Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models

Hyeokho Choi, *Member, IEEE,* and Richard G. Baraniuk, *Senior Member, IEEE*

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

***Index Terms*—Hidden Markov tree, segmentation, texture modeling, wavelets.**

I. Introduction

A. Image Segmentation

AN IMAGE segmentation algorithm aims to assign a *class label* to each pixel of an image based on the properties of the pixel and its relationship with its neighbors. A "good" segmentation separates an image into simple regions with homogeneous properties, each with a different "texture" [1].

Recently, many authors have applied Bayesian statistical techniques to jointly estimate the region shapes and determine their classes [2]–[4]. Bayesian techniques regard a sampled image $\boldsymbol{x}$ as a realization of a random field $\boldsymbol{X}$ with distinct and consistent stochastic behavior in different regions.[1] In an image region $\boldsymbol{X}_r \subset \boldsymbol{X}$ of class c, the pixels are assumed distributed with joint probability density function (pdf) $f(\boldsymbol{x}_r|c)$. In these terms, the image segmentation problem can be rephrased as: given an image $\boldsymbol{x}$, estimate for each pixel a class label $c \in \{1, 2, \ldots, N_c\}$. The *labeling field* $\boldsymbol{C}$ records the class label of each pixel. Maximum likelihood (ML) segmentation partitions the image into regions $\boldsymbol{x}_r$ that maximize the likelihood $f(\boldsymbol{x}_r|c)$ over the regions. Maximum *a posteriori* (MAP) segmentation in addition weights the likelihoods by the prior probability of each c.

The two key ingredients to any segmentation scheme are 1) a description of the possible image regions $\boldsymbol{x}_r$ and 2) a set of joint pixel pdfs $\{f(\boldsymbol{x}_r|c) : c = 1, 2, \ldots, N_c\}$.

The primary difficulty in image segmentation arises because there are simply too many possible region shapes, and it is intractable to specify the joint pixel pdf for each possibility. Moreover, even if the joint density could be specified for each possible region shape, the cost of computing the optimal ML or MAP segmentation would be prohibitive. In practice, we must impose structures on both the possible image regions and on the pixel pdfs.

B. Multiscale Image Segmentation

Many segmentation algorithms employ a *classification window* of some size in the hope that all pixels in the window belong to the same class. A typical segmentation then consists of classifying 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, a large window produces accurate segmentations in large, homogeneous regions but poor segmentations along the boundaries between regions. A small window reduces the possibility of having multiple classes in the window but sacrifices classification reliability due to the paucity of statistical information. Thus, a small window is more appropriate near the boundaries between regions.

To capture the properties of each image region to be segmented, both the large and small scale behaviors should be utilized to properly segment both large, homogeneous regions and detailed boundary regions. In *multiscale segmentation* [5], [6] the results of many classification windows of different sizes are combined to obtain an accurate segmentation at fine scales.

In this paper, we will employ the *dyadic squares* (or blocks) to implement classification windows of different sizes. Given an initial $2^J \times 2^J$ square image $\boldsymbol{x}$ of $n := 2^{2J}$ pixels, the dyadic squares are obtained simply by recursively dividing the image into four square subimages of equal size [see Fig. 1(a)]. Since the four "child" squares nest inside their "parent" square at the next coarser scale, the dyadic squares have a convenient quad-tree structure; each node in the quad tree in Fig. 1(b) corresponds to a dyadic square. Denote a dyadic square at scale

Manuscript received October 26, 1999; revised May 31, 2001. This work was supported by the NSF under Grants MIP-9457438 and CCR-9973188, by DARPA/AFOSR Grant F49620-97-1-0513, the ONR under Grant N00014-99-1-0813, and the Texas Instruments Leadership University Program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pierre Moulin.

The authors are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005-1892 USA (e-mail: choi@ece.rice.edu; richb@rice.edu).

Publisher Item Identifier S 1057-7149(01)07466-8.

[1]We denote deterministic quantities using small letters, random variables using capital letters, and vectors using boldface letters.

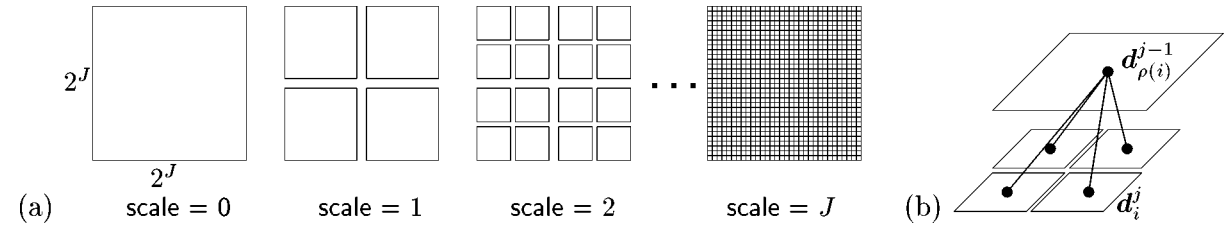

Fig. 1. (a) Image $\boldsymbol{x}$ divided into dyadic squares $\boldsymbol{d}_i^j$ at different scales. Each dyadic square can be associated with a subtree of Haar wavelet coefficients. (b) Quad-tree structure of dyadic squares. The dyadic square $\boldsymbol{d}_{\rho(i)}^{j-1}$ splits into four child squares at scale j.

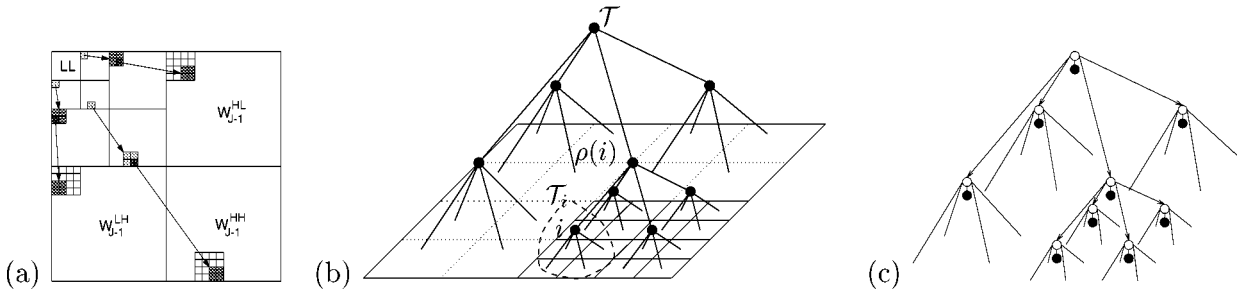

Fig. 2. (a) Parent–child dependencies of the three 2-D wavelet transform subbands: Each arrow points from a parent wavelet coefficient to its four children at the next finer scale. (b) More detailed view of the quad-tree structure for one subband. Each black node corresponds to a wavelet coefficient. The figure also illustrates our tree indexing notation: $\mathcal{T}_i$ is the subtree of coefficients rooted at node i, and $\rho(i)$ is the parent of node i. (c) A 2-D wavelet hidden Markov tree (HMT) model for one subband. We model each wavelet coefficient (black node) as a Gaussian mixture controlled by a hidden state variable (white node). To capture the persistence across scale property of wavelet transforms, we connect the states vertically across scale in Markov-1 chains.

j by $\boldsymbol{d}_i^j$ (with i an abstract index enumerating the squares at this scale). At the two extremes, $\boldsymbol{d}_0^0$ (root of the tree) is the entire image $\boldsymbol{x}$, and each $\boldsymbol{d}_i^J$ (leaf of the tree) is an individual pixel. Given a random field image $\boldsymbol{X}$, the dyadic squares are also random fields, denoted $\boldsymbol{D}_i^j$. In the sequel, when we speak of a generic square, we will often drop the j.

With this structure for representing regions, we will segment images by estimating the class label c for each dyadic square $\boldsymbol{d}_i$. This estimation requires a pixel pdf model for each class that is suited to the dyadic squares. Help is close at hand with the dyadic wavelet decomposition and wavelet-based statistical models.

C. Multiscale Statistical Models and Wavelets

Models of different image textures play a fundamental role in image classification and segmentation, since the complete joint pixel pdf is typically overly complicated or unavailable in practice. Transform-domain models are based on the idea that often a linear, invertible transform will "restructure" an image, leaving transform coefficients whose structure is simpler to model. Most real-world images, especially gray-scale texture images, are well characterized by their *singularity* (edge and ridge) structure. The wavelet transform provides a powerful transform domain for modeling singularity-rich images [7].

The wavelet transform can be interpreted as a multiscale edge detector that represents the singularity content of an image at multiple scales and three different orientations. Wavelets overlying a singularity yield large wavelet coefficients; wavelets overlying a smooth region yield small coefficients. Four wavelets at a given scale nest inside one at the next coarser scale, giving rise to a quad-tree structure of wavelet coefficients that mirrors that of the dyadic squares [see Fig. 2(a)]. In particular, with the *Haar wavelet transform*, each wavelet coefficient node in the wavelet quad-tree corresponds to a wavelet supported exactly on the corresponding dyadic image square.

In combination, the multiscale singularity detection property and tree structure imply that image singularities manifest themselves as cascades of large wavelet coefficients through scale along the branches of the quad-tree [7]. Conversely, smooth regions lead to cascades of small coefficients.

This multiscale singularity characterization makes the wavelet domain natural for modeling texture images. A number of statistical models have been developed for modeling textures [8]–[11]; here we concentrate on the *hidden Markov tree* (HMT) model of Crouse *et al.* [12]. The HMT approximates both the marginal and joint wavelet coefficient statistics. The HMT associates with each wavelet coefficient a (hidden) state variable that controls whether it is "large" or "small." The marginal density of each coefficient is then modeled as a two-density Gaussian mixture, using a large variance Gaussian for the large state and a small variance Gaussian for the small state. This Gaussian mixture closely matches the nonGaussian wavelet coefficient marginal statistics observed in natural images [9], [10], [13]. The HMT captures the persistence across scale of large/small coefficients using Markov-1 dependencies between the hidden states that chain across scale in a tree structure that parallels that of the wavelet coefficients and dyadic squares [see Fig. 2(c)]. Grouping the model parameters (Gaussian mixture variances and Markov state transition probabilities) into a vector $\mathcal{M}$, the HMT can be viewed as a high-dimensional yet highly structured Gaussian mixture model $f(\boldsymbol{w}|\mathcal{M})$ that approximates the overall joint pdf of the wavelet coefficients $\boldsymbol{W}$.

The computational efficiency of the wavelet transform carries over to HMT-based processing. The HMT model parameters can be estimated using the iterative expectation-maximization

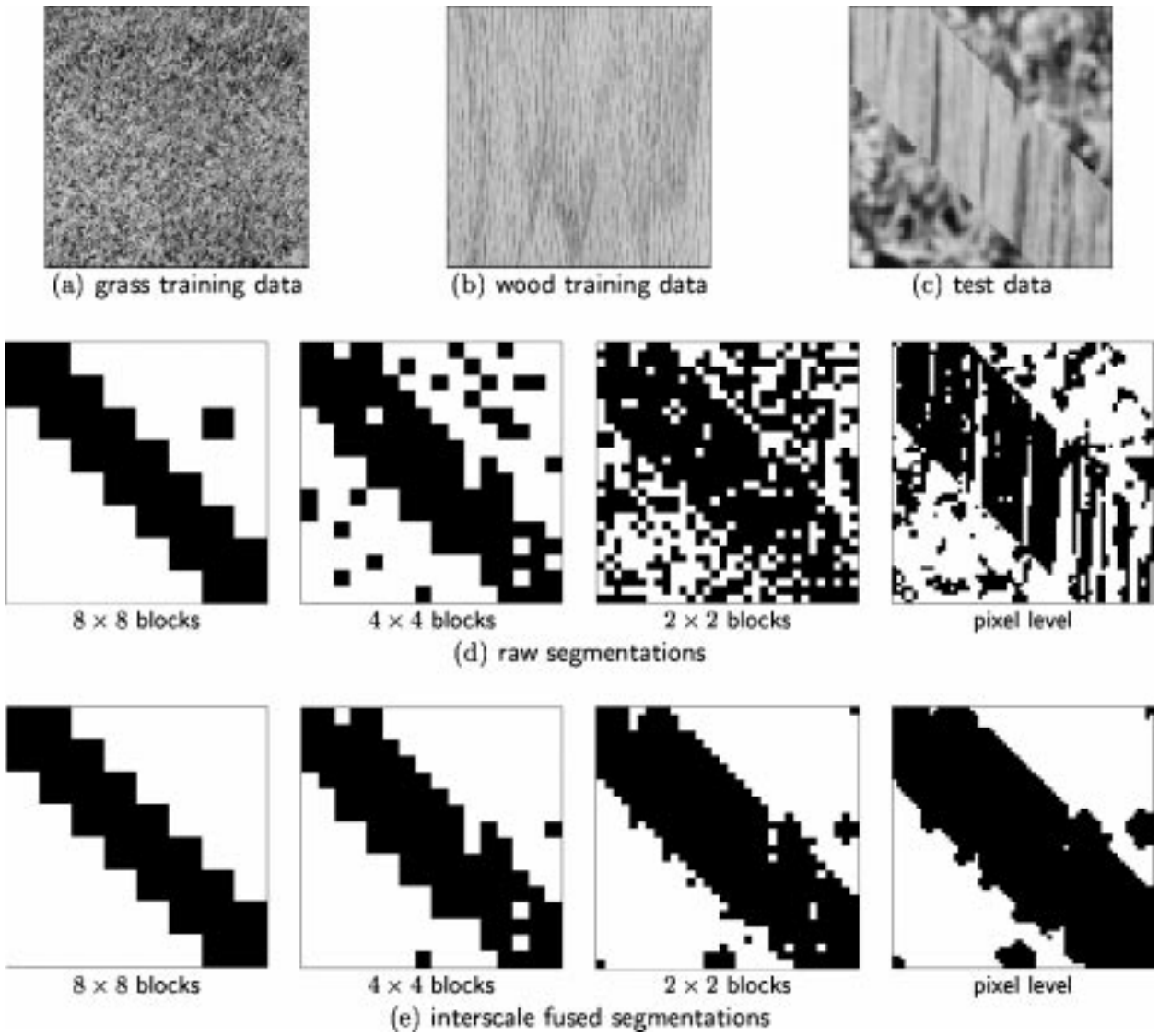

Fig. 3. HMTseg applied to a synthetic test image. (a) A 512×512 grass texture image [14]. (b) A 512×512 wood texture image [14]. (c) A 64×64 grass/wood mosaic test image $\tilde{\boldsymbol{x}}$ to be segmented. (d) Raw HMT-based multiscale classifications $\hat{\boldsymbol{c}}^j_{\mathrm{ML}}$ of $\boldsymbol{x}$ for $8 \times 8, 4 \times 4, 2 \times 2$, and pixel-sized dyadic squares. Classification accuracy increases with block size (toward coarser scales), because more statistical information is available for the class label decision. However, this comes at a cost of reduced boundary resolution. (e) Final segmentations $\hat{\boldsymbol{c}}^j_{\mathrm{MAP}}$ using Bayesian context-based interscale fusion.

(EM) algorithm at a cost of $O(n)$ computations per iteration [12]. More importantly, given the wavelet transform $\tilde{\boldsymbol{w}}$ of a test image $\tilde{\boldsymbol{x}}$ and a set of HMT parameters $\mathcal{M}$, computation of the likelihood $f(\tilde{\boldsymbol{w}}|\mathcal{M})$ that $\tilde{\boldsymbol{w}}$ is a realization of the HMT model requires only a simple $O(n)$ upsweep through the HMT tree from leaves to root [12].

The HMT has a convenient nesting structure that matches that of the dyadic squares. Each subtree of the HMT is itself an HMT, with the HMT subtree rooted at node i modeling the statistical behavior of the wavelet coefficients corresponding to the dyadic square $\boldsymbol{d}_i$. Serendipitously, the partial likelihood calculations obtained at intermediate scales of the HMT tree as part of the leaves-to-root upsweep give the likelihoods $f(\boldsymbol{d}_i|\mathcal{M})$ of *each dyadic subsquare of the image* under the HMT model (more details in Section IV-B).

These tools enable a simple multiscale image classification algorithm. Suppose that for each texture class $c \in \{1, 2, \ldots, N_c\}$ we have specified or trained an HMT with parameters $\mathcal{M}_c$. Now, given the wavelet transform $\tilde{\boldsymbol{w}}$ of an image $\tilde{\boldsymbol{x}}$ consisting of a montage of these textures, applying the above multiscale likelihood calculation to each HMT yields the likelihoods $f(\tilde{\boldsymbol{d}}_i|\mathcal{M}_c)$, $c \in \{1, 2, \ldots, N_c\}$ for each dyadic subimage $\tilde{\boldsymbol{d}}_i$. With the multiscale likelihoods at hand, the simplest ML classification

$$\hat{c}_i^{\mathrm{ML}} := \arg \max_{c \in \{1, 2, \ldots, N_c\}} f(\tilde{\boldsymbol{d}}_i \Big| \mathcal{M}_c) \tag{1}$$

then informs us of the most likely label $\hat{c}_i^{\mathrm{ML}}$ for each dyadic subimage $\tilde{\boldsymbol{d}}_i$. This classification process, which we call the *raw ML segmentation*, can be completed in just $O(n)$ computations for an n-pixel image. It yields a set of J different segmentations $\boldsymbol{c}^j_{\mathrm{ML}}$, $j = 0, 1, \ldots, J-1$, one for each different scale j of dyadic square.

Fig. 3 illustrates the process. After training HMT models on the grass and wood textures from Fig. 3(a) and (b), we per-

formed the multiscale classification (1) on the test image (c) to obtain the raw segmentations (d) at various scales.

D. Interscale Decision Fusion

While quick and easy, as Fig. 3(d) attests, the raw ML segmentations suffer from the classical "blockiness versus robustness" tradeoff that leaves no single $\hat{\boldsymbol{c}}^j_{\mathrm{ML}}$ desirable. To obtain a high-quality segmentation, clearly we should combine the multiscale results to benefit from both the robustness of large block sizes and the resolution of small block sizes.

Since finer scale dyadic squares nest inside coarser scale squares, the dyadic squares will be statistically dependent across scale for images consisting of fairly large, homogeneous regions. Hence, (reliable) coarse-scale information should be able to help guide (less reliable) finer-scale decisions.

If the dyadic square $\boldsymbol{d}_i^{j-1}$ was classified as class c, then it is quite likely that its four children squares at scale j belong to the same class, especially when j is large (at fine scales). Hence, we will guide the classification decisions for the child squares based on the decision made for their parent square. This will tend to make the class labels of the four children the same unless their likelihood values strongly indicate otherwise, thus reducing the number of misclassifications due to slight perturbations in child likelihood values. In addition to the parent square, we can also use the neighbors of the parent to guide the decision process. Similar multiscale decision ideas have been successfully applied to document segmentation in [6].

To exploit the parent–child dependencies between the dyadic squares, we will build yet another tree-structured probability model, the *labeling tree* (more details in Section IV-C). Akin to the HMT, the labeling tree models the dependencies between dyadic squares across scale in a Markov-1 fashion, where the dyadic squares at scale j are assumed to depend only on the squares at scale $j-1$. (Dependencies between squares within the same scale are captured through the squares' common ancestors.) Using tree-based modeling, we gain tremendous "economies of scale" [4], [12], [15]–[17].

Markov modeling leads us to a simple scale-recursive classification of the dyadic squares, where we classify $\boldsymbol{d}_i^j$ based on its likelihood and guidance from the previous scale $j-1$. This Bayesian *interscale decision fusion* computes a MAP estimate of the class label $\hat{c}_i^{\mathrm{MAP}}$ of each dyadic square $\boldsymbol{d}_i$. Stopping the fusion at scale j, we obtain the MAP segmentation $\hat{\boldsymbol{c}}^j_{\mathrm{MAP}}$. As we see from Fig. 3(e), multiscale decision fusion greatly improves the robustness and accuracy of the segmentation.

Combining the above tools results in a robust and accurate yet simple and efficient segmentation algorithm that we call *HMTseg* [18]. It relies on three separate tree structures: the wavelet transform quad-tree, the HMT, and the labeling tree.

E. Related Work

We group related work in texture classification and segmentation into four broad classes.

Markov Random Fields: Markov random fields (MRFs) [19]–[21] have been extensively applied to model the pixel pdf $f(\boldsymbol{x})$. However, while they enable spatially local processing, they capture only local interactions and thus have only a limited ability to describe large scale behavior. MRF's can be improved by incorporating more neighboring pixels, but this rapidly increases their complexity. More recently, there have been attempts to approximate MRFs using tree structured models [15], [22].

Scaling Coefficient Models: Multiscale autoregressive models approximate the multiscale statistics of the *scaling* coefficients rather than wavelet coefficients [5]. While the main advantage of multiscale image segmentation is to avoid the *ad hoc* choice of the classification window size, the divide-and-conquer segmentation algorithm of [5] (and a similar one in [23]) still requires a proper choice.

Wavelet-Domain Features: Wavelet-domain features are not new to texture classification and segmentation. In [24], Unser extracts parameters at different wavelet scales to facilitate texture classification. Li *et al.* [25] employ wavelet coefficient statistics to classify different textures in document images. Gross *et al.* [26] use a neural network to model the textural features of wavelet coefficients for classification purposes.

Multiscale Decision Algorithms: The multiscale labeling model of Bouman *et al.* [4], [16], [17] does not use an explicit model for the image pixels. Rather it indirectly models the pixel pdf using a multiscale model of the class labels only. The technique in [4] is a general systematic method for combining multiscale information. However, because it considers only the behavior of the class labels across scale without actually considering the joint statistics of the image pixels (it assumes that the pixels are independent given the class label), the algorithm is useful only for certain types of images (such as the SAR images considered in [4]) in which pixel values can be considered independent. The algorithms recently proposed in [16], [17] generalize [4] further. However, because these algorithms still do not perform direct modeling and decision of class labels at multiple scales, they require complicated statistical learning methods based on manually prepared training data. Furthermore, they still model the wavelet coefficients as independent, which is not accurate for singularity-rich data such as textures, because of the strong residual correlations between wavelet coefficients. In followup work to [25], Li *et al.* [6] propose a divide-and-conquer multiscale decision algorithm that incorporates deterministic context information. While intuitively appealing, their proposed decision rules are deterministic and presented with little justification.

As far as we know, HMTseg is the first attempt to combine parametric wavelet-domain statistical modeling, direct likelihood calculation, and multiscale Bayesian decision fusion (via the labeling tree, which is inspired by [4], [16], and [17]). We believe that these features give HMTseg several distinct advantages over existing segmentation techniques. In particular, since we obtain the multiscale likelihoods and classifications directly through the HMTs, the multiscale information fusion simplifies considerably. As a result, unlike the algorithms in [16], [17], we are able to extract the labeling tree parameters from the given image to be segmented, without additional training data.

F. Paper Organization

In Sections II and III, we study two of the basic ingredients of HMTseg: the wavelet transform and the wavelet HMT model.

We describe the third basic element, the labeling tree, and construct the algorithm in Section IV. Section V demonstrates the performance of HMTseg through a number of examples. We conclude in Section VI by pointing to some remaining issues and suggesting directions for further research.

II. Wavelet Transform

A. Wavelet Transform and Dyadic Squares

The wavelet transform represents the singularity content of an image at multiple scales. There are several different interpretations; we will find the pyramidal multiscale construction for discrete images cleanest for our purposes [27].

We will focus on the simplest wavelet transform, that of Haar. The construction of Haar wavelet coefficients of an image can be explained using four 2-D wavelet filters: the local smoother $h_{\mathrm{LL}} = 1/2\left(\begin{smallmatrix}1 & 1\\ 1 & 1\end{smallmatrix}\right)$, horizontal edge detector $g_{\mathrm{LH}} = 1/2\left(\begin{smallmatrix}1 & 1\\ -1 & -1\end{smallmatrix}\right)$, vertical edge detector $g_{\mathrm{HL}} = 1/2\left(\begin{smallmatrix}1 & -1\\ 1 & -1\end{smallmatrix}\right)$, and diagonal edge detector $g_{\mathrm{HH}} = 1/2\left(\begin{smallmatrix}1 & -1\\ -1 & 1\end{smallmatrix}\right)$.

To compute the wavelet transform of a $2^J \times 2^J$ discrete image $\boldsymbol{x}$, first set $\boldsymbol{u}_J[k, l] := \boldsymbol{x}[k, l]$, $0 \le k, l \le 2^J - 1$. Next, convolve $\boldsymbol{u}_J$ with the filters h_{LL}, g_{LH}, g_{HL}, and g_{HH} and discard every other sample in both the k and l directions. The resulting *subband* images—$\boldsymbol{u}_{J-1}$, $\boldsymbol{w}_{J-1}^{\mathrm{LH}}$, $\boldsymbol{w}_{J-1}^{\mathrm{HL}}$, and $\boldsymbol{w}_{J-1}^{\mathrm{HH}}$, respectively—are each of size $2^{J-1} \times 2^{J-1}$. The 4-pack can be compactly stacked back into a $2^J \times 2^J$ matrix

$$\begin{bmatrix} \boldsymbol{u}_{J-1} & \boldsymbol{w}_{J-1}^{\mathrm{HL}} \\ \boldsymbol{w}_{J-1}^{\mathrm{LH}} & \boldsymbol{w}_{J-1}^{\mathrm{HH}} \end{bmatrix}.$$

The filtering and downsampling process can now be continued on the $\boldsymbol{u}_{J-1}$ image and the procedure iterated up to J times [see Fig. 2(a)].

The *scaling coefficient* matrices $\boldsymbol{u}_j$, $0 \le j \le J-1$ are progressively smoothed versions of the original image $\boldsymbol{u}_J$. The *wavelet coefficient* matrices $\boldsymbol{w}_j^{\mathrm{LH}}$, $\boldsymbol{w}_j^{\mathrm{HL}}$, and $\boldsymbol{w}_j^{\mathrm{HH}}$ are high- and band-pass filtered, *edge-detected*, versions of the image that respond strongly to edges in the horizontal, vertical, and diagonal orientations, respectively. For example, the wavelet coefficient $\boldsymbol{w}_{J-1}^{\mathrm{LH}}[k, l]$, $0 \le k, l \le 2^{J-1} - 1$, is large if the 2×2 image block

$$\begin{bmatrix} \boldsymbol{x}[2k, 2l] & \boldsymbol{x}[2k, 2l+1] \\ \boldsymbol{x}[2k+1, 2l] & \boldsymbol{x}[2k+1, 2l+1] \end{bmatrix}$$

contains a horizontal edge and small otherwise.

The iterative computation of each Haar wavelet coefficient from a 2×2 block in a finer-scale image leads naturally to a quad-tree structure on the wavelet coefficients in each subband, as illustrated in Fig. 2(a) and (b) [28]. First assume that we carry out the iterated filtering to scale $j = 0$, and consider only the LH subband. Then the root of the tree lies at $\boldsymbol{w}_0^{\mathrm{LH}}[0, 0]$ and the leaves at $\boldsymbol{w}_{J-1}^{\mathrm{LH}}[k, l]$, $0 \le k, l \le 2^J - 1$. As we move down the tree, we move from coarse to fine scale, adding details as we go. More specifically, each *parent* wavelet coefficient $\boldsymbol{w}_j^{\mathrm{LH}}[k, l]$ analyzes the same region in the original image as its four *children* $\boldsymbol{w}_{j+1}^{\mathrm{LH}}[2k, 2l]$, $\boldsymbol{w}_{j+1}^{\mathrm{LH}}[2k, 2l+1]$, $\boldsymbol{w}_{j+1}^{\mathrm{LH}}[2k+1, 2l]$, and $\boldsymbol{w}_{j+1}^{\mathrm{LH}}[2k+1, 2l+1]$. Coefficients on the path to the root are *ancestors*; coefficients on the paths to the leaves are *descendents*. If we terminate the iterated filtering at a scale $j > 0$, then there will be more than one coarsest scale wavelet coefficient in each subband, leading to a forest of quad-trees in each subband [12].

To keep the notation manageable in the sequel, let $\boldsymbol{w}$ denote the collection of all wavelet coefficients, and let $\boldsymbol{w}^{\mathrm{LH}}$, $\boldsymbol{w}^{\mathrm{HL}}$, $\boldsymbol{w}^{\mathrm{HH}}$ denote the collections of all coefficients in the respective subbands. Let w_i denote a generic wavelet coefficient, with the subband under consideration determined by context. In our statistical modeling framework, we will regard w_i as a realization of the random variable W_i and $\boldsymbol{w}$ as a realization of the wavelet random field $\boldsymbol{W}$. Define by $J(i)$ the scale of coefficient i in the subband quad-tree. Define $\rho(i)$ as the parent of tree node i. In a given subband, define $\mathcal{T}_i$ as the subtree of wavelet coefficients with root node i; that is, $\mathcal{T}_i$ contains coefficient w_i and all of its descendents [see Fig. 2(b)].

With the 2-D Haar wavelet transform, there is an obvious correspondence between the wavelet coefficients and the dyadic squares [recall Fig. 1(a)], which are obtained by iteratively dividing the image into equal-size quadrants. Recall that $\boldsymbol{d}_i^j$ denotes a dyadic square at scale j, with i an abstract index for the square. (In the sequel, superscripts will always denote scale and subscripts will always denote position within a scale.) Each $\boldsymbol{d}_i^j$ is obtained by dividing a square at scale $j-1$ (the "parent" square, $\boldsymbol{d}_{\rho(i)}^{j-1}$) into four quadrants (the "child" squares). To each dyadic square of pixels $\boldsymbol{d}_i$ there corresponds a unique set of wavelet coefficients with a special property: all wavelet coefficients in the subtree $\mathcal{T}_i$ rooted at w_i depend exclusively on the pixel values in $\boldsymbol{d}_i$.

The same procedure of wavelet transform construction can be applied to other wavelet systems besides the Haar. While larger wavelet filters are more appropriate for representing smooth images, the Haar system is more appropriate for our purpose of classifying dyadic squares, due to its direct connection with the dyadic squares. We will see that the Haar system is more than adequate for the HMTseg algorithm.

Because each wavelet coefficient is computed locally, the wavelet transform efficiently represents with large coefficients only the edges of images, resulting in a very sparse and compact representation of singularity-rich images [7], [12], [28]. Furthermore, the approximate decorrelation of wavelet coefficients [7], [12] enables us to use very simple local modeling of joint statistics between wavelet coefficients.

III. Two-Dimensional Hidden Markov Tree Model

The wavelet hidden Markov tree (HMT) [12], [29] models both the nonGaussian marginal pdfs of the wavelet coefficients and the persistence of large/small coefficients across scale. As indicated in [12], HMT models can be developed for wavelet transforms of any dimensionality; here we focus on 2-D images and thus quad trees.

A. Modeling the Marginal Distribution

The energy compaction property [7], [12], [27] of the wavelet transform implies that the transform of most real-world images consists of a small number of large coefficients and a large

number of small coefficients. We can consider the population of large coefficients as outcomes of a pdf with a large variance. Similarly, we can consider the collection of small coefficients as outcomes of a pdf with a small variance. Hence, the pdf $f(w_i)$ of each wavelet coefficient is well approximated by a two-density *Gaussian mixture model* [9], [10], [12], [13].[2]

To each wavelet coefficient $\boldsymbol{W}_i$, we associate a discrete hidden state S_i that takes on the values $m = \mathtt{S}, \mathtt{L}$, signifying the small and large variance, with probability mass function (pmf) $p_{S_i}(m)$. Conditioned on $S_i = m$, $\boldsymbol{W}_i$ is Gaussian with mean $\mu_{i,m}$ and variance $\sigma^2_{i,m}$. Thus, the overall pdf of $\boldsymbol{W}_i$ is given by

$$f(w_i) = \sum_{m=\mathtt{S},\,\mathtt{L}} p_{S_i}(m) f(w_i|S_i = m) \tag{2}$$

where $f(w_i|S_i = m) \sim N(\mu_{i,m}, \sigma^2_{i,m})$ and $p_{S_i}(\mathtt{S}) + p_{S_i}(\mathtt{L}) = 1$. In Fig. 2(c) we depict the wavelet coefficients using black nodes and their associated hidden states using white nodes.

B. Modeling Across-Scale Dependencies

Working in the Gaussian mixture marginal distribution framework, we can characterize the dependencies between the wavelet coefficients by specifying the joint pmf of the hidden states. Thanks to the approximate decorrelating power of the wavelet transform [7], the most important correlations are the parent–child interactions due to magnitude persistence across scale [12].[3]

To capture the scale persistence of coefficient magnitudes, we connect the hidden states in a directed Markov-1 probabilistic graph [12]. For each parent–child pair of hidden states $\{S_{\rho(i)}, S_i\}$, the state transition probabilities $\epsilon^{\rho(i),m}_{i,m'}$ for $m, m' = \mathtt{S}, \mathtt{L}$ represent the probability for W_i to be small/large when its parent $W_{\rho(i)}$ is small/large. For each i, we thus have the state transition probability matrix

$$\begin{bmatrix} \epsilon^{\rho(i),\mathtt{S}}_{i,\mathtt{S}} & \epsilon^{\rho(i),\mathtt{S}}_{i,\mathtt{L}} \\ \epsilon^{\rho(i),\mathtt{L}}_{i,\mathtt{S}} & \epsilon^{\rho(i),\mathtt{L}}_{i,\mathtt{L}} \end{bmatrix} = \begin{bmatrix} \epsilon^{\rho(i),\mathtt{S}}_{i,\mathtt{S}} & 1-\epsilon^{\rho(i),\mathtt{S}}_{i,\mathtt{S}} \\ 1-\epsilon^{\rho(i),\mathtt{L}}_{i,\mathtt{L}} & \epsilon^{\rho(i),\mathtt{L}}_{i,\mathtt{L}} \end{bmatrix}.$$

For typical gray-scale images, we expect $\epsilon^{\rho(i),\mathtt{S}}_{i,\mathtt{S}}$ and $\epsilon^{\rho(i),\mathtt{L}}_{i,\mathtt{L}}$ to be large due to the persistence property.

C. The 2-D HMT Model

A complete 2-D image wavelet transform comprises three subbands with three parallel quad-tree structures [as in Fig. 2(b)]. In particular, node i in the LH, HL, and HH quad trees corresponds to the same dyadic square $\boldsymbol{d}_i$ in the image. While the three subbands are clearly dependent on each other, for tractability reasons, we assume the following.

Subband Independence Assumption: The three subbands of the 2-D wavelet transform are statistically independent.

The complete 2-D wavelet HMT model $\mathcal{M}$ consists of three HMT's [one for each wavelet subband as shown in Fig. 2(c)]. Denoting the parameter vectors for the three subband HMTs as $\boldsymbol{\Theta}^{\mathrm{LH}}$, $\boldsymbol{\Theta}^{\mathrm{HL}}$ and $\boldsymbol{\Theta}^{\mathrm{HH}}$, respectively, we have $\mathcal{M} := \{\boldsymbol{\Theta}^{\mathrm{LH}}, \boldsymbol{\Theta}^{\mathrm{HL}}, \boldsymbol{\Theta}^{\mathrm{HH}}\}$. The HMT is thus a parametric model (multidimensional Gaussian mixture) for the joint pdf of the wavelet coefficients. Under the subband independence assumption, we can write

$$f(\boldsymbol{w}|\mathcal{M}) := f(\boldsymbol{w}^{\mathrm{LH}}|\boldsymbol{\Theta}^{\mathrm{LH}}) f(\boldsymbol{w}^{\mathrm{HL}}|\boldsymbol{\Theta}^{\mathrm{HL}}) f(\boldsymbol{w}^{\mathrm{HH}}|\boldsymbol{\Theta}^{\mathrm{HH}}). \tag{3}$$

In addition to modeling the wavelet coefficients, we can separately model the scaling coefficients—using a mixture density, for example [12]. However, for HMTseg, we will intentionally ignore the scaling coefficients. Since the scaling coefficients equal local pixel averages, we thus build into our statistical modeling independence to the local brightness level. This is a desirable feature for many segmentation applications, because even in regions of homogeneous statistical properties, the local brightness often varies in different parts of the region.

As it stands, the HMT has a large number of parameters (approximately $4n$ for an n-pixel image). This can make model training difficult when only a small number of training images are available. Fortunately, wavelet coefficients at the same scale tend to exhibit similar statistical characteristics [12], [30]; thus we can use just one set of parameters for each scale. This nodal *tying* reduces the number of parameters considerably (to approximately $4J$ for a J-scale transform), avoiding the risk of overfitting the model [12], [31].

IV. Multiscale Segmentation Using HMT Models

We now describe the HMTseg algorithm's HMT training, fast HMT-based likelihood calculation, and multiscale Bayesian decision fusion.

A. Training the HMT Models

Before we begin, we must acquire training data representative of each texture and train an HMT model for each. We typically obtain training images either by picking homogeneous regions from the given image or from completely different images having homogeneous regions representative of the candidate textures. For each class $c \in \{1, \ldots, N_c\}$, we train a 2-D wavelet HMT model $\mathcal{M}_c$ using the iterative expectation-maximization (EM) algorithm [12], [31], [32].

The EM algorithm finds the locally optimal (in the ML sense) set of model parameters $\mathcal{M}$ for a given set of training data. In each iteration, the E step defines a likelihood surface based on the current parameters. The M step then updates the parameters to maximize the likelihood that the training data came from the model [31], [32]. The EM algorithm derived for 1-D HMT models in [12] applies with little modification in 2-D if we interpret the parent–child relations between nodes appropriately for quad-trees.[4] In the HMT, each EM iteration consists of an up/down sweep through the tree [$O(n)$ cost for n wavelet coefficients].

[2]We can use more than two mixture densities to provide a fit to the actual $f(w_i)$ with any desired fidelity. In practice, however, we have seen no real performance benefit to using more than two.

[3]While the HMT focuses on across-scale dependencies, it does not ignore within-scale dependencies. Correlations between coefficients at the same scale are modeled through their mutual ancestors in the HMT quad tree.

[4]The only necessary change is in [12, p. 900, Eq. (22)], where the product now covers four child coefficients on the quad tree rather than two children on the binary tree. For more information on probabilistic graphs and training algorithms, see [32].

To maximize the spatial shift invariance of HMTseg, we can either use intra-scale tying (using the same mixture variances and state transition matrices for all wavelet coefficients at the same scale) as proposed in [12] or prepare a training set containing different shifts of the training textures. In our experiments, we have found only minimal performance reduction (but significant parameter reduction) using intra-scale tying. Furthermore, in all of our experiments, reliable HMT training required fewer than ten training images.

Given m training images of n pixels each, the total computational cost per EM iteration is $O(mn)$. While the total training cost can be enormous for large m and n, we have several avenues for reducing the amount of computation. First, intra-scale tying significantly reduces both the number of parameters to train and the number of required training images m. Second, we note that coarse-scale wavelet coefficients correspond to large dyadic squares that likely contain a number of different textures. Since these squares supply little information for segmentation, we can clip the upper scales from the HMT, creating a forest of smaller HMTs. These smaller HMT's naturally share the same parameters and thus reduce the size n of our required training images (more on this in Section IV-E).

While EM training is notorious for its slow convergence, in our experiments (including the two examples in Section V) we reached convergence in fewer than twenty iterations using an intelligent parameter initialization.

B. Multiscale Likelihood Computation

Given a set of 2-D HMT model parameters $\mathcal{M}$ and the wavelet transform $\tilde{\boldsymbol{w}}$ of a test image, it is straightforward to compute the *likelihood* $f(\tilde{\boldsymbol{w}}|\mathcal{M})$ that the image was generated by the model [12]. Furthermore, thanks to the dyadic multiscale structure of the wavelet transform and the HMT, we can obtain the *likelihoods of all dyadic squares of the image* simultaneously in a single upward sweep through the tree [a fast $O(n)$ algorithm].

Consider first the likelihood calculation for a subtree $\mathcal{T}_i$ of wavelet coefficients rooted at w_i in one of the subbands [12]. Suppose this subband has HMT parameters $\boldsymbol{\Theta}$. Given the conditional likelihoods $\beta_i(m) := f(\mathcal{T}_i|S_i = m, \boldsymbol{\Theta})$ obtained by sweeping up the quad-tree from the leaves to node i (see [12]), the likelihood of the coefficients in $\mathcal{T}_i$ can be computed as

$$f(\mathcal{T}_i|\boldsymbol{\Theta}) = \sum_{m=\mathrm{S},\mathrm{L}} \beta_i(m) p(S_i = m|\boldsymbol{\Theta}) \tag{4}$$

with $p(S_i = m|\boldsymbol{\Theta})$ state probabilities obtained directly from $\boldsymbol{\Theta}$ (or computed during training).

Now recall the connection with the dyadic squares. It is easy to see that the wavelet coefficients of the square $\boldsymbol{d}_i$ consist of the triple $\{\mathcal{T}_i^{\mathrm{LH}}, \mathcal{T}_i^{\mathrm{HL}}, \mathcal{T}_i^{\mathrm{HH}}\}$, each a subtree of one of the three wavelet subband quad-trees. Using three HMT upsweeps, we can easily compute the likelihood (4) for each of these subtrees. Then, using the subband independence assumption, we have

$$f(\boldsymbol{d}_i|\mathcal{M}) = f(\mathcal{T}_i^{\mathrm{LH}}|\boldsymbol{\Theta}^{\mathrm{LH}}) f(\mathcal{T}_i^{\mathrm{HL}}|\boldsymbol{\Theta}^{\mathrm{HL}}) f(\mathcal{T}_i^{\mathrm{HH}}|\boldsymbol{\Theta}^{\mathrm{HH}}). \tag{5}$$

Using (5), we can compute the likelihood under each texture model of each dyadic square down to 2×2 block scale.

Direct block-by-block comparison of the likelihoods (1) computed using (5) yields the ML raw segmentations at a range of scales [recall Fig. 3(d)]. (We discuss how to carry this down to pixel-sized blocks in Section IV-D.) We refer to this block-by-block classification as "raw," because we do not exploit any possible relationships between the classifications at different scales. We expect the raw decisions to be more reliable at coarser scales (where we have more image pixels per block) but more finely localized at finer scales (where the blocks are smaller). Clearly it is in our best interests to overcome this blockiness versus robustness tradeoff by folding the coarse-through-fine likelihoods into our final segmentation recipe.

C. Context-Based Interscale Fusion

We can improve the raw segmentation considerably by considering the dependencies between the class decisions at different scales. We will do this by modeling the multiscale dependencies between the dyadic blocks. Our approach is inspired by the Bayesian multiscale segmentation framework of Bouman *et al.* [4].

1) Bayesian Segmentation: In a Bayesian segmentation framework, we treat each class label c_i as a random variable C_i taking a value from $\{1, 2, \ldots, N_c\}$. Given the posterior distribution $p(c_i|\boldsymbol{x})$ of C_i given the image $\boldsymbol{X}$, the MAP classification of dyadic square $\boldsymbol{d}_i$ corresponds to the class label that maximizes the posterior distribution

$$\hat{c}_i^{\mathrm{MAP}} := \arg \max_{c_i \in \{1, 2, \ldots, N_c\}} p(c_i|\boldsymbol{x}). \tag{6}$$

By Bayes rule, the posterior is given by

$$p(c_i|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|c_i) p(c_i)}{f(\boldsymbol{x})}. \tag{7}$$

Let $\boldsymbol{d}^j := \{\boldsymbol{d}_i^j\}$ denote the collection of all dyadic squares at scale j; note that each $\boldsymbol{d}^j$ contains complete information on the image $\boldsymbol{x}$. A posterior equivalent to (7) is thus

$$p(c_i^j|\boldsymbol{d}^j) = \frac{f(\boldsymbol{d}^j|c_i^j) p(c_i^j)}{f(\boldsymbol{d}^j)}. \tag{8}$$

Since computation and maximization of (8) is intractable in practice, we will perform a succession of manipulations and simplifications to arrive at a practical MAP classifier. Just as the HMT models the pdfs $f(\boldsymbol{w})$ and $f(\boldsymbol{x})$ by echoing the structure of the wavelet coefficient quad tree, we will construct a probabilistic tree to model the posterior (8) based on the dyadic square quad tree of Fig. 1(b). The resulting *labeling tree* model will capture the interscale dependencies between dyadic blocks and their class labels and enable a *multiscale Bayesian decision fusion*. There are many ways to capture these multiscale dependencies [4], [6], [16], [17]; here we outline one possible approach that balances accuracy with tractability.

2) Image Model with Hidden Class Labels: Rather than modeling the joint statistics of the dyadic squares $\boldsymbol{D}_i$ directly, we will model the statistics of the associated class labels C_i. We assume that class label C_i controls the textural properties of square $\boldsymbol{D}_i$; that is, each $\boldsymbol{D}_i$ is generated based on the distribution $f(\boldsymbol{d}_i|c_i)$ independently of all other c_k and $\boldsymbol{d}_k$, $k \neq i$. Let

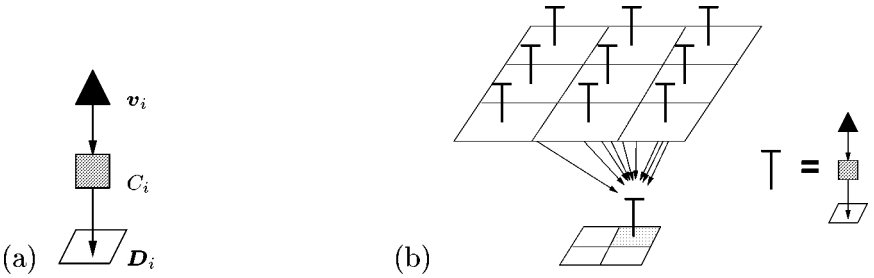

Fig. 4. (a) Context, class label, and dyadic square form a Markov-1 chain: $\boldsymbol{v}_i \rightarrow C_i \rightarrow \boldsymbol{D}_i$. (b) Context labeling tree. The context of the child square is determined by the decision results of its parent plus eight neighbors.

$\boldsymbol{C}^j := \{C_i^j\}$ denote the collection of all class labels at scale j. Then, given $\boldsymbol{C}^j = \boldsymbol{c}^j$, all $\boldsymbol{D}_i^j$ are independent

$$f(\boldsymbol{d}^j|\boldsymbol{c}^j) = \prod_i f(\boldsymbol{d}_i^j|c_i^j). \tag{9}$$

The class labels play a role analogous to the hidden states in the HMT (recall that given the values of the states, all wavelet coefficients are independent [12], [31], [32]).

Under the conditional independence of the $\boldsymbol{d}_i^j$s given $\boldsymbol{c}^j$, our MAP classification problem transforms to maximizing the posterior $f(c_i^j|\boldsymbol{d}^j)$ [recall (8)], the marginal of

$$p(\boldsymbol{c}^j|\boldsymbol{d}^j) = \frac{f(\boldsymbol{d}^j|\boldsymbol{c}^j)p(\boldsymbol{c}^j)}{f(\boldsymbol{d}^j)} = \frac{p(\boldsymbol{c}^j)}{f(\boldsymbol{d}^j)} \prod_i f(\boldsymbol{d}_i^j|c_i^j). \tag{10}$$

Here we have used (9). Unfortunately, marginalizing (10) to obtain (8)—integrating out all c_k^j, $k \neq i$—for the MAP decision statistic is difficult in general, unless the joint distribution $p(\boldsymbol{c}^j)$ has a special structure.

3) Multiscale Prior and Contexts: To simplify the determination and marginalization of the joint prior distribution $p(\boldsymbol{c}^j|\boldsymbol{d}^j)$ in (10), we assume that the joint distribution of the class labels C_i^j at scale j is completely determined by the C_i^{j-1} at the previous coarser scale. Combined with the assumption that $\boldsymbol{D}_i$ is conditionally independent of all C_k, $k \neq i$ given C_i, we have that C_i^{j-1}, C_i^j, and $\boldsymbol{D}_i$ form a Markov chain: $\{C_i^{j-1}\} \rightarrow C_i \rightarrow \boldsymbol{D}_i$. This heuristic models the interscale dependencies between the class labels that we motivated in Section I-D. Thus, given $\boldsymbol{C}^{j-1} = \boldsymbol{c}^{j-1}$, the C_i^js at scale j are independent, and we can write the *multiscale prior distribution*

$$p(\boldsymbol{c}^j|\boldsymbol{c}^{j-1}) = \prod_i p(c_i^j|\boldsymbol{c}^{j-1}). \tag{11}$$

Due to the high dimensionality of the conditioning vector $\boldsymbol{c}^{j-1}$, estimating the marginalized class prior distribution $p(c_i^j|\boldsymbol{c}^{j-1})$ still requires a prohibitive amount of training data. *Contexts* provide a useful further simplification of (11) [33]. To each dyadic square $\boldsymbol{D}_i^j$ with hidden class label C_i^j, we assign the (deterministic) context vector $\boldsymbol{v}_i^j$, which is formed from information about the $\boldsymbol{c}^{j-1}$.

The triple $\boldsymbol{v}_i \rightarrow C_i \rightarrow \boldsymbol{D}_i$ forms a Markov-1 chain [see Fig. 4(a)]. That is, $\boldsymbol{v}_i$ encodes sufficient information regarding $\boldsymbol{c}^{j-1}$ such that, given its value, we can treat C_i^j and $\boldsymbol{D}_i$ as independent of all other C_k and $\boldsymbol{D}_k$. If $\boldsymbol{v}_i$ is chosen as a discrete vector of small dimension, then it simplifies the modeling considerably. In the multiscale prior model (11), we let $\boldsymbol{v}_i^j$ be a function of the $\boldsymbol{c}^{j-1}$. Let $\boldsymbol{v}^j$ be the collection of all contexts at scale j.

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 model. Among many candidate contexts, we can determine the effective contexts based on known training data. In some sense, the decision-tree based algorithm in [16] is a general form of the context-based fusion algorithm applicable when sufficient training data is available for reliable estimation of the decision parameters.

Contexts allow us to write

$$p(\boldsymbol{c}^j|\boldsymbol{v}^j) = \prod_i p(c_i^j|\boldsymbol{v}_i^j). \tag{12}$$

Since $\boldsymbol{D}_i$ is independent of $\boldsymbol{v}_i$ given C_i (by the Markov-1 property), conditioning (10) on the contexts yields

$$\begin{aligned} p(\boldsymbol{c}^j|\boldsymbol{d}^j, \boldsymbol{v}^j) &= \frac{f(\boldsymbol{d}^j|\boldsymbol{c}^j)p(\boldsymbol{c}^j|\boldsymbol{v}^j)}{f(\boldsymbol{d}^j|\boldsymbol{v}^j)} \\ &= \frac{1}{f(\boldsymbol{d}^j|\boldsymbol{v}^j)} \prod_i \left[f(\boldsymbol{d}_i^j|c_i^j)p(c_i^j|\boldsymbol{v}_i^j) \right] \end{aligned} \tag{13}$$

and the marginalized, context-based posterior

$$f(c_i^j|\boldsymbol{d}_i^j, \boldsymbol{v}_i^j) \propto f(\boldsymbol{d}_i^j|c_i^j)p(c_i^j|\boldsymbol{v}_i^j). \tag{14}$$

This is a greatly simplified version of the MAP posterior (7) for use in the MAP equation (6). Here, the $f(\boldsymbol{d}_i|c_i)$ are the likelihoods of the dyadic square $\boldsymbol{d}_i$ given the different class values C_i, which are computed using an HMT likelihood upsweep on each texture model. The prior $p(c_i|\boldsymbol{v}_i)$ supplies information on the C_i^j provided by the C_k^{j-1}s through $\boldsymbol{v}_i$.

4) Context Labeling Tree: The interscale dependency modeling between the class labels (11) yields a tree of class labels, where the dependencies march down the tree in a Markov fashion. Compared with dependency modeling at each individual scale (with, say, a MRF), causal tree-based dependency modeling is both simple and effective.

While each context $\boldsymbol{v}_i^j$ is potentially a function of all C_k^{j-1} at scale $j-1$, here we will employ a simplified tree organization: each $\boldsymbol{v}_i^j$ at scale j will receive information from nine scale $j-1$ class labels, the parent label $C_{\rho(i)}^{j-1}$ plus the parent's eight nearest neighboring C_k^{j-1} [see Fig. 4(b)]. We term this context organization the *context labeling tree*. The limit of coarser scale information to just nine blocks is easily justified by noting that $\boldsymbol{v}_i^j$ will receive information from a region of pixels centered around and 36 times larger than its square $\boldsymbol{d}_i^j$.

While the context choice is very general and we may need a very complex context to accurately summarize the information conveyed by the $\boldsymbol{C}^{j-1}$, over-complicated contexts run the risk of *context dilution*, especially with insufficient training data [16], [17], [33].[5]

Inspired by the success of hybrid tree model in [4], we will employ a simple context structure in HMTseg. Each context vector $\boldsymbol{v}_i$ will contain two entries: the value of the class label $C_{\rho(i)}$ of the parent square (which will be a MAP estimate in practice) and the majority vote of the class labels of the parent plus its eight neighbors. Given N_c different textures, each context can take on N_c^2 different values. Let the number of different values $\boldsymbol{v}_i$ can take be N_v ($= N_c^2$ in the algorithm); thus, $\boldsymbol{v}_i \in \{\overline{\boldsymbol{v}}_1, \ldots, \overline{\boldsymbol{v}}_{N_v}\}$. While this simple context is *ad hoc*, it is more than sufficient for demonstrating the effectiveness of multiscale decision fusion. Furthermore, context training can be accomplished reliably based on the given image only, without requiring extra training data for estimating the context related probabilities.

Since the $p(c_i^j|\boldsymbol{v}_i)$s at scale j depend on the C_k^{j-1}s from scale $j-1$, we will evaluate and maximize (14) in a multiscale, coarse-to-fine manner to fuse the HMT likelihoods $f(\boldsymbol{d}_i|c_i)$ (precomputed as in Section IV-B) using the labeling tree prior $p(c_i^j|\boldsymbol{v}_i)$. Our fusion will pass the MAP decisions down through scale to aid the segmentation of fine scale dyadic squares. The result is simple, yet effective.

5) Interscale Fusion EM Algorithm: The fusion proceeds as follows. Start at a coarse enough scale $j-1$ such that the ML raw segmentations $\hat{\boldsymbol{c}}_{\mathrm{ML}}^{j-1}$ are statistically reliable. Use these and all coarser ML decisions as the MAP decisions $\hat{\boldsymbol{c}}_{\mathrm{MAP}}^{j-1}$. This is entirely reasonable; at coarse scales (large dyadic squares), the next coarser scale (very large dyadic squares) provides little prior information for segmentation.

Now move down to the next finer level j. Fix the context values $\boldsymbol{v}_i^j$ from the $\hat{\boldsymbol{c}}_{\mathrm{MAP}}^{j-1}$ at scale $j-1$ (from the parent label and its eight nearest neighbors). We are given the likelihood $f(\boldsymbol{d}_i^j|c_i^j)$ in (14) from the HMT likelihood computation step. Hence, after computing $p(c_i^j|\boldsymbol{v}_i^j)$, we can choose the label for $\boldsymbol{c}_{\mathrm{MAP}}^j$ that maximizes the product (14).

To compute $p(c_i|\boldsymbol{v}_i)$, we use an ML estimate averaged over the collection of *all* dyadic squares $\boldsymbol{d}_k$ at scale j. Since this collection is precisely the image $\boldsymbol{x}$, we can write (by the chain rule of conditioning)

$$f(\boldsymbol{x}|\boldsymbol{v}^j) = \prod_{J(i)=j} \sum_{l=1}^{N_c} f(\boldsymbol{d}_i^j|c_i = l)p(c_i = l|\boldsymbol{v}_i). \qquad (15)$$

Here we sum over the N_c candidate textures and use the fact that all blocks at the same scale j are independent given the contexts $\boldsymbol{v}^j$. Because $p(c_i|\boldsymbol{v}_i)$ represents the relation between the context and the class label, it is reasonable to assume that it is same for all i within each scale. The ML estimate of $p(c_i|\boldsymbol{v}_i)$ is that which maximizes the likelihood of the image given the $\boldsymbol{v}_i$s [given in (15)]. Maximizing (15) is possible because the likelihoods $f(\boldsymbol{d}_i^j|c_i = l)$ are already available from the multiscale HMT likelihood computations. Note that $p(c_i|\boldsymbol{v}_i)$ is chosen in the ML sense by averaging over the entire image $\boldsymbol{x}$ in (15).

The EM algorithm again comes to our rescue; in fact, we can use it to compute and maximize the posterior (14) directly. We do not specify $p(c_i|\boldsymbol{v}_i)$ directly, but rather specify $p(\boldsymbol{v}_i|c_i)$ and apply Bayes rule

$$p(c_i|\boldsymbol{v}_i) = \frac{p(\boldsymbol{v}_i|c_i)p(c_i)}{p(\boldsymbol{v}_i)}. \qquad (16)$$

Assuming these probabilities to be constant at each scale, set

$$e_{j,m} := p_{c_i}(m), \quad \alpha_{j,\overline{\boldsymbol{v}}_k,m} := p(\boldsymbol{v}_i = \overline{\boldsymbol{v}}_k|c_i = m) \qquad (17)$$

for all i in scale j and $m \in \{1, \ldots, N_c\}$, $k \in \{1, \ldots, N_v\}$. The set of probabilities $\boldsymbol{P} := \{e_{j,m}, \alpha_{j,\overline{\boldsymbol{v}}_k,m}\}$ can be computed using an EM algorithm on the context labeling tree (see the Appendix for details). The context-based Bayes classification then finds the class label that maximizes the contextual posterior distribution $p(c_i|\boldsymbol{d}_i, \boldsymbol{v}_i)$ from (14) [see (18) in the Appendix].

While EM iterations are necessary at each scale to estimate the fusion parameters $e_{j,m}$ and $\alpha_{j,\overline{\boldsymbol{v}}_k,m}$, we note that the algorithm converges rapidly with the initial parameters set to the values estimated in the previous coarser scale. This is because the parameters change little from scale to scale, especially at fine scales where EM iterations are more expensive. Furthermore, at very fine scales, we can actually use the fusion parameters estimated in the coarser scales without re-estimation. This is particularly helpful when the likelihoods $f(\boldsymbol{d}_i^j|c_i)$ at very fine scales are less robust and maximizing $f(\boldsymbol{x}|\boldsymbol{v}^j)$ in (15) does not give the desired $p(c_i|\boldsymbol{v}_i)$s. We employed this technique in the document segmentation example of Section V-B.

D. Pixel-Level Segmentation

Since the Haar wavelet HMT characterizes the joint statistics of the dyadic image squares only down to 2×2 blocks, we do not directly obtain pixel-level segmentations. While the collection of all wavelet and scaling coefficients completely characterizes the original image, the HMT subband independence assumption and the fact that we ignore the scaling coefficients limit our reach to 2×2 blocks. Pixel-level segmentation requires a model for the pixel brightness of each texture class. However, obtaining an appropriate model can be difficult, since in many images the local brightness varies considerably due to shading, etc. For such images, the 2×2 block segmentations will be far more robust, since they rely on inter-pixel dependencies and not local brightness.

Pixel brightness corresponds to the pdf of a single pixel. For our purposes, we fit a Gaussian mixture to the pixel values for each training texture. We can then compute the likelihood of each pixel and extend the above interscale scale fusion algorithm from 2×2 blocks to the pixel level.

E. Implementation Issues

As described above, the interscale fusion algorithm starts at the root node of the context labeling tree and descends to the finest scale to combine all possible coarse scale information. However, at very coarse scales, the likelihoods of the dyadic

[5]Effective contexts can be selected from a library of possible contexts using a classification algorithm such as that proposed in [16], provided that sufficient manually prepared training data are available.

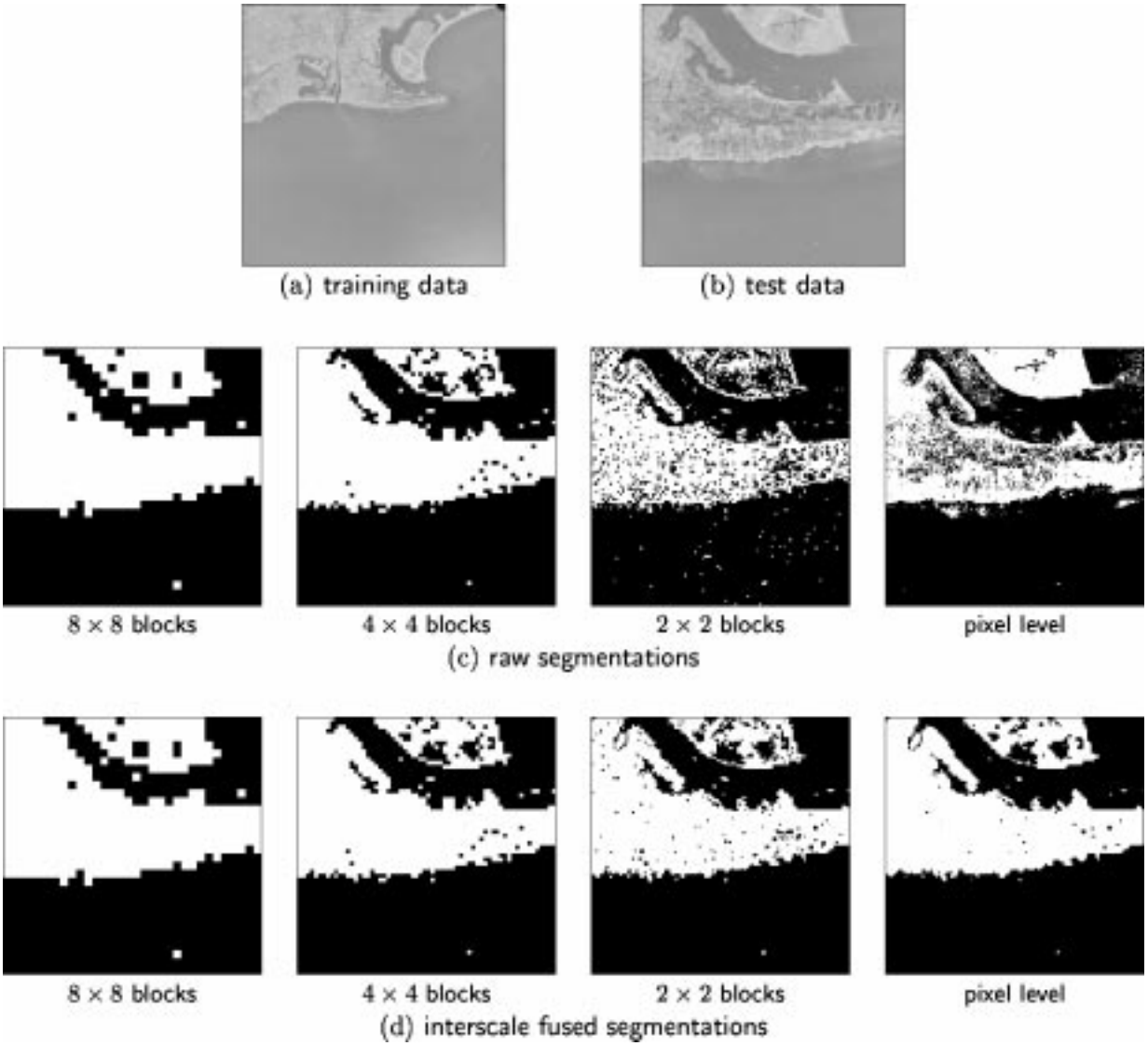

Fig. 5. Aerial photo segmentation using HMTseg. (a) A 1024×1024 aerial photo and (b) 256×256 test subimage $\boldsymbol{x}$. The homogeneous ground/sea regions outside the region (b) were used to train two HMT's. (c) Raw HMT-based multiscale classifications $\widehat{\boldsymbol{c}}_{\mathrm{ML}}^{j}$ of $\boldsymbol{x}$ for 8×8, 4×4, 2×2, and pixel-sized dyadic squares. (d) Final segmentations $\widehat{\boldsymbol{c}}_{\mathrm{MAP}}^{j}$ using Bayesian context-based interscale fusion. The erroneous segmentation of the ground regions in the upper middle portion of the image is due to the large expanses of concrete (runways), whose texture is closer to that of sea than ground in this case.

squares do not contain significant information, since the squares are large and hence are likely to contain several differently textured regions. When fusing multiscale classification results, we therefore ignore the information at very coarse scales.

Ignoring the coarsest scales has the side benefit. As described in Section IV-A, reducing the size of the HMT reduces the computation required for training and likelihood determination. If we start fusing at scale $j_0 > 0$, then we only need the wavelet coefficients, HMT models, and likelihoods at scales $j \geq j_0$. With the Haar transform, starting at scale $j_0 > 0$ is equivalent to dividing the image into the dyadic squares $\boldsymbol{D}_i^{j_0}$ and then performing the HMT likelihood computation independently on each of these squares. This saves a considerable amount of computation and reduces the size of the required homogeneous training images to $2^{J-j_0} \times 2^{J-j_0}$. In practice, we set the starting scale j_0 such that the coarsest raw segmentations have sufficient reliability.

F. Summary of HMTseg Algorithm

The final segmentation algorithm consists of three steps; HMT training, multiscale likelihood computation, and multiscale fusion.

HMTseg Algorithm

1) **Train wavelet-domain HMT models** for each texture using homogeneous training images. To obtain pixel-level segmentation, also train a pixel brightness pdf model.

2) **Compute multiscale likelihoods.** Using the likelihood computation algorithm for the HMT model [12] and (4), compute the likelihood of each dyadic image square at each different scale. This gives the likelihoods $f(\boldsymbol{d}_i^j | c_i^j)$ for each dyadic square. If the trained HMT model is smaller than the test image, repeat the likelihood computations for image subblocks assuming that the blocks are independent (see Section IV-E).

Fig. 6. Document segmentation using HMTseg. (a) A 512×512 training image was hand-segmented, and homogeneous regions were used to train HMTs for text, image, and background textures. (b) A 512×512 test image $\boldsymbol{x}$. (c) Raw HMT-based multiscale classifications $\widehat{\boldsymbol{c}}_{\mathrm{ML}}^{j}$ of $\boldsymbol{x}$ for 8×8, 4×4, 2×2, and pixel-sized dyadic squares. Black, gray, and white represent text, image, and background, respectively. Classification accuracy clearly decreases at fine scales. (d) Final segmentations $\widehat{\boldsymbol{c}}_{\mathrm{MAP}}^{j}$ using Bayesian context-based interscale fusion correctly classify even the angled text on the books. Adding a fourth class (large text) would allow us to correctly classify the text at the bottom of $\boldsymbol{x}$.

3) **Fuse multiscale likelihoods using the labeling tree** to form the multiscale MAP classification. Pick the starting scale j such that the ML classifications of the $\boldsymbol{d}_i^{j-1}$s at scale $j-1$ are reliable enough to obtain the $\boldsymbol{v}_i^j$s. Estimate the parameters $e_{j,m}$ and $\alpha_{j,\overline{\boldsymbol{v}}_k,m}$ to maximize $f(\boldsymbol{x}|\boldsymbol{v}^j)$ in (15) using the EM algorithm in the Appendix. Each EM iteration updates the contextual posterior distribution $p(c_i|\boldsymbol{d}_i, \boldsymbol{v}_i)$. When converged, determine the c_i that maximizes $p(c_i|\boldsymbol{d}_i, \boldsymbol{v}_i)$. Continue fusion at scale $j+1$ based on the $\boldsymbol{v}^{j+1}$ formed using the c_i^js obtained at scale j. Continue the process for all scales until the finest scale is reached.

V. Examples

Fig. 3 demonstrated the HMTseg process on a synthetic data example. Here we illustrate two real-world image segmentation problems.

A. Aerial Photo Segmentation

We trained wavelet HMTs for "sea" and "ground" textures using hand-segmented blocks from the 1024×1024 aerial photo [14] in Fig. 5(a). For training data, we extracted 100×100 homogeneous ground [upper-left corner of Fig. 5(a)] and sea [lower-right corner of Fig. 5(a)] images. Then, from each 100×100 image, we randomly picked ten (overlapping) 64×64 blocks. With this training data and intra-scale tying in the HMT models, the EM training algorithm converged in less than 15 iterations.

Choosing $j_0 = 3$ for the starting scale (corresponding to 6-scale quad-trees on 64×64 image blocks), we segmented the 256×256 test image in Fig. 5(b).

Fig. 5(c) shows the raw classification results. Pixel-level raw segmentation was obtained using 2-density Gaussian mixture models for pixel brightness of the ground and sea textures. Fig. 5(d) illustrates the segmentation resulting from 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 concrete texture more resembling sea), we observe excellent segmentation results at all scales.

B. Document Segmentation

We trained HMT and pixel brightness models for "text," "image," and "background" textures using hand-segmented blocks from the 512×512 document in Fig. 6(a) [34]. Again, we randomly picked ten 64×64 homogeneous regions for each texture from Fig. 6(a) as training data set. The EM trainings of the models converged within 20 iterations.

Choosing $j_0 = 4$ for the starting scale (corresponding to 6-scale quad-trees on 64×64 image blocks), we segmented the 512×512 test image in Fig. 6(b). Fig. 6(c) shows the raw classification results. As expected, we observe many classification errors. The pixel-level segmentation, in particular, is not reliable (all text was classified as imagery). Fig. 6(d) illustrates the segmentation resulting from coarse-to-fine interscale fusion. All text regions are segmented well, including the text surrounded by images on the books. At the bottom, we observe the large-font title text segmented as imagery. This is because the homogeneous texture inside each large letter has properties more similar to imagery than (small-font) text. The background regions are correctly segmented, even though the brightness of the background varies in different areas and is corrupted by a noise-like feature caused by text on the reverse side of the page. In the fusion step, we estimated the fusion parameters only down to 2×2 block scale, since incorrect pixel likelihoods make the estimation unreliable.

VI. Conclusions

In this paper, we have developed a new framework for multiscale Bayesian image segmentation based on wavelet-domain HMT models. By concisely modeling and fusing the statistical behavior of textures at multiple scales, the HMTseg algorithm produces a robust and accurate segmentation of texture images. HMTseg yields not one final segmentation but a range of segmentations at different scales.

While we have illustrated with an aerial photograph and a document image, HMTseg can be applied to many different image types, including radar/sonar images [35] and medical images. Furthermore, because the HMT modeling framework extends trivially to higher-dimensional data, we can employ HMTseg to segment multidimensional data such as geophysical surveys. One-dimensional signals, such as speech and geophysical well-logs, are also within HMTseg's purview.

As an added bonus, HMTseg has the potential to segment wavelet-compressed data directly without re-expanding to the space domain. HMTseg thus provides a natural vehicle for developing joint segmentation/compression algorithms [36].

Promising avenues for future HMTseg research include the investigation of wavelet basis representation different from Haar, simplified universal HMT modeling [30], more accurate (but complicated) interscale fusion algorithms, and the analysis of multiscale classification errors [37].

Appendix
EM Algorithm for Context Labeling Tree

Our goal is to find $p(c_i|v_i)$ maximizing $f(\boldsymbol{x}|\boldsymbol{v}^j)$ in (15). We precompute the conditional likelihoods $f(\boldsymbol{d}_i^j|c_i)$ for all $c_i \in \{1, \ldots, N_c\}$ using (5) by sweeping up the HMTs from the leaves to node i [12]. Recall the definitions of $e_{j,m}$, $\alpha_{j,\overline{\boldsymbol{v}}_k,m}$, and $\boldsymbol{P}$ from (17). The EM algorithm runs as follows.

Initialize: Set $I = 0$ and choose $\boldsymbol{P}^0$.

(A natural choice for $\boldsymbol{P}^0$ is the set of parameters obtained in the previous, next coarser scale.)

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

$$p\left(c_i = m \middle| \boldsymbol{d}_i^j, \boldsymbol{v}_i^j\right) = \frac{e_{j,m}\alpha_{j,\boldsymbol{v}_i,m} f\left(\boldsymbol{d}_i^j \middle| c_i = m\right)}{\sum_{l=1}^{N_c} e_{j,l}\alpha_{j,\boldsymbol{v}_i,l} f\left(\boldsymbol{d}_i^j \middle| c_i = l\right)}. \quad (18)$$

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

$$e_{j,m} = \frac{1}{2^{2j}} \sum_i p\left(c_i = m \middle| \boldsymbol{v}_i^j, \boldsymbol{d}_i^j\right) \quad (19)$$

$$\alpha_{j,\overline{\boldsymbol{v}}_k,m} = \frac{1}{2^{2j} \cdot e_{j,m}} \sum_{i \text{ with } \boldsymbol{v}_i^j = \overline{\boldsymbol{v}}_k} p\left(c_i = m \middle| \boldsymbol{v}_i^j, \boldsymbol{d}_i^j\right) \quad \text{for each } \overline{\boldsymbol{v}}_k,\ k \in \{1, \ldots, N_v\}. \quad (20)$$

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

References

[1] R. Haralick and L. Shapiro, "Image segmentation techniques," *Comput. Vis., Graph., Image Process.*, vol. 29, pp. 100–132, 1985.

[2] C. Therrien, "An estimation-theoretic approach to terrain image segmentation," *Comput. Vis., Graph., Image Process.*, vol. 22, pp. 313–326, 1983.

[3] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, Jan. 1987.

[4] C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Processing*, vol. 3, pp. 162–177, Mar. 1994.

[5] C. Fosgate, H. Krim, W. Irving, W. Karl, and A. Willsky, "Multiscale segmentation and anomaly enhancement of SAR imagery," *IEEE Trans. Image Processing*, vol. 6, pp. 7–20, Jan. 1997.

[6] J. Li, R. M. Gray, and R. A. Olshen, "Multiscale image classification by hierarchical modeling with two dimensional hidden Markov models," *IEEE Trans. Inform. Theory*, vol. 46, no. 5, pp. 1826–1841, 2000.

[7] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 1998.

[8] E. P. Simoncelli, "Statistical models for images: Compression, restoration and synthesis," in *Proc. 31st Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Nov. 1997, pp. 673–678.

[9] J. Pesquet, H. Krim, and E. Hamman, "Bayesian approach to best basis selection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing '96*, Atlanta, GA, 1996, pp. 2634–2637.

[10] F. Abramovich, T. Sapatinas, and B. W. Silverman, "Wavelet thresholding via a Bayesian approach," *J. R. Statist. Soc. B*, vol. 60, pp. 725–749, 1998.

[11] P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized-Gaussian priors," in *Proc. IEEE Signal Processing Int. Symp. Time-Frequency Time-Scale Analysis*, Pittsburgh, PA, Oct. 6–9, 1998, pp. 633–636.

[12] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, pp. 886–902, Apr. 1998.

[13] H. Chipman, E. Kolaczyk, and R. McCulloch, "Adaptive Bayesian wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 92, 1997.

[14] The usc-sipi image database. [Online]. Available: http://sipi.usc.edu/services.html
[15] C. Wu and P. C. Doerschuk, "Tree approximations to Markov random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 4, pp. 391–402, 1995.
[16] H. Cheng and C. A. Bouman, "Trainable context model for multiscale segmentation," in *Proc. IEEE Int. Conf. Image Processing '98*, Chicago, IL, Oct. 4–7, 1998.
[17] H. Cheng, C. A. Bouman, and J. P. Allebach, "Multiscale document segmentation," in *Proc. IS&T 50th Annu. Conf.*, Cambridge, MA, May 18–23, 1997, pp. 417–425.
[18] H. Choi and R. G. Baraniuk, "Image segmentation using wavelet-domain classification," in *Proc. SPIE Conf. Math. Modeling, Bayesian Estimation, Inverse Problems*, vol. 3816, Denver, CO, July 1999, pp. 306–320.
[19] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, 1984.
[20] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. R. Statist. Soc. B*, vol. 36, pp. 192–225, 1974.
[21] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 959–963, 1985.
[22] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss–Markov random field models for texture segmentation," *IEEE Trans. Image Processing*, vol. 6, pp. 251–267, Feb. 1997.
[23] A. Kim and H. Krim, "Hierarchical stochastic modeling of SAR imagery for segmentation/compression," *IEEE Trans. Signal Processing*, vol. 47, pp. 458–468, Feb. 1999.
[24] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, pp. 1549–1560, Nov. 1995.
[25] J. Li and R. M. Gray, "Text and picture segmentation by the distribution analysis of wavelet coefficients," in *Proc. IEEE Int. Conf. Image Processing '98*, Chicago, IL, Oct. 1998.
[26] M. H. Gross, R. Koch, L. Lippert, and A. Dreger, "Multiscale image texture analysis in wavelet spaces," in *Proc. IEEE Int. Conf. Image Processing '94*, Austin, TX, 1994.
[27] M. Vetterli and J. Kovačević, *Wavlets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[28] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
[29] R. Nowak, "Multiscale hidden Markov models for Bayesian image analysis," in *Bayesian Inference in Wavelet Based Models*, P. Müller and B. Vidakovic, Eds. Berlin, Germany: Springer-Verlag, 1999, pp. 243–266.
[30] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden Markov models," *IEEE Trans. Image Processing*, vol. 10, pp. 1056–1068, July 2001.
[31] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.
[32] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge, MA: MIT Press, 1998.
[33] M. S. Crouse and R. G. Baraniuk, "Simplified wavelet-domain hidden Markov models using contexts," in *Proc. 31st Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Nov. 1997.
[34] H. Choi and R. Baraniuk, "Multiscale document segmentation using wavelet-domain hidden Markov models," in *Proc. IST/SPIE 12th Annu. Symp. Electronic Imaging 2000*, San Jose, CA, Jan. 2000.
[35] V. Venkatachalam, H. Choi, and R. G. Baraniuk, "Multiscale SAR image segmentation using wavelet-domain hidden Markov tree models," in *Proc. SPIE 14th Int. Symp. Aerospace/Defense Sensing, Simulation, Control*, Orlando, FL, April 2000.
[36] R. L. de Queiroz, R. Buckley, and M. Xu, "Mixed raster content (MRC) model for compound image compression," in *Proc. IS&T/SPIE Symp. Electronic Imaging, Visual Communication, Image Processing*, San Jose, CA, Feb. 1999.
[37] B. Hendricks, H. Choi, and R. Baraniuk, "Analysis of wavelet-domain multiscale classification using Kullback–Leibler distances," in *Proc. 33rd Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Oct. 1999.

Hyeokho Choi (M'98) was born in Korea in 1969. He received the B.S. degree in control and instrumentation engineering (summa cum laude) from Seoul National University, Seoul, Korea, in 1991, and the M.S. and Ph.D degrees in electrical engineering from the University of Illinois at Urbana-Champaign, in 1993 and 1998, respectively. His research was in the area of computed imaging systems and signal processing.

Since January 1998, he has been at Rice University, Houston, TX, where he is currently a Research Professor with the Department of Electrical and Computer Engineering. His current research interests lie in the area of statistical signal processing, pattern recognition, wavelet theory, and imaging systems.

Richard G. Baraniuk (S'85–M'93–SM'98) received the B.Sc. degree in 1987 from the University of Manitoba, Winnipeg, MB, Canada, the M.Sc. degree in 1988 from the University of Wisconsin, Madison, and the Ph.D. degree in 1992 from the University of Illinois at Urbana-Champaign, all in electrical engineering.

In 1986, he was a Research Engineer with Omron Tateisi Electronics, Kyoto, Japan. After spending 1992–1993 with the Signal Processing Laboratory of Ecole Normale Supérieure, Lyon, France, he joined Rice University, Houston, TX, where he is currently a Professor of electrical and computer engineering. He spent Autumn 1998 at the Isaac Newton Institute, Cambridge University, Cambridge, U.K., as the Rosenbaum Fellow. His research interests lie in the area of signal and image processing and include wavelets, probabilistic models, networks, and time-frequency analysis. He serves on the editorial board of *Applied and Computational Harmonic Analysis*.

Dr. Baraniuk received a NATO Postdoctoral Fellowship from NSERC in 1992, the National Young Investigator award from the National Science Foundation in 1994, a Young Investigator Award from the Office of Naval Research in 1995, the Rosenbaum Fellowship from the Newton Institute in 1998, the C. Holmes MacDonald National Outstanding Teaching Award from Eta Kappa Nu in 1999, the Charles Duncan Junior Faculty Achievement Award from Rice in 2000, and the ECE Young Alumni Achievement Award from the University of Illinois in 2000. He was co-author on a paper (with M. Crouse and R. Nowak) that received the IEEE Signal Processing Society (SPS) Junior Paper Award in 2001. He is a member of the Signal Processing Theory and Methods Technical Committee of the IEEE SPS.