

REMOVING IMAGE NOISE BASED ON FUZZY LOGIC CONCEPT

Nada N. Saleem* Warqaa Y.IbraheemSahar Kh. Ahmed****

**College of Computer Sciences and Mathematics/ Software Engineering Dept.*

**College of Engineering/ Computer Engineering Dept.*

ABSTRACT:In the present research algorithms employing fuzzy logic on median and mean filters for improving impulse noise removal performance for image processing have been developed. These algorithms can achieve significantly better image quality and capable of preserving the intricate details of the image than classical arithmetic and mean filters when the images are corrupted by impulse noise. The proposed fuzzy image filters (Filter1, Filter2 and Filter3) are based on a combination of fuzzy impulse detection and restoration of corrupted pixels. Fuzzy knowledge base required for detection of impulses. The research also presents an adaptive fuzzy filter system (filter 4) for noisy image enhancement combining smoothing and sharpening. The method is automatically obtaining an optimum parameter value adaptively by evaluating the local features. We present the results for different levels of impulse noise corruption on several real images, and the performance of our proposed filters is compared with statistical noise removal methods to show the effectiveness of the proposed techniques.

Keywords: fuzzy logic, image enhancement, impulse noise.

Introduction

As the population in the Internet swells, more and more kinds of digital data are being delivered in the network.

No matter what kind of digital data it is: text, sound and image all included, it may very easily be damaged in unsound circuits or on unsafe channels. This blur caused during transmission is called noise. Noise is a big common problem in electronic communication; it may cause data loss or misplacement [1].

The most common noise in the Internet communication is impulse noise [2][3], which usually corrupts pixels into relatively low or high pixel values when we transfer

uncompressed image on the Internet [4].

Preprocessing steps is important to eliminate noise before subsequent processing such as object recognition, edge detection, image segmentation, feature extraction and pattern recognition [4][5]. Conventional image enhancement techniques such as mean and median filtering have been employed in various applications, but have the disadvantage of performing poorly when it comes to preserving intricate details of an image [6][7].

Towards overcoming the disadvantages, fuzzy logic which mimics human reasoning and tolerance ambiguities well are

increasing being looked into as alternatives to these conventional techniques. When using fuzzy logic to any particular application fuzzy rule base should be decided and the parameters of the membership functions used in the inference process should be selected.

Usually, human intuition is used to decide these rule bases and parameters with the intention of seeing a human-like behavior from the system [8].

In this research, fuzzy logic is employed to show how it could be used in practical digital image processing system to remove heavy impulse noise from corrupted images. This is specifically true in automated processes where the human intervention is a minimum.

Fuzzy Image Processing

Fuzzy image understanding:

To apply the idea of fuzzy set to image processing problems, one should develop a new image understanding. We need a new image definition, some ways to fuzzify the images and their features, and finally, an extension of digital topology, which plays a pivotal role in image representation and local operations respectively.

An image X of size $M \times N$ with L gray levels $g: 0,1,2,\dots,L-1$, can be defined as an array of fuzzy singletons indicating the membership values μ_{mn} of each image point X_{mn} regarding to a predefined image property .

$$X = \bigcup_{m=1}^M \bigcup_{n=1}^N \frac{\mu_{mn}}{X_{mn}} \quad (1)$$

The definition of the membership value μ_{mn} depends on the specific requirements of actual application and the corresponding expert knowledge.

Fuzzy image processing is a kind of nonlinear and knowledge-based image processing. The difference to other methodologies is that fuzzy techniques operate on membership values. Therefore image fuzzification is always the first processing step [9][11].

Structure of fuzzy image processing:

Fuzzy image processing consists generally of three steps fuzzification (image coding), operations in the membership plane, and finally defuzzification (decoding of results).

Fuzzification does mean that we assign the image (its gray levels, features, segments,...) with one or more membership values regarding to the interesting properties. After transformation of image into the membership plain, a suitable fuzzy approach aggregates AND/OR modifies the membership values. To achieve new results the output of membership plain should be decoded (defuzzification). It means that the membership values are retransformed into the gray level plane [10].

Fuzzy Image Noise Detection

Fuzzy logic is a very powerful tool which has strong capabilities to deal with set of data having vagueness and complex variation. [8]

Thus, fuzzy knowledge base can be employed to make decision of the noisiness of the pixels in an image corrupted with high impulsive noise and assign each

pixel of the noisy image a degree of noisiness (between 0&1) which would represent as to what extent a given pixel has been corrupted.

The Proposed Fuzzy Image Filters

Filter 1: Median fuzzy filter

Median filter was initially introduced to eliminate impulse noise [12]. To recover original pixel values from corrupted image we develop a fuzzy noise detection process. We use fuzzy reasoning to detect if the pixel is noisy and to reconstruct the pixel by using median filter.

Step1: The source image is partitioned into overlapping 2-D blocks of size 5×5 , which are processed sequentially in a raster scan fashion, left to right and top to bottom sliding window of 5×5 is shown below.

$$\begin{bmatrix} W_{01} & W_{02} & W_{03} & W_{04} & W_{05} \\ W_{06} & W_{07} & W_{08} & W_{09} & W_{10} \\ W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{16} & W_{17} & W_{18} & W_{19} & W_{20} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \end{bmatrix}$$

Let us suppose the pixel to be processed is W_{00} .

Step2: In this step of method, we divide the 5×5 neighborhood region consisting of 25 pixels into eight sub-regions of 3×3 pixel windows, each containing nine pixels as shown in Figure 1.

W_{01}	W_{02}	W_{03}	W_{04}	W_{05}
W_{06}	W_{07}	W_{08}	W_{09}	W_{10}
W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
W_{16}	W_{17}	W_{18}	W_{19}	W_{20}
W_{21}	W_{22}	W_{23}	W_{24}	W_{25}

(a)

W_{01}	W_{02}	W_{03}	W_{04}	W_{05}
W_{06}	W_{07}	W_{08}	W_{09}	W_{10}
W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
W_{16}	W_{17}	W_{18}	W_{19}	W_{20}
W_{21}	W_{22}	W_{23}	W_{24}	W_{25}

(b)

Figure 1. Eight (8) sub-region created inside the 5×5 sliding window

Step3: Next we calculate the median values of each of these sub regions of the 5×5 neighborhood as shown below:

$M_1 = \text{median-of } (W_k \mid k=0, 1, 2, 3, 6, 7, 8, 11, 12)$

$M_2 = \text{median-of } (W_k \mid k=0, 2, 3, 4, 7, 8, 9, 12, 13)$

$M_3 = \text{median-of } (W_k \mid k=0, 3, 4, 5, 8, 9, 10, 13, 14)$

$M_4 = \text{median-of } (W_k \mid k=0, 6, 7, 8, 11, 12, 15, 16, 17)$

$M_5 = \text{median-of } (W_k \mid k=0, 8, 9, 10, 13, 14, 17, 18, 19)$

$M_6 = \text{median-of } (W_k \mid k=0, 11, 12, 15, 16, 17, 20, 21, 22)$

$M_7 = \text{median-of } (W_k \mid k=0, 12, 13, 16, 17, 18, 21, 22, 23)$

$M_8 = \text{median-of } (W_k \mid k=0, 13, 14, 17, 18, 19, 22, 23, 24)$

Step 4: We transfer the region within the 5×5 neighborhood window into a virtual 3×3 window as shown below by using the median values M_1, \dots, M_8 .

$$\begin{bmatrix} M_1 & M_2 & M_3 \\ M_4 & W_{00} & M_5 \\ M_6 & M_7 & M_8 \end{bmatrix}$$

Now the median value calculated above are used during the noise detection process instead of the actual pixels, which can be corrupted themselves.

Step5: The input variables to the fuzzy knowledge base are the intensity median differences given by

$$D_j = \text{abs}(M_j - W_{00}) \quad j=1, \dots, 8 \quad (2)$$

Since gray scale images are being processed, the input variable interval for D_j is [0-255]

Step6: To evaluate the differences calculated in step 5, two input fuzzy sets, name Difference_High (DH) and Difference_Low (DL) are define with Gaussian membership functions, as shown in Figure 2.

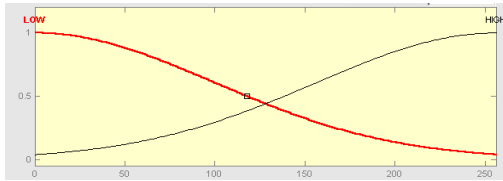


Figure 2. Fuzzy sets for input variables to the Fuzzy Knowledge Base

The output variable of the fuzzy inference engine noisiness $\in [0,1]$, is a number of two trapezoidal fuzzy sets, Very_Low (VL) and Very_High (VH) given as Figure 3:

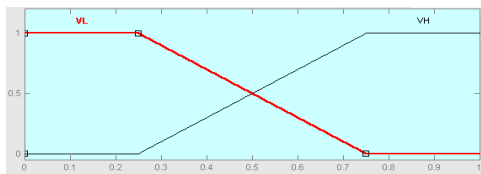


Figure 3. Fuzzy sets for output of Fuzzy Inference engine

There are 16 rules in the fuzzy rule base as shown in table (1). The rules are built using intuition of how the

intensity differences determine the existent of the noisy pixel.

Table 1: fuzzy rule-base

Rule number	D_1	D_2	D_3	D_4	Noisiness
Rule 1	DL	DL	DL	DL	VL
Rule 2	DL	DL	DL	DH	VL
Rule 3	DL	DL	DH	DL	VL

Rule 16	DH	DH	DH	DH	VH

For example in the antecedent side of rule 16 the median difference intensity of D_1 , D_2 , D_3 and D_4 is High then in the consequent side the noisiness is Very High .

Each rule is designed to deal with a particular pattern of intensity difference among the virtual neighboring pixels M_1, \dots, M_8 and the pixel W_{00} .

Given a 16 fuzzy rule base, the output noisiness as given in equation (3) uses a singleton fuzzifier, Mamdani product inference engine and center average defuzzifier.

$$\text{Noisiness} = \frac{\sum_{l=1}^R \bar{y}^l (\prod_{i=1}^n \mu_{A_i^l} D_i)}{\sum_{l=1}^R (\prod_{i=1}^n \mu_{A_i^l} D_i)} \quad (3)$$

Where R: represent no. of rules in the rule base.

\bar{y}^l :represent the center of output sets.

Step7: The output of the fuzzy inference engine which represents the noisiness of the pixel under consideration in the corrupted image is used to calculate the new gray level of the pixel which will replace the corrupted pixel.

$$W_{00}^{(new)} = N_d \times R_{value} + (1 - N_d) \times W_{00} \quad (4)$$

Where $R_{\text{value}} = \text{median}(M_j \text{ and } j=1, \dots, 8)$ is the reconstructed value, N_d is the degree of noisiness of the selected pixel.

Step8: Go to second step with new 5×5 sliding window from the corrupted image, until reach last block in the image.

4.2 Filter 2: Mean fuzzy filter

In this method we use the algorithm of the fuzzy median filter for noise detection process which explained in previous section to recover the original pixel value from the corrupted image. But we calculate in this fuzzy filter the mean values of each sub regions of the 5×5 sliding windows and the mean values for the variable R_{value} which used in equation (4) to calculate the enhancement pixel.

4.3 Filter3: Hybrid Fuzzy Filter

The algorithm compare the intensity difference between a selected pixel and the neighboring pixels in a sliding window of size 3×3 , then fuzzy reasoning used to detect if the pixel is noisy and to reconstruct the pixel by using mean and median filter.

Step1: The source image is partitioned into overlapping 2-D block of 3×3 , which are processed sequentially in a raster scan fashion, left to right and top to bottom.

A 3×3 sliding window is shown below.

$$\begin{bmatrix} W_1 & W_2 & W_3 \\ W_4 & W_0 & W_5 \\ W_6 & W_7 & W_8 \end{bmatrix}$$

Step2: Calculate the input variables to the fuzzy knowledge base by take the intensity differences given by

$$X_i = \text{abs}(W_i - W_0) \quad i=1,2,3,4 \quad (5)$$

Where $-W_0$ is the pixel to be processed.

$-W_i$ represents the pixel W_1 to W_4 that have been processed.

The intensity difference values for pixels W_5 to W_8 are uncertain because they may be corrupted by noise, so they are not used as fuzzy inputs.

Step3: Fuzzification the four input variables using two input fuzzy sets named Difference_High (DH) and Difference_Low (DL) using trapezoidal membership functions as show in Figure 4.

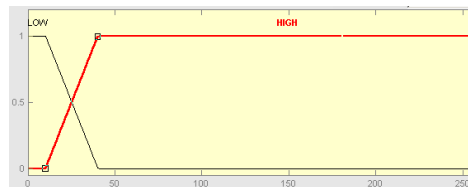


Figure 4. Fuzzy membership functions for input variables to the Fuzzy Knowledge Base

The output variable of the fuzzy inference engine noisiness $\in [0,1]$, is a member of two fuzzy set Very_Low (VL) and Very_High (VH) shown in Figure 3.

There are 16 rules in fuzzy rule base; these rules are built using institution of how the intensity difference determines the existence of a noisy pixel. Each rule has been designed to deal with a particular pattern of intensity

difference among the neighboring pixels.

The output noisiness given in equation (3) uses a singleton fuzzifier, Mamdani fuzzy inference engine and center average defuzzification .

Step4: The pixel in this step is reconstructed by using the median value when intensity gradient is high and the mean value when the intensity gradient is low.

The intensity gradients are calculated below:

$$G(1) = \text{abs}(W_4 - W_5).$$

$$G(2) = \text{abs}(W_2 - W_7).$$

$$G(3) = \text{abs}(W_3 - W_6).$$

$$G(4) = \text{abs}(W_1 - W_8).$$

The median value O_{median} is calculated using a 3×3 sliding window, while the mean value O_{mean} is calculated using the two outer pixels corresponding to the minimum gradient. The minimum gradient fuzzified using two fuzzy sets Low_Gradient (GL) and High_Gradient (GH) by equation (6).

Then the reconstructed pixel value is obtained by take the weights for both mean and median.

$$O_{\text{rec}} = GL(X) \times O_{\text{mean}} + GH(X) \times O_{\text{median}} \quad (7)$$

Now the enhanced pixel output is given by equation (8)

$$O_{\text{enhanced}} = \text{noisiness} \times W_0 + (1 - \text{noisiness}) \times O_{\text{rec}} \quad (9)$$

Step 5: Repeat steps 2-4 with new intensity sliding window until last image block.

4.4 Filter 4: Adaptive fuzzy filter

The noisy image can be enhancement by combined sharpening and smoothing techniques in the same processing image [13]. The former aims at increase the luminance difference between the center pixel and its neighborhood, while the latter aims at reducing the increase in noise. The developed enhancement technique algorithm is implemented as following:

Step1: The image enhancement technique operates on a window of size 3×3 as shown below.

$$\begin{bmatrix} W_1 & W_2 & W_3 \\ W_4 & W_{i,j} & W_5 \\ W_6 & W_7 & W_8 \end{bmatrix}$$

Let $W_{i,j}$ be the pixel luminance at location (i,j) , and let A denoted the set of $N=8$ neighboring pixels.

Step2: We compute the noise amplitude by considering fuzzy relations between the center pixel and its neighbors using the following relationship

$$\Delta W_{i,j} = \frac{k\alpha}{N} \left[\sum_{X_{m,n} \in A} \mu_{R_1}(W_{i,j}, W_{m,n}, \alpha) - \sum_{X_{m,n} \in A} \mu_{R_2}(W_{i,j}, W_{m,n}, \alpha) \right] \quad (10)$$

Where $k=1$ and R_v represent the class of fuzzy relations described by the parameterized membership functions: (10)

$$\mu_{R_v}(W_{i,j}, W_{m,n}, \alpha) = \begin{matrix} \text{MAX} \left[\frac{(1 - |W_{i,j} - W_{m,n} - \alpha|)}{2\alpha}, 0 \right] \\ \text{MAX} \left[\frac{(1 - |W_{i,j} - W_{m,n} + \alpha|)}{2\alpha}, 0 \right] \end{matrix}$$

Step3: Obtain the possible noise free value $Y_{i,j}$ of the pixel luminance at location (i,j) by subtracting the noise estimate $\Delta W_{i,j}$ from the original pixel luminance $W_{i,j}$:

$$Y_{i,j} = W_{i,j} - \Delta W_{i,j} \quad (11)$$

Step4: Repeat the operations from Step1 to Step3 for all image pixels. At the end of processing all image pixels in Step4 smoothing image was implemented.

Step5: Now the sharpening operation is performed on the result of smoothing operation by repeat Step1 to Step 4 using the same fuzzy equations as defined by (1-2) with $k=2$, but the output $\Delta W_{i,j}$, added to the original pixel luminance $W_{i,j}$

$$Y_{i,j} = W_{i,j} + \Delta W_{i,j} \quad (12)$$

4.4.1 The strategy to choose an optimum value for α

An appropriate choice of α in the smoothing case permits us to remove noise in the image, the larger the α , the more smoothing occurs (increase of detail blur), if α is small, the noise might not be removed.

In the case of sharpening, a large value of α causes sharper contrast among pixels, this decreases the smoothness of uniform areas.

We use fuzzy membership functions to adaptively obtain the

parameter α so that, the appropriate amount of smoothing and sharpening is applied to each pixel, this is done by tuning the α parameter according to a membership function which reflects the local noise pattern and can be defined by (13)

Where $\mu_{W_{i,j}}$ represents the degree of compatibility of a neighboring pixel $W_{m,n}$, the $\mu_{W_{m,n}}$ is a decreasing function of the scaled residual

$$\alpha_{sharp} = \min((\mu_c)(L-1), L-1)$$

$$\mu_{W_{m,n}} = \frac{(W_{i,j} - W_{m,n})^2}{\beta_{i,j}}$$

In which $\hat{\alpha}$ is the scale parameter and can be determined on the basis of the variations in pixel intensities in a given window. $\hat{\alpha}$ reflects the variance of luminance differences between the center pixel and its neighboring pixels as described by the function:

$$\beta_{W_{i,j}} = \frac{1}{N} \sum_{W_{m,n} \in A} (W_{i,j} - W_{m,n})^2 \quad (14)$$

To find whether a particular center pixel is an impulse noise pixel, we have to consider the compatibility of all the neighboring pixels by taking the mean of $\mu_{W_{i,j}}$ as described by the function:

$$\mu_c = \frac{1}{N} \sum \mu_{W_{i,j}} \quad 0 \leq \mu_c \leq 1$$

$$\mu_{W_{i,j}} = \exp\left(-\frac{(W_{i,j} - W_{m,n})^2}{\mu_{W_{i,j}} \beta_{i,j}}\right) \quad (15)$$

If the center pixel $W_{i,j}$ is an impulse noise pixel, the compatibility will be small, resulting in a small μ_c if the center pixel $W_{i,j}$ is part of a uniform area, μ_c would be large.

We will denote α_{smooth} for smoothing and α_{sharp} for sharpening, α_{smooth} should decrease as μ_c increases so as not to blur edges. α_{smooth} is given by the following relationship:

$$\alpha_{smooth} = \min\left(\exp\left(\frac{1}{\mu_c}\right), L-1\right)$$

α_{sharp} should be small in the presence of impulse noise and larger in areas which are relatively noiseless. The resulting membership function of α_{sharp} is an increasing function of μ_c .

Experimental Results

We use PSNR (Peak Signal to Noise Ratio) to evaluate the difference between pre and post processing images. PSNR is defined as follows:

$$MSE = \frac{1}{(MN)} \sum_{i=1}^M \sum_{j=1}^N (I_{i,j} - \bar{I})^2 \quad (18)$$

$$PSNR = 10 \log_{10} \frac{(255)^2}{MSE} \quad (19)$$

The variables M and N respectively denote the height and width of an image, I and \bar{I} stand for the versions of source and processing image. The larger the PSNR value is the more similar two images are.

To examine the effectiveness of the developed fuzzy filtering system, the fuzzy filters were implemented on four different standard test images: (Lenna,

Pepper, Car and Girl) contain a mixture of details, flat regions, shading and texture that do a good job of testing various image processing algorithms and having size 256×256 pixels. First of all, these images are corrupted by three different noise levels viz. 10%, 20%

and 30% salt and pepper to demonstrate the results of application of these methods. Figure 5.(a) and Figure 6.(a) show the Original images, while images corrupted by salt and Pepper noise with corruption levels of 20% and 30% are given in Figure 5.(b) and Figure 6.(b)

respectively. The results for Lenna and Pepper images after applying different filters are shown in Figure 5.(c-h), Figure 6.(c-h) respectively.

The resulting PSNR for various filters is given in Table 2 for Lenna image, in Table 3 for Pepper image, in Table 4 for the Girl image and in Table 5 for Car image.

Table 2: Performance of various filters on the Lenna image

	PSNR		
	10%	20%	30%
Noise Image	25.5032	22.3372	20.6123
Mean	25.9907	24.0996	23.9594
Median	28.9515	27.8850	26.8310
Filter1	31.4019	30.6188	29.0119
Filter2	30.5885	30.5498	29.8672
Filter3	30.4359	30.0937	29.0123
Filter4	28.0359	27.7984	26.1083

Table 3: Performance of various filters on the Pepper image

	PSNR		
	10%	20%	30%
Noise Image	24.9012	21.0025	19.4462
Mean	26.4040	26.0651	25.7399
Median	28.6182	28.5822	27.5664
Filter1	34.1832	32.9987	32.2302
Filter2	30.9851	29.7432	28.9841
Filter3	32.2188	31.6695	31.6688
Filter4	28.3974	28.6850	27.6182

Table 4: Performance of various filters on the Girl image

	PSNR		
	10%	20%	30%
Noise Image	25.2615	21.5277	20.3160
Mean	28.2865	27.8169	27.3554
Median	30.0582	29.9777	29.9252
Filter1	34.1919	33.0086	32.4882
Filter2	31.7907	30.2275	29.9930
Filter3	31.4949	31.0812	30.8035
Filter4	29.5931	28.7521	28.1096

Table 5: Performance of various filters on the Car image

	PSNR		
	10%	20%	30%
Noise Image	25.5795	22.5275	20.6496
Mean	25.9054	25.6175	25.3588
Median	28.4944	27.3676	26.7759
Filter1	33.3482	32.0271	31.4547
Filter2	30.9630	30.3385	30.0261
Filter3	31.6511	30.8577	30.5248
Filter4	28.0305	27.1039	26.4739

Conclusion

In this paper, the combination of fuzzy logic techniques and simple filters is achieved for impulse noise reduction based on the detailed information on the difference between pixels. Fuzzy rule based enhancement is an attractive solution to removing noise while preserving the integrity of edges as much as possible.

An effective method combining smoothing and sharpening in the presence of impulse noise also employed for construct enhancement to reduce image noise. In the proposed method, the different parameter values are adaptively obtained according to fuzzy membership function.

The performance is examined under salt and pepper noise model. The final images from different fuzzy filters are compared with those of other statistical filters using real images such as Lenna, Peppers, Car and Girl. In the experiments, it has been observed that the results of the proposed fuzzy filters are showed better performance and are an improvement over the classical arithmetic filters mean and median on the PSNR. A compatibility measure PSNR is calculated at each pixel which represents the noise content in it, PSNR shows the ability to remove impulse noise while preserving edge sharpness and the results are shown in tables (2, 3, 4, 5).

References

- [1].Chin-chen Chang, J.Hsiao, and C.Hsieh, December (2003). "A Fast Noise Reduction method Based on Human Visual System". ICICS-PCM 2003, singapore.

- [2].J.Astola and P.Kuosmanen, (1997). "Fundamentals of non-linear Digital Filtering". CRC, Boca Raton, Fl.
- [3].T.C.Chen, K.K Ma and L.H che, Dec.(1999). "Tri-state median filter for image denosing". IEEE Transactions on Image Processing, Vol. 8: No.12, pp.1834-1838.
- [4].Ho-Ming Lin, A.N.Willson, June (1988). "Median Filters with Adaptive Length". IEEE Transaction on circuits and system, Vol. 35: No. 6.
- [5].X.Wang, (1992). "Generalized Multi stage Median Filters". IEEE Trans. On image proc. Vol.4, pp.543-545.
- [6].E.Abreu, M.lightstone, S.K.Mitra, and K.Arakawa, June (1996). "A new efficient approach for the removal of input noise from highly corrupted images". IEEE Transaction Image processing, Vol. 5, pp.1012-1025.
- [7].H.L.Eng and K.K.Ma, (2001). "Noise adaptive soft-switching median filter". IEEE Trans. On Image processing, Vol. 10, pp. 242-251.
- [8].L.Wang, (1997). "A course in fuzzy systems and control". USA, prentice Hall.
- [9].Hamid R. Tizhoosh, (1998). "Fuzzy Image Processing: Potential and state of the ART". 5th International Conference on Soft Computing, Japan, October 16-20, Vol 1, pp 321-324.
- [10].Tinka Acharya and A.K. Ray, (2005). "Image Processing principles and applications". Wiley-Interscience puplicaton, Canada.
- [11].Hamid R. Tizhoosh, (2004). "Fuzzy Image Processing". Springer-Verlag.
- [12].How-Lung, Eng and Kai-Kuang Ma, (2000). "Noise Adaptive Soft-Switching Median Filter for Image Denoising". IEEE Intl. Conference on Image processing, Japan, pp 2175-2178.
- [13].F.Russo, Aug.(2002). "An Image Enhancement Technique Combining Sharpening and Noise Reduction". IEEE Trans. Instrum. Meas., Vol. 51, pp 824-828.



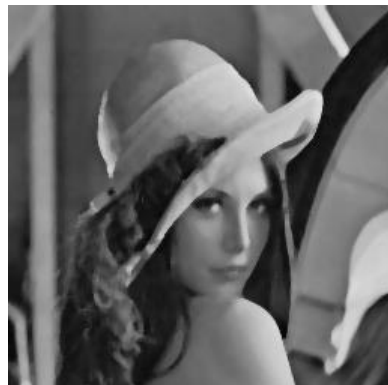
(a)



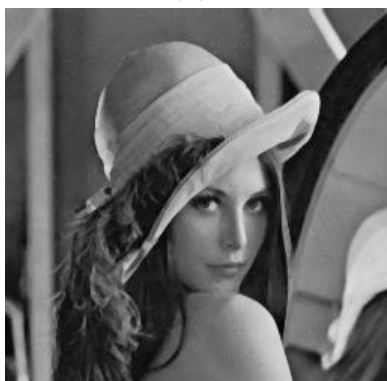
(b)



(c)



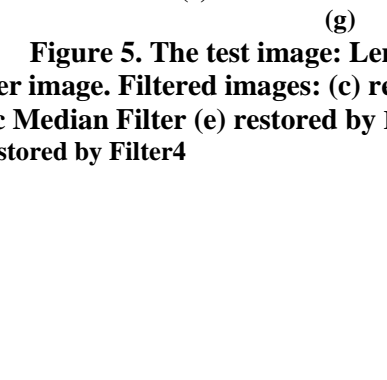
(d)



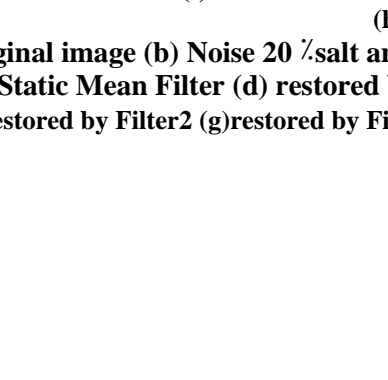
(e)



(f)



(g)



(h)

Figure 5. The test image: Lena, (a) Original image (b) Noise 20 %salt and pepper image. Filtered images: (c) restored by Static Mean Filter (d) restored by Static Median Filter (e) restored by Filter1 (f) restored by Filter2 (g) restored by Filter3 (h) restored by Filter4

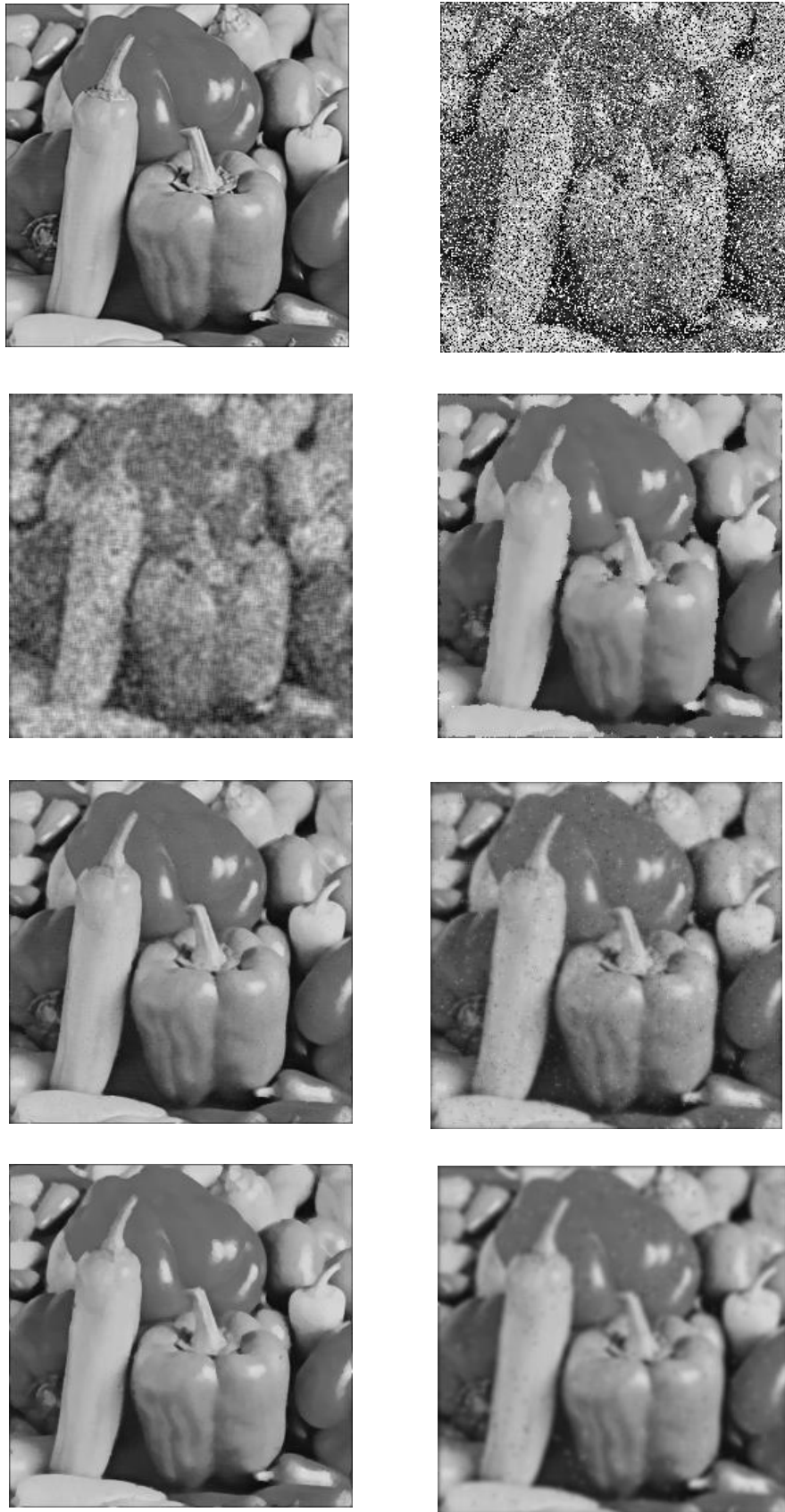


Figure 6. The test image: Peppers, (a) Original image (b) Noise 30 %salt and pepper image. Filtered images: (c) restored by Static Mean Filter (d) restored by Static Median Filter (e) restored by Filter1 (f) restored by Filter2 (g) restored by Filter3 (h) restored by Filter4.

أزالة ضوضاء الصورة باستخدام تقنيات المنطق المضبيب

د.ندى نعمت سليم سحر خالد احمد ورقاء يونس ابراهيم

E.mail:scicolanb@yahoo.com

الخلاصة:

في هذا البحث تم اقتراح خوارزميات تستخدم أو تطبق المنطق المضبيب على المرشح الوسطي والمرشح المعدلي لغرض تحسين طريقة إزالة الضوضاء في معالجة الصور. وهذه الخوارزميات حسنت من كفاءة ونوعية الاحتفاظ بتفاصيل الصورة مقارنة بالمرشحات التقليدية المتبعة لإزالة الضوضاء من الصور التي تعرضت إلى ضوضاء شديدة.

مرشحات الصور المضببية المقترحة تعتمد على إيجاد وتحديد نسبة الضوضاء أولاً باستخدام المنطق المضبيب ثم استرجاع قيمة النقطة التي تعرضت للضوضاء. ولتحديد الضوضاء نحتاج إلى قاعدة معلومات مضببية والتي تشمل تحديد القوانين المضببية وتحديد البيانات المضببية الداخلة والخارجة للقاعدة المضببية.

وفي هذا البحث تم أيضاً تحسين الصور المعرضة للضوضاء وذلك بدمج طريقة التنعيم وطريقة تحديد تفاصيل الصورة مع المنطق المضبيب، كما إن هذه الطريقة تغير من قيم المعاملات المستخدمة بالاعتماد على الصفات والمعلومات المتوفرة آنياً للحصول على القيمة المثالية.

تم حساب النتائج لعدة أنواع من الصور التي تعرضت لمستويات (كميات) مختلفة من الضوضاء. وتم مقارنة النتائج التي توصلنا إليها باستخدام الطرق المقترحة لإزالة الضوضاء مع المنطق المضبيب والطرق الرياضية التقليدية وهي المرشح الوسطي والمرشح المعدلي لإزالة الضوضاء لبيان كفاءة الطرق المقترحة.