An Optimal Fuzzy Filter for Gaussian Noise in Color Images Using Bacterial Foraging Algorithm

Om Prakash Verma, Anil Singh Parihar, Shobha Tyagi
Department of Information Technology
Delhi Technological University
New Delhi, India
opverma.dce@gmail.com, parihar ani@gmail.com, tyagishobha@gmail.com

Abstract— This paper presents an optimal fuzzy filter for Gaussian noise in color images using Bacterial Foraging Algorithm (BFA) and cosine similarity. The filter makes use of the relationship between different color components of a pixel to remove the noise from the color images. Three color components of the RGB color space are paired as red-green, red-blue and green-blue. The adaptive cosine similarity between the central pixel and the neighboring pixels is estimated using these color pairs for noise removal. The membership function Large is defined and used to fuzzyf similarity of each color component. Mean Square Error is used as an objective function, which is optimized using the bacterial foraging algorithm to learn the parameters of membership function Large. The correction term for the Gaussian filter is calculated using weighted average of the weights of all the neighboring pixels. The proposed Gaussian filter is found to be effective in eliminating noise from color images with the significant improvement in image quality. The experimental result on several color images proves the efficacy of the proposed fuzzy filter.

Keywords— Fuzzy, Gaussian noise, Cosine similarity, Color-pair and BFA.

I. INTRODUCTION

The noise removal has been an important aspect of image processing. Noise free image is primary requirement for many applications such as object detection, segmentation, biometrics etc. Noise may get added in the image during acquisition by camera sensors and transmission in the channel etc. Two additive noises, Gaussian and Impulse has been major concern in image processing applications. The Gaussian noise among two, is more difficult to remove because of its nature. There are several works, in the field of noise reduction reported, in the literature. Tuan-Anh Nguyen et al. [1] proposed spatially adaptive de-noising algorithm for a single image corrupted by the Gaussian noise. The algorithm is consisting of two stages; first noise detection and then noise removal filtering. Local weighted mean, local weighted activity and local maximum were defined to corporate desirable properties into de-noising process. Russo [2] introduced a multi-pass fuzzy filter consisting of three cascaded blocks. Each block is hooked to a fuzzy operator that attempts to cancel the noise while preserving the image structure. D. Androustos, et al. [3] described the strong potential of fuzzy adaptive filters for multichannel signal applications, the filters use fuzzy transformations of the angles among the different vectors to adapt to local data in the image. Fuzzy filters are easy to realize by means of simple fuzzy rules that characterize a particular noise. Major problem in removing Gaussian noise is to differentiate between noise and edges. In [4], the effective fuzzy derivatives are used for differentiating the noise and edge pixels in images corrupted with Gaussian noise. Schulte et al. [5] consider the fuzzy distance between color pairs as a weight to perform the weighted average filtering for the removal of the Gaussian noise in color images. Russo [6] proposes a method for Gaussian noise filtering that combines a nonlinear algorithm for detail preserving and smoothing of noisy data, and a technique for automatic parameter tuning base on noise estimation. An efficient fuzzy filter for edge preservation is proposed in [7] using fuzzy technique for color images. In [8] a new fuzzy-logic-control based filter is introduced with the ability to remove impulsive noise and smooth Gaussian noise, while preserving edges and image details.

Some of the latest developments in this area is discussed as follows: Tuan-Anh Nguyen et al. [9] propose a spatially adaptive de-noising algorithm using local statistics for a single image corrupted by Gaussian noise. The algorithm consists of two stages: noise detection and noise removal filtering. To corporate desirable properties into de-noising process, local weighted mean, local weighted activity, and local maximum are defined. Using the local statistics, constraint for noise detection is defined. A modified Gaussian noise removal filter based on the local statistics is also used to control the degree of noise suppression.

Tzu-Chao Lin [10] uses decision-based fuzzy averaging (DFA) filter consisting of a D–S (Dempster–Shafer) noise detector and a two-pass noise filtering mechanism. Bodies of evidence are extracted, and the basic belief assignment is developed using the simple support function, which avoids the counter-intuitive problem of Dempster’s combination rule. A fuzzy averaging method, where the weights are constructed using a predefined fuzzy set, is developed to achieve noise cancellation. A simple second filter is employed to improve the final filtering performance.

The peer group of an image pixel is a pixel similarity based concept which has been successfully used to devise image de-
noising methods. The fuzzy peer group concept, which extends the peer group concept in the fuzzy setting, is described in [11]. A fuzzy peer group will be defined as a fuzzy set that takes a peer group as support set and where the membership degree of each peer group member will be given by its fuzzy similarity with respect to the pixel under processing. The fuzzy peer group of each image pixel are determined by means of a fuzzy logic-based procedure. The fuzzy peer group concept is used to design a two-step color image filter cascading a fuzzy rule-based switching impulse noise filter by a fuzzy average filtering over the fuzzy peer group. Both steps use the same fuzzy peer group, which leads to computational savings according to [11].

Hanagaki and osana in [12] proposes a similarity-based image retrieval considering artifacts by self-organizing map with refractoriness. In [13], Contemporary fuzzy logic is used to implement a relative pixel similarity value algorithm. The bacterial foraging scheme appeared in Passino [15] (2002), Liu and Passino [16] (2002). Foraging can be modeled as an optimization process where bacteria seek to maximize the energy obtained per unit time spent during foraging. In this scheme, an objective function is posed as the effort or a cost incurred by the bacteria in search of food. Recently the bacterial foraging algorithm has been applied in image enhancement and edge detection [17-19] processing. In [17] Madasu Hanmandlu et al. presented a new approach for the enhancement of color images using the fuzzy logic and bacterial foraging. A new approach for edge detection using a combination of bacterial foraging algorithm (BFA) and probabilistic derivative technique derived from Ant Colony Systems is presented by Verma et al. [18]. Biswas et al. [20-21] proposed a hybridization of bacterial foraging optimization algorithm with another very popular optimization technique of current interest called Differential Evolution. Dasgupta et al. [22] developed an algorithm which adopted swarm-intelligence technique, well known as the bacterial foraging optimization. Minimization of new fuzzy objective function which was derived for the edge map of a given image with an adaptive version of the BFO algorithm leads to the automatic detection of circles on the image. Analyze the chemotactic step [23] of a one dimensional BFOA in the light of the classical Gradient Descent Algorithm (GDA). Their analysis points out that chemotaxis employed in BFOA may result in sustained oscillation, especially for a flat fitness landscape, when a bacterium cell is very near to the optima. Datta et al. [24] proposed an improved adaptive approach involving bacterial foraging algorithm (BFA) to optimize both the amplitude and phase of the weights of a linear array of antennas for maximum array factor at any desired direction and nulls in specific directions. Kim et al. [25] proposed a hybrid approach involving genetic algorithms (GA) and bacterial foraging (BF) algorithms for function optimization problems.

In this paper, we have extended our previous work [29] on noise reduction in color images where Euclidian distance is used to measure the distance between the noisy pixels. The distance calculates similarity between the central pixel and the neighboring pixels. In proposed approach, cosine similarity is calculated between the central pixel and the neighboring pixels to estimate the similarity between the color pairs. Membership function large is defined which is used to fuzzify all the three color components. Mean Square Error is selected as an objective function, which is optimized with the help of the bacterial foraging algorithm to learn the parameters of membership function Large.

Organization of the paper is as follows: Section II presents the fuzzy filter based on the cosine similarity for Gaussian noise reduction. Objective function and its optimization using the bacterial foraging are discussed in section III. Algorithm for the proposed filter is given in section 4. Results and their analysis are discussed in Section 5. Finally, conclusions are drawn in Section 6.

II. FUZZY FILTER FOR NOISE REMOVAL

As Gaussian noise is additive, a color pixel in RGB color space with co-ordinates \((x, y, z)\) degraded by random noise is expressed [38] as:

\[
  f(x, y, z) = I(x, y, z) + \eta(x, y, z)
\]

(1)

Where \(f(x, y, z)\) is the noisy color image, \(I(x, y, z)\) is original color image both defined in RGB color space and \(\eta(x, y, z)\) represents the signal independent additive random noise in the same color space. The \(x\) and \(y\) represents the coordinates of the image pixel and \(z = 1, 2, 3\) represents the red, green and blue (RGB) color components at \((x, y)\) respectively. The methods for the reduction of Gaussian noise adopt the weighted average of neighborhood pixel values of the central pixel value [2]. The key point here is to select the weights to the neighborhood pixels in such a way as to obtain the corrected value. The use of color pairs to assign weights to the neighborhood pixels leads to a reduction in the ensuing artifacts. Adaptive fuzzy cosine similarity between the color pairs gives similarity between the central pixel and the neighborhood. This Adaptive fuzzy similarity helps in preserving edges by way of giving less weight to the noisy pixels and more weight to the similar pixels. Therefore, if the adaptive fuzzy similarity value is more than more weight is assigned and vice-versa.

A. Cosine Similarity

The Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. The result of the Cosine function is equal to 1 when the angle is 0, and it is less than 1 when the angle is of any other value. Calculating the cosine of the angle between two vectors thus determines whether two vectors are pointing in roughly the same direction.

Cosine of two vectors can be easily derived by using the Dot Product formula:

\[
  A \cdot B = \|A\| \|B\| \cos \theta
\]

(2)

Given two vectors of attributes, \(A\) and \(B\), \(n\) is the dimension of the vector space, the cosine similarity is given as

\[
  \cos \theta = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=0}^{n} A_i \times B_i}{\sqrt{\sum_{i=0}^{n} (A_i)^2} \times \sqrt{\sum_{i=0}^{n} (B_i)^2}}
\]

(3)
The similarity measure can be determined by various combination of the Euclidean distance and cosine similarity. An way to think about cosine similarity is as measuring the relative proportions of the various features or dimensions. When all the dimensions between two vectors are in proportion, then maximum similarity is obtained. Cosine similarity and Euclidean distance capture almost similar information. However, whereas Euclidean distance measures an actual distance between the two points of interest, Cosine can be thought of as measuring their apparent distance as viewed from the origin. Adaptive fuzzy similarity is found between each color pair of the central pixel and that of the neighborhood pixels. Color pairs are denoted in terms of the image function 'f' as follows [38]:

- Red-Green: \( f(x, y, 1), f(x, y, 2) \)
- Red-Blue: \( f(x, y, 1), f(x, y, 3) \)
- Green-Blue: \( f(x, y, 2), f(x, y, 3) \)  \( (4) \)

Adaptive similarity between a color pair of central pixel and that of neighborhood pixel, say between red-green pairs is found from:

\[
S_{rb}(x+i, y+j) = \frac{f(i, j, 1) \ast f(x+i, y+j) + f(i, j, 2) \ast f(x+i, y+j)}{(f(i, j, 1) + f(i, j, 2))^2 + (f(x+i, y+j))^2} \]  \( (5) \)

Where \( f(i, j) \) represents neighboring pixels and \( f(x, y) \) is the central pixel for the window of size \( k \times k \). Similarly, we can find \( S_{rb}(x+i, y+j) \) and \( S_{gb}(x+i, y+j) \) for adaptive similarity between the red-blue and green-blue components. The adaptive fuzzy similarity between the color pairs is obtained by fuzzifying the adaptive similarity using the membership function 'Large' to be introduced in the next section.

B. The Filter Structure

In the proposed method the weighted average of the neighboring pixels in the window of interest is calculated. The weights to the neighboring pixels are determined according to the following fuzzy rules.

For the Red component

IF \( S_{rb}(x+i, y+j) \) is large AND \( S_{gb}(x+i, y+j) \) is large
THEN weight \( W(x+i, y+j, 1) \) is a large.  \( (6) \)

For the Green component

IF \( S_{rb}(x+i, y+j) \) is large AND \( S_{gb}(x+i, y+j) \) is large
THEN weight \( W(x+i, y+j, 2) \) is a large.  \( (7) \)

For the Blue component

IF \( S_{rb}(x+i, y+j) \) is large AND \( S_{gb}(x+i, y+j) \) is large
THEN weight \( W(x+i, y+j, 3) \) is a large.  \( (8) \)

To express the degree to which an adaptive similarity is Large, the adaptive similarities are fuzzified using the membership function Large, defined as:

\[
\mu_L = \begin{cases} 
\exp \left( \frac{-(\frac{x}{t})^2}{2} \right), & \lambda \geq t \\
0, & \lambda < t 
\end{cases} \]  \( (9) \)

This membership function for the set “Large” is shown in Figure 1. The parameter ‘t’ is the minimum similarity between a color pair of a central pixel and that of the neighborhood in a window. The values for b and c in the above function are discussed in section 3.

![Membership Function for "Large"](image)

Fig. 1. The membership Function for “Large”

The above fuzzy rules are implemented by calculating the adaptive fuzzy similarity using the membership function “Large”. For example, fuzzy adaptive similarity between the red-green color pairs of a pixel at \((x, y)\) and a neighboring pixel at \((x+i, y+j)\) is represented as: \( \mu_{rg}(S_{rg}(x+i, y+j)) \)

where, \( \mu_{rg} \) is the membership function of the red-green color pair defined in equation (9). The weights for a neighboring pixel at the location \((x+i, y+j)\) corresponding to red, green and blue components are derived from fuzzy rules as:

\[
W(x+i, y+j, 1) = \max \left[ \mu_{rg}(S_{rg}(x+i, y+j)), \mu_{gb}(S_{gb}(x+i, y+j)) \right] \]  \( (10) \)

\[
W(x+i, y+j, 2) = \max \left[ \mu_{rb}(S_{rb}(x+i, y+j)), \mu_{gb}(S_{gb}(x+i, y+j)) \right] \]

\[
W(x+i, y+j, 3) = \max \left[ \mu_{rb}(S_{rb}(x+i, y+j)), \mu_{gb}(S_{gb}(x+i, y+j)) \right] \]

The weights for red, green and blue components follow similarly and the final corrected value of a pixel at location \((x, y)\) for the red component is given by:

\[
I(x, y, 1) = \frac{\sum_{i=-k}^{k} \sum_{j=-k}^{k} W(x+i, y+j, 1) \ast f(x+i, y+j, 1)}{\sum_{i=-k}^{k} \sum_{j=-k}^{k} W(x+i, y+j, 1)} \]  \( (11) \)

Where \( k \) is the size of the window, similarly, we can find \( I(x, y, 2) \) and \( I(x, y, 3) \) for green and blue components respectively.
III. PARAMETER OPTIMIZATION

The filtering action of the Gaussian filter is dependent on two parameters b and c. In our approach, we are training the values of these parameters with the help of bacterial foraging optimization technique. The search space of bacteria foraging technique is two dimensional and the movement of bacteria finds the minimum value of Mean Square Error. Parameter training with the help of bacterial foraging gives different values of parameters for different noise concentration level this is in contrast with the given constant values for these parameters. Mean Square Error is given as

\[ \text{MSE}(I, f) = \frac{\sum_{i=1}^{3} \sum_{j=1}^{M} \sum_{k=1}^{N} (I(x, y, z) - f(x, y, z))^2}{3 \times M \times N} \]  

(12)

Where, \( I \) is the original test image and \( f \) is the filtered image of \( M \times N \) size. MSE gives the similarity between two images. By minimizing MSE with the help of BF optimization technique, we are reducing the mean square error between the noisy image and the de-noised image and obtain the optimized values for the unknown parameters b and c to get the best possible results. Results are discussed in the next section.

Optimization problem is to find minima of mean square error therefore MSE is used as the food function for bacterial foraging. The objective function in our problem is given as

\[ J = \text{MSE}(I, f) \]  

(13)

The cell-to-cell attractant function of BF optimization algorithm is used to obtain high density food area, which is less important in this scenario; hence it is not used here.

A. Initialization of Parameters

This includes two sets of parameters: the parameters to be optimized, i.e., b and c in the original objective function J, and the parameters related to the BF algorithm [15] for facilitating the optimization. The initialization of the former will be discussed in the next section, and the initialization of the latter is now taken up.

The dimension of search space \( p = 2 \), the bacteria population \( S \) is taken 10, number of chemotactic step before reproduction \( N_c = 4 \), the bacteria split ratio \( S_r = S / 2 \), swimming length \( L = 4 \), number of reproduction steps \( N_r = 2 \), number of elimination and dispersion events \( N_e = 2 \), and the probability of elimination/dispersion event \( P_e = 0.25 \).

The movement of bacteria is given by following equation:

\[ \theta'(j + 1, k, l) = \theta(j + 1, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta'(i) \Delta(i)}} \]  

(14)

Where \( \theta'(j, k, l) \) is position of \( i^{th} \) bacterium in \( j^{th} \) chemotaxis, \( k^{th} \) reproduction and \( l^{th} \) elimination-dispersion step. \( C(i) \) is step size of bacteria movement and the direction movement is decided by vector \( \Delta(i) \).

IV. RESULT AND DISCUSSION

A color image consisting of an \( M \times N \times 3 \) array of pixel at locations \( (x, y) \) may be viewed as a “stack” of three scale images corresponding to RGB components. The color images “Lena”, “Parrots”, and “Flower” of size 256 × 256 with the Gaussian noise is considered as test images. The original images are shown in Figure 2.

![Image](image-url)

Fig. 2. Original Image (a)Lena, (b)Parrots, and (c)Flower

The mean square error (MSE) is selected as the measure of performance as defined above (12). Another measure of performance is peak signal to noise ratio (PSNR) given as:

\[ \text{PSNR}(I, f) = 10 \log \left( \frac{1}{\text{MSE}(I, f)} \right) \]  

(15)

Higher the PSNR value, better the de-noised image. Experiments are performed using different sizes of windows and the results for these experiments are shows that the window size of 3×3 is the most suitable. The performance of Gaussian noise filter is evaluated over the four test colour images with \( \sigma = 10, 20 \) and 30. The parameters b=0.2, c=0.9 are taken as the initial values to the Bacterial foraging algorithm.

<table>
<thead>
<tr>
<th>Variance (( \sigma ))</th>
<th>Initial Values</th>
<th>Optimize Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>( c )</td>
<td>MSE</td>
</tr>
<tr>
<td>10</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td>20</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td>30</td>
<td>0.2</td>
<td>0.9</td>
</tr>
</tbody>
</table>

It shows that there is a great difference between the values of the membership function parameters and MSE before and after the optimisation and hence the results. The results of the proposed approach are compared in terms of MSE with methods developed by Om Prakash Verma et al. (FFNRCI) [29], Tzu-Chao Lin, Decision-based fuzzy image restoration for noise reduction based on evidence theory (DBFIR) [12], Restoration of images corrupted by mixed Gaussian-impulse noise via l₁ - l₀ minimization (RICMG) [13], Fuzzy Peer Groups for Reducing Mixed Gaussian-Impulse Noise From Color Images [14] (FPGA), Spatially Adaptive De-noising Algorithm for a Single Image Corrupted by Gaussian Noise(SADA)[15].
The above result shows that proposed filter is better from many existing filters. Fig. 3 shows the comparison of PSNR values for different filter for Lena image with 30% Gaussian noise. The images in Fig. 4 also show visual difference between resulting images from different filters.

V. CONCLUSION

The removal of Gaussian noise is accomplished via fuzzy adaptive similarity between the color components of a pixel of interest and the neighbourhood pixel. The Adaptive fuzzy similarity between colour pairs approach produces a de-noised image with all the significant details preserved. It was shown that this filter is capable of reducing Gaussian noise up to $\sigma = 30$. This is observed that the proposed filter produces the better texture than the previous works. The use of the adaptive fuzzy similarity between colour components in the RGB space is resulted in better results.

REFERENCES


