Multiscale SAR Image Segmentation using Wavelet-domain Hidden Markov Tree Models

Vidya Venkatachalam, Hyeokho Choi, and Richard G. Baraniuk

Computational Mathematics Laboratory
Department of Electrical & Computer Engineering
Rice University
Houston, TX, USA

ABSTRACT

We study the segmentation of SAR imagery using wavelet-domain Hidden Markov Tree (HMT) models. The HMT model is a tree-structured probabilistic graph that captures the statistical properties of the wavelet transforms of images. This technique has been successfully applied to the segmentation of natural texture images, documents, etc. However, SAR image segmentation poses a difficult challenge owing to the high levels of speckle noise present at fine scales. We solve this problem using a "truncated" wavelet HMT model specially adapted to SAR images. This variation is built using only the coarse scale wavelet coefficients. When applied to SAR images, this technique provides a reliable initial segmentation. We then refine the classification using a multiscale fusion technique, which combines the classification information across scales from the initial segmentation to correct for misclassifications. We provide a fast algorithm, and demonstrate its performance on MSTAR clutter data.

Keywords: segmentation, wavelets, SAR, ATR, MSTAR

1. INTRODUCTION

The primary objective of Synthetic Aperture Radar (SAR) image segmentation is the separation of the SAR image into simple regions with homogeneous properties, each with a different "texture".[1] There exist several texture segmentation schemes which are designed for natural/photographic images. However, unlike natural images, SAR images characteristically have a particular kind of noise, called "speckle," which is introduced due to the coherent imaging process. Speckle is caused by the random interactions (constructive and destructive) of the reflections from the scatterers on the imaged surface. At fine resolutions, this noise becomes significant and often masks the underlying signal. This causes serious problems in the segmentation process. Hence, segmentation techniques that work successfully on natural images do not perform as well on SAR images.

It is clear that the resolution at which we image the terrain becomes important in the segmentation process. The optimal resolution level at which the speckle noise ceases to be dominant is a function of the spatial distribution of the scatterers on the imaged surface, and is not in general, fixed. In order to deal with this situation, what we need is a scheme that adapts itself to different resolution levels. This suggests the use of multiscale (multiresolution) segmentation schemes.

Choi and Baraniuk[2] proposed an image texture segmentation algorithm, HMTseg, based on wavelets and the Hidden Markov Tree (HMT) model. The strength of HMTseg is in the fact that it produces segmentations at different resolution levels. This is very significant when resolution is an issue in the segmentation process, as it is with SAR images. In this paper, we study HMTseg and adapt it to SAR image segmentation.

In order to make this paper reasonably self-contained, we first explain the fundamental features of the HMTseg algorithm (complete details can be obtained from Choi and Baraniuk[2]), and then detail how it can be modified for SAR images. Finally, we demonstrate its performance through simulation results on MSTAR clutter data.

Further author information: Email: {vidyav,hchoi,richb}@rice.edu, Web: www.dsp.rice.edu.
This work was supported by DARPA/AFOSR grant F49620–97–1–0513, NSF grants MIP–9457438 and CCR–9973188, ONR grant N00014–99–1–0813, and the Texas Instruments Leadership University Program.

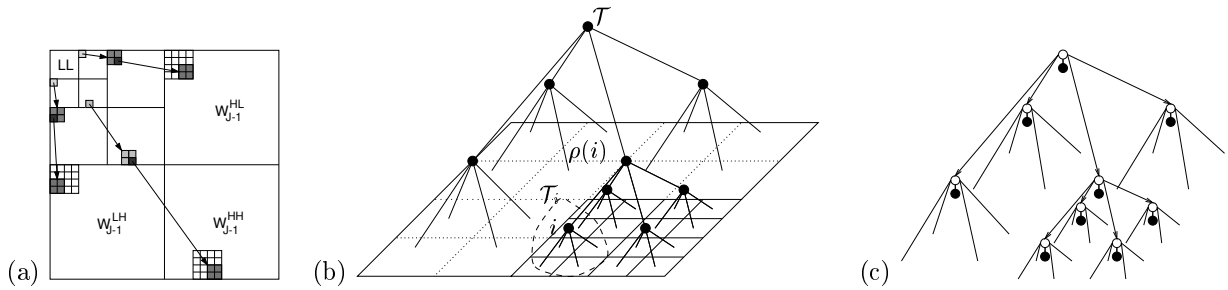

Figure 1. *(a) Parent-child dependencies of the three 2-D wavelet transform subbands: Each arrow points from a parent wavelet coefficient to its four children at the next finer scale. (b) More detailed view of the quad-tree structure in one subband. Each black node corresponds to a wavelet coefficient. 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) 2-D wavelet Hidden Markov Tree (HMT) model. 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 (**W6**), we connect the states vertically across scale in Markov-1 chains.*

2. WAVELET-BASED MULTISCALE IMAGE SEGMENTATION

2.1. Multiscale segmentation

In most segmentation algorithms, we process images through a *classification window* of a fixed size and assume that all pixels in the window belong to the same class. A typical segmentation would then assign one class label to all pixels in the window. 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 (scale corresponds to the window size) behaviors should be utilized to properly segment both large, homogeneous regions and detailed boundary regions. This is the key idea in *multiscale segmentation*, where we combine the results of many classification windows of different sizes.

2.2. Wavelets and multiscale analysis

The wavelet transform lends itself naturally to a multiscale texture segmentation. We focus here on the pyramidal multiscale Haar wavelet construction for discrete images.[3] The construction of Haar wavelet coefficients of an image can be explained using four 2-D wavelet filters: the local smoother $h_{\mathrm{LL}} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, horizontal edge detector $g_{\mathrm{LH}} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$, vertical edge detector $g_{\mathrm{HL}} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$, and corner detector $g_{\mathrm{HH}} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$.

To compute the wavelet transform of a $2^J \times 2^J$ discrete image $\boldsymbol{x}$, first we set $\boldsymbol{u}_J[k,l] := \boldsymbol{x}[k,l]$, $0 \le k,l \le 2^J - 1$. Next, we 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 — $\boldsymbol{u}_{J-1}$, $\boldsymbol{w}^{\mathrm{LH}}_{J-1}$, $\boldsymbol{w}^{\mathrm{HL}}_{J-1}$, and $\boldsymbol{w}^{\mathrm{HH}}_{J-1}$, 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 $\begin{bmatrix} \boldsymbol{u}_{J-1} & \boldsymbol{w}^{\mathrm{HL}}_{J-1} \\ \boldsymbol{w}^{\mathrm{LH}}_{J-1} & \boldsymbol{w}^{\mathrm{HH}}_{J-1} \end{bmatrix}$. 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. 1(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 $\boldsymbol{w}^{\mathrm{LH}}_j$, $\boldsymbol{w}^{\mathrm{HL}}_j$, and $\boldsymbol{w}^{\mathrm{HH}}_j$ are high- and band-pass filtered, *edge-detected*, versions of the image that correspond strongly to edges in the horizontal, vertical, and diagonal orientations, respectively. For example, the wavelet coefficient $\boldsymbol{w}^{\mathrm{LH}}_{J-1}[k,l]$, $0 \le k,l \le 2^{J-1} - 1$, 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. 1(a) and (b).[4] First assume

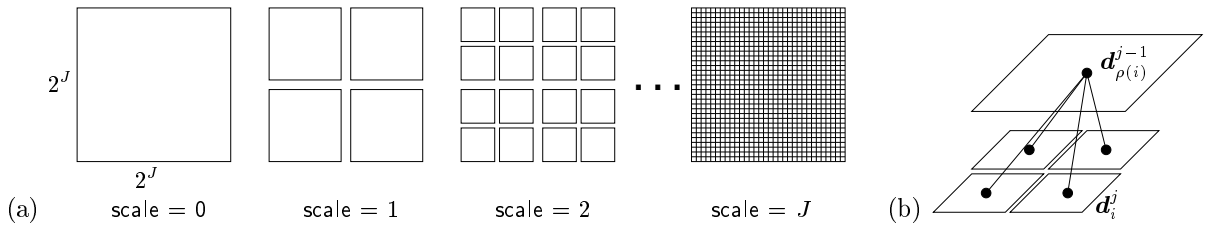

Figure 2. *(a) Image $\boldsymbol{x}$ divided into dyadic squares $\boldsymbol{d}_i^j$ at different scales. Each dyadic square can be associated with a subtree of Haar wavelet coefficients. (b) Quad-tree structure of dyadic squares. The dyadic square $\boldsymbol{d}_{\rho(i)}^{j-1}$ splits into four child squares at scale j.*

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* $\boldsymbol{w}_{j+1}^{\mathrm{LH}}[2k,2l]$, $\boldsymbol{w}_j^{\mathrm{LH}}[2k,2l+1]$, $\boldsymbol{w}_j^{\mathrm{LH}}[2k+1,2l]$, and $\boldsymbol{w}_j^{\mathrm{LH}}[2k+1,2l+1]$. 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.[5]

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. 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 descendants (see Fig. 1(b)).

With the 2-D Haar wavelet transform, there is an obvious correspondence between the wavelet coefficients and the dyadic squares (see Fig. 2(a)), which are obtained by iteratively dividing the image into equal-size quadrants. Note 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 wavelet coefficient w_i with a special property: all wavelet coefficients in the subtree $\mathcal{T}_i$ rooted at w_i depend exclusively on the pixel values in $\boldsymbol{d}_i$. This results in a *natural scaling* of the image information in terms of dyadic squares of different sizes. While the above analysis holds true even for larger wavelet filters (which are better suited to represent smooth images), the Haar system is more appropriate for our purpose due to its direct connection with dyadic squares in the image.

2.3. Multiscale wavelet-based statistical models

Our segmentation approach belongs to the class of Bayesian statistical techniques. Such 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. In an image subregion $\boldsymbol{X}_r \subset \boldsymbol{X}$ of class c, the pixels are assumed distributed with joint probability density function (pdf) $f(\boldsymbol{x}_r|c)$. In this framework, the image segmentation problem is characterized 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. Since we adopt a multiscale segmentation routine on dyadic squares, we require a pixel pdf model for each class that is suited to the dyadic squares.

Owing to the practical difficulties in determining the complete joint pixel pdf, we often impose structures on both the possible image regions and on the pixel pdfs. This leads to the idea of statistical image models, which are playing an increasingly fundamental role in image classification and segmentation. In many cases, we can further simplify the problem by working in a suitable transform domain. The idea is that the transform-domain model obtained from a linear, invertible transform "restructures" the image, leaving transform coefficients whose structure is easier to model.

The wavelet transform described earlier can be interpreted as a multiscale edge detector that represents the singularity content of an image at multiple scales and three different orientations. Wavelet transform coefficients have certain important properties which are outlined below[5,2]:

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

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

W3. Energy Compaction: The wavelet transforms of real-world images tend to be sparse, except at edges.

W4. Decorrelation: The wavelet coefficients of real-world images tend to be approximately decorrelated (and the only correlations that remain are local, see **W6** below).

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

W6. Persistence and Clustering: Large/small values of wavelet coefficients tend to propagate across scale in the wavelet quad-tree.[6,7] If a particular wavelet coefficient is large/small, then adjacent coefficients are very likely to also be large/small.[5,8]

These properties indicate that the wavelet transforms of real-world images have a local dependency structure that should not be ignored. With these facts in mind, we now turn to modeling images in the wavelet domain.

2.3.1. The wavelet-HMT model

Owing to the multiscale singularity detection property and tree structure (Fig.1) of wavelet coefficients, image singularities manifest themselves as cascades of large wavelet coefficients through scale along the branches of the quad-tree.[9] Conversely, smooth regions lead to cascades of small coefficients. This multiscale singularity characterization makes the wavelet domain natural for modeling texture images. Crouse et al.[5] developed the *hidden Markov tree* (HMT) model, a parametric statistical model for wavelet transforms. The HMT owes its flexibility to two key features:
1) It differentiates between "large" and "small" wavelet coefficients by associating with each coefficient a binary state variable that controls its size. Assuming that each coefficient is Gaussian distributed when conditioned on its state, the marginal distribution of each coefficient is modeled as a Gaussian mixture. Thus, to each wavelet coefficient W_i, we associate a discrete hidden state S_i that takes on the values $m = \mathsf{S},\mathsf{L}$, signifying the small and large coefficient values, 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

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

where $f(w_i|S_i = m) \sim N(\mu_{i,m}, \sigma^2_{i,m})$ and $p_{S_i}(\mathsf{S}) + p_{S_i}(\mathsf{L}) = 1$.
2) In order to capture the fact that large and small wavelet coefficients cascade through scale, the states are connected in a Markovian probabilistic quad-tree that mirrors that of the wavelet transform. Each state-to-state link has an underlying state transition matrix that controls (probabilistically) the persistence of large and small states down the tree. In particular, consider the modeling of one subband of the wavelet transform. In Fig. 1(c) we picture the wavelet coefficients as black nodes and their associated hidden states as white nodes. To capture **W6**, we connect the hidden states in a directed Markov-1 probabilistic graph. For each parent-child pair of hidden states $\{S_{\rho(i)}, S_i\}$, the state transition probabilities $\epsilon^{\rho(i),m}_{i,m'}$ for $m, m' = \mathsf{S},\mathsf{L}$ 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 $\begin{bmatrix} \epsilon^{\rho(i),\mathsf{S}}_{i,\mathsf{S}} & \epsilon^{\rho(i),\mathsf{S}}_{i,\mathsf{L}} \\ \epsilon^{\rho(i),\mathsf{L}}_{i,\mathsf{S}} & \epsilon^{\rho(i),\mathsf{L}}_{i,\mathsf{L}} \end{bmatrix} = \begin{bmatrix} \epsilon^{\rho(i),\mathsf{S}}_{i,\mathsf{S}} & 1-\epsilon^{\rho(i),\mathsf{S}}_{i,\mathsf{S}} \\ 1-\epsilon^{\rho(i),\mathsf{L}}_{i,\mathsf{L}} & \epsilon^{\rho(i),\mathsf{L}}_{i,\mathsf{L}} \end{bmatrix}$.
The above two features together complete specify the HMT model for one subband. In our HMT algorithm, we assume that all the three subbands of the 2-D wavelet transform are *independent*. Each is then separately modeled exactly as described above.

The quad-tree HMT model is thus completely determined by the following parameters

$$\boldsymbol{\Theta} := \begin{cases} \mu_{i,m},\ \sigma^2_{i,m} & \text{(mixture means and variances)} \\ \epsilon^{\rho(i),m}_{i,m'} & \text{(state transition probabilities from } S_{\rho(i)} \text{ to } S_i) \\ p_{S_0}(m) & \text{(probability mass function for the root node state).} \end{cases}$$

Given $\boldsymbol{\Theta}$, the HMT models the joint pdf of the subband wavelet coefficients. The complete wavelet HMT model $\mathcal{M}$ consists of three HMTs (one for each wavelet subband). Denoting the parameter vectors for the three subband HMTs as $\boldsymbol{\Theta}^{\mathrm{HH}}, \boldsymbol{\Theta}^{\mathrm{HL}}$ and $\boldsymbol{\Theta}^{\mathrm{LH}}$, respectively, we have $\mathcal{M} := \{\boldsymbol{\Theta}^{\mathrm{HH}}, \boldsymbol{\Theta}^{\mathrm{HL}}, \boldsymbol{\Theta}^{\mathrm{LH}}\}$. The HMT is thus a parametric model for the joint pdf of the wavelet coefficients. Using the independent subband assumption, we can write

$$f(\boldsymbol{w}|\mathcal{M}) := f(\boldsymbol{w}^{\mathrm{LH}}|\boldsymbol{\Theta}^{\mathrm{LH}})\, f(\boldsymbol{w}^{\mathrm{HL}}|\boldsymbol{\Theta}^{\mathrm{HL}})\, f(\boldsymbol{w}^{\mathrm{HH}}|\boldsymbol{\Theta}^{\mathrm{HH}}). \tag{2}$$

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 amount of training data is available. Fortunately, wavelet coefficients tend to exhibit similar statistical characteristics within the same scale,[5,10] and so we can often use the same parameters for those coefficients. This nodal *tying* reduces the number of parameters considerably, avoiding the risk of overfitting the model.[5,11] Grouping the model parameters into the vector $\mathcal{M}$, the result is a highly structured Gaussian mixture model $f(\boldsymbol{w}|\mathcal{M})$ that approximates the overall joint pdf of the wavelet coefficients $\boldsymbol{W}$.

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

3. THE HMT BASED SEGMENTATION ALGORITHM

The HMTseg algorithm relies on three separate tree structures: the wavelet transform quad-tree, the HMT, and a labeling tree. The key steps in the algorithm are given below.

3.1. Wavelet-domain HMT model training

We can *train* the wavelet HMT model parameters for each texture to match a set of training data consisting of homogeneous images. The iterative expectation-maximization (EM) algorithm finds the locally optimal (in the Maximum Likelihood (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. Iteration of the two steps is guaranteed to converge to an $\mathcal{M}$ that locally maximizes the likelihood.[11] In the HMT, each EM iteration consists of an up/down sweep through the tree ($O(n)$ cost for n wavelet coefficients). Once trained, the HMT provides a close approximation to the full joint pdf of the wavelet coefficients.

3.2. Multiscale likelihood computation

Using the likelihood computation algorithm for the HMT model,[5] we can compute the likelihood $f(\widetilde{\boldsymbol{d}}_i|\mathcal{M}_c)$, $c \in \{1, 2, \ldots, N_c\}$ of each dyadic image square $\widetilde{\boldsymbol{d}}_i$ at each different scale. Specifically, given a set of 2-D HMT model parameters $\mathcal{M}$ and the wavelet transform $\widetilde{\boldsymbol{w}}$ of a test image, we can compute the *likelihood* $f(\widetilde{\boldsymbol{w}}|\mathcal{M})$ that the image was generated by the model.[5] Furthermore, due 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.[5] Suppose this subband has HMT parameters $\boldsymbol{\Theta}$. Given the conditional likelihood $\beta_i(m) := f(\mathcal{T}_i|S_i = m, \boldsymbol{\Theta})$ obtained by sweeping up the quad-tree from the leaves to node i (see Crouse et al.[5]), the likelihood of the coefficients in $\mathcal{T}_i$ can be computed as

$$f(\mathcal{T}_i|\boldsymbol{\Theta}) = \sum_{m=\mathsf{S},\mathsf{L}} \beta_i(m)\, p(S_i = m|\boldsymbol{\Theta}), \tag{3}$$

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

*The EM algorithm derived for 1-D HMT models in Crouse et al.[5] applies without modification in 2-D if we interpret the parent-child relations between nodes appropriately for quad-trees. For a general theory of probabilistic graphs and training algorithms, see Frey.[12]

Now we make the connection with the dyadic squares in the image. 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 quadtrees. Using three upsweeps, we can easily compute the likelihood (3) for each of these subtrees. Then, under the independent subband assumption, we have

$$f(\boldsymbol{d}_i|\mathcal{M}) = f(\mathcal{T}_i^{\mathrm{LH}}|\boldsymbol{\Theta}^{\mathrm{LH}})\, f(\mathcal{T}_i^{\mathrm{HL}}|\boldsymbol{\Theta}^{\mathrm{HL}})\, f(\mathcal{T}_i^{\mathrm{HH}}|\boldsymbol{\Theta}^{\mathrm{HH}}). \tag{4}$$

Based on these likelihoods, the ML classification is given by

$$\widehat{c}_i^{\mathrm{ML}} := \arg\max_{c\in\{1,2,\ldots,N_c\}} f(\widetilde{\boldsymbol{d}}_i|\mathcal{M}_c) \tag{5}$$

This yields the ML *raw classifications* $\widehat{c}_i^{\mathrm{ML}}$ for a range of scales j. This raw ML segmentation process can be completed in just $O(n)$ computations for an n-pixel image. It yields a set of J different segmentations $\boldsymbol{c}_{\mathrm{ML}}^j$, $j = 0, 1, \ldots, J-1$, one for each different scale j of dyadic square. Thus, we obtain segmentation at different resolution levels of the image using the HMT model.

3.3. Interscale decision fusion

The basic HMT segmentation algorithm provides raw segmentation at different scales. The next step is to intelligently combine these decisions to obtain a final high-quality segmentation which has both the robustness of the large block segmentations, and the resolution of the small block segmentations. We do this using a Bayesian interscale fusion technique.[13]

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.

One way to exploit these parent-child dependencies between the dyadic squares is to build yet another tree-structured probability model, the *labeling tree*. Akin to the HMT, the labeling tree models the dependencies between dyadic squares across scale in a Markov 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.) 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 maximum a posteriori (MAP) estimate of the class label $\widehat{c}_i^{\mathrm{MAP}}$ of each dyadic square $\boldsymbol{d}_i$ as follows. Since this is a Bayesian 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 of C_i given the image, denoted here by $p(c_i|\boldsymbol{x})$, the MAP classification of dyadic square $\boldsymbol{d}_i$ corresponds to the class label that maximizes $p(c_i|\boldsymbol{x})$; i.e.,

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

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 $\boldsymbol{d} := \{\boldsymbol{d}_i\}$ denote the collection of all dyadic squares (at all scales), and note that $\boldsymbol{d}$ contains complete information on the image $\boldsymbol{x}$ (many times over). A posterior equivalent to (7) is thus

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

Since computation and maximization of (8) is intractable in practice, we 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

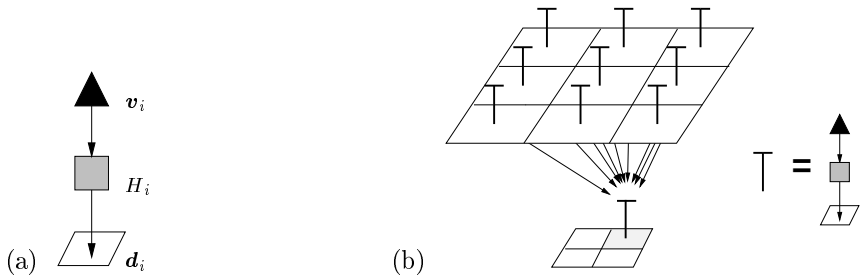

Figure 3. *(a) The dyadic square $\boldsymbol{d}_i$, with the assigned hidden feature variable $\boldsymbol{H}_i$ and the associated context $\boldsymbol{v}_i$ (b) Contextual labeling tree. The context of the child square is determined by the decision results of the parent plus its eight neighbors.*

echoing the structure of the wavelet coefficient quad-tree, we can construct a probabilistic tree to model the posterior (8) based on the dyadic square quad-tree of Fig. 2(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; the approach adopted here uses the concept of *context.*[14] The main idea of the contextual model is as follows. To each dyadic square $\boldsymbol{d}_i^j$, we assign the hidden feature variable $\boldsymbol{H}_i^j$ that we assume controls the textural properties of the square. The hidden feature variables play a role analogous to the hidden states in the HMT: given the values of the states, all wavelet coefficients are independent. Now, to each $\boldsymbol{d}_i^j$ with hidden feature variable $\boldsymbol{H}_i^j$, we assign the (deterministic) context vector $\boldsymbol{v}_i^j$, which is formed from information about other dyadic squares and hidden feature variables.

Based on our training data, we adopt the following simplification. While each $\boldsymbol{v}_i$ is potentially a function of all $\boldsymbol{H}_k$, $k \neq i$, we can employ a tree organization; i.e., each $\boldsymbol{v}_i^j$ at scale j will receive information from nine scale $j-1$ feature variables, the parent feature variable $\boldsymbol{H}_{\rho(i)}$ and the parent's eight nearest neighboring $\boldsymbol{H}_k$ (see Fig. 3). We term this organization the *contextual 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 $\boldsymbol{d}_i^j$.

To further simplify the problem, we set $\boldsymbol{H}_i = C_i$, so that the feature variable controlling the texture of each square is merely the texture label itself. Next, inspired by the success of the hybrid tree model,[15] we use a simple context structure, wherein each context vector $\boldsymbol{v}_i$ contains 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. With these simplifications, we obtain an easily computable expression for the posterior. Complete details of the exact algorithm used to compute the posterior are found in Choi and Baraniuk.[2] We thus obtain a final segmentation which takes the raw HMT-based segmentation, and builds on it using a context-based decision-guiding process from the reliable coarse scale segmentations to fine scales.

3.4. Pixel-level segmentation

This is the final step in the HMTseg algorithm. Since the Haar wavelet HMT characterizes the joint statistics of dyadic image squares only down to 2×2 blocks, we do not directly obtain pixel-level segmentations.[†] Pixel-level segmentation requires a model for the pixel brightness of each texture class. 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.[2] We can then compute the likelihood of each pixel and extend the above interscale scale fusion algorithm to the pixel level. This gives us the final classification.

4. HMTseg ADAPTED TO SAR DATA

The HMTseg algorithm described above gives very good results when applied to natural images; however, it performs unsatisfactorily on SAR images. The main reason is that SAR image information at fine resolutions is highly unreliable owing to speckle noise, and consequently, should not be used in the training of the HMT model. We propose

[†]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 limits our reach to 2×2 blocks.

here a version of the HMT that discards this unreliable information, while maintaining the attractive properties of the original HMT model.

The main difference between this algorithm and the original HMTseg lies in the training process. When using SAR images as training data, it is crucial to first determine which resolution levels are acceptable as input. This information is obtained using knowledge of the imaging process, relative positions of the scatterers on the imaged surface, receiver positions, etc.

Suppose we determine that the highest resolution acceptable is $\boldsymbol{d}^J$ (we choose this to be dyadic in keeping with the wavelet framework). Then we use only those wavelet coefficients corresponding to dyadic squares no smaller than $\boldsymbol{d}^J$ when training the HMT model. Thus, given an image $\boldsymbol{x}$ of size $2^L \times 2^L$, we take the L level 2-D discrete Haar wavelet transform, yielding the quad-tree structure. Noting that at level j, the corresponding $\boldsymbol{d}^j$ is of size $2^{L-j} \times 2^{L-j}$, we truncate or prune the tree to remove the $L-J-1$ finer scale coefficients (levels $L-1, L-2, \cdots, J+1$), and use this pruned tree to train our model. The actual training and likelihood computation is exactly the same as in HMTseg. This gives us an HMT model with far fewer parameters than the original model. Effectively, we are training the model using a coarse representation of the original image. This is the basis of the "truncated" HMT model approach.

In the segmentation process, the main modification we need to make is in the interscale fusion process. Note that we need to use very coarse scale raw classification results when running the context model. Also, we do not get classification at the pixel level, but only upto dyadic squares of the size of $\boldsymbol{d}^{J+1}$. In practice, this level of resolution is sufficient for SAR segmentation. However, there may be situations where the pixel intensity information has added relevance. In this case, we combine the information from the pixel intensity segmentation (obtained by fitting Gaussian mixtures to pixel intensity values, and performing an ML segmentation[2]), and the final segmentation at scale $J+1$ to obtain a more refined segmentation. In this work, we adopt a very simple combination strategy. Starting with the pruned HMT based final segmentation, we modify it depending on the pixel intensity based segmentation as follows. For a block of size of, say $\boldsymbol{d}^j$, in the pixel intensity based segmented image, if more than a fixed percentage of the values have been labeled of one class, then we assign that class label to the corresponding block in the final pruned HMT segmentation; else we keep the original HMT based segmented label. This is a very simple process, but works well for the SAR images we tested.

5. SIMULATION RESULTS

In this section, we present simulation results on MSTAR clutter data using the truncated HMTseg. We train the HMT models using two different textures - forest and field imagery from the MSTAR clutter data set.[16] Training was done on homogeneous blocks of forest and field of size 256 by 256. This corresponds to 8 levels in the 2-D discrete wavelet transform tree. We prune the tree to take only the top 4 levels (coarse scales) which implies using a minimum block size of 32 by 32. The model was then trained using only these 4 coarse scale information. We tested the "truncated" HMT models on the SAR image HB06170 from the same data set (see Fig. 4). The image contains both forest and field regions, and was not used in the training process. The raw and final segmentation results, at different resolution levels, are presented in Figs. 5 and 6 respectively. Except for the road in the field, which has been classified, for the most part, as forest, the classification is sufficiently accurate. If we train our algorithm to obtain a model for road texture, then we can correct for this error; unfortunately, we were hampered by the absence of sufficient training data for road texture class. In Fig. 7, we present the combined segmentation results using the final HMT based segmentation (at scale 16×16) and the pixel intensity based segmentation. In this case, we set a hard threshold of 90% to get the combined segmentation result. Observe that we obtain a reliable and clean final segmentation using the combined approach. We tested the "truncated" HMTseg algorithm on several different SAR data sets with similar results.

6. CONCLUSIONS

We have presented an intuitive, albeit simple, modification to the HMTseg algorithm to adapt it to SAR imagery. Our modification possesses all the attractive multiscale properties of the original HMTseg algorithm, is relatively simple to implement, and reliable, since it is based entirely on coarse scale data. Moreover, it can be applied directly in the compressed wavelet domain, which is of considerable importance when dealing with large SAR images.

ACKNOWLEDGMENTS

We would like to thank Paul Haley, Andrew Miklich, Bob Mitchell, and Mike Hoffelder from Northrop Grumman Corp., Pittsburgh, PA, for many useful discussions, suggestions, and help with understanding the SAR imaging process in general, and the specific problems related to SAR ATR.

REFERENCES

1. R. Haralick and L. Shapiro, "Image segmentation techniques," *Comput. Vision Graphics Image Processing* **29**, pp. 100–132, 1985.
2. H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden markov models," *submitted to IEEE Trans. on Image Processing* , Oct. 1999. `Preprint available at www.dsp.rice.edu/publications`.
3. M. Vetterli and J. Kovačević, *Wavlets and Subband Coding*, Prentice Hall, Englewood Cliffs:NJ, 1995.
4. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Proc.* **41**, pp. 3445–3462, Dec. 1993.
5. 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.
6. S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analysis and Machine Intelligence* **14**, pp. 710–732, July 1992.
7. S. Mallat and W. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. on Info. Theory* **38**(2), pp. 617–643, 1992.
8. M. T. Orchard and K. Ramchandran, "An investigation of wavelet-based image coding using an entropy-constrained quantization framework," in *Data Compression Conference '94*, pp. 341–350, (Snowbird, Utah), 1994.
9. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, 1998.
10. J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden Markov models," in *Proceedings of SPIE technical conference on Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, vol. 3816, pp. 31–44, (Denver, CO), July 1999. Extended version available at `www.dsp.rice.edu/publications`.
11. L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE* **77**, pp. 257–285, Feb. 1989.
12. B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*, MIT Press, Cambridge, MA, 1998.
13. H. Choi and R. G. Baraniuk, "Image segmentation using wavelet-domain classification," in *Proceedings of SPIE technical conference on Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, vol. 3816, pp. 306–320, (Denver, CO), July 1999.
14. 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.
15. C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Proc.* **3**, pp. 162–177, March 1994.
16. *MSTAR (Public) Clutter DBA7B02AA 2-CD set* , September 1995. Further information available at `www.mbvlab.wpafb.af.mil/public/MBVDATA`.

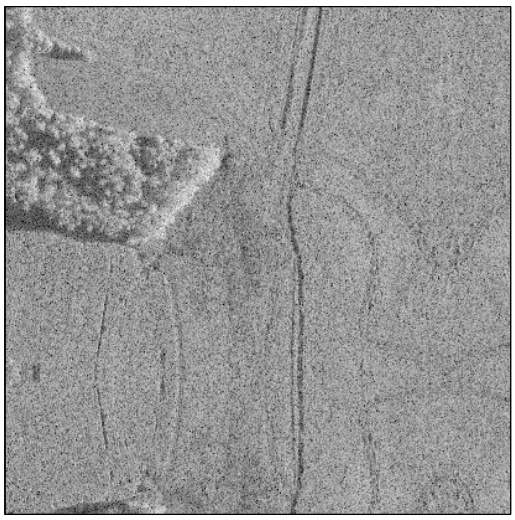

Figure 4. *Original SAR image – MSTAR clutter data HB06170 consisting of forest and field regions.*

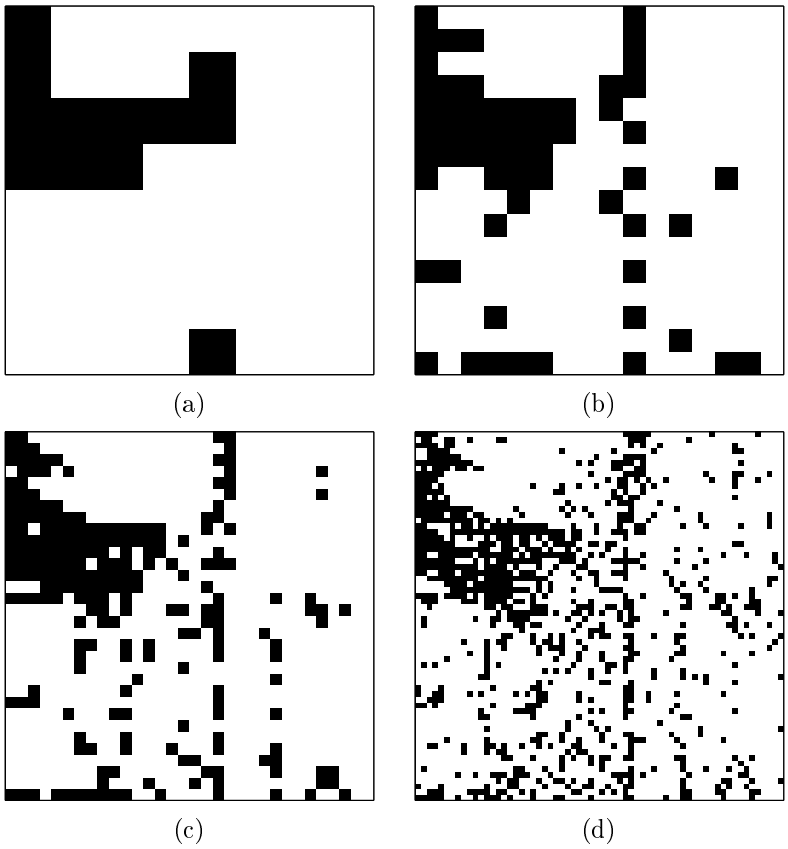

Figure 5. *Raw multiscale HMT segmentation using the 4 coarsest scales of the 2-D discrete Haar wavelet transform (white–field, black–forest): (a) Scale* 256×256 *(b) Scale* 128×128 *(c) Scale* 64×64 *(d) Scale* 32×32.

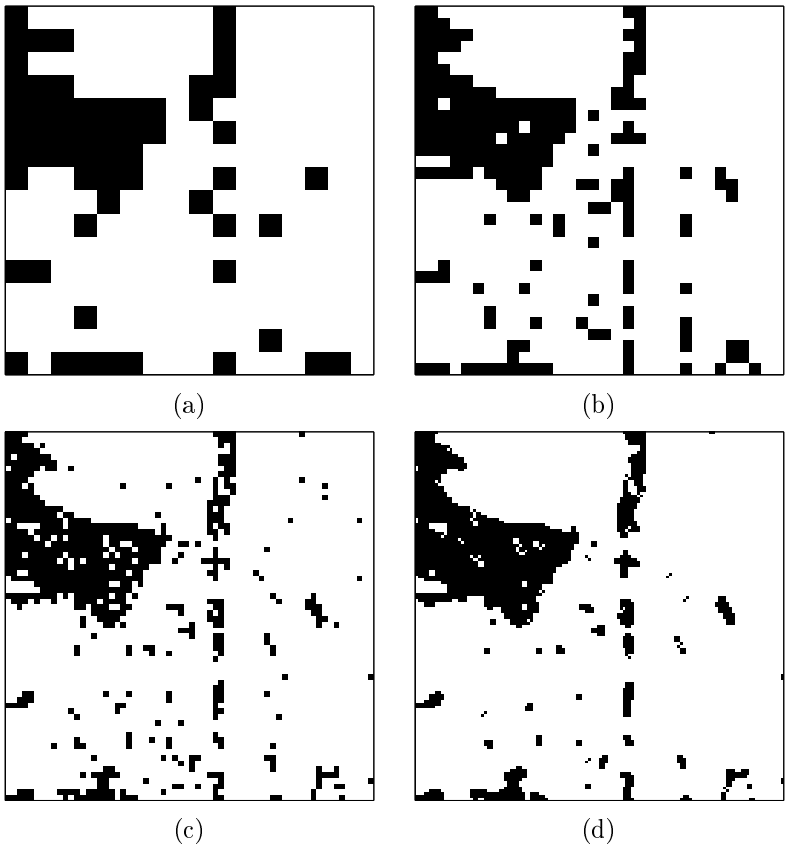

Figure 6. *Final context-based multiscale HMT segmentation using the 4 coarsest scales of the 2-D discrete Haar wavelet transform (white–field, black–forest): (a) Scale* 128×128 *(b) Scale* 64×64 *(c) Scale* 32×32 *(d) Scale* 16×16.

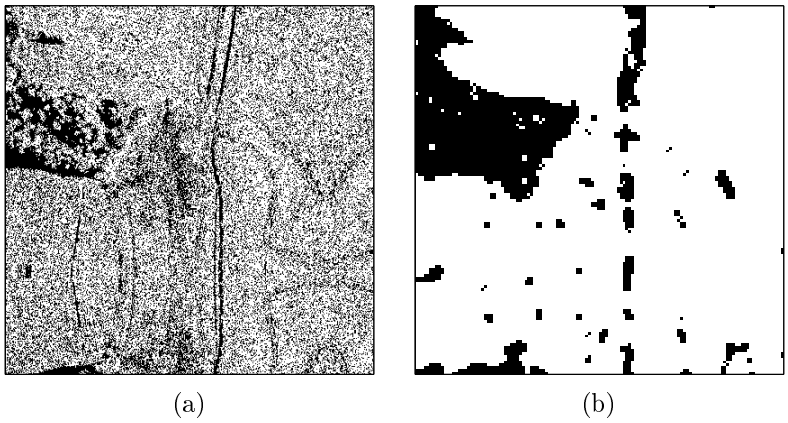

Figure 7. *Combined final segmentation using the context-based multiscale HMT segmentation and the pixel intensity based segmentation at scale* 16×16 *(white–field, black–forest): (a) Pixel intensity-based segmentation (b) Final combined segmentation.*