Texture Edge Detection using Multi-resolution Features and SOM

Lalit Gupta and Sukhendu Das
Department of Computer Science and Engineering,
Indian Institute of Technology - Madras, Chennai - 600 036, India.
{ {lalit}@cse., sdas@ } iitm.ernet.in

Abstract

Texture boundaries or edges are useful information for segmenting a texture image. We propose a texture edge detection algorithm using a bank of 1-D multi-channel, multi-resolution filters and self organizing map (SOM). 2-D filtering smears the effect of an edge in all directions, and hence we used 1-D filtering that will only smear the edge in the direction of filtering. SOM reduces the dimension of a feature vector by producing a 1-D map which plots the similarities of data by grouping similar items together. The output of SOM is processed to obtain the texture edge map. The proposed methodology is tested on simulated as well as natural texture images and produces satisfactory results.

1. Introduction

Texture is found in most natural images and consequently the analysis of texture has played a prominent role in computer vision research in the past decades. Most of the earlier edge detection methods, are however based on the pixel-wise changes of gray level intensities [12], and are thus not appropriate for texture edge detection. Textures are formed by some patterned placement of texture elements consisting of a group of pixels. Extraction of texture boundaries in an image is one of the major issues in discriminating and segmenting the different regions in an texture image. It has several applications in remote sensing, defense surveillance, robot navigation etc. We have used 1-D multi-channel, multi-resolution filters: Gabor filter and Discrete Wavelet Transform (DWT) for filtering a texture image, that resemble our Human Visual System [8]. 2-D filtering has the disadvantage of smearing the edge in all directions, whereas 1-D filters will only process the information in a particular direction along which it is applied [13]. We exploit this advantage of 1-D processing to detect edges which is a line-based feature. We have used self-organizing map (SOM), that reduces the dimension of the feature vector by producing a 1-D dimensional map, which plots the similarities of the data by grouping similar data items together [4]. In the following we present a review of the related work.

Traditionally, the methods used for texture edge detection were based on statistical approaches. Wechsler [12] discussed the disadvantages of using statistical approaches for texture edge detection. The problem with statistical approaches is that they do not consider multi-channel, multi-resolution approach of human vision system [8]. Gabor filter and discrete wavelet transform (DWT) are two band pass filters that resemble the human visual system. Dunn et. al. [2] presents an algorithm to design Gabor filters specially tuned to segment images with bipartite textures. Results are shown mostly on simulated and a few real world samples.

Methods of texture segmentation based on DWT [11] [6] have been widely used. A method based on hierarchical wavelet decomposition technique is introduced by Salari and Ling [11], where he uses Daubechies filer to decompose the original image into three detail and one approximate sub-band images followed by k-means clustering for segmentation. Results are shown on few regular and homogeneous real-world textures.

Maxwell et al. [7] proposed an extension of the compass operator to make it applicable to texture edge detection. Results are shown on real images consists of outdoor scenes. A 2-D filtering smears an edge between two texture regions. It is thus difficult to detect sharp edges using the conventional processing methods of 2-D filters. Yegnanarayana et. al. [13] showed that 1-D Gabor filtering smears an edge only along the direction in which it is applied. Hence the edge information in the direction orthogonal to the filter is unattenuated. We exploit this property of 1-D filtering to detect edges (a 1-D property) in 2-D images. Results are shown on simulated texture images [13].

Self-organizing map (SOM) is a data visualization technique invented by Professor Teuvo Kohonen [3]. which reduces the dimension of the data through the use of self-organizing neural networks. SOM has been used for texture classification and edge-detection [9] [4].

Liu et. al. [4] proposed a method for texture edge de-

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

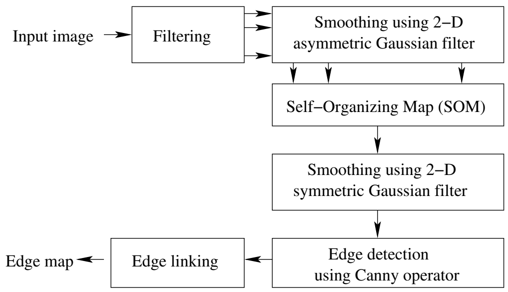

Figure 1. Stages of processing for Texture edge detection

tection using SOM. The texture features are first extracted as n-dimensional vectors by using a 2-D Gabor filter bank. They are then projected onto a 1-D feature map using SOM, where they are encoded as scalars by replacing Gabor features by the corresponding location index on feature map. Then, the predictive relationship between an encoded feature and its neighbors is computed using MLP. The variation of eight prediction errors are smoothed using a Gaussian filter to suppress local fluctuations. The final edge map is produced by Canny's edge-detection. Results are shown on the images containing five textures.

In this paper we propose a propose a technique for texture edge detection using 1-D multi-channel, multi-resolution filters: Gabor filter and DWT (Daubechies 8-tap and Haar); and the SOM. The advantages of using the combination of Gabor filter and DWT for texture classification has been discussed in [10]. The overall methodology followed for texture edge detection is shown in Fig. 1. The texture features are first extracted as n-dimensional vectors by using 1-D Gabor filter bank and DWT in both directions (horizontal and vertical parallel lines of an image). They are then smoothed using asymmetric Gaussian filter and are projected onto 1-D feature map using SOM. The feature map is smoothed with a 2-D symmetric Gaussian filter, and the final map is obtained by Canny's edge detection [1].

2. Texture edge detection

The stages of texture edge detection (Fig. 1) are described below. The outputs of intermediate stages of processing are shown for each step for the image in Fig. 2(a).

Step 1 (filtering). A 1-D Gabor filter and wavelet filter are applied along the set of all parallel lines of an image **I** in one direction (say, along all the vertical lines of the image). For a Gabor filter, the output is given as

$$H_k(x_c, y) = I(x_c, y) * g_k \tag{1}$$

where $*$ indicates the 1-D convolution operator, g_k represents 1-D Gabor filter with the parameter set $k = (\omega_k, \sigma_k)$, $I(x_c, y)$ represents the ${x_c}^{th}$ column of the image **I**, and $\mathbf{H}_k$ denotes k^{th} the filter response. For a set of four Gabor filters the corresponding filtered images obtained are $\mathbf{H}_1, ..., \mathbf{H}_4$. Fig. 2(b) and (c) show the Gabor filter responses with parameters $\sigma = 2$ and $\omega = 2.5$, along horizontal and vertical directions respectively for the input image shown in Fig. 2(a).

Similarly, image **I** is filtered using two most commonly used 1-D DWT (Daubechies 8-taps and Haar filters) along the same directions as discussed above. Let $\mathbf{H}_5, \mathbf{H}_6, \mathbf{H}_7$ and $\mathbf{H}_8$ denote the filter responses. Where $\mathbf{H}_5$ and $\mathbf{H}_6$ denote detail subband of level-1 and level-2 decomposition of Daubechies filter respectively and $\mathbf{H}_7$ and $\mathbf{H}_8$ denote detailed subband of level-1 and level-2 decomposition of Haar filter respectively. Fig. 2(d)-(e) show the images filtered using Daubechies filter along horizontal and vertical directions respectively for the input image shown in Fig. 2(a). Fig. 2(f) shows images filtered using Haar filter along horizontal direction for the input image shown in Fig. 2(a).

Step 2 (smoothing). Filtered images obtained in step 1 are smoothed with asymmetric Gaussian filter.

$$V_i(x, y) = H_i(x, y) ** L(x, y) \tag{2}$$

where, $L(x, y)$ denotes a Gaussian filter with $\sigma_x = 8$ x σ_y if images are filtered along parallel vertical lines and $\sigma_y = 8$ x σ_x if images are filtered along parallel horizontal lines $\mathbf{V}_i(x, y)$ denotes the convolved output of i^{th} image where $i = 1, ..., 8$. Fig. 3(a)-(c) show the smoothed images for the images shown in Fig. 2(c)-(e) respectively.

Step 3. Steps 1 and 2 are repeated to obtain filtered images in orthogonal direction. Let $F_i(x, y)$ denote all the filtered images, where $i = 1, ..., 16$. Thus a 16-dimensional vector $F(x, y)$ is obtained as:

$$\mathbf{F}(x, y) = [F_1(x, y), ..., F_{16}(x, y)] \tag{3}$$

Step 4 (SOM). A one dimensional feature map Γ over the vectors $\{\mathbf{F}(x, y)\}$ is generated using the Kohonen's SOM algorithm. For each pixel (x, y), the scaler index $M(x, y)$ of the reference vector closest to $\{\mathbf{F}(x, y)\}$ is assigned

$$M(x, y) = \arg\min_i ||\mathbf{F}(x, y) - \mathbf{w}_i|| \quad \text{for all } \mathbf{w}_i \in \Gamma \tag{4}$$

In this way we transform the vector image to a scalar image. Feature map **M** is smoothed with a symmetric Gaussian

$$E(x, y) = M(x, y) ** L(x, y) \tag{5}$$

where, $L(x, y)$ denotes Gaussian filter and **E** denotes a smoothed image. Fig. 4(a) shows the feature map (scalar image) obtained from the filtered images of the image in

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

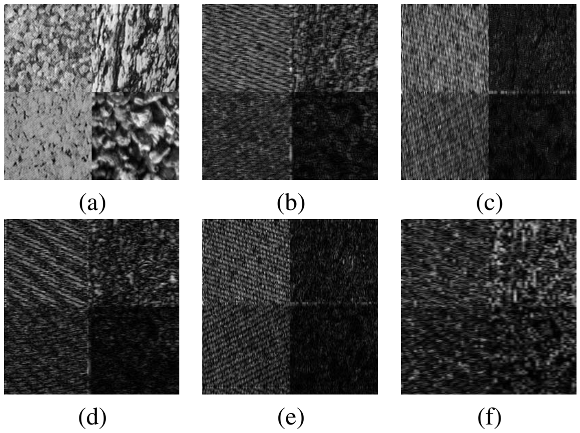

Figure 2. Filter responses for the input image shown in (a) is filtered using: (b) Gabor filter along horizontal direction; (c) Gabor filter along vertical direction; (d) Daubechies filter along horizontal direction; (e) Daubechies filter along vertical direction; (f) Haar filter along horizontal direction.

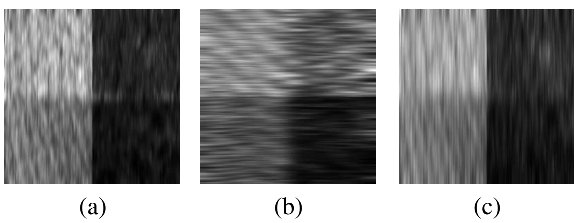

Figure 3. (a)-(c) images obtained by smoothing images in Fig. 2(c)-(e) using asymmetric Gaussian filter.

Fig. 2(a). Fig. 4(b) shows the smoothed image of the image in Fig. 4(a).

Step 5 (canny edge detection). Canny's edge-detection method is applied to the smoothed feature map image E obtained in previous step to obtain the edge map.

Step 6 (edge linking). The discontinuities in the edge map obtained using canny edge detection are linked using the similar algorithm as proposed in [5]. The steps of the algorithm are as follows: **(i)** Find all the connected segments $c_1, c_2, ..., c_b, ..., c_B$ in the edge map obtained using canny edge detection. Where c_b denotes a connected edge segment and B denotes the total number of connected segments. Repeat step (ii) for $b = 1, ..., B$; **(ii)** Find the end points of the connected segment c_b. Where, end points are the points, which contains only one edge pixel among their 8-Neighbors. Connect the end points of the c_bth segment

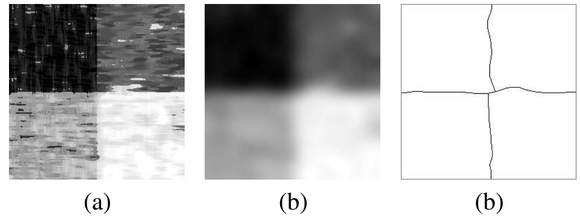

Figure 4. (a) Feature map obtained using SOM from the filtered images of the image in Fig. 2(a); (b) Smoothed feature map obtained by smoothing image in (a); (c) Final edge map obtained after applying Canny's edge dedector and edge linking on image in (b).

Table 1. Gabor filter parameters for the filter bank

Filter(k)	1	2	3	4	5	6
σ_k	1.5	2	2	2.5	2.0	1.5
ω_k	1.5	2	2.5	1.0	3.5	1.0

to the nearest edge pixel (other than pixels in c_bth segment) or the boundary pixels of an image; **(iii)** Final edge map is obtained. Fig. 4(c) shows the final edge map obtained for the input image shown in Fig. 2(a).

3. Experiments

The proposed approach has been tested on many simulated and natural texture images composed of different types of textures. Fig. 5 shows the edges obtained on six images containing 2-5 different texture regions, with edges of different orientations. Parameters for the Gabor filters are obtained using trial and error method, and are given in table 1. The width of asymmetric Gaussian filter along the major axes is taken as 1/4th the number of rows/columns. The 16-dimensional feature vector is mapped on the 1-D feature map with corresponding indices of size 256. It can be observed that proposed technique gives accurate results for the images composed of 2-5 textures, as shown in Fig. 5. Fig. 5(f)-(g) show images where it is difficult to find edges using naked eyes. The minor errors in the image shown in Fig. 5(f) are due to the set of features selected, which do not completely distinguish between different textures.

4. Conclusion

In this paper we have proposed a method based on 1-D multi-channel, multi-resolution filters and SOM for texture

0-7695-2521-0/06 $20.00

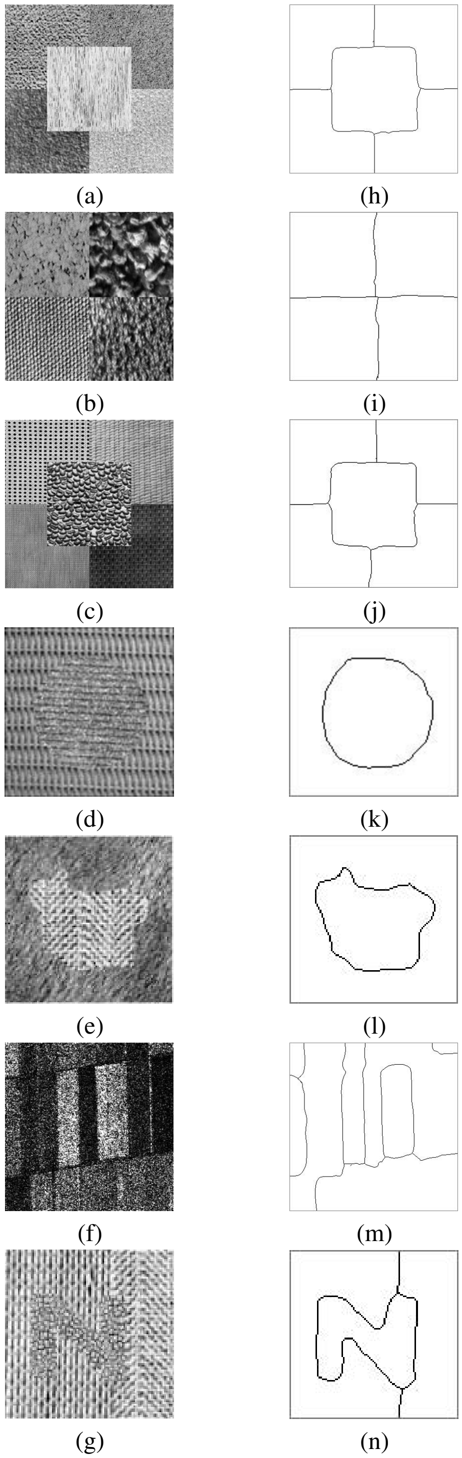

Figure 5. (a)-(g) Texture images; (h)-(n) Final edge map obtained using proposed algorithm on the images in (a)-(g) respectively.

edge detection. The method combines the advantages of both Gabor filter and DWT for texture edge detection. SOM reduces the dimension of feature vector by producing 1-D map which plots the similarities of data by grouping similar items together. Results show that texture edges are accurately detected by the proposed technique for a large variety of simulated and natural texture images. Results can further be improved by obtaining an optimal set of Gabor filter parameters. The edge information obtained by this method can be used to improve the segmentation of textures.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, June 1986.

[2] J. W. D. Dunn, W. E. Higgins. Texture segmentation using 2-d Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, 1994.

[3] T. Kohonen. The self-organizing map. In *Proc. IEEE*, volume 78, pages 1464–1480, September 1990.

[4] J.-C. Liu and G. Pok. Texture edge detection by feature encoding and predictive model. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1105–1108, March 1999.

[5] S.-M. Liu, W.-C. Lin, and C.-C. Liang. An iterative edge linking algorithm with noise removal capability. In *9th International Conference on Pattern Recognition*, volume 2, November 14-17 1988.

[6] C. Lu, P. Chung, and C. Chen. Unsupervised texture segmentation via wavelet transform. *Pattern Recognition*, 30:729–742, 1997.

[7] B. A. Maxwell and S. J. Brubaker. Texture edge detection using the compass operator. In *Proceedings British Machine Conference 2003*, September 9-11 2003.

[8] D. A. Pollen and S. F. Ronner. Visual cortical neurons as localized spatial frequency filters. *IEEE Transactions on system, Man and Cybernetics*, 13(5), 1983.

[9] P. P. Raghu, R. Poongodi, and B. Yegnanarayana. Texture classification using a two-stage neural network approach. In *International Joint Conference on Neural Network*, volume 3, pages 2195–2198, 1993.

[10] S. G. Rao, M. Puri, and S. Das. Unsupervised segmentation of texture images using a combination of Gabor and wavelet features. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 370–375, Kolkata, India, December 16-18 2004.

[11] E. Salari and Z. Ling. Texture segmentation using hierarchical wavelet decomposition. *Pattern Recognition*, 28:1819–1824, 1995.

[12] H. Wechsler. Texture analysis a survey. signal processing. 2:271–282, September 1980.

[13] B. Yegnanarayana, G. P. Kumar, and S. Das. One-dimensional Gabor filtering for texture edge detection. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 231–237, New Delhi, India, December 21-23 1998.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE