

# Mutiscale Texture Segmentation Using Contextual Hidden Markov Tree Models

Din-Chang Tseng, *Member, IACSIT* and Ruei-Lung Chen

**Abstract**—A multiscale texture segmentation approach based on contextual hidden Markov tree (*CHMT*) model and boundary refinement is proposed. A hidden Markov tree (*HMT*) model is a probabilistic model for capturing persistence properties of wavelet coefficients without considering clustering properties. We have proposed the *CHMT* model to enhance the clustering properties by adding extended coefficients associated with wavelet coefficients in every scale. In this study, we train the *CHMT* parameters for every texture and then use them to compute maximum likelihoods for every dyadic square region at every scale in an image which will be segmented. Then the boundary refinement algorithm is adopted to fuse the different-scale segmented results to improve the final results. We demonstrate the performance of the proposed method on synthetic and aerial images; moreover, the comparison with other methods is also provided to show the effectiveness of the proposed method.

**Index Terms**—Contextual hidden Markov tree model, maximum likelihood, multiscale texture image segmentation, wavelet transform.

## I. INTRODUCTION

Texture image segmentation is a fundamental technique to categorize pixels into classes based on homogeneous properties. Many researchers solved the texture analysis problems with various approaches which are broadly classified into structural, statistical, model-based, and transform approaches [1]-[3]. Structural methods use texture elements to describe textures. It is good for image synthesis applications. Statistical methods use gray-level relationship between neighboring pixels to describe to local texture property in first-order, second-order, or higher-order statistics. The methods are good for invariant texture analysis and classification. Model-based methods model images as different probability or linear combination models [3] and use model parameters to describe their texture features, such as autoregressive models, fractal models [4], Gaussian-mixture models (*GMM*) [5], hidden Markov models (*HMM*) [6], [7], Markov random fields (*MRF*) [8] and so on. The transform methods transfer images into a frequency domain to describe textures. The methods usually use Fourier, Gabor, or wavelet transform. Bharati *et al.* [1] and Porter *et al.* [9] compared different methods for texture segmentation and got the result that wavelet method appears to be more powerful in texture analysis because it keeps the space and frequency information simultaneously and it has least sensitivity to noise.

Crouse *et al.* [10] developed a wavelet-based *HMM* called *HMT* model for signal processing. The model describes inheritances of wavelet coefficients from coarser to finer scales in wavelet trees. It connects wavelet coefficients between scales with Markov chain and preserves the persistence properties. Choi and Baraniuk [11], [12] used the *HMT* model to propose a *HMTseg* algorithm for multiscale Bayesian image segmentation. In the Bayesian segmentation, an image  $x$  is regarded as a realization of a random field  $X$  [13]. Then the image segmentation problem is interpreted as giving an image  $x$  and estimating a region  $x_r$ , a class label  $c_r \in \{1, \dots, N_c\}$  which is one of  $N_c$  texture candidates with maximum likelihood  $f(x_r|c_r)$  [12]. Following the Choi and Baraniuk's method, Fan and Xia [14] developed a *HMT-3S* method based on *HMTseg*. Xu *et al.* [15] applied the *HMT*-based method to improve the color texture classification by taking account of the dependencies between different color channels. Mor and Aladjem [16] fused the class labels at different scales with boundary refinement. Liu *et al.* [17] used a complex wavelet domain *HMT* model to improve the problems of shift variance and lack of directionality. Owing to the analysis accuracy is not high enough for real-world applications [18], the studies of multiscale texture analysis in wavelet-based *HMM* are still progressively improved. Kim *et al.* [19] presented a wavelet-based texture segmentation using multilayer perceptron networks and *MRFs* in a multiscale Bayesian framework. Zhang *et al.* [18] implemented an energy minimization algorithm based on graph cut to fuse the multiscale classification results. Moreover, Zheng *et al.* [20] employed a fuzzy method to estimate parameters of multiresolution Markov random field to segment texture images.

The above-mentioned methods usually concentrate only on inter relationships. In this paper, we proposed the *CHMT* model to enhance the clustering properties by adding extended coefficients associated with wavelet coefficients in every scale. The *CHMT* model strengthens the clustering of wavelet coefficients and holds the persistence properties, simultaneously. So both the interscale and intrascale dependences between wavelet coefficients are considered. The *CHMT* model had worked well for image denoising [21]. Here we extend the application of the *CHMT* model for texture segmentation.

The procedure of the proposed approach is shown in Fig. 1. First, we use iterative expectation-maximization (*EM*) algorithms to train the *CHMT* parameters for all dyadic blocks and the *GMM* parameters for all pixels of images with several texture types. Then, a raw segmentation is obtained by computing the maximum likelihoods of each dyadic block of the testing image to decide the class label of every dyadic

Manuscript received October 1, 2014; revised December 29, 2014.

Din-Chang Tseng and Ruei-Lung Chen are with the Institute of Computer Science and Information Engineering, National Central University, Zhongli, 32001 Taiwan (e-mail: tsengdc@ip.csie.ncu.edu.tw, rlchen17@gmail.com).

block and pixel. There are two procedures in the labeling. In the first procedure, a multiscale segmentation uses the trained *CHMT* parameters to estimate the likelihoods of every dyadic block in the wavelet tree of the testing image. We ignore the scaling coefficients in *CHMT* model, because coarser scale provides little information for image segmentation. In the second procedure, a pixel-level segmentation uses the trained *GMM* parameters to estimate the likelihoods of each pixel in the testing image. At last, both segmentation results at various scales are combined by the boundary refinement method [16] to get a better-quality final segmentation result.

The remaining sections of this paper are organized as follows. Section II introduces the *CHMT* model. The segmentation algorithm using *CHMT* model and boundary refinement is presented in Section III. Section IV reports the experiment and other comparison results. At last, conclusions are presented in Section V.

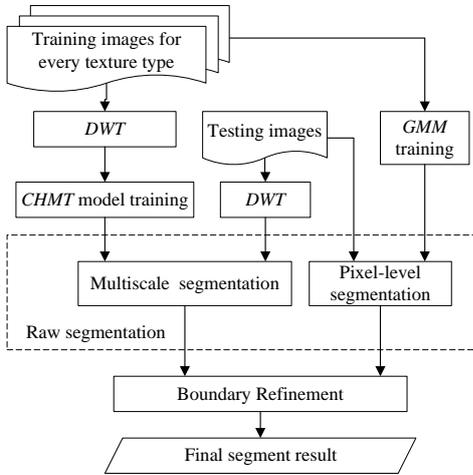


Fig. 1. The procedure of the proposed method.

## II. CONTEXTUAL HIDDEN MARKOV TREE MODELS

The 2D wavelet coefficients have a quad-tree pyramid structure like Fig. 2, in which a node  $i$  has a parent node  $p(i)$  in the last coarser scale and four child nodes  $c(i)$  in next finer scale. In the quad-tree structure, a tree with the root node at the coarsest scale can be expressed as  $T_0$ . The trees with root nodes at the other scales are sub-trees. The tree  $T_i$  is one of the sub-trees of tree  $T_0$ .

Assuming a wavelet coefficient  $w_i$  is a stochastic variable  $W_i$  and a hidden state  $s_i$  is associated with the wavelet coefficient as another stochastic variable  $S_i$ . The pair  $(w_i, s_i)$  is called a node with complete information of the wavelet tree structure. Fig. 3 shows dependencies of the nodes  $a$  to  $g$  in the *CHMT* model, in which the white dots are the coefficients and the black dots are the hidden states.

In a  $M$ -state *GMM*, the wavelet coefficient can be expressed as a probability mass function (*pmf*)

$$f_{w_i}(w_i) = \sum_{m=1}^M p_{S_i}(m) f_{w_i|S_i}(w_i | s_i = m), \quad (1)$$

where  $p_{S_i}(m)$  is the probability of state  $m$  and  $f_{w_i|S_i}(w_i | s_i = m)$  is the conditional probability function of wavelet coefficient  $w_i$  at state  $m$ . Here, the conditional probability function

$f_{w_i|S_i}(w_i | s_i = m)$  is set as a Gaussian function. Actually, the probability density of wavelet coefficients is not a Gaussian distribution according to the clustering characteristic [11]. However a *GMM* that can approach any density distribution with arbitrarily shape can model wavelet coefficients as a multi-state Gaussian distribution.

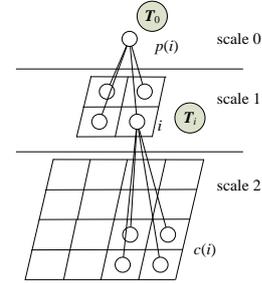


Fig. 2. The pyramid structure of a 2-D wavelet transform.

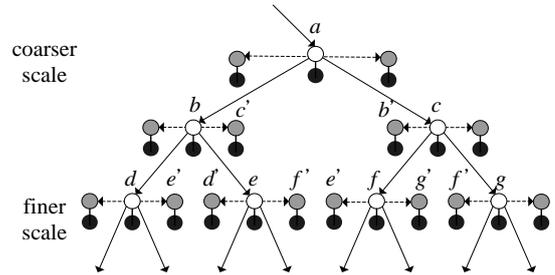


Fig. 3. Dependencies in the *CHMT* model.

The *HMT* model holds the interscale dependences between different scales in wavelet tree structure as the solid arrows in Fig. 3. To keep the clustering property with no drawbacks of the network structure, we add extended nodes which are made by copying adjacent nodes at the same scale in the *HMT* model to construct a *CHMT* model. A node connects to its extended adjacent nodes as the dashed arrows in Fig. 3 and those nodes are treated as tree leaves in every scale. The gray nodes  $b'$  to  $g'$  mean the extended nodes which copy from nodes  $b$  to  $g$ , separately. So that *CHMT* model keeps both persistence and light clustering property without losing a tree structure. In addition to interscale, each node of the *CHMT* model gets light intrascale dependences from adjacent coefficients cause of the extended adjacent nodes.

In the *CHMT* model, the hidden state variable  $S_{e(i)}$  of an extended node is only dependent on the hidden state variable  $S_i$  of node  $i$  and independent to other hidden state variables; that is, the intrascale dependence only influences on the coefficient and its extended coefficients and is not propagated to others.

The *CHMT* model is a parametric model to set the joint *pmf* of the wavelet coefficients. It has strong interscale dependences and light intrascale dependences. The parameter set  $\theta_{CHMT}$  of the *CHMT* model is given as

$$\theta_{CHMT} = \{p_{S_0}(m), \varepsilon_{c(i),i}^{nm}, \delta_{e(i),i}^{nm}, \mu_{i,m}, \sigma_{i,m}^2 | m, n = 1, 2, \dots, M\}. \quad (2)$$

$p_{S_0}(m)$  is the *pmf* value of the root state variable  $s_0$  in state  $m$ .  $\varepsilon_{c(i),i}^{nm}$  is the state transition probability of the Markov chain between the hidden states of parent-child nodes and is given as the conditional probability

$$\varepsilon_{e(i),i}^{mm} = p_{S_{e(i)}|S_i}(n | s_i = m). \quad (3)$$

$\delta_{e(i),i}^{mm}$  is the state transition probability of the Markov chain between the hidden states of node  $j$  and its extended node, which is given as the conditional probability

$$\delta_{e(i),i}^{mm} = p_{S_{e(i)}|S_i}(n | s_i = m). \quad (4)$$

$\mu_{i,m}$  and  $\sigma_{i,m}^2$  are the mean and variance of Gaussian mixture model for the wavelet coefficient  $w_i$  given  $s_i$  in state  $m$ .

The *CHMT* model builds a hierarchical Markov chain by the way of setting state translation probabilities between states of wavelet coefficients in a wavelet tree structure. The transition probabilities  $\varepsilon$ 's are used to define the interscale dependences, and  $\delta$ 's are used to define the intrascale dependences.

### III. TEXTURE SEGMENTATION USING *CHMT* MODELS

A wavelet-based multiscale segmentation method using the *CHMT* model is proposed. The key steps of this method are model training, multiscale segmentation, and boundary refinement.

#### A. Data Structures for Segmentation

Traditional segmentation algorithms employ a fix-sized window to aggregate the same-classed pixels in the window. Obviously, the size of the classification window is crucial to the segmentation result. A larger window usually strengthens the classification reliability because more pixels provide more rich statistical information, but has more risks in which pixels belong to different classes. Thus, a larger window produces accurate segmentations in large and homogeneous regions but poor segmentation on boundaries between different regions. A smaller window reduces the possibility of pixels having multiple classes in a window, but has lower classification reliability for less statistical information.

To capture the properties of every texture region for segmentation, both large and small windows should be utilized. A natural way to achieve the purpose is analysis in multiscale. In a multiscale segmentation, the results of different-sized classification windows are combined to obtain a better segmentation result. In this study, we employ the dyadic squares to implement classification windows of different sizes as shown in Fig. 4(a). The dyadic squares are from dividing the image into four subimages of equal size recursively until the size of dyadic square is equal to one pixel. We use  $d_i^j$  to indicate the dyadic square at scale  $j$  with an abstract index enumerating  $i$  like the numbers. Given a  $2J \times 2J$  hierarchy image, the dyadic square  $d_0^0$  denotes the whole image and  $d_i^j$  denotes an individual pixel. Every square  $d_i^j$  has a parent square  $d_{p(i)}^{j-1}$  and four children squares  $d_{c(i)}^{j+1}$ .

In a wavelet tree, each subtree  $T_i$  in *LH* or *HL* or *HH* represents the information of a certain square region which corresponds to one dyadic square in an image. Depending on the tree structure, the abstract index enumerating of wavelet subtree can be set as one example shown in Fig. 4(b). In

accordance with the index numbers in the squares and subtree, the information of subtree  $T_i$  in *LH*, *HL*, and *HH* bands can be merged to the relative dyadic square  $d_i$  such as Fig. 4(c).

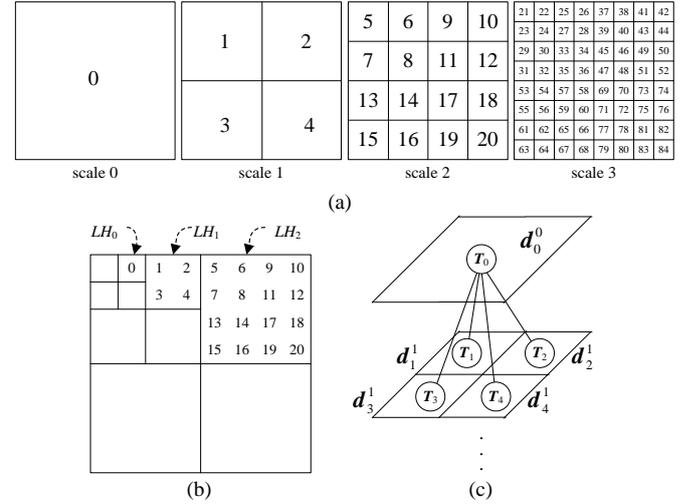


Fig. 4. The relationship of dyadic squares and wavelet subtrees. (a) The dyadic squares in an image. (b) The *LH* subtree in a wavelet tree. (c) Quad-tree structure of dyadic squares.

#### B. *CHMT* Model Training

The purpose of the proposed segmentation method is to assign a suitable class label to every dyadic square in all scales. Before the segmentation, the *CHMT* models of all textures must be trained. At first, each texture image is transformed into wavelet domain, then the hidden states of wavelet coefficients are characterized.

Let  $T_i$  be the subtree with root at node  $i$ , so the subtree  $T_i$  contains coefficient  $w_i$  and all its descendants. If  $T_j$  is a subtree of  $T_i$ , then we define  $T_{ij}$  to be the tree  $T_i$  removed its subtree  $T_j$ . Thus,  $T_0$  is the entire tree of all observed coefficients including extended coefficients. Let  $\mathbf{w}$  is the set of all wavelet coefficients in the entire tree  $T_0$ .  $c(i,j)$  is defined as the  $j$ 'th child node of node  $i$  and  $e(i,j)$  is defined as the  $j$ 'th extended node of node  $i$ . For each subtree  $T_i$ , we define the conditional likelihood functions  $\beta_i$ 's as

$$\beta_i(m) = f(\mathbf{T}_i | S_i = m, \boldsymbol{\theta}_{CHMT}), \quad (5)$$

$$\beta_{i/c(i,j)}(m) = f(\mathbf{T}_{i/c(i,j)} | S_i = m, \boldsymbol{\theta}_{CHMT}), \quad (6)$$

$$\beta_{i/e(i,j)}(m) = f(\mathbf{T}_{i/e(i,j)} | S_i = m, \boldsymbol{\theta}_{CHMT}), \quad (7)$$

and joint probability  $\alpha_i$ 's for all tree nodes are

$$\alpha_i(m) = p(S_i = m, \mathbf{T}_{0/i} | \boldsymbol{\theta}_{CHMT}). \quad (8)$$

The trees  $T_i$  and  $T_{0/i}$  are independent given the state variable  $S_i$ ; that is, the  $\alpha$ 's and  $\beta$ 's are independent. Then the joint probability of node  $i$  in state  $m$  and entire tree is written as

$$p(S_i = m, \mathbf{T}_0 | \boldsymbol{\theta}_{CHMT}) = \alpha_i(m) \beta_i(m) \quad (9)$$

and the entire likelihood function of observed coefficients  $\mathbf{w}$  is

$$f(\mathbf{w} | \boldsymbol{\theta}_{CHMT}) = \sum_{m=1}^M p(S_i = m, \mathbf{w} | \boldsymbol{\theta}_{CHMT}) = \sum_{m=1}^M \alpha_i(m) \beta_i(m). \quad (10)$$

Based on the chain rule of transition probability, the entire likelihood function can be obtained from the joint probabilities of node  $i$  in its child node  $c(i,j)$  in state  $n$

$$f(\mathbf{w} | \boldsymbol{\theta}_{CHMT}) = \sum_{n=1}^M \sum_{m=1}^M \beta_{c(i,j)}(n) \varepsilon_{c(i,j),i}^{nm} \alpha_i(m) \beta_{i/c(i,j)}(m), \quad (11)$$

and extended node  $e(i,j)$  in state  $n$

$$f(\mathbf{w} | \boldsymbol{\theta}_{CHMT}) = \sum_{n=1}^M \sum_{m=1}^M \beta_{e(i,j)}(n) \delta_{e(i,j),i}^{nm} \alpha_i(m) \beta_{i/e(i,j)}(m). \quad (12)$$

After  $\alpha$ 's and  $\beta$ 's are determined, the *pmf* for each node and each chain of two nodes can be calculated by Bayes's rule,

$$p(S_i = m | \mathbf{w}, \boldsymbol{\theta}_{CHMT}) = \frac{\alpha_i(m) \beta_i(m)}{\sum_{n=1}^M \alpha_i(m) \beta_i(m)}, \quad (13)$$

$$\begin{aligned} p(S_{c(i,j)} = n, S_i = m | \mathbf{w}, \boldsymbol{\theta}_{CHMT}) \\ = \frac{\beta_{c(i,j)}(n) \varepsilon_{c(i,j),i}^{nm} \alpha_i(m) \beta_{i/c(i,j)}(m)}{\sum_{n=1}^M \sum_{m=1}^M \beta_{c(i,j)}(n) \varepsilon_{c(i,j),i}^{nm} \alpha_i(m) \beta_{i/c(i,j)}(m)}, \end{aligned} \quad (14)$$

and

$$\begin{aligned} p(S_{e(i,j)} = n, S_i = m | \mathbf{w}, \boldsymbol{\theta}_{CHMT}) \\ = \frac{\beta_{e(i,j)}(n) \delta_{e(i,j),i}^{nm} \alpha_i(m) \beta_{i/e(i,j)}(m)}{\sum_{n=1}^M \sum_{m=1}^M \beta_{e(i,j)}(n) \delta_{e(i,j),i}^{nm} \alpha_i(m) \beta_{i/e(i,j)}(m)}. \end{aligned} \quad (15)$$

All state variables in the *CHMT* model are interdependent. We use *EM* algorithm to compute the *CHMT* model parameters based on the maximum likelihood principle [10]. The algorithm consists of three steps: up expectation, down expectation, and maximization steps. Based on the parameter set  $\boldsymbol{\theta}_{CHMT}$ , the up step determines the likelihood of subtrees from the finest scale to the coarsest scale of wavelet coefficients in a wavelet tree and the down step determines the joint probability of subtrees from the coarsest scale to the finest scale in a wavelet tree. Combining these two steps, all state transition probabilities are evaluated. Finally, we get new model parameters  $\boldsymbol{\theta}_{CHMT}$  by calculating the mean of expectation results in the maximization step [22].

For reducing the number of parameters we use the universal tying strategy [12] to tie parameters for estimation. The wavelet coefficients tend to display similar statistical characteristics within the same scale, so we can often use the same parameters for those coefficients in the same scale.

### C. Multiscale Segmentation

We use the *CHMT* model parameters for every texture image to compute the likelihood functions of dyadic squares at different scales in segmented images. Then a class label with maximum likelihood is assigned to each dyadic square. A 2-D image wavelet transform comprised three subbands

with three quad tree structures. The subtrees  $\mathbf{T}_i^{LH}$ ,  $\mathbf{T}_i^{HL}$ , and  $\mathbf{T}_i^{HH}$  in the *LH*, *HL*, and *HH* subbands respectively correspond to the dyadic square  $\mathbf{d}_i$  in the image as shown in Fig. 4. For making computation problem to be more tractable, we assume that the three wavelet trees are statistical independent. Then the likelihood function of a dyadic square can be computed as

$$f(\mathbf{d}_i | \boldsymbol{\theta}_{CHMT}) = f(\mathbf{T}_i^{LH} | \boldsymbol{\theta}_{CHMT}^{LH}) f(\mathbf{T}_i^{HL} | \boldsymbol{\theta}_{CHMT}^{HL}) f(\mathbf{T}_i^{HH} | \boldsymbol{\theta}_{CHMT}^{HH}), \quad (16)$$

where

$$f(\mathbf{T}_i | \boldsymbol{\theta}_{CHMT}) = \sum_{m=high,low} \beta_i(m) p(S_i = m | \boldsymbol{\theta}_{CHMT}). \quad (17)$$

In this study, we model each wavelet coefficient as a mixture model with two states. The ‘‘high’’ state represents that a wavelet coefficient has large value and contains significant contribution to signal energy, and ‘‘low’’ state indicates a wavelet coefficient having small value and little signal energy. The conditional likelihood function  $\beta_i(m)$  obtained by sweeping up the quadtree from fine-scale to coarse-scale and the state probabilities  $p(S_i = m | \boldsymbol{\theta}_{CHMT})$  can be got from the parameter set  $\boldsymbol{\theta}_{CHMT}$ . Each dyadic square is raw classified using the maximum likelihood segmentation as

$$c_i = \arg \max_c f(\mathbf{d}_i | \boldsymbol{\theta}_{CHMT}). \quad (18)$$

This segmentation does not exploit any relationship between the different scales.

### D. Pixel-Level Segmentation

The wavelet tree characterize the joint statistics of dyadic image squares only down to  $2 \times 2$  blocks, so a pixel-level segmentation is needed separately to implement.

Pixel intensity is taken as the probability density function value for an individual pixel. We use an *EM* algorithm to train a two-state Gaussian mixture model to fit the probability density function for each training datum. Let  $x$  denote the set of intensity values, and  $x_i$  be a pixel value. Let  $\mu_m$ ,  $\sigma_m^2$ , and  $\rho_m$  denote the mean, variance, and probability mass function of state  $m$ , respectively. Let  $\boldsymbol{\theta}_{GMM}$  be a parameter set which consists of  $\{\mu_m, \sigma_m^2, \rho_m\}$  at state  $m$ . We start the *EM* algorithm for the *GMM* with an initial parameter set  $\boldsymbol{\theta}_{GMM}^0$ . At the  $k$ 'th iteration, the *E* step is compute as

$$p(S_i = m | x_i, \boldsymbol{\theta}_{GMM}^k) = \frac{p(x_i | S_i = m, \boldsymbol{\theta}_{GMM}^k) p(S_i = m | \boldsymbol{\theta}_{GMM}^k)}{p(x_i | \boldsymbol{\theta}_{GMM}^k)}, \quad (19)$$

$$m = 1, 2.$$

Then we update the parameter set  $\boldsymbol{\theta}_{GMM}^k$  to  $\boldsymbol{\theta}_{GMM}^{k+1}$  in  $M$  step with

$$\hat{\mu}_m = \frac{\sum_{i=1}^n x_i p(S_i = m | x_i, \boldsymbol{\theta}_{GMM}^k)}{\sum_{i=1}^n p(S_i = m | x_i, \boldsymbol{\theta}_{GMM}^k)}, \quad (20)$$

$$\hat{\sigma}_m^2 = \frac{\sum_{i=1}^n (x_i - \hat{\mu}_m)^2 p(j | x_i, \theta_{GMM}^k)}{\sum_{i=1}^n p(S_i = m | x_i, \theta_{GMM}^k)}, \quad (21)$$

and

$$\hat{\rho}_m = \frac{1}{n_p} \sum_{i=1}^n p(S_i = m | x_i, \theta_{GMM}^k), \quad (22)$$

where  $n_p$  is the number of pixel in an image. We can then compute the likelihood of each pixel of segmented image to categorize pixels into classes.

### E. Boundary Refinement

Through the *CHMT*-based and pixel-level segmentations, the class label of each dyadic square in every scale has been decided; that is, every dyadic square has been judged the belonging of one texture class. The segmentation result suffers from low resolution at coarse scales and instability at finer scales. To improve the results, a multiscale decision fusion method was used to combine the results of the *CHMT* and pixel-level results. Choi and Baraniuk [12] proposed a context-based interscale fusion method to fuse the class decisions at different scales; but there, we adopt Mor and Aladjem's [16] boundary refinement method because it has higher performance.

Given the posterior probability  $p(c_i | x)$ , the maximum a posterior (*MAP*) classification of image  $x$  corresponds to the class label that maximizes the posterior probability

$$\hat{c}_i^{MAP} = \arg \max_{c_i \in \{1, 2, \dots, N_c\}} p(c_i | \mathbf{x}). \quad (23)$$

By Bayesian rule, the posterior distribution is

$$p(c_i | \mathbf{x}) = \frac{f(\mathbf{x} | c_i) p(c_i)}{f(\mathbf{x})}. \quad (24)$$

Let  $\mathbf{d}^j$  denote the set of all dyadic squares at scale  $j$  and each  $\mathbf{d}^j$  contains complete information of image  $x$ . Thus, the posterior can be described as

$$p(c_i^j | \mathbf{d}_i^j) = \frac{f(\mathbf{d}_i^j | c_i^j) p(c_i^j)}{f(\mathbf{d}_i^j)}, \quad (25)$$

where  $c_i^j$  is the class label of dyadic squares  $\mathbf{d}_i^j$  in scale  $j$ .

The boundary refinement sets boundary probability to solve the *MAP* problem. At first, the prior probability of texture class was got from the class labels of previous scale

$$p(c_i^j) = \frac{N(c_i^j)}{9}, \quad (26)$$

where  $N(c_i^j)$  is the number of dyadic squares which have the same class label with  $c_i^j$  in square  $\mathbf{d}_{p(i)}^{j-1}$  and its eight neighbor squares. Then we use the Bayes law, to establish the raw posterior probability for the  $M$  classes

$$p(c_i^j | \mathbf{d}_i^j) = \frac{f(\mathbf{d}_i^j | c_i^j) p(c_i^j)}{\sum_{c_i^j=1}^M f(\mathbf{d}_i^j | c_i^j) p(c_i^j)}, \quad (27)$$

where  $f(\mathbf{d}_i^j | c_i^j)$  is given by (34).

In order to get more correct segmentation, we prefer the class label at finer scale near boundaries and coarser scale at smooth regions. For this purpose, the boundary variable  $b_i^j$  is build up and used to express the position characteristic,

$$b_i^j = \begin{cases} 0, & \text{if } \{c_{ch(i)}^{j+1}\} = c_i^j \\ 1, & \text{otherwise} \end{cases}. \quad (28)$$

We consider that the class label  $c_i^j$  of square  $\mathbf{d}_i^j$  is same with all the class labels  $c_{ch(i)}^{j+1}$  of its four children squares in homogeneous regions then the variable  $b_i^j$  is set to zero. Otherwise, the variable  $b_i^j$  is set to one. Thus the probability  $p(b_i^j = 0)$  will be high in homogeneous regions and the probability  $p(b_i^j = 1)$  is anticipated to be high in the regions with boundaries.

Given the boundary variable at parent node, the conditional posteriori probability is computed

$$p(c_i^j | \mathbf{d}_{A(i)}^L, b_{p(i)}^{j-1}) = \begin{cases} p(c_{p(i)}^{j-1} | \mathbf{d}_{A(p(i))}^L), & \text{if } b_{p(i)}^{j-1} = 0 \\ p(c_i^j | \mathbf{d}_i^j), & \text{if } b_{p(i)}^{j-1} = 1 \end{cases}, \quad (29)$$

where  $A(i)$  means the ancestor of node  $i$  at the scale  $L$  and  $\mathbf{d}_{A(i)}^L$  is the set of  $\{\mathbf{d}_i^j, \mathbf{d}_{p(i)}^{j-1}, \mathbf{d}_{p(p(i))}^{j-2}, \dots, \mathbf{d}_{A(i)}^L\}$ . The averaged posteriori of the class label of ancestor dyadic squares is

$$p(c_i^j | \mathbf{d}_{A(i)}^L) = \sum_{m=0}^1 p(c_i^j | \mathbf{d}_{A(i)}^L, b_{p(i)}^{j-1}) p(b_{p(i)}^{j-1} = m). \quad (30)$$

If  $b_{p(i)}^{j-1} = 1$ , the conditional posterior probability  $p(c_i^j | \mathbf{d}_{A(i)}^L, b_{p(i)}^{j-1})$  is the original raw posterior at the current scale. The finer scale raw posterior probabilities are used in boundary regions. In smooth regions,  $b_{p(i)}^{j-1} = 0$ , the average posterior probability  $p(c_{p(i)}^{j-1} | \mathbf{d}_{A(p(i))}^L)$  has sufficient statistical property, so that it is used in place of the conditional posterior probability of the current finer scale.

In (30), the boundary probability  $p(b_{p(i)}^{j-1} = m)$  is usually unknown in advance. We use an iterative procedure to estimate the boundary probability and then substitute (29) and (30). The estimation is iteratively refined until the boundary probabilities have converged. For estimating the  $p(b_{p(i)}^{j-1})$ , we assume the four class labels  $\{c_{ch(p(i))}^j\}$  are independent given their ancestor dyadic squares  $\mathbf{d}_{A(q)}^L$ . The probability of class labels  $\{c_{ch(p(i))}^j\}$  being equal to the same class label  $C$  is

$$p(\{c_{ch(p(i))}^j\} = C) = \prod_q p(c_q^j = C | \mathbf{d}_{A(q)}^L), \quad (31)$$

where  $q$  is an abstract index of the four children of  $p(i)$ . Substituting (29) and (30) into (31) and summing over all class labels, we can get

$$p(b_{p(q)}^{j-1} = 0) - \sum_{C=1}^M \prod_q \{p(c_{p(q)}^{j-1} = C | \mathbf{d}_{A(p(q))}^L) p(b_{p(q)}^{j-1} = 0) + p(c_{p(q)}^j = C | \mathbf{d}_q^j) p(b_{p(q)}^{j-1} = 1)\} = 0. \quad (32)$$

Due to the fact  $p(b_{p(q)}^{j-1} = 1) + p(b_{p(q)}^{j-1} = 0) = 1$ , equation (32) becomes a function of a single variable and can be solved by any standard root finding method. Here, we use Secant method [23] to solve (32); the solution converges quickly and the initial values do not influence the result.

Finally, the *MAP* segmentation problem of (23) is transformed to

$$c_i^j = \arg \max_{c_i^j} p(c_i^j | \mathbf{d}_{A(i)}^L), \quad c_i^j = 1, \dots, M. \quad (33)$$

This merging procedure is run repeatedly from a suitable scale to the finest scale; that is, the results of raw segmentation are fused by considering the stability of coarse segmented images at smooth regions and the finer segmented images near texture boundaries.

#### IV. EXPERIMENTS

We demonstrate the performance of the proposed method on four  $256 \times 256$  synthetic images, “mix-2”, “mix-3”, “mix-4”, and “mix-5”, shown in Fig. 5. Their ground-truth segmentation results were given in Fig. 6. We compare the results of raw and final segmentations with *HMT*seg [12] and Mor’s [16] methods.

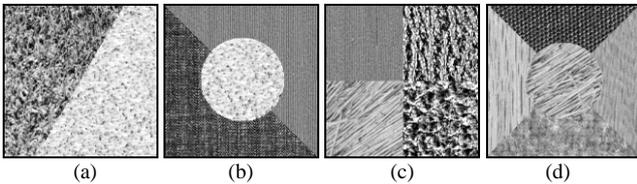


Fig. 5. The synthetic test images (a) “mix-2”, (b) “mix-3”, (c) “mix-4”, and (d) “mix-5”.

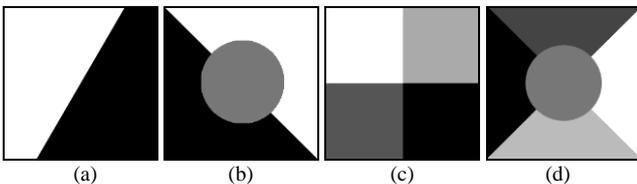


Fig. 6. The ground-truth segmentation results of (a) “mix-2”, (b) “mix-3”, (c) “mix-4”, and (d) “mix-5”.

We use the texture data from the usc-sipi image database. For every texture, we randomly selected ten  $128 \times 128$  images for training *CHMT* parameters. The *EM* algorithm converged in 10 iterations. In our experiments, the raw segmentations at dyadic square size  $16 \times 16$  are usually considered as good enough for fusing the smaller dyadic squares. The correction rate is used to evaluate the segmented results and is defined the ratio of correct classified pixels to all pixels.

Table I shows the correction rates of raw segmentation at different dyadic square size using *HMT* and *CHMT* model. Table II shows the correction rates of raw pixel-level segmentation. Fig. 7-Fig. 10 show the raw segmented results.

Table III shows the correction rates of final segmentation from the three test methods. Fig. 11 and Fig. 12 show the final segmentation results, respectively.

TABLE I: THE CORRECTION RATES OF RAW SEGMENTATION

Image	Model	16×16 square	8×8 square	4×4 square	2×2 square
mix-2	<i>HMT</i>	0.9090	0.8786	0.8543	0.7663
	<i>CHMT</i>	0.9207	0.8645	0.8806	0.8567
mix-3	<i>HMT</i>	0.8593	0.8796	0.8466	0.7136
	<i>CHMT</i>	0.8626	0.8810	0.8849	0.8174
mix-4	<i>HMT</i>	0.8633	0.8389	0.7517	0.5976
	<i>CHMT</i>	0.8906	0.8857	0.8174	0.7178
mix-5	<i>HMT</i>	0.7337	0.7340	0.6057	0.4325
	<i>CHMT</i>	0.7836	0.7943	0.6956	0.5877

TABLE II: THE CORRECTION RATES OF RAW PIXEL-LEVEL SEGMENTATION

Image	mix-2	mix-3	mix-4	mix-5
pixel-level square	0.8552	0.8593	0.5292	0.5675

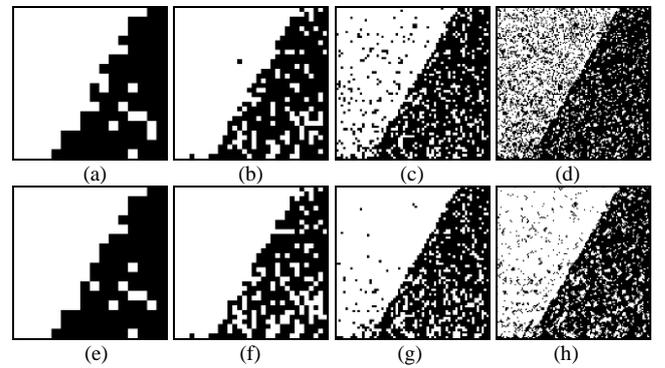


Fig. 7. Raw segmentation results of image “mix-2” in *HMT* model with (a)  $16 \times 16$ , (b)  $8 \times 8$ , (c)  $4 \times 4$ , (d)  $2 \times 2$  dyadic squares and in *CHMT* model with (e)  $16 \times 16$ , (f)  $8 \times 8$ , (g)  $4 \times 4$ , (h)  $2 \times 2$  dyadic squares.

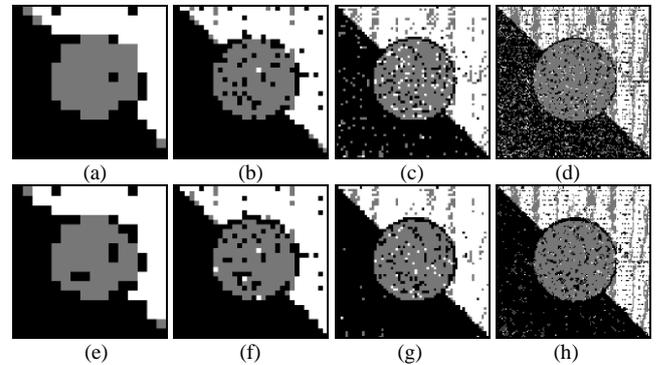


Fig. 8. Raw segmentation results of image “mix-3” in *HMT* model with (a)  $16 \times 16$ , (b)  $8 \times 8$ , (c)  $4 \times 4$ , (d)  $2 \times 2$  dyadic squares and in *CHMT* model with (e)  $16 \times 16$ , (f)  $8 \times 8$ , (g)  $4 \times 4$ , (h)  $2 \times 2$  dyadic squares.

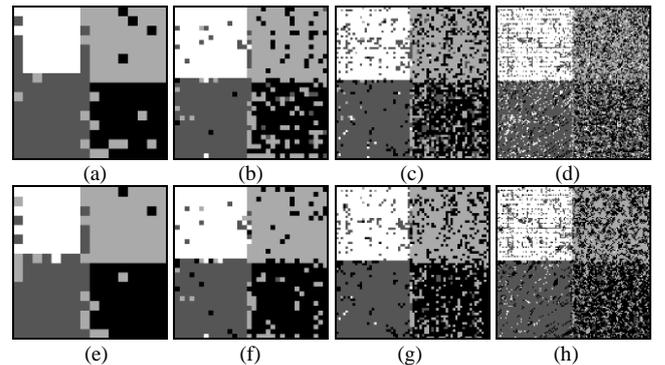


Fig. 9. Raw segmentation results of image “mix-4” in *HMT* model with (a)  $16 \times 16$ , (b)  $8 \times 8$ , (c)  $4 \times 4$ , (d)  $2 \times 2$  dyadic squares and in *CHMT* model with (e)  $16 \times 16$ , (f)  $8 \times 8$ , (g)  $4 \times 4$ , (h)  $2 \times 2$  dyadic squares.

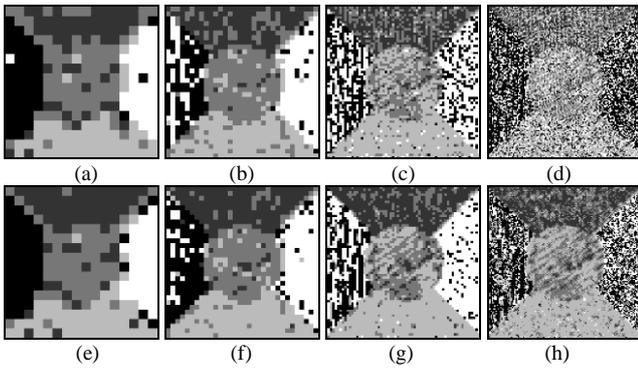


Fig. 10. Raw segmentation results of image “mix-5” in *HMT* model with (a)  $16 \times 16$ , (b)  $8 \times 8$ , (c)  $4 \times 4$ , (d)  $2 \times 2$  dyadic squares and in *CHMT* model with (e)  $16 \times 16$ , (f)  $8 \times 8$ , (g)  $4 \times 4$ , (h)  $2 \times 2$  dyadic squares.

TABLE III: THE CORRECTION RATES OF FINAL SEGMENTATION

Image	Method	8×8 square	4×4 square	2×2 square	pixel-level square
mix-2	<i>HMTseg</i>	0.8927	0.8941	0.8957	0.9593
	Mor's method	0.9466	0.9535	0.9594	0.9705
	The proposed method	0.9563	0.9594	0.9626	0.9733
mix-3	<i>HMTseg</i>	0.8676	0.9008	0.8681	0.9127
	Mor's method	0.8987	0.9290	0.9390	0.9595
	The proposed method	0.9207	0.9475	0.9546	0.9750
mix-4	<i>HMTseg</i>	0.8652	0.8843	0.7319	0.7768
	Mor's method	0.9502	0.9570	0.9597	0.9659
	The proposed method	0.9561	0.9629	0.9650	0.9716
mix-5	<i>HMTseg</i>	0.7808	0.7549	0.4817	0.4372
	Mor's method	0.8788	0.8963	0.9039	0.9197
	The proposed method	0.8979	0.9149	0.9181	0.9318

Because the *CHMT* model enhances the clustering property in wavelet domain with connecting extended adjacent nodes, the segmentation results in Table I shows that *CHMT* model method has better correction rates than *HMT* model. Obviously, the images in Fig. 7-Fig. 10 show that *CHMT* model method in the smooth region has more precise segmentation results. In Table III, the final segmentation using boundary refinement gets better correction rates than intescale fusion in *HMTseg*. After improving the raw segmentations by *CHMT* model, the boundary refinement method can get superior correction rates from the better raw segmentations. The experiment results also show the goodness of multiscale segmentation. The finer scales image usually have less correction rates because the small windows with little information for computing statistic model. Contrary, the appropriate coarser scales have more correction rates. Using boundary refinement to combine the raw segmentation results from coarser to finer scales, the good segmentation results at coarser scales can refine the raw segmentation results.

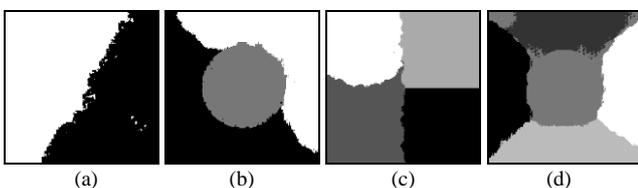


Fig. 11. The final segmentation results of image (a) “mix-2”, (b) “mix-3”, (c) “mix-4”, and (d) “mix-5” by Mor's method.

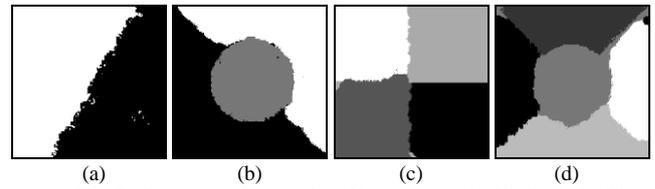


Fig. 12. The final segmentation results of image (a) “mix-2”, (b) “mix-3”, (c) “mix-4”, and (d) “mix-5” by the proposed method.

Two  $256 \times 256$  aerial photos: land region and water region were used for practically testing as shown in Fig. 13(a) and 14(a). Fig. 13(b) and Fig. 14(b) show the results from Mor's method. Fig. 13(c) and Fig. 14(c) show the results from the proposed method. We can see that there are more white labels in the water region in Fig. 13(b) and Fig. 14(b). So the final segmentation results show that the performance of the proposed method is better.

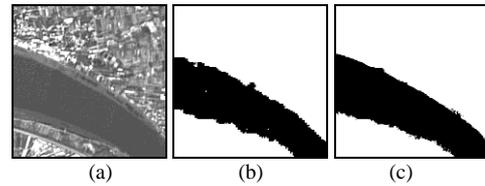


Fig. 13. The test aerial photo, (a) “aerial-1”, and the final segmentation from (b) Mor's method and (c) the proposed method.

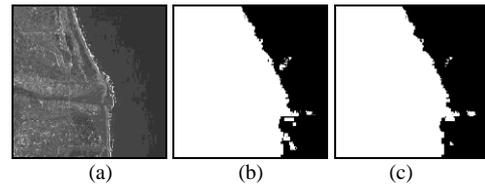


Fig. 14. The test aerial photo, (a) “aerial-2”, and the final segmentation from (b) Mor's method and (c) the proposed method.

## V. CONCLUSIONS

In this paper, we use the proposed *CHMT* model combining the multiscale boundary refinement method to segment textures. By adding extended coefficients associated with wavelet coefficients in every scale, the *CHMT* model holds persistence properties as the *HMT* model and enhances the clustering properties in wavelet domain. The experiments showed that the proposed method uses more parameters and more computation costs but get better segmentation performance than *HMTseg* and Mor's methods.

There are still some tasks for future research. We will study more appropriate wavelet model that include shift invariant property and high frequency information to fit the real image data more precisely. The boundaries between textures are the critical regions that usually make the performance worse. So the performance of classification in boundary regions must be improved. For more real image applications, the parallel computation for reducing executed time is also an important issue.

## REFERENCES

- [1] M. H. Bharati, J. J. Liu, and J. F. MacGregor, “Image texture analysis: methods and comparisons,” *Chemometrics and Intelligent Laboratory Systems*, vol. 72, pp. 57-71, June 2004.
- [2] G. Castellano, L. Bonilha, L. M. Li, and F. Cendes, “Texture analysis of medical images,” *Clinical Radiology*, vol. 59, no. 2, pp. 1061-1069, Dec. 2004.

- [3] J. G. Zhang and T. N. Tan, "Brief review of invariant texture analysis methods," *Pattern Recognition*, vol. 35, pp. 735-747, Mar. 2002.
- [4] L. M. Kaplan, "Extended fractal analysis for texture classification and segmentation," *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1572-1585, Nov. 1999.
- [5] T. M. Nguyen and Q. M. J. Wu, "Gaussian-mixture-model-based spatial neighborhood relationships for pixel labeling problem," *IEEE Trans. Systems, Man, and Cybernetics Part B-Cybernetics*, vol. 42, no. 1, pp. 193-202, Feb. 2012.
- [6] J.-L. Chen and A. Kundu, "Automatic unsupervised texture segmentation using hidden Markov model," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Minneapolis, Minnesota, Apr. 27-30, 1993, pp. 21-24.
- [7] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden Markov model," *IEEE Trans. Signal Processing*, vol. 48, no. 2, pp. 517-533, Feb. 2000.
- [8] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss-Markov random field models for texture segmentation," *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 251-267, Feb. 1997.
- [9] R. Porter and N. Canagarajah, "Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes," *IEE Proc. Vision Image Signal Processing*, vol. 144, no. 3, pp. 180-188, June 1997.
- [10] 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, no. 4, pp. 886-902, Apr. 1998.
- [11] H. Choi and R. G. Baraniuk, "Image segmentation using wavelet-domain classification," in *Proc. SPIE 3816, Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, Denver, Colorado, July 1999, pp. 306-320.
- [12] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models," *IEEE Trans. Image Processing*, vol. 10, no. 9, pp. 1039-1321, Sep. 2001.
- [13] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Processing*, vol. 3, no. 2, pp. 162-177, Mar. 1994.
- [14] G. L. Fan and X. G. Xia, "Wavelet-based texture analysis and synthesis using hidden Markov models," *IEEE Trans. Circuits and Systems I-Fundamental Theory and Applications*, vol. 50, no. 1, pp. 106-120, Jan. 2003.
- [15] Q. Xu, J. Yang, and S. Y. Ding, "Color texture analysis using wavelet-based hidden Markov model," *Pattern Recognition Letters* vol. 26, no. 11, pp. 1710-1719, Aug. 2005.
- [16] E. Mor and M. Aladjem, "Boundary refinements for wavelet-domain multiscale texture segmentation," *Image and Vision Computing*, vol. 23, no. 13, pp. 1150-1158, Nov. 2005.
- [17] X.-Z. Liu, B. Fang, and Z.-W. Shang, "Texture image segmentation using complex wavelet transform and hidden Markov models," in *Proc. 1st Int. Conf. Wavelet Analysis and Pattern Recognition*, Baoding, China, July 12-15, 2009, pp. 396-401.
- [18] Y. H. Zhang, Y. S. Zhang, Z. F. He, and X. Y. Tang, "Multiscale fusion of wavelet-domain hidden Markov tree through graph cut," *Image and Vision Computing*, vol. 27, pp. 1402-1410, Aug. 2009.
- [19] T. H. Kim, I. K. Eom, and Y. S. Kim, "Multiscale Bayesian texture segmentation using neural networks and Markov random fields," *Neural Computing & Application*, vol. 18, no. 2, pp. 141-155, Feb. 2009.
- [20] C. Zheng, Q. Qin, G. Liu, and Y. Hu, "Image segmentation based on multiresolution Markov random field with fuzzy constraint in wavelet domain," *IET Image Processing*, vol. 6, no. 3, pp. 213-221, 2012.
- [21] D.-C. Tseng and M. Y. Shih, "Wavelet-based image denoising using contextual hidden Markov tree model," *Journal of Electronic Imaging*, vol. 14, no. 3, pp. 1-12, July-Sep. 2005.
- [22] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceeding of the IEEE*, vol. 77, no. 2, pp. 257-285, Feb. 1989.
- [23] F. V. L. Charles, *Introduction to Scientific Computing: A Matrix Vector Approach Using MATLAB*, Prentice Hall, New York, 1996.



**Din-Chang Tseng** received his Ph.D. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in June 1988. He has been a professor in the Department of Computer Science and Information Engineering at National Central University, Zhongli, Taiwan since 1996. He is a member of IEEE. His current research interests include computer vision, image processing, and virtual reality, especially in the topic: computer vision system for advanced safety vehicle, computer vision techniques for human computer interaction, and view-dependent multi-resolution terrain modeling.



**Ruei-Lung Chen** received his B.S. degree in electronic engineering from Chung Yuan Christian University, Zhongli, Taiwan and M.S. degree in electrical engineering from National Chung Cheng University, Chiayi, Taiwan, in 1998 and 2000 respectively. He is pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering at National Central University, Zhongli, Taiwan. His research interests include image processing and pattern recognition.