

Boundary refinements for wavelet-domain multiscale texture segmentation

Etai Mor, Mayer Aladjem*

Department of Electrical and Computer Engineering Ben-Gurion University of the Negev P.O. Box 653, Beer-Sheva 84105, Israel

Received 14 June 2003; received in revised form 14 August 2004; accepted 26 July 2005

Abstract

We propose a method based on the Hidden Markov Tree (HMT) model for multiscale image segmentation in the wavelet domain. We use the inherent tree structure of the model to segment the image at a range of different scales. We then merge these different scale segmented images using boundary refinement conditions. The final segmented image utilizes the reliability of coarse scale segmented images and the fineness of finer scales segmented images. We demonstrate the performance of the algorithm on synthetic data and aerial photos.

© 2005 Elsevier Ltd All rights reserved.

Keywords: Hidden Markov models; Wavelets; Boundary refinements; Texture segmentation

1. Introduction

The task of image segmentation is to separate a given image into different regions each with homogenous properties. In a texture segmentation algorithm, we assign a class label (identifying the texture) to each pixel based on its properties and its relationship with its neighborhood [13]. Recently a wavelet-domain Hidden Markov Tree (HMT) model was proposed which is well suited to texture images (edge and ridge features) [8]. The HMT characterizes the joint statistics of the wavelet coefficients and their neighborhood relationship. Because the HMT is based on the wavelet transform, its parameters are naturally arranged into the form of a quad tree [14]. This structure presents an efficient way to calculate the likelihoods of a given image, to different classes. The HMT likelihood is a robust and reliable property for classifying large homogenous image blocks. It is not robust for small blocks because it does not capture enough neighborhood information in order to assign the correct label. Recently, researchers proposed multiscale

Bayesian techniques, which apply contextual behavior in the coarser scale to guide decisions on a finer scale, e.g. [1,2]. They adopt the sequential maximum a posteriori estimator to assign each wavelet coefficient a class label. Each of these methods defines a ‘context’ which is a reference to surrounding information. Then a multiscale context model is trained in order to estimate each pixel class label. Such models were developed in [3,4] to characterize multiscale contextual labeling with off-line context model training. In [6,7] an online training of the context model was proposed. In [10,11] contextual behavior was accumulated across scales and via multiple context models.

In this work, we suggest a method for image segmentation, which overcomes the training of the multiscale context model. We rely on the fact that in homogenous regions, coarse segmentations are reliable and sufficient fine. We therefore do not need to use finer segmentation at these homogenous blocks. On the other hand near boundaries, coarse scale segmentations are not adequate enough and we need to use finer scales in order to refine our coarse scale segmented image.

We have developed a method that uses the likelihoods of different scales in order to estimate the segmented image at pixel resolution level. We compare our method to the algorithm named HMTseg [6], which also relies on the HMT model. Results show that our method outperforms the HMTseg algorithm. We also apply the algorithm to the

* Corresponding author. Fax: +972 8 6472949.

E-mail addresses: mor_etai@yahoo.com (E. Mor), aladjem@ee.bgu.ac.il (M. Aladjem).

segmentation of remotely sensed images. Excellent performance results suggest that the algorithm can be applied to various image types, including radar/sonar images and medical image, where fast and accurate segmentations are needed.

2. The wavelet transform

The wavelet transform is a multiresolution technique, which is intended to transform signals (1-D or 2-D) into a representation in which both spatial and frequency information is present [14]. There are several different implementations of the transform. We use the pyramidal multiscale construction for discrete images [20]. We will also concentrate on the simple Haar wavelet transform, which is appropriate for our purpose [7]. The Haar wavelet transform is based on the following filters (named Haar filters[9]):

1. The local smoother: $h_{LL} = (1/2) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$
2. Horizontal edge detector: $g_{LH} = (1/2) \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$
3. Vertical edge detector: $g_{HL} = (1/2) \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$
4. Diagonal edge detector: $g_{HH} = (1/2) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

To obtain the Haar wavelet coefficients of a given image u_j of size $N \times N$ ($J = \log_2 N$), we convolve the image with the four Haar filters and discard every other sample in both horizontal and vertical directions. This results in four coefficient matrixes of size $(N/2) \times (N/2)$ for each sub-band. The w_{j-1}^{LH} , w_{j-1}^{HL} , w_{j-1}^{HH} coefficients matrixes (which are outputs of the g_{LH} , g_{HL} , g_{HH} filters, respectively) are the finest scale wavelet coefficients in each sub-band (LH, HL, HH). The coefficient matrix u_{j-1}^{LL} , which is the output of the h_{LL} filter is called the scaling matrix and is used to obtain next scale wavelet coefficients. We continue recursively by applying the same procedure (convolving and down-sampling) to the resulting scaling matrix u_{j-1}^{LL} . Each iteration (scale) results in four new coarser (lower resolution) views of the image. This procedure imposes a maximal level of $J = \log_2 N$ decompositions, which leads naturally to a quad tree structure in each sub-band [18].

Fig. 1a demonstrates a three-scale wavelet transform implementation. Each sub-band is painted in a different color. At each scale, we calculate three new wavelet coefficient matrixes for the sub-bands LH, HL and HH, and a new scaling matrix for the sub-band LL. Fig. 1b illustrates the resulting quad tree structure of the wavelet coefficients in each sub-band [6]. The coefficients in coarse scales have four child coefficients in the next finer scale. The arrows point from father coefficients to their four children (from coarse scale coefficients to next finer scale coefficients).

Each wavelet coefficient analyses the same image region as its four children coefficients. These image regions will be referred to as dyadic blocks d_i^j , where i is an abstract index enumerating the dyadic blocks at scale j [6]. Given an initial image u_j of size $2^j \times 2^j$, the dyadic squares are obtained by recursively dividing the image into four square sub-images of equal size. At the two extremes d_0^0 (root of the tree) is the entire image u_j and each d_i^j (leaf of the tree) is an individual pixel. In Section 3.2, we will use this quad tree structure in order to model the wavelet coefficients into a hidden Markov tree model.

3. Wavelet-domain statistical image models

Following [8], we regard the texture as a random realization from a family or distribution of images. We present two statistical models that operate in the wavelet-domain of the image. The first is the Independent Mixture Model (IMM) [8], which is a simple model that assumes that the wavelet coefficients are independent. We then extend the IMM model to the Hidden Markov Tree (HMT) model in order to capture the key dependencies between wavelet coefficients [5]. In Section 4, we utilize the HMT model in order to segment the image.

3.1. The independent mixture model (IMM)

The IMM of wavelet coefficient was first introduced in [5,8]. The model utilizes the fact that the wavelet transform implies almost uncorrelated wavelet coefficients. If we ignore the dependencies between adjacent wavelet coefficients, we obtain the joint probability density function (pdf) of the wavelet coefficients W

$$f(W) = \prod_i f(w_i) \quad (1)$$

where w_i is a single wavelet coefficient and $f(w_i)$ is the univariate pdf. In this case, we need to model each coefficient density $f(w_i)$ independently.

The compression property of the wavelet transform [8] states that the transform of typical images consists of a small number of large coefficients and a large number of small coefficients. This property combined with looking at an image as a realization drawn from a probability distribution leads to the following model. We model each coefficient w_i as being in one of two states (hidden states): ‘high’—corresponds to a wavelet component containing significant contribution of image energy, or ‘low’—representing coefficients with little energy. We associate each state with a Gaussian pdf. We set a zero mean high variance pdf for the high state coefficients and a zero mean low-variance pdf for the low state coefficients. Finally, we define a two-state Gaussian mixture model for each

wavelet coefficient

$$f(w_i) = p_s(0)g(w_i|s = 0) + p_s(1)g(w_i|s = 1), \quad (2)$$

where

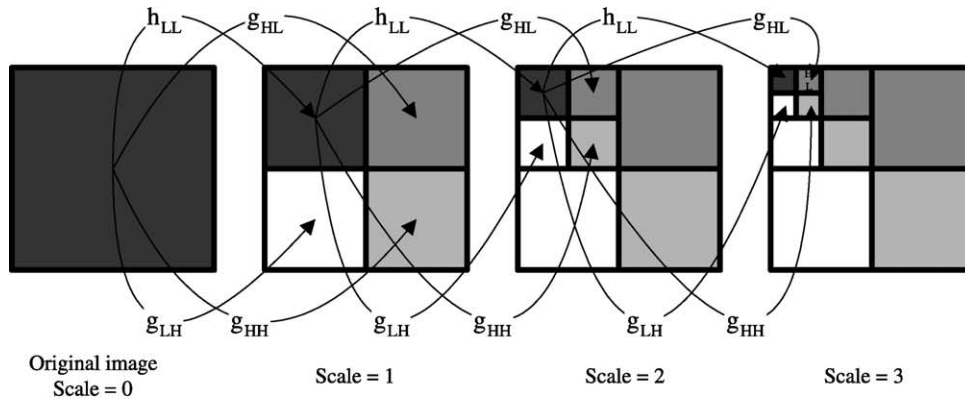
$$g(w_i|s = m) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left\{\frac{-w_i^2}{2\sigma_m^2}\right\} \text{ for } m = 0, 1. \quad (3)$$

The hidden state variable denoted by s can be in one of two states, $s=0$, representing low variance coefficients or $s=1$, representing high variance coefficients. The model is

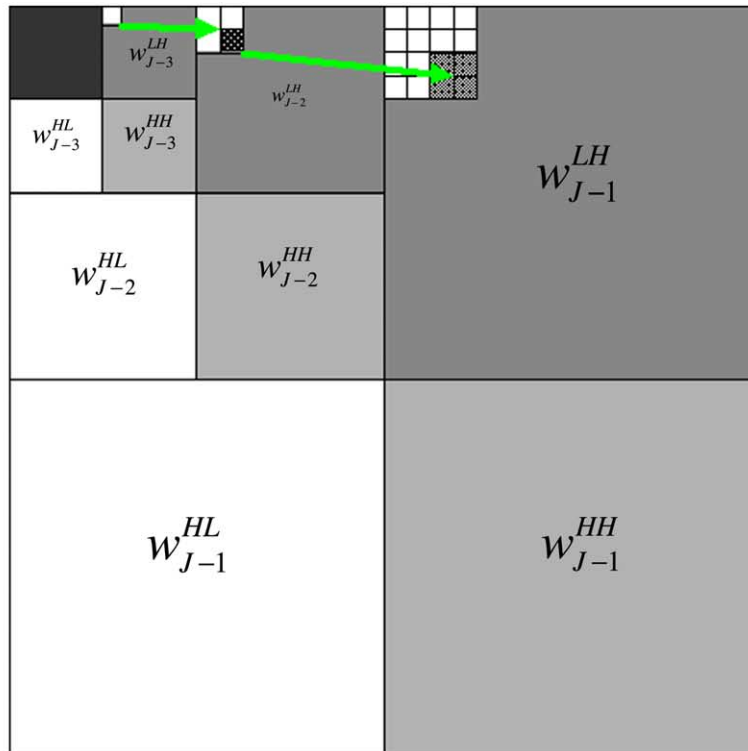
completely parameterized by the prior probabilities $p_s(m)$ of s and the variances σ_m^2 for $m=0, 1$. The parameters $p_s(m)$ and σ_m^2 can be estimated using a small amount of training data [8].

3.2. The hidden Markov tree model (HMT)

The HMT [8] extends the IMM model by also modeling the relationships between wavelet coefficients. The HMT models the key dependencies between wavelet coefficients by utilizing two properties of the wavelet transform.



(a) Wavelet transform implementation.



(b) Quad tree structure of wavelet coefficients

Fig. 1. The iterative procedure for constructing the Haar wavelet coefficients. (a) The scale wavelet coefficients are produced using the four Haar filters and the previous scale scaling coefficient matrix. (b) The resulting quad tree structure; each wavelet coefficient has four child coefficients in the next finer scale.

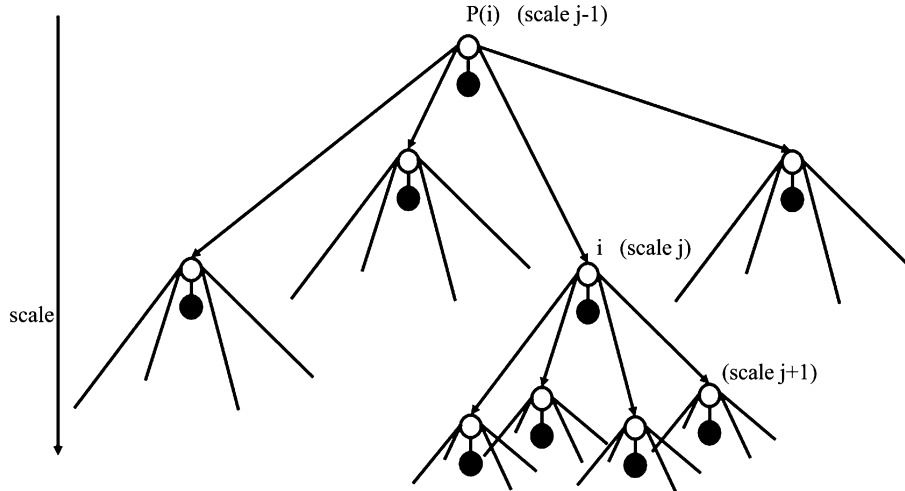


Fig. 2. A 2-D wavelet hidden Markov tree model for a sub-band. Each wavelet coefficient (black node) is modeled as a Gaussian mixture, controlled by a hidden state (white node). The arrows are from parent coefficients hidden states to their four children hidden states.

The first property is clustering, which states that if a particular wavelet coefficient is large/small, then adjacent coefficients are very likely to also be large/small [17]. The second property is persistence across scale, which states that large/small values of wavelet coefficients tend to stay large/small across scales [15,16]. These dependencies can be described by a probabilistic graph [12] shown in Fig. 2. Each black node represents a single wavelet coefficient. Each white node represents a hidden state variable associated with the wavelet coefficient. The relationship between two wavelet coefficients is described using a connection between two wavelet’s hidden states. The connection is between the parent coefficients $P(i)$ in scale $j-1$ to its four child coefficients i in the next finer scale j . This type of relationship results a quad tree structure for each sub-band b (see Fig. 1b). We define T_i^b as the set of wavelet coefficient in a given sub-band and dyadic block d_i^j . The wavelets coefficients T_i^b are arranged in a sub-tree structure, where the coefficient w_i^j is the root of the tree. Each state coefficient hidden state variable is denoted by s_i^b .

Thus the HMT for each sub-band b is parameterized by:

$$\theta_{\text{HMT}}^b = \{p_0^b(m), \varepsilon_{j,j-1}^b(m, n), \sigma_{b,j,m}^2 | b \in \{\text{HL}, \text{LH}, \text{HH}\};$$

$$j = 0, \dots, J-1; \quad m, n = 0, 1\} \quad (4)$$

Here $p_0^b(m)$ is the prior probability of the root coefficient state variable s_o^b , $\varepsilon_{j,j-1}^b(m, n)$ is the transition probability of the Markov chain from scale $j-1$ to scale j in the sub-band b (conditional probability of variable s_i^b being in state m given its parent $s_{P(i)}^b$ being in state n), and $\sigma_{b,j,m}^2$ is the variance of the Gaussian component, corresponding to each state m , scale j and sub-band b .

3.2.1. Model training

Following [8], we compute the HMT likelihood for each sub-band in a recursive fine to coarse fashion. First, we

calculate the conditional likelihood $\beta_i^b(m)$ of the sub tree T_i^b to the HMT model θ_{HMT}^b given that its hidden variable is in state m

$$\beta_i^b(m) = f(T_i^b | s_i^b = m, \theta_{\text{HMT}}^b) \quad (5)$$

For the finest scale $\beta_i^b(m) = g(w_i^b | s_i^b = m)$ (see Eq. (3)). Then, we calculate the conditional likelihood $\beta_{i,P(i)}^b(m)$ of the sub-tree T_i^b to θ_{HMT}^b , given that its parent is in state m

$$\beta_{i,P(i)}^b(m) = f(T_i^b | s_{P(i)}^b = m, \theta_{\text{HMT}}^b)$$

$$= \sum_{n=0,1} \varepsilon_{i,P(i)}^b(n, m) \beta_i^b(n) \quad (6)$$

For the next coarser level, we calculate the conditional likelihood

$$\beta_{P(i)}^b(m) = f(w_{P(i)}^b | m, \theta_{\text{HMT}}^b) \prod_{j \in C(P(i))} \beta_{j,P(i)}^b(m), \quad (7)$$

where $C(P(i))$ means the four children of $P(i)$. We iterate the calculation of Eqs. (6) and (7) until we reach the root of the tree.

The likelihood of the sub-tree T_i^b to a specific model θ_{HMT}^b (4) is

$$f(T_i^b | \theta_{\text{HMT}}^b) = \sum_{m=0,1} \beta_i^b(m) p(s_i = m) \quad (8)$$

Using the assumption that the trees $T_i^{\text{HL}}, T_i^{\text{LH}}, T_i^{\text{HH}}$ of the sub-bands HL, LH and HH are independent [7], we have the likelihood of the wavelet coefficients by

$$f(W | \theta_{\text{HMT}}) = \prod_{b=\text{HL,LH,HH}} f(T_0^b | \theta_{\text{HMT}}^b) \quad (9)$$

Here W denotes all coefficients and T_0^b represents all coefficients for $b=\text{HL}, \text{LH}$ and HH . The maximum

likelihood estimation of θ_{HMT} is

$$\hat{\theta}_{\text{HMT}} = \underset{\theta_{\text{HMT}}}{\operatorname{argmax}} \prod_{k=1}^K f(W_k | \theta_{\text{HMT}}), \quad (10)$$

were $W_1 \dots W_K$ are the wavelet coefficients computed for K training images. The computation of $\hat{\theta}_{\text{HMT}}$ can be done efficiently by the tree-structured EM algorithm [8].

4. Multiscale segmentation using HMT model

Here, we employ the HMT model (Section 3.2) for image segmentation. In Section 4.1, we describe a multiscale raw classification algorithm, based on the HMT likelihood [6,7]. It classifies each dyadic block using a set of independent ‘raw’ segmented images down to resolution of 2×2 blocks. In Section 4.2, we explain the classification algorithm for pixel resolution level which results in ‘raw’ segmentations at coarse and finer scales [6,7]. The coarse scales are very reliable, because of the large amount of wavelet coefficients, and the relationships between them. On the other hand, finer scales are much more finely localized, but suffer from poor classification because of a smaller amount of data. In Section 4.3, we develop a method called boundary refinement segmentation. The idea is to combine the fine and coarse ‘raw’ image segmentations in order to obtain a robust and also finely localized result.

4.1. Multiscale raw segmentation [6,7]

In order to segment the image, we first acquire training data representative of each texture data and using (10), we estimate θ_{HMT}^c for each texture class $c \in \{1, \dots, M\}$.

Using the Haar wavelet transform, there is an obvious correspondence between the wavelet coefficients and dyadic squares of the image. Each dyadic block of pixels d_i^j corresponds to the three trees of wavelet coefficients: T_i^{LH} , T_i^{HL} , T_i^{HH} . Using the sub-band independence assumption, we can calculate the likelihood of each dyadic block d_i^j to a specific model θ_{HMT}^c by

$$f(d_i^j | \theta_{\text{HMT}}^c) = f(T_i^{\text{LH}} | \theta_{\text{HMT}}^{\text{LH},c}) f(T_i^{\text{HL}} | \theta_{\text{HMT}}^{\text{HL},c}) f(T_i^{\text{HH}} | \theta_{\text{HMT}}^{\text{HH},c}) \quad (11)$$

where the likelihoods of each sub-tree is computed by (8). Each dyadic block is classified using the maximum likelihood criteria

$$c_i^j = \underset{c}{\operatorname{argmax}} f(d_i^j | \theta_{\text{HMT}}^c) \quad (12)$$

The latter classification implies a set of segmented images each at a different scale. We refer to these segmentations as ‘raw’ segmentations because they do not exploit any possible relationships between the different scales. On coarse scales each dyadic block contains more wavelet coefficients than on finer scales, resulting a more robust HMT model, which captures a lot of the wavelet

coefficients relationships. But as we move to finer scales each block becomes smaller, resulting a more finely localized segmentation, but less robust.

4.2. Pixel-level segmentation [6,7]

Because the HMT model characterizes the joint statistics of dyadic image squares, only down to 2×2 blocks, we do not directly obtain the pixel level segmentation. This is because we ignored the scaling coefficients, which characterize pixel brightness. In order to obtain pixel-level segmentation we need a model for the pixel brightness of each texture class. We use a Gaussian mixture model for the pixel values of each trained texture. We then obtain the likelihood of each pixel and compute the raw segmented images down to pixel level¹.

4.3. Segmentation using boundary refinements

Sections 4.1 and 4.2 introduced the raw segmentation algorithm, which suffers from low resolution at coarse scales and from instability at finer scales. In order to obtain high-quality segmentations we propose a method that combines results from different scales. Our method differs from the methods in [6,7,10,11] by the fact it does not train a context model. Instead it uses a weighted average between the results of different scales using a boundary probability function. The boundary probability function is computed online using an iterative method. Performance results also show significant improvement over the method in [6,7].

The main problem of coarse scale segmentations is on the boundaries between different textures, where a more finely localized segmentation is needed. On the other hand on smooth regions (not near boundaries), coarse segmentations are sufficiently fine and are also more robust. We propose to merge coarse and fine scale segmentations into a single image, in order to improve segmentation results. We start at a coarse enough scale L such that the raw segmentations are statistically reliable and move down to finer scales. At each scale, we calculate an averaged segmented image, which is a refinement of the previous averaged segmented image. Each averaged segmented image is obtained using the following algorithm:

4.3.1. Step A—calculation of the class prior and raw posterior probabilities

We estimate the texture class prior probability given the previous scale segmentations

$$p(c_i^j) = \frac{N(c_i^j)}{9}. \quad (13)$$

¹ In many real world images pixel brightness varies considerably due to shading. For such images the 2×2 block segmentations will be far more robust, since they rely just on inter scale pixel dependencies.

where c_i^j is the class label of dyadic block d_i^j , and $N(c_i^j)$ is the number of dyadic blocks that were classified as class c_i^j in the previous stage's segmented image (i 's parent and its eight neighbors).

Then, we calculate the raw posteriors for the M classes using Bayes rule

$$p(c_i^j | d_i^j) = \frac{f(d_i^j | c_i^j) p(c_i^j)}{\sum_{c_i^j=1}^M f(d_i^j | c_i^j) p(c_i^j)} \quad (14)$$

with $f(d_i^j | c_i^j)$ obtained by (11).

4.3.2. Step B—merging posteriori probabilities

The dyadic block d_i^j becomes smaller as we move to finer scales resulting in less accurate estimations of the posteriors in (14). In order to overcome this fine scale inaccuracy we merge posteriors calculated from (14) so that near boundaries we will prefer the next finer raw segmented image and on smooth regions, we will prefer the previous stage segmented image. For this purpose, we introduce the boundary variable b_i^j , which states if a dyadic square d_i^j is composed of more than one class label.

$$b_i^j = \begin{cases} 0, & \text{if } \{c_{ch(i)}^{j+1}\} = c_i^j \\ 1 & \text{otherwise} \end{cases} \quad (15)$$

where $\{c_{ch(i)}^{j+1}\}$ are the four children of the class label c_i^j . We expect that in homogenous regions most of the class labels c_i^j will be classified as their children $c_{ch(i)}^{j+1}$, and therefore, the probability of b_i^j being equal 0 ($p(b_i^j = 0)$) will be high. On the contrary, near boundaries we expect $p(b_i^j = 1)$ to have high values.

Given $b_{P(i)}^{j-1}$, we can calculate the conditional posteriori

$$p(c_i^j | d_{A(i)}^L, b_{P(i)}^{j-1}) = \begin{cases} p(c_{P(i)}^{j-1} | d_{A(P(i))}^L) & \text{where } b_{P(i)}^{j-1} = 0 \text{ (smooth region)} \\ p(c_i^j | d_i^j) & \text{where } b_{P(i)}^{j-1} = 1 \text{ (boundary region)} \end{cases}, \quad (16)$$

where $A(i)$ denotes the ancestor of coefficient i at scale L and $p(c_{P(i)}^{j-1} | d_{A(P(i))}^L)$ is the averaged posteriori calculated in the previous scale by:

$$p(c_i^j | d_{A(i)}^L) = \sum_{m=0}^1 p(c_i^j | d_{A(i)}^L, b_{P(i)}^{j-1} = m) p(b_{P(i)}^{j-1} = m) \quad (17)$$

The conditional posteriori $p(c_i^j | d_{A(i)}^L, b_{P(i)}^{j-1})$ equals the current scale raw posterior if $b_{P(i)}^{j-1} = 1$. By this means, we incorporate finer scale raw posteriors in boundary regions. On the other hand in homogenous regions ($b_{P(i)}^{j-1} = 0$), we use the averaged posteriori $p(c_{P(i)}^{j-1} | d_{A(P(i))}^L)$ calculated in the

previous scale because it is sufficiently fine in homogenous regions.

We name the computations (16,17) merging the posteriori probabilities. We use the dyadic block d_i^j and all its ancestors; therefore it is conditioned on $d_{A(i)}^L$ which contains all the data in $\{d_i^j, d_{P(i)}^{j-1}, d_{P(P(i))}^{j-2}, \dots, d_{A(i)}^L\}$.

4.3.3. Step C—calculation of the boundary probability

Equations (16) and (17) provide an efficient way for merging the posteriori probabilities from different scales. However, Eq. (17) is based on the boundary probability $p(b_{P(i)}^{j-1} = m)$, $m=0,1$ which is usually unknown in advance. In order to estimate the boundary probability we present an iterative procedure which starts by guessing an estimate for $p(b_{P(i)}^{j-1} = m)$, $m=0,1$ and then calculating (16,17) using this estimate. We then use $p(c_i^j | d_{A(i)}^L)$ calculated in (17) to obtain a new estimate for $p(b_{P(i)}^{j-1} = m)$, $m=0,1$. We continue this iterative procedure until the boundary probabilities have converged.

In order to calculate the boundary probabilities, we use the definition of b_i^j provided in Eq. (15). From (15), it is clear that $p(b_{P(i)}^{j-1} = 0)$ is the probability of all children of $c_{P(i)}^{j-1}$ being equal to their parents' class.

Using Eq. (17) and assuming the four class labels $\{c_{ch(P(i))}^j\}$ are independent given their ancestor dyadic block $d_{A(q)}^L$, we obtain the probability of $\{c_{ch(P(i))}^j\}$ being equal to class label C by:

$$p(\{c_{ch(P(i))}^j\} = C) = \prod_q p(c_q^j = C | d_{A(q)}^L) \quad (18)$$

where q is an abstract index iterating the 4 children of $P(i)$.

By substituting (16), (17) into (18) and summing over all values of C we obtain

$$p(b_{P(q)}^{j-1} = 0) - \sum_{C=1}^M \prod_q \left\{ p(c_{P(q)}^{j-1} = C | d_{A(P(q))}^L) p(b_{P(q)}^{j-1} = 0) + p(c_q^j = C | d_q^j) p(b_{P(q)}^{j-1} = 1) \right\} = 0 \quad (19)$$

Using the fact that $p(b_{P(q)}^{j-1} = 1) = 1 - p(b_{P(q)}^{j-1} = 0)$, Eq. (19) becomes a function of a single variable $p(b_{P(q)}^{j-1} = 0)$. Equation (19) can be solved using any standard root finding method. We used the Secant method [21] which converges quickly.

4.3.4. Step D—segmentation

We segment the image using the averaged posteriori $p(c_i^j | d_{A(i)}^L)$

$$c_i^j = \underset{c_i^j}{\operatorname{argmax}} p(c_i^j | d_{A(i)}^L), \quad c_i^j = 1 \dots M. \quad (20)$$

This segmentation is much more reliable than the raw segmentation (12) because it is based on $d_{A(i)}^L$ that contains all other finer scale dyadic blocks. The raw segmentations (12) are based just on d_i^j which became smaller as we moved to finer scales, resulting in unreliable segmentations.

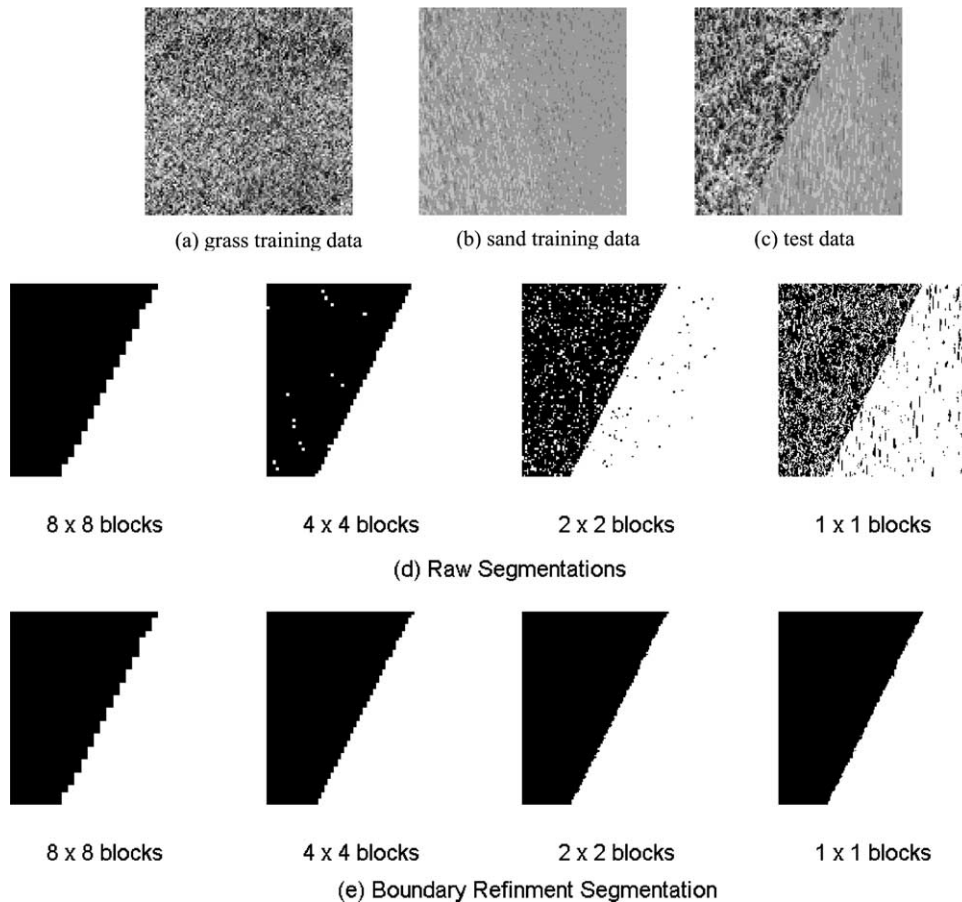


Fig. 3. Boundary refinement example. (a) 512×512 grass texture image. (b) 512×512 sand texture image. (c) A 256×256 grass/sand mosaic test image. (d) Raw HMT-based multiscale segmentations of the test image. Segmentation of size 8×8 , 4×4 , 2×2 and pixel size are presented. (e) Our method of segmentation.

We iterate steps A–D, until we reach the finest scale $j=J$. At this point, the segmented image is a merging of all raw segmented images, which achieves the stability of coarse segmented images on homogenous regions and the finer segmentations of finer scales near texture boundaries.

Fig. 3 demonstrates the boundary refinement process. We trained the HMT model for grass and sand textures [19] shown in Fig. 3a and b, respectively. The test image is shown in Fig. 3c. Fig. 3d shows the resulting raw segmentations [6,7]. We can see that at coarse scales, the segmentation image is robust, but not finely localized on the boundary between the grass and sand textures. As we move to finer scales, the boundary becomes more finely localized but the segmented image has more misclassifications. Fig. 3e shows our boundary refinement average segmented images. At each scale, a finer segmentation is achieved using the previous stage averaged posteriori. Each segmented image preserves the robustness of the previous scale segmented image while refining the boundary between the sand and grass textures. The final segmentation is robust and also finely localized.

5. Experiments

Here, we present the results of our comparisons using simulations and real data applications. We first compare our method to the HMTseg method proposed in [6,7] (Section 5.1) using a simulation of texture mosaics set in [12]. Then, we show the performance results on an aerial photo image (Section 5.2) and compare it to HMTseg results.

5.1. Simulation results

We set the starting coarse scale L to 4 and ran our method (Section 4.3) and the HMTseg method [6,7] on the texture mosaic example proposed in [12]. This mosaic is composed of nine different textures (Fig. 4a) having ground truth segmentation shown in Fig. 4b. We trained the HMT model using 256×256 textures taken randomly from [19]. Fig. 4c and d shows the segmentations performed by the HMTseg and our method. Visually inspecting the images shows that our method outperforms the HMTseg method. We also measure segmentation performance in terms of classification rates P_a , P_b and P_c [11], where, P_a is the percentage of pixels which are correctly classified, showing accuracy,

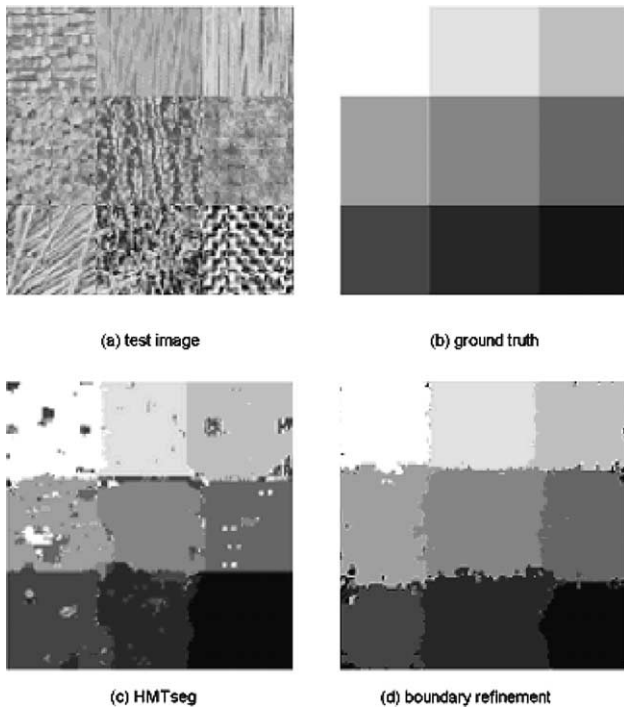


Fig. 4. (a) Test image (b) Ground truth segmentation (c) Segmentation using HMTseg method (d) Our segmentation method.

Pb the percentage of boundaries that coincide with true ones, showing specificity, and Pc the percentage of true boundaries that can be detected, showing sensitivity. Table 1 summarizes the performances of HMTseg and our boundary refinement method. We see that our method improves the segmentation results over the HMTseg in terms of Pa, Pb and Pc (see 3rd row of Table 1).

5.2. Aerial photo segmentation

In Fig. 5, we show the segmentation result on a real aerial photo. We trained the HMTs for ‘sea’ and ‘ground’ textures using 256×256 hand-segmented blocks from the 1024×1024 aerial photo in [11]. Fig. 5a presents the test image. Fig. 5b shows the segmentation of the HMTseg method [6, 7] and Fig. 5c shows the segmentation result of our method. We can see that our method outperforms the HMTseg method yielding robust, but also finely localized segmentation.

Table 1
The performance of our ‘boundary refinement’ method and the HMTseg method [6,7]

Method	Pa (%)	Pb (%)	Pc (%)
HMTseg	88.10	11.47	51.09
Boundary Refinement	95.15	14.7	54.5
Improvement	7.05	3.23	3.41



(a) Test Image



(b) HMTseg



(c) Boundary refinement

Fig. 5. (a) 256×256 -building/water mosaic test image. (b) HMTseg segmentation result. (c) Our method.

6. Conclusions

In this paper, we proposed a multiscale image segmentation method based on the HMT model. The method accumulates statistical context behavior across scales in order to produce a robust and accurate segmentation of texture images. It attempts to refine coarse scale segmentations mainly on texture boundaries, where finer segmentations are needed. We do not train a context model as is done in [6,7,10,11]. By this, we reduce the running time of the segmentation significantly. Performance results on different kinds of texture showed excellent results compared to the HMTseg method. Promising avenues for future research include the investigation of different wavelets models [12] and extending the algorithm to the unsupervised segmentation task.

Acknowledgements

This work was supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University of the Negev, Israel.

References

- [1] C.A. Bouman, C.A. Bouman, B. Liu, Multiple resolution segmentation of textured images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (2) (1991) 99–113.
- [2] C.A. Bouman, C.A. Bouman, M. Shapiro, A multiscale random field model for Bayesian image segmentation, *IEEE Transactions on Image Processing* 3 (2) (1994) 162–177.

- [3] H. Cheng, H. Cheng, C.A. Bouman, Trainable context model for multiscale segmentation Proc. of IEEE Int'l Conf. on Image Proc., vol. 1 1998 pp. 610–614, Chicago, IL, October.
- [4] H. Cheng, H. Cheng, C.A. Bouman, Multiscale Bayesian segmentation using a trainable context model, IEEE Transactions on Image Processing 10 (1) (2001) 511–525.
- [5] H. Chipman, E. Kolaczyk, R. McCulloch, Adaptive Bayesian wavelet shrinkage, Journal American Statistical Association 440 (92) (1997) 1413–1421.
- [6] H. Choi, H. Choi, R.G. Baraniuk, Multiscale image segmentation using wavelet-domain hidden markov models, IEEE Transactions on Image Processing 10 (9) (2001) 1309–1321.
- [7] H. Choi, H. Choi, R.G. Baraniuk, Image segmentation using wavelet-domain classification Proc. of SPIE vol. 3816 (1999), p. 306. pp. 306–320, Denver, CO.
- [8] M.S. Crouse, R.D. Nowak, R.G. Baraniuk, Wavelet-based statistical signal processing using hidden Markov models, IEEE Transactions on Signal Processing 46 (4) (1998) 886–902.
- [9] I. Daubechies, I. Daubechies, Ten Lectures on Wavelets, SIAM, New York, 1992.
- [10] G. Fan, G. Fan, X.G. Xia, Multiscale texture segmentation using hybrid contextual labeling tree, Proc. IEEE Int. Conf. Image Proc., 2000 Vancouver, Canada.
- [11] G. Fan, G. Fan, X.G. Xia, A joint multi-context and multiscale approach to bayesian image segmentation, IEEE Transactions on Geosciences and Remote Sensing 39 (12) (2001) 2680–2688.
- [12] G. Fan, X.-G. Xia. On Context-Based Bayesian Image Segmentation: Joint Multi-context and Multiscale Approach and Wavelet-Domain Hidden Markov Models, in Proceedings of the 35th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 4–7, 2001.
- [13] R. Haralick, L. Shapiro, Image segmentation techniques, Computing and Visualization in Graphics, Image Processing 29 (1985) 100–132.
- [14] S. Mallat, Tour of Signal Processing, Academic, New York, 1998.
- [15] S. Mallat, S. Mallat, S. Zhong, Characterization of signals from multiscale edges, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 710–732.
- [16] S. Mallat, S. Mallat, W. Hwang, Singularity detection and processing with wavelets, IEEE Transactions on Information Theory 38 (2) (1992) 617–643.
- [17] M.T. Orchard, M.T. Orchard, K. Ramchandran, An investigation of wavelet-based image coding using an entropy-constrained quantization framework Data Compression Conference, Snowbird, Utah, 1994. pp. 341–350.
- [18] J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Transactions on Signal Processing 41 (1998) 3445–3462.
- [19] The usc-sipi image database. [Online]. Available: <http://sipi.usc.edu/services/database/Database.html>.
- [20] M. Vetterli, M. Vetterli, J. Kovacevi'c, Wavelets and subband coding, Prentice Hall, Englewood Cliffs, NJ, 1995.
- [21] F. Charles. Van Loan, Introduction to Scientific Programming: a matrix-vector approach. Department of Computer Science, Cornell University, 2000, pp. 289–293.