Color Texture segmentation using Multichannel Filtering

Ramchandra Manthalkar and P. K. Biswas
Department of E & ECE, I. I. T. Kharagpur, PIN 721302, India
{rrm,pkb}@ece.iitkgp.ernet.in

ABSTRACT

Gabor filters are extensively used for texture segmentation because of their good joint space and spatial frequency localization. A texture segmentation method using multichannel Gabor filters for color texture images is presented. Only even symmetric Gabor filters are used which reduce the computational complexity. Principal component analysis is used to reduce the dimensionality of feature vector. The results are encouraging.

1. Introduction

The goal of texture segmentation is to partition an image into homogeneous regions and identify the boundaries, which separate regions of different textures. Many approaches are suggested in the literature. Rao and Lohse [1] found that repetitiveness, orientation and complexity are the major features, which account for most of the variability of different textures. Fourier power spectrum, second order gray level statistics, co-occurrence statistics, gray level run length statistics, fractal dimension, transform and filter banks are used to obtain texture features. Wezka et al [2] and Conners and Harlow [3], compared a number of statistical features and concluded that co-occurrence features were the best features for texture characterization. Du Buf et al [4] reported that several features had roughly the same performance. There is no particular method, which performs uniformly for all textures. A recent study by Trygve Randen and John Husoy [5], concludes that no single approach performs best or very close to the best for all images. Classification error and computational complexity are two very important issues in deciding a particular feature set. They have advised to test the Gabor filter bank, AR and co-occurrence features in case that they may perform well for the specific dataset at hand; however the co-occurrence and AR schemes do have a significant computational complexity.

Research suggested that the analysis of a stimulus by the visual system might involve a set of quasi-independent mechanisms, called channels, which could be conveniently characterized in the spatial frequency domain. The channels in the visual system were modeled by filters defined in the spatial frequency domain with characteristics based on the results of psychophysical and neurological studies. Gabor functions have been shown to be good fits to the receptive field profiles of simple cells in the striate cortex [6]. Statistical methods in the past have proven to be superior to frequency domain techniques. This was due to the lack of locality in these early frequency analysis methods. Joint spatial/spatial-frequency techniques are inherently local in nature, and have characteristics that compare favorably with those of statistical methods [7]. The major advantage of multichannel filtering approach to texture analysis is that simple statistics of gray values can be used as texture features. The main issues involved in the multichannel filtering approach are functional characterization of the different channels and the number of channels, extraction of appropriate features from the filtered images, the relationship between channels (dependent versus independent), and integration of texture features from different channels to produce segmentation. In this paper a strategy similar to that of Jain and Farrokhnia [8] is used for image segmentation. The texture segmentation scheme is shown in figure 1. Each of R, G, and B image is passed through the channels and features images are obtained. Afterwards a feature reduction scheme is employed using principal component analysis. To the best our knowledge color texture segmentation is not attempted using even symmetric Gabor filters.

2. Even symmetric Gabor filter

Based on psychophysical grounds Malik and Perona [10] provide some justification for using even symmetric filters only. Response of an even symmetric Gabor filter is given by,

$$h(x, y; u, \theta) = \exp\left(-\tfrac{1}{2}\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right) \cos(2\pi u x') \quad (1)$$

$$x' = x\cos\theta + y\sin\theta \quad (2)$$

$$y' = -x\sin q + y\cos q \qquad (3)$$

where u is the frequency of the sinusoidal wave along the direction q from the x-axis. s_x and s_y specify the Gaussian envelope along x and y axes respectively, which determine the bandwidth of the Gabor filter. The Fourier transform of equation (1) is

$$H(U,V) = A\left(\exp\left\{-\tfrac{1}{2}\left[\frac{(U-u)^2}{s_u^2}+\frac{V^2}{s_v^2}\right]\right\}+\exp\left\{-\tfrac{1}{2}\left[\frac{(U+u)^2}{s_u^2}+\frac{V^2}{s_v^2}\right]\right\}\right) \qquad (4)$$

where $s_u = \frac{1}{2ps_x}$, $s_v = \frac{1}{2ps_y}$ and $A = 2p\,s_x s_y$, Figure 2 shows one spatial domain impulse response and one frequency domain response for 0^0 degree orientation filters on a 128×128 grid. Most Gabor filters have slight response to regions of uniform luminance (or DC). This results in sensitivity to background luminance levels, which, for texture discrimination, signifies a first order difference between regions. This can be corrected by subtracting the mean pixel value of each filter from each of pixel value. In frequency domain this results in subtraction of a sinc function with the same value as the filter at the origin, thereby creating a filter with no DC response. Unfortunately, the sinc function induces a slight rippling in the filter's response, especially at low frequencies. However, the amplitude of the sinc if sufficiently low then these ripples is not detectable in the spectral plots for the filters [11]. Gabor filter is built in Fourier domain, and at $u = v = 0$, $H(u,v)$ is kept zero. This ensures that the filters do not respond to regions with constant intensity.

In addition to radial frequency and orientation, frequency and orientation bandwidths are also very important. For Gabor filter defined by equation (4), B_r and B_q, the half peak magnitude bandwidths, are given by

$$B_r = \log_2\left(\frac{u+(2\ln 2)^{0.5}s_u}{u-(2\ln 2)^{0.5}s_u}\right) \qquad (5)$$

$$B_q = 2\tan^{-1}\left(\frac{(2\ln 2)^{0.5}s_v}{u}\right) \qquad (6)$$

B_r is in octaves and B_q is in degrees.

The filter parameters are decided such that the space frequency plane is covered nearly uniformly. A class of self-similar functions, referred to as Gabor wavelets is constructed as follows [14]. Let $h(x,y)$ be the mother Gabor wavelet, then this self-similar filter set can be obtained by appropriate dilations and rotations of mother wavelet through the generating function:

$$h_{mn}(x,y) = a^{-m}h(x',y';u,q), \quad a>1,$$
$$m,n = \text{int } eger$$

$$x' = a^{-m}(x\cdot\cos q + y\cdot\sin q),$$
$$y' = a^{-m}(-x\cdot\sin q + y\cdot\cos q),$$

where $q = np/K$ and K is the total number of orientations. The scale factor a^{-m} in equation (7) ensures that the energy is independent of m.

$$E_{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|g_{mn}(x,y)\right|^2 dxdy \qquad (8)$$

This ensures that all filters in the set have the same energy. Let U_l and U_h denote the lower and upper center frequencies of interest. Let K be the number of orientations and S be the number of scales in the multi-resolution decomposition. Then the design strategy is to ensure that the half peak magnitude cross-sections of the filter responses in the frequency spectrum touch each other. This results in the following formulas for computing the filter parameters σ_u and σ_v (and thus σ_x and σ_y).

$$a = \left(U_h/U_l\right)^{\frac{1}{S-1}} \qquad (9)$$

$$s_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}} \qquad (10)$$

$$s_v = \tan\left(\frac{p}{2k}\right)\left[U_h - 2\ln 2\left(\frac{s_u^2}{U_h}\right)\right]\cdot\left[2\ln 2 - \frac{(2\ln 2)^2 s_u^2}{U_h^2}\right]^{-\frac{1}{2}} \qquad (11)$$

where $u = U_h, q = p/K \qquad m = 0,1,...S-1$

Here *m* is scale. In order to eliminate sensitivity of the filter response to absolute intensity values, the real (even) components of the 2-D Gabor filters are biased by adding a constant to make them zero mean. If an image is of size $N \times N$ pixels, the choice of radial frequencies u are $1\sqrt{2}$, $2\sqrt{2}$, $4\sqrt{2}$, $8\sqrt{2}$,$...(N/4)\sqrt{2}$. The unit is cycles per image width. In psychophysics, frequencies are expressed in cycles per degree of visual angle subtended on the eye. The frequencies in cycles per image width can be converted to cycles per degree if the width of the image in degrees of visual angle is known. Several experiments have shown that frequency bandwidth of simple cells in the visual cortex is about one octave [8]. Four orientations of $0^0, 45^0, 90^0, 135^0$ are used. The maximum spatial frequency is so chosen that the highest radial frequency fall inside the image. Thus the total number of filters in a filter set is given by $4\log_2(N/2)$. Orientation and frequency bandwidths of each filter is kept 45^0 and 1 octave, respectively. Figure 2 shows a filter in spatial and spatial frequency domain. Figure 3 shows the spatial frequency response of Gabor filters orientated at $0^0, 45^0, 90^0, 135^0$ and spatial frequencies of $4\sqrt{2}, 8\sqrt{2}, ...(64/4)\sqrt{2}$ cycles per image width (for an image of 256×256). The lower frequencies are not used because they respond to low frequencies and most textures consist of mid frequencies. Thus, the filterbank covers the total spatial frequency domain nearly uniformly.

3. Computing feature images

For computing feature image, each filtered image is passed through following nonlinear transformation as suggested in [8].

$$y(t) = \tanh(0.25t) = \frac{1 - e^{-0.5t}}{1 + e^{-0.5t}} \qquad (12)$$

The application of nonlinear transformation converts the sinusoidal transformations in the filtered images to square modulations. Average Absolute Deviation (AAD) from mean in small overlapping windows is calculated as feature. This is similar to texture energy measure proposed by Laws [13]. The feature image is computed corresponding to each filtered image as

$$f_i = \frac{1}{M^2} \sum_{(a,b) \in W_{xy}} |y(r_k(a,b)| \qquad (13)$$

where $y(.)$ is the nonlinear function and W_{xy} is a $M \times M$ window centered at the pixel location (x, y). The size of averaging window is very important parameter. Accurate edge preservation and accurate energy estimation are conflicting goals. Gaussian weighted windows are likely to result in more accurate texture boundaries. So each of the filtered image is passed through a Gaussian low pass filter suggested below.

$$h_g(n) = \frac{1}{\sqrt{2p}\, s_s} e^{\frac{-0.5n^2}{s^2_s}} \qquad (14)$$

in each dimension, where $s_s = \frac{1}{(2\sqrt{2})u}$. This s_s is chosen such that it is proportional to average intensity variations in the image. Thus, each of the filtered image is passed through a Gaussian low pass filter. This is repeated for R, G and B image to get feature images.

4. Feature reduction

Principal Component Analysis is used for feature reduction. For R, G, B image, after getting feature images, a vector is formed for each pixel of original image called feature vector which consists the corresponding pixel value of pixel in filtered images in all R, G and B planes respectively. The components, which give variations, more that eighty five percent in data are chosen as significant features. This is done for all three images. The features selected for R, G, B (independently) are concatenated together to form feature vector for each pixel.

5. Features for segmentation

Let there be t_n textures present in image. If the texture features are capable of discriminating these categories then patterns belonging to each category will form a cluster in feature space. This cluster should be compact and isolated from other texture categories. Here fuzzy c-means clustering algorithm is used for segmentation. For each pixel in original image a feature vector is formed by taking the value of the corresponding pixel in each feature image. This feature set is given to the clustering algorithm. The data is normalized to zero mean and constant variance for better performance of clustering algorithm. The number of different textures in the image is explicitly provided to the algorithm.

6. Experimental results

The texture images are created by collaging subimages of Vistex texture database. Before collaging an image for texture segmentation, normalization for intensity and contrast is done. The experiments are carried out by keeping the intensity at 128 and contrast at 60 for 256 gray level original images. That is $I_{\text{int}} = 128$, $C = 60$. Figure 4 shows the results of segmentation for 2, 3 and 4 texture images. Table 1 gives the percentage mis classification for test images.

7. Conclusion

Color texture segmentation is achieved using multichannel filtering (even symmetric Gabor filters are used for multichannel filtering) and results are encouraging.

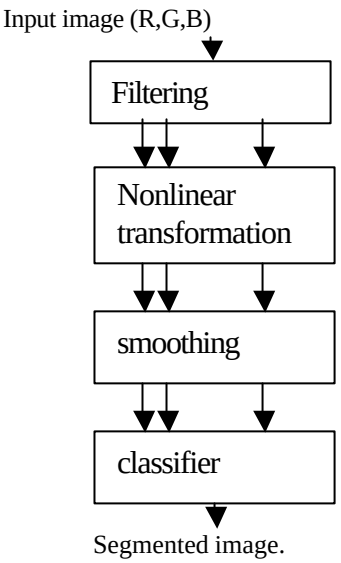

Figure 1: Scheme for the experiment

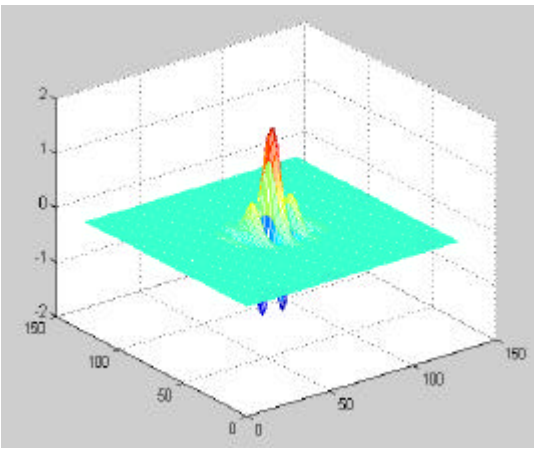

Figure 2(a)): spatial impulse response for u=5.54 c/I, (c/iw is cycles per image width).

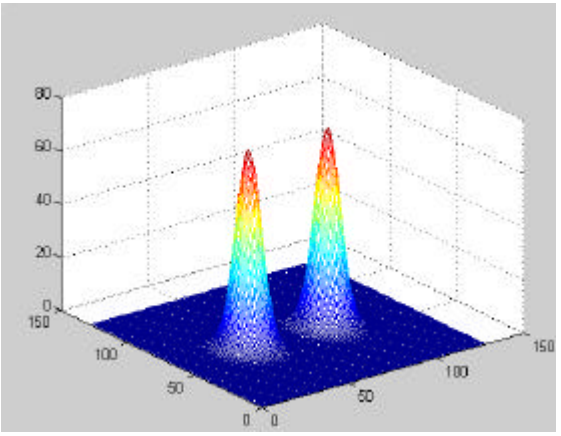

Figure 2(b) Frequency Domain representation for u=45.25 c/iw

Figure 2: Gabor Filter in spatial and spatial frequency domain.

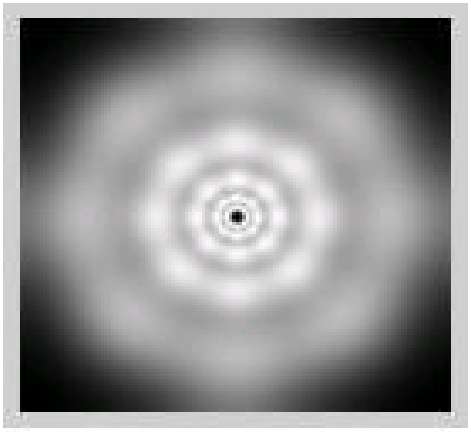

Figure 3: The spatial frequency coverage of Gabor Filter bank

Table 1: Segmentation results

Texture collage	Percentage misclassification
flowers4, Food10	**1.28**
fabric13, grass02, water05	**5.04**
fabric13, grass02, water05, leaves04	**2.35**
Clouds01, grass02, water05, leaves04	**4.68**

REFERENCES

[1] A. Ravishankar Rao, Gerald L. Lohse, " Identifying high level features of texture perception", Computer Vision, Graphics, and Image Processing, Vol. 55, no. 1, pp. 218-233, 1993.

[2] J. S. Wezka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification, " IEEE trans. On systems , man and cybernetics, Vol. 86, pp. 376-380, 1976.

[3] R. W. Conners, C. A. Harlow, "A theoretical comparison of texture algorithms," IEEE trans. On pattern analysis and machine intelligence, Vol. 2, pp. 204-221, 1980.

[4] J. M. H. Du Buf, M. Kardan, M. Spann, "Texture feature performance for image segmentation," Pattern Recognition, Vol. 23(3/4), pp. 291-309, 1990.

[5] Trygve Randen, Hakon Husoy, "Filtering for texture classification: a comparative study," IEEE trans. On pattern analysis and machine intelligence, Vol. 21, no. 4, pp. 291-310, 1999.

[6] S. Marcelja, "Mathematical description of the responses of simple cortical cells, " J. Opt. Soc. Am., Vol. 70, pp. 1297-1300, April 1980.

[7] Todd R. Reed and J. M. H. Du Buf, "A review of recent texture segmentation and feature extraction techniques," Computer Vision, Graphics, and Image Processing, Vol. 57, no. 3, pp. 359-372, 1993.

[8] A. K. Jain, Farshid Farrokhnia, "Unsupervised texture segmentation using Gabor filters," Pattern Recognition, Vol. 4, no. 12, pp. 1167-1186, 1991.

[9] A. C. Bovik, M. Clark, W. S. Geisler, "Multichannel texture analysis using localized spatial filters," IEEE trans. On pattern analysis and machine intelligence, Vol. 21, no. 1, pp. 55-73, 1990.

[10] J. Malik, P. Perona, "Preattentive texture discrimination with early vision mechanisms," J. Opt. Soc. Am. A, Vol. 7, pp 923-932, 1990.

[11] M. R. Turner, "Texture discrimination by Gabor functions, "Biological Cybernetics," Vol. 55, pp. 71-82, 1986.

[12] K. I. Laws, "Textured image segmentation," Technical report USCCIPI-940, Image Processing Institute, University of South California, 1980.

[13] W. Y. Ma. B. S. Manjunath, " Edge flow: a framework of boundary detection and image segmentation", Proc. CVPR, pp. 744-49, 1997.

[14] Wei-Ying Ma, "Netra: A toolbox for navigating large image databases," Ph. D. dissertation, Electrical and Computer Engineering department, Uni. Of California Santa Barbara, June 1997.

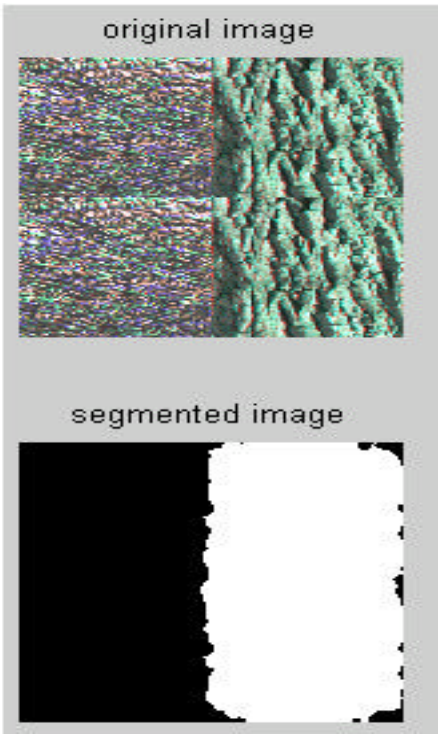

Figure 4(a): 2 texture segmentation

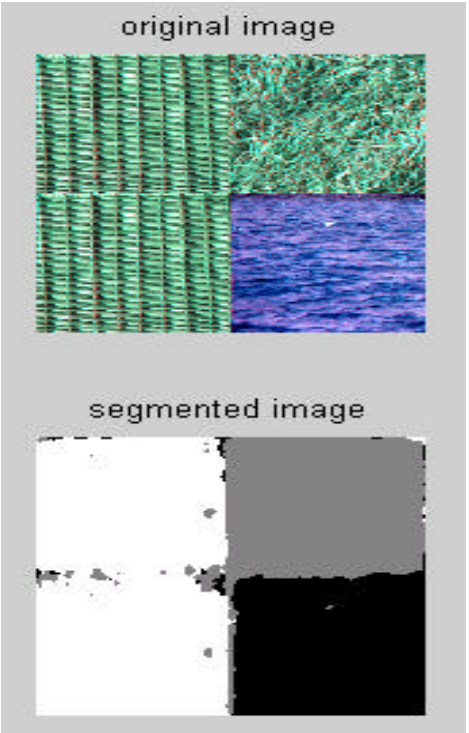

Figure 4(b) : 3 texture segmentation

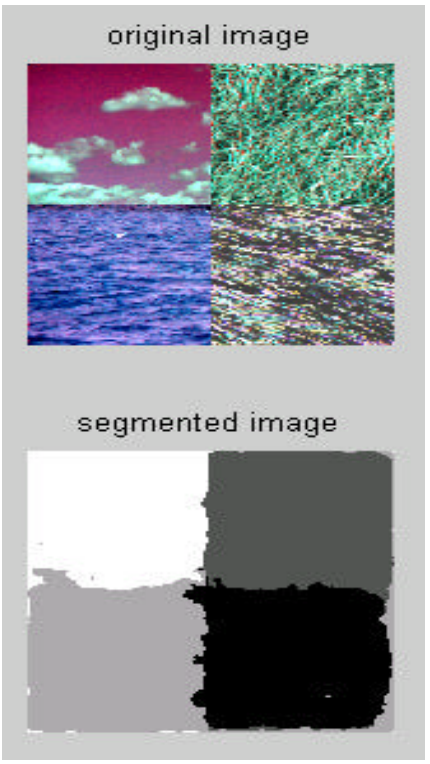

Figure 4(c): 4 texture segmentation

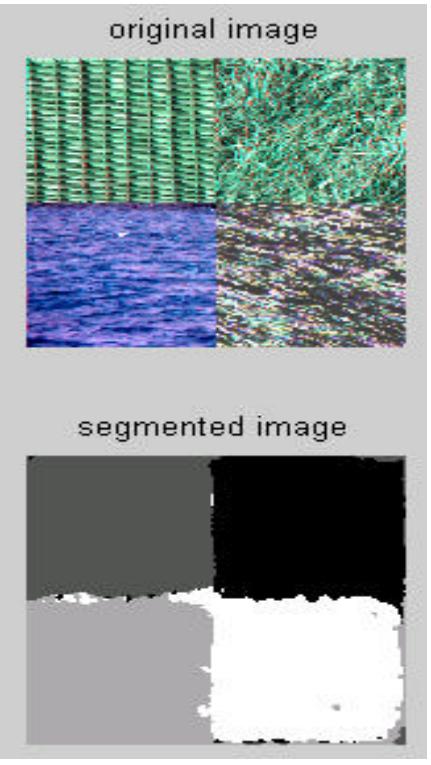

Figure 4(d): 4 texture segmentation

Figure 4: Segmentation result