Texture Image Segmentation using Reduced Gabor Filter Set and Mean Shift Clustering

Guodong Guo, Stan Z. Li, Kap Luk Chan and Hong Pan

School of EEE, Nanyang Technological University
Nanyang Avenue, Singapore 639798
email: egdguo@ntu.edu.sg

ABSTRACT

This paper presents an unsupervised texture image segmentation algorithm using reduced Gabor filter set and mean shift clustering. Two criteria are proposed in order to construct a feature space of reduced dimensions for texture image segmentation, based on selected Gabor filter subset from a predefined Gabor filter set. An unsupervised clustering algorithm using the mean shift clustering method is then applied to the reduced feature space to obtain the number of clusters, i.e. the number of texture regions. A simple Euclidean distance classification scheme is used to group the pixels into corresponding texture regions. Experiments on a mixture of Brodatz textures or mosaic of textures generated by random field model show the proposed algorithm of using the reduced Gabor filter set and mean shift clustering gives satisfactory results in terms of the number of regions and region shapes.

Keywords: *unspuervised image segmentation, textures, Gabor filter selection, feature selection, mean shift clustering.*

1. Introduction

Texture image segmentation involves identifying regions with "uniform" textures in a given image. It is a difficult task in several aspects. First, it is difficult to define the properties that can be used effectively to characterize all textures. Second, it is difficult to find a set of properties that can be used to distinguish textures found in a given image. Third, it is difficult to determine how many textures there is in a given image. Forth, it is difficult to determine the texture region boundary accurately because texture is a region property rather than a point property. In this paper, we address the first three issues while the fourth issue can be addressed by a better image segmentation routine.

A large number of techniques for analyzing image texture have been proposed in the past two decades [12] [11] [18]. Recent work on texture analysis mainly focuses on the following two well-established area: statistical approach and Filtering approach. In the first approach, it involves statistical modeling which characterizes textures as arising from probability distributions on random fields. These modeling approaches involve only a small number of parameters, thus provide a concise representation for textures. Markov random fields (MRF) [5] [15] are the most popular among these models. However, these models are usually of limited forms, thus suffer from the lack of expressive power to large scale behavior. The second approach can be said to be inspired by the multi-channel filtering mechanism discovered and generally accepted in neurophysiology. This mechanism suggests that the visual system decomposes the retinal image into a set of sub-bands. This can be realized by filtering the image with a bank of linear filters followed by some nonlinear procedures. The filtering theory developed along this direction includes the Gabor filters [6] and wavelet pyramids [16]. The filtering methods have demonstrated excellent performance in texture classification and segmentation [14]. A thorough review of texture segmentation using Gabor filter banks can be found in [1].

Significant features in the image correspond to high density regions in the feature space. A simple nonparametric technique to estimate the density gradient based on the mean shift principle was proposed in [9], and generalized by [3]. The mean shift algorithm has been used recently in color image segmentation [4].The mean shift approach is adopted for texture feature analysis here.

In this paper, we choose the Gabor filters as the feature extractor [1] [7] [8] [13] [14] [17]. A set of frequency and orientation selective filters are used to filter the input image, resulting in a set of filtered images. At least one feature can be extracted from a filtered image. The texture feature dimensions are then usually large. In order to reduce the feature dimensions, we propose two criteria to select the features. Since the selection is based on filtered images, it is in a sense a filter selection. The first selection criterion is based on the energy contained in each filtered image and second selection criterion is to reduce the redundancy between the feature images. After the texture features are determined for segmentation, mean shift clustering is applied to determine the number of texture regions in the given image and their characteristic feature vectors from the cluster centers. An overview of our texture segmentation algorithm is shown in Figure 1.

The paper is organized as follows. In Section 2, we describe the texture feature extraction by Gabor filters. We propose the two criteria for filter selection and feature dimension reduction. The mean shift algorithm applied to feature space clustering is described in Section 3. Experimental results on unsupervised texture segmentation are shown and discussed

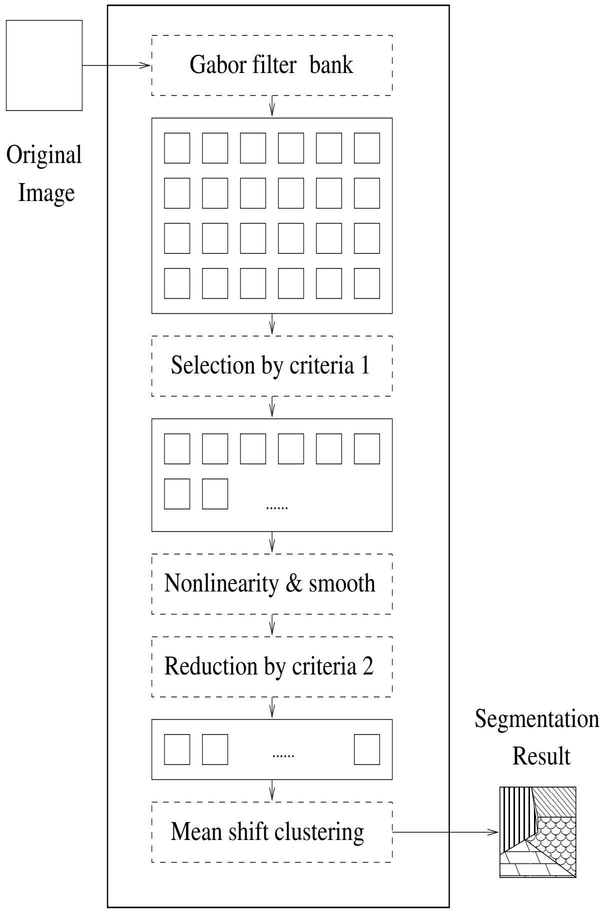


Figure 1. The framework of proposed texture segmentation algorithm. The dash blocks represent operations while the solid blocks contain the filtered or feature images.

in Section 4. Finally, Section 5 gives the conclusion.

2. Texture Feature Extraction

2.1. The Gabor Filter Set

A two dimensional Gabor function $g(x, y)$ and its Fourier transform $G(u, v)$ can be written as:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \quad (1)$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} \quad (2)$$

where W is the frequency of a sinusoidal plane wave along the x-axis, and σ_x and σ_y are the space constants of the Gaussian envelope along the x and y axes, respectively. While $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$. Filtering a signal with this basis provides a localized frequency characterization. Filters with arbitrary orientations can be obtained via a rigid rotation of the x-y coordinate system:

$$g^{'}(x,y) = g(x^{'}, y^{'}), \quad (3)$$

where

$$x^{'} = x\cos\theta + y\sin\theta, \quad y^{'} = -x\sin\theta + y\cos\theta, \quad (4)$$

where θ is the rotation angle.

A Gabor filter bank is usually designed to cover all available the frequency spectrum [14] [17]. The Gabor filter set is constructed such that the half-peak magnitude of the filter in the frequency spectrum touch each other. This results in the following formula to compute the filter parameters σ_u and σ_v:

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{S-1}}, \quad W = a^m U_l, \quad (5)$$

$$\sigma_u = \frac{(a-1)W}{(a+1)\sqrt{2\ln 2}}, \quad (6)$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right)\left[W - \frac{(2\ln 2)\sigma_u^2}{W}\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{W^2}\right]^{-\frac{1}{2}} \quad (7)$$

where U_l and U_h denote the lower and upper center frequencies of interest. $m \in \{0, 1, \ldots, S-1\}$ and $n \in \{0, 1, \ldots, K-1\}$, are the index of scale and orientation, respectively. K is the number of orientations and S is the number of scales.

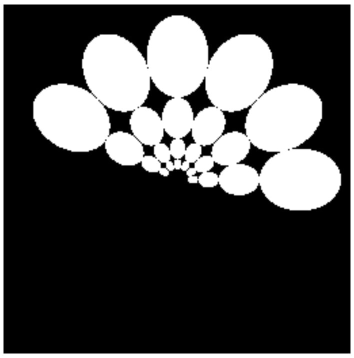


Figure 2. The filter set in the spatial-frequency domain. There is a total of 24 Gabor filters shown at the half-peak magnitude.

In our texture segmentation experiments, we select $U_h = \sqrt{2}/4$, and $U_l = \sqrt{2}/32$, and hence a total of four scales, $S = 4$. Six orientations, $K = 6$, are defined within each scale. The half-peak supports of the Gabor filter bank are shown in Figure 2. The differences in the strength of the responses of different texture regions is the key to the multi-channel approach to texture analysis.

2.2. Filter Selection and Feature Dimension Reduction

Using the Gabor filter set defined here for image filtering results in 24 filtered images for an input texture image. The dimensions of texture features is at least 24. Since the key to successful classification and segmentation is the different responses to the filters due to different textures. Some filtered images may show similar response to different textures because the textures may share the same spatial frequency properties over these bands and orientations. Hence, not all the

24 filtered images are always useful. Reducing the feature dimensions by using only a subset of the filtered images can reduce the computational burden at later stages.

We propose two criteria for filter selection, or more precisely filtered image selection. One is based on the power of each filter; the other is based on the redundancy reduction between the feature images.

2.2.1 Filter Selection by Filtered Image Energy

Let $f_i(x, y)$ be the ith filtered image and $F_i(u, v)$ be its discrete Fourier transform(DFT). Denote the energy of each filtered image as E_i, $i = 1, 2, \ldots, N$, N is the number of the filters, here $N = 24$. The sum of the energy of all the filtered images is E. Due to the overlaps between the Gabor filters in the filter bank, we have the following approximation:

$$E \approx \sum_{i=1}^{N} E_i, \tag{8}$$

where

$$E_i = \sum_{x,y} [f_i(x, y)]^2 = \sum_{u,v} |F_i(u, v)|^2, \tag{9}$$

We sort the filters by the energy of the corresponding filtered images, such that $E_{k_1} < E_{k_2} < \ldots < E_{k_N}$.

How many filters should be discarded ? We argue that the sum of energy of the discarded filtered images should be small enough so that the reserved filtered images contain most of the energy of E. A reconstruction from the reduced filtered images should not be visually different from original image. Let's define

$$R \approx \frac{\sum_{j \in \Lambda} E_j}{E}, \tag{10}$$

where Λ is a subset of the original filter set Ω, it consists of the first c filters which will be discarded. We discard as many filters as needed as long as R is no more that 5%. The reserved or selected filtered images correspond to the filters in the set $(\Omega \setminus \Lambda)$.

Jain and Farrokhnia [14] described a filter selection scheme based on reconstruction of the input image. However, that scheme has much computation burden, so they take an approximation scheme whose result is similar to that obtained by our first criterion but with different interpretation. Even though, the feature dimensions is often still very high (above 13) for subsequent segmentation.

2.2.2 Redundancy Reduction Between Feature Images

The second criterion aims to reduce the feature dimensions further by the distances between any two feature images. If the distance is small, the two feature images are similar. We just choose one with higher energy and discard the other.

Suppose there are M filtered images selected by the first criterion, denoted by $\{1, 2, \ldots, M\}$ for simplicity. These images undergo firstly a similar nonlinear processing as in [14] :

$$\psi(t) = tanh(\alpha t) = \frac{1 - e^{-2\alpha t}}{1 + e^{-2\alpha t}} \tag{11}$$

where α is a constant (α = 0.25 in our work). Then, the feature image $e_k(x, y)$ corresponding to filtered image $f_k(x, y)$ is given by:

$$e_k(x, y) = \sum_{a,b \in A_{xy}} |\psi(f_k(a, b)) \cdot s(a, b)| \tag{12}$$

and

$$s(a, b) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(a - x)^2 + (b - y)^2}{2\sigma^2}\right) \tag{13}$$

where A_{xy} is an $A \times A$ window centered at the pixel (x, y), and $s(a, b)$ is the Gaussian centered at (x, y) for smoothing.

The process of feature dimension reduction consists of the following steps:

1. Normalize the M feature images into the same range, such as $[0, 1]$ without loss of generality. Suppose the input image has n data points, then we calculate the average distance (or difference):

$$D(i, j) = \frac{1}{n} \sum_{x,y} |\tilde{e}_i(x, y) - \tilde{e}_j(x, y)|^2 \tag{14}$$

where $\tilde{e}_i$ is the normalized value of e_i, and $i, j \in \{1, 2, \ldots, M\}$.

So we obtain a table or matrix of size $M \times M$. We call it the Normalized Distance Matrix (NDM), which has "0" elements along its main diagonal, i.e., $D(i, i) = 0$.

2. Calculate the minimal distance between i and any other $j \in \{1, 2, \ldots, M\}$, $j \neq i$, from the NDM:

$$\tilde{D}_i = \min_{j \neq i} D(i, j) \tag{15}$$

and also record the index i^* corresponding to the minimal distance.

$$i^* = \arg\min_{j \neq i} D(i, j) \tag{16}$$

3. Find the minimal distance of i^* to all the other feature images, denoted by $\tilde{D}_{i^*}$, corresponding to index i^{**}:

$$\tilde{D}_{i^*} = \min_{j \neq i^*} D(i^*, j) \tag{17}$$

$$i^{**} = \arg\min_{j \neq i^*} D(i^*, j) \tag{18}$$

4. Sort the distance $\tilde{D}_i$, $i \in \{1, 2, \ldots, M\}$ in ascending order, and do comparison:

$$\tilde{D}_i \stackrel{?}{=} \tilde{D}_{i^*}, \qquad i \stackrel{?}{=} i^{**}, \tag{19}$$

where "$\stackrel{?}{=}$" means *whether equal or not*. From these comparisons, we divide all the feature images into two sets, S_1 and S_2:

If $\tilde{D}_i = \tilde{D}_{i^*}$, and $i = i^{**}$, the feature image index i and i^* will be put into the set S_1. We iteratively select all the pairs of i and i^* into set S_1, and put all other index into set S_2.

5. Obtain the maximal distance in set S_1, denoted by $\tilde{\tilde{D}}$:

$$\tilde{\tilde{D}} = \max_{i \in S_1} \tilde{D}_i = \max_{i^* \in S_1} \tilde{D}(i^*) \tag{20}$$

which is called the *maximin* distance, because $\tilde{D}_i$ (or $\tilde{D}_{i^*}$) is the minimal distance in Eq. 15 (or 17). The value $\tilde{\tilde{D}}$ will be used as the threshold for later use.

6. Compare the energy difference from each pair of i, i^*:

$$p = \sum_{x,y} |e_i(x, y)|^2 - \sum_{x,y} |e_{i^*}(x, y)|^2 \tag{21}$$

and make decision:

$$\begin{cases} Reserve \;\; i, discard \;\; i^* \;\; in \;\; S_1, & \text{if } p > 0; \\ Reserve \;\; i^*, discard \;\; i \;\; in \;\; S_1, & \text{if } p < 0. \end{cases} \tag{22}$$

After above processing, the size of set S_1 will become half of its original.

7. Do another decision for all the elements in set S_2:

$$put \;\; l \;\; into \;\; S_1, \;\; if \;\; \tilde{D}_l > \tilde{\bar{D}} \;\; and \;\; l \in S_2 \tag{23}$$

8. Finally, all the elements in S_1 are the index of the final selected feature images.

Usually, the final feature dimension (6-7 in our experiments for 4 or 5 texture classes) is much smaller than 24, the size of the original filter bank, and also M (above 13) of the selected features by criterion 1. This will largely reduce the computation burden for later feature space analysis.

3. Segmentation using Mean Shift Clustering

The idea of *mean shift* is to shift iteratively a fixed size window to the average of the data points within it. It estimates the gradient of a density function without assuming any *a priori* distribution structure of the data.

Assume the reduced feature dimensions is d and the number of pixels in an image is n, so we have a data set, denoted by $\{\mathbf{x}_i\}, i = 1, 2, \ldots, n$. Then, the *multivariate kernel density estimate* obtained with kernel $K(\mathbf{x})$ and window radius r, computed at point $\mathbf{x}$ is defined as [19]:

$$\hat{f}(\mathbf{x}) = \frac{1}{nr^d}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{r}\right) \tag{24}$$

The optimal kernel yielding minimal mean integrated square error is the Epanechnikov kernel:

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2}(d+2)(1-\mathbf{x}^T\mathbf{x})/v_d & \text{if } \mathbf{x}^T\mathbf{x} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where v_d is the volume of the unit d-dimensional sphere [19].

The use of a differentiable kernel allows to define the estimate of the density gradient as the gradient of the kernel density estimate,

$$\hat{\nabla} f(\mathbf{x}) \equiv \nabla \hat{f}(\mathbf{x}) = \frac{1}{nr^d}\sum_{i=1}^{n} \nabla K\left(\frac{\mathbf{x}-\mathbf{x}_i}{r}\right) \tag{26}$$

The density gradient estimate (Eq. 26) with respect to the Epanechnikov kernel (Eq. 25) is

$$\hat{\nabla} f(\mathbf{x}) = \frac{1}{n(r^d v_d)}\frac{d+2}{r^2}\sum_{\mathbf{x}_i \in S_r(\mathbf{x})}[\mathbf{x}_i - \mathbf{x}] \tag{27}$$

$$= \frac{n_{\mathbf{x}}}{n(r^d v_d)}\frac{d+2}{r^2}\left(\frac{1}{n_{\mathbf{x}}}\sum_{\mathbf{x}_i \in S_r(\mathbf{x})}[\mathbf{x}_i - \mathbf{x}]\right) \tag{28}$$

where the hypersphere $S_r(\mathbf{x})$ with radius r, has volume $r^d v_d$, centered on $\mathbf{x}$, and contains $n_{\mathbf{x}}$ data points. Define

$$M_r(\mathbf{x}) \equiv \frac{1}{n_{\mathbf{x}}}\sum_{\mathbf{x}_i \in S_r(\mathbf{x})}[\mathbf{x}_i - \mathbf{x}] = \frac{1}{n_{\mathbf{x}}}\sum_{\mathbf{x}_i \in S_r(\mathbf{x})}\mathbf{x}_i - \mathbf{x} \tag{29}$$

which is called the *sample mean shift* [4]. If we use a kernel different from the Epanechnikov kernel, we will have a weighted mean computation in Eq. (29).

The quantity $\frac{n_{\mathbf{x}}}{n(r^d v_d)}$ is the kernel density estimate $\hat{f}(\mathbf{x})$ computed with the hypersphere $S_r(\mathbf{x})$, we can write Eq. (27) as

$$\hat{\nabla} f(\mathbf{x}) = \hat{f}(\mathbf{x})\frac{d+2}{r^2}M_r(\mathbf{x}), \tag{30}$$

which yields

$$M_r(\mathbf{x}) = \frac{r^2}{d+2}\frac{\hat{\nabla} f(\mathbf{x})}{\hat{f}(\mathbf{x})} \tag{31}$$

The above equation was first derived in [10]. It shows that an estimate of the normalized gradient can be obtained by computing the sample mean shift in a uniform kernel centered on $\mathbf{x}$.

The mean shift vector always points to the direction of the maximal increase of the density, it can define a path leading to a local density maximum, *i.e.*, the mode of the density.

The iterative procedure of *mean shift* is as follow: (i) Compute the mean shift vector $M_r(\mathbf{x})$; and (ii) Move the window $S_r(\mathbf{x})$ by $M_r(\mathbf{x})$. The two steps are iterated until no more mean shifts occurs within an iteration.

The above mean shift procedure can move gradually towards the mode of a data set from any starting point. One can take all the data points as the starting points, and then move simultaneously to convergence, *i.e.*, stopping at the modes. This is called "blurring" in [3]; Alternatively, selects randomly part of the data points as the starting points. In our experiments, we choose the latter approach as it is not necessary to use all data points as starting points and thus reduce the amount of computation.

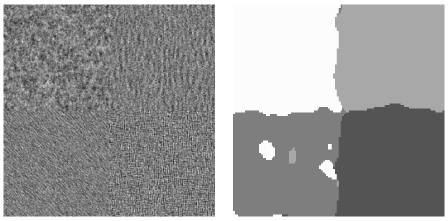
Given a search radius, and the starting points, the mean shift procedure can take place iteratively until convergence. The number of modes and the position of each mode can be obtained simultaneously. Then we calculate the Euclidean distance of each sample to all the detected modes, and classify it to the nearest.

4. Experiments on Texture Segmentation

4.1. Experimental Results

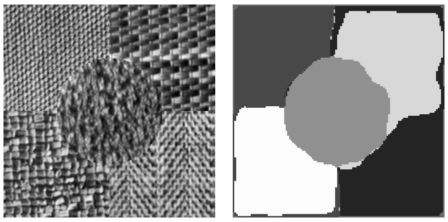
We applied our filter selection and dimension reduction criteria and mean shift clustering to two representative texture images. These images are created by collating subimages of natural as well as artificial textures. We started with 24 Gabor filters in each case. We then use our two criteria to reduce the feature dimensions. The selected feature images are used as input to the mean shift clustering procedure with 6% of the total image points randomly selected as starting points in the feature space. The same percentage is used in all experiments. The segmentation results are displayed as gray-level images, where regions belonging to different categories are illustrated as different gray levels.

Figure 3a shows a 256×256 image (GMRF-4) containing four textures generated by the Gaussian Markov random field (GMRF) model . These textures are generated using noncausal finite lattice GMRFs and cannot be discriminated on the basis of their mean gray value. The number of filters selected by our two criteria is 7. The radius of the mean shift



(a) (b)

Figure 3. (a) A 256×256 image (GMRF-4) containing four Gaussian Markov random field textures; (b) segmentation obtained by mean shift procedure with radius r=0.4, using a total of 7 Gabor filters.



(a) (b)

Figure 4. (a) A 256×256 image (Nat-5) containing five textures (D77, D55, D84, D17 and D24) from the Brodatz album; (b) five modes segmentation obtained by mean shift procedure with radius r=0.4, using a total of 6 Gabor filters.

search window is set to be 0.4 for normalize features in the range $[0, 1]$. The segmentation result is shown in Figure 3b. 4 modes, i.e. 4 textures were correctly found and segmented.

Figure 4a shows another 256×256 image (Nat-5) containing textures $D77$, $D55$, $D84$, $D17$ and $D24$ from the Brodatz album [2]. Only 6 Gabor filters are selected. The radius of the mean shift search window is also 0.4. Five modes were correctly found and the segmentation of this image is shown in Fig. 4b.

4.2. Comments on the Performance

The two classical texture images used in our experiments have been used by Jain and Farrokhnia [14] before. Here, we compare our approach with theirs. For the first image, they selected 11 Gabor filters, while we just select 7 filters. The number of clusters detected by their modified Hubert index is 4. The mean shift clustering also gives 4 modes. Our segmentation result is visually similar to theirs without the use of spatial coordinates x, y as two additional features. Better result can be obtained if the contextual information in the image space is involved.

For the second image, Jain and Farrokhnia [14] selected 13 Gabor filters, while we just select 6 filters by our criteria. Their modified Hubert index can not detect the 5 clusters, while our mean shift program can deliver 5 modes correctly. The segmentation result is comparable to theirs. We do not use the pixel coordinate x, y as additional features.

The lack of appropriate quantitative measures for the goodness of a segmentation makes it very difficult to evaluate and compare different texture segmentation algorithms [14]. So we do not make quantitative comparison. Our emphases are mainly on the feature reduction criteria and mean shift clustering.

In summary, the experiments on texture segmentation show that both of our filter selection criteria and mean shift clustering gives effective and efficient unsupervised texture segmentation.

5. Conclusion and Discussion

We have presented a texture segmentation algorithm that uses a reduced set of Gabor filtered images. We showed that the two criteria for filter selection can largely reduce the feature dimensions without leading to any wrong segmentation of the textures in the images. The mean shift method can accurately determine the number of textures in a given image and leads to successful segmentation. The number of modes and the value of each mode can be obtained simultaneously and automatically. The segmentation results are visually satisfactory. The segmentation results can be further improved if more sophisticated segmentation routine is used after feature space clustering.

References

[1] A. C. Bovik, M. Clark and W. S. Geisler, Multichannel texture analysis using localized spatial filters, *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 12, 55-73, Jan. 1990.

[2] P. Brodatz, *Textures: A Photographic Album for Artists & Designers*. New York: Dover, 1966.

[3] Y. Cheng, Mean shift, mode seeking, and clustering, *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 17, 790-799, 1995.

[4] D. Comaniciu and P. Meer, Robust analysis of feature space: Color image segmentation, in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 750-755, 1997.

[5] G. R. Cross and A. K. Jain, Markov random field texture models. *IEEE, PAMI*, 5:25-39, 1983.

[6] J, Daugman, Uncertainty relation for resolution in space, spatial frequency and orientation optimized by tow-dimensional visual cortical filters, *Journal of Optical Society of America*, vol. A, 1160-1169, 1985.

[7] D. F. Dunn, W. E. Higgins and J. Wakeley, Texture segmentation using 2-d Gabor elementary functions, *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 16, Feb., 1994.

[8] D. F. Dunn and W. E. Higgins, Optimal Gabor filters for texture segmentation, *IEEE trans. on Image Processing*, no. 4, 947-064, 1995.

[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second Ed., Boston: Academic Press, 1990.

[10] K. Fukunaga, L. D. Hosteler, The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE trans. Info. Theory*, vol. IT-21, 32-40, 1975.

[11] L. Van Gool, P. Dewaele and A. Oosterlinck, Texture analysis anno 1983, *Comput. Vision Graphics Image Process*. 29, 336-357, 1985.

[12] R. M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE* 67, 786-804, 1979.

[13] T. Hofmann, J. Puzicha and J. M. Buhmann, Unsupervised texture segmentation in a deterministic annealing framework, *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 20, 803-818, Aug., 1998.

[14] A. K. Jain and F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, *Pattern Recognition*, vol. 16, no. 12, 1167-1186, 1991.

[15] S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer, 1995.

[16] S. Mallat, Multiresolution approximation and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 315:69-87, 1989.

[17] B. Manjunath and W. Y. Ma, Texture features for browsing and retrieval of image data, *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 18, 837-842, Aug., 1996.

[18] T. R. Reed and J. M. H. D. Buf, A review of recent texture segmentation and feature extraction techniques, *Computer Vision Graphics and Image Processing*, vol. 3, no. 57, 359-372, 1993.

[19] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall, 1986.