Removing Gaussian Noise from Color Images by Varying the Size of Fuzzy Filters

G.Sudhavani
Associate Professor,
Department of ECE,
RVR & JC College of Engineering, Guntur.

G.Madhuri
Student (M.Tech),
Department of ECE,
RVR & JC College of Engineering, Guntur.

P.Venkateswara Rao
Associate Professor,
Department of CSE,
RVR & JC College of Engineering, Guntur.

K.Satya Prasad
Professor,
Department of ECE,
JNTU Kakinada, Kakinada.

ABSTRACT

The primary issue involved in image and signal processing is to efficiently remove noise from a digital color image while preserving its features. A fuzzy filter is presented for the reduction of additive noise for digital color images. The filter consists of two sub filters. The first sub filter computes fuzzy distances among the color components of the central pixel and its neighborhood. These distances decide in what extent each component should be corrected. The objective of the second sub filter is to calculate the color components differences to retain the fine details of the image. One distance measure as the Minkowski's distance and other as the Absolute distance are selected and compared their performances using Peak Signal to Noise Ratio $(PSNR)$. In this paper the performance of the fuzzy noise filter with two distance measures is compared by changing the noise ratio and window size.

Keywords

Absolute distance, Additive noise, fuzzy filter, fuzzy rule-based systems, Minkowski's distance.

1. INTRODUCTION:

To carry and transfer information images are one of the most important tools. The research field of image processing not only comprehends technologies for capture and transfer images, but also techniques to anatomize these images. Image processing is a growing research field. Images have always played an important and essential role in human life, not only as a way to communicate, but also for other applications like scientific, commercial, military and industrial. In this paper we will focus on a technique that removes noise from color images using fuzzy logic.

Especially there are three types of noise exist: Gaussian noise, Salt & pepper noise, Speckle noise. Gaussian noise is statistical noise that has a probability density function of the normal distribution (also known as Gaussian distribution). It is most commonly used as additive white Gaussian noise (AWGN). Salt & pepper noise itself as randomly occurring white and black pixels. A noise reduction method which uses a median filter is suitable for salt & pepper noise reduction. This type of noise can be described as dark pixels on bright background and bright pixels on dark background. Images corrupt with Salt and pepper noise when quick transients, such as faulty switching, take place. Speckle noise is also known as multiplicative noise, i.e. the gray levels of speckle noisy pixels are in direct proportion to the local gray levels in any area of an image.

Distance computation plays an important role in removal of noise from color images. The distance among two couples is calculated according to the Minkowski's and absolute distances. To calculate scaling factors of the filter Takagi-sugeno fuzzy model is used. In this method three 2-D distances (distance between red–green, red–blue, and green–blue) along with three fuzzy rules are used to compute the scaling factors. The concept behind evolution of fuzzy rules is to assign large scaling factors to the neighbors that have similar colors as the centre pixel.

Fuzzy set theory and fuzzy logic [1] offer us powerful tools to describe and process human knowledge. Some of the fuzzy filters designed earlier for noise reduction are a new fuzzy filter for image enhancement [2], noise adaptive soft switching median filter [3], noise reduction by fuzzy image filtering [4], fuzzy two-step color filter [5], a fuzzy impulse noise detection and reduction method [6], and so on. The different methods using fuzzy filters for reduction of noise are explained in the literature [7]-[9]. Most of these state-of-art methods are mainly developed for the reduction of fat-tailed noise like impulse noise. These filters are able to outperform rank-order techniques (such as median based filters). Nevertheless, most of the current fuzzy techniques do not produce convincing results for additive noise, which is illustrated in [10], [11]. Another shortcoming of the current methods is that most of these filters are especially developed for grayscale images. It is, of course, possible to extend these filters to color images. However, this introduces many artifacts, especially on edge or texture elements. Therefore, this paper presents a new and simple fuzzy technique for filtering color images corrupted with narrow-tailed and medium narrow-tailed noise (e.g., Gaussian noise) without introducing these artifacts.

A digital color image (denoted as X) can be represented by using a certain color space (e.g., RGB, HSV, L*a*b*). The most commonly used color space is called "RGB," which has one overwhelmingly important characteristic i.e., every scanner and digital camera can produce images with RGB-encoded color. And also every image-handling device and color-aware application can handle images with RGB-encoded color. It is the official default color space for the World Wide Web. Colors in the RGB model are represented by a 3-D vector, with the first vector element being the red, the second being the green and the third being the blue, respectively. The red, green and blue components are called the three primary components, each component is quantized to the range $[0,\ 2^m - 1]$. In practice, a digital color image X can be represented by a 2-D array of vectors where an address (k,l) defines a position in X, called a pixel or picture element. If $X(k,l,1)$ denotes the red component, $X(k,l,2)$ the green component and $X(k,l,3)$ the blue component of a pixel at position (k,l) in an (noise-free) image X, then we can denote the noisy color image NI at position (k,l) as follows:

$$\left[NI(k,l,1)\ NI(k,l,2)\ NI(k,l,3)\right] = \left[\left(X(k,l,1)+\partial_1\right)\ \left(X(k,l,2)+\partial_2\right)\ \left(X(k,l,3)+\partial_3\right)\right] \quad (1)$$

With ∂_1, ∂_2 and ∂_3 three separate randomly Gaussian distributed values with means $(\mu_1,\ \mu_2 \text{ and } \mu_3)$ and standard deviations $(\sigma_1,\ \sigma_2 \text{ and } \sigma_3)$, respectively. NI is the color image that is corrupted with additive white Gaussian noise.

2. FIRST SUBFILTER WITH FUZZY RULES:

While capturing or transmission of images the effect of noise may degrade the quality of the image so it is desirable to perform some sort of noise reduction on an image. The general idea for noise reduction is to average a pixel using other pixel values from its neighborhood, but simultaneously take into account the important image structures such as edges and color component distances, which should not be destroyed by the filter. The goal of the first subfilter of is to discriminate between local variations due to noise and due to image structures such as edges. This is realized by using the color component distances instead of component differences as done in most current filters. For example, to filter a certain red component at position (k,l), we use the distances between red–green and red–blue of a certain neighborhood centered at (k,l) instead of just a neighborhood in the red component array. The difference between this new proposed filter and other vector based approaches as [12]–[13] is that we do not calculate the 3-D distances between pixels (distances between two pixels where a pixel is considered as a vector), but we use three 2-D distances (distances between red–green, red–blue, and green–blue) together with three fuzzy rules to calculate the weights used for the Takagi–Sugeno fuzzy model [14] explained in the next section.

2.1 Distance determination:

The red, green, and blue component at a certain pixel position of a noisy input image NI is denoted as $NI(k,l,1)$, $NI(k,l,2)$ and $NI(k,l,3)$, respectively. So, for each pixel position, we have three components that define the color. For each pixel position (k,l) we define the following pairs: the pair red and green denoted as $RG(k,l) = \left(NI(k,l,1), NI(k,l,2)\right)$ the pair red and blue denoted as $RB(k,l) = \left(NI(k,l,1), NI(k,l,3)\right)$ and the pair green and blue denoted as $GB(k,l) = \left(NI(k,l,2), NI(k,l,3)\right)$. To filter the current image pixel, we use a window of size $(2I+1)\times(2I+1)$ centered at position (k,l). Next we assigned certain scaling factors to each of the pixels in the window. The scaling factors $w(k+i,l+j,1)$, $w(k+i,l+j,2)$ and $w(k+i,l+j,3)$ are for the red, green and blue components at position $(k+i,l+j)$ respectively. These scaling factors are assigned according to the following fuzzy rules, where a Takagi-sugeno fuzzy model [10] was used.

Fuzzy rule 1: The first fuzzy rule defines the weight $w(k+i,l+j,1)$ for the red component of the neighbor $NI(k+i,l+j,1)$, i.e.

IF the distances are small between the pairs $RG(k,l)$, $RG(k+i,l+j)$ and $RB(k,l)$, $RB(k+i,l+j)$ **THEN** the scaling factor $w(k+i,l+j,1)$ is large.

Fuzzy rule 2: The second fuzzy rule defines the weight $w(k+i,l+j,2)$ for the green component of the neighbor $NI(k+i,l+j,2)$, i.e.

IF the distances are small between the pairs $RG(k,l)$, $RG(k+i,l+j)$ and $GB(k,l)$, $GB(k+i,l+j)$ **THEN** the scaling factor $w(k+i,l+j,2)$ is large.

Fuzzy rule 3: The third fuzzy rule defines the weight $w(k+i,l+j,3)$ for the blue component of the neighbor $NI(k+i,l+j,3)$, i.e.

IF the distances are small between the pairs $RB(k,l)$, $RB(k+i,l+j)$ and $GB(k,l)$, $GB(k+i,l+j)$ **THEN** the scaling factor $w(k+i,l+j,3)$ is large.

The concept behind these simple fuzzy rules is to assign large scaling factors to the neighbors that have similar colors as the center pixel. In color images it is optimum to consider pixels as colors instead of taking them as three separate color components. When only the separate color components are considered, more artifacts are introduced, especially on the contour of objects. The distance between two couples is calculated according to the Minkowski's distances. This is illustrated in equation (2) for red–green pair

$$D\left(RG\left(k,l\right),RG\left(k+i,l+j\right)\right) = \left[\begin{array}{l}\left(NI\left(k+i,l+j,1\right)-NI\left(k,l,1\right)\right)^{2}+\\ \left(NI\left(k+i,l+j,2\right)-NI\left(k,l,2\right)\right)^{2}\end{array}\right]^{1/2} \quad (2)$$

Another distance measure which is the Absolute distance is also illustrated in the following equation (3) for red-green pair.

$$D\left(RG\left(k,l\right),RG\left(k+i,l+j\right)\right) = \left|NI(k+i,\ l+j,\ 1)-NI(k,\ l,\ 1)\right| + \left|NI\left(k+i,l+j,2\right)-NI\left(k,l,2\right)\right| \quad (3)$$

To compute the value that expresses the degree to which the distance of two couples is small, we will make use of the fuzzy set *small*. Fuzzy sets are commonly represented by membership functions. From such functions we can derive the corresponding membership degrees. If the distance between two couples has a membership degree one (zero) in the fuzzy set *small*, it means that this distance is considered as (not) small for sure. Membership degrees between zero and one indicate that we do not know for sure if such distance is small or not, so there is some kind of uncertainty. For more background information about fuzzy logic, we refer to [1], [15], and [16]. A membership function *small*, denoted as μ_{small} is shown in Fig. 1. An alternative notation for the membership function *small* is given in equation (4)

$$\mu_{small}\left(y\right) = \begin{cases}\left(\dfrac{t-y}{t}\right)^{2}, & \text{if } y \le t\\ 0, & \text{if } y > t\end{cases} \quad (4)$$

With $t \in \left]0,\sqrt{2}\right]$ for the normalized input image NI

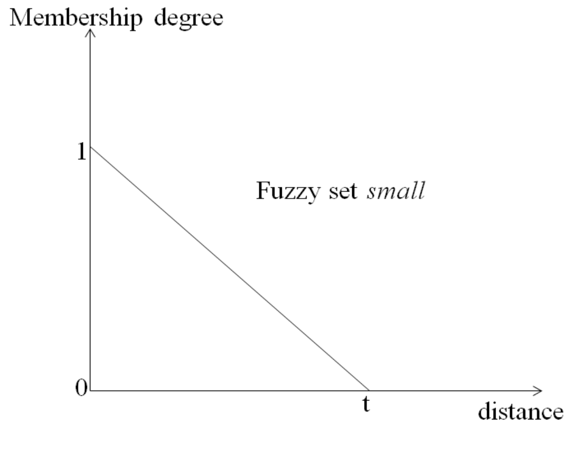

Figure1: Membership function small

We have define three such fuzzy sets, one for each pair (one for the pair red–green, another for the pair red–blue, and the last one for the pair green–blue). All these fuzzy sets depend on the parameter as seen in (4) and Fig. 1. These parameters are denoted as P_{RG}, P_{RB} and P_{GB}. The parameter P_{RG} is determined adaptively as follows. Similarly the remaining two parameters are calculated for remaining two pairs RB and GB.

$$P_{RG}\left(k,l\right) = \max_{k,l \in \Omega}\left(\gamma_{RG}\left(k,l,i,j\right)\right) \quad (5)$$

Where Ω defines the $\left(2I+1\right) \times \left(2I+1\right)$ neighborhood around the central pixel, i.e. $i,j \in \left\{-I,\ -I+1,...,0,...,I-1,I\right\}$, and where $\gamma_{RG}\left(k,l,i,j\right)$, $\gamma_{RB}\left(k,l,i,j\right)$ and $\gamma_{GB}\left(k,l,i,j\right)$ were used as convenient notation for the distances.(i.e., $\gamma_{RG}\left(k,l,i,j\right) = D\left(RG(k,l),RG\left(k+i,l+j\right)\right)$, similarly for $\gamma_{RB}\left(k,l,i,j\right)$ and $\gamma_{GB}\left(k,l,i,j\right)$).

So, the parameters (for the spatial position $\left(k,l\right)$) $P_{RG}\left(k,l\right)$, $P_{RB}\left(k,l\right)$, and $P_{GB}\left(k,l\right)$ are equal to the maximal distance between the red–green, red–blue, and green–blue pairs in a $\left(2I+1\right) \times \left(2I+1\right)$ neighborhood around the centre pixel $\left(k,l\right)$.The γ's of expression (6) are used to calculate the scaling factors as introduced by fuzzy rules. In these rules, we can observe an intersection of two fuzzy sets, which is generally specified by mapping T. We have used the algebraic product T-norms. This means for instance that the fuzzification of the antecedent of fuzzy rule 1 is $\mu_{small1}\left(\gamma_{RG}\left(k,l,i,j\right)\right) \cdot \mu_{small2}\left(\gamma_{RB}\left(k,l,i,j\right)\right)$, where μ_{small1} and μ_{small2} are equal to membership function *small*, shown in expression (2), with parameters $P_{RG}\left(k,l\right)$ and $P_{RB}\left(k,l\right)$, respectively. The obtained value $\left(\mu_{small1}\left(\gamma_{RG}\left(k,l,i,j\right)\right) \cdot \mu_{small2}\left(\gamma_{RB}\left(k,l,i,j\right)\right)\right)$ is called the activation degree of the fuzzy rule and is used to obtain corresponding scaling factor, i.e. $w\left(k+i,l+j,1\right) = \mu_{small1}\left(\gamma_{RG}\left(k,l,i,j\right)\right) \cdot \mu_{small2}\left(\gamma_{RB}\left(k,l,i,j\right)\right)$, similarly the remaining two scaling factors are calculated by using activation degrees.

Where μ_{small1}, μ_{small2} and μ_{small3} are equal to membership function *small*, shown in expression (2), with parameters $P_{RG}\left(k,l\right)$, $P_{RB}\left(k,l\right)$ and $P_{GB}\left(k,l\right)$ respectively. The output of the first subfilter can finally be illustrated for the red component, where the output image is denoted as A, i.e.,

$$A\left(k,l,1\right) = \frac{\sum_{i=-I}^{+I}\sum_{j=-I}^{+I} w\left(k+i,l+j,1\right) \cdot NI\left(k+i,l+j,1\right)}{\sum_{i=-I}^{+I}\sum_{j=-I}^{+I} w\left(k+i,l+j,1\right)} \quad (6)$$

The filtering method for the green and blue component is similar to the one above.

3. SECOND SUBFILTER TO RETAIN FINE DETAILS:

The Second subfilter is a complementary filter to the first one. The objective of this subfilter is to improve the first method by reducing the noise in the color components differences without deteriorating the fine details of the image. This is realized by calculating the local differences in the red, green, and blue environment individually. These differences are then combined to calculate the local estimation of the centre pixel.

Analogous to the first subfilter, we use a window of size $(2J+1)\times(2J+1)$, where J is not necessarily equal to I, centered at (k,l) to filter the current image pixel at position (k,l). Next, we calculate the local differences (also called gradients or derivatives) for each element of the window denoted as BD_R, BD_G and BD_B for the red, green, and blue environment, respectively. If the output image of the previous subfilter is denoted as A, then the local difference for red component is calculated by using (7). Similarly the remaining differences are calculated.

$$BD_R(i,j) = A(k+i,l+j,1) - A(k,l,1) \tag{7}$$

For all $i, j \in \{-J,...,0,....,+J\}$. These differences are finally combined to calculate the following correction terms:

$$e(i,j) = \frac{1}{3}\left(BD_R(i,j) + BD_G(i,j) + BD_B(i,j)\right) \tag{8}$$

i.e., we calculate the average of the difference for the red, green, and blue component at the same position.

3.1. Output of the second subfilter:

Finally, the output of the second subfilter, denoted as B, for red component is determined as follows. Like red component the outputs for green and blue components are determined.

$$B(k,l,1) = \frac{\sum_{i=-J}^{+J}\sum_{j=-J}^{+J}\left(A(k+i,l+j,1)+e(i,j)\right)}{(2J+1)^2} \tag{9}$$

With $B(k,l,1)$ is the red component of the output image and where $e(i,j)$ is the correction term for the neighboring pixel $A(k+i,l+j,1)$.

4. SIMULATION RESULTS:

As a measure of objective similarity between a filtered image and the original one, we use the peak signal-to-noise ratio $(PSNR)$ in decibels (dB)

$$PSNR(FI,NI) = 10\log_{10}\frac{S^2}{MSE(FI,NI)} \tag{10}$$

$$MSE(FI,NI) = \frac{\sum_{c=1}^{3}\sum_{k=1}^{N}\sum_{l=1}^{M}\left[NI(k,l,c) - FI(k,l,c)\right]^2}{3NM} \tag{11}$$

The numerical results for Minkowski's distance and for absolute distance in terms of *PSNR* values are shown in table I and table II. Table I shows the numerical results for colored Lena image of size (512x512) for different noise levels with respect to variation in window size. Table II shows the numerical results for colored Baboon eye image of size (512x512) for different noise levels with respect to variation in window size. Figure-6 to Figure-16 shows the visual observations for the results tabulated in table I. Figure-2 to figure-5 shows the graphical representation of variation in PSNR values for different size of windows with respect to different levels of noise for both Minkowski's and Absolute distances.

Table-1

Comparison of proposed method with other filtering methods for the (512x512) colored Lena image

		***PSNR* (dB)**				
		Lena image				
σ		**5**	**10**	**20**	**30**	**40**
Noisy		37.6898	32.6498	29.9416	29.1524	28.7689
Minkowski's distance	(3x3,3x3)	32.7361	30.6698	29.0070	28.4260	28.1320
	(3x3,5x5)	31.6998	30.1381	28.7946	28.2851	28.0453
	(3x3,7x7)	31.0647	29.8505	28.7020	28.2438	28.0057
	(3x3,9x9)	30.5690	29.6030	28.6167	28.2005	27.9810
Absolute distance	(3x3,3x3)	38.5100	37.4666	35.1361	33.9245	33.2771
	(3x3,5x5)	37.1350	36.6597	35.0819	34.0835	33.4906
	(3x3,7x7)	36.3272	36.0885	34.9429	34.0845	33.5494
	(3x3,9x9)	35.7535	35.6591	34.8166	34.0916	33.5905

Table II

Comparison of proposed method with other filtering methods for the (512x512) colored Baboon eye image

		***PSNR* (dB)**				
		Baboon image				
σ		**5**	**10**	**20**	**30**	**40**
Noisy		37.1812	32.0382	29.4092	28.5992	28.2410
Minkowski's distance	(3x3,3x3)	34.0968	30.8973	28.8649	28.2453	27.9835
	(3x3,5x5)	31.7804	30.0869	28.6232	28.1142	27.8922
	(3x3,7x7)	30.5080	29.5292	28.4631	28.0407	27.8392
	(3x3,9x9)	29.8011	29.1596	28.3276	27.9694	27.7971
Absolute distance	(3x3,3x3)	44.7091	41.0410	36.4986	34.7709	33.8436
	(3x3,5x5)	42.6113	40.6683	36.9627	35.2639	34.3868
	(3x3,7x7)	40.7347	39.7010	36.9103	35.3839	34.4346
	(3x3,9x9)	39.4863	38.8561	36.7473	35.3917	34.4542

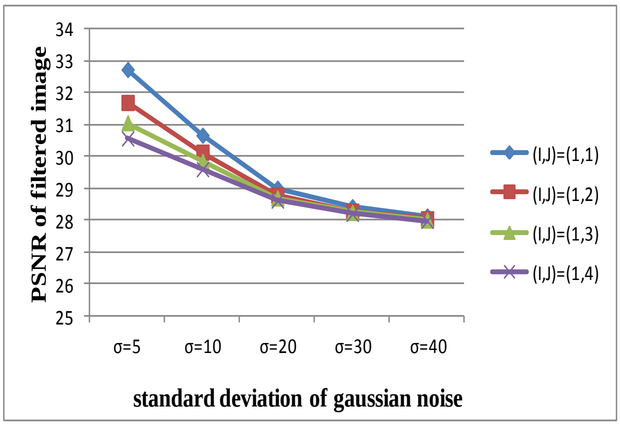

Figure2: Graphical representation of Minkowski's distance for Lena image for different noise levels and window size

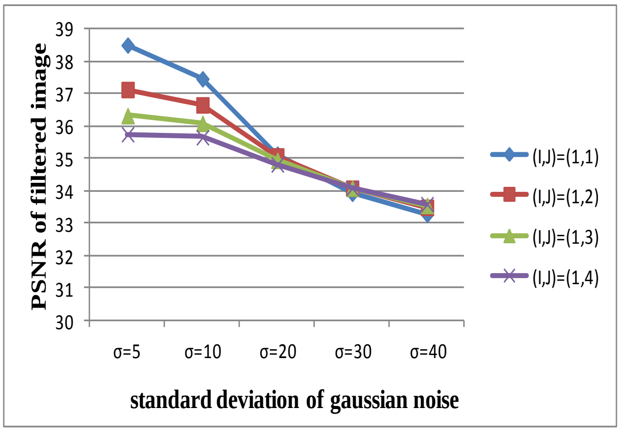

Figure3: Graphical representation of Absolute distance for Lena image for different noise levels and window size

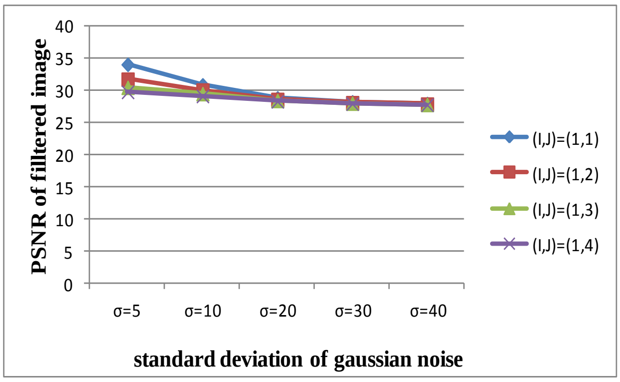

Figure4: Graphical representation of Minkowski's distance for Baboon eye image for different noise levels and window size

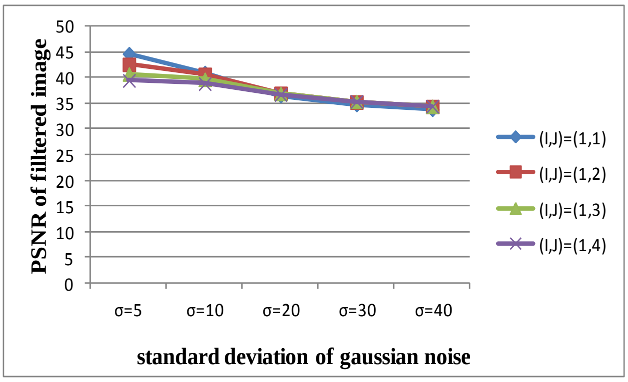

Figure5: Graphical representation of Absolute distance for Baboon eye image for different noise levels and window size

Figure6: Original Lena image

Figure7: Illustration of Minkowski's distance results for Lena image with $\sigma = 5$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window

Figure8: Illustration of Minkowski's distance results for Lena image with $\sigma = 10$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window

Figure9: Illustration of Minkowski's distance results for Lena image with $\sigma = 20$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window

Figure10: Illustration of Minkowski's distance results for Lena image with $\sigma = 30$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window.

Figure11: Illustration of Minkowski's distance results for Lena image with $\sigma = 40$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window

Figure12:Illustration of Absolute distance results for Lena image with $\sigma = 5$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window.

Figure13:Illustration of Absolute distance results for Lena image with $\sigma = 10$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window.

Figure14:Illustration of Absolute distance results for Lena image with $\sigma = 20$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window.

Figure15:Illustration of Absolute distance results for Lena image with $\sigma = 20$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window.

Figure16:Illustration of Absolute distance results for Lena image with $\sigma = 20$ (a) Noisy image (b) Corresponding output image for (3x3, 3x3) window (c) Corresponding output image for (3x3, 5x5) window (d) Corresponding output image for (3x3, 7x7) window (e) Corresponding output image for (3x3, 9x9) window.

5. CONCLUSION:

This paper proposed a fuzzy noise filter for removal of additive noise for any type of digital color images. The main advantage of this filter is, it is showing better results at higher levels of noise. Numerical measures and visual observations have shown convincing results. By observation from tables I & II it is clear that the fuzzy noise filter with varying window sizes is performed well and also it is showing better results at higher noise levels.

6. REFERENCES:

[1] E. E. Kerre, Fuzzy Sets and Approximate Reasoning, Xian, China: Xian Jiaotong Univ. Press, 1998.

[2] Farzam Farbiz, Mohammad Bager Menhaj, Seyed A. Motamedi, and Martin T. Hagan "A New Fuzzy Logic Filter for Image Enhancement" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 30, NO. 1, FEBRUARY 2000.

[3] How-Lung Eng, Student Member, *IEEE*, and Kai-Kuang Ma "Noise Adaptive Soft-Switching Median Filter" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 2, FEBRUARY 2001

[4] Dimitri Van De Ville, Mike Nachtegael, Dietrich Van der Weken, Etienne E. Kerre, Wilfried Philipsand Ignace Lemahieu "Noise Reduction by Fuzzy Image Filtering"IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 11, NO. 4, AUGUST 2003.

[5] Stefan Schulte, Valérie De Witte, Mike Nachtegael, Dietrich Van der Weken, and Etienne E. Kerre "Fuzzy Two-Step Filter for Impulse Noise Reduction From Color Images" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 15, NO. 11, NOVEMBER 2006.

[6] Stefan Schulte, Mike Nachtegael, Valérie De Witte, Dietrich Van der Weken, and Etienne E. Kerre "A Fuzzy Impulse Noise Detection and Reduction Method" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 15, NO. 5, MAY 2006.

[7] Stefan Schulte "Fuzzy and Nonlinear Restoration and Analysis Techniques for Digital Images".

[8] P.Venkatesan & G.Nagarajan "Removal of Gaussian and Impulse Noise in the Color Image Progression with Fuzzy Filters" International Journal of Electronics Signals and Systems (IJESS), ISSN: 2231- 5969, Vol-3, Iss-1, 2013.

[9] Tom Mélange, Mike Nachtegael, and Etienne E. Kerre "Fuzzy Random Impulse Noise Removal from Color Image Sequences" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 20, NO. 4, APRIL 2011.

[10] S. Schulte, B. Huysmans, A. Piˇzurica, E. E. Kerre, and W. Philips, "A new fuzzy-based wavelet shrinkage image denoising technique," Lecture Notes Comput. Sci., vol. 4179, pp. 12–23, 20

[11] Stefan Schulte, Valérie De Witte, and Etienne E. Kerre "A Fuzzy Noise Reduction Method for Color Images" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 16, NO. 5, MAY 2007 1425

[12] C. Vertan and V. Buzuloiu, "Fuzzy nonlinear filtering of color images," in Fuzzy Techniques in Image Processing, E. E. Kerre and M. Nachtegael, Eds., 1st ed. Heidelberg, Germany: Physica Verlag, 2000, vol. 52, pp. 248–264.

[13] S. M. Guo, C. S. Lee, and C. Y. Hsu, "An intelligent image agent based on soft-computing techniques for color image processing," Expert Syst Appl., vol. 28, pp. 483–494, Apr. 2005

[14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst., Man, Cybern., vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[15] L. A. Zadeh, "Fuzzy sets," Inf. Control, vol. 8, no. 3, pp. 338–353, 1965.

[16] ——, "Fuzzy logic and its application to approximate reasoning," Inf. Process., vol. 74, pp. 591–594, 1973.