International Journal of Recent Trends in Engineering, Vol 2, No. 1, November 2009

Noise Removal Algorithm for Images Corrupted by Additive Gaussian Noise

Shyam Lal[1], Mahesh Chandra[2] and Gopal Krishna Upadhyay[3]
[1]Department of E & C Engineering, Moradabad Institute of Technology, Moradabad-244001(U.P)-India
[2]Department of E & C Engineering, Birla Institute of Technology, Mesra, Ranchi (Jharkhand)-India
[3]Director, Sri Sri Institute of Technology & Management (SSITM), Kasganj (U.P.) -India
[shyam_rao24@rediffmail.com, shrotriya@bitmesra.ac.in, gkupadhyay2003@yahoo.com]

ABSTRACT-This paper presents noise removal algorithm for gray scale images corrupted by additive Gaussian noise. A robust open close sequence filter based on mathematical morphology for high probability additive Gaussian noise removal is discussed. First, an additive Gaussian noise detector using mathematical residues is to identify pixels that are contaminated by the additive Gaussian noise. Then the image is restored using specialized open-close sequence algorithms that apply only to the noisy pixels. Finally, black and white blocks that degrade the quality of the image will be recovered by a block smart erase method. Simulation and experimental results demonstrate that the robust open close sequence filter outperforms a number of other existing algorithms and is particularly effective for highly corrupted images.

Keywords: Dilation, Erosion, Opening, Closing, Median filter, CWM Filter, PWMAD Filter, DWT & DTDWT

I. INTRODUCTION

The goal of removing additive Gaussian noise is to suppress the noise while preserving the integrity of edge and detail information associated with the original image. To this end, nonlinear techniques have been found to provide more satisfactory results in comparison with linear methods. The traditional median filters process all the pixels of the image, without taking account of whether or not a pixel is corrupted by additive Gaussian noise [1]. To avoid this disadvantage, it is important to detect whether the current pixel is corrupted. These approaches involve a preliminary identification of corrupted pixels in an effort to prevent alteration of true pixels. Among those are center weighted median (CWM) filter [2-4], pixel-wise MAD (PWMAD) [5-6],discrete wavelet transform[18-22], dual tree discrete wavelet transform[18-22] and recently, additive Gaussian noise removal based on mathematical morphology has been attracting research effort [7-12]. Mathematical morphology is nonlinear image processing methodology. After that mathematical morphology has become popular in the image processing field due to its rigorous mathematical description and its proven applicability in a number of imaging problems including noise elimination, feature extraction and image compression [13-16]. To remove the commonly found noise without losing details of the image, morphological filters with directional, multiple structuring elements, or multiscale filtering have been used. Although the above filters & wavelet transforms give a better performance in terms of additive Gaussian noise removal. In this paper, the performance of a robust open-close sequence (OCS) filter has been evaluated and compared with the performance of other existing filters and discrete wavelet transform to restore images that are corrupted by noise variance from 10 to 80.

II. MATHEMATICAL MORPHOLOGY

Mathematical morphology is nonlinear image processing methodology that is based on the application of lattice theory to spatial structures. After that, mathematical morphology has become popular in the image processing field, due to its rigorous mathematical description and its proven applicability in a number of imaging problems, including noise elimination, feature extraction, and image compression [13-16].

Two fundamental mathematical morphological operations are: dilation and erosion. In fact, many of the morphological algorithms are based on these two primitive operations. In gray scale images, we develop algorithms for boundary extraction via a morphological gradient operation, and for region partitioning based on texture content. Mathematical morphological operations are also useful in smoothing and sharpening, which often are useful as per or post processing steps [13-16].

A. Dilation:

If A and B as sets in Z^2, the dilation of A and B, denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \phi \right\} \quad (1)$$

This equation is based on obtaining the reflection of B about its origin and shifting this reflection by z. the dilation of A and B then is the set of all displacement, z, such that $\widehat{B}$ and A overlap by at least one element. Gray scale dilation of *f* by *b*, denoted $f \oplus b$, is defined as

$$(f \oplus b)(s,t) = \max\{ f(s-x, t-y) + b(x,y) \mid (s-x), (t-y) \in D_f ; (x,y) \in D_b \} \quad (2)$$

where D_f and D_b are the domains of f and b, respectively.

B. Erosions:

If A and B as sets in Z^2, the erosion of A and B, denoted $A \Theta B$, is defined as

$$A \Theta B = \left\{ z \mid (\hat{B})_z \subseteq A \right\} \quad (3)$$

This equation indicates that the erosion of A and B is the set of all points z such that B, translated by z, is contained in A. Gray scale erosion of *f* by *b*, denoted $f \Theta b$, is defined as

$$(f \ominus b)(s,t) = \min\{f(s+x,t+y) - b(x,y) \mid (s+x),(t+y) \in D_f; (x,y) \in D_b\} \quad (4)$$

where D_f and D_b are the domains of f and b, respectively.

C. Opening:

Opening of image f by subimage (structuring element) b, denoted $f \circ b$ is

$$f \circ b = (f \ominus b) \oplus b \quad (5)$$

Opening is simply the erosion of f by b, followed by a dilation of the result by b. The opening simplifies by removing the bright components that do not fit within the structuring element.

D. Closing:

Closing of image f by subimage (structuring element) b, denoted $f \bullet b$ is

$$f \bullet b = (f \oplus b) \ominus b \quad (6)$$

Closing is simply the dilation of f by b, followed by *a* erosion of the result by b. The closing simplifies by removing the dark components that do not fit within the structuring element.

III. DISCRETE WAVELET TRANSFORM

A. Wavelet Transform:

The simplest wavelet transform for multi-dimensional digital data is the critically-sampled separable wavelet transform. This transform uses a 1-D wavelet transform in each dimension and is the one that is conventionally used. However, one way to improve the performance of wavelet-based signal and image processing algorithms is to use specialized wavelet transforms in place of the conventional wavelet transform. There are several advances in the design of specific wavelet transforms that lead to substantially improved performance. For example, the undecimated wavelet transform, the steerable pyramid and curvelet transform all give improved results in applications involving multidimensional data. Recently dual-tree transform (an oriented complex-valued wavelet transform) have proved to be highly beneficial for multi-dimensional signal processing. This transform has advantages of near shift invariance, directional selectivity and improved energy compaction [18-22] over the conventional multi-dimensional wavelet transform.

B. Separable Discrete Wavelet Transform (SDWT)

B.1. 1-D Discrete Wavelet Transform

The analysis filter bank decomposes the input signal x(n) into two sub band signals, c(n) and d(n). The signal c(n) represents the low frequency part of x(n), while the signal d(n) represents the high frequency part of x(n). We denote the low pass filter by af1 (analysis filter 1) and the high pass filter by af2 (analysis filter 2). As shown in the Fig.1, the output of each filter is then down sampled by 2 to obtain the two sub band signals c(n) & d(n).

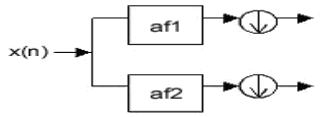

Fig.1. The Analysis filter bank

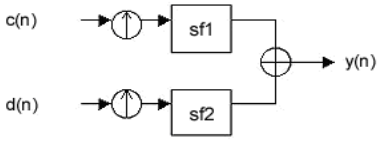

Fig. 2. The Synthesis filter bank

The Synthesis filter bank combines the two sub band signals c(n) & d(n) to obtain a single signal y(n). The synthesis filters bank up-samples each of the two sub band signals. The signals are then filtered using a low pass and a high pass filter. We denote the low pass filter by sf1 (synthesis filter1) and the high pass filter by sf2 (synthesis filter 2) as shown in the Fig.2. The signals are then added together to obtain the signal y(n). If the four filters are designed so as to guarantee that the output signal y(n) equals the input signal x(n), then the filters are said to satisfy the perfect reconstruction condition[19-21].

B.2. 2-D Discrete Wavelet Transform

a. 2-D Filter Banks

When wavelet transform is used in image processing applications it is required to implement a 2D version of the analysis and synthesis filter banks. In the 2D case, the 1D analysis filter bank is first applied to the columns of the image and then applied to the rows. If the image has N1 rows and N2 columns, then after applying the 1D analysis filter bank to each column we have two sub band images, each having N1/2 rows and N2 columns; after applying the 1D analysis filter bank to each row of both of the two sub band images, four sub band images are obtained, each having N1/2 rows & N2/2 columns. This is illustrated in the Fig.3. The 2D synthesis filter bank combines the four sub band images to obtain the original image of size N1 by N2.

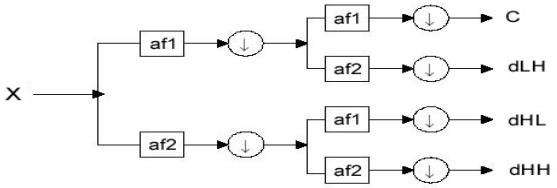

Fig. 3. One stage in multi-resolution wavelet decomposition of an image

The 2D discrete wavelet transform of a signal x is implemented by iterating the 2D analysis filter bank on the low-pass sub-band image [18-25].

C. Bivariate Shrinkage Functions for Wavelet Based image enhancement.

We have considered non-Gaussian bivariate probability distribution function to model the statistics of wavelet coefficients of natural images. The model captures the dependence between a wavelet coefficient and its parent. Using Bayesian estimation theory we derive from this model a simple non-linear shrinkage function for wavelet denoising, which generalizes the soft thresholding approach of Donoho and Johnstone. The new shrinkage function, which depends on both the coefficient and its parent, yields improved results for wavelet-based image denoising [18-20].

Let w2 represent the parent of w1 (w2 is the wavelet coefficient at the same spatial position as w1, but at the next coarser scale). Then

$$y = w + n \tag{7}$$

where w = (w1,w2), y = (y1,y2) and n = (n1,n2). The noise values n1, n2 are iid zero-mean Gaussian with variance \sigma_n^2. Based on the empirical histograms we have computed and define the following non-Gaussian bivariate pdf

$$p_w(w) = \frac{3}{2\pi\sigma^2}.\exp\left(-\frac{\sqrt{3}}{\sigma}\sqrt{w_1^2 + w_2^2}\right) \tag{8}$$

With this pdf, w1 and w2 are uncorrelated, but not independent. The MAP estimator of w1 yields the following bivariate shrinkage function

$$\hat{w} = \frac{\left(\sqrt{y_1^2 + y_2^2} - \frac{\sqrt{3}\sigma_n^2}{\sigma}\right)}{\sqrt{y_1^2 + y_2^2}}.y_1 \tag{9}$$

For this bivariate shrinkage function, the smaller the parent value, the greater will be the shrinkage. This is consistent with other models, but here it is derived using a Bayesian estimation approach beginning with the new bivariate non-Gaussian mode [18-37].

D. Local Adaptive Image Enhancement

Using the bivariate shrinkage function shown in Eqa.(9), we developed an effective and low complexity locally adaptive image denoising algorithm in [18-20]. This shrinkage function requires the prior knowledge of the noise variance and the signal variance for each wavelet coefficient. Therefore the algorithm first estimates these parameters [18-20] for the recipe of the estimation rules.

Briefly, the algorithm is summarized as follows:

1. Calculate the noise variance.
2. For each wavelet coefficient,
   i. Calculate signal variance,
   ii. Estimate each coefficient using the bivariate shrinkage function.

E. MATLAB Implementation Procedure

1. Set the window size. The signal variance of a coefficient will be estimated using neighboring coefficients in a rectangular region with this window size.
2. Set how many stages will be used for the wavelet transform.
3. Extend the noisy image. The noisy image will be extended using symmetric extension in order to reduce the boundary problem. Calculate the forward wavelet transform.
4. Estimate the noise variance. The noise variance will be calculated using the robust median estimator.
5. Process each subband separately. We will process each subband in a loop since our implementation stores the wavelet coefficients in a cell array. First the coefficient and the corresponding parent matrices are prepared for each subband.
6. Estimate the signal variance and the threshold value: The signal variance for each coefficient is estimated using the window size and the threshold value for each coefficient will be calculated and stored in a matrix with the same size as the coefficient matrix.
7. Estimate the coefficients. The coefficients will be estimated using the noisy coefficient, its parent, and the estimated threshold value with the Bivariate Shrinkage Function.
8. Calculate the inverse wavelet transform.
9. Extract the image. The necessary part of the final image is extracted in order to reverse the symmetric extension operation.

F. Dual Tree Discrete Wavelet Transform (DTDWT)

There are two versions of the 2D dual-tree wavelet transform: the real 2-D dual-tree DWT is 2-times expansive, while the complex 2-D dual-tree DWT is 4-times expansive. Both types have wavelets oriented in six distinct directions. We describe the real version first [18] [20-25].

a). Real 2-D Dual-tree Wavelet Transform

The real 2-D dual-tree DWT of an image x is implemented using two critically-sampled separable 2-D DWTs in parallel. Then for each pair of subbands we take the sum and difference [18-25].

b) Complex 2-D Dual-tree Wavelet Transform

The dual-tree complex DWT (DT-CDWT) of a signal x is implemented using two critically-sampled DWTs in parallel on the same data, as shown in the Fig. 4.

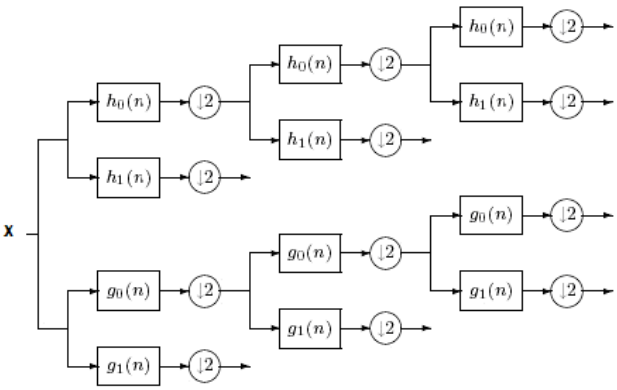

Fig.4. Dual Tree Complex Discrete wavelet Transform

This transform is 2-times expansive because for an N-point signal it gives 2N DWT coefficients. If the filters in the upper and lower DWTs are the same, then no advantage is gained. However, if the filters are designed in a specific way then the subband signals of the upper DWT can be interpreted as the real part of a complex wavelet transform and subband signals of the lower DWT can be interpreted as the imaginary part. Equivalently, for specially designed sets of filters, the wavelet associated with the upper DWT can be an approximate Hilbert transform of the wavelet associated with the lower DWT. When designed in this way, the dual-tree complex DWT is nearly shift-invariant, in contrast with the critically-sampled DWT. Moreover, the dual-tree complex DWT can be used to implement 2D wavelet transforms where each wavelet is oriented, which is especially useful for image processing. For the separable 2D DWT, recall that one of the three wavelets does not have a dominant orientation. The dual-tree complex DWT outperforms the critically-sampled DWT for applications like image denoising and enhancement.

The complex 2-D dual-tree DWT also gives rise to wavelets in six distinct directions, however, in this case there are two wavelets in each direction as will be illustrated

below. In each direction, one of the two wavelets can be interpreted as the real part of a complex-valued 2D wavelet, while the other wavelet can be interpreted as the imaginary part of a complex-valued 2D wavelet. Because the complex version has twice as many wavelets as the real version of the transform, the complex version is 4-times expansive. The complex 2-D dual-tree is implemented as four critically-sampled separable 2-D DWTs operating in parallel. However, different filter sets are used along the rows and columns. As in the real case, the sum and difference of sub-band images is performed to obtain the oriented wavelets [18-37].

One of the advantages of the dual-tree complex wavelet transform is that it can be used to implement 2D wavelet transforms that are more selective with respect to orientation than is the separable 2D DWT.

G. MATLAB Implementation Procedure:

1. Set the window size. The signal variance of a coefficient will be estimated using neighboring coefficients in a rectangular region with this window size.
2. Set how many stages will be used for the wavelet transform.
3. Extend the noisy image. The noisy image will be extended using symmetric extension in order to improve the boundary problem.
4. Calculate the forward dual-tree DWT.
5. Estimate the noise variance. The noise variance will be calculated using the robust median estimator.
6. Process each subband separately in a loop. First the real and imaginary parts of the coefficients and the corresponding parent matrices are prepared for each subband
7. Estimate the signal variance and the threshold value: The signal variance for each coefficient is estimated using the window size and the threshold value for each coefficient will be calculated and stored in a matrix with the same size as the coefficient matrix.
8. Estimate the magnitude of the complex coefficients. The coefficients will be estimated using the magnitudes of the complex coefficient, its parent and the threshold value with the Bivariate Shrinkage Function.
9. Calculate the inverse wavelet transform.
10. Extract the image. The necessary part of the final image is extracted in order to reverse the symmetrical extension.

IV. ROBUST OPEN-CLOSE SEQUENCE FILTER

The robust open-close sequences (OCS) filter to restore images that are corrupted by noise variance from 10 to 80 probability of additive Gaussian noise. The scheme of the OCS f[illegible]n in Fig.5.

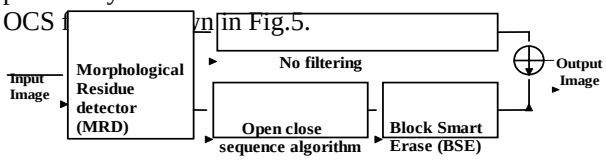

Fig. 5. Open–Close sequence (OCS) filter

Main components of the open close sequence (OCS) filter are as follows:-

- Morphological Residue Detector (MRD)
- Open-close sequence algorithms
- Block Smart Erase (BSE)

In this section, we start by describing the noise detector based on morphological residues. Then we introduce the open-close sequence algorithms and block smart erase method. Finally, the open-close sequence filter is implemented & studied.

A. Morphological Residue Detector (MRD):

In this method, we detect noisy pixels using mathematical morphology. As usual, erosion outputs the minimum value in the structuring element domain, and dilation outputs the maximum value in the structuring element domain. Therefore, the opening & closing operation removes additive Gaussian noise. The determination of the additive Gaussian noise is easily done by comparing the difference between the value of the pixel and the result of opening and closing with a flat structuring element. Let D_o and D_c be the opening and closing absolute distance from input signal respectively

$$D_o = f - f \circ b \tag{10}$$

$$D_c = f \bullet b - f \tag{11}$$

Here D_o and D_c are nonnegative. Noise pixels are detected by comparing D_o and D_c with a threshold T.

$$r(i,j) = \begin{cases} 1, & D_o \geq T \text{ and } D_c = 0 \\ -1, & D_o = 0 \text{ and } D_c \geq T \\ 0, & \text{Otherwise} \end{cases} \tag{12}$$

If $r(i,j)$ is 1, it is regarded as salt noise; if $r(i,j)$ is -1 , it is regarded as a pepper noise; otherwise, if $r(i,j)$ is 0, then $f(i,j)$ is regarded as an original pixel. This detector uses two parameters: the threshold T and the size of the structuring element *b*. It is easy to optimize the parameters by means of experiment. Whenever these noises are not or cannot be detected, the pixel value is considered noiseless, and it is passed thorough. When a certain noise is detected, the corresponding generalized open-close sequence algorithm is selected; otherwise, the input signal is put forward [38-40].

B. Open-Close Sequence Algorithm:

Two filters using open-close sequences are employed to the detected additive Gaussian noise. The first one called open-close filter (OCF) is defined as follows:

$$OCF(f) = (f \circ b_1) \bullet b_2 \tag{13}$$

Multiscale structure elements are introduced to the open-close sequence algorithm. The size of b_1 must be small enough to preserve more details of the image, and the size of b_2 is larger than that of b_1. The opening efficiently eliminates the detected Gaussian noises, but at the same time, the Gaussian noises are magnified by the first erosion. So it is necessary to use closing to remove the additional Gaussian noises. However, infected Gaussian noises whose sizes exceed that of b_1 cannot be eliminated. To achieve better performance, the scale of b_2 must be much larger than that of b_1. It is appropriate that the size of b_1 is 5x 5 (or 7x 7), which is not so large as to remove lots of image details. Then the Gaussian noise can be eliminated powerfully by the following closing while more image details can be

preserved. The second sequence called close-open filter (COF) is defined as follows:

$$COF(f) = (f \bullet b_1) \circ b_2 \quad (14)$$

Corresponding to the first filter, the second one applied to remove the detected Gaussian noises consists of one closing and one opening**.** Both OCF and COF filters are combinations of opening and closing operators and perform efficiently to remove the corresponding Gaussian noises. However, the noises whose size is larger than the size of b_1 will not be eliminated and propagated in the image. The filtered image appears to have some uncomfortable characteristics: undesirable white (or black) blocks are generated during the OCF (or COF) procedure and reserved in the filtered result. Their sizes are large enough and cannot be eliminated by above two sequence filters [38-40].

C. Block Smart Erase Algorithm

The existence of white and black blocks degrades the filtered image significantly and makes the filtered image look uncomfortable. A simple and efficient algorithm called block smart erase (BSE) algorithm is discussed. The BSE algorithm is based on median technology and takes the place of the extreme value (black or white) pixel by the median value of their surrounding pixels [38-40]. The OCS filter can be defined as

$$OCS(f) = \left[\frac{BSE_w(OCF(f)) + BSE_b(COF(f))}{2} \right] \quad (15)$$

D. MATLAB Implementation Procedure:

1) For an NxN window centered at the test pixel, where N would normally be 5, 7, 9,… and larger value should be suggested.
2) If *f(i,j) = 0* or *f(i,j)* = 511, *f(i,j)* is an absolute extreme value pixel that must be estimated; go to step 3, Otherwise, the value of *f(i,j)* is not altered; go to step 4.
3) When an extreme value pixel is detected, its gray level is substituted by the median value of the window.
4) The procedure is repeated for the next window

V. SIMULATION & EXPERIMENT RESULTS

The 8-bit images of dimensions MlxM2 (= 512x512) pixels is used for simulations. The pixels s(i, j) for $1 \le i \le M1$ and $1 \le j \le M2$, of the image is corrupted by adding additive Gaussian noise with noise variance ranging from 10 to 80. In all the simulations, square windows of dimensions NxN (i.e. N=5, 7, 9..) pixels are used. The superiority of filter is demonstrated by conducting two experiments. The peak signal to noise ratio (PSNR) in dB as defined in equation (16) is the metric used to compare the noise removal capability of different filters & discrete wavelet transforms.

A. Peak signal to noise ratio (PSNR): The PSNR between the filtered output image y(i, j) and the original image s(i, j) of dimensions MlxM2 pixels is defined as:

$$PSNR = 20 * \log_{10}\left(\frac{MAX_I^2}{\sqrt{MSE}} \right) \quad (16)$$

Where MAX_I is the maximum pixel value of the image and MSE is is mean squared error and it is defined as

$$MSE = \frac{\sum_{i}^{M1} \sum_{j}^{M2} [y(i, j) - s(i, j)]^2}{M1 \times M2} \quad (17)$$

It can be seen that peak signal to noise ratio (PSNR) is closely related to mean square error (MSE).

Table I. PSNR values for different filters on 'Lena" image

Algorithm	Noise Variance							
	10	20	30	40	50	60	70	80
DWT	44.1888	40.3086	38.2161	36.8147	35.7194	34.8318	34.0989	33.4461
DTDWT	45.458	41.494	39.461	38.058	36.926	36.043	35.299	34.732
Median	55.593	55.593	55.593	55.593	55.593	55.592	55.592	55.593
CWM	65.493	65.304	65.023	64.625	64.202	63.691	62.998	62.469
PWMAD	65.509	65.210	64.941	64.505	64.182	63.557	63.049	62.483
OCS	73.534	73.511	73.362	73.317	72.969	72.981	72.412	72.017

Table II. PSNR values for different filters on "Peppers" image

Algorithm	**Noise Variance**							
	10	20	30	40	50	60	70	80
DWT	41.765	38.479	36.565	35.164	34.054	33.228	32.412	31.746
DTDWT	42.244	39.345	37.556	36.208	35.172	34.306	33.508	32.895
Median	49.519	49.519	49.519	49.516	49.518	49.518	49.518	49.518
CWM	65.081	64.851	64.553	64.217	63.698	63.279	62.696	61.865
PWMAD	65.114	64.869	64.555	64.221	63.662	63.168	62.681	62.031
OCS	71.175	71.173	71.175	71.106	71.118	71.039	70.961	70.935

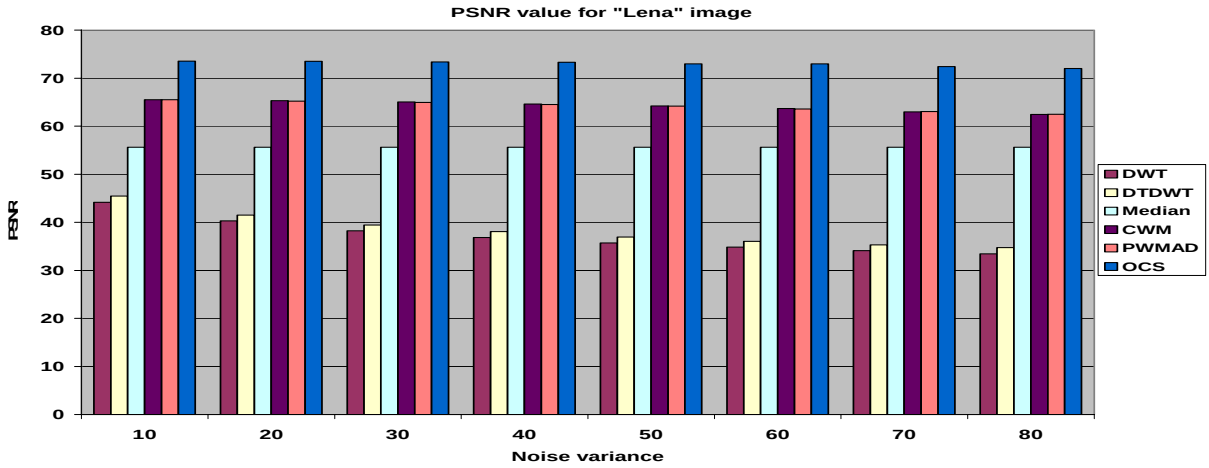

Fig.6. PSNR value on “Lena” image

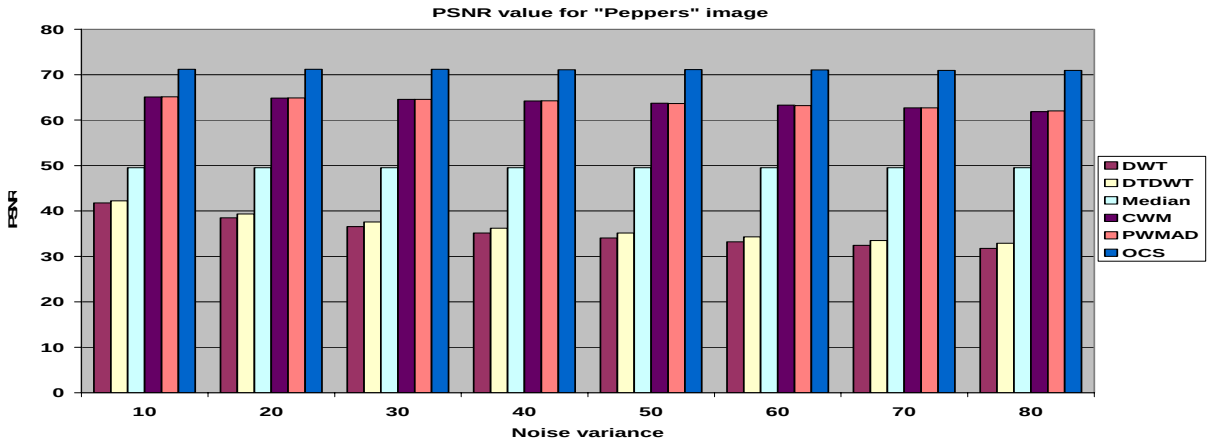

Fig. 7. PSNR value on “Peppers” Image

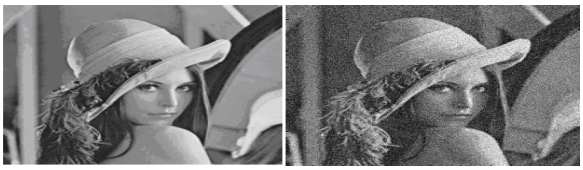

Fig.8.(a)Original image (b) Noisy image with Noise variance 30

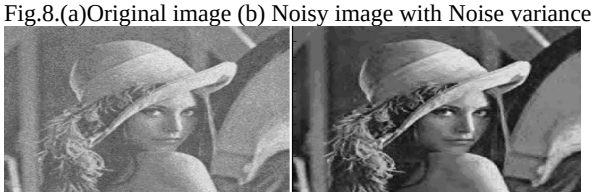

Fig.8.(c) Output from median filter (d) Output from DWT

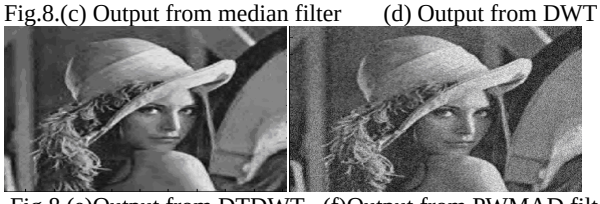

Fig.8.(e)Output from DTDWT (f)Output from PWMAD filter

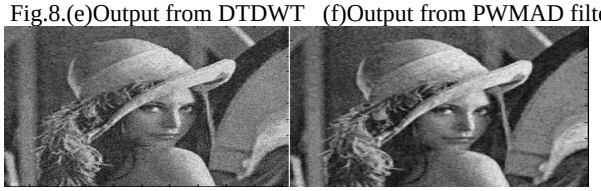

Fig.8.(g)Output from CWM filter (h)Output from OCS filter

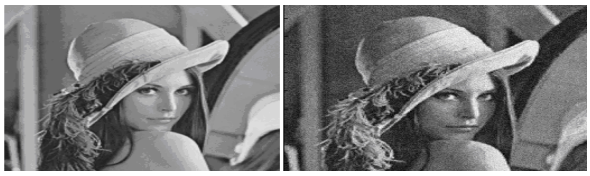

Fig.9.(a)Original image(b) Noisy image with Noise variance 80

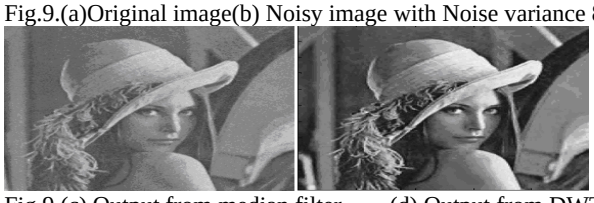

Fig.9.(c) Output from median filter (d) Output from DWT

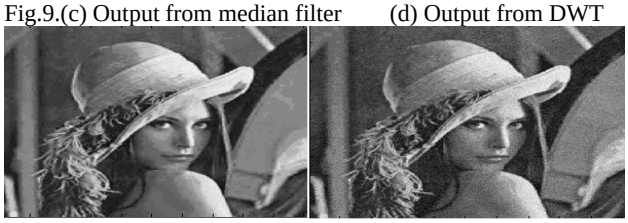

Fig.9.(e)Output from DTDWT (f)Output from PWMAD filter

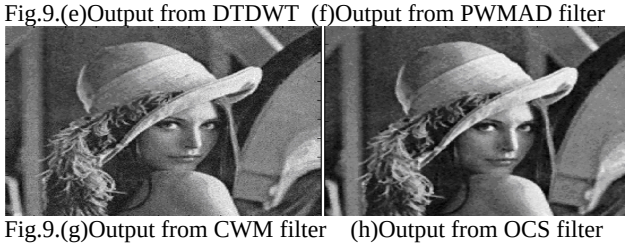

Fig.9.(g)Output from CWM filter (h)Output from OCS filter


Fig.10.(a)Original image(b)Noisy image with Noise variance 30

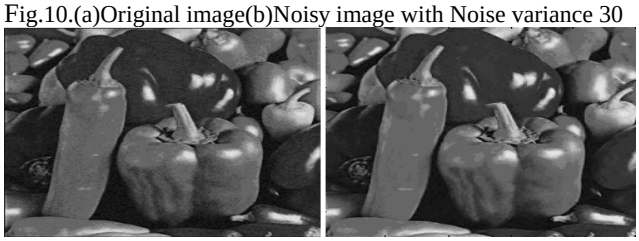
Fig.10.(c) Output from median filter (d) Output from DWT

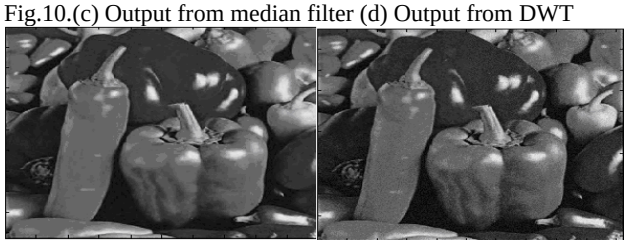
Fig.10.(e)Output from DTDWT (f)Output from PWMAD filter

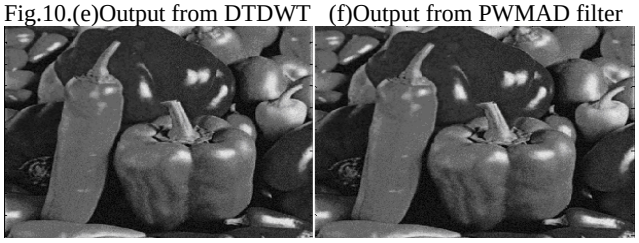
Fig.10. (g)Output from CWM filter (h)Output from OCS filter

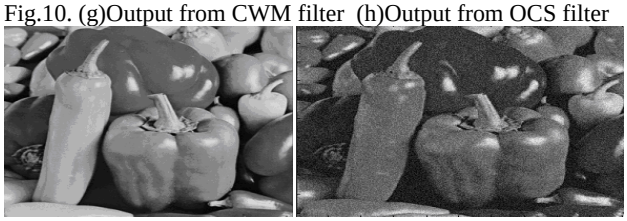
Fig.11.(a)Original image(b)Noisy image with Noise variance 80

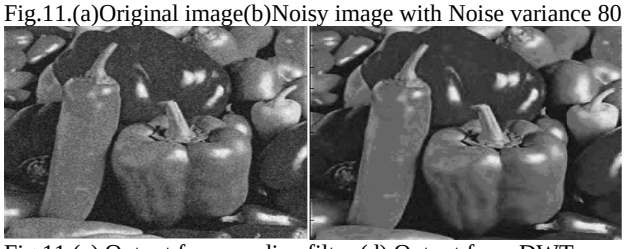
Fig.11.(c) Output from median filter (d) Output from DWT

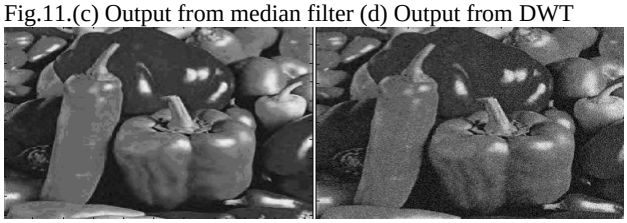
Fig.11.(e)Output from DTDWT (f)Output from PWMAD filter

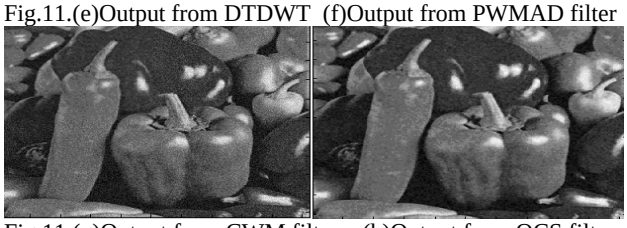
Fig.11.(g)Output from CWM filter (h)Output from OCS filter

B. ***Experiment 1:***

Lena & Peppers images are corrupted with different noise variance ranging from 10 to 80.Varios standard schemes like as median filter, Centre weighted median (CWM) filter, Pixel Wise MAD (PWMAD) filter, DWT, DTDWT and Open close sequence (OCS) filter are simulated .Peak signal to noise ratio (PSNR) is obtained from various schemes for Lena & Peppers images which are plotted and shown in the figure 6 & 7. It can be noticed from figure 6 & 7 that the OCS noise removal scheme outperform in comparison with others existing scheme and it is particularly effective for highly corrupted image.

C. Experiment 2:

To visualize the subjective image enhancement performance, the enhanced Lena and Peppers images are compared with result of median filter, PWMAD filter, CWM filter, DWT, DTDWT and OCS filter and it is shown in figure 8, 9, 10 & 11 with noise variance 30 & 80 for Lena & Peppers images.

VI. CONCLUSION

This paper highlighted the noise removal algorithm for gray scale images corrupted by additive Gaussian noise. In this study the performance of robust open-close sequence filter have been evaluated and compared with the performance of other nonlinear filters and wavelet transforms. The morphological residue detector powerfully determinates the additive Gaussian noise with a low percentage error. The simulation results indicate that the robust open close filter performs better than other nonlinear filtering techniques & wavelet transforms for noise removal from gray scale images. The simulation results also indicate that the robust open close filter also provides better PSNR as compared to other non linear techniques used.

REFERENCE

[1]. W. K. Pratt, “Median filtering,” Image Proc. Institute, University of Southern California, Los Angeles, Tech. Rep., September,1975.

[2]. R.K. Brownrigg, “The weighted median filter,” Communication of ACM, vol. 27 , Issue 8, pp. 807–818,1984.

[3]. T. Sun, M. Gabbouj and Y. Neuvo, "Center weighted median filters: some properties and application in image processing," Signal Processing of ACM, vol. 35, pp. 213-229, 1994.

[4]. T.Sun, M.Gabb,J. and Y.Neuv, “Deterministic properties center weighted median filter” Proceeding of 1992 International Conference on Communication Technology, Beijing, China, Sep. 16-18, , pp.06.09.1-06.09.4., 1992

[5]. T. Chen and H.Wu, “Adaptive impulse detection using center-weighted median filters,” IEEE Signal Process. Lett., vol. 8, no. 1, pp. 1–3, Jan 2001.

[6]. V. Crnojevic, V. Senk, and Z. Trpovski, “Advanced impulse detection based on pixel-wise mad,” IEEE Signal Process. Lett., vol. 11, no. 7, pp. 589–592, Jul. 2004.

[7]. E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, “A new efficient approach for the removal of impulse noise from highly corrupted images,” IEEE Trans. Image Process., vol. 5, no. 6, pp. 1012–1025, Jun. 1996.

[8]. J. Oh and L. F. Chaparro, “Adaptive fuzzy morphological filtering of images,” in Proc. IEEE Int. Conf. Acoustic, Speech, Signal Processing, vol. 5, pp. 2901–2904, May 1998.

[9]. R. H. Chan, C.-W. Ho and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail preserving regularization," IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1479–1485, 2005.
[10]. P.-E. Ng and K.-K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," IEEE Transactions on Image Processing, vol. 15, pp. 1506–1516, 2006.
[11]. A. Bovik, Handbook of Image and Video Processing. Academic Press, 2000.
[12]. Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images, "IEEE Transactions on Circuits and Systems II," vol. 46, pp. 78–80, 1999.
[13]. Shyam Lal, Gaurav Saxena & Akhilesh R Upadhyay, "Analysis of Digital Images with Morphological Technique", Proc.Technologia-2007, MPCCET Bhilai, Chhattisgarh-India, March 02-03, 2007.
[14]. J. Serra, Image, "Analysis and Mathematical Morphology", Academic Press, New York, 1988.
[15]. J. Song and E. Delp, "The analysis of morphological filters with multiple structuring elements," Comput. Vis. Graph., Image Process., vol. 50, pp. 308–328, Jun. 1990.
[16]. S. Mukhopadhysy and B. Chanda, "An edge preserving noise smoothing technique using multi-scale morphology," Signal Processing of ACM, vol. 82, pp. 527–544, 2002.
[17]. S. Zhang and M. A. Karim, "A new impulse detector for switching median filters," IEEE Signal Process. Lett., vol. 9, no. 11, pp. 360–363, Nov. 2002.
[18]. R.Gomathi & S.Selvakumaran, "A Bivariate Shrinkage Function For Complex Dual Tree DWT Based Image Denoising ," In Proc. ICWAMS-2006, Bucharest, Romania, October 16-18, 2006.
[19]. L. Sendur and I. W. Selesnick, " Bivariate shrinkage with local variance estimation," IEEE Signal Processing Letters, 9(12), December 2002.
[20]. L. Sendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," IEEE Trans. on Signal Processing, 50(11):2744-2756, November 2002.
[21]. L. Sendur, I.W. Selesnick, "A bivariate shrinkage function for wavelet-based denoising", IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002.
[22]. L. Sendur, I.W. Selesnick, "Subband adaptive image denoising via bivariate shrinkage", IEEE International Conference on Image Processing (ICIP), 2002.
[23]. N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," Applied and Computational Harmonic Analysis, 10(3):234-253, May 2002.
[24]. N. G. Kingsbury, "Image processing with complex wavelets," Phil. Trans. Royal Society London A, September 1999.
[25]. W. T. Freeman and E. H. Adelson, "The design and use of steerable filters", IEEE Trans. Patt. Anal. Mach. Intell., 13(9):891-906, September 1991.
[26]. A. F. Abdelnour and I. W. Selesnick, "Nearly symmetric orthogonal wavelet bases", In Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP), May 2001.
[27]. F. Abramovich and Y. Benjamini,"Adaptive thresholding of wavelet coefficients," *Comput. Statist. Data Anal.*, vol. 22, pp. 351–361, 1996.
[28]. F. Abramovich, T. Sapatinas, and B. Silverman, "Wavelet thresholding via a Bayesian approach," *J. R. Stat.*, vol. 60, pp. 725–749, 1998.
[29]. Z. Cai, T. H. Cheng, C. Lu, and K. R. Subramanian, "Efficient wavelet based image denoising algorithm," *Electron. Lett.*, vol. 37, no. 11, pp. 683–685, May 2001.
[30]. S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Processing*, vol. 9, pp. 1532–1546, Sept. 2000.
[31]. S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Trans. Image Processing*, vol. 9, pp. 1522–1531, Sept. 2000
[32]. M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, pp. 886–902, Apr. 1998.
[33]. D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, May 1995.
[34]. M. A. T. Figueiredo and R. D. Nowak, "Wavelet-based image estimation: An empirical bayes approach using Jeffrey's noninformative prior," *IEEE Trans. Image Processing*, vol. 10, pp. 1322–1331, Sept. 2001.
[35]. A. Hyvarinen, E. Oja, and P. Hoyer, "Image denoising by sparse code shrinkage," in *Intelligent Signal Processing*, S. Haykin and B. Kosko, Eds. Piscataway, NJ: IEEE, 2001.
[36]. N. G. Kingsbury, "Image processing with complex wavelets," *Phil. Trans. R. Soc. London A*, Sept. 1999.
[37]. E. P. Simoncelli, "Bayesian denoising of visual images in the wavelet domain," in *Bayesian Inference in Wavelet Based Models*, P. Müller and B. Vidakovic, Eds. New York: Springer-Verlag, 1999.
[38]. V. Strela, J. Portilla, and E. Simoncelli, "Image denoising using a local Gaussian scale mixture model in the wavelet domain," in Proc. SPIE 45th Annu. Meet., 2000.
[39]. Deng Ze Feng, Yin Zhou Ping, and Xiong You Lou. "High probability impulse noise removing algorithm based on mathematical morphology", IEEE Signal Process. Lett., vol. 14, no. 1, pp. 31–34, Jan. 2007.
[40]. Shyam Lal, Akhilesh Shukla, Amit Sharma, Pradeep Gupta , Sanjay Kumar ,"A Robust Open-Close Sequence Filter Based on Mathematical Morphology", IJIES, Vol. 2, pp. 23-29, Chennai (T.N.), India, Nov. 2008.