PERFORMANCE ANALYSIS OF A MODIFIED DECOMPOSITION FILTER FOR NON IDENTICAL NOISES

K. Vasanth\textsuperscript{1} and S. Karthik\textsuperscript{2}
\textsuperscript{1}Sathyabama University, Tamil Nadu, India
E-mail: vasanthece_k@yahoo.co.in
\textsuperscript{2}Cognizant Technology solutions, Tamil Nadu, India
E-mail: skarthick76@gmail.com

Abstract
The proposed work aims in the restoration of images corrupted by Gaussian noise, impulse noise. The new algorithm significantly removes different noises and produce better image quality than standard median filter (SMF), Centre weighted median filter (CWF) and threshold decomposition filter (TDF). The proposed algorithm (PA) is tested on different images corrupted by all two noises and is found to produce better results in terms of the qualitative and quantitative measures of the image for noise densities up to 30\% noise level for impulse noise, mean zero and 0.9\% variance of Gaussian noise. The filter works well for speckle noise up to 0.8\% variance.

Keywords:
Impulse Noise, Median Filter, Threshold Decomposition, Non-linear Filter

1. INTRODUCTION

Images are often corrupted by noise, due to degradation introduced at the input channels, transmission medium, sensor and/or digitizer. Common types of degradation are blurring, distortion, additive random noise such as Gaussian white noise and salt-and-pepper impulse noise, signal-dependent noise such as speckle, film grain noise and quantization noise [2]. In order to restore back these images, a proper filter should be carefully chosen. A good noise removal filter would remove the additive noise distributions exactly, restoring the original image from the noisy image completely. To do this, the filtering algorithm must be specially designed to remove a particular noise distribution. In reality, no matter how well a noise removal filter is designed, the restored image always exhibits a certain degree of deviation in its pixel values from the original image. Excessive deviation often renders the restored image useless. In other words, the restored image may be visually unacceptable if subjected to human inspection [3]. The additive white Gaussian noise which are caused by poor image acquisition or by transferring the image data in noisy communication channels. Gaussian noise removal can be effectively done by linear filtering methods. Impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel. Two common types of impulse noise are the salt-and-pepper noise and the random-valued noise. For images corrupted by salt-and pepper noise, the noisy pixels can take only the maximum and the minimum values while in the case of random-valued noise; they can take any random value in the dynamic range. Speckle is a random, deterministic, interference pattern in an image formed with coherent radiation of a medium containing many sub-resolution scatterers. The texture of the observed speckle pattern does not correspond to underlying structure. The local brightness of the speckle pattern, however, does reflect the local echogenicity of the underlying scatterers [3]. There are two basic approaches to image de-noising, spatial filtering methods and transform domain filtering methods [4]. A traditional way to remove noise from image data is to employ spatial filters. Spatial filters can be classified into non-linear and linear filters. Many non-linear filters fall into the category of order statistic neighbor operators [5]. This means that the local neighborhoods are sorted into ascending order and this list is processed to give an estimate of the underlying image brightness. The simplest order statistic operator is the median [6], where the central value in the ordered list is used for the new value of the brightness. The median is good at reducing impulse noise. However, A mean or average filter is the optimal linear filter for Gaussian noise removal which tend to blur sharp edges, destroy lines and other fine image details, and perform poorly in the presence of signal-dependent noise. This paper is organized as follows. Section II describes noise model. Section III gives a brief review of related work on Image De-noising using proposed algorithm. Section IV deals with Exhaustive Experimental Results and Discussions and finally Concluding Remarks are given in Section V.

2. NOISE MODEL

Let the true image \(x\) belong to a proper function space \(S(\Omega)\) on \(\Omega = [0: 1]^2\), and the observed digital image \(y\) be a vector in \(R_{m \times m}\) indexed by \(A = \{1,2,..,m\} \times \{1,2,..,m\}\). The image degradation can be modeled as \(y = N(Hx)\), where \(H : S(\Omega) \rightarrow R_{m \times m}\) is a linear operator representing blurring, and \(N : R_{m \times m} \rightarrow R_{m \times m}\) models the noise. Usually, \(y = Hx + \alpha n\) where \(\alpha n \in C R_{m \times m}\) is an additive zero-mean Gaussian noise with standard deviation \(\sigma\geq 0\). Outliers are modeled as impulse noise. For an overview, see [7].

\[
y' = Hx + \alpha g \quad \text{(1)}
\]

\[
y = N(y') \quad \text{(2)}
\]

where \(N\) represents the impulse noise. There are two common models for impulse noise: the salt-and-pepper noise and the random-valued noise. If \([d_{\text{min}}; d_{\text{max}}]\) denote the dynamic range of \(y'\), i.e., \(d_{\text{min}} \leq y'ij \leq d_{\text{max}}\) for all \((i,j)\), then they are denoted by Salt-and-pepper noise: the gray level of \(y\) at pixel location \((i,j)\) is

\[
y_{ij} = d_{\text{min}}; \text{with probability } p;
\]

\[
d_{\text{max}}; \text{with probability } q;
\]

\[
y'_{ij}; \text{with probability } 1 - p - q;
\]

Where \(s = p + q\) denotes the salt-and-pepper noise level [7].
3. PROPOSED WORK

In the existing threshold decomposition techniques, threshold levels from 0-255 are used, based upon which the pixels in the window are decomposed into strings of 1s and 0s, depending on whether the pixel intensity is greater than or lesser than the threshold level. Then the majority function is found out at each level which is recombined to produce the median value. The pixel to be processed is then replaced by the median value. Large number of threshold levels and bit comparisons are used in determining the majority function at each level, which increases the complexity of the process and the time taken for processing. The complexity of the process can be described as follows:

Stage 1: The stage involves the process of decomposing the pixels into 1s are required and 0s demands 256 one bit comparisons for each pixel.

Stage 2: The process of computing the majority function involves 9 one-bit comparisons at each threshold level. So, 256X9 comparisons are required for a 3X3 window.

Stage 3: 255 one bit comparisons are required for the process of recombining the 1's, to obtain the median value.

3.1. PROPOSED ALGORITHM

The aim of the work is to apply the proposed filter over an image corrupted by mixed noises (zero mean Gaussian and impulse noise). Figure 1 denotes the aim of the work. To overcome this problem, a new algorithm is proposed in which the pixel intensity itself is considered as the threshold and decomposed into its equivalent string of 1s, thereby reducing the number of thresholds. The median is found eliminating the process of finding out the majority function which in turn eliminates the process of comparison. Proposed algorithm is given as follows:

**STEP 1:** A 2D window of size 3x3 is selected. Assume the pixel to be processed is p(x,y).

**STEP 2:** Every pixel of the window is decomposed into its number equivalent strings of 1's considering the pixel intensity itself as the threshold. Here the decomposition is done with the help of a counter ROW1, which eliminates the comparison involved in decomposition process of the conventional and existing threshold decomposition techniques. Simultaneously, the number of 1’s in each column is counted with the help of a counter and its number equivalent is stored in COL1 simultaneously.

**STEP 3:** The values of COL1 counter are decomposed into its equivalent strings of 1’s and the number of 1’s at each column is recombined to obtain the pixel intensities of the window sorted in descending order with the help of counter VAL. The fifth element of the VAL or the number equivalent of the fifth column counter gives the median of the window considered. After the computation of median, the centre pixel of the window is replaced by the evaluated median. Subsequently, the window moves towards the right for a new set of window values; this processing as well as the updating procedure are repeated until the end of the image element is reached. Fig 2 denotes the methodology of proposed algorithm.

4. SIMULATION RESULTS

This Section experimentally analyzes the performance of developed image denoising algorithm with various median

![Fig.1. Insight of the proposed filter on mixed noises](image1)

![Fig.2. Methodology of the proposed algorithm](image2)
filters, such as, standard median filter (inbuilt MATLAB function) SMF, Center weighted median filter (CWF), Threshold decomposition filter (TDF), for Gaussian, Speckle and Salt & Pepper noise added on images such as Lena, Barbara, Baby, girl, Pepper and Cameraman image. It is experimentally proved that the proposed algorithm is as optimal for better denoising of different noises. Filtering performance can be evaluated by computing Peak Signal to Noise Ratio (PSNR). Image enhancement factor (IEF) and time using (matlab inbuilt functions) which are the estimates of the quality of a filtered image compared with an original image. The PSNR is calculated using the formulae.

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right)$$

$$\text{MSE} = \frac{\sum_{i,j} (r_{ij} - x_{ij})^2}{MN}$$

Where, r - Original image, MxN - size of image, x - restored image. The Image enhancement factor is calculated using the formulae

$$\text{IEF} = \frac{\left( \sum_{i,j} n_{ij} - r_{ij} \right)^2}{\left( \sum_{i,j} x_{ij} - r_{ij} \right)^2}$$

Where n - corrupted image, r - original image x - restored image [1].

The PSNR, IEF, and CPU computation time in seconds for impulse noise, zero mean Gaussian noise and Speckle noise are calculated for the PA and compared with SMF, CWF and TDF, in Tables 1 to 3 for lena.gif. The important aspect of the PA is that it uses a fixed 3x3 window for processing and thus leads to smaller computation time amongst the existing threshold decomposition filters or stack filters and centre weighted median filter. MATLAB 7.0(R14) on a PC equipped with 2-GHz CPU and 1GB of RAM memory has been employed for the evaluation of computation time of all algorithms. It was found from tables 1-3 that the proposed algorithm has better performance in removing impulse noise up to 30%. From table 5 and 6 it was observed that the proposed algorithm has capability to eliminate zero mean with 0.9% Gaussian noise and speckle noise up to 0.8%. Considering the discussions made before, Subsequent Tables 4 to 6 represents the performance of the SMF, CWF, TDF and PA for five different images by above said compositions of noises respectively. Table 7 and 8 shows the performance of the PA is better in terms of PSNR, IEF and optimum time when compared with SMF, CWF, and TDF for various types of images corrupted by all three types of noises in proportion. Fig 3-11 illustrates the performance of the PA over other filters for impulse noise, Gaussian noise and speckle noise. In fig 12-13 PA has higher PSNR, IEF when tested on different images which is corrupted by 30% impulse noise. In fig 15, 16 PA has slightly better PSNR, IEF over other filters that are used for denoising zero mean variance 0.9% Gaussian noise tested on various images. It was observed that for the images which have gray levels varying more (details of an image) such as cameraman.bmp, barbera.tif, girl.jpg the PA performance is average when compared with other filters. For the images whose gray levels is uniform (details of the image) such as babay.jpg, pepper.bmp the performance of the PA is good when compared with other filters. In fig 18 the performance of PA is good when compared with other filters for 0.8% speckle noise. From fig 19 we understand such that depending upon the variation in grey levels in an image the performance is good or average. IEF of the PA good on par with other filters if the grey level changes are more else the performance is average. Fig 21-22 gives the performance of PA over different images corrupted by mixed noises in some proportion has a good PSNR and IEF. Fig 24-27 shows pictorial representation obtained by employing various filters. Fig 5, 8, 11, 14, 17, 20, 23 denotes the optimum computational speed at which the PA works.

![Fig.3. Noise density versus PSNR for various filters for Lena image corrupted by impulse noise](image)

Table.1. PSNR, IEF, TIME for LENA.GIF (512 X 512) image corrupted by impulse noise at different noise densities

| ND  | SMF     | TDF     | CWF     | PA     | SMF     | TDF     | CWF     | PA     | SMF     | TDF     | CWF     | PA     | SMF     | TDF     | CWF     | PA     |
|-----|---------|---------|---------|--------|---------|---------|---------|--------|---------|---------|---------|--------|---------|---------|---------|--------|
| 10% | 34.927  | 32.775  | 35.234  | 35.934 | 89.055  | 38.253  | 95.903  | 99.675 | 1.544   | 421.871 | 24.804  | 46.743 |
| 20% | 30.305  | 27.841  | 28.136  | 31.713 | 61.079  | 25.055  | 37.278  | 67.702 | 1.404   | 441.934 | 20.545  | 45.968 |
| 30% | 23.992  | 23.369  | 22.262  | 25.395 | 21.415  | 19.642  | 14.428  | 23.638 | 1.342   | 457.816 | 21.107  | 48.544 |
| 40% | 19.023  | 19.012  | 17.853  | 19.238 | 9.181   | 9.226   | 6.947   | 9.586  | 1.373   | 481.09  | 28.548  | 49.024 |
| 50% | 15.934  | 15.32   | 14.38   | 15.393 | 4.953   | 4.885   | 3.925   | 4.956  | 1.373   | 497.975 | 21.091  | 49.349 |
Table 2. PSNR, IEF, TIME for LENA.GIF (512 X 512) image corrupted by zero mean gaussian noise at different noise densities

| Noise Density | PSNR | IEF | TIME |
|---------------|------|-----|------|
| 60%           | 12.36| 11.748 | 12.357 | 2.958 | 2.986 | 2.572 | 2.946 | 1.357 | 509.552 | 19.375 | 49.347 |
| 70%           | 10.085 | 10.019 | 9.62 | 10.042 | 2.036 | 2.014 | 1.835 | 2.022 | 1.326 | 519.921 | 24.321 | 50.525 |
| 80%           | 8.159 | 8.103 | 7.973 | 8.143 | 1.496 | 1.483 | 1.429 | 1.492 | 1.388 | 519.314 | 21.185 | 50.774 |
| 90%           | 6.607 | 6.608 | 6.569 | 6.62 | 1.182 | 1.181 | 1.167 | 1.183 | 1.373 | 526.499 | 19.516 | 51.45 |

Table 3. PSNR, IEF, TIME for LENA.GIF (512 X 512) image corrupted by speckle noise at different noise densities

| Noise Density | PSNR | IEF | TIME |
|---------------|------|-----|------|
| 0.001         | 34.08 | 29.384 | 34.092 | 34.126 | 2.656 | 1.078 | 2.575 | 2.597 | 1.444 | 227.29 | 34.092 | 40.863 |
| 0.002         | 32.403 | 28.807 | 32.211 | 32.438 | 3.462 | 1.806 | 3.315 | 3.49 | 1.458 | 233.09 | 32.211 | 41.049 |
| 0.003         | 31.213 | 28.307 | 30.909 | 31.229 | 3.963 | 2.328 | 3.699 | 3.983 | 1.513 | 239.424 | 30.909 | 40.216 |
| 0.004         | 30.276 | 27.841 | 29.931 | 30.341 | 4.232 | 2.745 | 3.927 | 4.314 | 1.414 | 245.428 | 29.931 | 40.184 |
| 0.005         | 29.557 | 27.406 | 29.163 | 29.577 | 4.487 | 3.038 | 4.102 | 4.494 | 1.583 | 252.896 | 29.163 | 40.352 |
| 0.006         | 28.926 | 27.051 | 28.473 | 28.972 | 4.622 | 3.319 | 4.2 | 4.692 | 2.014 | 256.614 | 28.473 | 41.337 |
| 0.007         | 28.361 | 26.653 | 27.921 | 28.405 | 4.739 | 3.508 | 4.274 | 4.768 | 1.382 | 265.054 | 27.921 | 40.49 |
| 0.008         | 27.919 | 26.386 | 27.434 | 27.923 | 4.854 | 3.682 | 4.382 | 4.881 | 1.387 | 265.209 | 27.434 | 40.573 |
| 0.009         | 27.434 | 26.064 | 26.982 | 27.544 | 4.906 | 3.849 | 4.398 | 4.993 | 1.379 | 275.593 | 26.982 | 40.415 |

Table 4. PSNR, IEF, TIME for different images corrupted by impulse noise at 30% noise density

| Images           | PSNR | IEF | TIME |
|------------------|------|-----|------|
| BABY.JPG(292X425) | 22.172 | 23.076 | 21.591 | 23.973 | 16.694 | 23.199 | 14.524 | 24.995 | 1.335 | 98.674 | 11.75 | 24.736 |
| CAMERAMAN.BMP(256X256) | 20.698 | 19.826 | 20.352 | 21.418 | 11.022 | 8.875 | 10.135 | 12.821 | 0.995 | 116.883 | 7.178 | 11.66 |
| BARBERA.TIF(512X512) | 21.038 | 21.327 | 20.041 | 21.147 | 10.917 | 10.075 | 8.722 | 11.239 | 1.38 | 457.713 | 14.24 | 62.22 |
| PEPPER.BMP(512X512) | 10.588 | 22.864 | 21.651 | 23.667 | 2.22 | 13.634 | 13.133 | 20.539 | 1.777 | 461.69 | 14.136 | 63.651 |
| GIRL.JPG(600X900) | 10.232 | 23.65 | 21.753 | 23.613 | 11.907 | 17.817 | 21.753 | 23.614 | 1.918 | 87.308 | 21.753 | 23.65 |
Table 5. PSNR, IEF, TIME for different images corrupted by zero mean Gaussian noise for variance 0.9

| IMAGES                | PSNR   | IEF   | TIME |
|-----------------------|--------|-------|------|
|                       | SMF    | TDF   | CWF  | PA    | SMF    | TDF   | CWF  | PA    | SMF     | TDF     | CWF     |
| BABY.JPG (292X425)    | 27.74  | 27.82 | 27.34 | 27.917 | 4.952  | 3.964  | 4.489 | 5.111 | 1.354  | 91.194  | 19.62  | 27.116 |
| CAMERAMAN.BMP (256X256) | 24.361 | 21.778 | 25.651 | 24.427 | 2.246  | 1.266  | 2.15  | 2.292 | 0.017  | 82.554  | 25.199 | 11.245 |
| BARBERA.TIF (512X512) | 23.246 | 4.642 | 23.316 | 23.287 | 1.875  | 0.988  | 1.892 | 1.892 | 1.472  | 614.141 | 14.045 | 60.599 |
| PEPPER.BMP (512X512)  | 20.657 | 25.566 | 26.669 | 27.065 | 1.128  | 2.148  | 4.043 | 4.742 | 1.656  | 136.334 | 14.281 | 62.035 |
| GIRL.JPG (600X900)    | 20.839 | 26.973 | 32.285 | 27.366 | 2.116  | 2.778  | 1.939 | 4.584 | 1.782  | 211.644 | 69.512 | 79.246 |

Table 6. PSNR, IEF, and TIME for different images corrupted by speckle noise for variance 0.8%

| IMAGES                | PSNR   | IEF   | TIME |
|-----------------------|--------|-------|------|
|                       | SMF    | TDF   | CWF  | PA |
| BABY.JPG (292X425)    | 28.53  | 28.446 | 28.153 | 28.623 | 2.784  | 2.504  | 3.007 | 3.021 | 1.346  | 88.697  | 8.813  | 25.755 |
| CAMERAMAN.BMP (256X256) | 25.934 | 22.345 | 26.308 | 26.979 | 0.864  | 0.4    | 0.943 | 0.874 | 0.969  | 40.015  | 11.5   | 17.61 |
| BARBERA.TIF (512X512) | 24.523 | 24.969 | 24.842 | 24.563 | 0.52   | 0.42   | 0.558 | 0.522 | 1.157  | 166.657 | 23.062 | 69.656 |
| PEPPER.BMP (512X512)  | 30.407 | 27.271 | 30.342 | 30.396 | 2.345  | 0.664  | 2.316 | 2.341 | 1.593  | 164.657 | 24.719 | 70.515 |
| GIRL.JPG (600X900)    | 29.147 | 31.125 | 32.607 | 32.689 | 1.082  | 1.859  | 0.153 | 1.893 | 2.095  | 166.673 | 24.453 | 78.261 |

Table 7. PSNR, IEF, TIME for LENA.GIF, GIRL.JPG and BABY.JPG images corrupted by 20% impulse noise plus zero mean 0.9% variance Gaussian noise

| IMAGES                | PSNR   | IEF   | TIME |
|-----------------------|--------|-------|------|
| LENA.GIF (512X512)    | 24.599 | 16.596 | 4.14 | 12.14 | 9.238  | 5.056  | 24.145 | 17.386 | 3.699 |
| GIRL.JPG (600X900)    | 22.631 | 10.569 | 39.798 | 22.51 | 11.413 | 60.134 | 22.676 | 12.6072 | 21.484 |
| BABY.JPG (292X425)    | 23.946 | 14.854 | 242.255 | 24.42 | 13.971 | 419.09 | 24.92 | 18.48 | 211.781 |
| PA                    | 24.704 | 17.015 | 113.295 | 24.643 | 18.591 | 183.885 | 25.074 | 20.999 | 66.235 |

Table 8. PSNR, IEF, TIME for BARBARA.TIF, PEPPER.BMP, CAMERAMAN.BMP images corrupted by 20% impulse noise plus zero mean 0.9% variance Gaussian noise

| IMAGES                | PSNR   | IEF   | TIME |
|-----------------------|--------|-------|------|
| BARBARA.TIF (512X512) | 21.593 | 8.399 | 3.378 | 12.515 | 1.812  | 3.378 | 22.023 | 9.693  | 4.169 |
| PEPPER.BMP (512X512)  | 21.134 | 7.509 | 29.632 | 22.289 | 10.213 | 29.632 | 21.514 | 8.754  | 36.738 |
| CAMERAMAN.BMP (256X256) | 21.962 | 7.547 | 286.573 | 23.46 | 10.437 | 286.573 | 22.173 | 7.005 | 311.423 |
| PA                    | 21.987 | 8.565 | 150.454 | 24.244 | 16.011 | 150.454 | 22.274 | 10.59 | 155.469 |
K. VASANTH AND S. KARTHIK: PERFORMANCE ANALYSIS OF A MODIFIED DECOMPOSITION FILTER FOR NON IDENTICAL NOISES

Fig.4. Noise density versus IEF for various filters for Lena image corrupted by impulse noise

Fig.5. Noise density versus TIME for various filters for Lena image corrupted by impulse noise

Fig.6. Variance versus PSNR for various filters for Lena image corrupted by Gaussian noise

Fig.7. Variance versus IEF for various filters for Lena image corrupted by Gaussian noise

Fig.8. Variance versus TIME for various filters for Lena image corrupted by Gaussian noise

Fig.9. Variance versus PSNR for various filters for Lena image corrupted by Speckle noise
Fig. 10. Variance versus IEF for various filters for Lena image corrupted by Speckle noise.

Fig. 11. Variance versus TIME for various filters for Lena image corrupted by Speckle noise.

Fig. 12. PSNR for various filters applied over different images corrupted by 30% impulse noise.

Fig. 13. IEF for various filters applied over different images corrupted by 30% impulse noise.

Fig. 14. TIME for various filters applied over different images corrupted by 30% impulse noise.

Fig. 15. PSNR for various filters applied over different images corrupted by zero mean and 0.9% variance Gaussian noise.
Fig. 16. IEF for various filters applied over different images corrupted by zero mean and 0.9% variance Gaussian noise

Fig. 17. TIME for various filters applied over different images corrupted by zero mean and 0.9% variance Gaussian noise

Fig. 18. PSNR for various filters applied over different images corrupted by 0.8% variance Speckle noise

Fig. 19. IEF for various filters applied over different images corrupted by 0.8% variance Speckle noise

Fig. 20. TIME for various filters applied over different images corrupted by 0.8% variance Speckle noise

Fig. 21. PSNR for various filters applied over different images corrupted by 20% impulse noise, 0.9% variance Gaussian noise
Fig. 22. IEF for various filters applied over different images corrupted by 20% impulse noise, 0.9% variance Gaussian noise.

Fig. 23. TIME for various filters applied over different images corrupted by 20% impulse noise, 0.9% variance Gaussian noise.

Fig. 24. Cameraman.bmp, Barbara.tif, lena.gif (a) original image (b) impulse noise affected from by 30% (c) images restored by SMF (d) images restored from by TDF (e) images restored by CWF (f) images restored by proposed algorithm.

Fig. 25. Cameraman.bmp, Barbara.tif, lena.gif (a) original image (b) Zero mean and 0.9% variance Gaussian noise (c) images restored by SMF (d) images restored from by TDF (e) images restored by CWF (f) images restored by proposed algorithm.
Fig. 26. Cameraman.bmp, Barbara.tif, lena.gif (a) original image (b) 0.8% variance Speckle noise (c) images restored by SMF (d) images restored from by TDF (e) images restored by CWF (f) images restored by proposed algorithm

Fig. 27. Barbara.tif, pepper.bmp, lena.gif, Cameraman.bmp, baby.jpg, girl.jpg (a) original image (b) Impulse noise 20% plus zero mean 0.9% variance Gaussian noise (c) images restored by SMF (d) images restored from by CWF (e) images restored by TDF (f) images restored by proposed algorithm
5. CONCLUSION

From the exhaustive experiments, conducted for different noise types for different images for different median filters, we conclude that, the highest PSNR (dB) and IEF is not obtained for PA for different images and for different noise type. However, on overall basis, i.e., in an average sense, PA gives good performance for low density impulse noise up to 20%, zero mean 0.9% variance Gaussian noise removal. When compared to their class of decomposition filters such as TDF in specific, the PA exhibits better performance for Salt & Pepper noise removal up to 30% and reduces smaller proportion of zero mean 0.9% variance Gaussian noise. The proposed filter also exhibits good noise removal up to 0.8% speckle noise. In our method, time complexity of Threshold Decomposition is removed by considering the pixel intensity itself as threshold. Hence, the proposed method shows good performance with fewer complexities. The Proposed algorithm has good average computation time such that it’s twice faster in comparison to TDF and exhibits optimum computation speed when compared with other filters.

REFERENCES

[1] K.Vasanth and S.Karthik, 2009, “A New class of decomposition algorithm for the reduction of low density impulse noise”, International conference on ARTCOM2009, India, pp.203-207.

[2] N. D. Sidiropoulos, J. S. Baras and C A Berenstein, 1994, “Optimal filtering of digital binary images corrupted by union/intersection noise”, IEEE Trans. on Image Processing, Vol.3, No.4, pp.382-403.

[3] A.K.Jain, 1989, “Fundamentals of digital image processing,” Prentice-Hall Inc., Englewood Cliffs, New Jersey.

[4] Motwani. M.C., Gadiya. M.C., Motwani. R.C.and Harris Jr. F.C., 2004, “Survey of Image Denoising Techniques”, Proceedings of GSPx, Santa Clara, CA.

[5] G.R.Arce, N.C.Gallagher, and T.Nodes, 1986, “Median filters: Theory and applications”, in Advances in Computer Vision and Image Processing, T.Huang, Ed.Greenwich.

[6] I.Pitas and A.N.Venetsanopoulos, 1990, “Nonlinear Digital Filters: Principles and applications”, Boston, MA:Kluwer.

[7] A. Bovik, 2000, Handbook of Image and Video Processing, Academic Press.

[8] Amandeep Kaur, Karamjeet Singh, 2010, “Speckle noisereduction by using wavelets”, NCCT’10, pp.198-203.