Texture-based Image Segmentation using Information measure

Anil Kumar Mishra
The Northcap University, Gurgaon-122017

Latika Singh
The Northcap University, Gurgaon-122017

Abstract - This paper describes a texture image segmentation method based on fuzzy descriptors. We have attempted to capture natural textures which are not very regular. This texture segmentation is rather a challenging task to identify textures because the repetitive texture patterns do not have any fixed size. The texture patterns are visually identifiable, but statistically they are not very regular. For texture segmentation, we derive fuzzy descriptors from the fuzzy texture features by computing the fuzzy responses of each pixel and we have implemented the Improved Mountain Clustering technique for clustering the descriptor values to segment the image.

Index Terms: Texture segmentation, Fuzzy features, Descriptors, Quotient response, Entropy, Energy

I. INTRODUCTION

The literature is ripe with so many approaches for characterizing the texture. For texture image segmentation, Bayesian approaches are proved to be effective as they can integrate both texture features and the prior knowledge about the contextual properties into a segmentation algorithm. The Markov random field (MRF) provides a convenient way for modeling prior context knowledge [3]. Many researchers have proposed multiscale wavelet representation of an observed image for Bayesian segmentation [4, 5, 6]. In Blobworld [1, 2], the region segmentation is performed by classification of joint color and texture vectors with Expectation-Maximization (EM) technique. There is hardly any attempt to characterize texture by fuzzy logic. This provides an opportunity to devise fuzzy features for describing the texture. As the texture is a region concept, it is hoped that suitable local information will represent the underlying texture. By incorporating neighborhood information in the fuzzification function, it is possible to represent texture by means of fuzzy logic techniques. At the same time, it is also required to compare the performance of fuzzy features with any of the existing textural features. The self-similarity of repetitive patterns existing in the textures paves the way to utilize fractal features for representing textures. However, the fractal dimension does not show much variation to distinguish different textures. The fuzzy features are therefore explored in this paper.

The ambiguities in texture that arise due to fuzzy nature of the image function give an opportunity to devise fuzzy texture features. Since texture is region based, we consider arrangement of image functions (i.e. intensities) of pixels in a local region, say, a window, in order to characterize the texture using the Gaussian type membership function. Next, these features are utilized to construct texture descriptors which in turn will be used for texture segmentation.

The paper is organized as follows: Section 2 presents the feature extraction of fuzzy features of images. Section 3 describes the Improved Modified Mountain Clustering Technique. We also present the fuzzy descriptor based texture segmentation algorithm. Results and comparisons of texture segmentation are shown in Section 4. Finally conclusions are drawn in Section 5.

II. FEATURE EXTRACTION

2.1 Extraction of Fuzzy features

To convert the spatial domain image into the fuzzy domain, we consider the spatial arrangement of gray levels of pixels over a window. The fuzzy property can be expressed in terms of a membership function. A membership function to this effect is defined by the Gaussian type function.

$$\mu_{(k,l)}(i,j) = \exp\left[-\{(x(i,j) - x(k,l))/\tau\}^2\right] \quad (1)$$

where x(i, j) is the gray level of the pixel at the (i,j)th position and τ is the fuzzifier which is taken as the window size. In our experiment, we take the value of τ as 5. We note that

$$\mu_{(i,j)}(k,l) = 1 \text{ if } x(i,j) = x(k,l) \quad (2)$$

To take the effect of the neighboring pixels, we obtain the cumulative response of (i, j)th pixel as follows

$$y(i,j) = \sum_{k,l} \mu_{(k,l)}(i,j) * x(k,l) / \sum_{k,l} \mu_{(k,l)}(i,j) \quad (3)$$

This is the defuzzified response of the (i,j)th pixel over the window of size 5. This process is repeated for all pixels in the image to yield a texture image consisting of all the defuzzified values. For convenience of notation, we designate the matrix formed by y(i, j) as the response matrix, Y.

2.2 Texture Descriptors

We will derive descriptors so as to capture the essence of the texture of a subimage of size w×w from the original image of size M×N (w«M and w«N). The following descriptors are used to represent the information content in the defuzzified response matrix.

Descriptor 1: Quotient Response

This descriptor gives an idea about the relation between the maximum and the minimum of $y(i,j)$ in a window. This quotient is defined as

$$Q_y = \frac{Max\{y(i,j)\} - Min\{y(i,j)\}}{Max\{y(i,j)\} + Min\{y(i,j)\}} \quad (4)$$

Descriptor 2: Entropy

This second feature is derived from the defuzzified response $y(i,j)$ in a window. The entropy of $y(i,j)$ is calculated as:

$$E_y = -\frac{1}{N_b \log(2)} \sum_i \sum_j y'(i,j).\log y'(i,j) \quad (5)$$

where $y'(i,j) = \frac{y(i,j)}{\sum_k \sum_l y(k,l)}$ such that $\sum_i \sum_j y'(i,j) = 1$ and N_b is the number of pixels in a window.

Descriptor 3: Energy

The third feature gives the measure of the energy of the defuzzified response in a window. This energy is calculated following an analogy with the power equation (i.e., Current2 ×Resistance) as:

$$I_y = \frac{1}{N_b} \sum_i \sum_j [y'(i,j)]^2 \quad (6)$$

Where $y'(i,j)$ is as defined above.

The three descriptor values for each pixel will be used to segment the image into logical regions where textural patterns are similar in nature. We employ Improved Modified Clustering technique for segmenting the image on the basis of descriptor values. In the following section, we will present the algorithm for descriptor based texture segmentation.

III. TEXTURE SEGMENTATION

3.1 Improved Mountain Clustering

All clustering techniques cannot uncover all the clusters present in the data [7] with equal facility, because the clustering algorithms often contain implicit assumptions about cluster shape or multiple-cluster configurations based on the similarity measures and grouping criteria used. Humans can compete with the automatic clustering procedures in two dimensions, but most real problems involve clustering in higher dimensions. It is difficult for humans to obtain an intuitive interpretation of data embedded in a high-dimensional space. In addition, data hardly follow the “ideal” structures (e.g., hyper spherical, linear). This explains the reason behind the development of a large number of clustering algorithms in the literature. We consider Improved Mountain Clustering (IMC) [8] technique for texture segmentation because it is better in terms of optimum number of clusters with less computational complexity

IMC Algorithm

Step 1: Normalize each dimension of hyper-space, so that the data points are bounded by the unit hypercube.

We define the j^{th} data in $\mathbf{x}$ hyperspace as:

$$\mathbf{x}^j = \left\{x_1^j, x_2^j, ..., x_D^j\right\}$$

The normalized data points $\overline{\mathbf{x}}^j$, are defined as:

$$\overline{\mathbf{x}}^j = \frac{\left\langle \mathbf{x}^j - (\mathbf{x}^j)_{\min} \right\rangle}{\left\langle (\mathbf{x}^j)_{\max} - (\mathbf{x}^j)_{\min} \right\rangle}; \quad \forall j = 1, 2, ...n \tag{7}$$

where

$$(\mathbf{x}^j)_{\min} = \left\{ \min_{j=1}^{n} x_1, \min_{j=1}^{n} x_2, ..., \min_{j=1}^{n} x_D, \right\} \tag{8}$$

$$(\mathbf{x}^j)_{\max} = \left\{ \max_{j=1}^{n} x_1, \max_{j=1}^{n} x_2, ..., \max_{j=1}^{n} x_D, \right\} \tag{9}$$

and n is the total number of data points.

Step 2: Determine the values of the positive constants d_1 and d_2 in the neighborhood of the data point for each window. We compute these from the heuristic:

$$d_1 = d_2 = \frac{1}{2n} \sum_{j}^{n} \left(\frac{\min\left(\mathbf{x}^j\right)}{\sum_{i=1}^{D} x_i^j} \right) \tag{10}$$

Step 3: Calculate the potential value of each point using the mountain function, which is a function of distance $d^2(\overline{x}^r, \overline{x}^j) = (\overline{x}^r - \overline{x}^j)Q(\overline{x}^r - \overline{x}^j)$ between $\overline{\mathbf{x}}^r$ and all other data points.

$$P_1^r = \sum_{j=1}^{n} \exp\left[-\left(\frac{d^2\left(\overline{\mathbf{x}}^r, \overline{\mathbf{x}}^j\right)}{d_1^2} \right) \right]; \quad \forall r = 1, 2, ...n \tag{11}$$

Step 4: Select the first cluster center according to the highest value of P_1^r as:

$$\overline{c}_1 = \overline{\mathbf{x}}^{1^*} \Leftarrow P_1^* = \max_{r=1}^{n} \left(P_1^r\right) \tag{12}$$

Step 5: Assign those data points to the first cluster whose Euclidean distance from the first cluster center is less than the threshold, d_2, i.e.

If $d^2\left(\overline{\mathbf{x}}^r, \overline{c}_1\right) \le d_2;\ \forall r = 1, 2, \ldots n$ (13)

then $\overline{\mathbf{x}}^r$ is assigned to the first cluster.
Step 6: Remove all those data points from the dataset which are assigned to the first cluster.
Step 7: Repeat Steps 2 to 5 for the remaining or reduced dataset to make successive clusters. Similarly for the selection of m^{th} cluster center, potential values are found for the reduced dataset and m^{th} cluster center is selected with the highest value of P_m^r as under:

$$\overline{c}_m = \overline{\mathbf{x}}^{m^*} \Leftarrow P_m^{\ *} = \max_{r=1}^{n}\left(P_m^r\right) \tag{14}$$

Step 8: Determine the optimum number of clusters using the validity function S defined by,

$$S = \frac{\sum_{m=1}^{M}\sum_{r=1}^{n}\left(\overline{\mu}_m^r\right)^2 . \left\|\overline{\mathbf{x}}^r - \overline{c}_m\right\|^2}{n.\min_{i \ne j}\left\|\overline{c}_i - \overline{c}_j\right\|^2}\text{; where, } \quad \overline{\mu}_m^r = \frac{\mu_m^r}{\sum_{m=1}^{M}\mu_m^r} \tag{15}$$

and the membership function μ_m^r represents the degree of association of the r^{th} data point to the m^{th} cluster center and is defined as

$$\mu_m^r = \exp\left[-\left(\frac{d^2\left(\overline{\mathbf{x}}^r, \overline{c}_m\right)}{d_2^2}\right)\right] \tag{16}$$

The optimum number of clusters for a given dataset is decided by the validity function S, which is the ratio of compactness to separation. The compactness is related to the closeness of all other points with respect to a cluster center whereas the separation indicates the distance between the cluster centers.
Step 9: Form an optimum number of clusters, using the Steps 2 to 7 and separate out these clusters from the whole dataset. The rest of the data points are distributed among the already formed clusters depending upon their Euclidean distance, i.e. nearness to the respective clusters.

3.2 An Algorithm for Texture Segmentation

Step 1: Convert the spatial domain image into the fuzzy domain. Calculate the correlation of each (i, j)th pixel about its neighborhood in a window of size 5 using the Gaussian membership function given in equations (1), (2).

Step 2: Obtain the defuzzified response of the (i, j) th pixel using the formula given in (3), over the window of size 5. We designate the matrix formed by all the defuzzified response values by a response matrix, Y.

Step 3: Determine the three descriptors – Quotient Response, Entropy, and Energy for each pixel (i, j). Apply the formulas in (4), (5), and (6) on the response matrix Y in a window of size w to compute the values of descriptors at (i, j) th pixel. The window size w is considered as 5, 7,..,etc. depending on the size texture pattern.

Step 4: Apply Improved Mountain Clustering technique on the descriptor values to cluster the pixels having the same texture features.

IV. RESULTS AND DISCUSSIONS

In this section, we present the texture segmentation of natural images taken from the real life situation. We employ the descriptor based approach to segment the images of size 150×100. The defuzzified response is calculated for each pixel over the window of size 5x5. This response matrix is further used to find the texture descriptors for each pixel. The three descriptors – quotient response, entropy and energy are calculated for each pixel over the window of

size wxw. The size of window should be sufficient to capture the pattern of textures present in the natural image. We will apply Improved Mountain Clustering technique on descriptor values to segment the images.
Figure 1 show the texture segmentation of a Zebra Butterfly image of size 120×90 and descriptor values are given in Table 1. We take the number of clusters as 3 and the descriptor window size as 7x7.

(a) (b) (c)

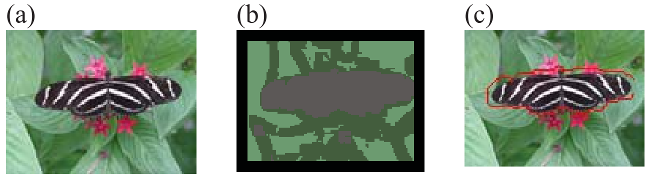

Fig. 1. Texture Segmentation of Zebra Butterfly Image

(a) Original Image
(b) Texture Segmented image with the descriptor window size 7
(c) Segmented mask of zebra butterfly

TABLE 1
Descriptor values - Zebra Butterfly image

	Quotient Response	Entropy	Energy
Cluster 1	0.109678	0.11454	0.000417766
Cluster 2	0.332316	0.114165	0.000428657
Cluster 3	0.784438	0.107171	0.000658509

The texture descriptors are derived for each pixel from the fuzzy response of the image. The Improved Modified Clustering technique is applied to segment the image based on the descriptor values of each pixels. Table 1 gives the three descriptor values for each cluster center. It may be observed from Fig. 1 (b) that the natural texture of the butterfly is identified correctly.
We will now compare the results of texture segmentation with the those of Blobworld [1]. In order to segment each image automatically, Blobworld models the joint distribution of color, texture, and position features with a mixture of Gaussians. Expectation-Maximization (EM) algorithm has been used to estimate the parameters of this model that results in the segmentation of the image.
Fig. 2 depicts the results of comparison of the segmented Zebra images from both the proposed and the Blobworld approaches.

(a) (b) (c) (d) (e)

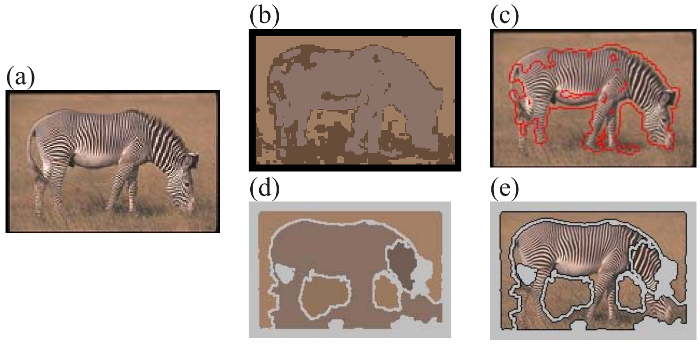

Fig. 2. Comparison of Texture Segmentation results of Zebra 1 image

(a) Original Image
(b) Texture Segmented image with the descriptor window size 5
(c) Segmented mask of Zebra 1
(d) Texture Segmented image by Blobworld
(e) Segmented Mask of Zebra 1 by Blobworld

TABLE 2
Descriptor values - Zebra 1 image

	Quotient Response	Entropy	Energy
Cluster 1	0.0190574	0.185751	0.00160019
Cluster 2	0.133934	0.185615	0.00160759
Cluster 3	0.335112	0.184672	0.0016582

In Figs. 2 (b) and (c), we can see that a small portion in the middle of the body and on the back side are not included in the segment of the zebra, because in those portions, the texture of zebra is not clearly visible. In the segmentation using the Blobworld shown in Figs.2 (d) and 2 (e), we observe that the head of the zebra doesn't belong to the segment of zebra and the ground which is not a part of the zebra, belongs to the segment of the zebra. It may be noticed that the descriptor based method yields the better segmentation than that of the Blobworld.

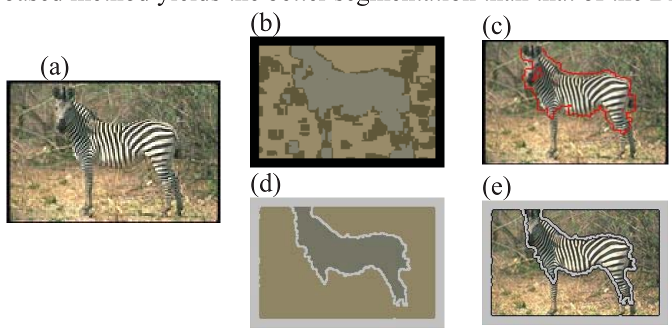

Fig. 3. Comparison of Texture Segmentation results of Zebra 2 image

(a) Original Image
(b) Texture Segmented image with the descriptor window size 5
(c) Segmented mask of Zebra 2
(d) Texture Segmented image by Blobworld
(e) Segmented Mask of Zebra 2 by Blobworld

TABLE 3
Descriptor values - Zebra 2 image

	Quotient Response	Entropy	Energy
Cluster 1	0.257646	0.114344	0.000423329
Cluster 2	0.409468	0.113947	0.000434287
Cluster 3	0.618911	0.113049	0.000460804

In Fig.3, we have observed that descriptor based segmentation and Blobworld segmentation for zebra image are almost same.

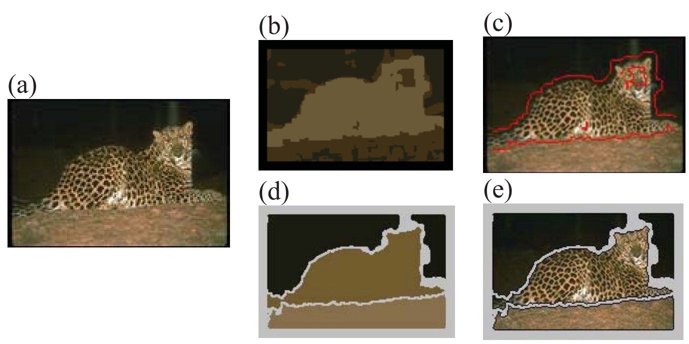

Fig. 4. Comparison of Texture Segmentation results of Tiger 1 image

(a) Original Image
(b) Texture Segmented image with the descriptor window size 7
(c) Segmented mask of Tiger 1

(d) Texture Segmented image by Blobworld
(e) Segmented Mask of Tiger 1 by Blobworld

TABLE 4
Descriptor values - Tiger 1 image

	Quotient Response	Entropy	Energy
Cluster 1	0.051038	0.114575	0.000416798
Cluster 2	0.209717	0.114423	0.000421171
Cluster 3	0.71041	0.111816	0.000490781

In Figs.4 (b) and (c), we can see that a portion in the face of tiger is not shown as part of the face of the tiger because that portion does not possess the texture of its body. Here we have chosen the size of descriptor window as 7 to capture the texture pattern of the tiger in the image.

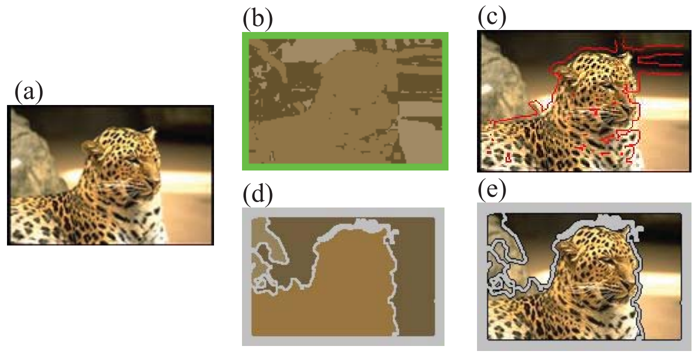

Fig. 5. Comparison of Texture Segmentation results of Tiger 2 image

(a) Original Image
(b) Texture Segmented image with the descriptor window size 7
(c) Segmented mask of Tiger 2
(d) Texture Segmented image by Blobworld
(e) Segmented Mask of Tiger 2 by Blobworld

TABLE 5
Descriptor values - Tiger 2 image

	Quotient Response	Entropy	Energy
Cluster 1	0.0257579	0.185748	0.00160037
Cluster 2	0.15947	0.185502	0.00161391
Cluster 3	0.488816	0.183449	0.00172174

The tiger image shown in Fig. 5 (a) has an irregular texture pattern. We adopt the descriptor based texture segmentation again with the descriptor window size of 7 and the results are shown in Figs. 5 (b) and (c).

V. CONCLUSIONS

The descriptor based texture segmentation of the images is presented in this paper. The proposed texture descriptor algorithm is useful in identifying a segment of an image having any particular pattern. In the examples considered, it may be noticed that the patterns in Butterfly, Zebra, and Tiger images are not vey regular and the texel sizes in these images vary considerably. Hence we keep our descriptor window size sufficiently large to accommodate the texel of any pattern in the image.

The results of segmentation obtained from the Blobworld system are used to compare with our results. In Blobworld, color and texture are both used for segmentation. But, we have attempted to segment the image based on texture only. Our major concern in this work is to identify an object in an image and it may be observed that the desired object is clearly recognizable by its texture information. Comparing the Blobworld results with ours, we find that both the segmented images are alike with minor deviation and in some cases our segmentation is better.

REFERENCES

[1] C. Carson, S. Belongie, J. H. Greenspan, J. Malik, “Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying ”. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, No. 8, August 2002.

[2] J. Shi and J. Malik, “Normalized cuts and image segmentation”, *IEEE Trans. On Pattern Analysis and Machine Intelligence (PAMI)*, 22(8), Aug. 2000, pp. 888-905.

[3] Bouman, C., and Shaprio, M.: ‘A Multiresolution random field model for Bayesian image segmentation’, *IEEE Trans. Image Process.*, 1994, 3, (2), pp. 162–177

[4] Choi, H., and Baraniuk, R.: ‘Image segmentation using wavelet-domain classification’, Proc. SPIE–Int. Soc. Opt. Eng., 1999, 3816, pp. 306–320

[5] Choi, H., and Baraniuk, R.: ‘Multiresolution image segmentation using wavelet-domain hidden Markov models’, IEEE Trans. Image Process., 2001, 10, (9), pp. 1309–1321

[6] Noda, H., Shirazi, M.N., and Kawajuchi, E.: ‘MRF based texture segmentation using wavelet decomposition images’, Pattern Recognit., 2002, 35, (4), pp. 771–782

[7] A K Jain, M.N.Murthy, P.J.Flynn, “Data Clustering: A Review” , *ACM Computing Surveys*, Vol. 31,No. 3,September 1999

[8] N.K. Verma and M. Hanmandlu, “Color Segmentation via improved Mountain Clustering technique”, International Journal of Image and Graphics (in press) 2006.

[9] M. Hanmandlu, Anirban Das, “Content-based Image Retrieval by Information Theoretic Measure”, Defense Science Journal, vol. 61, pp. 415-430, 2011.