Image Segmentation Using RGB Decomposition and Modified Bacterial Foraging Optimization

Bobbinpreet Kaur[1], Raju Sharma[2]
[1] *Student BBSBEC Fatehgarh Sahib,*
[2] *Assistant Professor, BBSBEC Fatehgarh Sahib*

ABSTRACT:

This paper addresses the problem of segmenting the image into various color components by using combined approach of Modified Bacterial foraging optimization approach and RGB Decomposition. The original image is decomposed into separate planes of R G and B and then modified Bacterial Foraging algorithm is applied on three planes separately to calculate three different thresholds. Segmentation is performed on the basis of Thresholding. Since Image segmentation is the basic step in many image processing applications so faithful segmentation algorithm must be developed for successful implementation of the processing applications. Main aim of image segmentation is to extract the information which is of interest for a particular application. This methodology will be able to separate three different colors of original image. The accuracy of any algorithm varies with input image. This is an extensive research area as many applications depend upon results of the algorithm for image segmentation, but it is still difficult to assess whether one algorithm produces more accurate results than another for all type of images.

KEYWORDS: *Modified BFO, RGB planes, Segmentation, Thresholding*

I. INTRODUCTION

Image segmentation deals with dividing the image according to either similarity or dissimilarity. It is an important prospect in image analysis. Color is a perceptual phenomenon related to human response to different wavelengths in the visible electromagnetic spectrum [1].Color is the most prominent feature of any image. Extracting color information from any image has many applications related to computer vision algorithms. Color of an image can carry much more information than gray level [2].

In a broad sense the colored images segmentation are classified as follows:-

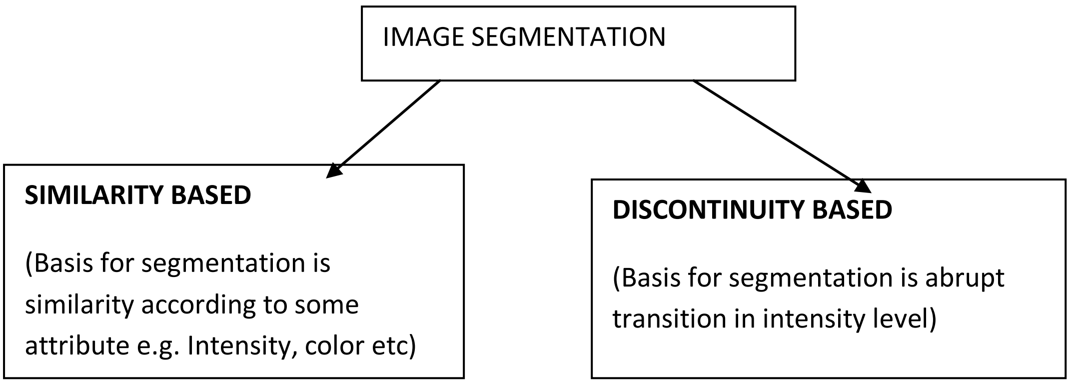


Figure 1:- Types of Image Segmentation

1.1 Similarity Based Segmentation Techniques:-

Similarity based segmentation techniques consists of Thresholding methods and region based methods. Thresholding methods convert grayscale image into binary image (Black and white) image by first choosing a gray level in the original image and then turning every pixel black or white according to whether its gray value is greater or less than T.[3]

A pixel = White if gray value >T

= Black if gray value <T

Region based methods **-**The main principle behind region growing method is a collection of pixels with similar properties (color, intensity level etc.) to form a region. Region growing method partitions an image into regions that are similar according to given criteria, such as gray character, color character or texture character.

1.2 Discontinuity Based Segmentation Techniques Discontinuity based segmentation techniques consists of Edge detection, Line detection and Point detection methods. Edge is a boundary between two homogeneous regions. Edge detection refers to the process of identifying and locating sharp discontinuities in an image.

II. BACTERIA FORAGING OPTIMIZATION

This algorithm is one in class of nature inspired algorithm developed in order to solve number of optimization problems. It is a widely accepted algorithm for optimization based on social foraging behavior of E.coli bacteria. Bacteria move towards a particular direction in search of food based upon gradients of chemicals present in the environment. Foraging means locating, handling, and ingesting food [4].Bacterial foraging optimization algorithm (BFOA) has been widely accepted as a global optimization algorithm of current interest for distributed optimization and control The course(process) of natural selection tends to eradicate animals having poor foraging strategies and favor the propagation of genes of those animals that have flourishing foraging strategies, since they are more credible to enjoy reproductive success[5]. BFO is designed to tackle non gradient optimization problems and to handle complex and non differentiable objective functions. After many generations, weak foraging strategies are either eliminated or shaped into good ones. This stroke (action) of foraging led the researchers to use it as optimization process. The *Escherichia Coli or E. coli* bacteria that are present in our intestines also experience a foraging strategy. The control system of these bacteria that dictates how foraging should continue consists of four principal mechanisms, namely chemotaxis, swarming, reproduction, and elimination dispersal.BFO has been successfully applied on various applications like Option Model calibration [6], image processing [7], RFID Network scheduling [8] and many other applications.
The four basic steps in BFOA are explained below [9]

1.Chemotaxis:- In the original BFO, a unit walk of the bacteria with random direction represents a "tumble" and a unit walk with the same direction in the last step indicates a "run." It can move in two different strategies. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime.

2. Swarming: - A group of E.coli cells arrange themselves in a travelling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of succinate, release an attractant aspartate, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density.

3. Reproduction:- The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the population of bacteria constant.

4. Elimination and Dispersal: **-** The chemotaxis provides a basis for searching the local best solution
And reproduction process speeds up the convergence which has been simulated by the classical BFO. The bacteria with the best positions are kept and the remaining bacteria population is killed. The bacteria with best positions are then moved to another position within the environment.

III. PROPOSED WORK

The proposed modified bacterial foraging algorithm is implemented on the input image in the following steps:
Step-1: The input image is first converted into its RGB image components. Therefore, we get three images in red, green and blue components.
Step-2: Maximum No. of Colors in any color component image = 256 (Black to Red, Black to Green nd Black to Blue)

Step-3: Initialize Threshold in each RGB component image as TR = 0, TG = 0 and TB = 0
Step-4: Take the red component image. Compute the size of the image into row x column giving rise to bacterial search area.
Step-5: Initialize the Nc and Ns steps at 1. Nc = 1 and Ns=1; Chemo tactic and swim length.
Step-6: Compute the health status of all image pixels by using the image histogram. The health status of ith color pixel Hi is given by:

Hi = Pi / (row x column), where Pi is the no. of pixels of ith color.

Step-7: Compute the Euclidean distance ED between the adjacent pixels as ED = F(r,c) – F(r,c+1)
Where F(r,c) and F(r,c+1) are the pixel color value of two adjacent pixels.
Step-8: If ED is less than some threshold ED, then replace the F(r,c) by F(r,c+1), thereby reducing the no. of colors in the entire image.
Step-9: Now compute the difference of health status of H(r,c) and H(r,c+1) pixel . If H(r,c) and H(r,c+1) are less than the threshold health status, then they are the unpopular colors and can be eliminated to produce a new color.
Step-10: Keep on adding the color value to TH, TG or TR as may be the case from step-4.
Step-11: Move the pixel pointer over the entire image.
Step-12: Repeat from step-4 to 11 for green and blue component images in the same manner and compute the TR, TG and TB.
Step-13: Compute the individual thresholds as given by:

TH = TH/Image Size, TG = TG/Image Size TB = TB/Image Size

Step-14: Compute the final threshold as given by: T = (TH + TG + TR)/3
Step-15: Apply the final threshold over the original image and compute the performance indices as standard deviation, entropy, PSNR and class variance.

IV PERFORMANCE INDICES

Following performance indices are evaluated for measuring performance of the proposed algorithm
PSNR: The peak-signal to noise ratio (PSNR) was used to evaluate the reconstructed image quality. The PSNR is defined as follows:

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{N\times N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(f(i,j)-\hat{f}(i,j)\right)^2}dB,$$

where N × N is the size of the original image and f(i,j) and ˆf(i,j) are the gray-level pixel values of the original and reconstructed images, respectively.

Standard Deviation (SD): The standard variation of an image is given by:

$$\hat{\sigma}^2 = \frac{1}{n\times n}\sum_{j=1}^{n}\sum_{i=1}^{m}(x_{i,j}-\hat{\mu})^2,$$

This corresponds to the degree of deviation between the gray levels and its mean value, for the overall image.

Entropy E: The expression of the information entropy of an image is given by:

$$H = -\sum_{i=0}^{L-1} p_i \ln p_i,$$

Where L denotes the number of gray level, pi equals the ratio between the number of pixels whose gray value equals i (0 to L - 1) and the total pixel number contained in an image. The information entropy measures the richness of information in an image. If pi is the const for an arbitrary gray level, it can be proved that the entropy will reach its maximum. Below given figures shows the results of the segmentation obtained by Ostu algorithm and proposed algorithm. Fig. 1 is the original image, while fig. 2 and 3 shows the segmented image obtained after applying the Otsu and modified proposed algorithm.Class Variance: Class variance of the segmented image is computed by the following computation method:If the histogram is divided into two classes by the gray-level intensity t (threshold), then the probabilities of the respective classes can be expressed as:

$$p_1(t) = \sum_{i=0}^{t} p(i) \quad \text{and} \quad p_2(t) = \sum_{i=t+1}^{N-1} p(i)$$

Also, the class means m1 and m2 are given by:

$$m_1(t) = \sum_{i=0}^{t} ip(i) / p_1(t)$$

$$m_2(t) = \sum_{i=t+1}^{N-1} ip(i) / p_2(t)$$

The two class variances are given by:

$$\sigma_1^2(t) = \sum_{i=0}^{t} (i - m_1)^2 \frac{p(i)}{p_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{N-1} (i - m_2)^2 \frac{p(i)}{p_2(t)}$$

The total class variance (σ_T) is given by:

$$\sigma^2{}_T = \sigma_2{}^B + \sigma^2{}_W$$

Where $\sigma^2{}_B$ is the between class variance and $\sigma^2{}_W$ is the within class variance and given by following equations.

$$\sigma^2{}_W\,(t) = p_1(t)\,\sigma_1{}^2(t) + p_2(t)\,\sigma^2{}_2(t)$$

$$\sigma^2{}_B\,(t) = p_1(t).p_2(t)\,\{m1(t) - m2(t)\}^2$$

V RESULTS

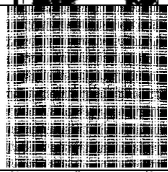
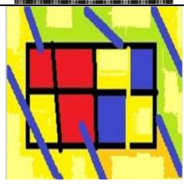
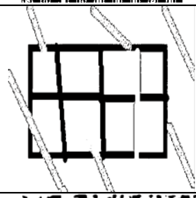
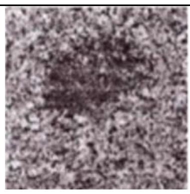
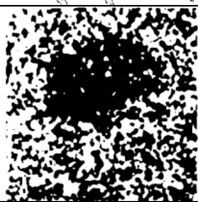
The proposed algorithm is implemented on different images using matlab code. Table 1 depicts the values of the performance indices for all the images. Threshold, Standard Deviation, Class Variance and Entropy is being calculated and shown. The performance indices calculated through the code proved the results are better than the previously suggested techniques.

Table 1

Images	Threshold	Standard Deviation	Class Variance	Entropy
Hunterman	0.3705	0.4821	0.0070	0.9601
Leena	0.4638	0.4342	0.0049	0.9134
BW Pattern	0.5956	0.4594	0.0098	0.8409
Colored Pattern	0.3184	0.3933	0.0401	0.7258
Dotted Pattern	0.6095	0.4373	0.0045	0.9522
Airplane	0.5683	0.4297	0.0034	0.8860

Table 2 shows the results of the matlab code for the above said images. Original as well as segmented images are being shown in the tabular form.

Table 2

Image Name	Original Image	Segmented Image with modified BFO
Hunterman		
Leena		
BW Pattern		
Colored Pattern		
Dotted Pattern		
Airplane		

V1 CONCLUSION

It has been observed from the results that the proposed algorithm offers a more resolved thresholding in comparison to any other thresholding method like Otsu or watershed. The image decomposition into its RGB components and computation of threshold in each component image using BFO algorithm prove to be a good tool in order to faithfully segment or threshold the image. The proposed algorithm finds some limitation at the end of time of computation. This is because the algorithm runs three times on the input image in R, G and B components. Further if the size of the image increases, the time may again increase. However, the speed of algorithm can be optimized when running the same on high performance machine. The final threshold value has been computed by taking the mean of the three components thresholds. Further work may be carried out in order to integrate the three thresholds to fine tune the application.

REFERENCES

[1] Rosenfeld and Kak, Digital Picture Processing. 2, Academic Press, New York, 1982

[2] H. D. Cheng, X. H. Jiang, Y. Sun, and J. L. Wang, “Color image segmentation: Advances and prospects,” Pattern Recognition., Volume 34 Issue 12 December 2011 pp 2259-2281

[3] Alasdair Mc Andrew, An Introduction to Digital Image Processing with Matlab, Cengage learning

[4] S.Saadi, A.Kouzou, A.Guessoum, M.Bettayeb “Bacterial Foraging Algorithm for Neutron Radiography Image Quality Improvement”] 201O 7th International Multi-Conference on Systems, Signals and Devices

[5] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems Magazine*, vol. 22, pp. 52–67, 2002.optimization and control,” Control Systems Magazine, vol. 22, no. 3, pp. 52-67, Jun 2002.

[6] Kevin M Passino, “Biomimicry for optimisation,control,automation” Springer-Verlag London,UK 2005

[7] Bakwad KM, Pattnaik SS, Sohi BS, Devi S, Panigrahi BK, Sastri GSVR, “Bacterial Foraging Optimization Technique Cascaded with Adaptive Filter to Enhance Peak Signal to Noise Ratio from Single Image”, IETE Journal of Research, Vol 55 (4), Jul-Aug 2009.

[8] Liu W, Chen H, Chen H, and Chen M, "RFID Network Scheduling Using an Adaptive Bacteria Foraging Algorithm", Journal of Computer Information Systems (JCIS), Vol. 7(4), 2011, pp. 1238-1245.
[9] Sambarta Dasgupta, Swagatam Das "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications" Studies in Computational intelligence volume 203,2009 pp23-55