Reduction of Impulsive Noise in Continuous-Tone Images by Regression Analysis

Md. Tanvir Al Amin

Abstract— ***Obtaining a clear and lucid image by reducing the noise to a minimal level is one of the most fundamental research topics in image processing. Necessity for reducing noise is not only for aesthetic purposes, but also due to its vital role in the success of image processing and image understanding algorithms. As there is no way for total elimination, several methods have been developed, depending on the type of noise. In this paper, we devise a method for reducing impulsive noise and study its noise detection and image restoration performance. A sweeping window of certain dimension calculates how well a plane can be fitted over the pixels currently inside the window and then examines each pixel according to some rules whether it is part of noise or part of signal. Final decision about each pixel is taken by counting the majority verdicts about it. Values of the corrupted pixels are found by fitting a general paraboloid equation only through the uncorrupted pixels in each window and taking the mean of all suggested values. The method can be applied to a vast area of real world applications like digital photography, medical imaging, computer vision and so on. Simulation reveals that on average, our method has success rate of nearly 97% in case of low noise density and nearly 93% in case of medium noise density.***

Index Terms— **Edge Pixel, Impulsive noise, Continuous Tone Image, Noise Density.**

I. Introduction

Noise Reduction in digital images is an interesting topic for researchers. As some sort of error is always associated with signal handling devices, restoring the real signal from noise becomes a challenging task.

Due to several reasons e.g. Inappropriate Acquisition, Analog to digital conversion, Quantization, exposure setting, Camera ISO speed, scanning problem, dust in lens, film grain etc. random noise can be initiated in digital images leading image quality degradation [1],[3].

The process of minimizing the noise and thus presenting a lucid image is a necessary objective of myriad research fields related to Remote image acquisition, Satellite Imagery, Digital Photography, Image Processing and Understanding, Medical Imaging, Computer Vision [1].

Md. Tanvir Al Amin, Author is a student of Department of Computer Science and Engineering (#0305012), Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh. E-mail : tanviralamin@gmail.com

Mathematically, the goal is to produce the best possible estimate $\hat{a}[m,n]$ of the original image $a[m,n]$ from a recorded image $c[m,n]$. The measure of success is usually an error measure between the original $a[m,n]$ and the estimate $\hat{a}[m,n]$ as no mathematical error function is known that corresponds to human perceptual assessment of error [1].

If several images of same object differing only in noise realization are available then temporal averaging reduces the noise [1]. When temporal averaging is not possible, methods developed from linear filtering or smoothing is sometimes used.

In natural images, to distinguish the distorting pixel from the real one is an ill posed problem since it is not always known or well defined which measures actually answers whether a pixel is corrupted or not [5]. Most of the solutions on this problem work under certain assumptions, conditions and using some priori knowledge about the noise involving phenomenon. [2].

Early techniques involved general statistical ideas or frequency domain concept [1],[2]. Lee [3] describes methods for contrast enhancement and white noise filtering of a digital image based on local statistics.

Frequency based noise reduction is more or less established on the fact that, a small window in a continuous tone image contains less high frequency components. So, controlling high frequency components results in reduction of noise.

But these blind methods don't take the position of the error into account. So directly applying these methods often leads to undesired decrease in image sharpness. Applying a low or mid pass filter often makes the image blurred at edges and causes loss of information. Young *et al.* [1] compares several of these techniques.

A classical solution in this regard is Wiener filter [1] which works with apriori noise and power spectra. But, as the noise and power spectra are unknown in most of the cases, several extensions or alternatives exist considering fully or partly frequency based statistics and wavelet decomposed methods [6]. Research works also included Fractal based methods or combining Fractal, wavelet and frequency based methods [5],[12].

Modern techniques on this topic use algebraic manipulations. Introductory concepts of image restoration based on algebraic methods, Zero order approximation, Bilinear Interpolation for Gray Level Threshold, constrained and unconstrained Least squares are discussed in [2].

In most of the cases the measurement of error is done by Least Squares method. Though Rudin *et al* [8] emphasizes on using L_2 norm, gain in performance is not that high compared to the extra overhead needed. To avoid massive amount of calculations we use L_1 norm for the Least squares criteria in the paper.

Moreover, Eng and Ma [4] present a scheme for impulse noise detection and denoising. Their system decides the corrupted pixels first by applying median filter of certain window size determined by the noise density. By applying fuzzy set rules, the system discriminates between edge pixel and non-isolated impulse noise, and then denoises the image using median filter or FWM filtering.

However, these methods suffer from a common problem of extensive computation or iterative denoising, which is not feasible many cases, especially when handheld devices are involved and real time processing is necessary. Hence, existing methods are not easily employable in small handheld devices like mobile or handicam, with low power operation facility.

Depending on several parameters, image noise can be classified in two groups - dependent (Gaussian) or independent (random). Our concern here is one kind of random noise, also known as salt and peeper noise [7], which appears on the image as additive random impulsive dots or small regions. We introduce the method in [9], to determine the impulsive noise using window based local statistics and regression analysis. However in this paper we perceive the method thoroughly, evaluate both noise detection and removal parts by simulation and explore further observations.

Our basic assumptions here are:

1) Noise is distributed throughout the whole image. There is a pre-assumed noise density, which tells about how dense the corrupted pixels are.

2) Noisy pixels vary more than a threshold value with most of the surrounding pixels.

To decide whether a pixel is corrupted or not, our method uses up to s^2 judgments for a single pixel. Each judgment is done by fitting a general plane on a square window of dimension $s \times s$, which outputs its verdicts about each of the $s \times s$ pixels inside.

Generally it is difficult to discriminate between and edge and an impulse. Strength of our method is, chance to detect an edge pixel as impulsive noise is very low, because final decision about a pixel is made by combining the output of up to s^2 judgments. To get the estimated value for the corrupted pixel we have approximated a paraboloid i.e a bivariable second degree equation per window taking only good pixels in consideration, because this partly supports the gradual rise and fall nature, which is perfectly suited for natural images having continuous tone.

II. The Method

Our proposed method works basically in two stages.

Stage 1 : Detect the pixels which are corrupted.

Stage 2 : Keep the uncorrupted pixels intact. Estimate values for the corrupted pixels from its neighboring good pixels.

Throughout the discussion, let W be an image of dimension $m \times n$ corrupted by impulsive noise of known average strength and density. Let the value of a pixel at (u, v) position of the image = $W(u,v)$

A. Noise Detection

For each position *(x,y)*, $1 \le x \le m - s + 1 \quad 1 \le y \le n - s + 1$ in the image, our method processes a square window of dimension $s \times s$ stretching from (x, y) to $(x+s-1, y+s-1)$, where s is provided by user. s mainly depends on noise density.

1) Plane Approximation for Windows: If (x, y) is the starting point of a window $W_{x,y}$ and $(i, j) \quad 0 \le i, j \le s-1$ is the offset of a pixel, then $(i+x, j+y)$ is the global position of that pixel. For each window, the method first fits a plane with the pixel-values in the window by Multiple Linear Regression [10][11]

TABLE I
Suggestion For Window Dimension s

Noise Density	$s \times s$
$0 < \rho \le 0.3$	4×4
$0.3 < \rho \le 0.6$	6×6
$0.6 < \rho \le 0.8$	8×8

A general plane inside the window starting at *(x,y)* is: $Z_{x,y}(i, j) = c_1 + c_2 i + c_3 j$. For a Least squares fit, the linear system formed is

$$A \times C_{x,y} = B_{x,y} . \tag{1}$$

Several methods are described in [10],[11] to solve the system.

$$\text{where, } A = \sum_{i=0}^{s-1} \sum_{j=0}^{s-1} \begin{bmatrix} 1 & i & j \\ i & i^2 & ij \\ j & ij & j^2 \end{bmatrix} \tag{2}$$

$$B_{x,y} = \sum_{i=0}^{s-1} \sum_{j=0}^{s-1} \begin{bmatrix} W(i+x, j+y) \\ i.W(i+x, j+y) \\ j.W(i+x, j+y) \end{bmatrix} \tag{3}$$

$$\text{and } C_{x,y} = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}^T \tag{4}$$

Matrix A depends only on window size. For s = 4, calculated value of A is :

$$A = \begin{bmatrix} 16 & 24 & 24 \\ 24 & 56 & 36 \\ 24 & 36 & 56 \end{bmatrix} \tag{5}$$

For s = 6,

$$A = \begin{bmatrix} 36 & 90 & 90 \\ 90 & 330 & 225 \\ 90 & 225 & 330 \end{bmatrix} \tag{6}$$

For s = 8,

$$A = \begin{bmatrix} 64 & 224 & 224 \\ 224 & 1120 & 784 \\ 224 & 784 & 1120 \end{bmatrix} \tag{7}$$

2) Local Classification: From the plane determined for the window starting at *(x,y),* tentative value of $Z_{x,y}(i,j)$ $0 \le i, j \le s-1$ is calculated. Plane Pixel Deviation for each offset *(i,j)*

$$PPD_{x,y}(i,j) = \left| Z_{x,y}(i,j) - W(i+x, j+y) \right| \quad (8)$$

To find the most of the corrupted pixels our method depends on two external input parameters.

Deviation Parameter δ = Maximum Plane Pixel deviation that let the window render it uncorrupted.

Density Parameter ρ = Average density of corrupted pixels. It is assumed that corrupted pixels are distributed everywhere in the image $\rho \in [0,1]$. Experiment shows, taking a slightly larger value for Density parameter, than the assumed Noise Density yields better result.

If $PPD_{x,y}(i,j) > \delta$ current window reports the pixel at offset (i,j) as corrupted. If total number of reported corrupted pixels in a window exceeds $s^2\rho$, judgment of that window is not accepted.

For example, if s=4, suppose a window:

$$W_{u,v} = \begin{bmatrix} 40 & 52 & 55 & 58 \\ 60 & 62 & 90 & 60 \\ 5 & 70 & 60 & 58 \\ 55 & 61 & 64 & 25 \end{bmatrix} \quad (9)$$

$C_{u,v} = [52.14 \quad -1.98 \quad 3.68]$ calculated from $W_{u,v}$.

$$Z_{u,v} = \begin{bmatrix} 52.14 & 55.81 & 59.49 & 63.16 \\ 50.16 & 53.84 & 57.51 & 61.19 \\ 48.19 & 51.86 & 55.54 & 59.21 \\ 46.21 & 49.89 & 53.56 & 57.24 \end{bmatrix} \quad (10)$$

$$PPD_{u,v} = \begin{bmatrix} 12.14 & 3.81 & 4.49 & 5.16 \\ 9.84 & 8.16 & 32.49 & 1.19 \\ 43.19 & 18.14 & 4.46 & 1.21 \\ 8.79 & 11.11 & 10.44 & 32.24 \end{bmatrix} \quad (11)$$

Taking δ =25, pixel at offset (1,2), (2,0) and (3,3) are reported corrupt by the current window as PPD there exceed 25.

3) Global Classification by majority vote: Final decision about a pixel is taken according to majority vote i.e. verdict of most of the windows about it.

After processing the windows is complete, for each pixel *(u,v)* we have,

J(u,v) : number of accepted judgments for pixel at (u,v)

R(u,v): number of verdicts reporting pixel at (u,v)

uncorrupted $\alpha(u,v) = \dfrac{R(u,v)}{J(u,v)}$, when $J(u,v) \ne 0$ (12)

Verdict of a window is not accepted when noise density reported there exceeds the assumed density parameter ρ. This may happen in the region where the image has high contrast grainy parts or near some sharp edge of the object as there is an abrupt change in pixel value. So the plane approximation may misclassify some edge or object boundary pixels and noise presented there, especially in case of high noise.

To discriminate between edge and noise we introduce: Classifier Parameter, $\Omega \in (0,1)$. Pixel at (u,v) is assumed belonging to Edge part or boundary region of some object in the image if $J(u,v) < \Omega s^2$, i.e. in case of the pixels near edges, a certain fraction of the windows yield unreliable answer. Value of Ω depends on sharpness of the image. $\Omega = 0.4 \sim 0.5$ for natural images.

Threshold Ratio, φ = Minimum $\alpha(u,v)$ needed for a pixel to be declared uncorrupted globally. $\varphi \in (0,1)$. We define two Threshold ratios,

φ_e : used for edge pixel or grainy part

φ_n : used for flat or non grainy regions

If $s = 4$ value of $\varphi_e = 0.6 \sim 0.75$ and $\varphi_n = 0.75 \sim 0.9$.

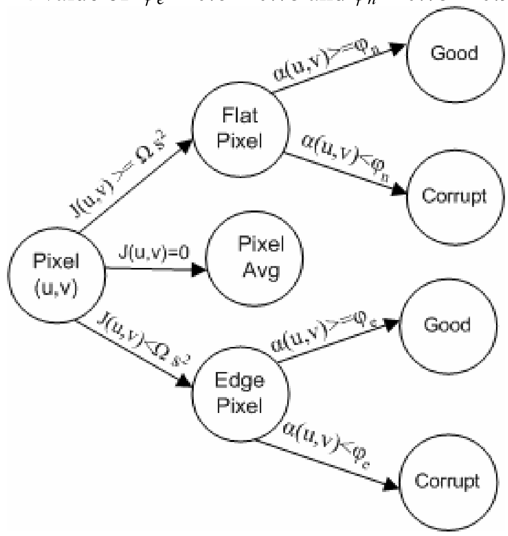


Fig. 1. Decision Tree to decide about a pixel

When $J(u,v) = 0$ the pixel is neither corrupted nor good. But in the next phase, during filtering, it is applied an average of the neighboring values. Fig. 1 summarizes the whole process of classification.

B. Noise Filtering

The values of the corrupted pixels are estimated from the values of nearby uncorrupted pixels. As we are dealing with impulsive noise, which change the pixel value too much, the value of a noisy pixel is not taken into account. Let, $\hat{W}(u,v)$ be the estimated value for corrupted pixel *(u,v).*

1)Paraboloid Approximation: For each window of dimension $s \times s$ at *(x,y)* the method fits a general paraboloid $K_{x,y}$ through the uncorrupted pixels inside that window.

$$K_{x,y}(i,j) = c_1 + c_2 i + c_3 j + c_4 i^2 + c_5 j^2 + c_6 ij \quad (13)$$

$$C_{x,y} = [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6]^T, \quad (14)$$

$$A \times C_{x,y} = B_{x,y} \quad (15)$$

is solved for *C.*

$$B_{x,y} = \sum_{i=0}^{s-1}\sum_{j=0}^{s-1}\left(\begin{bmatrix} w(i+x,j+y) \\ i.w(i+x,j+y) \\ j.w(i+x,j+y) \\ i^2.w(i+x,j+y) \\ j^2.w(i+x,j+y) \\ i.j.w(i+x,j+y) \end{bmatrix}, \ w(i+x,j+y) \text{ is uncorrupted}\right) \quad (16)$$

$$A = \sum_{i=0}^{s-1}\sum_{j=0}^{s-1}\left(\begin{bmatrix} 1 & i & j & i^2 & j^2 & ij \\ i & i^2 & ij & i^3 & ij^2 & i^2j \\ j & ij & j^2 & i^2j & j^3 & ij^2 \\ i^2 & i^3 & i^2j & i^4 & i^2j^2 & i^3j \\ j^2 & ij^2 & j^3 & i^2j^2 & j^4 & ij^3 \\ ij & i^2j & ij^2 & i^3j & ij^3 & i^2j^2 \end{bmatrix}, w(i+x,j+y) \text{ is uncorrupted}\right) \quad (17)$$

Due to absence of information in corrupted pixels A may be singular. In that case a paraboloid approximation is not possible and a Plane approximation

$$K_{x,y}(i,j) = c_1 + c_2 i + c_3 j \quad (18)$$

is taken.

2) Local Suggestion: From the regression coefficients, the window at *(x,y)* suggests for each corrupted pixel

$$\hat{w}(i+x, j+y) = K_{x,y}(i,j) \quad (19)$$

3) Global Estimation: For each corrupted pixel *(u,v)* we have, *S(u,v)* : sum of suggested values for *(u,v)*

T(u,v) : number of suggestions for *(u,v)*

Final Estimation, $\hat{W}(u,v) = \dfrac{S(u,v)}{T(u,v)}$, when $T(u,v) \neq 0$ (20)

If $T(u,v) = 0$ or $J(u,v) = 0$, value of that pixel is determined by averaging the neighboring pixels, after final estimation of other corrupted pixel is complete.

III. Simulation

Performance of the method can be calculated only by simulation when the error is known. Simulation is done by Matlab with standard test images artificially corrupted by Noise Strength approximately 50. When noise density is low or medium result seems better for window dimension 4×4, and discrimination between edge pixels and noisy pixels is also better.

A. Simulation of Noise Detection

Effect of several parameters on performance of noise detection was examined.

We define Performance of the Noise Detection part by two functions:

Classification Efficiency (CE) ,

$$= \frac{\text{Number of Pixels classified correctly}}{\text{Total number of Pixels in the image}} \times 100\% \quad (21)$$

Error Detection Efficiency (EDE),

$$= \frac{\text{Number of corrupted pixels correctly detected}}{\text{Total number of corrupted pixels}} \times 100\% \quad (22)$$

1) Performance at various Noise Density: Change in behavior of the method at various noise density is one of the most crucial parts related to performance. We consider our method for various levels of corrupted images. Noise Detection efficiency for different images are not same but they have similar trends.Here we graphically plot the results for 3 standard test images, 'lenna', 'peppers' and 'airplane'. Parameters are taken : δ=30, φ_e = 0.7 and φ_n = 0.85, Ω = 0.5

For 'lenna', Classification Efficiency and Error Detection Efficiency is shown in Fig. 2, and Fig. 3 respectively. CE is 98% at noise level 6%. It decreases as noise increases. At noise level 55% it becomes 77%.

Noise Detection is most efficient (88%) at noise level 10% to 40%. But the efficiency doesn't fall that much as noise in increased. At noise level 55% Detection efficiency is 83%.

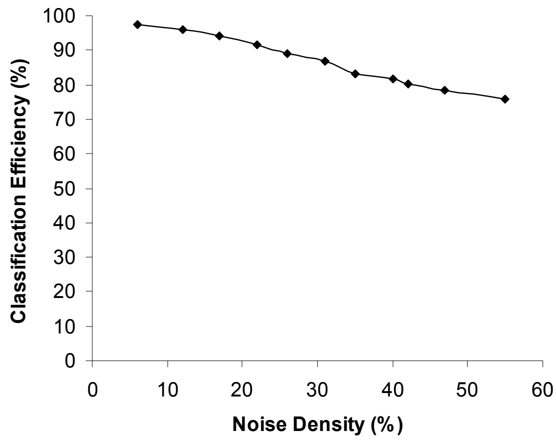


Fig. 2. Effect of Noise Density in Classification Efficiency

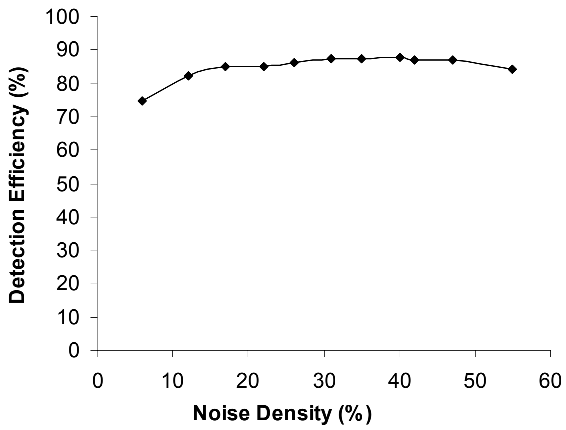


Fig. 3. Effect of Noise Density in Detection Efficiency

CE and EDE for ‘peppers’ are shown in Fig.4 and Fig. 5. CE is about 99% for ‘peppers’ at 6% error and reduces to 91% as density of corrupted pixel grows toward 40%

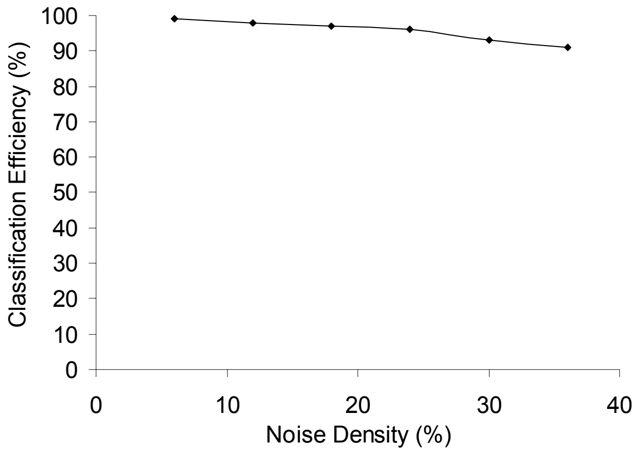


Fig. 4. Effect of Noise Density in Classification Efficiency

Behavior of EDE for ‘lenna’ and ‘peppers’ are similar, a maximum value is obtained near 20% to 30% Noise Density.

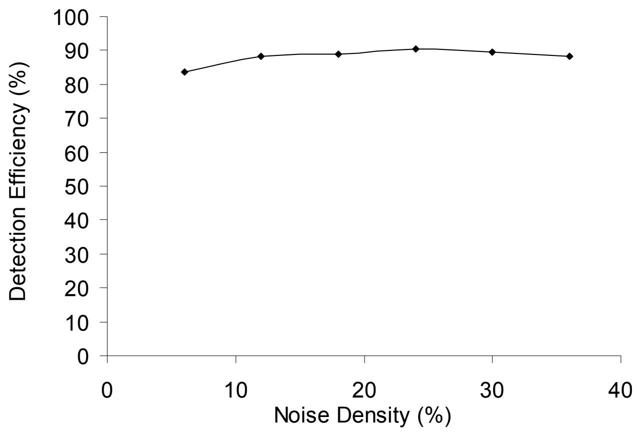


Fig. 5. Effect of Noise Density in Detection Efficiency

CE and EDE for ‘airplane’ is in Fig. 6 and in Fig. 7. CE is 97% at 6% error and is 85% at 35% error. Behavior of EDE for the range of error <40% for ‘airplane’ is different from ‘peppers’ and ‘lenna’.

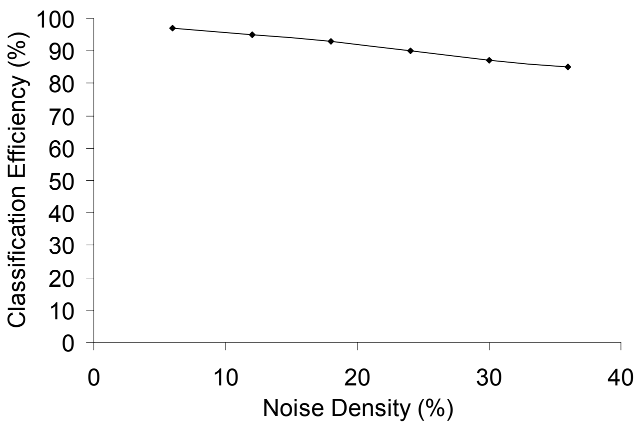


Fig. 6. Effect of Noise Density in Classification Efficiency

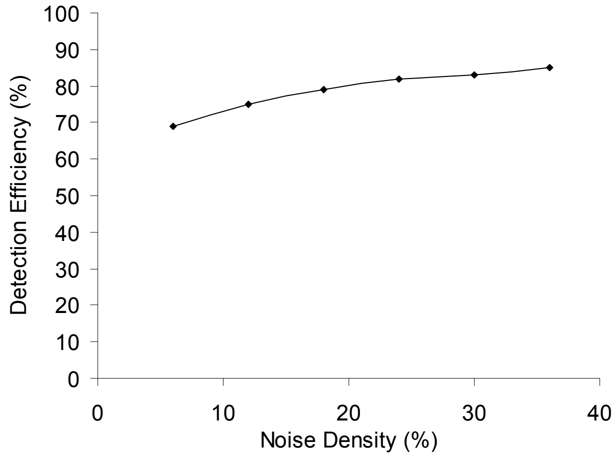


Fig. 7. Effect of Noise Density in Detection Efficiency

2) Effect of Deviation Parameter: The effect of deviation parameter for noise density 26% is shown in Fig. 8 and Fig. 9. The test image used is ‘lenna’

Parameters taken :$\varphi_e = 0.7$ and $\varphi_n = 0.85$, $\rho = 0.34$, $\Omega = 0.5$, $s = 4$

Highest performance is found at δ=30, Detection Efficiency about 87% and Classification Efficiency about 91%.

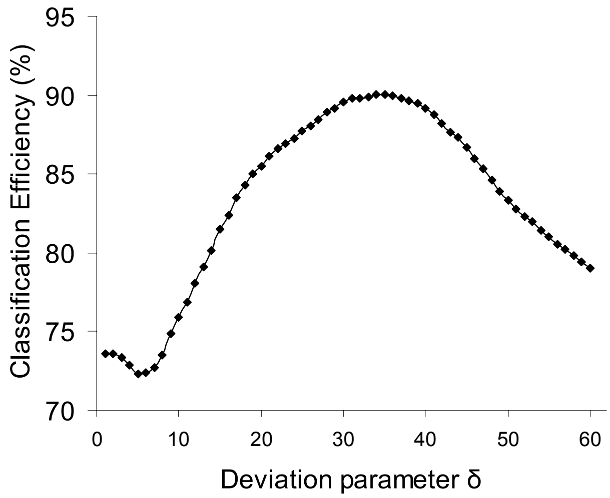


Fig. 8. Effect of Parameter δ in Classification Efficiency

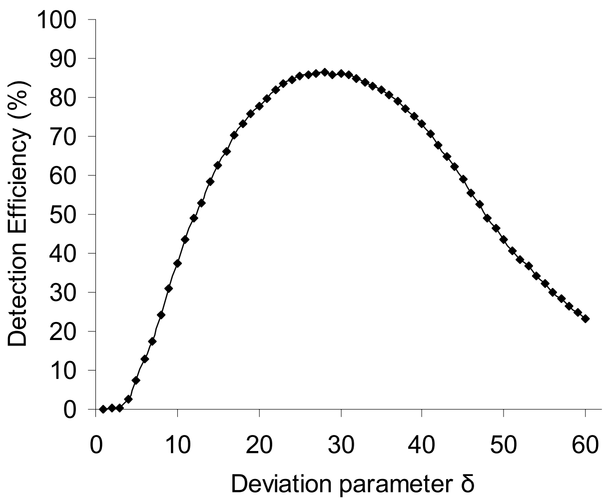


Fig. 9. Effect of δ in Detection Efficiency

3) Effect of Density Parameter for fixed Noise Density: For noise density 30% optimal value of ρ is 0.4 as depicted in Fig. 10 for test image ‘lenna’

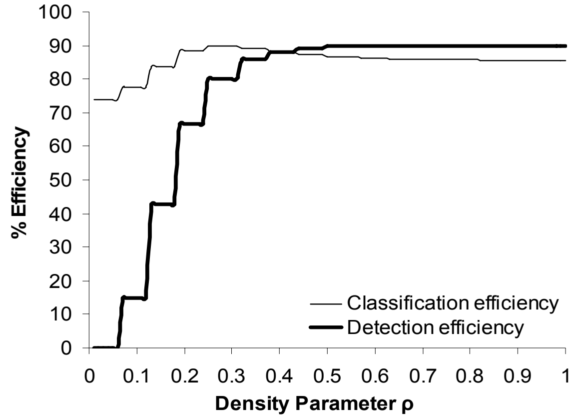


Fig. 10. Effect of ρ in Efficiency for Noise Density 30%

4) Effect of Threshold Ratio: With decreasing Threshold Ratio, Error Detection Efficiency decreases and Classification Efficiency increases for test image ‘lenna’ (Fig. 11) Reason is, as we try to include all the noise, some of the uncorrupted pixels are also treated as noise which decreases Classification Efficiency, and vice versa. So there is a tradeoff between these two.

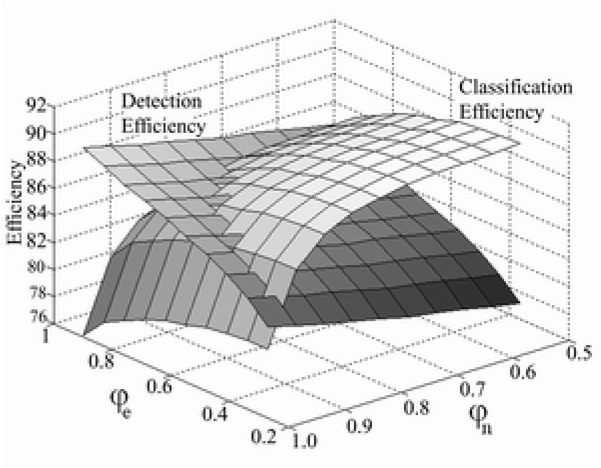


Fig. 11. Effect of φ in Efficiency

For ρ = 0.4, Ω = 0.5, s=4, Noise Density = 30%, optimal value of φ_e = 0.7 and φ_n = 0.85.

B. Noise Filtering

Evaluating Noise Filtering performance mathematically is possible only in simulation when the we actually know the corrupted pixels. Here we use Peak Signal to Noise Ratio (PSNR) as the objective.

If I denotes original image and K denotes restored one. MAX_I = 255, maximum pixel value

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|I(i,j)-K(i,j)\right|^2 \quad (23)$$

then, $PSNR = 20\log_{10}\frac{MAX_I}{\sqrt{MSE}}$ (24)

Noise Filtering i.e. estimation of the values of corrupted pixels is best when noise density is low to medium. Here we measure filtering performance by Peak Signal to Noise Ratio (PSNR) and also show the input output images visually.

Here PSNR curves and input output images are shown for 3 standard test images, ‘lenna’, ‘peppers’ and ‘airplane’.

1) lenna: PSNR for ‘lenna’ is shown in Fig. 12, and input output images are Fig. 13(a)(b), Fig. 14(a)(b), Fig. 15(a)(b) and Fig. 16(a)(b) for 6%, 12%, 24% and 36% noise density respectively.

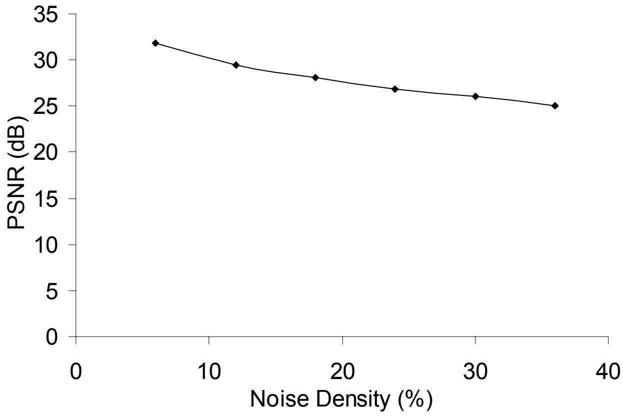


Fig. 12. PSNR vs. Noise Density (‘lenna’)

(a) (b)

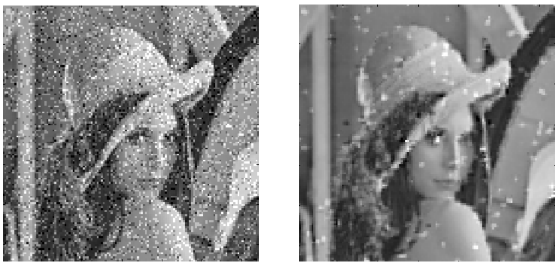
Fig. 13. (a) 6% corrupted image (b) Output at 6% noise

(a) (b)

Fig. 14. (a) 12% corrupted image (b) Output at 12% noise

(a) (b)

Fig. 15. (a) 24% corrupted image (b) Output at 24% noise



(a) (b)

Fig. 16. (a) 36% corrupted image (b) Output at 36% noise

2)Peppers: PSNR for 'peppers' is shown in Fig. 17, and input output images are Fig. 18(a)(b), Fig. 19(a)(b), Fig. 20(a)(b), Fig. 21(a)(b) for 6%, 12%, 24% and 36% noise density respectively.

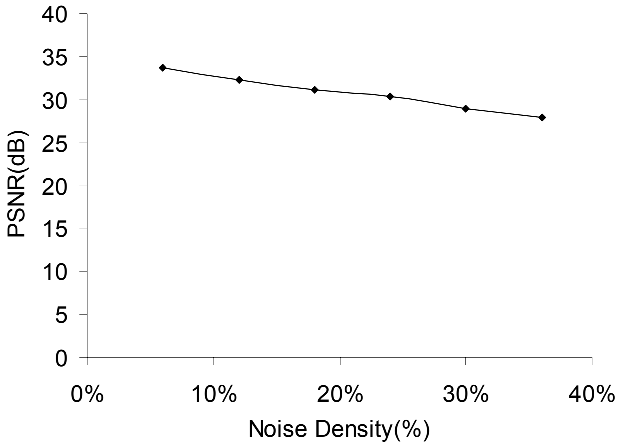


Fig. 17. Peak Signal to Noise Ratio vs. Noise Density ('peppers')

For 'peppers', falling tendency in PSNR curve is less than 'lenna'. This follows from the observation that, 'peppers' have more continuous tone parts than 'lenna'.

(a) (b)

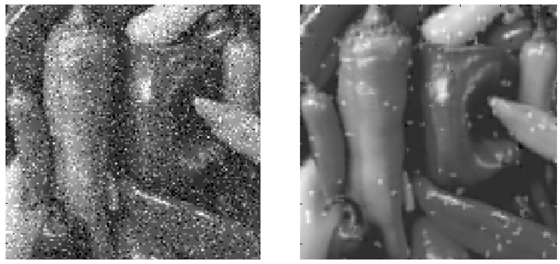
Fig. 18. (a) 6% corrupted image (b) Output at 6% noise

(a) (b)

Fig. 19. (a) 12% corrupted image (b) Output at 12% noise

(a) (b)

Fig. 20. (a) 24% corrupted image (b) Output at 24% noise



(a) (b)

Fig. 21. (a) 36% corrupted image (b) Output at 36% noise

3)airplane: PSNR for 'airplane' is shown in Fig. 22, and input output images are Fig. 23(a)(b), Fig. 24(a)(b), Fig. 25(a)(b), Fig. 26(a)(b) for 6%, 12%, 24% and 36% noise density respectively.

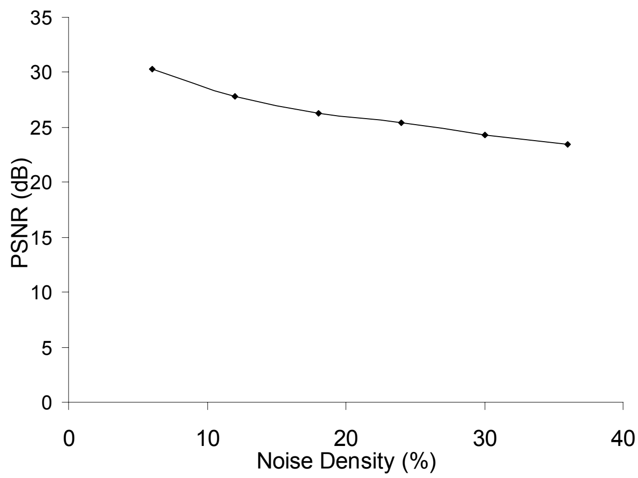
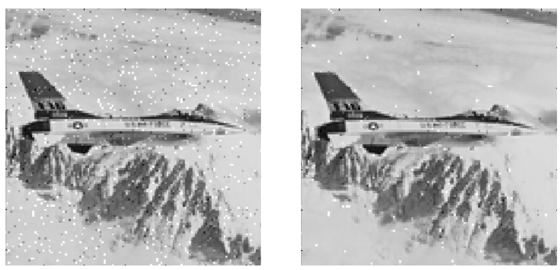
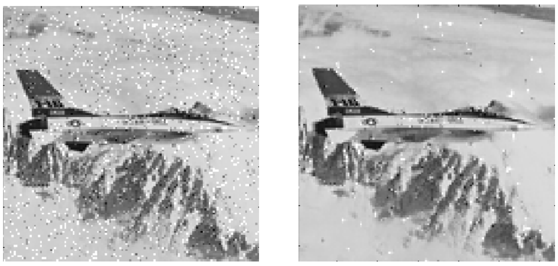


Fig. 22. Peak Signal to Noise Ratio vs. Noise Density ('airplane')



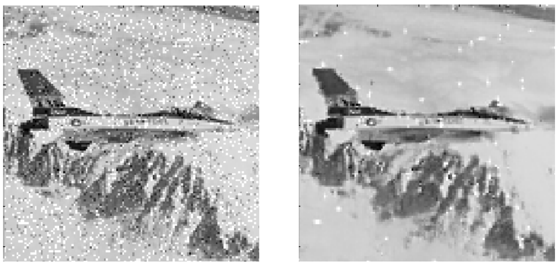
(a) (b)

Fig. 23. (a) 6% corrupted image (b) Output at 6% noise



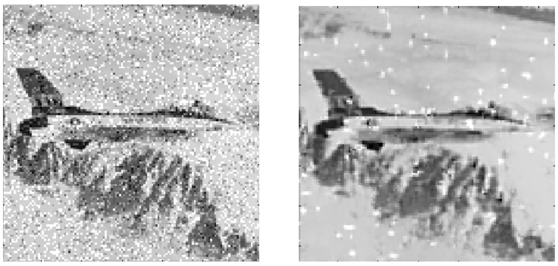
(a) (b)

Fig. 24. (a) 12% corrupted image (b) Output at 12% noise



(a) (b)

Fig. 25. (a) 24% corrupted image (b) Output at 24% noise



(a) (b)

Fig. 26. (a) 36% corrupted image (b) Output at 36% noise

IV. Complexity Analysis

Number of windows $= (m - s + 1)(n - s + 1)$

Cost per window for Local classification: $O(s^2)$

Time for Global Error Classification : $O(mn)$

Filtering can be done in $O(\rho s^2)$ per window

For Final Estimation $O(mn)$

So, Total Cost :

$O((m\text{-}s+1)(n\text{-}s+1)s^2 + mn + \rho s^2(m\text{-}s+1)(n\text{-}s+1)+mn)$

$= O(mns^2(1+\rho))$

V. Outcome Of The Research

The method performs best in noise density < 40%, which is a good limit for natural images.

Main successes of the method are:

1) It doesn't use blind mean or median filtering here, so the output doesn't suffer from unwanted loss in sharpness.
2) No complicated mathematical operations or transformation are needed and also specialized data structure is not necessary.
3) Implementation logic is easy and economical with resources.

Shortcomings of the method are:

1) Noise detection is done in single pass, and filtering is also done in another single pass. Multilevel detection and filtering would improve performance.
2) For Regression, L_1 norm is used. Less calculation is needed, but it results in less accuracy.
3) Present method only concentrates in algebraic methods. Considering frequency information and wavelet based statistics along with, would yield better result in noise detection and removal.
4) For images having high noise density >50%, output suffers from information loss.

VI. Conclusion

We developed and experimentally analyzed an easy to implement method to detect and eliminate impulsive noise successfully from images having continuous tone. As the detection system is based on multiple judgment and verdicts of the windows employed, and the restoration system is based on multiple suggestions about the tentative value of a pixel, it is context aware and less prone to misclassification and loss of sharpness. This majority vote system is simple yet powerful and can be adapted for other types of noise also.

Combining the present method with frequency based methods may give better estimation of the actual value of the corrupted pixel. We target to stack the scheme on a neural network based architecture, involving "experienced" and "learned" agents for further highly sophisticated and careful reduction of all type of noise to minimum. Developing strict theoretical grounds for our scheme with probabilistic models about its behavior is also in process.

References

[1] I. T. Young, J. J. Gerbrands, and L. J. v. Vliet, *Fundamentals of Image Processing,* 1995, Delft: PH Publications. pp. 91-98

[2] R. C. Gonzalez, and R. E. Woods, *Digital Image Processing,* 1st Edition, Addison-Wesley, 1992, pp. 253-254, 268-270, 298-302

[3] J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE PAMI,* vol. 2, no. 2, pp. 165-168, 1980.

[4] H. L. Eng, and K. K. Ma, "Noise adaptive soft-switching median filter for image denoising," *Proc. IEEE ICASSP*, 2000, vol 4, pp. 2175-2178

[5] K. U. Barthel, H. L. Cycon, and D. Marpe, "Image denoising using fractal and wavelet based methods", *Proc. SPIE*, Vol. 5266, pp. 39-47, Feb. 2004

[6] Y. XU, J. B. Weaver, D. M. Healy, Jr., and J. Lu, "Wavelet transform domain filters: spatially selective noise filtration technique", *IEEE Trans. on Image Processing,* Vol. 3, No. 6, Nov. 1994

[7] Wikipedia contributors, "Noise reduction," *Wikipedia, The Free Encyclopedia,* http://en.wikipedia.org/w/index.php?title=Noise_reduction&oldid=58396722 (June 16, 2006).

[8] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D* 60 (1992), 259—268.

[9] M. T. A. Amin, and A. Bhattacharjee, "Impulsive noise reduction in natural images by plane and paraboloid regression," *9th International Conference on Computer and Information Technology,* Dhaka, Bangladesh, December 2006

[10] Chapra, Steven C., and Canale, Raymond P., *"Numerical Methods for Engineers with Software and Programming Applications"*, Fourth Edition, The McGraw-Hill Companies Inc., 2002, pp 440 – 466

[11] Scarborough, James B., *"Numerical Mathematical Analysis"*, Sixth Edition, The Johns Hopkins Press, Baltimore, Md., 1966, pp 460-492

[12] Simoncelli, Eero P., and Adelson, Edward H., “Noise removal via Bayesian Wavelet coring”, *Proceedings, 3rd Annual Intl. Conf. on Image Processing*, Laussanne, Switzerland, 1996. IEEE.