixture densities $\mu_{i,m}, \sigma^2_{i,m}$ and the state transition probabilities of the hidden states $\epsilon^{mn}_{i,\rho(i)}$, $m, n = \mathsf{S}, \mathsf{L}$. These parameters plus the pmf for the state of the root node $p^m_{S_1}$ can be grouped into a model parameter vector $\boldsymbol{\Theta}$.

The large number of parameters in a general HMT make model training difficult when the amount of training data is small. Since wavelet coefficients at a certain scale exhibit similar statistical behavior at different spatial locations, we can often assume that the parameters are the same for all nodes in the same scale. This *tying* of nodes reduces the number of model parameters considerably (from $6N^2 + 1$ to $6J + 1$), which reduces the risk of overfitting the model.[10,17]

A 2-D wavelet HMT model $\mathcal{M}$ consists of three Markovian quadtree structured probabilistic graphs corresponding to the three wavelet subbands. Denoting the three model parameter vectors for the HH, HL and LH subbands as $\boldsymbol{\Theta}^{\mathrm{HH}}, \boldsymbol{\Theta}^{\mathrm{HL}}$, and $\boldsymbol{\Theta}^{\mathrm{LH}}$, respectively, we have $\mathcal{M} := \{\boldsymbol{\Theta}^{\mathrm{HH}}, \boldsymbol{\Theta}^{\mathrm{HL}}, \boldsymbol{\Theta}^{\mathrm{LH}}\}$.

4.3. Training and likelihood computation

We can *train* the wavelet HMT model to best match a set of training data in the ML sense. The iterative HMT Expectation Maximization (EM) algorithm finds the "best" set of model parameters $\boldsymbol{\Theta}$ for given set of training data.† The Expectation step defines the likelihood surface based on the current state of the model. Then, the Maximization step updates the model parameters to maximize the likelihood. Iteration of the two steps is guaranteed to converge to a $\boldsymbol{\Theta}$ that locally maximizes the likelihood.[17] Each EM iteration consists of an up/down sweep through the tree ($O(N^2)$ cost for an $N \times N$ image). Once trained, the HMT provides an excellent approximation to the full joint pdf of the wavelet coefficients $f(\boldsymbol{w})$.

Given a set of 2-D HMT model parameters $\mathcal{M}$ and an image, we can also compute the *likelihood* $f(\boldsymbol{w}|\mathcal{M})$ that the image was generated by the model. Thanks to the dyadic multiscale structure of the wavelet transform and HMT, we can obtain the likelihoods of all dyadic squares of the image simultaneously by a single upward sweep through the tree (a fast $O(N^2)$ algorithm). In order to classify a dyadic square, we will compute the likelihood of the corresponding subtree of wavelet coefficients under the HMT models for each candidate texture class; the class giving the maximum likelihood is chosen as the texture of the dyadic square.

Let us provide some more details. Suppose we are given a trained 2-D HMT model $\mathcal{M}_c = \{\boldsymbol{\Theta}^{\mathrm{HH}}_c, \boldsymbol{\Theta}^{\mathrm{HL}}_c, \boldsymbol{\Theta}^{\mathrm{LH}}_c\}$ for each class c and a set of wavelet coefficients $\boldsymbol{w}$ to be segmented. For $\mathcal{T}^{\mathrm{HH}}_i$, the subtree of wavelet coefficients with root node i in the HH subband, we compute the conditional likelihood $\beta_i(m) = f(\mathcal{T}^{\mathrm{HH}}_i | S_i = m, \boldsymbol{\Theta}^{\mathrm{HH}}_c)$ using a single up sweep on the quadtree. Then, the likelihood $f(\mathcal{T}^{\mathrm{HH}}_i | \boldsymbol{\Theta}^{\mathrm{HH}}_c)$ can be computed as

$$f(\mathcal{T}^{\mathrm{HH}}_i | \boldsymbol{\Theta}^{\mathrm{HH}}_c) = \sum_{m=\mathsf{S},\mathsf{L}} \beta_i(m) p_{S_i}(m), \tag{2}$$

where $p_{S_i}(m) = p(S_i = m | \boldsymbol{\Theta}^{\mathrm{HH}}_c)$ is the state probability obtained when the HMT is trained. For the HL and LH subbands, we can similarly compute the likelihoods $f(\mathcal{T}^{\mathrm{HL}}_i | \boldsymbol{\Theta}^{\mathrm{HL}}_c)$ and $f(\mathcal{T}^{\mathrm{LH}}_i | \boldsymbol{\Theta}^{\mathrm{LH}}_c)$, respectively.

Under the independent subband assumption (**S1**), the product of the three likelihood values yields the total likelihood $f(\mathcal{T}^{\mathrm{HH}}_i, \mathcal{T}^{\mathrm{HL}}_i, \mathcal{T}^{\mathrm{LH}}_i | \mathcal{M}_c)$ of the collection of wavelet coefficients $\{\mathcal{T}^{\mathrm{HH}}_i, \mathcal{T}^{\mathrm{HL}}_i, \mathcal{T}^{\mathrm{LH}}_i\}$ corresponding to the dyadic square D^j_i. Thus, the overall likelihood of D^j_i is given by

$$f(D^j_i | \mathcal{M}_c) = f(\mathcal{T}^{\mathrm{HH}}_i | \boldsymbol{\Theta}^{\mathrm{HH}}_c) f(\mathcal{T}^{\mathrm{HL}}_i | \boldsymbol{\Theta}^{\mathrm{HL}}_c) f(\mathcal{T}^{\mathrm{LH}}_i | \boldsymbol{\Theta}^{\mathrm{LH}}_c). \tag{3}$$

†We can use the 1-D HMT EM algorithm from Crouse et al.[10] in 2-D without modification if we interpret the parent-child relationships between nodes appropriately for quadtrees. For a general theory of probabilistic graphs and training algorithms, see Frey.[18]

5. MULTISCALE SEGMENTATION USING HMT MODELS

We now provide more details of the HMTseg algorithm first set out in Section 1.3.

5.1. Training of the HMT models

To classify different textures, we require training data from each texture to train the corresponding HMT models. We assume that homogeneous texture images are available to train a model for each texture. We obtain these training images either by picking out the homogeneous regions of the given image or from a set of completely different images having homogeneous regions representative of the candidate textures.

Suppose we have P images of each homogeneous texture $c \in \{1, \ldots, L\}$. We then train a 2-D wavelet HMT model $\mathcal{M}_c$ for each texture c to capture the properties of the wavelet coefficients of the texture. When the number of training images P is small, we use the intra-scale tying to avoid overfitting of the model.[10]

5.2. Maximum likelihood classification

With trained HMT models in hand for each class, the simple one-to-one correspondence between the dyadic squares and the Haar wavelet coefficients enables us to use the HMT as a multiscale classifier.

To classify each dyadic square, we use the ML classification principle.[‡] The classification at each scale is accomplished by comparing the likelihoods $f(D_i^j|\mathcal{M}_1)$ through $f(D_i^j|\mathcal{M}_L)$ for every node i, based on Eqs. (2) and (3). For each i, the dyadic square D_i^j of pixels is classified as the texture that maximizes the likelihood. The result is a "raw" classification of the image down to 2×2 image blocks.

We refer to this as a raw classification, because we do not exploit any possible relationships between the classifications at different scales. We expect the raw classification to be more reliable at coarse scales, because the classification of larger image blocks is more reliable due to its richness of statistical information. However, at a coarse scale, the boundaries between two different textures cannot be represented accurately. At fine scales, the segmentation will be less reliable, but with potentially more accurate boundaries.

Because the models for the finest 2-D Haar wavelet coefficients characterize the joint statistics of 2×2 blocks of the image, we obtain raw classifications down to only 2×2 blocks.[§] In order to obtain a pixel-level segmentation, we must model the brightness of the pixels in each class. Then, pixel brightness information can be fused with the raw multiscale segmentation using the interscale fusion algorithm explained in Section 5.3 without adding significant overhead to the algorithm.

5.3. Context-based Interscale Fusion

To obtain accurate segmentations, we combine the raw multiscale classification results by considering the dependencies between the dyadic squares across scale. The reliable classifications at coarse scales will help reduce the decision errors at fine scales inside homogeneous regions.[¶].

5.3.1. Labeling graph of dyadic squares

As motivated in Section 2.2, we will model the dependencies between the dyadic squares by building a *labeling graph*, where each node represents a dyadic square. While the three trees in the HMT model the joint statistics of the pixels within each dyadic square, the labeling graph models the *dependencies between different dyadic squares.* At each node of the labeling graph, we store the statistical information of the corresponding dyadic square as a vector. These *node state* vectors are jointly determined based on the data available at each node and the statistical information passed from the neighboring nodes. The purpose of the labeling graph is to decide the unknown class label for each node, based on the statistical information at each node, available through the state vectors. Figure 4(a) shows a generic example of the type of statistical graph we will use to represent the dependencies. For a complete description of the statistical information at each node, the state vectors must be continuous quantities and have high dimensionality.

At each node of the labeling graph, we also assign a random variable $c_i \in \{1, \ldots, L\}$, representing the unknown class of the corresponding dyadic square. Consider the decision process for node i of the graph, which corresponds

[‡]We can also use MAP classification if the appropriate prior distributions are available.

[§]Although the collection of all wavelet coefficients contains the same information as the original image, the subband independence assumption (**S1**) used to build the 2-D HMT models limits the classification down to 2×2 blocks.

[¶]For more information, see Choi et al.[15]

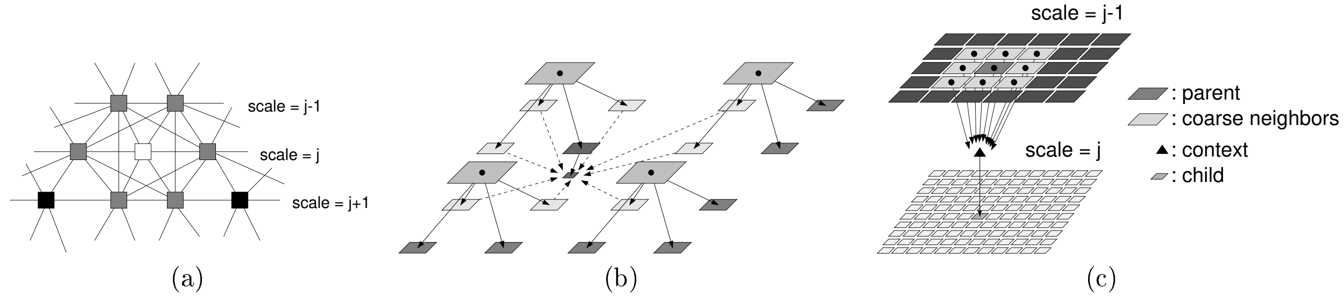

Figure 4. *(a)* Generic labeling graph *to model the dependencies between dyadic squares. The (white) dyadic square D_i^j is connected to the (gray) squares in ∂D_i^j. The dependency is modeled by conditional probabilities between squares. (b) Simplified* directed labeling tree *with causal dependencies in scale. Each dyadic square is dependent only on the 9 closest squares at the next coarser scale only. The solid lines represent the dependencies between parent and children. The dotted lines represent the dependencies on the coarse scale neighbors. (c) Even further simplified* context-based labeling tree. *To each dyadic square (child square in the figure), we augment a context node (black triangle). The context is a function of the statistical information of the dyadic squares at the next coarser scale. For the examples in Section 6, the context is a length-2 vector containing the label of the parent square and the majority vote of the class labels of the 9 coarse scale squares.*

to the classification of the dyadic square D_i. Denote the statistical information passed to node i from the connected nodes by ∂D_i. That is, ∂D_i is the collection of node state vectors for the nodes connected to the node i. We also use the same notation ∂D_i to denote the set of nodes connected to D_i, when there is no chance of confusion.

We are interested in the conditional probability $p(c_i|D_i, \partial D_i)$. After computing the conditional probabilities, we label D_i as the class l that maximizes $p(c_i = l|D_i, \partial D_i)$. Using the rule of Bayes (ruB), we can write the conditional probability as

$$p(c_i|D_i, \partial D_i) = \frac{f(D_i|c_i, \partial D_i)p(c_i|\partial D_i)}{f(D_i|\partial D_i)}. \tag{4}$$

Because $f(D_i|\partial D_i)$ is not dependent on c_i, the label of D_i is determined as

$$l_i = \arg\max_l p(c_i = l|D_i, \partial D_i) = \arg\max_l f(D_i|c_i = l, \partial D_i)\, p(c_i = l|\partial D_i). \tag{5}$$

With an additional assumption that D_i is independent of ∂D_i given c_i, we have $f(D_i|c_i, \partial D_i) = f(D_i|c_i)$, and we can write $l_i = \arg\max_l f(D_i|c_i = l)p(c_i = l|\partial D_i)$. Because we compute the likelihoods $f(D_i|c_i = l) = f(D_i|\mathcal{M}_l)$ using wavelet-domain HMT models, the remaining task is to compute the probabilities $p(c_i|\partial D_i)$. These are precisely the quantities encoded in the labeling graph.

5.3.2. Simplifying the labeling graph

For a general labeling graph, the difficulty of specifying the conditional probabilities $p(c_i|\partial D_i)$ stems from the following:

D1. All of the conditional probability functions (or the node states) must be estimated jointly.

D2. The dimension of the data ∂D_i is high ("curse of dimensionality").

D3. The data ∂D_i are *continuous* quantities.

Due to the above reasons, a reliable estimation (training) of the fusion parameters require a large amount of training data. To simplify the modeling problem, we simplify the structure of the graph by making several assumptions.

The requirement for the joint estimation (**D1**) can be relieved by introducing the directions of dependency relations, that is, by modeling the dependencies as a *directed* graph. Following the intuition behind a simplified labeling graph considered in Section 2.2, we use a directed labeling graph with a causal, Markov dependency across scale. Figure 4(b) shows an example of the labeling graph we employ. The arrows indicate the direction of dependencies. Due to the multiscale nesting of the dyadic squares, we can build a tree-structured directed graph of dyadic squares. We call this directed labeling graph the *labeling tree*.

On the scale-causal, Markov labeling tree, we assume that the decision on the dyadic square D_i^j is dependent only on the previous coarse scale information, i.e., ∂D_i^j is a set of dyadic squares at scale $j-1$ (causality in scale). Furthermore, we assume that given the data at scale $j-1$, that is D_i^{j-1}'s, the data at scale j are independent of all other scales (Markov in scale). With these simplifications of the graph, the joint estimation of all the graph model parameters is not required; rather we can estimate the graph parameters each scale at a time as we go down the labeling tree to finer scales.

The difficulty (**D2**) of ∂D_i^j being high dimensional can be partly relieved by restricting ∂D_i^j to the set of dyadic squares at scale $j-1$ having high correlation with D_i. For example, we can restrict ∂D_i^j to the parent square $D_{\rho(i)}^{j-1}$ and its neighbors. The dimension of ∂D_i can be further reduced by introducing the concept of the *context*[19] that is also used to overcome the difficulty (**D3**). To simplify the specification of the conditional probabilities $f(c_i|\partial D_i)$, we can define a function $g(\partial D_i)$, so that we have an approximation

$$f(c_i|\partial D_i) \approx f(c_i|g(\partial D_i)). \tag{6}$$

If we choose the function g properly, we can simplify the specification and estimation of the conditional probabilities enormously. Let's define $\mathbf{V}_i = g(\partial D_i)$ as the *context* for the dyadic square D_i. The idea is that if $\mathbf{V}_i$ summarizes the data ∂D_i properly, the approximation in Eq. (6) is valid. Often, we choose $\mathbf{V}_i$ as a discrete vector having only finitely many possible values, while it characterizes the key features of ∂D_i that affect the statistics of c_i.

The use of the context $\mathbf{V}_i$ instead of using ∂D_i relieves the difficulties (**D2**) and (**D3**), because $\mathbf{V}_i$ has a much lower dimension or takes only finitely many possible values. These simplifications of the modeling alleviates the requirement for a large amount of the training data, and we can estimate the dependency parameters without any extra training data, leading to an empirical extraction of the fusion parameters based on the given image to be segmented

5.3.3. Contextual labeling tree

We define the context for dyadic square D_i as a length-P vector $\mathbf{V}_i := [V_{i,1}, V_{i,2}, \ldots V_{i,P}]$ formed as a function of the likelihoods or the class labels of other parent/neighboring dyadic squares (see Figure 4(c)). 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 the each node of the labeling tree as independent.

By conditioning the likelihood of dyadic square D_i^j of labeling tree on $\mathbf{V}_i$ (with the added assumption that $\mathbf{V}_i$ and D_i^j are independent given the class label c_i), we have the context-based mixture model for D_i^j:

$$f(D_i^j|\mathbf{V}_i) = \sum_{l=1}^{L} p_{C_i|\mathbf{V}_i}(c_i = l|\mathbf{v}_i) f(D_i^j|\mathcal{M}_l) \tag{7}$$

where the likelihoods $f(D_i|\mathcal{M}_l)$ are computed using the HMT model for texture l, $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$ is the probability that the 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 $J(i) = j$).

We compute the probabilities $p_{C_i|\mathbf{V}_i}$ empirically for the image to be segmented based on the likelihoods computed using the HMT models. Denote the entire image as D_0. Let k be the scale under consideration. And, let $\mathbf{v}^j$ denote the collection of all contexts at scale j. The contextual Bayes classification starts by estimating the probabilities $p_{C_i|\mathbf{V}_i}(n|\mathbf{v}_i)$ to maximize the likelihood of the entire image computed in scale j, assuming all subtrees with root node at scale j are independent, given as

$$f(D_0|\mathbf{v}^j) = \prod_{J(i)=j} \sum_{l=1}^{L} p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i) f(D_i^j|c_i = l). \tag{8}$$

The values of $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$ are obtained by averaging over all the nodes in that scale.

For proper estimation of the context probabilities $p_{C_i|\mathbf{V}_i}(l|\mathbf{v}_i)$, the accuracy of the raw classification that is based on the likelihoods $f(D_i^j|c_i = l)$ should be high, which is true at scales coarse enough.

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 $p_{C_i}(n)$ and $p_{\mathbf{V}_i|C_i}(\mathbf{v}_i|n)$. Because we assume these probabilities to be constant at each scale j, the probabilities we compute are $e_{j,n} = p_{C_i}(n)$ and $\alpha_{j,\mathbf{v},n} = p_{\mathbf{V}_i|C_i}(\mathbf{v}_i|n)$, $\forall\, i$ such that $J(i) = j$. The set of probabilities $\mathbf{P} = \{e_{j,n}, \alpha_{j,\mathbf{v},n}\}$ are computed using the EM algorithm below.

EM Algorithm for the Contextual Labeling Tree

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

$$p_{C_i|\mathbf{V}_i,D_i^j}(n|\mathbf{v}_i, D_i^j) = \frac{e_{J(i),n}\alpha_{J(i),v,n} f_{D_i^j|C_i}(D_i^j|n)}{\sum_{l=1}^{L} e_{i,l}\alpha_{i,\mathbf{v},l} f_{D_i^j|C_i}(D_i^j|l)}. \tag{9}$$

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

$$e_{j,m} = \frac{1}{2^j}\sum_k p_{C_k|\mathbf{V}_k,D_k^j}(n|\mathbf{v}_k, D_k^j), \tag{10}$$

$$\alpha_{j,\mathbf{v},n} = \frac{1}{e_{j,n}} \sum_{k \text{ s.t. } \mathbf{v}_k=\mathbf{v}_i} p_{C_k|\mathbf{V}_k,D_k^j}(n|\mathbf{v}_k, D_k^j). \tag{11}$$

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

Once $e_{j,n}$ and $\alpha_{j,v,n}$ are computed, the context-based Bayes classification is performed by finding class label so that $p_{C_i|\mathbf{V}_i,D_i^j}(n|\mathbf{v}_i, D_i^j)$ is maximized.

The choice of a good context model is crucial to the performance of HMTseg. We have a trade-off between the complexity of the context and the accuracy of the estimated model parameters.

Inspired by the success of hybrid tree model used to describe the multiscale dependency structure of the class labels in Bouman et al.,[14] we consider a very simple context structure where the context of a dyadic square depends on the class labels of nine squares in the previous coarse scale. For the decision of the class label of D_i^j, we use its parent $D_{\rho(i)}^{j-1}$ and its 8 neighbors (Fig. 4(c)) to specify the context. The context $g(\partial D_i^j)$ is defined as a length-2 vector $\mathbf{V}_i$. The first element $V_{i,1}$ is the class label of the parent square $D_{\rho(i)}^{j-1}$ taking L different values. The second element $V_{i,2}$ is the majority of the labels for the nine squares consisting of $D_{\rho(i)}^{j-1}$ and its eight neighbors. Hence, if there are L different class labels, we have the context $\mathbf{V}_i$ can take L^2 different values. For the three segmentation examples illustrated in Section 6, we used this simple context structure.

To obtain a pixel-level segmentation, the probability distribution of a single pixel for each of the textures is necessary. The pdf can be modeled from the histogram of the pixel brightness for each texture. For textures, it can be well approximated as a Gaussian mixture model. For the pixels of the training images, we fit a Gaussian mixture model to the pixel brightness values, and we compute the likelihood of each pixel. Then, using the same fusion algorithm to go down to a pixel level segmentation from the results at 2×2 block scale, we obtain the final pixel level segmentation.

5.4. Implementation issues

Ideally, the inter-scale fusion algorithm should start at the root node of the labeling tree and descend to the final finest scale to combine all possible information. However, at very coarse scales, the likelihoods of dyadic squares do not contain significant information, because the squares are large and hence likely contain several differently textured regions. When combining multiscale classification results, we will therefore ignore the information at very coarse scales (above scale j_0, say). As a bonus, we will need HMT models only for scales $j \geq j_0$.

6. EXAMPLES

6.1. Synthetic example

In the first set of simulations, we segment a simple synthesized texture image consisting of a combination of "grass" and "wood" Brodatz[20] textures.[‖] To keep things simple, we considered only 64×64 images in the simulation. First we randomly selected five 64×64 blocks from the original grass and wood images to train respective HMT models. The training was performed with intra-scale tying[10] to avoid overfitting. The mosaic test image was created from randomly chosen 64×64 grass and wood images (see Figure 2(a)); the (i, j) elements with $|i - j| \leq 18$ correspond to wood texture with the remaining grass texture.

Figure 2(b) illustrates the raw classification results before inter-scale fusion processing. At coarse scales, the classification is reliable, but the details at the texture boundaries are poorly represented. The boundaries are better represented 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-density Gaussian mixture pixel brightness model.[15]

Figure 2(c) shows the segmentation results resulting from inter-scale fusion of the raw segmentation results using the context labeling tree. Comparing Figures 2(b) and 2(c), we observe an enormous increase in the segmentation accuracy.

6.2. Aerial photo segmentation

We next segment a real aerial photograph (also from the USC Image Database). The original image is a 1024×1024 image with "ground" and "sea" regions as shown in Figure 5(a). Figure 5(b) shows a 256×256 subblock of the full image that is to be segmented using HMTseg. First, 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 of depth 6 in scale (corresponding to dyadic squares of sizes 64×64 and smaller).

Figure 5(c) shows the raw classification results of the aerial photo. Again, the pixel-level raw segmentation was obtained using a 2-density Gaussian mixture model for pixel brightness of ground and sea textures. Figure 5(d) illustrates the segmentation results after utilizing the coarse-to-fine interscale fusion. Except for some segmentation errors in the upper middle part of the image (caused by the ground there having a texture more like sea), we observe excellent segmentation results at all scales.

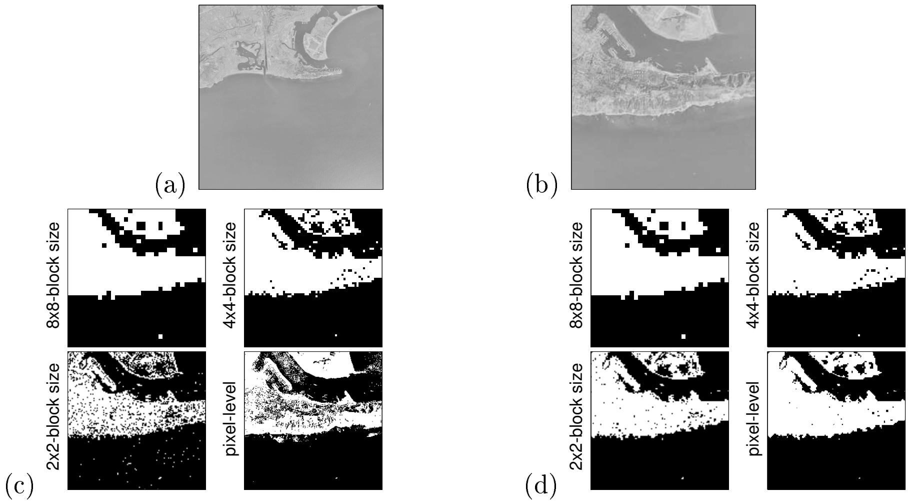

Figure 5. *(a) Original 1024×1024 aerial photo and (b) 256×256 test subimage. The homogeneous ground/sea regions outside the region (b) were used to train HMTs. (c) The raw classification is not accurate at fine scales. (d) The segmentation results using inter-scale fusion are much improved.*

‖Available thorough the USC Image Database at `sipi.usc.edu/services.html`.

6.3. Document segmentation

Finally, consider the problem of segmenting a scanned document into text, image, and background regions. Figure 6(b) shows the original 512 × 512 document image to be segmented by HMTseg. The wavelet-domain HMT models and pixel-level brightness models were trained based on the three different regions of the image shown in Figure 6(a).

Figure 6(c) shows the raw classification results. The final segmentation results after the Bayesian interscale fusion are shown in Figure 6(d). The labeled text, image, and background regions were displayed as black, gray, and white, respectively. All the text regions were segmented well, including the text surrounded by image regions. We observe at the bottom of the image that the titles consisting of large fonts were segmented as images. This is because the homogeneous texture inside each large letter has similar properties to an image. The background regions were correctly segmented, even though the brightness of the background varies in different areas and was corrupted by a noise-like feature caused by text on the reverse side of the page by the document scanner.

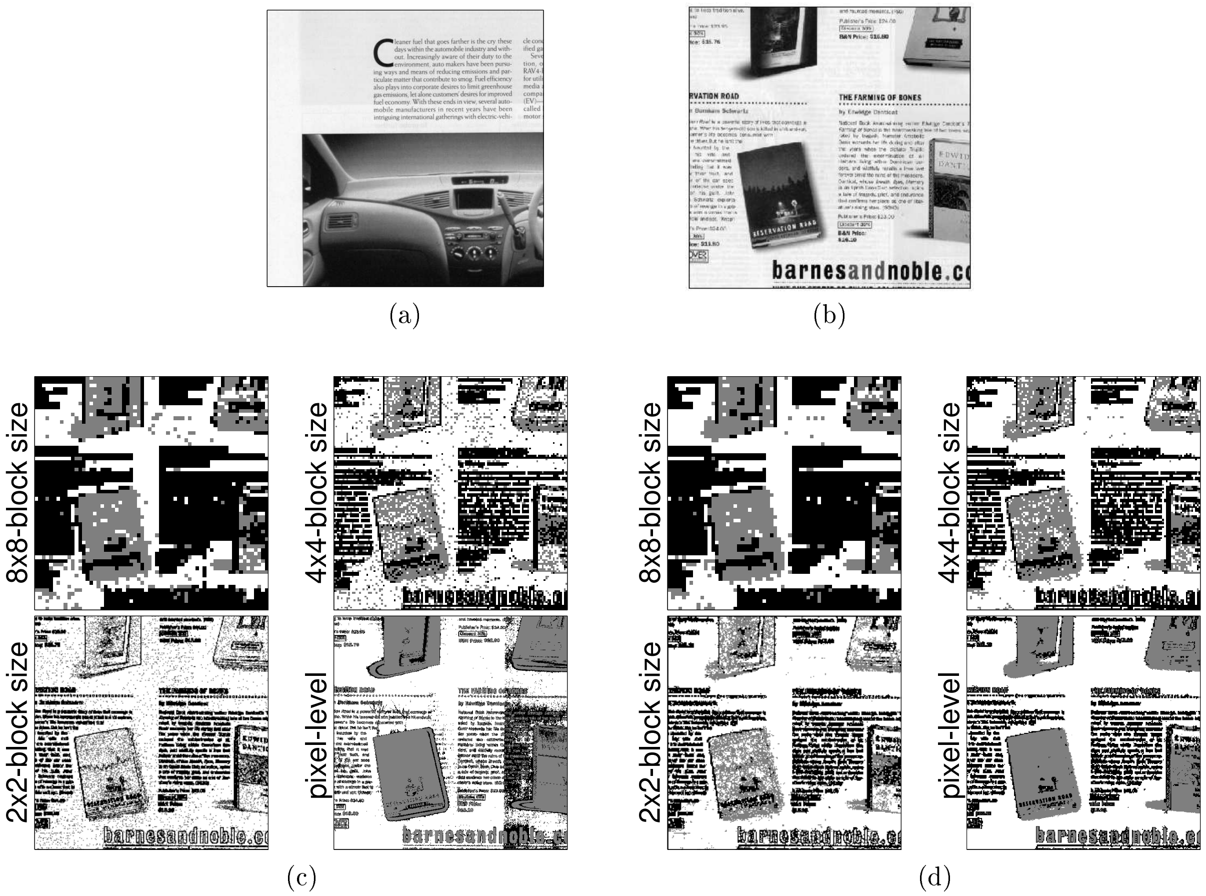

Figure 6. *(a) 512 × 512 document image used to train the HMT models for text, image, and background textures. The HMTs were trained on homogeneous regions obtained by manual segmentation. (b) Original 512 × 512 document image to be segmented. (c) Raw classification, with black, gray and white representing text, image and background, respectively. The classification accuracy decreases at fine scales. Pixel brightness is clearly not the proper feature for pixel-level classification; all texts was classified as image. (d) Segmentation resulting from inter-scale fusion; pixel-level segmentation is much improved.*

7. CONCLUSIONS

In this paper, we have developed a new framework for texture segmentation based on wavelet-domain HMT 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 texture images.

We believe the proposed segmentation algorithm can be applied to many different types of images, including radar/sonar images and medical images (CT/ultrasound). To incorporate the 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 HMT-based classifiers remains future research topic.

As an added bonus, the HMTseg algorithm has the potential to segment wavelet-compressed images directly in the compressed domain. Finally, while we concentrated on the image segmentation problem in this paper, HMTseg can be applied to 1-D and multi-D data as well.

REFERENCES

1. C. Therrien, "An estimation-theoretic approach to terrain image segmentation," *Comput. Vision Graphics Image Processing* **22**, pp. 313–326, 1983.
2. H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-9**, pp. 39–55, Jan. 1987.
3. S. Geman and D. Geman, "Stochastic relaxation, gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. on Pattern Anal. Machine Intell.* **6**, pp. 721–741, 1984.
4. R. Challappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acous. Speech. Signal Proc.* **33**, pp. 959–963, 1985.
5. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, 1998.
6. M. Vetterli and J. Kovačević, *Wavlets and Subband Coding*, Prentice Hall, Englewood Cliffs:NJ, 1995.
7. W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *J. Appl. Comp. Harm. Anal.* **3**(2), pp. 186–200, 1996.
8. S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, pp. 674–693, July 1989.
9. P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized-Gaussian priors," in *Proc. IEEE TFTS Symposium*, pp. 633–636, (Pittsburgh, PA), Oct. 6-9 1998.
10. M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Proc.* **46**, pp. 886–902, April 1998.
11. E. P. Simoncelli and E. H. Adelson, "Noise removal via Bayesian wavelet coring," in *IEEE Int. Conf. on Image Proc. — ICIP 1996*, (Switzerland), Sep. 1996.
12. S. Mallat and W. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. on Info. Theory* **38**(2), pp. 617–643, 1992.
13. C. Fosgate, H. Krim, W. Irving, W. Karl, and A. Willsky, "Multiscale segmentation and anomaly enhancement of SAR imagery," *IEEE Trans. on Image Proc.* **6**, pp. 7–20, Jan. 1997.
14. C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Proc.* **3**, pp. 162–177, March 1994.
15. H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden markov models," *Manuscript in Preparation - Available at www.dsp.rice.edu/publications/* , 1999.
16. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Proc.* **41**, pp. 3445–3462, Dec. 1993.
17. L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE* **77**, pp. 257–285, Feb. 1989.
18. B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*, MIT Press, Cambridge, MA, 1998.
19. M. S. Crouse and R. G. Baraniuk, "Simplified wavelet-domain hidden Markov models using contexts," in *Proc. 31st Asilomar Conf.*, (Pacific Grove, CA), Nov. 1997.
20. P. Brodatz, *Texture: A Photograph Album for Artists and Designers*, Dover, New York, 1956.