Asian Journal of Information Technology 9 (4): 218-230, 2010
ISSN: 1682-3915
© Medwell Journals, 2010

Enhanced Fuzzy Method for Noise Reduction of Color Image

[1]M. Rahmat Widyanto and [2]Surya Agustian
[1]Faculty of Computer Science, University of Indonesia, Depok, 16424, Indonesia
[2]Faculty of Information Technology, YARSI University, Jakarta, 10510, Indonesia

Abstract: A new methodology of fuzzy low pass filter is proposed to reduce additive noise on color image which is also good for impulse and multiplicative noise. This new methodology calculates distance of every single color channel between central and neighbor pixels in a spatial window and then calculates the weights by adopting low pass filter concept in electronics. Several approaches are also proposed, i.e., weighting adjustment that adopts Euclidean Norm concept and power weighted averaging function rather than Takagi-Sugeno-Kang (TSK) at defuzzification stage. These two approaches with correct combinations lead to a better result on reducing additive, multiplicative and impulse noise from color image compared to the state-of-the-art fuzzy method. Experimental results on some benchmark images show that the PSNR of the proposed Fuzzy-LPF methodology increase by around 30% of the FCG subfilter from it where we start this research. Due to the fast growing utilization of digital image communication in the world this new proposed method can be useful for noise reduction tasks on image processing.

Key words: Digital image processing, noise reduction, low pass filter, fuzzy method, pixels, additive

INTRODUCTION

Noise appeared on digital color image is caused by many factors, e.g., low resolution digital camera, insufficient lighting during picture capture, snap image from CCTV or cam coder and so on. Commonly, there are three classifications of noise on digital image, impulse noise, additive noise and multiplicative noise. Impulse noise is kind of discontinuous signals (pixels) that have an extreme different value from its neighbor pixels. The example of impulse noise is called salt and pepper with various intensity. Additive noise is when a value of some kind of distribution (e.g., Gaussian distribution) is added to the image signal. Multiplicative noise could be a multiplication or convolution of different noise signal, intensity or distribution to image signal that become hard to be removed from the original image (e.g. speckle noise). Because of the need of image enhancement in multimedia application, many methods have been developed to overcome noise problem on digital image. However, neither the results do not perform user satisfaction, research area in this problem is still widely opened.

Additive noise is the most common problem that occurred in digital color image. Recently, there are some fuzzy methods have been introduced to reduce that type of noise. Farbiz and Menhaj (2000) used fuzzy logic control to filter image. Van De Ville *et al.* (2000) introduced fuzzy-based filter for Gaussian noise. Pu-Yin Liu introduced some fuzzy techniques for Gaussian and impulse noise removaln by Liu and Li (2004). Nachtegael *et al.* (2006) introduced fuzzy filter for Gaussian noise in grayscale image. Schulte *et al.* (2006a) proposed fuzzy-shrink wavelet method and proposed Fuzzy k-NN and MFRB combination Chang and Lu (2006) and many more. The latest method that we investigate in the research which called FCG was introduced by Schulte *et al.* (2007a, b). The last mentioned method calculates distance between central pixel and neighbor pixel of two dimensional RGB color, i.e., red-green, red-blue and green-blue. Membership value of each distance is then recalculated for weighting modification and final filtered image was resulted by defuzzification process.

This research was performed as a curiosity to deeply explore the relatively best performance of last mentioned fuzzy 2-dimension distance filter which is introduced by Schulte which claimed to show much better result compared to any other methods. The filter is built on two sub-filters, the first is based on fuzzy theory and the second one is simply calculates the local difference between the central pixel and its neighbor of each color channel separately and calculate the mean to be used as correction factor. In the research, fuzzy filter performance is examined to answer a basic question, why this fuzzy filter needs complement (second subfilter) to make some correctness. The investigation results that the first subfilter generally has less capability in reducing noise,

Corresponding Author: M. Rahmat Widyanto, Faculty of Computer Science, University of Indonesia, Depok, 16424, Indonesia

based on visual observation and numerical analysis (PSNR). The second subfilter tends to reduce image quality, i.e., color, edge and sharpness.

Some enhancements of this fuzzy method are proposed. The first step is to modify the fuzzy subfilter by means of low pass filter concept. This modification results noise reduction performance significantly increased. The second step is to eliminate the second subfilter which is suspected to decrease image artifacts, like edge and sharpness.

Later on, several new fuzzy filters based on low pass filter concept are introduced to reduce additive and impulse noise. We also propose the use of power weighted averaging approach rather than (Takagi-Sugeno-Kang) TSK method for defuzzification process. This new approach gives better results with correct combinations where the images feature, i.e., sharpness and brightness are enhanced and PSNR is significantly increased.

FUZZY COLOR PRESERVING GAUSSIAN NOISE (FCG)

This research is an enhancement of fuzzy noise reduction method introduced by Schulte named Fuzzy Color preserving Gaussian noise reduction method (FCG). For better understanding, the steps and equations of the method is re-written.

Digital color images are represented by three $m \times n$ $\{m, n \in N\}$ pixel color vector, i.e., red, green and blue in which m and n corresponds to width and height of the image. If a noise-free or clean image vector on each pixel position (i, j) where $\{i, j \in N\}$ is represented by C (i, j, c) and a noisy image is represented by N(i, j, c) where $c \in \{1,2,3\}$ is representing color index, i.e., 1 for red, 2 for green and 3 for blue then a corrupted image can be formulated as:

$$[N(i,j,1)\ N(i,j,2)\ N(i,j,3)] = [(C(i,j,1)+\rho_1)\ (C(i,j,2)+\rho_2)\ (C(i,j,3)+\rho_3)] \quad (1)$$

where, ρ_1, ρ_2, $\rho_3 \in R$ are three separate noise components. This could be impulsive, additive or multiplicative.

The noise component is usually generated by some kind of mathematical function. For additive noise, Gaussian distribution is used with mean (μ_1, μ_2 , $\mu_3 \in R$) and standard deviation (σ_1, σ_2, $\sigma_3 \in R$).

FCG method consists of the two subfilters. The first fuzzy subfilter calculates the weights for three 2-D distance pixel in a window, i.e,. red-green, red-blue and green-blue and the second subfilter calculates the local differences between pixel in the same color for correction factor.

FCG Subfilter I: In the first subfilter, fuzzy distance is calculated between couples and each couple pixel is denoted as:

Red-green	rg(i, j) = (N(i, j, 1), N(i, j, 2))
Red-blue	rb(i, j) = (N(i, j, 1), N(i, j, 3))
Green-blue	gb(i, j) = (N(i, j, 2), N(i, j, 3))

Image pixel then be evaluated in segmented window of size $(2K+1)\times(2K+1)$ with central pixel position on (i, j). Weight w (i+k, j+l,1), w (i+k, j+l, 2), w (i+k, j+l, 3) is applied to each vector color pixel, respectively at position (i+k, j+l) with $k,l \in \{-K,..: K \in N\}$. Three fuzzy rules on Takagi-Sugeno model are applied to define weights for each color component:

Fuzzy rule 1

Red component: If the distance between the couple rg (i, j) and rg (i+k, j+l) is small and the distance between the couple rb (i, j) and rb (i+k, j+l) is small. Then the weight w(i+ k, j+l, 1) is large.

Fuzzy rule 2

Green component: If the distance between the couple rg (i, j) and rg (i+k, j+l) is small and the distance between the couple gb (i, j) and gb (i+k, j+l) is small. Then the weight w (i+k, j+l, 2) is large.

Fuzzy rule 3

Blue component: If the distance between the couple rb (i, j) and rb(i+k, j+l) is small and the distance between the couple gb (i, j) and gb (i+k, j+l) is small. Then the weight w (i+k, j+l, 3) is large. These rules imply large weights to the neighbor pixels which have similar color as the center in a window by treating pixels as colors and not as three separate color components.

The next step is to calculate the distance D between couples using Euclidean distance:

$$D(rg(i,j),rg(i+k,j+l)) = ((N(i+k,j+l,1)-N(i,j,1))^2 + (N(i+k,j+l,2)-N(i,j,2))^2)^{1/2} \quad (2)$$

The other distances for couples red-blue and green-blue can be derived from Eq. 2, respectively. To short the Eq. 2, a notation for each distance is introduced:

$$\begin{aligned} \gamma_{rg}(i,j,k,l) &= D(rg(i,j),rg(i+k,j+1)) \\ \gamma_{rb}(i,j,k,l) &= D(rb(i,j),rb(i+k,j+1)) \\ \gamma_{gb}(i,j,k,l) &= D(gb(i,j),gb(i+k,j+1)) \end{aligned} \quad (3)$$

The next step is to find the maximum values p for the three distances in Eq. 3 for each window. These values will be used to determine the membership function for each color pair in window.

$$P_{rg}(i,j)=\max_{k,l\in\Omega}(\gamma_{rg}(i,j,k,l))$$
$$P_{rb}(i,j)=\max_{k,l\in\Omega}(\gamma_{rb}(i,j,k,l)) \quad (4)$$
$$P_{gb}(i,j)=\max_{k,l\in\Omega}(\gamma_{gb}(i,j,k,l))$$

$k, l\in\Omega$: $k, l\in\{-K,-K+1,...,0,..., K-1, K\}$ In this notation, Ω represents the position of each pixel in window $(2K+1)\times(2K+1)$ (Fig. 1).

Membership function small is determined to express that a neighbor pixel is much similar to the central pixel. In this method, small means that the certain couple color distance of neighbor pixel to the central pixel is small, e.g., distance of red and green couple between neighbor pixel and central pixel and so on. It can not exactly determine how small the distance is.

For this uncertainty, the fuzzy set small is used that is if the membership value $\mu_s\in[0,1]$ is one (1) then the distance is the closest which means the central pixel and the neighbor have the same couple colors. Vice versa if the membership value close to zero (0) then the distance is far which means that the neighbor and central pixels are completely different.

To express the value of uncertainty between far and close distance of one color, the Eq. 5 is used. Based on this equation, a membership degree is drawn as shown in Fig. 2 where $p = \upsilon\sqrt{2}$ (maximum distance allowed by filter and x is spatial pixel distance at (i+k, j+l) to the center in window:

$$\mu_s(x)=\begin{cases}\left(\dfrac{p-x}{p}\right)^2, & \text{if} x \le p \\ 0, & \text{if} x > p\end{cases} \quad (5)$$

Finally, the distance value of couple color between neighbor and central pixel is expressed as an algebraic dot product T-norm of each single color A and B (A and B represent single color channel red, green or blue) membership value in Eq. 6.

$$\mu_{A\cap B}(x) = T(\mu_A(x), \mu_B(x)) = \mu_A(x)\,\mu_B(x) \quad (6)$$

This T-norm is then used as weight for each evaluated pixel in window. Using the notation in Eq. 3, the Eq. 6 then rewrite as follow:

$$w(i+k, j+l,1) = \mu_{s1}(\gamma_{rg}(i,j,k,l)).\mu_{s2}(\gamma_{rb}(i,j,k,l)),$$
$$w(i+k, j+l,2) = \mu_{s1}(\gamma_{rg}(i,j,k,l)).\mu_{s3}(\gamma_{gb}(i,j,k,l)), \quad (7)$$
$$w(i+k, j+l,3) = \mu_{s2}(\gamma_{rb}(i,k,j+1)).\mu_{s3}(\gamma_{gb}(i,j,k,l)),$$

where μ_{s1}, μ_{s2} and μ_{s3} are the membership value of each pixel position in window, denote the similarity degree

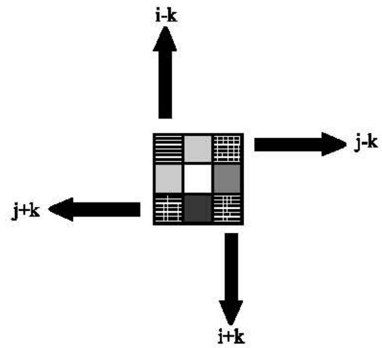

Fig. 1: Illustration of a spatial window consists of 9 pixels (K = 1) centered on (i, j)

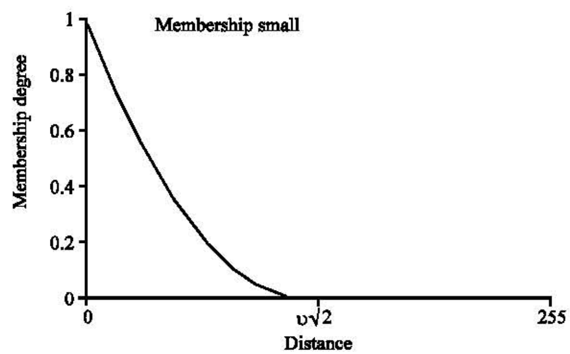

Fig. 2: Membership function of Eq. 5

between appropriate couple color of each pixel at (i+k, j+l) to the central pixel at (i, j). Next step, the weights will determine how the central pixel at (i, j) is corrected by the following equation:

$$F(i,j,c)=\frac{\sum_{k=-K}^{K}\sum_{l=-K}^{K} w(i+k,j+l,c).N(i+k,j+l,c)}{\sum_{k=-K}^{K}\sum_{l=-K}^{K} w(i+k,j+l,c)} \quad (8)$$

where, c is representing color code, i.e., 1 for red, 2 for green and 3 for blue and F is the corrected value of central pixel in window as the output of the first Fuzzy Subfilter.

FCG Subfilter II: Since the second subfilter is not based on fuzzy, the detail formulas are not written here. The second phase of this method simply calculates the local difference between the central pixel and its neighbor of each single color channel separately for each pixel position in window. An average value (mean) of those three local differences for every pixel position in window is determined and is used as correction factor. (Schulte *et al*., 2007a, b) for detail equations. The correction factor for each pixel position is then added to each single color channel for the appropriate pixel position in window. Later on it is observed that this linear averaging results a blur edge and degrade image quality.

FCG PERFORMANCE ANALYSIS

To see the performance of the FCG, simulation on some benchmark images is performed to each subfilter separately. The benchmark images are some common images which used widely in image processing research, e.g., lena, baboon, clown, flower, bird, etc. For comparison purpose, lena (512×512) and baboon (512×512) images are used as shown in Fig. 3.

Let's focus on the membership function first. From Eq. 5 and Fig. 2 it is found an implicit statement that only neighbor pixels with a very close value to the central pixel will have membership degree close to one (1). This membership function does not give any benefit for correction, since it only considers a very similar pixel to the center and reject others. For normalized distance with $p = \sqrt{2}$ which means maximum distance from neighbor to central pixel for each single color channel is one (of 255 scale), the membership function will give a value. Otherwise, it is canceled. As the consequences, algebraic product T-norm of two membership degrees between two color channels in Eq. 6 will result a very small value (if not cancelled by fuzzy membership) for other distance, i.e., approach to zero.

A zero-value of T-norm will happen when the central pixel is different from its environment (e.g. contaminated by noise) which is relatively far from its neighbors. It will cause any other distance from neighbor pixels will be canceled by filter. The reason is that filter only considers very similar pixels to have significant membership values. Since it is believed that any neighbor pixel is not similar to central noisy pixel, dot products of each pixel position in window will result close-to-zero value to further be the

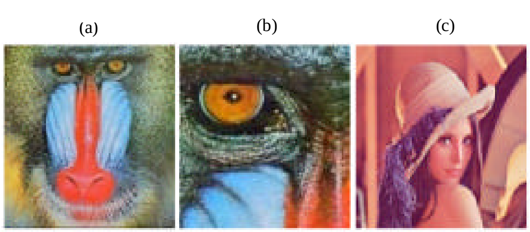

Fig. 3: Test images: (a) baboon, (b) baboon's eye for visualization part of baboon image in (a), (c) lena

weight components in Eq. 7. This weight will take into account to weight averaging for correction factor to central pixel (i, j) as expressed in the Eq. 8. As the consequent, correction will not occur since the product of weights on the appropriate neighbor pixels is very tiny and the pixel value of central pixel remains the same. In other word, the central pixel is still a noise.

Now let's having proof of this explanation by using the equations which form fuzzy subfilter1. Assume a K×K spatial window has a uniformly pixel value around a for each color channel, before noise added. Then a level of noise (could be impulse, additive or multiplicative) is added to the central pixel with value is υ. Let's assume:

$$N(i+k, j+l,1) = \lim_{x\to\alpha} x, \quad j, k \in [-K, K] \text{ and } j, k \neq 0$$
$$N(i+k, j+l,2) = \lim_{x\to\alpha} x, \quad j, k \in [-K, K] \text{ and } j,k \neq 0$$
$$N(i+k, j+l,3) = \lim_{x\to\alpha} x, \quad j, k \in [-K, K] \text{ and } j,k \neq 0$$
$$N(i, j,1) = \alpha+\upsilon$$
$$N(i, j,2) = \alpha+\upsilon$$
$$N(i, j,3) = \alpha+\upsilon$$

Where $\alpha, \upsilon \in [0,255]$ K is window size, i, j is position index. In this assumption, a spatial window image with pixels value is drawn for each certain color component is about similar. Since central pixel is noisy with value ($\alpha+\upsilon$) and the other pixels value in window are approach to α, the distances between central pixel and neighbor pixel at any position would be calculated as follow:

For $j,k \in [-K, K]$ and $j,k \neq 0$,

$$D\big(rg(i, j),rg(i+k, j+l)\big) = \Big(\big(N(i+k, j+l,1)-N(i, j,1)\big)^2 + \big((N(i+k, j+l,2)-N(i, j,2))^2\big)^{1/2} = \lim_{x\to\alpha} \Big(\big(x-(x+\upsilon)\big)^2 + \big((x-(x+\upsilon)^2\big)^{\frac{1}{2}} = \Big(\big(\alpha-(\alpha+\upsilon)\big)^2 + \big(\alpha-(\alpha+\upsilon)\big)^2\Big)^{\frac{1}{2}} = \upsilon\sqrt{2}$$

$$D(rb(i, j), rb(i+k, j+l)) = D(rg(i, j), rg(i+k, j+l))$$

Then,

$$\mu_{s1}\big(\gamma_{rg}\ (i, j,k,l)\big) = D\big(rg(i, j),rg(i+k, j+l)\big) = \upsilon\sqrt{2}$$
$$\mu_{s2}\ (\gamma_{rb}\ (i, j,k,l)) = D(rb(i, j),rb(i+k, j+l)) = \upsilon\sqrt{2}$$

Else

$$\mu_{s1}\big(\gamma_{rg}\ (i, j)\big) = D\big(rg(i, j),rg(i,j)\big) = 0$$
$$\mu_{s2}\big(\gamma_{rg}\ (i, j)\big) = D\big(rg(i, j),rg(i,j)\big)= 0$$

With the same calculation, it will result same value for other couples of the example. Since the evaluated window is uniformly colored and the central pixel is noisy, the distance will be approach to maximum (e.g., $\mu = \upsilon\sqrt{2}$) and membership value of Eq. 5 will approach to zero (or very close to zero) as shown in Fig. 2.

A wasting computation will be occurred when calculating weight for each pixel because only very small impact occurred for the weights and the result is again double approach to zero. For convincing this assumption, let's take a look on the weight calculations:

$$w(i+k, j+l,1) = \mu_{s1}\left(\gamma_{rg}\ (i,j,k,l)\right).\ \mu_{s2}\left(\gamma_{rb}\ (i,j,k,l)\right) = 0,$$
$$w(i+k, j+l,2) = \mu_{s1}\left(\gamma_{rg}\ (i,j,k,l)\right).\ \mu_{s3}\left(\gamma_{gb}\ (i,j,k,l)\right) = 0,$$
$$w(i+k, j+l,3) = \mu_{s2}\left(\gamma_{rb}\ (i,j,k,l)\right).\ \mu_{s3}\left(\gamma_{gb}\ (i,j,k,l)\right) = 0,$$
$$\text{for } k,l \neq 0 \ \text{ and } k,l \in \{-K,..K\}$$
$$w(i,j,1) = \mu_{s1}\left(\gamma_{rg}(i,j)\right).\ \mu_{s2}\left(\gamma_{rb}\ (i,j)\right) = 1,$$
$$w(i,j,2) = \mu_{s1}\left(\gamma_{rg}\ (i,j)\right).\mu_{s3}\left(\gamma_{gb}\ (i,j)\right) = 1,$$
$$w(i,j,3) = \mu_{s2}\left(\gamma_{rb}\ (i,j)\right).\ \mu_{s3}\left(\gamma_{gb}\ (i,j)\right) = 1, \text{ (at } k,l = 0)$$

However, Eq. 8 will be resource consuming, since the result F(i, j, n) will be remain the same as the noisy pixel N(i, j, n) which means the central pixel in the appropriate spatial window will not be corrected, i.e.,

$$F(i,j,n)=\frac{\sum_{k=-K}^{K}\sum_{l=-K}^{K} w(i+k,j+l,n).N(i+k,j+l,n)}{\sum_{k=-K}^{K}\sum_{l=-K}^{K} w(i+k,j+l,n)}=N(i,j,n)$$

For visual analysis, the simulation results for original image (a) with affected by Gaussian noise with $\sigma = 40$ are shown in Fig. 4. The PSNR for each picture is 16.46, 17.52 and 14.72 for noisy (b), subfilter1 output (c) and subfilter2 output (d), respectively.

It is found that using average local differences for correction factor in subfilter2 decreases the image quality, i.e., color and shape. Figure 4d is the result of subfilter2 when applied directly to filter noise (without subfilter 1).

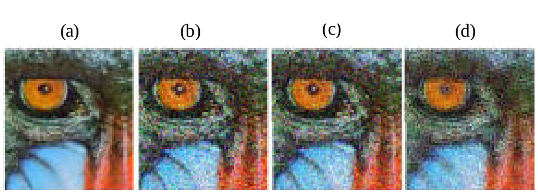

Fig. 4: Visualization for FCG performance, (K,L) = (1,2)

PERFORMANCE ENHANCEMENT

For fuzzy filter enhancement, we adopt Low Pass Filter (LPF) concept in electronic engineering (e.g., for filtering frequency) which will pass as much frequency until a determined cut off frequency. As ideal LPF is never be found in real electronic component, cut off frequency is determined when it's power output is 1/2 of maximum (pass band) power or the voltage is about $1/\sqrt{2}$ of the pass band voltage.

Hence the cut off frequency is also known as the 3 dB point because the voltage $1/\sqrt{2}$ is about -3 decibel (Kwang, 2005; Bourbaki, 1987). Figure 5a, b shows the description of LPF in electronics, ideal LPF and LPF approach ideal.

Some state-of-the-art membership functions which approach ideal LPF transfer function, e.g. Z-MF (Eq. 9) and Sigmoid-MF (Eq. 10) and square (Eq. 11) are investigated to be applied to the fuzzy filter enhancement which is drawn on Fig. 6. Varying membership function parameters will change slope and cut off. In this case, power (y-axes) is the degree of the distance (close or far),

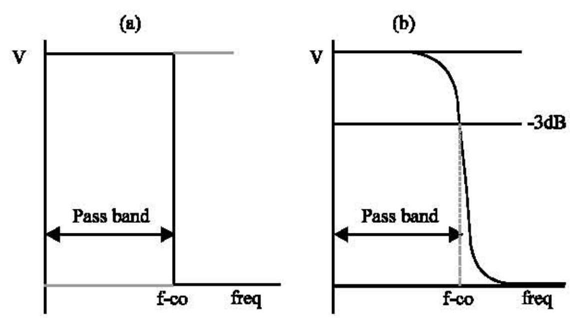

Fig. 5: Ilustration of low pass filter transfer function

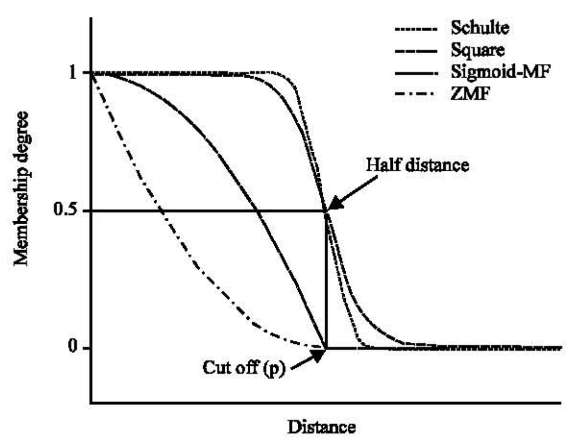

Fig. 6: Some Fuzzy Membership Functions (Schulte is the MF used by Schulte *et al.* (2007c)

the slope of the curve determine how much degree of the distance will be passed by filter which later correspond to its weighting function.

$$\mu_s(x)=\begin{cases} 1, & \text{if } x < a \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & \text{if } a \le x \le \frac{a+b}{2} \\ 2\left(\frac{b-x}{b-a}\right)^2, & \text{if } \frac{a+b}{2} \le x \le b \\ 0, & \text{if } x > b \end{cases} \quad (9)$$

$$\mu_s(x)=\frac{1}{1+e^{(-a(x-p))}} \quad (10)$$

$$\mu_s(x)=\begin{cases} 1-\frac{x^2}{p^2}, & \text{if } x \le p \\ 0, & \text{if } x > p \end{cases} \quad (11)$$

Following Schulte FCG method, the first thing we can enhance is modifying fuzzy distance membership function. Since we want to pass pixels that have as much correlation or very similar to central pixel, the LPF term is used. The mentioned fuzzy membership functions are used in the simulation.

From Fig. 6 it is clear that by definition LPF will pass the distance which is small than p (the distance cut off). The membership degree for any passed distance will determine the weight for the appropriate pixel position. Considering this issue and referring to Fig. 6, Z-MF and Sigmoid-MF is expected to produce better result than Schulte and Square. It is possible to use any other membership function as long as LPF concept related.

By assuming that the distance is the power of LPF and the cut off is when the membership value for the distance is a half of the maximum (e.g. 0.5), we define the cut off is the maximum distance in window for each color couple (Eq. 4). It will consider any distance contributions for correction weights. For Z-MF, the slope of the LPF membership function will be determined by value a and b where $a = p-\lambda$ and $b = p+\lambda$, $\lambda \in R$.

It is also possible to change the slope for seeking the best performance by change the value of λ. The simulation for this enhancement gives much better result than Subfilter1.

Caused by long computation time of Z-MF filter, Sigmoid-MF is considered. Later we find that Sigmoid-MF gives better result than Z-MF after several attempts in finding the optimal parameters. Besides, its computation time is about 7 times faster than Z-MF (in the simulation, Z-MF takes about 204 sec, Sigmoid-MF 28 sec). Similar to Z-MF, we define the cut-off distance for Sigmoid-MF is the maximum distance of color couple in a spatial window. A negative slope (a) is chosen to fit LPF concept. Like Z-MF, it is also possible to use different value from the proposed slope a.

FUZZY LOW PASS FILTER

Considering that color is not only formed by two primary color components as previous method in (Shulte *et al*., 2007c, d) but a fusion of three primary colors (RGB) we propose Fuzzy 3-D Low Pass Filter that considering all colors, either in distance measurements or weighting calculations. The basic research idea is to calculate the Euclidean distance between neighbor pixels to central pixel for each color component separately, calculate each membership degree of distance and manipulate the three colors membership degree for weighting. Rather than using the state-of-the-art weight average defuzzification like Takagi-Sugeno (TSK) method (Takagi and Sugeno, 1985; Jang *et al*., 1997; Kwang, 2005) we also manipulate this defuzzification manner. Some experiments are performed to investigate the influence of these manipulations and for some combination it gives a better result as expected.

The following detail is written to give more brief about the modification of FCG and mathematical manipulation in the experiments, e.g.:

- Pixel Distance (first flag) for distance measure, we use the Euclidean distance (flag: E), Hamming distance (flag: H), power-2 Hamming distance (flag: P)
- Color Dimension for distance measure (second flag) The filter measure distance of 1, 2 or 3 color vector (Flag: 1, 2 or 3, respectively)
- Membership function (third flag) we use Sigmoid-MF (flag: S) and Z-MF (flag: 2)
- Fuzzy Rules and Weighting (fourth flag) for weighting calculation we use algebraic product (dot-product) T-norm (flag: T), Euclidean norm or 2-norm (Bourbaki, 1987) (flag: E) and proportional Euclidean norm (flag: P). For 3-D distance, the 3-D membership degree is used as the weight for all colors (flag: n)
- Weight Averaging (5th flag) we propose another way in defuzzification rather than TSK method (flag: T) that we name power-2 weighted averaging (flag: W)

Flag is used for naming convention of experimented filter, express the combination of manipulated component, for example a filter code LPF-H3STT means a filter consi-dering 3 color components (3 Dimension) using Hamming distance pixel measurement with Sigmoid-MF, calculating T-norm for weight modification with TSK method weight averaging.

Pixel distance and color dimension: These two components are used together for filter variations. We treat different combinations of them to find the better result in simulation. The combinations format e.g. Hamming Distance (HD) of neighbor pixel to central pixel is formulated:

For 1-Dimension:

$$d_c = |N\ (i+k, j+l,c)\text{-}N\ (i, j,c)|$$

For 2-Dimension:

$$d_{xy} = |N(i+k, j+l, x)\text{-}N(i, j, x)\text{-}N(i,j,x| \\ +|N(i+k, j+l, y)\text{-}N(i, j, y)|$$

For 3-Dimension:

$$d_{xyz} =| N(i+k, j+l, x)\text{-}N(i, j, x)| \\ +|N(i+k, j+l, y)\text{-}N(i, j, y)| \\ +|N(i+k, j+l, z)\text{-}N(i, j, z)|$$

where, x, y, z, c represent the color index {1,2,3}, (i, j) is the pixel position index, k,l∈{-K,..,K} is the size of spatial window and d_c, d_{xy}, d_{xyz} represent the distance of the color code x, y, z and c. Power-2 Hamming Distance is the distance of 2nd ordered pixel between neighbor and center in a window is formulated:

For 1-Dimension:

$$d_c\ (i, j,k,l)= |N(i+k, j+l,c)^2 - N(i, j,c)^2|^{1/2}$$

For 2-Dimension:

$$d_{xy}\ (i, j,k,l)=|N(i+k, j+l, x)^2 - N(i, j, x)^2|^{1/2} \\ +| N(i+k, j+l, y)^2 \text{-}N(i, j, y)^2|^{1/2}$$

For 3-Dimension

$$d_{xyz}\ (i, j,k,l) = |N(i+k, j+l, x)^2 \text{-}N(i, j, x)^2|^{1/2} \\ +|N(i+k, j+l, y)^2 \text{-}N(i, j, y)^2|^{1/2} \\ +|N(i+k, j+l, z)^2 \text{-}N(i, j, z)^2|^{1/2}$$

Euclidean distance

For 2-Dimension:

$$d_{xy}\ (i, j,k,l) = \left(N(i+k, j+l, x)\text{-}N(i, j, x)\right)^2 \\ +\left(N(i+k, j+l, y)\text{-}N(i, j, y)^2\right)^{1/2}$$

For 3-Dimension:

$$d_{xyx}\left((i, j,k,l) = N(i+k, j+l, x)\text{-}N(i, j, x)\right)^2 \\ +\left(N(i+k, j+l, y)\text{-}N(i, j, y)\right)^2 \\ +\left(N(i+k, j+l, z)\text{-}N(i, j, z)^2\right)^{1/2}$$

Membership function: Z-MF and Sigmoid-MF as formulated in Eq. 9 and 10 is used in this experiment (Jang *et al*., 1997; Kwang, 2005). Parameters are adjusted to get the best result for each membership function while skip the rest combination to be unchanged. Either Z-MF or Sigmoid-MF is tent to give results that close each other. But Z-MF calculation time is not good which is about 6-7 times longer than Sigmoid-MF. The formulation of each distance membership degree is notated by:

Sigmoid function:

$$\mu_c = \text{sigmoidmf}\ \left(d_c\ (i, j,k,l),[a\ \max(d_c(i, j,k,l)]\right)$$

Z-MF function:

$$\mu_c = \text{zmf}\left(d_c\ (i, j,k,l),[a\ b]\right)$$

where, d_c is the distance of category c and c represents the subscript code for each dimension, i.e., for 1-Dimension: r (red), g (green), b(blue); for 2-Dimension: rg (red-green), rb (red-blue), gb (green-blue); for 3-Dimension: rgb (red-green-blue).

Fuzzy rules and weighting: Fuzzy weighting has a strong correlation to the membership function and the distance method chosen, either 1-Dimension or 2-Dimension. Considering that a color is formed by 3 natural color components, each color conjugate takes into account in a certain pixel color weighting.

As all colors are already counted for 3-Dimension distance, the weights for the 3 color pixel come from the membership degree of the distance itself. In other words, the weights for the red, green and blue pixel are equal. Based on this reasoning, fuzzy rules for 1-D and 3-D distance are determined:

1-D distance fuzzy rule: If the distance between the central pixel N(i, j, 1) and neighbor pixel N(i+k, j+l, 1) is small and the distance between the central N(i, j, 2) and neighbor pixel N(i+k, j+l, 2) is small and the distance between the central N(i, j, 3) and neighbor pixel N(i+k, j+l, 3) is small then the weight w(i+k, j+l) for all colors is large.

3-D distance fuzzy rule: If the 3-D distance d_{rgb} (i, j,k,l) between the central pixel and neighbor pixel is small then the weight w(i+k, j+l) for all colors is large. Fuzzy rule and weighting function for 2-D distance is the same as previous method. To short the written equation, we simplify the following notation, $Dc = d_c$ (i+k, j+l) where index c represents the criteria as already mentioned.

Weighting calculations based on the Fuzzy Rules above are basically using algebraic product T-norm (Zadeh, 1965) i.e., For 1-D distance:

$$w_{rgb}\ (i+k, j+l) = \mu_S\ (D_r).\ \mu_S(D_g\ ).\ \mu_S(D_b\ )$$

For 3-D distance:

$$w_{rgb}\ (i+k, j+l) = \mu_S(D_{rgb})$$

Here, w_{rgb} is applied to all color components, in other words the weight for all colors is the same. Rather than use equal weight for each color we perform some mathematical manipulations to modify the weighting function. Experiment results show that for some noise levels, especially with high standard deviation (e.g., σ = 30, 40, 50) the weighting method (combined with other variation and modification) stand on the top 5 performance. We proposed a Euclidean-Norm (Eq. 12) and proportional Euclidean-Norm (Eq. 13) to be used in fuzzy weight calculation as follow:

$$\mu x1 \cap x2 \cap \ldots \cap xn = E(\mu_{x1}, \mu_{x1}, \ldots, \mu_{xn}) = \sqrt{(\mu_{x1})^2 + (\mu_{x1})^2 + \ldots + (\mu_{xn})^2} \quad (12)$$

$$\mu x1 \cap x2 \cap \ldots \cap xn = pE(\mu_{x1}, \mu_{x1}, \ldots, \mu_{xn}) = \sqrt{\alpha_1(\mu_{x1})^2 + a_2(\mu_{x1})^2 + \ldots + \alpha_n(\mu_{xn})^2} \quad (13)$$

where, α is determined by user proportionally. The normalized portion is 1 (α = 1) which is back to the Euclidean Norm (E-norm) definition. In this LPF variation, we use E-norm and proportional E-norm for weighting calculation, i.e., E-norm for 1-D distance:

$$w(i+k, j+l) = \sqrt{(\mu_s(D_r))^2 + (\mu_s(D_g))^2 + (\mu_s(D_b))^2}$$

E-norm for 2-D distance:

$$w(i+k, j+l, 1) = \sqrt{(\mu_s(D_{rg}))^2 + (\mu_s(D_{rb}))^2}$$
$$w(i+k, j+l, 2) = \sqrt{(\mu_s(D_{rg}))^2 + (\mu_s(D_{gb}))^2}$$
$$w(i+k, j+l, 3) = \sqrt{(\mu_s(D_{rb}))^2 + (\mu_s(D_{gb}))^2}$$

Proportional E-norm for 1-D distance

$$w(i+k, j+l, 1) = \sqrt{(\mu_s(D_r))^2 + \frac{(\mu_s(D_g))^2 + (\mu_s(D_b))^2}{2}}$$
$$w(i+k, j+l, 2) = \sqrt{(\mu_s(D_g))^2 + \frac{(\mu_s(D_r))^2 + (\mu_s(D_b))^2}{2}}$$
$$w(i+k, j+l, 3) = \sqrt{(\mu_s(D_b))^2 + \frac{(\mu_s(D_r))^2 + (\mu_s(D_b))^2}{2}}$$

Weight averaging: Besides using the state-of-the-art fuzzy inference method (e.g., TSK-method in Eq. 8 a different approach is proposed using Euclidean weighting concept as the following Eq. 14:

$$\bar{x} = \frac{\sum_{i=0}^{n}(w_i x_i)^2}{\sum_{i=0}^{n}(w_i)^2} = \frac{(w_1 x_1)^2 + (w_2 w_2)^2 + .. + (w_n x_n)^2}{(w_1)^2 + (w_2)^2 + .. + (w_n)^2} \quad (14)$$

This similar approach had been used by (Kurniawati *et al*., 1998; Megiddo and Zemel, 1986) for different applications and conditional case. By applying Eq. 14 to fuzzy inference system, the output can be determined as:

$$F(i, j, c) = \frac{\sum_{k=-k}^{k}\sum_{l=-k}^{k}\left(w(i+k, j+l, c).N(i+k, j+l, c\right)^2}{\sum_{k=-k}^{k}\sum_{l=-k}^{k}\left(w(i+k, j+l, c)\right)^2}$$

where, c is the index of color component (red = 1, green = 2, blue = 3).

SIMULATION RESULT AND ANALYSIS

Some combinations are chosen to form several fuzzy-LPFs. All LPFs research on the 3×3 spatial windows to reduce calculation time. We start to form those filters by almost all possible combinations which means it is starting from a set of combination options and simulate the filter for several noise levels. Then we change one option (e.g., Z-MF to Sigmoid-MF) and perform the same simulations. We compared the results, when they are significant it is consider to use the two options and continue to other options. On the other way, either the results are not significant for an option compared to the other, we leave the option. At a certain combination we stop when we found the best result and move to other options. In this research, we trash away the combination which give the worst result and keep some that fit to the expected PSNR (Peak Signal to Noise Ratio) level.

The PSNR is an objective measurement of output image quality compared to the original clean image which is calculated as following Eq. 15:

$$\text{PSNR} = 10\log\left(\frac{255^2}{\frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{c=1}^{3}(C(i,j,c)-F(i,j,c))^2}\right) \quad (15)$$

Where C and F are clean and filtered image, respectively with size N×M and c is image color index. The proposed LPFs are also tested to reduce impulse and multiplicative noise to see how robust the LPFs against any type of noises. We concern to increase PSNR value at high noise level by modifying MF attributes with consideration that it is not significant to have high PSNR for low noise level because both noisy and filtered images are relatively clean already. All the results for baboon and lena image are shown in Fig. 7-9 and the PSNR values of each filter in Table 1-3.

Additive noise: Based on selected parameters of each combination options, Fuzzy Low Pass Filters show better result compared to the previous method.

A crisp weight averaging filter which gives more weight to central pixel (in this simulation we name it CWA = Center Weighted Averaging Filter) seems to study at the preferred results for several Gaussian noise levels. It calculates an average of pixel value in window after central pixel multiplied by some number λ (e.g., 3, 4, 5, 6 and so on), then divided by (8+λ). Normal averaging (Mean Filter) is when λ = 1.

The experiments show that the best results of the filter are when the multiplication factor of central pixel (λ) is 4. Compared to this conventional method, some of the fuzzy LPFs at certain combination have higher PSNR values.

From Table 1 it is clear that the fuzzy LPF method is good to reduce high level Gaussian noise which is shown by PSNR values of almost fuzzy LPF combinations. It increases filtering quality about 30% of previous fuzzy performance.

For low level Gaussian noise, although some combinations have lower PSNR values they are still acceptable since the filtered image qualities are fit to human visualization. Overall, either combination of Hamming Distance on 1-D or 3-D and Sigmoid-MF or Z-MF with TSK method, have relatively show the best performance on high level Gaussian noise reduction in this experiments.

Impulse noise: PSNR values of tested filter as shown in Table 2 show that Low Pass Filter concept is robust against impulse noise and have increased the filtering quality of FCG with a significant number.

For each impulse noise level, some combinations of the LPFs outperform the overall performance. Low PSNR values and less visual image quality shown in Fig. 9 come from several combination of algebraic-product

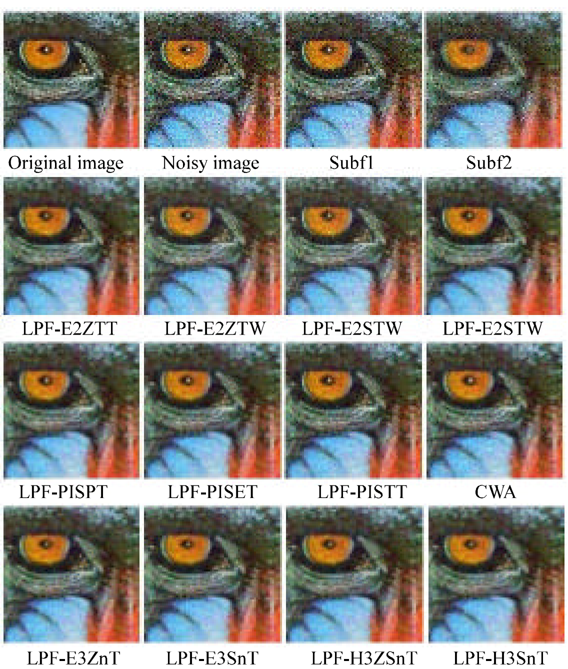

Fig. 7: Part of baboon image with Gaussian noise σ = 30 and the filters output

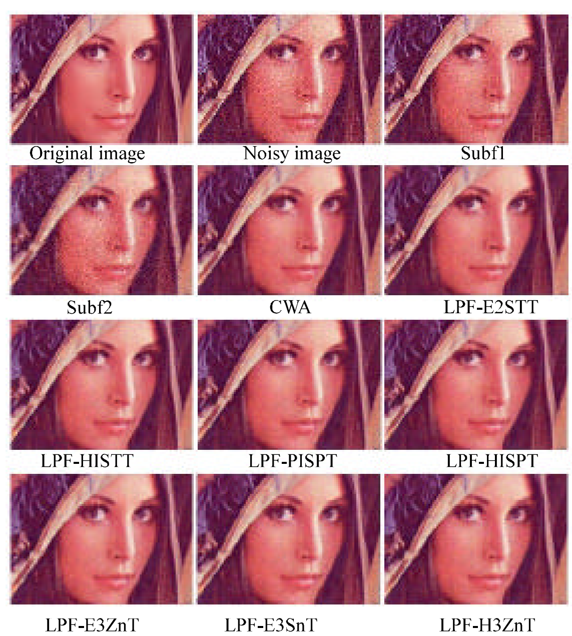

Fig. 8: Part of Lena image with Gaussian noise σ = 20 and the filters' output

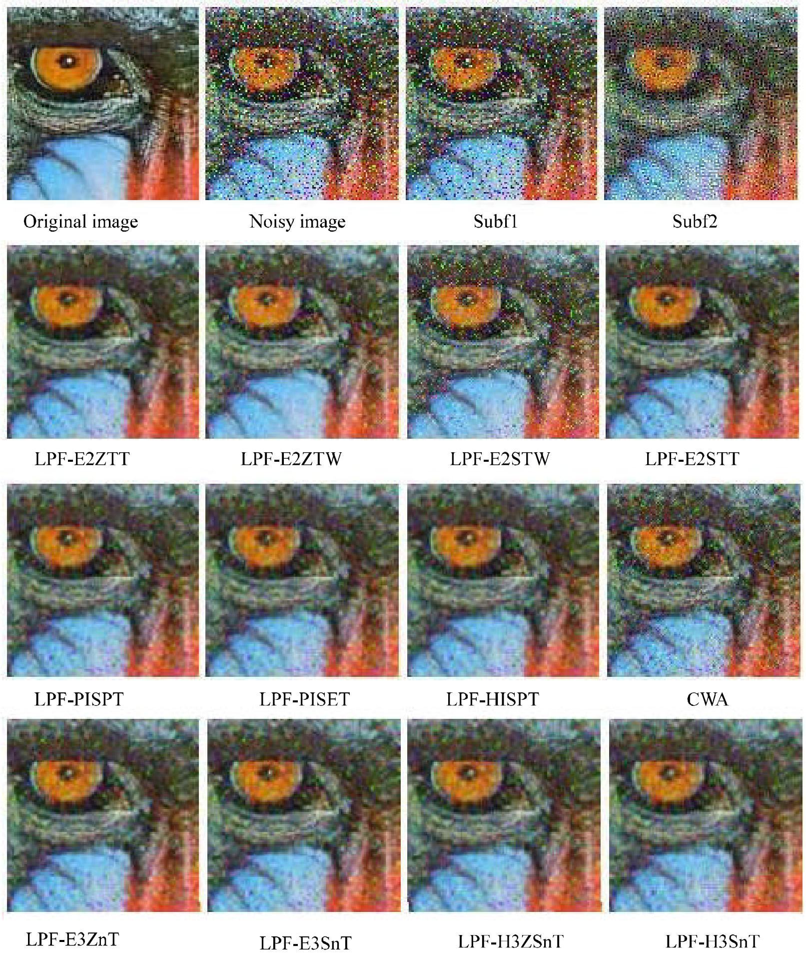

Fig. 9: Part of baboon image with 15% impulse noise (salt and pepper) and the filters output

Table 1: The comparison of PSNR values of some combinations of fuzzy low pass filter and enhanced fuzzy 2-D distance filter against various gaussian noise level

	PSNR of baboon image (in dB)						PSNR of Lena image (in dB)					
Image source	σ = 5	σ = 10	σ = 20	σ = 30	σ = 40	σ = 50	σ = 5	σ = 10	σ = 20	σ = 30	σ = 40	σ = 50
Noisy image	34.15	28.17	22.22	18.81	16.46	14.74	34.13	28.16	22.24	18.87	16.54	14.83
FCG-Subf1	34.32	29.38	23.45	19.94	17.52	15.75	35.09	29.28	23.33	19.93	17.57	15.83
FCG-Subf2 (3×3)	18.26	18.13	17.66	16.98	16.26	15.56	27.26	26.31	23.82	21.57	19.74	18.23
FCG-Subf2 (5×5)	15.78	15.72	15.48	15.13	14.72	14.31	23.75	23.36	22.06	20.61	19.23	18.03
CWA	26.67	26.30	25.07	23.61	22.18	20.95	34.30	32.49	28.80	25.98	23.81	22.13
LPF-E2STT	27.15	26.75	25.46	23.88	22.35	21.04	34.23	32.83	29.29	26.32	24.02	22.27
LPF-E2STW	28.00	27.46	25.71	23.61	21.61	19.96	34.33	32.81	28.83	25.42	22.77	20.76
LPF-E2ZTT	25.68	25.46	24.62	23.48	22.31	21.24	33.77	32.40	29.27	26.69	24.62	23.00
LPF-E2ZTW	25.82	25.58	24.56	23.19	21.77	20.52	33.88	32.27	28.76	25.98	23.76	21.99
LPF-E3SnT	25.51	25.26	24.43	23.38	22.32	21.28	33.59	32.32	29.34	26.98	24.83	23.24
LPF-E3ZnT	25.84	25.58	24.68	23.53	22.40	21.31	33.62	32.42	29.40	26.90	24.76	23.16
LPF-H3SnT	33.59	32.32	29.34	26.86	24.83	23.24	33.19	32.12	29.42	26.93	24.93	23.31
LPF-H3ZnT	33.62	32.42	29.40	26.83	24.76	23.16	32.72	31.78	29.30	26.90	25.01	23.43
LPF-H1SPT	25.14	24.94	24.25	23.33	22.32	21.34	32.85	31.88	29.37	26.99	25.01	23.37
LPF-H1STT	26.80	26.44	25.33	23.99	22.63	21.40	33.62	32.47	29.53	26.87	24.66	22.88
LPF-P1SPT	25.44	25.19	24.40	23.39	22.33	21.35	33.12	32.06	29.37	26.96	24.94	23.32
LPF-P1STT	27.86	27.28	25.70	24.04	22.50	21.18	34.39	32.84	29.33	26.46	24.20	22.43
LPF-P1STW	31.02	29.49	26.14	23.35	21.12	19.40	35.73	33.00	27.93	24.42	21.87	19.96
LPF-P1SETs	25.43	25.18	24.37	23.37	22.32	21.35	33.07	32.02	29.37	26.98	24.98	23.37

*Red font = top 5, bold = top 2 of PSNR values

Table 2: The comparison of PSNR values of some combinations of fuzzy low pass filter and enhanced fuzzy 2-D Distance filter against various impulse noise level

	PSNR of baboon image (dB)				PSNR of Lena image (dB)			
Image source	5%	10%	15%	25%	5%	10%	15%	25%
Noisy image	18.26	15.29	13.51	11.28	18.16	15.19	13.41	11.21
FCG-Subf1	18.55	15.73	14.04	11.87	18.36	15.55	13.86	11.73
FCG-Subf2 (3×3)	16.95	15.91	15.05	13.69	21.11	18.60	16.94	14.83
FCG-Subf2 (5×5)	15.19	14.64	14.13	13.21	20.38	18.43	17.00	15.03
CWA	23.16	21.14	19.67	17.57	25.21	22.31	20.43	18.00
LPF-E2STT	24.31	22.50	21.05	18.73	27.11	24.29	22.27	19.40
LPF-E2STW	23.11	20.80	19.14	16.72	24.27	21.45	19.51	16.91
LPF-E2ZTT	24.25	22.84	21.48	19.16	29.55	26.32	23.83	20.42
LPF-E2ZTW	24.05	22.37	20.78	18.14	28.94	25.52	22.81	19.23
LPF-E3SnT	23.66	22.09	20.78	18.69	27.31	24.41	22.34	19.56
LPF-E3ZnT	23.12	21.67	20.47	18.55	26.81	24.02	22.05	19.43
LPF-H3SnT	23.64	22.11	20.75	18.67	27.32	24.39	22.37	19.55
LPF-H3ZnT	23.11	21.69	20.44	18.52	26.82	24.01	22.09	19.43
LPF-H1SPT	23.38	21.94	20.71	18.71	27.16	24.34	22.35	19.65
LPF-H1STT	24.38	22.54	21.02	18.63	27.50	24.45	22.28	19.32
LPF-P1SPT	23.50	21.97	20.68	18.62	27.14	24.27	22.26	19.54
LPF-P1STT	24.58	22.42	20.74	18.24	27.15	24.01	21.82	18.85
LPF-P1STW	23.58	20.71	18.78	16.16	24.55	21.34	19.18	16.43
LPF-P1SET	23.41	21.86	20.58	18.56	26.93	24.08	22.10	19.45

*Red font = top 5, bold = top 2 of PSNR values

Table 3: The comparison of PSNR values of some combinations of fuzzy low pass filter and enhanced fuzzy 2-D distance filter against various multiplicative noise level

	PSNR of baboon image (dB)				PSNR of Lena image (dB)			
Image source	σ = 30	σ = 50	σ = 70	σ = 90	σ = 30	σ = 50	σ = 70	σ = 90
Noisy Img	24.19	19.91	17.16	15.13	23.97	19.83	17.15	15.17
FCG-Subf1	25.20	20.81	17.98	15.90	24.84	20.64	17.91	15.89
FCG-Subf2 (3×3)	17.87	17.22	16.44	15.65	24.73	22.21	20.10	18.37
FCG-Subf2 (5×5)	15.59	15.25	14.82	14.35	22.57	21.05	19.49	18.08
CWA	25.60	24.04	22.45	20.92	30.00	26.64	24.04	22.01
LPF-E2STT	25.55	24.13	22.53	20.94	30.30	26.98	24.22	22.06
LPF-E2STW	26.49	24.19	21.96	20.06	29.93	25.90	22.91	20.73
LPF-E2ZTT	24.92	23.66	22.31	20.96	29.99	26.88	24.41	22.48
LPF-E2ZTW	25.02	23.64	22.21	20.89	29.67	26.51	24.16	22.40
LPF-E3SnT	24.83	23.76	22.51	21.21	30.25	27.36	24.85	22.83
LPF-E3ZnT	25.04	23.78	22.43	21.10	30.20	27.08	24.59	22.64
LPF-H3SnT	23.95	23.12	22.11	21.01	29.99	27.38	25.04	23.09
LPF-H3ZnT	24.31	23.38	22.30	21.13	30.12	27.41	25.02	23.05
LPF-H1SPT	24.54	23.55	22.39	21.16	30.17	27.38	24.94	22.94
LPF-H1STT	25.79	24.30	22.72	21.16	30.41	27.24	24.55	22.40
LPF-P1SET	24.69	23.60	22.40	21.16	30.20	27.36	24.92	22.93
LPF-P1SPT	24.71	23.61	22.39	21.13	30.17	27.29	24.82	22.83
LPF-P1STT	26.25	24.32	22.53	20.88	30.19	26.70	23.98	21.88
LPF-P1STW	27.21	23.88	21.41	19.50	29.02	24.86	22.04	20.02

*Red font = top 5, bold = top 2 of PSNR values

T-norm and power-2 weighted averaging (flag code LPF-xxxTW). Overall, 2-Dimension combination tends to be suitable for impulse noise since the PSNRs are higher than others.

For low level impulse noise, the proposed LPF-filters have much better performance compared to the previous single subfilter method. The previous method fails to perform impulse noise reduction, since only very close distance of neighbor pixels to central pixel is consi-dered. Figure 10 shows 1-D, 2-D and 3-D filtered images with PSNR around 24 dB from low level impulse noise. Since the LPFs selected options above are not prepared for impulse noise removal, the performance may be less than other specific design filter (Schulte *et al*., 2006a-c, 2007a, b; Camarena *et al*., 2006; Smolka, 2008) (some of them are for grey scale images). But it could be alternatives for multimedia applications like cellular digital camera, cctv and so on, since the published all-in-one noise removal filters for color image are limited.

Multiplicative noise: To proof the robustness of our LPF method, these filters are also tested on some multiplicative noise (e.g., speckle noise) levels. Visual filter results of baboon image are shown in Fig. 11 and PSNR

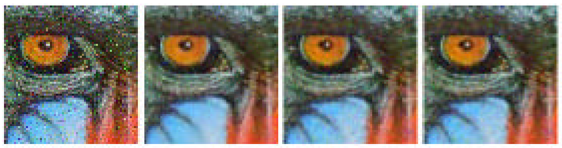

Fig. 10: Image with 5% impulse noise and filters' outputs (combination of 1-D, 2-D or 3-D with other options)

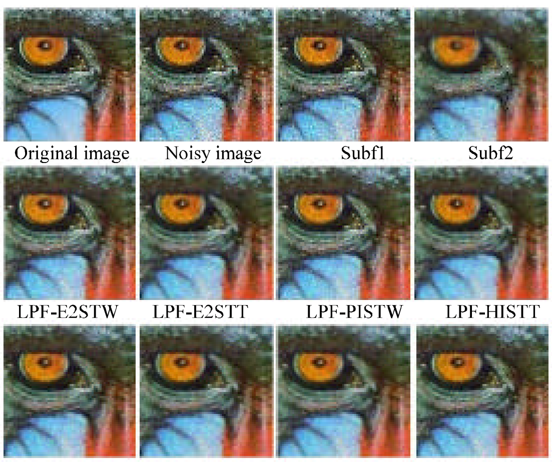

Fig. 11: Part of baboon image with multiplicative noise (speckle noise σ = 50) and filters output

measurements are shown in Table 3. Against multiplicative noise, LPF filters are applicable and perform good image qualities with much better PSNR values than previous method. Moreover, some of them outperform the other Crisp Weighting Filter (CWA) which can be seen visually in Fig. 11 and in Table 3. For colorful image like baboon, the Hamming 1-D option combined with Sigmoid-MF and algebraic T-norm and TSK method results better result for most of multiplicative noise level. But for less color image like lena image, Hamming 3-D combined with Z-MF or Sigmoid-MF performs the best results.

Combination of two or more LPFs or with any other filters that already well-known or proposed are possible to increase output image quality and PSNR level. It depends on user satisfaction to select which LPF combinations that fit to their requirement and the expected image quality they preferred.

CONCLUSION

A Fuzzy Low Pass Filter methodology is proposed to overcome any kinds of common noise (impulse, additive and multiplicative). This Fuzzy LPF methodology is based on the LPF concepts in electronics, since the basic idea is to pass and then give a weight to neighbor pixel that has a strong similar color to central pixel in windows. Each step from fuzzification to defuzzification process becomes options that user can combine to form a preferred LPF. Although not all combinations are showed in this study, basically some of them already show much better result than previous method, both in quality and robustness.

In this study we only concern on utilization of fuzzy concept and compare the fuzzy filter performance to the previous proposed fuzzy filter, especially on high level additive and multiplicative noise and any level of impulse noise. The future researcher will observe the performance of Fuzzy LPF if combined to well-known filters that already published.

REFERENCES

Bourbaki, N., 1987. Topological Vector Spaces. Springer, New York.

Camarena, J.G., V. Gregori, S. Morillas and G. Peris-Fajarnes, 2006. New Method for Fast Detection and Removal of Impulsive Noise Using Fuzzy Metrics. In: ICIAR06, Campilho, A. and M. Kamel (Eds.). LNCS. 4141, Springer-Verlag, Berlin, Heidelberg, pp: 359-369.

Chang, J.Y. and S.M. Lu, 2006. A two-stage fuzzy filtering method to restore images contaminated by mixed impulse and gaussian noises. Lecture Notes Comput. Sci., 4029: 740-749.

Farbiz, F. and M.B. Menhaj, 2000. A Fuzzy Logic Control Based Approach for Image Filtering. In: Fuzzy Technique Image Processing, Kerre, E.E. and M. Nachtegael (Eds.). Springer Physica Verlag, Heidelberg, Germany, pp: 194-221.

Jang, J.S.R., C.T. Sun and E. Mizutani, 1997. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall International, USA., ISBN-10: 0132610663, pp: 614.

Kurniawati, R., J.S. Jin and J.A. Shepherd, 1998. Efficient Nearest-Neighbour Searches Using Weighted Euclidean Metrics. In: Advances in Database BNCOD98, Embury, S.M. *et al*. (Eds.). LNCS. 1405, Springer-Verlag, Berlin, Heidelberg, pp: 64-76.

Kwang, H.L., 2005. First Course on Fuzzy Theory and Applications. Springer-Verlig Berlin, Heidelberg, New York, ISBN 3-540-22988-4, pp: 340.

Liu, P.Y. and H.X. Li, 2004. Fuzzy techniques in image restoration research. Int. J. Comput. Cognition, 2: 131-149.

Megiddo, N. and E. Zemel, 1986. An O(nlogn) randomizing algorithm for the weighted euclidean 1-center problem. J. Algoritms, 7: 358-368.

Nachtegael, M., S. Schulte, D. Van der Weken, V. De Witte and E.E. Kerre, 2006. Gaussian noise reduction in greyscale images. Int. J. Intell. Syst. Technol. Appl., 1: 211-233.

Schulte, S., B. H uysmans, A. Pizurica, E.E. Kerre and W. Phillips, 2006a. A new fuzzy-based wavelet shrinkage image denoising technique. Lecture Notes Comput. Sci., 4179: 12-23.

Schulte, S., M. Nachtegael, V. De Witte, D. Van der Weken and E.E. Kerre, 2006b. A fuzzy impulse noise detection and reduction method. IEEE Trans. Image Process., 15: 1153-1162.

Schulte, S., V. De Witte, M. Nachtegael, D. Van der Weken and E.E. Kerre, 2006c. Fuzzy two-step filter for impulse noise reduction from color images. IEEE Trans. Image Process., 15: 3567-3578.

Schulte, S., V. De Witte, M. Nachtegael, T. Melange and E.E.Kerre, 2007a. A New Fuzzy Additive Noise Reduction Method. In: Image Analysis and Recognition, Kamel, M. and A. Campilho (Eds.). Springer Berlin, Heidelberg, pp: 12-23.

Schulte, S., V. De Witte, M. Nachtegael, D. Van der Weken, E.E. Kerre, 2007b. Fuzzy random impulse noise reduction method. Fuzzy Sets Syst., 158: 270-283.

Schulte, S., V. De Witte and E.E. Kerre, 2007c. A fuzzy noise reduction method for color images. IEEE Trans. Image Process., 16: 1425-1436.

Schulte, S., S. Morillas, V. Gregori and E.E. Kerre, 2007d. A new fuzzy color correlated impulse noise reduction method. IEEE Trans. Image Process., 16: 2565-2575.

Smolka, B., 2008. Peer Group Filter for Impulsive Noise Removal in Color Images. In: CIARP08, Ruiz-Shucloper, J. and W.G. Kropatsch (Eds.). LNCS. 5179, Springer-Verlag, Berlin, Heidelberg, pp: 699-707.

Takagi, T. and M. Sugeno, 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst, Man Cybermetics, 15: 116-132.

Van De Ville, D., M. Nachtegael, D. van der Weken, W.R. Philips, I.L. Lemahieu and E. Kerre, 2000. New fuzzy filter for *Gaussian noise* reduction. Proc. SPIE, 4310: 1-9.

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control, 8: 338-353.