

Color Image Quantization based on Bacteria Foraging Optimization

Rajinder Kaur
GNDEC, Ludhiana, Punjab,
India

Akshay Girdhar
A.P. & H.O.D,
GNDEC, Ludhiana, Punjab,

Surbhi Gupta
Sr. Lecturer, RIEIT,
Railmagra, Punjab,

ABSTRACT

Bacterial Foraging Optimization (BFO) is optimization technique proposed by K. M. Passino in 2002 To tackle complex search problems of the real world, scientists have been drawing inspiration from nature and natural creatures for years. Bacterial Foraging Optimization is a burgeoning nature inspired technique to find the optimal solution of the problem. A Color images Quantization is necessary if the display on which a specific image is presented works with less colors than the original image. While a lot of color reduction techniques exist in the literature, they are mainly designed for image compression as they tend to alter image color structure and distribution, the researchers are always finding alternative strategies for color quantization so that they may be prepared to select the most appropriate technique for the color quantization. The objective of this research work, is to implement a new algorithm for Color Image Quantization based on Bacteria Foraging Optimization. To compare the designed algorithm with other swarm intelligence techniques and to validate the proposed work. The proposed algorithm is then applied to commonly used images including the phantom images. The conducted experiments indicate that proposed algorithm generally results in a significant improvement of image quality compared to other well-known approaches.

Keywords

Quantization, Bacteria Foraging Optimization, Swim, Tumble, Chemo-tactic, CMC, SI.

1. INTRODUCTION

Bacterial foraging behaviours are used as a source of engineering applications and computational model. A few models have been developed to bacterial foraging behaviours and been applied for solving practical problems [1, 2, 13]. Among them, Bacterial Foraging Optimization (BFO) is a population-based numerical optimization algorithm. Until date, BFO has been applied successfully to some engineering problems, such as optimal control [9], harmonic estimation [12], transmission loss reduction [17] and machine learning [10].

1.1. Bacterial foraging optimization algorithm

The motile bacteria such as *Escherichia coli* propel themselves by rotating their flagella. They rotate their flagella counter clockwise to move forward rotate also called as “swimming” (or “runs”). But to move the bacteria in random direction i.e. “tumble” they rotate their flagella clockwise and then swims again. Tumbling just changes the direction of movement of bacteria. The bacteria first of all tumble in random direction to search for food. As the bacteria found the food in a particular direction it then swims toward the food in

that direction. An alternation between “swim” and “tumble” enables the bacteria to search for food in random directions [2].

The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely, chemo taxis, reproduction, and elimination-dispersal. These are described as follows [2].

1.1.1 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.” Suppose $\phi^i(j, k, l)$ represents the bacterium at j^{th} chemotactic, k^{th} reproductive, and l^{th} elimination-dispersal step. $C(i)$ is the chemotactic step size during each run or tumble (i.e., run-length unit). Then in each computational chemotactic step, the movement of the i th bacterium can be represented as

$$\phi^i(j+1, k, l) = \phi^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}} \quad \dots(1.1)$$

Where $\Delta(i)$ is the direction vector of the j^{th} chemotactic step. When the bacterial movement is *run*, $\Delta(i)$ is the same with the last chemotactic step; otherwise, $\Delta(i)$ is a random vector whose elements lie in $[-1, 1]$. With the movement of run or tumble taken at each step of the chemotaxis process, a step fitness, denoted as $J(i, j, k, l)$, will be evaluated [2].

1.1.2 Reproduction

The fitness value of each bacterium is calculated as the sum of the step fitness during its life, that is

$$\sum_{j=1}^{N_c} J(i, j, k, l)$$

Where N_c is the maximum step in a chemotaxis process. All bacteria are sorted in descending order according to health status. In the reproduction step, only the first half of population survives. The surviving population is divided into two identical ones, which are then placed in the same locations at which their parents were. Thus the total population of bacteria keeps constant [2].

1.1.3 Elimination and Dispersal

The chemotaxis provides a basis for searching the local best solution, and the 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 [2].

1.1.4 BFO Algorithm

In what follows we briefly outline the original BFO algorithm step by step[1].

Step 1. Initialize parameters n, S, N_c, N_s, N_{re}

$N_{ed}, P_{ed}, C(i)$ ($i = 1, 2, \dots, S$), ϕ^i , where

n : dimension of the search space,

S : the number of bacteria in the colony,

N_c : Chemotactic steps,

N_s : Swim steps,

N_{re} : Reproductive steps,

N_{ed} : Elimination and dispersal steps,

P_{ed} : Probability of elimination,

Step 2. Elimination-dispersal loop: $l = l + 1$.

Step 3. Reproduction loop: $k = k + 1$.

Step 4. Chemotaxis loop: $j = j + 1$.

Substep 4.1. For $i = 1, 2, \dots, S$, take a chemotactic step for bacterium i as follows.

Substep 4.2. Compute fitness function, $J(i, j, k, l)$.

Substep 4.3. Let $j_{last} = J(i, j, k, l)$ to save this value since we may find better value via a run.

Substep 4.4. Tumble. Generate a random vector

$$\Delta(i) \in R^n$$

with each element $\Delta m(i)$, $m = 1, 2, \dots, n$, a random number on $[-1, 1]$.

Substep 4.5. Move. Let

$$\phi^i(j+1, k, l) = \phi^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

...(1.2)

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i .

Substep 4.6. Compute $J(i, j+1, k, l)$ with $\phi^i(j+1, k, l)$.

Substep 4.7. Swimming.

Let $m = 0$ (counter for swim length).

While $m < N_s$ (if has not climbed down too long), the following hold.

- Let $m = m + 1$.

- If $J(i, j+1, k, l) < j_{last}$ let $j_{last} = J(i, j+1, k, l)$, then another step of size $C(i)$ in this same direction will be taken as (2.2) and use the new generated.

- $\phi^i(j+1, k, l)$ to compute the new $J(i, j+1, k, l)$.

- Else let $m = N_s$.

Substep 4.8. Go to next bacterium ($i+1$). If $i \neq S$, go to Substep 4.2 to process the next bacterium.

Step 5. If $j < N_c$, go to Step 3. In this case, continue chemotaxis since the life of the bacteria is not over.

Step 6. Reproduction.

Substep 6.1. For the given k and l , and for each $i = 1, 2, \dots, S$, let

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l) \quad \dots(1.3)$$

be the health of the bacteria. Sort bacteria in order of ascending values (J_{health}).

Substep 6.2. The S_r bacteria with the highest J_{health} values die and the other S_r bacteria with the best values split and the copies that are made are placed at the same location as their parent.

Step 7. If $k < N_{re}$, go to Step 2. In this case the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

Step 8. Elimination-dispersal: for $i = 1, 2, \dots, S$, with probability P_{ed} , eliminate and disperse each bacterium, which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to Step 2; otherwise end [2].

1.2 Color image quantization

A color image quantization is a process that reduces the number of distinct colors used in an image, usually with the intention that the new image should be as visually similar as possible to the original image. Computer algorithms to perform color quantization on bitmaps have been studied since the 1970s. Color quantization is critical for displaying images with many colors on devices that can only display a limited number of colors, usually due to memory limitations, and enables efficient compression of certain types of images. Color image quantization can be formally defined as follows:

Given a set of $N_{S'}$ colors where $S' \subset R^{N_d}$ and N_d is the dimension of the data space. The color quantization is a map $f_q: S' \rightarrow S''$ where S'' is a set of $N_{S''}$ colors

such that $S'' \subset S'$ and $N_{S''} < N_{S'}$. The objective is to minimize the quantization error resulting from replacing a color $c \in S'$ with its quantized value $f_q(c) \in S''$ [11] [18].

Color image quantization can be viewed as stepwise process:

1. In the first step statistics on the used colors in the image that is to be quantized are generated (histogram analysis)
2. a) Based on the analysis the color lookup-table has to be filled with values.
b) The true color values are mapped to the values of the color table. The color values have to be mapped to the nearest color entries in the color table.
3. The original image is quantized. Each pixel is transformed to the appropriate index of the color table.
4. Optionally an error diffusion technique can be applied.

Color quantization is important because quantized image can be used in many applications including the following.

- It can be used in lossy compression techniques [18].

- It is suitable for mobile and hand-held devices where memory is usually small [14].
- It is suitable for low-cost color display and printing devices where only a small number of colors can be displayed or printed simultaneously [15].
- Most graphics hardware use color lookup tables with a limited number of colors [5].

So, the main objective of color image quantization is to map the set of colors in the original color image to a much smaller set of colors in the quantized image [11].

1.3 Lab color model

LAB stands for Luminance (or lightness) and A and B (which are chromatic components). According to this model A ranges from green to red, and B ranges from blue to yellow. This model was designed to be device independent. The Luminance ranges from 0 to 100, the A component ranges from -120 to +120 (from green to red) and the B component ranges from -120 to +120 (from blue to yellow) [6] [7]. The three parameters in the model represent the lightness of the color (L^* , $L^*=0$ yields black and $L^*=100$ indicates white), its position between magenta and green (a^* , negative values indicate green while positive values indicate magenta) and its position between yellow and blue (b^* , negative values indicate blue and positive values indicate yellow). While implementing the color image quantization using BFO we have considered the LAB color model as compared to RGB color model. The LAB color model is considered because this model was designed to be device independent. In other words by means of this model you can handle colors regardless of specific devices (such as monitors, printers, or computers) [6] [7].

1.4 CMC distance

CMC is a modification of CIELAB developed by the Color Measurement Committee of the Society of Dyers and Colorists. Color difference calculated using CMC method are believed to correlate better with visual assessment than color differences calculated using other instrumental systems [4]. It has two parameters: lightness (l) and chroma (c).

$$\Delta L_{cmc} = \frac{\Delta L^*}{lSL}$$

$$\Delta C_{cmc} = \frac{\Delta C^*}{cSC}$$

$$\Delta H_{cmc} = \frac{\Delta H^*}{SH}$$

$$\Delta E_{cmc} = \sqrt{\left(\frac{\Delta L^*}{lSL}\right)^2 + \left(\frac{\Delta C^*}{cSC}\right)^2 + \left(\frac{\Delta H^*}{SH}\right)^2}$$

Where

$$SL = \frac{0.040975L^*}{1 + 0.01765L^*} \quad \text{For } L^* > 16$$

$$SC = \frac{0.0638C^*}{1 + 0.0131C^*} + 0.638$$

Where

$$SH = (FT + 1 - F)SC$$

$$F = \sqrt{\frac{C^{*4}}{C^{*4} + 1900}}$$

$$T = 0.36 + |0.4 \cos(35 + h)|$$

For

$$h = 164^\circ \text{ or } h > 345^\circ$$

$$T = 0.56 + |0.2 \cos(168 + h)|$$

For $164^\circ < h < 345^\circ$

$$\Delta L^* = (cf)lSL$$

$$\Delta C^* = (cf)cSC$$

$$\Delta H^* = (cf)SH$$

L^* and C^* are from CIEL $^*C^*h$.

ΔE_{cmc} is the total color difference. ΔE_{cmc} Value allow the evaluation of the acceptability of a color match without regard to the color of the standard (e.g. two reds that have a ΔE_{cmc} of 0.5 have the same amount of visual color difference as two blues that have a ΔE_{cmc} of 0.5). As a result, a single ΔE_{cmc} limit value may be set to be used in evaluating color matches [8].

In this research work, the fitness function is taken as CMC distance to find out the distance between two food sources i.e. colors. Color difference calculated using CMC method are believed to correlate better with visual assessment than color differences calculated using other instrumental systems. The CMC distance formula shows convincing results on its property to better characterize perceptual color similarity. The rest of the paper is organized as follows. Section 2 surveys related work in the field of color image quantization. The proposed algorithm is presented in section 3, while an experimental evaluation of the algorithm is provided in section 4. Finally, section 5 concludes the paper and provides guidelines for future research.

2. RELATED WORK IN THE FIELD OF COLOR IMAGE QUANTIZATION

Several heuristic techniques for color image quantization have been proposed in the literature. Some of them are discussed below.

Static color look-up table divide the RGB cube into equal slices in each dimension. The cross product of this color levels can be used as the entities of the color look-up table. This kind of quantization can be made for the red axis and for the green axis into 8 levels each and the blue axis (the human is less sensitive to blue) into 4 levels. So that $8 * 8 * 4 = 256$ colors are uniformly spread over the color space are available. The mapping of an image value into a value out of this selection is simply done by rounding every component. An important drawback of this method is the artifact of appearing edges in the image and speaking the procedure of finding the best entries for the look up table is time and memory consuming [16].

The median cut algorithm (MCA) divides the color space repeatedly along the median into rectangular boxes until the desired number of colors is obtained [11].

Popularity Algorithm builds the color map by finding the K most frequently appearing colors in the original image. Therefore the colors are stored in a histogram. The K most frequently occurring colors are extracted and they are made the entries in the color table. Now the true image can be quantized [16].

The variance-based algorithm (VBA) [19] also divides the color space into rectangular boxes. However, in VBA the box with the largest mean squared error between the colors in the box and their mean is split.

The octree quantization algorithm [11] repeatedly subdivides a cube into eight smaller cubes in a tree structure of degree eight. Then adjacent cubes with the least number of pixels are merged. This is repeated until the required number of colors is obtained [11] [3].

Rui et al. [14] presented an initialization and training method for Self Organizing Maps (SOM) that reduces the computational load of SOM and at the same time generates reasonably good results [11].

M. G. Omran [11] in his paper proposes Color image quantization based on PSO. The proposed approach is of the class of quantization techniques that performs clustering of the color space. The proposed algorithm randomly initializes each particle in the swarm to contain K centroids (i.e. color triplets). The K-means clustering algorithm is then applied to each particle at a user-specified probability to refine the chosen centroids. Each pixel is then assigned to the cluster with the closest centroid. The PSO is then applied to refine the centroids obtained from the K-means algorithm.

3. PROPOSED ALGORITHM

Bacteria Foraging Optimization is a population oriented algorithm used to search optimal solution. In this research each Pixel of the image is considered as bacteria food and the color of the pixel is considered as bacteria food. The aim of the proposed algorithm is to minimize the food sources i.e. to reduce the number of colors in the image. In this research, all the pixels initially have some color and the purpose of this research is to optimize the number of colors in the image. All the colors in the image are evaluated as the number of pixels having that color. This evaluation defines the health status of all the colors present in the image. Depending upon the health status of the colors, all the colors in the image are divided into two categories popular colors and unpopular colors. If the health status of the color is high i.e. the color is present on too many pixels then that color is considered as popular color and all other colors whose health status is poor are considered unpopular colors. All the pixels in the image are compared with every other pixel in the image to find the most similar color to be eliminated. The fitness function is taken as CMC distance to find out the distance between two food sources i.e. colors. In this research we have considered CMC threshold value as 0.7 to further optimize the colors in the image. After comparing the colors of all the pixels in the image if the CMC distance between two colors is found less than 0.7 then the color with which the CMC distance is found and all other colors having that color are eliminated with the other color. After this elimination process the health status of all the colors is evaluated again because after elimination the health status of colors may change. After the elimination process, the unpopular colors are compared and if the CMC distance between two unpopular colors is found less than 0.7 then those two colors are combined to produce a new color. This process of producing the new color is called as reproduction. The colors from which the new color is produced are killed.

This new color is now dispersed at the pixels where the parents of new color were present.

BFO Consist of following basic principal mechanisms:-

- Chemo-taxis.
- Elimination.
- Reproduction.
- Dispersal.

3.1 Chemo-taxis

The motion patterns that the bacteria will generate in the presence of chemical attractants and repellents are called chemo-taxis. For E. coli, this process was simulated by two different moving ways: run or tumble. A Bacterium alternates between these two modes of operation its entire lifetime. The bacterium sometimes tumbles after a tumble or tumbles after a run. This alternation between the two modes will move the bacterium, and this enables it to "search" for nutrients (Chan et.al. 2009). In this research, each bacteria takes a unit step of size one in the same direction to find its nutrient i.e. each pixel takes a unit step of size one to find the most similar color. If the pixel find the most similar color after a unit walk fulfilling the fitness function i.e. CMC distance then it is called as swim where the pixel color is replaced with the color of that next pixel.

For two color of respective C, E Lab components (L_1, a_1, b_1) and (L_2, a_2, b_2) , CMC metrics define three components for distance measure as follow:

Chroma difference:

$$\Delta E = \sqrt{(\Delta H/SH)^2 + (\Delta L/L.SL)^2 + (\Delta C/C.SC)^2} \quad \dots(3.1)$$

$$\text{Where } \Delta C = \sqrt{a_1^2 + b_1^2} - \sqrt{a_2^2 + b_2^2}$$

$$\text{Lighting difference } \Delta L = L_1 - L_2$$

Here perceptual difference

$$\Delta H = \sqrt{(\Delta a)^2 + (\Delta b)^2} - (\Delta c)^2$$

I & C are application dependent coefficient. ΔE here represents the CMC distance between two colors.

If the most similar color is not found at the immediate next pixel position then the bacterium i.e. the pixel run to the next pixel positions with the unit steps, to find the most similar color. This process of swimming continued till the maximum number of similar colors is found.

3.2 Elimination

Elimination is performed in two steps. Primary elimination and secondary elimination.

Primary Elimination: In primary elimination if a pixel in the image found similar color following the fitness function then one of them becomes candidate pixel for the elimination.

Secondary Elimination: In secondary elimination firstly the health status of all the colors in the image is evaluated. Then based on the health status the colors are divided into two categories surviving i.e. popular and the un-surviving i.e. unpopular colors. The un-surviving colors following the fitness function become candidate for the elimination.

In this research, after comparing the colors of all the pixels in the image the elimination of colors in this step is based on the primary elimination.

3.3 Reproduction

All the colors in the image are evaluated as the number of pixels having that color.

$$\text{Health status} = \frac{N_i}{S}$$

Where N represents the number of pixels having i^{th} color. And S represents total number of pixels in the image. This evaluation defines the health status of all the colors present in the image. Depending upon the health status of the colors, all the colors in the image are divided into two categories surviving colors and un-surviving colors. If the health status of the color is high then that pixel is considered as surviving color and all other colors whose health status is poor are considered un-surviving colors. The unpopular colors are compared and if the CMC distance between two unpopular colors is found less than 0.7 then those two colors are combined to produce a new color. This process of producing the new color is called as reproduction.

3.4 Dispersal

As explained above in the reproduction, we can add new colors to our color palette. The un-surviving colors from which the new color is produced are eliminated. Elimination in this step is performed according to the secondary elimination. This new color is now dispersed i.e. allocated to the parents of new color. In 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. In this research, the colors with poor health status are eliminated and the colors with high health status are kept. The new colors dispersed to other pixels in the image where the parents of new color were present.

In the classical BFO, only the first half of population survives. In this research, instead of killing bacteria population the food sources are killed and reproduced. BFO has been implemented on fifteen images. It has been validated by applying the algorithm on phantom images.

3.5 Proposed algorithm

Step1. Initialize parameters

$S, i, N_s, N_c, k, l, n_i, N_u, \Delta E(k_i, k_i + 1)$. Where

S : Total number of pixels in the image (total number of bacteria).

i : Total colors in the image (number of food sources).

N_s : Number of swim steps ($N_s = 1$).

N_c : Number of chemo-tactic steps ($N_c = 1$).

k_i : Color of the current pixel (Current bacteria).

l : New color (new food source).

n_i : Number of pixels having same color (Number of bacteria following i th food source).

N_u : Number of pixels having unpopular color (Total number of bacteria with unpopular food source).

$\Delta E(k_i, k_i + 1)$: This is $\text{CMC}(k_i, k_i + 1)$ i.e. CMC distance between pixel's current color and the next pixel's color (i.e. CMC distance between Bacteria's current food source and nearest food source).

Food sources are divided into categories popular and unpopular depending upon how many bacteria are moving toward that particular food source. Colors in the image are

divided into two categories surviving colors and un-surviving colors depending upon how many pixels have the similar color.

Step 2. Chemo-tactic step: Compute

$$\Delta E = \sqrt{(\Delta H / SH)^2 + (\Delta L / l.SL)^2 + (\Delta C / C.SC)^2}$$

$$\text{Where } \Delta C = \sqrt{a_1^2 + b_1^2} - \sqrt{a_2^2 + b_2^2}$$

Lighting difference $\Delta L = L_1 - L_2$

Here perceptual difference

$$\Delta H = \sqrt{(\Delta a)^2 + (\Delta b)^2} - (\Delta c)^2$$

I & C are application dependent coefficient. ΔE here represents the CMC distance between two colors.

Step 2. Elimination step

For $k = 1 \dots S$. Take a chemo-tactic step of size one for pixel k as follows:

If $\Delta E(k_i, k_i + 1) \leq 0.7$

Eliminate k_i pixel's color and all other pixels

having i^{th} color with $k_i + 1$ pixel's color.

Else

$k = k + 1$

END

END

Step 3. Reproduction and dispersal step

$$\text{Health status} = \frac{N_i}{S}$$

Categorize the colors in the image into two categories popular and unpopular depending upon the health status.

Substep 3.1 For $k = 1 \dots N_u$ Take a chemo-tactic step of size one for pixel k as follows:

If (k_i = popular)

Continue;

Else $\Delta E(k_i, k_i + 1) \leq 0.7$

$$l = \frac{(k_i) + (k_i + 1)}{2} \dots (3.2)$$

Eliminate $k_i + 1$ pixel's color and k_i pixel's color. Disperse l^{th} color at the pixels were the parents of l^{th} color were.

END

END

4. RESULTS AND DISCUSSIONS

Our objective is to use the proposed Bacteria Foraging Optimization algorithm for Color image quantization. BFO has been implemented on fifteen images. It has been validated by applying the algorithm on phantom images. Images created entirely by digital means are called as phantom images. Phantom images are also called as computer generated images. This category collects images that are scans, screen captures, photos, and/or illustrations of The Phantom and related characters and intellectual properties. The following

figures show input image with original number of colors and resulting image with quantized colors.



Figure 4.1: Original image 'desert.png' with 6481 colors on left side and Quantized image 'desert.png' with 4676 colors on right side.



Figure 4.2: Original image 'flower1.jpg' with 9048 colors on left side and Quantized image 'flower1.jpg' with 5948 colors on right side.



Figure 4.3: Original image 'flower2.jpg' with 13357 colors on left side and Quantized image 'flower2.jpg' with 8629 colors on right side.



Figure 4.4: Original image 'image3.bmp' with 15116 colors on left side and Quantized image 'image3.bmp' with 10489 colors on right side.



Figure 4.5: Original image 'phantom6.jpg' with 8036 colors on left side and Quantized image 'phantom6.jpg' with 4948 colors on right side.

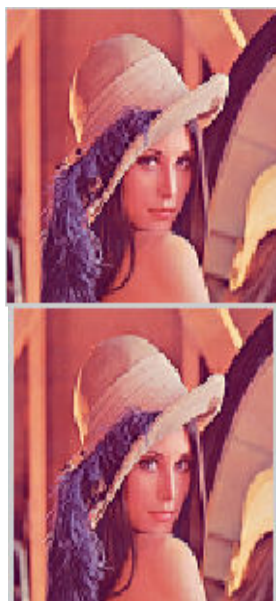


Figure 4.6: Original image 'lenna.png' with 9889 colors on left side and Quantized image 'lenna.png' with 5779 colors right side.

4.1 Histograms of R component

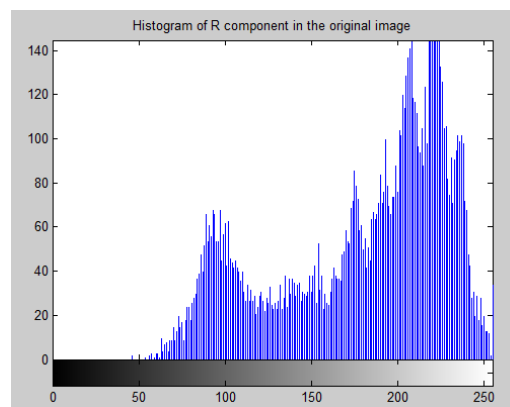


Figure 4.7: Histogram of original image 'lenna.png'.

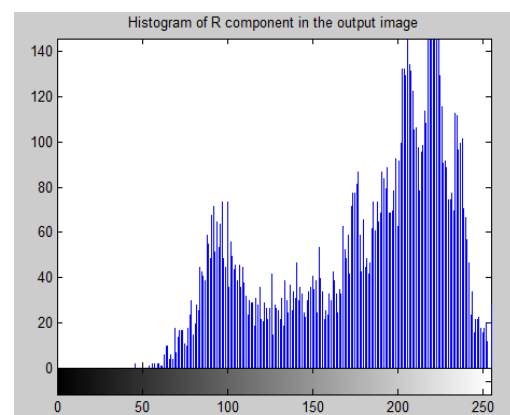


Figure 4.8: histogram of quantized image 'lenna.png'.

4.2 Histograms of G component

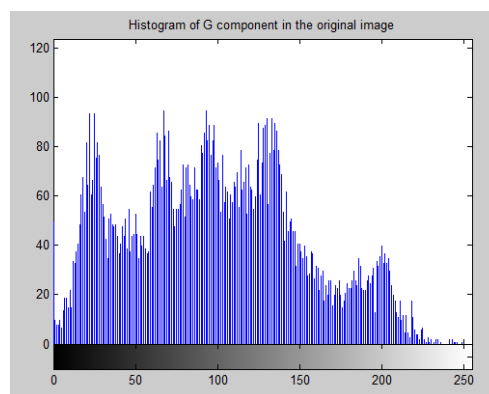


Figure 4.9: histogram of original image 'lenna.png'.

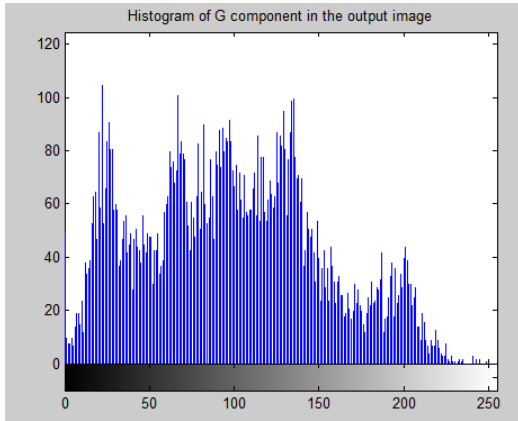


Figure 4.10: Histogram of quantized image 'lenna.png'

4.3 Histograms of B component

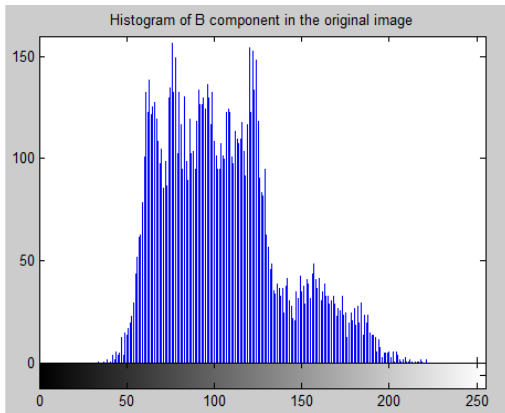


Figure 4.11: Histogram of original image 'lenna.png'

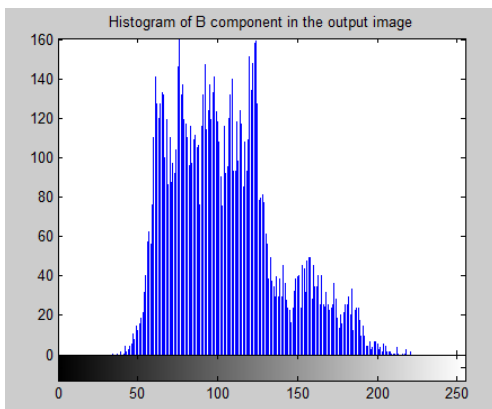


Figure 4.12: Histogram of quantized image 'lenna.png'

Here the histograms of R,G,B components of original and quantized image named 'lenna.png' shows visual similarity which represents that the image quality is preserved. The histograms shown above represents that very similar color has been replaced using the proposed algorithm. Image quality can be visually analyzed from the histograms of input and output image. The computational results which have been obtained using the proposed algorithm are shown below in a

table. These results have been analyzed based on LMSE and Euclidean metric.

Table 4.1 Computational Result & Analysis of results based on LMSE and Euclidean metric.

Name of the image	Original no. of colors	Colors after quantization	Euclidean distance	LMSE	Scaling coefficients
desert.png	6481	4676	157.04	0.0198	0.99, 0.06
flower1.jpg	9048	5948	248.87	0.0208	1.00,-0.02
flower2.jpg	13357	8629	247.45	0.0148	0.99, 0.05
image3.bmp	15116	10489	233.25	0.0136	0.99, 0.18
Lenna.png	9889	5779	238.62	0.0225	0.998,-0.18

From the above results it can be observed that perceptual uniformity is there in the output image. There is no degradation in the image quality i.e. The image perception from the viewer point of view is same in both. The difference between the colors in the input image and the processed image shows the total number of colors reduced. The processed image is visually similar to the input image. The performance of proposed algorithm is evaluated based on LMSE and Euclidean distance. In this research work the results which have been achieved using BFO for color quantization are compared with the technique based on Biogeography based optimization. The results of the proposed algorithm are analyzed by comparing it with the existing algorithm Biogeography Based Optimization and by implementing on following images. The following figures shows the processed images based on BFO and processed images based on BBO.



Figure 4.13: Quantized image 'lenna.png' with 5779 colors using BFO on left side and Quantized image 'lenna.png' with 8616 colors using BBO on right side.



Figure 4.14: Quantized image ‘phantom6.jpg’ with 4948 colors using BFO on left side and Quantized image ‘phantom6.jpg’ with 7354 colors using BBO on right side.

From the results shown above, the performance of BFO algorithm for image quantization can be compared with the technique based on BBO. The results obtained by using Bacteria foraging optimization are comparatively better than the results obtained by the results which have been obtained using Biogeography Based Optimization.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented Bacteria Foraging Optimization algorithm for color image quantization. The results presented are preliminary and there is a lot of scope for improvement to develop this algorithm. Based on the results presented in the previous chapter, I conclude that the image quantization based on Bacteria foraging optimization gives better results than the technique based on BBO. The BFO-CIQ uses the Lab color space. Although the Lab color model eliminates the weakness of RGB model. One of the weaknesses of RGB is that equal distances in the RGB color space may not correspond to equal distance in color perception. The advantage of LAB color model over RGB model is that LAB color model is device independent. In other words by means of this model you can handle colors regardless of specific devices (such as monitors, printers, or computers). In this research, BFO has been implemented on various types of images including the phantom images. This validates the proposed algorithm and it gives optimized results when implemented on the phantom images.

5.1 Future Work

In the proposed algorithm we have to consider each pixel and for large images the proposed algorithm may become slow. So the further research may focus on some modification of the proposed algorithm to enhance the speed. Further research work may focus on developing some new algorithms related to bacterial foraging to decrease the computational cost and time during global optimization. Also some modifications of the threshold value can be undertaken to enhance the performance of the said algorithm in the field of image quantization and image processing. Future research may try to apply the BFO-CIQ to other color spaces.

6. ACKNOWLEDGMENTS

I am extremely grateful to **Assistant Professor Akshay Girdhar, Head Of the Department I.T., GNDEC, LUDHIANA** helping in carrying the present work and is acknowledged with reverential thanks. I am extremely grateful to **Ms. Surbhi Gupta Senior lecturer RIEIT, Railmajra.** The help rendered by **Ms. Surbhi Gupta Senior lecturer RIEIT, Railmajra** for the experimentation is greatly acknowledged. Without the wise counsel and able guidance, it would have been impossible to complete this work in this manner.

7. REFERENCES

- [1] Bremermann H. J. and Anderson R.W., “An alternative to back- propagation: a simple rule of synaptic modification for neural net training and memory,” Tech. Rep. PAM-483, Cent for Pure and Applied Mathematics, University of California, San Diego, Calif, USA, 1990.
- [2] Chan H., Zhu Y., and Hu K., 2009 “ Cooperative Bacterial Foraging Optimization” Hindawi Publishing Corporation Discrete Dynamics in Nature and Society Volume2009, Article ID 815247, 17 pages.
- [3] Dekker A (1994) Kohonen neural networks for optimal colour quantization, Network: Computation in Neural Systems 5: 351- 367.
- [4] Foley J. D., Van Dam A., Feiner S. K., Hughes J. F., Computer Graphics - Principles and Practice, Second Edition, Addison- Wesley Publishing Company, 1990, ISBN 0-201- 12110-7.
- [5] Freisleben B, Schrader A (1997) An evolutionary approach to color image quantization, Proceedings of IEEE International Conference on Evolutionary Computation, 459-464.
- [6] Hunter Lab, L*a*b* versus CIE (1976) L*a*b* (http://www.hunterlab.com/appnotes/an02_01.pdf) (PDF)
- [7] Hunter labs (1996) “ Hunter lab color scale” Insight on color 8 9 (August 1-15, 1996). Reston, VA, USA: Hunter Associates Laboratories
- [8] Hunter lab Application notes (2008) “Insight on color CMC” Vol 8, No 13
- [9] Kim D. H. and Cho J. H., “Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization,” in Proceedings of the 3rd International Atlantic Web Intelligence Conference (AWIC ’05), vol. 3528 of Lecture Notes in Computer Science, pp. 231–235, Lodz, Poland, June 2005.
- [10] Kim D. H. and Cho C. H., “Bacterial foraging based neural network fuzzy learning,” in Proceedings of the Indian International Conference on Artificial Intelligence, pp. 2030–2036, Pune, India, December 2005.
- [11] Mahamed G. Omran and Andries P. Engelbrecht, Ayed Salman “A Color Image Quantization Algorithm Based on Particle Swarm Optimization Informatica (2005) 261– 269.
- [12] Mishra S., “A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation,” IEEE Transactions on Evolutionary Computation, vol. 9, no. 1, pp. 61–73, 2005.

- [13] Passino K. M., “Biomimicry of bacterial foraging for distributed optimization and control,” IEEE Control Systems Magazine, vol.22, pp. 52–67, 2002.
- [14] Rui X, Chang C, Srikanthan T (2002) On the initialization and training methods for Kohonen self-organizing feature maps in color image quantization, Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications.
- [15] Scheunders P (1997) A genetic C-means clustering algorithm applied to color image quantization, Pattern Recognition 30(6): 859-866.
- [16] Segenchuk S. “An Overview of Color Quantization Techniques”
- [17] Tripathy M., Mishra S., Lai L. L., and Zhang Q. P., “Transmission loss reduction based on FACTS and bacteria foraging algorithm,” in Proceedings of the Parallel Problem Solving from Nature (PPSN '06), pp. 222–231, Reykjavik, Iceland, September 2006.
- [18] Velho L, Gomes J, Sobreiro M (1997) Color image quantization by pairwise clustering, Proceedings of the 10th Brazilian Symposium on Computer Graphics and Image Processing, 203- 207.
- [19] Wan S, Prusinkiewicz P, Wong S (1990) Variancebased color image quantization for frame buffer display, Color Research and Application 15(1): 52- 58.