Image Quantization using HSI based on Bacteria Foraging Optimization

[1]**Dharminder Kumar**, [2]**Vinay Chopra**

[1]Student, M.Tech., [2]Assistant Professor,

Department of Computer Science & Engg,

D.A.V. I.E.T., Jalandhar, Punjab

Abstract: *Bacteria Foraging Optimization a nature-inspired optimization has drawn the attention of researchers because of its efficiency in solving real-world optimization problems arising in several application domains. Color image quantization is an important process of representing true color images using a small number of colors. Existing color reduction techniques tend to alter image color structure and distribution. Thus the researchers are always finding alternative strategies for color quantization. In cylindrical color spaces like HSI, color is represented by hue, saturation and intensity. These components are closer to the way human perceives and describes color. Hue, saturation and intensity can also reveal image features that are not so obvious in other color spaces. The objective of this research work, is to design an algorithm for Image Quantization using HSI color space based on Bacteria Foraging Optimization. To implement and test the proposed algorithm. To compare the designed algorithm with other quantization techniques. The conducted experiments indicate that proposed algorithm generally results in a significant improvement of image quality compared to other well-known approaches.*

Keywords: Color reduction, Bacteria Foraging Optimization, HSI color space, Euclidean distance, Swarm intelligence.

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, 9, 18]. 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 [12], harmonic estimation [14], transmission loss reduction [16] and machine learning [13].

1.1 Bacterial Foraging Optimization

Algorithm

During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an *E.coli* bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the above-mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment. When they get food in sufficient, they are increased in length and in presence of suitable temperature they break in the middle to from an exact replica of itself. This phenomenon inspired Passino to introduce an event of reproduction in Bacteria Foraging Optimization algorithm. Due to the occurrence of sudden environmental changes or attack, the chemotactic progress may be destroyed and a group of bacteria may move to some other places or some other may be introduced in the swarm of concern. This constitutes the event of elimination-dispersal in the real bacterial population, where all the bacteria in a region are killed or a group is dispersed into a new part of the environment.

The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely, chemo taxis, reproduction, and elimination-dispersal. These are described as follows [18].

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

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

Where Δ (*i*) is the direction vector of the *j*th chemotactic step. When the bacterial movement is *run*, Δ (*i*) is the same with the last chemotactic step; otherwise, Δ (*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[18].

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 [18].

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 [18].

1.1.4 BFO Algorithm

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

Step 1. Initialize parameters n, S, N_c, N_s, N_{re}, N_{ed}, P_{ed},
C(i) (i = 1, 2, . . . , 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= 1, 2, . . ., 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 *last* $j = J(i, j, k, l)$ to save this value since we may find better value via a run.

Substep 4.4. Tumble. Generate a random vector

$$\mathbf{\Delta}(i) \varepsilon\ x R^n$$

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

Substep 4.5. Move. Let

$$\emptyset^i(j+1,k,l) = \emptyset^i(j,k,l) + C(i)\frac{\mathbf{\Delta}(i)}{\sqrt{\mathbf{\Delta}^T(i)\mathbf{\Delta}(i)}} \qquad ...(1.2)$$

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 $\emptyset^i(j+1,k,l)$

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.

- ϕ^i (*j* +1, *k, l*) to compute the new *J*(*i, j*+ 1, *k, l*).

- Else let *m* = *s N* .

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, . . . , *S,* let

$$J^i_{health} = \sum_{j=1}^{N_c} J(i,j,k,l) \qquad ....(1.3)$$

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, . . . , *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 [15].

1.2 Color image quantization

Color image quantization is an important process of representing true color images using a small number of colors. With a good color quantization algorithm and some lossy compression algorithms (such as ones used by .jpg formats), the same image quality (at least visually) can mostly be restored from a much smaller file. The color image quantization can reduce not only storage requirement but also the transfer time of the image. These reductions are quite important for multimedia applications in the Internet where the communication delays are very concerned. Moreover, the color image quantization can be implemented as a preprocessing step for image compression algorithm.

- The color image quantization algorithm typically consists of four phases.
- The first phase, called sampling the original image, computes the image histogram for color statistics i.e. a number of distinct colors and their frequencies.
- The second phase, called colormap design, chooses the best possible set of representative colors from the color statistics.
- The third phase, called pixel mapping, maps each color in the original image to a representative color in the colormap.
- The fourth phase, called image quantizing, redraws the image by replacing the original color in every pixel with a representative color. 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.

- It is suitable for mobile and hand-held devices where memory is usually small [10].
- It is suitable for low-cost color display and printing devices where only a small number of colors can be displayed or printed simultaneously [7].
- Most graphics hardware use color lookup tables with a limited number of colors [6].

1.3 HSI color model

The choice of the color space can be a very important decision which can dramatically influence the results of the processing. The knowledge of various color spaces can ease the choice of the appropriate colour space. *Color* is the way the HVS (the *human visual system*) measures a part of the electromagnetic spectrum, approximately between 300 and 830 nm. A *color space* is a notation by which we can specify colours, ie the human perception of the visible electromagnetic spectrum.

When humans view a color object, we tend to describe it by its hue, saturation, and brightness.

Hue is an attribute that describes a pure color.

saturation gives a mesaure of the degree to which a pure color is diluted by white light.

Brightness is a subjective descriptor that is practically impossible to measure.

It embodies the achromatic description of *intensity* and is a key factor in describing color sensation. We do know that intensity (gray level) is a most useful descriptor of monochromatic images. This quantity definitely is measurable and easily interpretable. The Hue component describes the color itself in the form of an angle between [0,360] degrees. 0 degree mean red, 120 means green 240 means blue. 60 degrees is yellow, 300 degrees is magenta. The Saturation component signals how much the color is polluted with white color. The range of the S component is [0,1]. The Intensity range is between [0,1] and 0 means black, 1 means white.

While implementing the color image quantization using Bacteria Foraging Optimization we have considered the HSI color model as compared to RGB and LAB color model. RGB and LAB color space has some drawbacks which make researchers look to the other color spaces in computer vision tasks. One drawback is high correlation between individual components caused by aliasing of spectral sensitivity curves of three types of cones. Further, the individual components does not correspond the way human perceives and describes colors [1]. For example, it is hard to say, solely looking at the color, how much of individual components comprise the color. In cylindrical color spaces like HSI color is represented by hue, saturation and intensity (value, brightness). These components are closer to the way human perceives and describes color. Hue, saturation and intensity can also reveal image features that are not so obvious in RGB and LAB color space. Also, in HSI color space chromatic (hue and saturation) and achromatic (intensity) information are separated.

1.4 Euclidean distance

A central problem in image recognition and computer vision is determining the distance between images. Considerable efforts have been made to define image distances that provide intuitively reasonable results.

Among all the image metrics, Euclidean distance is the most commonly used due to its simplicity. The key advantages of this metric are:

1) Relative insensitivity to small perturbation (deformation);
2) Simplicity of computation;
3) It can be efficiently embedded in most of the powerful image recognition techniques.

Euclidean distance between two points in HSI color model is defined as follows:

$$F_{1}1 = (H_{1}1, S_{1}1, I_{1}1\ )$$
$$F_{1} = (H_{2}, S_{2}, I_{2})$$
$$\Delta HSI(F_{1}, F_{2}) = \sqrt{(\Delta a)^{2} + (\Delta b)^{2} + (\Delta c)^{2}} \qquad ..1.4$$

Where

$$\Delta a = (I_{1} - I_{2})^{\square}\ ,$$
$$\Delta b = (S_{1}\cos(H_{1}) - S_{2}\cos(H_{2}))^{\square}\ ,$$
$$\Delta c = (S_{1}\sin(H_{1}) - S_{2}\mathbf{sin}(H_{2}))^{\square}$$

∆HSI is the total color difference. A single ∆HSI limit value may be set to be used in evaluating color matches [8]. In this research work, the fitness function is taken as Euclidean distance to find out the distance between two food sources i.e. colors. Color difference calculated using Euclidean distance method are believed to correlate better with visual assessment than color differences calculated using other instrumental systems.

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.

The popularity algorithm generates the colormap by finding the densest regions in color distribution of the image. Hence, it simply selects the K colors with the highest occurrences from the image histogram and uses these K colors as the representative colors in the colormap[23].

The median-cut algorithm uses the splitting approach to repeatedly divide the color space into two smaller individual cells containing an approximately equal number of pixels at each step. The orientation of cutting plane is normal to one of the coordinate axes with a largest range of image pixels and pass through the median point of the color distribution projected on this axis. At the end of this operation, the final cells contain an equal number of image pixels[11].

The variance-based algorithm is schematically similar to the mediancut algorithm, with an exception that, at each step, a cell for further partition is the cell with the largest weighted variances of color distribution. The cutting plane is chosen to be perpendicular to the coordinate axis where the expected variance is most reduced[2].

The octree algorithm relies on a tree structure. The root of the octree is an entire cell and at each level of the tree each node has eight successors. The maximum depth of the octree is 8. At level 8, the terminal nodes of the octree are individual colors. The octree is then reduced by a process that replaces the terminal node with their parent node containing an average of the color in the terminal node. This process continues until the number of terminal nodes is equal K. Finally, the K terminal nodes are chosen as the representative colors in the colormap[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 Kmeans algorithm.

[21] In this paper the bacteria foraging optimization is applied to LAB color model for image quantization. The CMC distance metric is used for color difference between pixels. Color elimination and reproduction is done by evaluating the CMC distance. The threshold value of CMC distance is used for comparisons.

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 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 Euclidean distance to find out the distance between two food sources i.e. colors. Based on the value of euclidean distance between similar colors elimination of one of the colors is done. 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 based on euclidean distance values to combine colors 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.

BFO Consist of following basic principal echanisms:-

- 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. 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. Euclidean 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 HSI image components ($H_1$1, $S_1$1, $I_1$1) and (H_2, S_2, I_2), Euclidean distance metrics define two components for distance measure as follow:

$$\Delta HSI\big((H_1, S_1, I_1), (H_2, S_2, I_2)\big) = \sqrt{(\Delta a)^2 + (\Delta b)^2 + (\Delta c)^2}$$

Where

$$\Delta a = (I_1 - I_2)^{\square} \ ,$$

$$\Delta b = (S_1 \cos(H_1) - S_2 \cos(H_2))^{\square} \ ,$$

$$\Delta c = (S_1 \sin(H_1) - S_2 \mathbf{sin}(H_2))^{\square}$$

ΔHSI here represents the Euclidean 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 is 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.

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 Euclidean distance between two unpopular colors is found less than threshold value 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 and validated on by applying the algorithm on images as well phantom images.

3.5 Proposed algorithm

Step1. Initialize parameters

S , i , N_s , N_c , k , l , n , N_u , $\Delta HSI\ (k_i,\ k_{i+1})$

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).

$\Delta HSI\ \big((k_{i1},\ k_{i+1})\big)$: This is euclidean 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

$$\begin{aligned}&\boldsymbol{\Delta} HSI\big((H_1, S_1, I_1), (H_2, S_2, I_2)\big)\\&= \sqrt{(\boldsymbol{\Delta} a)^2 + (\boldsymbol{\Delta} b)^2 + (\boldsymbol{\Delta} c)^2}\end{aligned} \qquad ...3.1$$

Where

$$\boldsymbol{\Delta} a = (I_1 - I_2)^{\square} \ ,$$

$$\boldsymbol{\Delta} b = (S_1 \cos(H_1) - S_2 \cos(H_2))^{\square} \ ,$$

$$\boldsymbol{\Delta} c = (S_1 \sin(H_1) - S_2 \,\mathbf{sin}(H_2))^{\square}$$

Step 2. Elimination step

```
For $k = 1.........S$. Take a chemo-tactic step of size one for pixel $k$ as follows:
    If $\Delta HSI \big((k_{i1}, k_{i+1})\big) <= D_{max}$
        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 $k = 1.........N_u$ Take a chemo-tactic step of size one for pixel $k$ as follows:
    If ( $k_i$ = popular)
    Continue;
Else If $\Delta HSI \big((k_{i1}, k_{i+1})\big) <= D_{max}$
    $l = \frac{(k_i) + k_i + 1}{2}$                ...3.2
    Eliminate  pixel's color and  pixel's color. Disperse $l^{th}$ color at the pixels were the parents of $l^{th}$ color were.
    END
END
```

D_{max} is a constant to be chosen while evaluating similarity.

4. Results And Discussions

Our objective is to use the proposed Bacteria Foraging Optimization algorithm for Color image quantization using HSI color model. Bacteria Foraging Optimization using HSI color model has been validated by using D_{max} = 0.205 and applying the algorithm on images as well as phantom images by varying the size of image and number of bacteria. 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.

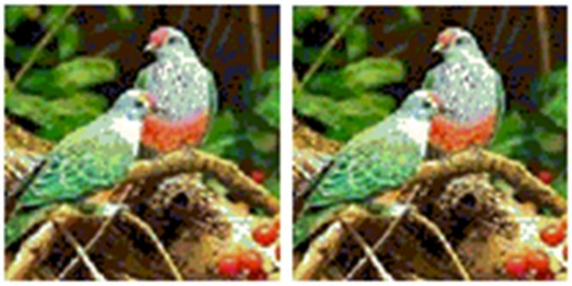


Figure 4.1:
Original image 'Image1.bmp' with 9719 colors on left and Quantized image 'Image1.bmp' with 5529 colors on right.

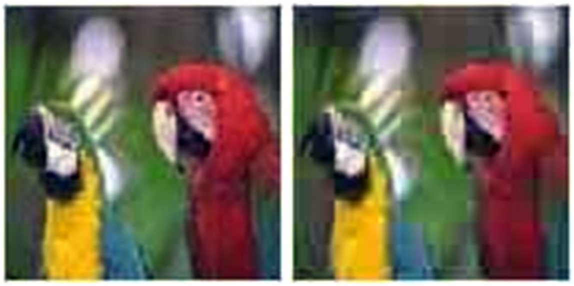


Figure 4.2:
Original image ‘Parrots.jpg’ with 8806 colors on left
and Quantized image ‘Parrots.jpg’ with 5503 colors on right.

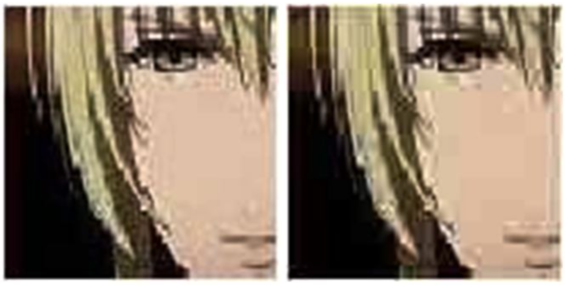


Figure 4.3:
Original image ‘Phantom1.jpg’ with 4721 colors on left
and Quantized image “Phantom1.jpg’ with 2974 colors on right.

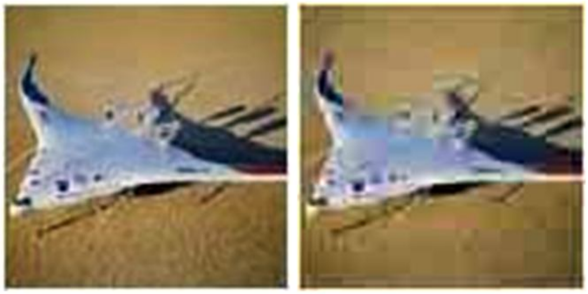


Figure 4.4:
Original image ‘Phantom2.jpg’
with 5918 colors on left
and Quantized image ‘Phantom2.jpg’ with 3754 colors on right.

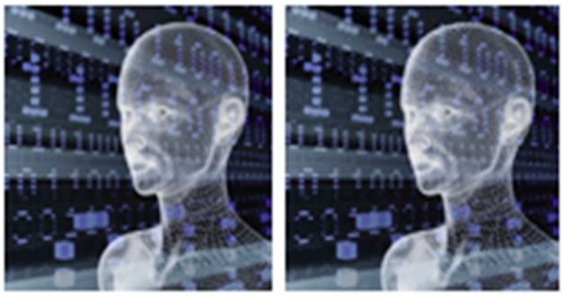


Figure 4.5:
Original image ‘Technology.png’ with 7643 colors on left
and Quantized image ‘Technology.png’ with 5042 colors on right.

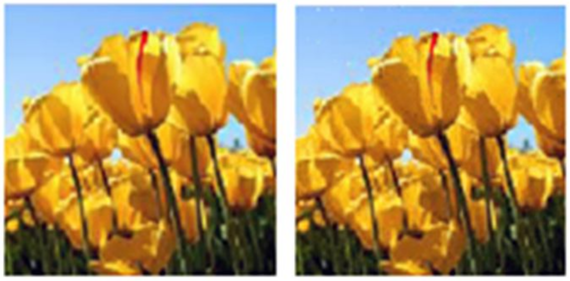
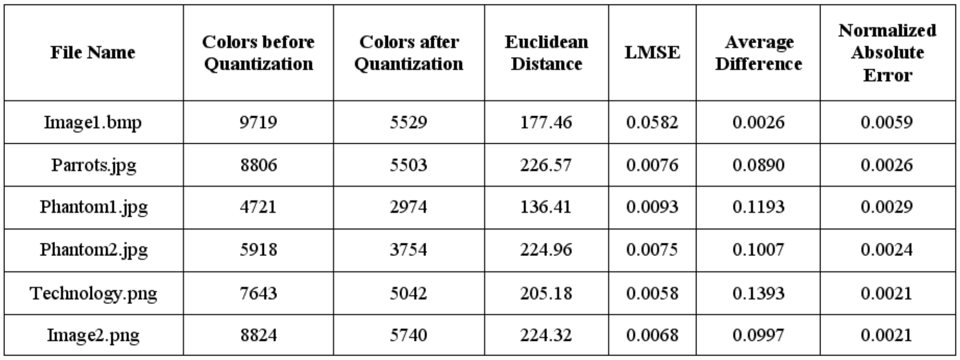


Figure 4.6: Original image ‘Image2.png’ with 8824 colors on left side and Quantized image ‘Image2.png’ with 5740 colors right side.

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, Euclidean distance for images, Normalized Absolute Error and Average Difference.

Table 4.1 Computational Result & Analysis of results based on LMSE, Euclidean distance, Average Difference and Normalized Absolute Error



File Name	Colors before Quantization	Colors after Quantization	Euclidean Distance	LMSE	Average Difference	Normalized Absolute Error
Image1.bmp	9719	5529	177.46	0.0582	0.0026	0.0059
Parrots.jpg	8806	5503	226.57	0.0076	0.0890	0.0026
Phantom1.jpg	4721	2974	136.41	0.0093	0.1193	0.0029
Phantom2.jpg	5918	3754	224.96	0.0075	0.1007	0.0024
Technology.png	7643	5042	205.18	0.0058	0.1393	0.0021
Image2.png	8824	5740	224.32	0.0068	0.0997	0.0021

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. The processed image is visually similar to the input image. The performance of proposed algorithm is evaluated based on LMSE, Euclidean distance for images, Normalized Absolute Error and Average Difference. In this research work the results which have been achieved using Bacteria Foraging Optimization for color quantization using HSI color model are compared with the other approaches. The results of the proposed algorithm are analyzed by comparing it with the existing techniques of color image quantization.

The following figures shows the processed images based on Bacteria Foraging Optimization for color quantization using HSI color model and processed images based on Bacteria Foraging Optimization for color quantization using LAB color model.

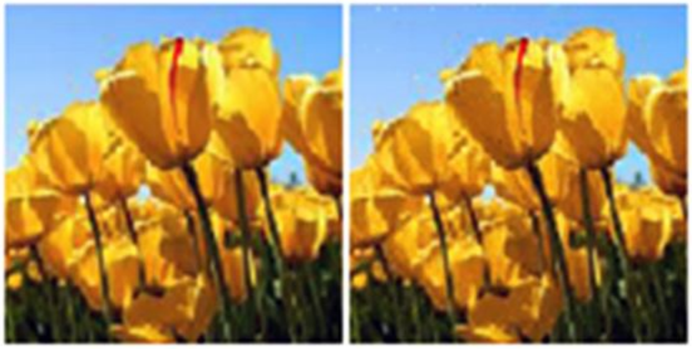


Figure 4.13:

Quantized image 'Image2.png' with 5756 colors using BFO-CIQ LAB on left and Quantized image 'Image2.png' with 5740 colors using BFO-CIQ HSI on right.

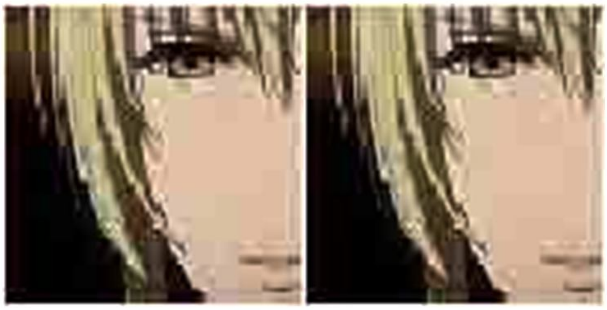


Figure 4.14:
Quantized image 'Phantom1.jpg' with 3083 colors using CIQ LAB on left and Quantized image 'Phantom1.jpg' with 2974 colors using BFO-CIQ HSI on right.

From the results shown above, the results obtained by using Bacteria foraging optimization using HSI model are comparatively better than the previous work.

5. Conclusions And Future Work

In this paper, we have presented Bacteria Foraging Optimization algorithm for color image quantization using HSI color model. Based on the results presented in the previous chapter, I conclude that the image quantization based on Bacteria foraging optimization using HSI color model gives better results. The HSI color model eliminates the weakness of RGB color model and LAB model. In HSI color model hue, saturation and intensity (value, brightness) components are closer to the way human perceives and describes color. Hue, saturation and intensity can also reveal image features that are not so obvious in RGB color space and LAB color space. In this research, Bacteria Foraging Optimization 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. Future research may try to apply the Bacteria Foraging Optimization algorithm for color image quantization to other color spaces.

6. Acknowledgements

I am extremely grateful to **Assistant Professor Vinay Chopra, Department of CSE & I.T., DAVIET, Jalandhar** helping in carrying the present work and acknowledged with reverential thanks. Without the wise counsel and able guidance, would have been impossible to complete this work in this manner.

7. References

1. Bremermann H. J. and Anderson R.W., *"An lternative to back-propagation: a simple rule of synaptic modification for neural net training and memory"*, Tech. Rep. PAM-483, Centre for Pure and Applied Mathematics, University of California, San Diego, Calif, USA, 1990.

2. Wan S., Prusinkiewicz P., Wong S.,*"Variance based color image quantization for frame buffer display"*, Color Research and Application 15(1): 52- 58, 1990.

3. Foley J. D., Van Dam A., Feiner S. K., Hughes J. F., *"Computer Graphics - Principles and Practice"*, Second Edition, Addison- Wesley Publishing Company, ISBN 0-201- 12110-7 , 1990.

4. Gonzalez and Woods, *"Derivation of HSI-to-RGB and RGB-to-HSI conversion equations"*, Digital Image Processing, Ist ed. Addison Wesley,1992

5. Evgeny Roytman, Craig Gotsman, *"Dynamic color quantization of video sequences",* IEEE transactions on visualization and computer graphics, vol. 1, No. 3, september 1995.

6. Freisleben B., Schrader A., *"An evolutionary approach to color image quantization"*, Proceedings of IEEE International Conference on Evolutionary Computation, 459-464, 1997.

7. Scheunders P.,*"A genetic C-means clustering algorithm applied to color image quantization"*, Pattern Recognition 30(6): 859-866,1997.

8. Velho L, Gomes J, Sobreiro M, *"Color image quantization by pairwise clustering"*, Proceedings of the 10th Brazilian Symposium on Computer Graphics and Image Processing, 203- 207, 1997.

9. Passino K. M., *"Biomimicry of bacterial foraging for distributed optimization and control"*,IEEE Control Systems Magazine, vol.22, pp. 52-67, 2002.

10. Rui X., Chang C., Srikanthan T., *"On the initialization and training methods for Kohonen selforganizing feature maps in color image quantization"*, Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications, 2002.

11. M. G. Omran, A. P. Engelbrecht, and A. Salman, *"A Color Image Quantization Algorithm Based on Particle Swarm Optimization",* Journal of Informatica, Vol.29, No.3, pp. 261-269, 2005.

12. 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.

13. 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.

14. 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.

15. Kyung W. Kim, Young J. Kim, "Perceived Visibility Measurement Using the HSI Color Difference Method", Journal of the Korean Physical Society, Vol. 46, No. 5, pp. 1243-1250 , May 2005.

16. 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.

17. A Pujol, L Chen, *"Color quantization for image processing using self information"*, Proceedings of the 6th International Conference on Information, Communications and Signal Processing (IEEE ICICS), pp. 1-5, December 2007.

18. Hanning Chen, Yunlong Zhu, and Kunyuan Hu, *"Cooperative Bacterial Foraging Optimization"*, Hindawi Publishing Corporation , Discrete Dynamics in Nature and Society, Article ID 815247, 17 pages doi:10.1155/2009/815247, Volume 2009.

19. K. M. Bakwad, S.S. Pattnaik, B. S. Sohi, S. Devi, M.R. Lohakare, *"Parallel Bacterial Foraging Optimizationfor Video Compression"*, International Journal of Recent Trends in Engineering (IJRTE),Volume 1, Number 1, Issue on Computer Science,Page(s): 118-122, May 2009.

20. H.-C. Huang, Y.-H. Chen, and A. Abraham, *"Optimized Watermarking Using Swarm-Based Bacterial Foraging",* Journal of Information Hiding and Multimedia Signal Processing, vol. 1, no. 1, pp. 51-58, Jan. 2010.

21. Kaur Rajinder; Girdhar Akshay; Gupta Surbhi, *"Color Image Quantization based on Bacteria Foraging Optimization"*, International Journal of Computer Applications, vol. 25, issue 7, pp. 33-42, July 2011.

22. Dekker A (1994) Kohonen *"Neural networks for optimal colour quantization"*, Network: Computation in Neural Systems 5: 351- 367.

23. Segenchuk S. "*An Overview of Color Quantization Techniques.*"

24. Liwei Wang, Yan Zhang, Jufu Feng, *"On the Euclidean Distance of Images"*

25. Rachel Hager, *"An Introduction to Image Processing with MatLab"*

* * * * *