TEXTURE SEGMENTATION USING STATISTICAL CHARACTERISTICS OF SOM AND MULTI-SCALE BAYESIAN ESTIMATION

Tae Hyung Kim, Il Kyu Eom**, Yoo Shin Kim**

*Dept. of Electronic Engineering, Pusan National University, South Korea
** Dept. of Info. & Comm. Engineering, Miryang National University, South Korea
*{thyunkim,kimys}@pusan.ac.kr, **ikeom@mnu.ac.kr

ABSTRACT

This paper presents a novel texture segmentation method using Bayesian estimation and SOM (self organizing feature map). Multi-scale wavelet coefficients are used as input for SOM, and likelihood probabilities for observations are obtained from trained SOMs. Texture segmentation is performed by the likelihood probability from trained SOMs and ML (maximum likelihood) classification. The result of texture segmentation is improved using contextual information. The proposed segmentation method performed better than segmentation method using HMT (hidden Markov trees) model. In addition, texture segmentation results by SOM and multi-scale Bayesian image segmentation technique called HMTseg also performed better than those by HMT and HMTseg.

1. INTRODUCTION

The goal of texture segmentation is to divide an image into homogeneous textured regions and identify the boundaries between regions. Texture segmentation can be categorized as supervised and unsupervised segmentation. In supervised segmentation problems, the number of textures and their associated model parameters are known, or estimated before segmentation. In unsupervised segmentation problems, such knowledge is not available before segmentation. Thus segmentation methods are determined according to each segmentation problem.

Another factor that influences the choice of segmentation method is the texture description or feature extraction method [2]. Of all methods for the texture feature extraction, signal processing methods are attractive due to their simplicity and supported by psychophysical research which has given evidence that the human brain does a frequency analysis of the image [1]. Fourier or wavelet transform [4] can be used to extract texture image features. The resulting transformed features in multi-scale are efficient for texture segmentation [4]. Therefore, the HMT model in multi-scale wavelet-domain was used for texture segmentation [4]. Kohonen neural networks [3], or fuzzy c-means, with some features extracted from wavelet transform, were also used for texture segmentation. Neural networks can represent any distribution of inputs without complicated modeling methods [5]. SOM can be trained in unsupervised training mode [8]. A trained SOM forms prototypes for training data [5-6], [8].

In this paper, we propose a novel method of supervised texture segmentation using SOM in a multi-scale wavelet domain and a multi-scale Bayesian image segmentation technique called HMTseg. Firstly, a method for obtaining the likelihood probability of observations from SOM is proposed. Texture segmentation is performed by using likelihood from SOM and ML classification. Secondly, a method using contextual information is proposed for improving texture segmentation by SOM. Finally, the proposed method uses HMTseg to fuse the multi-scale segmentations and to improve texture segmentation at the finest scale.

2. THE LIKELIHOOD ESTIMATION FROM SOM

In this section, we explain the method to estimate probability density from training samples or *prototypes* [5]. Then, from this estimation method we propose a method for estimating a distribution from SOM.

2.1. Statistical characteristics of SOM

SOM learns by self-organizing and competition [8]. SOM has one layer, which is called the competitive layer, and neuron nodes in that layer are arranged in a lattice. The weight vectors are connected to the neuron nodes in the competitive layer and have identical dimensions as the input vectors. When an input vector is fed into SOM, the neuron whose weight vector is closest to the input vector according to some measure (e.g. Euclidean distance) is called the winning neuron for that input vector.

In SOM training, if a training vector $\boldsymbol{G}$ is fed into SOM, the weight vectors, which are connected to a winning neuron node i^* and all neuron nodes within a certain neighborhood $N_{i^*}(d)$ of the wining neuron, are updated using the Kohonen rule [8]. Here, $N_{i^*}(d) = \{ j, d_{i^*j} \le d \}$, and d_{i^*j} is the distance between the winning neuron node i^* and the neuron node j in the competitive layer (the distance between positions of neuron nodes on the lattice). Thus,

when a vector ***G*** is presented, the weights of the wining neuron and its close neighbors move toward ***G***. Consequently, after many presentations, weight vectors of neighboring neurons will be close to each other in a vector space. SOM allocates many neighboring neurons when recognizing a region which contains many training vectors in the training vector space, and relatively few neurons when recognizing a region which contains few training vectors. Thus, SOMs learn both the distribution and topology of the input vectors they are trained on. Then the weight vectors of the neurons in the layer become *prototypes* of the training vectors. SOM projects high dimensional training vectors on nodes which are arranged in a two-dimensional lattice.

2.2. Estimation of probability density from *prototypes* [5]

The probability P that vector $\mathbf{x}$ will fall in a region R is given by

$$P = \int_R p(\mathbf{x}')\, d\mathbf{x}' \,. \tag{1}$$

Thus P is an averaged version of the density function $p(\mathbf{x})$, and we can estimate the smoothed value of $p(\mathbf{x})$ by estimating the probability P. Suppose that n samples, $\mathbf{x}_1,\ldots,\mathbf{x}_n$, are drawn independently and identically distributed according to a probability law, $p(\mathbf{x})$. The probability that k of these n fall in R is given by the ratio k/n which will be a good estimate for the probability P. If we now assume that $p(\mathbf{x})$ is continuous and that the region R is so small that $p(\mathbf{x})$ does not vary appreciably within it, we can write

$$\int_R p(\mathbf{x}')d\mathbf{x}' \approx p(\mathbf{x})V \,, \tag{2}$$

where V is the volume enclosed by R. Combining Eqs. 1 and 2, we arrive at the following obvious estimate for $p(\mathbf{x})$,

$$p(\mathbf{x}) \approx \frac{k/n}{V} \,. \tag{3}$$

If we want to obtain $p(\mathbf{x})$ rather than just an averaged version of it, we must be prepared to let V approach zero. From a practical standpoint, the number of samples is always limited. Thus, the volume V cannot be allowed to become arbitrarily small. To estimate $p(\mathbf{x})$ from n training samples or *prototypes* we can center a cell about $\mathbf{x}$ and let it grow until it captures k_n samples, where k_n is some specified function of n. These samples are the k_n nearest-neighbors of $\mathbf{x}$. If the density is high near $\mathbf{x}$, the cell will be relatively small (good resolution). If the density is low, the cell will grow large. In either case, we can take the estimation $p_n(\mathbf{x})$ of $p(\mathbf{x})$ for n training samples as follows:

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n} \,, \tag{4}$$

where V_n is the volume of the cell which contains k_n samples. If the estimation $p_n(\mathbf{x})$ is to converge to density $p(\mathbf{x})$, three conditions appear to be required as follows:

$$\lim_{n\to\infty} V_n = 0, \quad \lim_{n\to\infty} k_n = \infty, \quad \lim_{n\to\infty} k_n/n = 0 \,. \tag{5}$$

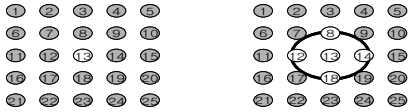


Fig. 1. Elements of K neighboring neurons on the two-dimensional lattice of neurons; left: when K=1; right: when K=5.

In other words, the smaller the region R (the smaller V_n becomes), in which the number of k_n nearest-neighbors of $\mathbf{x}$ are sufficiently many, the more accurate the $p_n(\mathbf{x})$ becomes.

2.3. The likelihood estimation from SOM

We propose a method for estimating likelihood probability for an input vector $\mathbf{x}$ using SOM, which had been trained by training vectors of class c, as follows.

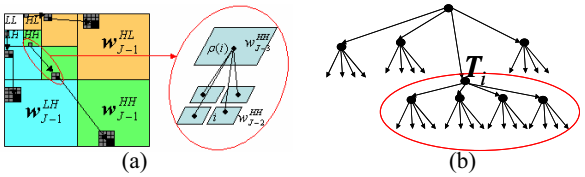
$$f(\mathbf{x}\,|\,c) \approx \frac{K/N}{V_N} = \frac{K/N}{\pi\, r_{c,K}(\mathbf{x})^2} \,, \tag{6}$$

where, $c \in \{1,2,\ldots,N_c\}$ is a class index, N is the number of all neurons in SOM, V_N is a circle with the radius $r_{c,K}(\mathbf{x})$. The radius $r_{c,K}(\mathbf{x})$ is the longest distance of all of the distances between input $\mathbf{x}$ and the weights of K neighboring neurons that are neighbors of winning neuron for the input $\mathbf{x}$ on the two-dimensional lattice of the competitive layer [see Fig. 1]. Thus, if the value of K is fixed, then $r_{c,K}(\mathbf{x})$ can be computed.

Estimating the likelihood probability more exactly from SOM, a proper K exists depending on the size of SOM and the amount of training data. Increasing the value of K results in an increment of the region R ($\pi r_{c,K}(\mathbf{x})^2$) when estimating the distribution of training data. The smaller the amount of training data, the larger the region R ($\pi r_{c,K}(\mathbf{x})^2$) grows according to increasing value of K. When the amount of training data is abundant, increasing the value of K improves the reliability of the estimation of the distribution of training data, and decreases the influence of noise. But, when there is few training data, increasing the value of K degrades the precision of the estimation of the distribution (Refer to Eqs. 3, 4, and 5). In this study, the values of K for SOMs are searched by experiments, and then the value of K for each SOM is fixed.

3. TEXTURE SEGMENTATION USING SOM

For texture segmentation we propose to use likelihood probability by SOM in multi-scale wavelet domain. The multi-level Haar wavelet transform forms a pyramid structure through all scales. Haar wavelet transforms of three levels are shown in Fig. 2-(a). As can be seen, the coarse-scale coefficient w_{J-3} corresponds to four coefficients in the next finer scale and the dependency of these coefficients across scales has a quad-tree structure. The multi-scale wavelet coefficients, which analyze a common sub-region of an image, have persistence across the scale. The dependency between these coefficients can be represented as the dependency between parent and child nodes of a wavelet quad-tree [see Fig. 2-(b)]. In Fig. 2-(b), a



(a) (b)

Fig. 2. (a) The Harr wavelet transform and the quad-tree structure of the wavelet coefficients; (b) The quad-tree structure of the wavelet coefficients in a wavelet sub-band and sub-tree T_i rooted at node i and its elements.

black node represents a wavelet coefficient, and $\boldsymbol{T}_i$ is a sub-tree rooted at node i. In this paper, we will often drop the scale (J, J-1, …) and the direction (LH, HL, HH), if they are not confused. To estimate likelihood in multi-scale, the proposed system has one SOM for each wavelet scale, and each texture class.

The input vector of SOM was determined by considering the dependency of the wavelet coefficients across the scale. The input vector $\mathbf{g}_i$ of SOM is defined as

$$\mathbf{g}_i \equiv \{T_i^{HL}, T_i^{LH}, T_i^{HH}, \mathbf{I}_i\} . \quad (7)$$

Here, the $\mathbf{g}_i$ is for the region of the sub-image analyzed by sub-tree $\boldsymbol{T}_i$ [see Fig. 2-(b)]. The sub-tress $\{T_i^{LH}, T_i^{LH}, T_i^{LH}\}$ rooted at node i in the three wavelet sub-band quad-trees contain the wavelet coefficients analyzing a common sub-region of an image. $\mathbf{I}_i$ is the intensity of the pixels of the sub-image analyzed by node i.

The classification of an input vector $\mathbf{g}_i$ in j'th wavelet scale is derived as follows:

$$\hat{c} = \arg\max_{c\in\{1,2,\ldots,N_c\}} f(\mathbf{g}_i \mid c) \approx \arg\max_{c\in\{1,2,\ldots,N_c\}} \frac{K/N}{\pi r_{c,K}^{j}(\mathbf{g}_i)^2} \quad (8)$$

$$\propto \arg\max_{c\in\{1,2,\ldots,N_c\}} \frac{1}{r_{c,K}^{j}(\mathbf{g}_i)^2} ,$$

where $r_{c,K}^{j}(\mathbf{g}_i)$ is a distance computed by using Eq. 6 from SOM of scale j for class c. In this study, the Euclidean distance is used as a distance measure. The multi-scale texture segments are obtained by using Eq. 8.

If the contextual information of an image is used to classify the texture for one of the nodes of the wavelet quad-tree, then the multi-scale texture segmentations have been greatly improved. The multi-scale texture segments can be obtained by using contextual information as follows.

$$\hat{c}_i = \arg\max_{c_i \in \{1,2,\ldots,N_c\}} f(\mathbf{D}_i \mid c_i), \quad (9)$$

where $\mathbf{D}_i = \{\mathbf{g}_i, \mathbf{D}_{Ni}, \mathbf{g}_{\rho(i)}, \mathbf{D}_{N\rho(i)}\}$ is an observation that considers the context of node i of the wavelet quad-tree; $\mathbf{D}_{N\rho(i)} \equiv \{\mathbf{g}_{N\rho(i),1}, \mathbf{g}_{N\rho(i),2}, \ldots, \mathbf{g}_{N\rho(i),8}\}$ is an observation on the eight neighbors of the parent node $\rho(i)$ of node i; $\mathbf{D}_{Ni} \equiv \{\mathbf{g}_{Ni,1}, \mathbf{g}_{Ni,2}, \ldots, \mathbf{g}_{Ni,8}\}$ is an observation on the eight neighbors of node i; and c_i is the class for node i of the wavelet quad-tree. If we assume that the wavelet coefficients in one scale are independent, Eq. 9 is rewritten as follows:

(a) (b)

Fig. 3. (a) Test texture images ; (b) ideal texture segmentations.

$$f(\mathbf{D}_i \mid c_i) = f(\mathbf{g}_i \mid c_i) f(\mathbf{D}_{N_i} \mid c_i) f(\mathbf{g}_{\rho(i)} \mid c_i) f(\mathbf{D}_{N_{\rho(i)}} \mid c_i) \quad (10)$$

$$\propto f(\mathbf{g}_i \mid c_i) f(\mathbf{D}_{N_i} \mid c_i) f(\mathbf{g}_{\rho(i)} \mid c_{\rho(i)}) f(\mathbf{D}_{N_{\rho(i)}} \mid c_{\rho(i)}) ,$$

where $f(\mathbf{D}_{N_i} \mid c_i) = \prod_{l=1}^{8} f(\mathbf{g}_{N_i,l} \mid c_i) \propto \prod_{l=1}^{8} \frac{1}{r_{c_i,K}(\mathbf{g}_{N_i,l})^2}$, $f(\mathbf{D}_{N_{\rho(i)}} \mid c_{\rho(i)}) = \prod_{l=1}^{8} f(\mathbf{g}_{N_{\rho(i)},l} \mid c_{\rho(i)}) \propto \prod_{l=1}^{8} \frac{1}{r_{c_{\rho(i)},K}(\mathbf{g}_{N_{\rho(i)},l})^2}$.

Texture segmentations in multi-scale have compromises between reliability and minuteness according to scale. Segmentation in coarse scale is accurate for large, homogeneous regions but poor along the boundaries between regions. Segmentation in fine scale is accurate near the boundaries between regions but has poor classification reliability due to the paucity of statistical information. A study modeled the distribution of the multi-scale wavelet coefficients in HMT, and performed texture segmentations by using HMT in multi-scale wavelet domain, and showed that image segmentation in the finest scale is improved by using the HMTseg algorithm that fuses image segmentations in multi-scale by HMT [4]. Therefore, to fuse multi-scale texture segments, we use the HMTseg algorithm. The HMTseg algorithm fuses the multi-scale segmentation one by one from coarse to fine scale and then finally improves texture segmentation at the finest scale.

4. EXPERIMENTS AND RESULTS

In this paper, 16 Brodatz textures are used in the experiments. From each 512×512 Brodatz texture image, we randomly picked ten (overlapping) 64×64 blocks. Then the multi-level wavelet transform (3 levels) of those blocks were used as training data. Neuron nodes in the competitive layer of SOM are arranged in a 7×7 planar square lattice. The image of scale j has $2^j \times 2^j$ pixels (If an image has 64 × 64 pixels, then the image has scale j = 6.). Therefore, when the full resolution image has 64 × 64 pixels, SOMs of scale j have $(((4^{6-j} - 4) / 3 + 1) \times 2 + ((4^{7-j} - 4) / 3 + 1))$ input nodes (Refer Eq. 7.). Euclidean distance was used to measure the distance between an input vector and the weight vector of a neuron. Weights of SOMs are updated by using the MATLAB neural network toolbox [6] with the earning rate which is decreasing 0.9 to 0.1 for 100 iterations (The neighboring distance is also decreasing maximum to 1 for 100 iterations). The parameters of HMT were estimated using EM algorithm (The threshold value 10^{-7} is used to determine the model convergence) with an intelligent parameter initialization [7]. The test images for the experiments are shown in figure 3.

A proper K in SOM will result in good segmentation performance. As shown on figure 4-(a) and 4-(b), SOM with K=5 presents good performance in scale j=6, and SOM with K=1 presents good performance in scales j=3, 4, and 5 (In figure 4, scale j =6 is finer than scale j =3.). The reason

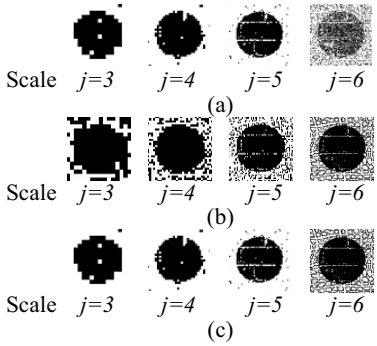


Fig. 4. Multi-scale texture segmentations by each system before applying HMTseg; (a) by SOM with K=1; (b) by SOM, K=5; (c) by SOM, K values set according to each scale (K=1, 1, 1, and 5 from left to right of figures).

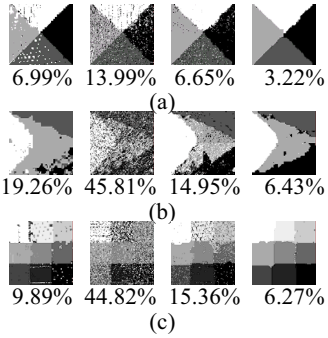


Fig. 5. Texture segmentation results and error rates by each segmentation method and HMTseg; (a) first column: by HMT and HMTseg [4]; second column: by SOM, K =1 and HMTseg; third column: by SOM, K =1,1,1, and 5 from the coarsest scale to the finest scale, and HMTseg; fourth column: by SOM, context $\boldsymbol{D}_i = \{\mathbf{g}_i, \mathbf{D}_{Ni}, \mathbf{g}_{\rho(i)}, \mathbf{D}_{N\rho(i)}\}$ and using K=1, 1, 1, and 5, and HMTseg; (b), (C) by the same method as (a).

for these results is because the amount of training data is more abundant for the fine scales (especially, for scale j=6) than for the coarse scales and segmentation for the fine scales is more sensitive to noise. Therefore, the segmentation performance by SOM with K=5 is excellent at the finest scale j=6. But, for the coarse scales which had less training data, increasing the value of K reduces the reliability of the likelihood estimation. Thus, SOMs with different values of K according to each scale have good texture segmentation performance. Texture segmentation results by SOM with different values of K according to each scale are shown in figure 4-(c) (The multi-scale segmentation results by SOM with K=1, 1, 1, and 5 according to the scale, j=3, 4, 5, and 6).

Segmentation results for other test images are in figure 5. The figure 5 shows texture segmentation results by each segmentation method and HMTseg only in the final pixel domain (scale j =6). Under the picture of each segmentation result, the error rate between the ideal segmentation (Fig. 3) and the resulting segmentation (Fig. 5) is given. The error rate is the rate of the number of misclassified pixels to the total number of pixels in an image. Let's consider figure 5-(a) and 5-(b) (the test image contains 4 texture classes). SOMs with using the proposed methods perform better than HMT except for SOM that uses K =1 for all of the scales. However, figure 5-(c) indicates that SOM using K values that differed according to each scale did not perform as well as HMT. It seems that 7×7-sized SOM is more sensitive to noise than the Gaussian mixture model at the finest scale (scale j =6) if the number of texture classes increases. However, when SOM uses contexts to reduce the influence of noise, it performs better than HMT. Comparing the second and third columns of figure 5 reveals that segmentation by SOM that uses proper values of K differing according to each scale is much better than SOM that uses K=1 for all of the scales

5. CONCLUSION

We proposed a novel method of supervised texture segmentation using SOM in a multi-scale wavelet domain and a multi-scale Bayesian image segmentation technique. A method for obtaining the likelihood probability of observations from SOM was proposed. To estimate the likelihood more exactly from SOM, the values of K for each SOM were searched by experiments. The performance of texture segmentation improved when K was set to different values for SOMs according to each scale or according to the amount of training data. We also propose a likelihood estimation method that used contextual information. This method improved texture segmentation. The results of texture segmentation by the proposed methods are much better than those by HMT and HMTseg.

6. REFERENCES

[1] M.A. Georgeson, "Spatial Fourier Analysis and Human Vison," Chapter 2, in *Tutorial Essays in Psychology, A Guide to Recent Advance,* N.S. Sutherland (ed.), Vol 2, Lawrence Earlbaum Associate, Hillsdale, N.J., 1979.

[2] C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), *The Handbook of Pattern Recognition and Computer Vision. 2nd Edition*, World Scientific Publishing Co. pp. 207-248, 1998.

[3] Z. Chen, T.J. Feng, Z. Houkes, "Texture segmentation based on wavelet and Kohonen network for remotely sensed images," *IEEE International Conf. on Systems, Man, and Cybernetics*, Tokyo, Japan, pp. 816-821, 1999.

[4] Hyeokho Choi, Richard G. Baraniuk, "Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models," *IEEE Transaction on image processong*, vol. 10, No. 9, September 2001.

[5] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification. 2nd Edition*, A Wiley Interscience publication pp.161-192, pp. 576-582, 2001

[6] Howard Demuth, Mark Beale, *Neural Network Toolbox For Use with MATLAB. User's Guide Version 4*, The MathWorks, Inc., pp. 277-298.

[7] Guoliang Fan, Xiang-Gen Xia, "Improved Hidden Markov Models in the Wavelet-Domain," *IEEE Transaction on signal processong*, vol. 49, No. 1, January 2001.

[8] T. Kohonen, "The Self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464-1480, 1990.