ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2010.0001


# VIDEO DENOISING USING SWITCHING ADAPTIVE DECISION BASED ALGORITHM WITH ROBUST MOTION ESTIMATION TECHNIQUE

**V. Jayaraj[1], D. Ebenezer[2] and S. Manikandan[3]**
[1]*Department of Electronics and Communication Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, India*
E-mail: jayarajmevlsi@gmail.com[1]
[2]*Department of Electronics and Communication Engineering, RMK Engineering College, Chennai, India*
[3]*DRDO, India*
E-mail: mani567@gmail.com

***Abstract***
***A Non-linear adaptive decision based algorithm with robust motion estimation technique is proposed for removal of impulse noise, Gaussian noise and mixed noise (impulse and Gaussian) with edge and fine detail preservation in images and videos. The algorithm includes detection of corrupted pixels and the estimation of values for replacing the corrupted pixels. The main advantage of the proposed algorithm is that an appropriate filter is used for replacing the corrupted pixel based on the estimation of the noise variance present in the filtering window. This leads to reduced blurring and better fine detail preservation even at the high mixed noise density. It performs both spatial and temporal filtering for removal of the noises in the filter window of the videos. The Improved Cross Diamond Search Motion Estimation technique uses Least Median Square as a cost function, which shows improved performance than other motion estimation techniques with existing cost functions. The results show that the proposed algorithm outperforms the other algorithms in the visual point of view and in Peak Signal to Noise Ratio, Mean Square Error and Image Enhancement Factor.***

***Keywords:***
***Impulse Noise, Gaussian Noise, Motion Estimation, Median Based Filter, Minimum Median Square Error***

## 1. INTRODUCTION

Videos are often corrupted by artifacts, impulses, and Gaussian noise while transmitting over channels and due to bad reception of television pictures [1] – [3]. Some noise sources are located in a camera and become active during image acquisition under bad lightning conditions. Noise reduction in image sequences is used for various purposes, e.g. for visual improvement in video surveillance. It is achieved through some form of linear or non-linear operation on correlated picture elements. In the recent past, a number of non-linear techniques for video processing have been proposed and were proved superior to linear techniques. It is well known that linear filtering techniques fail when the noise is non-additive and are not effective in removing impulse noise This has lead the researchers to the use of non-linear signal processing techniques. A class of widely used non-linear digital filters is median filters. Median filters are known for their capability to remove impulse noise as well as preserve the edges. The main drawback of a standard median filter (SMF) is that it is effective only for low noise densities. At high noise densities, SMFs often exhibit blurring for large window sizes and insufficient noise suppression for small window sizes.

The filters designed for image processing are required to yield sufficient noise reduction without losing the high-frequency content of image edges. However, most of the median filters operate uniformly across the image and thus tend to modify both noise and noise-free pixels. Consequently, the effective removal of impulse often leads to images with loss of fine details. It is desirable that the filtering should be applied only to corrupted pixels while leaving uncorrupted pixels intact. Inclusion of a noise-detection process to discriminate between uncorrupted pixels and the corrupted pixels prior to applying nonlinear filtering significantly improves the image quality. Standard Median Filter [4] Adaptive Median Filter [5], [7], Progressive Switching Median Filter (PSMF) [6], Decision Based Median Filter [8], or switching median filters have been proposed with this objective. Possible noisy pixels are identified and replaced by using median value or its variant while leaving uncorrupted pixels unchanged. The performance of AMF is good at lower noise density levels, due to the fact that there are only fewer corrupted pixels that are replaced by the median values. At higher noise densities, the number of replacements of corrupted pixel increases considerably; increasing window size will provide better noise removal performance; however, the corrupted pixel values and replaced median pixel values are less correlated. As a consequence, the edges are smeared significantly. The main drawback of decision-based or switching median filter is that defining a robust decision measure is difficult, because the decision is usually based on a predefined threshold value. An additional drawback is that the noisy pixels are replaced by some median value in their vicinity without taking into account local features such as possible presence of edges. Hence, details and edges are not recovered satisfactorily, especially when the noise level is high. Robust Estimation Algorithm's (REA) [9] are effective in removing impulse noise, but their performance is poor in removing Gaussian noise. Alpha-Trimmed mean filter (ATMF) (11) is useful for the multiple types of noise such as the combination of salt-and-pepper and Gaussian noise. But its efficiency is low. Alpha-Trimmed Midpoint (ATMP) [10] filter is recognized as a good compromise between the midpoint and median smoothers. It is effective in suppressing uniform noise when the trimming parameter, alpha value is close to zero, but destroys the image boundaries. Mean filter [1] compute the average value of the corrupted image in the area defined by the window. Noise is reduced as a result of blurring. Wiener filter [1] is a low pass filter. It filters a grayscale image that has been degraded by constant power additive noise. Wiener uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. Wiener filter works best when the noise additive noise, such as Gaussian noise. Its performance is poor in removing impulse noise. Gaussian noise removal involves smoothing pixels inside distinct regions of an image with the side effect of blurring edges and details significantly To solve this problem, non-linear methods, that utilize local measures such as weighting to smooth the noise in the image, have been proposed, e.g., the bilateral filter by Tomasi and Manducci [12]. Many Denoising algorithms are effective in removing salt & pepper noise or gaussian noise separately[21]. But those algorithms are very complex and time consuming. Their performance in mixed noise removal is also poor [13] – [20]. To overcome the above drawbacks, a new algorithm is proposed in this paper which will be explained in the consecutive section. If the noise is detected to be Gaussian in nature based on the variance value, then alpha-trimmed mean filter is used else if the pixel is corrupted by salt and pepper noise, the proposed adaptive decision based filter is used.

Motion estimation (ME) is a process to estimate the pixels of the current frame from reference frames. Over the last two decades, many fast motion estimation algorithms have been proposed to give a faster estimation with similar block distortion compared to Full Search (FS). The most well known fast Block Matching Algorithms (BMA) are the Three-Step Search (3SS), the New Three-Step Search (N3SS), the Four-Step Search (4SS), the Diamond Search (DS), the Cross-Diamond Search (CDS) and Small Cross-Diamond search [21] – [27]. As the characteristic of center-biased Motion Vector Distribution (MVD) which inspired many fast BMA in last decade, more than 80% of the blocks can be regarded as stationary or quasi-stationary blocks, i.e. most of the motion vectors are enclosed in the neighboring blocks and introduced a halfway-stop technique to achieve crucial speedup for stationary and quasi-stationary blocks. Four-Step Search also exploits the center-biased properties of motion vectors distribution by using halfway-stop techniques and smaller square search pattern compared to Three-Step Search central 5x5 (blocks) area. This center-based characteristic can even be found in the fast-motion sequences. In this paper, to fit cross-center-biased motion vector distribution property of the most real world sequences, a novel fast Block Matching Algorithm called Improved Cross-Diamond Search algorithm (ICDS) with new cost function termed as Minimum Median of Square Error (MMedSE) [23]- [24] is proposed. It uses a small cross-shaped search patterns in the first two steps and speed results in higher the motion estimation of stationary and quasi-stationary blocks.

First, we review the existing denosing methods for salt & pepper noise and gaussian noise in section 1. In section 2, we propose a new method for the calculation of the noise variance and choosing appropriate filter based on the noise variance. In this section we propose a new adaptive decision based median filter for the removal of salt and pepper noise. In section 3, we introduce an improved cross diamond search block matching algorithm for motion estimation. In section 4 the proposed method is compared with several other filters used for the removal of salt & pepper and gaussian noise.

## 2. ADAPTIVE DECISION-BASED ALGORITHM

The proposed adaptive decision based algorithm consists of two stages. Stage1 is spatial filtering which includes calculation of noise variance and choosing of filtering techniques depending on the noise variance. Stage2 is temporal processing, where the robust motion estimation technique Improved Cross Diamond Search algorithm is used. Here the cost function is Minimum Median Square Error. The block diagram of the proposed algorithm is given in Fig.1.

### 2.1 STAGE 1(SPATIAL FILTERING)

In this step, the noisy video is converted to frames and the frames are used for the spatial processing. The spatial processing includes the calculation of the noise variance and choosing appropriate filter based on the noise variance to remove the noise in the frames.

#### *2.1.1 Pseudo code for Noise Variance calculation and filter selection:*

Let $n_{i,j}$ be a image frame in a video sequence corrupted salt & pepper and gaussian noise. Let R represents the number of rows in the image and C represents number of columns.

```
for i= 1 to R
 for j= 1 to  C
  set the window size W=3. Assume that pixel in the processing
window are stored in S i,j.
   h= histogram (S i,j )   // find the histogram of the
                              current window
   h_max = maximum (h)      // maximum value in h
   h_size= size (h_max)     // find the size of h
     for k = 1 to h_size
        if h(k) = = h_max
            index = k       // find the index value
        end
     end
     σ = index * 0.2661 – 0.782
        if  0 < σ  < ζ
process for Gaussian noise using alpha trimmed mean filter
          else
 processing using adaptive decision based median filter for salt
and pepper noise
            end
    end
end
```

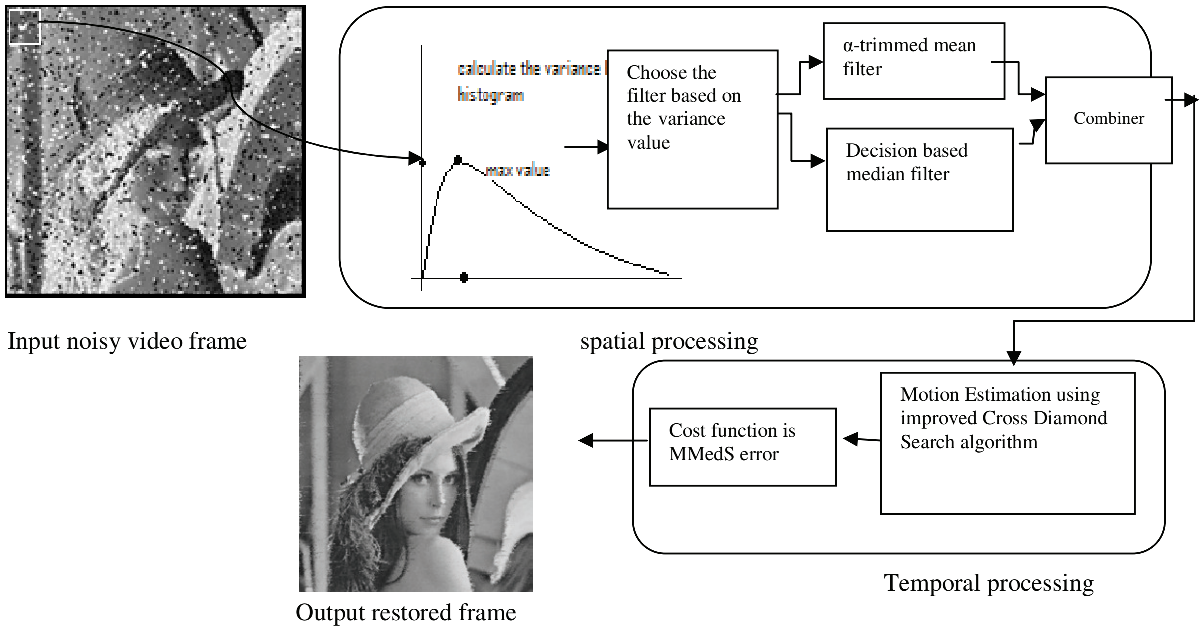



Fig.1. Block diagram of the proposed algorithm

A noise threshold ζ is chosen which normally varies for different images. If the estimated noise variance is greater than this threshold, median based filter is used. Otherwise alpha-trimmed mean filter is used. After doing many simulations on variety of images, it is found that for $\zeta = 20$ better result is obtained.

***2.1.2. New Mean/Median based Filter to remove salt and pepper noise:***

The proposed median based filter algorithm is as follows:

Step 1: A 2-D window "$S_{xy}$" of size 3 × 3 is selected with the pixel to be processed "P(x,y)" as centre.

Step 2: Determine the median ($P_{med}$) of the 9 pixels inside this window.

Step 3: P(x,y) is an uncorrupted pixel if $0 < P(x,y) < 255$, and is left unchanged.

Step 4: If P(x,y) is a corrupted pixel, then it is processed as follows.

Step 4(a): If $P_{med}$ is an uncorrupted pixel, then P(x,y) is replaced with $P_{med}$.

Step 4(b): If Pmed is a corrupted pixel, then another window of size 5 × 5 is selected with P(x,y) as centre and the median ($P_{med5}$) of these 25 pixels are determined.

Step 4(c): If $P_{med5}$ is an uncorrupted pixel, then P(x,y) is replaced with $P_{med5}$.

Step 4(d) : If $P_{med5}$ is a corrupted pixel, then window $S_{xy}$ is again considered and the number of uncorrupted pixels (Ns) in window $S_{xy}$ is counted.

Step 4(e): If Ns is even, then P(x,y) is replaced with mean of the uncorrupted pixels in the window $S_{xy}$.

Step 4(f): Else P(x,y) is replaced with median of the uncorrupted pixels in the window $S_{xy}$.

Step 5: Repeat steps 1 to 4 for all the pixels in the image.

***2.1.3. Alpha-Trimmed Mean Filter to remove Gaussian noise:***

Let X={x(i), x(i-1),·········x(i-n+1)}, where n= 2N+1 be a set of n sample signal values observed in a window, $W_i$. If these values are arranged in ascending order of their amplitude, the order statistics result is $x_1(i) \le x_2(i) \le x_3(i) \le \cdots. \le x_n(i)$, where $x_1(i)$ is the minimum , $x_n(i)$ is the maximum and $x_{N+1}(i)$ is the median of the above set of signal values. The output of the alpha-trimmed mean filter, is y(i,α)

$$y_n(i;\propto) = \frac{1}{n-2[\propto n]}\sum_{j=[\propto n]+1}^{n-[\propto n]} x_{(j)}(i) \qquad (1)$$

where $0 \le \alpha < 0.5$. $\alpha$ indicates the percentage of the trimmed samples. Therefore, the alpha-trimmed mean filter performs like a median filter when is close to 0.5, and moving average filter when $\alpha$ is close to 0. If we drop the time index and denote the trimmed-mean filter by m (α), then the moving average filter is m (0). . Although α never gets equal to 0.5, for simplicity we will represent median filter as m (0.5) since α is very close to 0.5 for this filter.

## 2.2 STAGE 2(TEMPORAL FILTERING)

The stage II is the temporal filtering where the process is done between the frames. The temporal median filter smoothes out sharp transitions in intensity at each pixel position; it not only denoise's the whole frame and removes blotches but also helps in stabilizing the illuminating fluctuations. This is beneficial both visually and for motion estimation performance in the next stage. Temporal median filtering removes the temporal noise in the form of small dots and streaks found in some videos. In this approach, dirt is viewed as a temporal impulse (single-frame incident) and hence treated by inter-frame processing by taking into account at least three consecutive frames. A pixel is flagged as dirt if the corresponding absolute differences between the current frame and each of the previous and next frames were high. Generally speaking, although motion compensation can potentially become an essential component of a dirt detection algorithm, it is well known that it does not degrade gracefully when motion estimation fails and may thus generate unpredictable results. For such a reason temporal filtering may be regarded as a useful tool as a complement to motion compensated approaches. In this robust block matching algorithm is used with least median square error as the cost function.

## 3. NEW ROBUST BLOCK MATCHING ALGORITHM

The idea behind block matching is to divide the current frame into a matrix of macro blocks. The blocks are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to ‘p’ pixels on all fours sides of the corresponding macro block in previous frame. This ‘p’ is called as the search parameter.

### 3.1 IMPROVED CROSS DIAMOND SEARCH (ICDS) ALGORITHM

In this section we first describe search pattern used in the algorithm and later the search path strategy will be explained. The Search Patterns used in ICDS is explained below.

The search-point configuration used in the ICDS is divided in 2 different shapes: Cross-shaped pattern and diamond-shaped pattern. Fig.2 (a) shows the Small Cross-Shaped Pattern (SCSP) and the Large Cross-Shaped Pattern (LCSP). The same search pattern from DS: Small Diamond-Shaped Pattern (SDSP) and Large Diamond-Shaped Pattern (LDSP) are shown in Fig.2 (b).

### 3.2 ALGORITHM FOR IMPROVED CROSS DIAMOND SEARCH

Step 1 (Starting - Small Cross Shape Pattern SCSP): A MMedSE is found from the 5 search points of the SCSP (Small Cross-Shaped Pattern) located at the center of search window. If the MMedSE point occurs at the center of the SCSP (0,0), the search stops (First Step Stop); otherwise, go to Step 2.

Step 2 (SCSP): With the vertex (MMedSE point) in the first SCSP as the center, a new SCSP is formed. If the MMedSE point occurs at the center of this SCSP, the search stops (Second Step Stop); otherwise go to Step 3.

Step 3 (Guiding - Large Cross Shape Pattern - LCSP): With the MMedSE point found in the previous step, a Large Cross Shape Pattern is formed, in which the step is trying to guide the possible correct direction for the subsequent steps. And then go to Step 4.

Step 4 (Diamond Searching): A new Large Diamond Search Pattern LDSP is formed by repositioning the MMedSE found in previous step as the center of the LDSP. If the new MMedSE point is at the center of the newly formed LDSP, then go to Step 5 for converging the final solution; otherwise, this step is repeated.

Step 5 (Ending – Converging step): With the MMedSE point in the previous step as the center, a SDSP (Small Diamond- Shaped Pattern) is formed. Identify the new MMedSE point from the SDSP, which is the final solution for the motion vector.

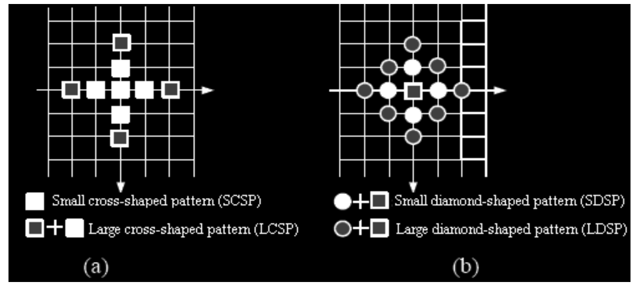



Fig.2. Search Patterns used in the improved-cross-diamond search

### 3.3 ANALYSIS OF ICDS ALGORITHM

To compare the Cross Diamond Search(CDS) and the Diamond Search(DS), the main improvement of this algorithm is the speed performance (the number of searching point). Improved Cross Diamond Search (ICDS) reduces the number of search points significantly if there is stationary block or quasi-stationary blocks. To fit the cross-center-biased MV distribution characteristics, it provides more chance to save up the searching points for motion vectors. In Fig.3, it shows 4 typical examples of ICDS and each candidate point is marked with the corresponding step number. Fig.3 shows two halfway-stop examples. The ICDS only takes 5 (first step stop) and 8 (second step stop), whereas the CDS took 9 and 11 search points, and the DS took 13 search points for searching the same block respectively. Another two search paths for r > 1 are shown in Fig.3 Due to the chance of being trapped to local minimum if the algorithm is keeping a SCSP in first and second search step, a guiding step in Step 3 of NCDS is trying to guide a possible correct direction by using a larger search pattern when r > 1. Thus, we employed LCSP step to avoid the algorithm being trapped by local minimum, which will influence the distortion performance. After step 3, the subsequent steps will be exactly the diamond search.

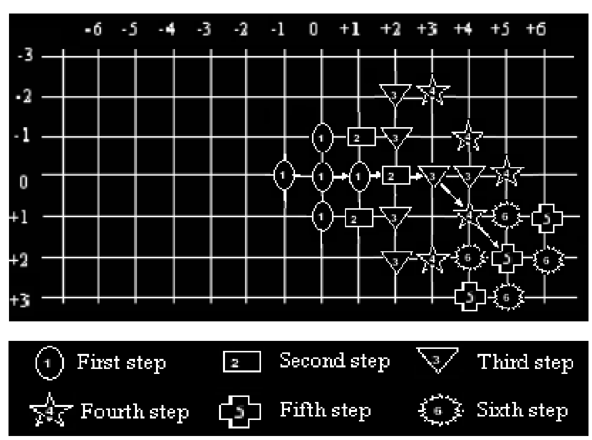



Fig.3. Examples of the ICDS: (a) first-step-stop with MV(0,0), Second-step-stop with MV(1,0), an unrestricted search path for MV(-2,-2) and (5,2) respectively.

### 3.4 ALGORITHM FOR MINIMUM MEDIAN SQUARE ERROR (MMedSE)

Compute the square error for the 25 elements in the macro block as

$(C(x,y) – R(x,y))^2 \quad ; \quad 1 \le x \le 5, 1 \le y \le 5$ (2)

where, C(x,y) - pixel in current frame, R(x,y) – pixel in reference frame,

Find the median of the 25 elements. Similarly, compute median of square error for all the search points. The minimum of all these search points gives the Least Median of Square Error (MMedSE) value.

The steps of MMedSE to find the global motion can be outlined as follows:

*Step 1:* For each pixel at [x y]' in the previous frame, obtain the corresponding point at [x’ y’]′ in current frame. The correspondence can be found from the ARPS block-matching algorithm. The estimated motion vector is

$[u,v]^t = [x’,y’]^t – [x,y]^t$.

*Step 2:* Select 3 pixels $[x_1,y_1]^t$, $[x_2,y_2]^t$, $[x_3,y_3]^t$ from the previous frame. From their corresponding points in the current frame, find a set of GMP, say Tk**,** using:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1' & y_1' \\ x_2' & y_2' \\ x_3' & y_3' \end{bmatrix} = \begin{bmatrix} a_1 & a_4 \\ a_2 & a_5 \\ a_3 & a_6 \end{bmatrix} \quad (3)$$

*Step 3:* For each pixel at $[x\ y]^t$ in the previous frame, use the ***Tk*** to estimate the corresponding point [$x_e$’ $y_e$’]' in the current frame by *Step 1* and so the estimated global motion vector $[u_e,v_e]^t$ by *Step 2.*

$$e = \sqrt{[(u_e - u)^2 + (v_e - v)^2]} \quad (4)$$

*Step 4:* For each $[x,y]^t$ evaluate the motion vector error ***e,*** i.e. the Euclidean distance between the estimated motion vector $[u,v]^t$ and estimated global motion vector $[u_e,v_e]^t$.

*Step 5:* Find the median of the motion vector errors by sorting them out.

*Step 6:* Repeat *Step 2* to *Step 5* g times to obtain g error medians.

*Step 7:* Compare the g error medians. Choose the GMP that associates with the least error median as the final solution.

Finding all possible solutions is not feasible as the number of trials g for choosing all combination of 3-pixels is very large. However, g can be reduced substantially while maintaining a high probability of selecting all 3 pixels from the background. According to *Step 3*, the probability of successfully selecting 3 background pixels is

$s = 1-(1-a^3)^g$ where a is the fraction of background pixels in the frame.

## 4. RESULTS AND DISCUSSIONS

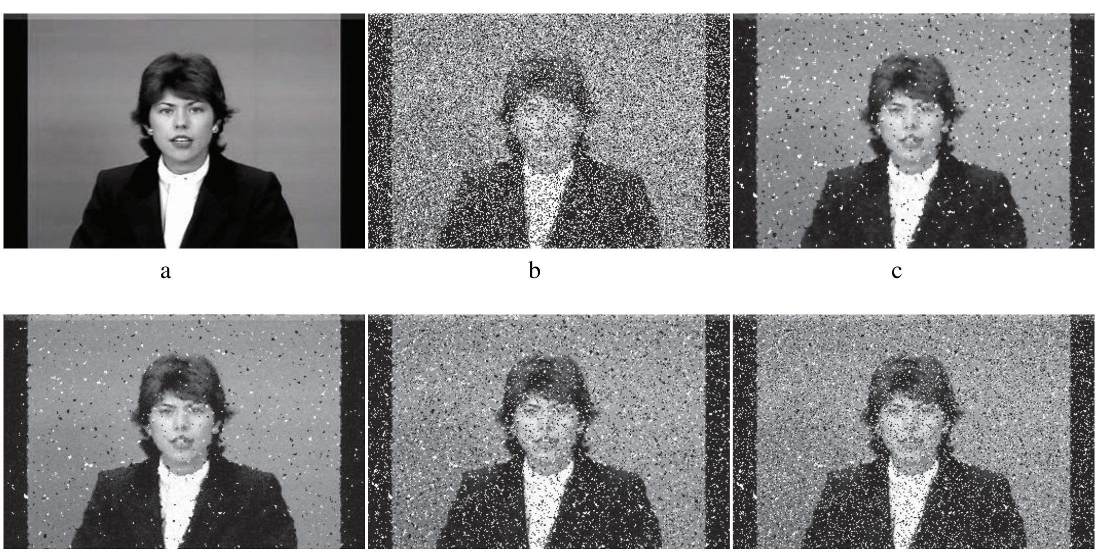
All Simulations are carried out to verify the noise removing capability of the proposed algorithm and the results are compared with several exiting filters. Video sequences are mixed with noise at different densities for evaluating the performance of the algorithm. A quantitative comparison is performed between several filters and the proposed algorithm in terms of Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and Image Enhancement Factor (IEF). Our method produced results superior to other methods in both visual image quality and quantitative measures. The performance of the algorithm for various images at different noise is studied and results are shown in Fig.4 – Fig. 15. The quantities for comparison are defined as follows and Tables 1 – 5 display the quantitative measures. Let N be the corrupted image; r be original image; M X N is size of image; x be the restored image.

$$PSNR = 10log10(255^2/MSE) \quad (5)$$

$$MSE = \frac{1}{MN}\sum_{i-1}^{M}\sum_{j-1}^{N}\left(r_{ij} - x_{ij}\right)^2 \quad (6)$$

$$IEF = \frac{\sum_{i-1}^{M}\sum_{j-1}^{N}\left(n_{ij}-r_{ij}\right)^2}{\sum_{i-1}^{M}\sum_{j-1}^{N}\left(x_{ij}-r_{ij}\right)^2} \quad (7)$$

The simulation results show that our approach provides a visually appealing output.



a b c

d e f

g h i

j k l

m

Fig.4. Results of different filters for frame 16 of Claire video sequence. (a) Original Sequence. (b) Image sequence corrupted by 30% salt & pepper noise and σ= 20. (c) Output of SMF. (d) Output of PSMF (e) Output of AMF. (f) Output of RAMF. (g) Output of DBA. (h) Output of REMF. (i) Output of ATMF. (j) Output of Mean filter (k) Output of Wiener filter. (l) Output of Bilateral. (m) Output of PA

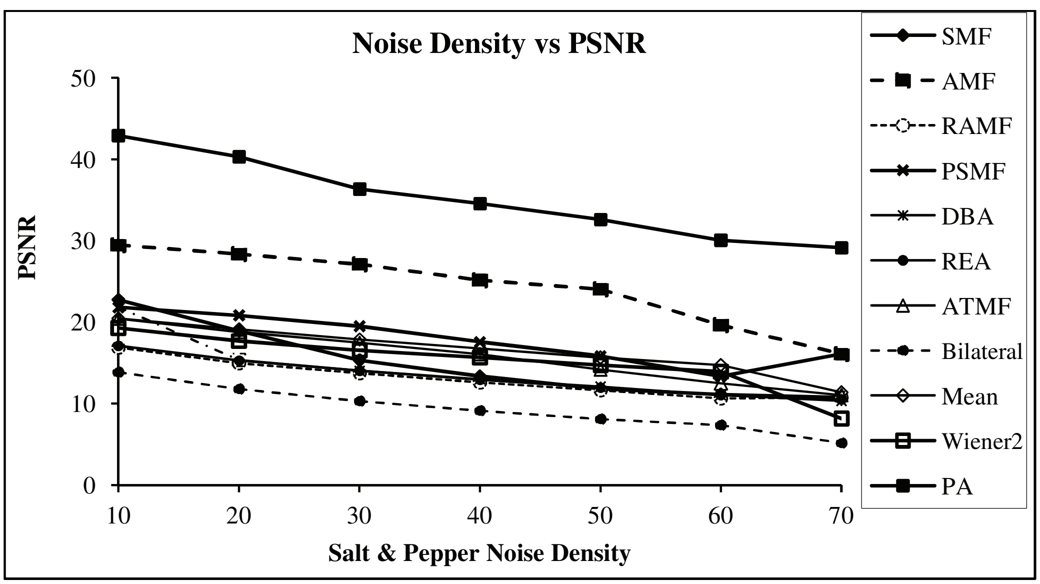



Fig.5. Comparison graph of PSNR at different noise densities for frame 16 of Claire video sequence

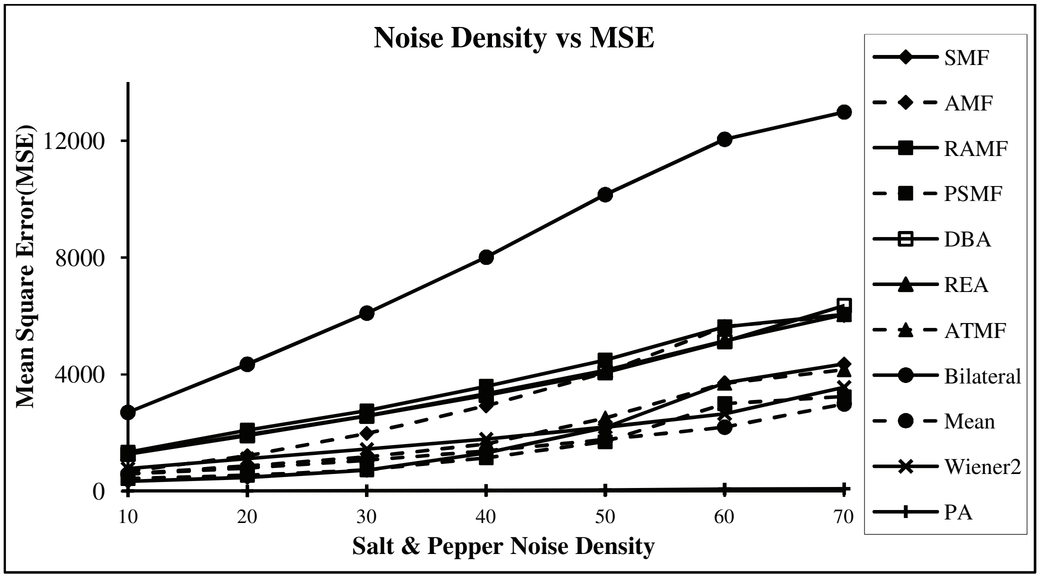



Fig.6. Comparison graph of MSE at different noise densities for frame 16 of Claire video sequence

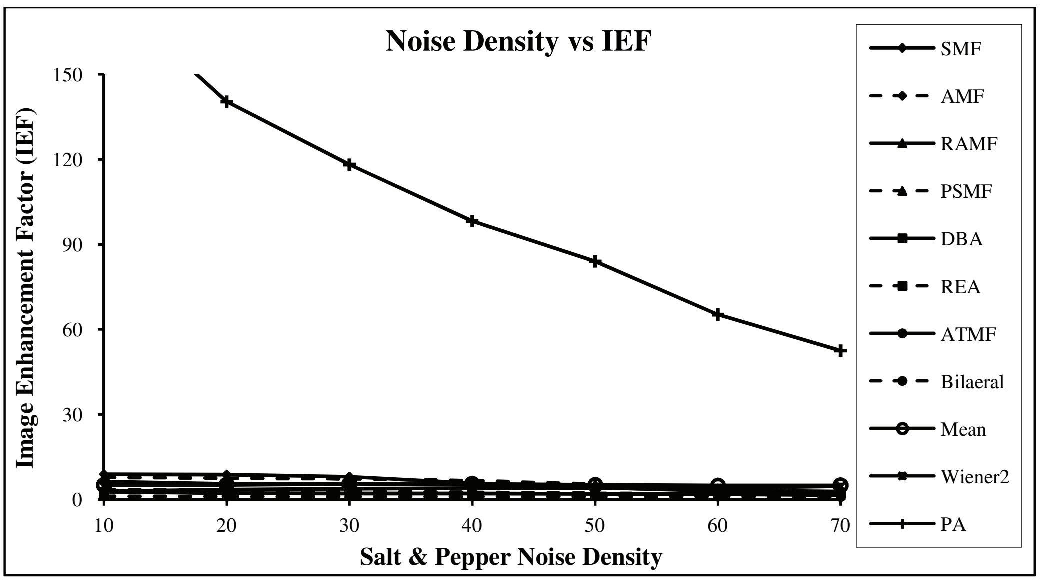



Fig.7. Comparison graph of IEF at different noise densities for frame 16 of Claire video sequence

Table.1. Noise Density vs PSNR for frame 16 of Claire video sequence

| σ =20 Salt & Pepper Noise density (%) | Peak Signal to Noise Ratio (PSNR) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SMF | AMF | RAMF | PSMF | DBA | REA | ATMF | Bilateral | Mean | Wiener2 | PA |
| 10 | 22.75 | 29.48 | 16.88 | 21.815 | 21.815 | 17.084 | 20.426 | 13.832 | 20.4559 | 19.2695 | 42.92 |
| 20 | 18.93 | 28.36 | 14.933 | 20.813 | 15.336 | 15.279 | 18.791 | 11.755 | 19.1349 | 17.6905 | 40.28 |
| 30 | 15.3 | 27.1 | 13.724 | 19.485 | 14.036 | 14.009 | 17.452 | 10.283 | 17.8801 | 16.5415 | 36.32 |
| 40 | 13.36 | 25.13 | 12.584 | 17.572 | 12.974 | 12.899 | 16.046 | 9.0969 | 16.7603 | 15.6362 | 34.57 |
| 50 | 11.82 | 24.04 | 11.616 | 15.833 | 12.054 | 11.968 | 14.16 | 8.0658 | 15.6516 | 14.7295 | 32.62 |
| 60 | 11.09 | 19.64 | 10.629 | 13.368 | 11.03 | 11.019 | 12.459 | 7.3225 | 14.7329 | 13.9093 | 30.05 |
| 70 | 10.72 | 16.1 | 10.72 | 16.1 | 10.33 | 10.521 | 11.014 | 5.151 | 11.351 | 8.151 | 29.13 |

Table.2. Noise Density vs MSE for frame 16 of Claire video sequence

| σ =20 Salt & Pepper Noise density (%) | Mean Square Error (MSE) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SMF | AMF | RAMF | PSMF | DBA | REA | ATMF | Bilateral | Mean | Wiener2 | PA |
| 10 | 338 | 674 | 1334 | 428 | 1262 | 1273 | 589 | 2690 | 585 | 769 | 3 |
| 20 | 474 | 1216 | 2088 | 539 | 1903 | 1928 | 859 | 4340 | 793 | 1106 | 6 |
| 30 | 715 | 1976 | 2758 | 732 | 2567 | 2583 | 1169 | 6091 | 1059 | 1441 | 15 |
| 40 | 1299 | 2921 | 3586 | 1137 | 3278 | 3335 | 1616 | 8006 | 1371 | 1776 | 22 |
| 50 | 2184 | 4050 | 4482 | 1697 | 4052 | 4133 | 2495 | 10151 | 1769 | 2188 | 35 |
| 60 | 3699 | 5628 | 5625 | 2994 | 5130 | 5142 | 3691 | 12046 | 2186 | 2643 | 64 |
| 70 | 4356 | 6015 | 6048 | 3242 | 6348 | 6047 | 4157 | 12987 | 2981 | 3562 | 79 |

Table.3. Noise Density vs. IEF for frame 16 of Claire video sequence

| σ =20 Salt & Pepper Noise density (%) | Image Enhancement Factor | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SMF | AMF | RAMF | PSMF | DBA | REA | ATMF | Bilateral | Mean | Wiener2 | PA |
| 10 | 8.9562 | 3.5819 | 2.7616 | 7.8578 | 2.8237 | 2.814 | 6.4283 | 1.1443 | 5.084 | 2.9823 | 179.81 |
| 20 | 8.8056 | 3.36 | 2.2725 | 7.5641 | 2.3814 | 2.3519 | 5.6247 | 1.1429 | 5.1959 | 3.571 | 140.35 |
| 30 | 8.116 | 2.988 | 2.1936 | 7.3314 | 2.2916 | 2.1763 | 5.5563 | 1.1223 | 5.5276 | 3.8955 | 118.25 |
| 40 | 5.8655 | 2.5795 | 2.0999 | 6.7268 | 2.2304 | 2.191 | 5.15 | 1.0869 | 5.5206 | 4.0641 | 98.31 |
| 50 | 4.2258 | 2.3043 | 2.0476 | 5.4145 | 2.193 | 2.1549 | 4.2178 | 1.0591 | 5.2673 | 4.0877 | 84.06 |
| 60 | 2.9965 | 1.9726 | 1.9542 | 3.6608 | 2.067 | 2.0662 | 3.3713 | 1.0388 | 5.0387 | 4.018 | 65.28 |
| 70 | 2.0453 | 1.389 | 1.514 | 2.546 | 2.001 | 2.034 | 2.871 | 1.001 | 5.0028 | 4.645 | 52.52 |

a b

Fig.8. (a) Frame 28 from Claire video sequence corrupted by 30% impulse noise and Gaussian noise of σ = 20 (b) Restored image

a b

Fig.9. (a) Frame 16 from Salesman video sequence corrupted by 30% impulse noise and Gaussian noise of σ = 20 (b) Restored image


Fig.10. (a) Frame 19 from Foreman video sequence corrupted by 30% impulse noise and Gaussian noise of σ = 20 (b) Restored image

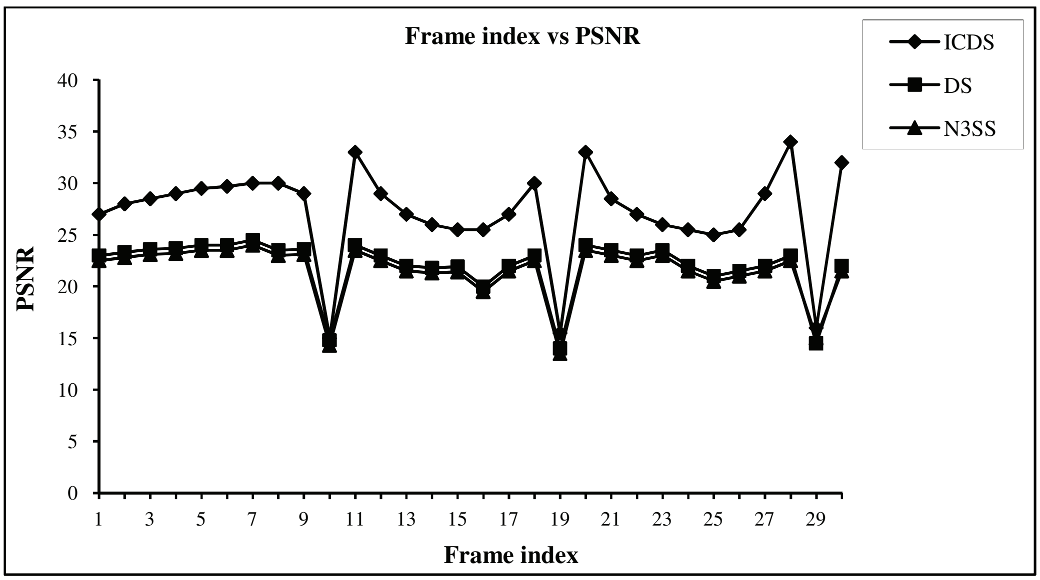



Fig.11. PSNR comparison graph of Foreman black and white video film

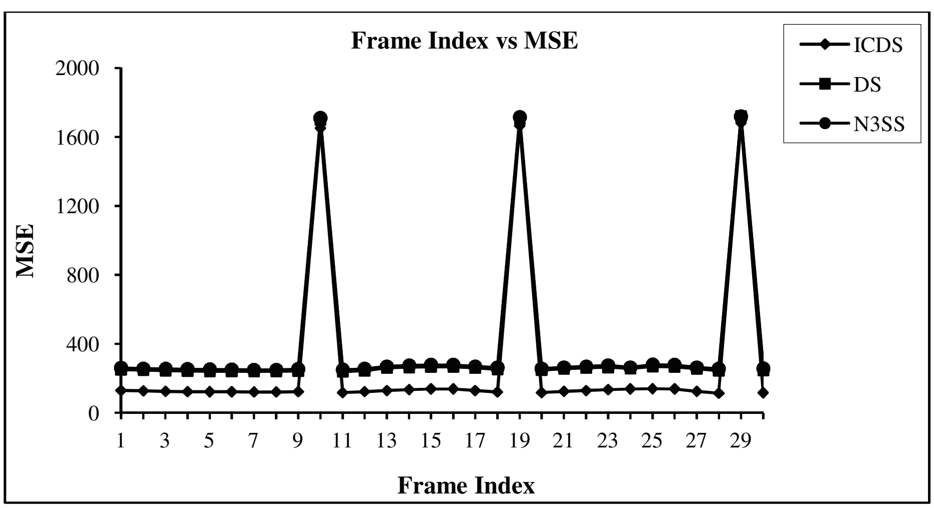


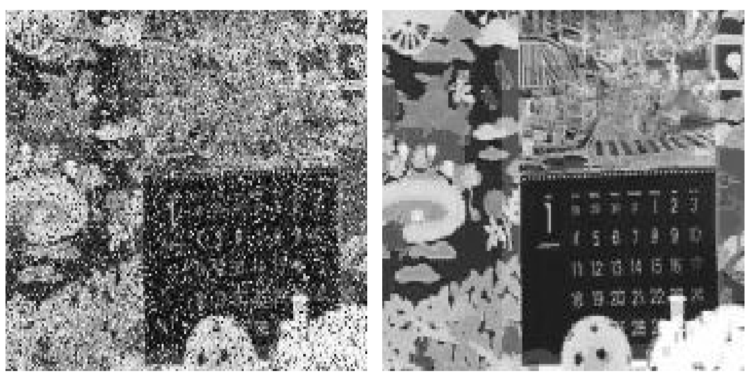

Fig.12. MSE comparison graph of Foreman black and white video film

Table.4. PSNR and MSE for N3SS, DS and ICDS for different black and white video sequences

| Black & White Noisy Video sequence | PSNR (dB) | | | MSE | | |
|---|---|---|---|---|---|---|
| | N3SS | DS | ICDS | N3SS | DS | ICDS |
| Claire | 25.327 | 25.6684 | 35.755 | 269.643 | 259.446 | 189.1647 |
| Salesman | 23.027 | 23.3278 | 30.022 | 362.767 | 332.9737 | 226.4941 |
| Foreman | 21.676 | 21.9276 | 26.4344 | 466.576 | 436.9912 | 248.9755 |

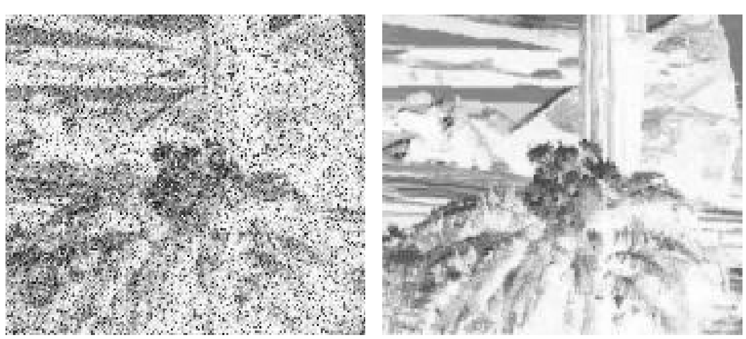
a b

Fig.13. (a) Lena image corrupted by 30% impulse noise and Gaussian noise of σ = 20 (b) Restored image



a b

Fig.14. (a) Frame from Mobile color video sequence corrupted by 30% impulse noise and Gaussian noise of σ = 20 (b) Restored image



a b

Fig.15. (a) Frame from Tempte color video sequence corrupted by 30% impulse noise and Gaussian noise of σ = 20 (b) Restored image

Table.5. PSNR and MSE for N3SS, DS and ICDS for different color video sequences

| Color Noisy Video Sequence | PSNR (dB) | | | MSE | | |
|---|---|---|---|---|---|---|
| | N3SS | DS | ICDS | N3SS | DS | ICDS |
| Mobile | 17.047 | 17.1141 | 27.1724 | 797.637 | 789.6514 | 262.036 |
| Tempete | 20.326 | 20.6187 | 30.4426 | 537.227 | 502.7345 | 224.547 |
| Paris | 21.970 | 22.1546 | 26.7645 | 300.92 | 249.5181 | 181.7603 |

## 4. CONCLUSIONS

In this paper, an adaptive decision based algorithm with robust motion estimation technique is proposed. The proposed algorithm detects and removes mixed Gaussian and impulse noise. The noise detector shows good performance in identifying noise even in mixed noise models. The proposed filter has excellent performance in simultaneous removal of both impulse noise and Gaussian noise. The proposed algorithm performs well by without introducing any visual artifacts and also giving better PSNR, MSE and IEF values. An improved cross diamond search for motion estimation is proposed. Its performance is better when compared to existing several block matching algorithms used for motion estimation. In future work, the motion estimation technique is intended to improve with fewer operations and with more efficiency.

## REFERENCES


[1] Gonzalez and Woods, 2001, Digital Image Processing, Prentice Hall, 2nd edition.

[2] Pitas I. and Venetsanopoulos A. N., 1990, Nonlinear Digital Filters Principles and Applications. Norwell, MA: Kluwer.

[3] Astola. J and Kuosmanen .P, 1997, Fundamentals of Nonlinear Digital Filtering. Boca Raton, CRC.

[4] Ahmad, M.O. and Sundararararajan, D, November 1987: “A Fast algorithm for two dimensional median filtering”, IEEE Trans. Circuits and Systems, Vol. 34, No.11, pp.1364–1374.

[5] Hwang, H. and Haddad, R. A, 1995: “Adaptive median filters: new and results” IEEE Trans. on Image Processing, Vol. 4, No.4, pp. 499-502.

[6] Wang, Z. and Zhang, D, 1999: “Progressive Switching Median filter for the removal of impulse noise from highly corrupted Images”, IEEE Trans. Circuits and Systems, Vol.46, No.1, pp.78-80.

[7] Chan R. H., C.-W. Ho, and Nikolova M, October 2005, “Salt-and-pepper noise removal by median-type noise detectors and detail preserving regularization,” IEEE Trans. Image Process., vol. 14, no. 10, pp. 1479 – 1485.

[8] Srinivasan, K. S. and Ebenezer, D, 2007: “A New Fast and Efficient Decision- Based Algorithm for Removal of High-Density Impulse Noises”, IEEE Signal Processing letters, Vol. 14, No. 3, pp.189 -192.

[9] Jayaraj . V, Ebenezer. D, Aiswarya. K, 2009: “High Density Salt and Pepper Noise Removal in Images using Improved Adaptive Statistics Estimation Filter” International Journal on Computer Science and Network Security, Vol. 9 , No. 11, pp. 170-176.

[10] Srinivasan. E and Ebenezer. D, 2008: “New non-linear filtering strategies for eliminating medium and long tailed noise in images with edge preservation properties”, International Journal of Signal Processing Vol.4, No.3, pp.175 – 181.

[11] Peterson, S.R., Lee, Y.H. Kassam, S.A. Moore, May 1998: “Some Statistical Properties of Alpha-Trimmed Mean and Standard Type M Filters” IEEE Tans. on ASSP, Vol.36, No.5.

[12] Tomasi C.and Manduchi R., 1998:“Bilateral filtering for gray and color images,” Proc. IEEE Int. Conf. Computer Vision, pp.839–846.

[13] W.-K. Chen, *Linear Networks and Systems* Belmont, CA: Wadsworth, 1993, pp. 123–135.

[14] Javier Portilla, Vzsily Strela, Martin J. Wainwright and Ero P.Simoncelli, 2003: “Image Denoising using scale mixtures of Gaussians in the Wavelet domain”, IEEE Trans. on Image Processing, Vol.12, No.11.

[15] Zlokolica V., Pizurica A., Philips W., October 2004: “Recursive temporal denoising and motion estimation of video” International Conference on Image Processing, Singapore.

[16] Rusanovskyy D. and Egiazarian K., September 2005: “Video denoising algorithm in sliding 3D DCT domain,” in Advanced Concepts for Intelligent Vision Systems, ACIVS, (Antwerp, Belgium).

[17] Zlokolica V., Pizurica A., Philips W., July 31-August 4, 2005: "Wavelet based motion compensated filtering of Color Video Sequences", Wavelets XI, SPIE Symposium on Optical Science and Technology, San Diego, CA.

[18] Zlokolica V., Pizurica A., and Philips W., August 2006: “Wavelet-domain video denoising based on reliability measures,” IEEE Trans. Circuits and Systems for Video Tech., vol. 16, pp. 993–1007.

[19] Lian N., Zagorodnov V., and Y. Tan, May 2006: “Video denoising using vector estimation of wavelet coefficients,” in Proc. IEEE International Symposium on Circuits and Systems, pp. 2673–2676.

[20] Rahman S. M. M., Ahmad M. O., and Swamy M. N. S., February 2007: “Video denoising based on inter-frame statistical modeling of wavelet coefficients,” IEEE Trans. Circuits and Systems for Video Tech., Vol. 17, pp. 187–198.

[21] S. Manikandan and D. Ebenezer, 2008: “A Nonlinear Decision-Based Algorithm for Removal of Strip Lines, Drop Lines, Blotches, Band Missing and Impulses in Images and Videos” EURASIP Journal on Image and Video Processing Volume 2008, Article ID 485921, 10 pages.

[22] Chi-Wai Lam, Lai-Man Po and Chun Ho Cheung, 14-17 Dec. 2003: “A New Cross-Diamond Search Algorithm for Fast Block Matching Motion Estimation” Proceedings Of

The 2003 International Conference on Neural Networks and Signal Processing.

[23] J. Ren and T. Vlachos, March 1999: “Dirt detection for archive film restoration using an adaptive spatiotemporal approach” IEEE International Conference on Acoustics and Signal Processing, pp. 3101–3104.

[24] Li, R., Zeng, B. and Liou, M. L., August 1994: “A New three-step search algorithm for block motion estimation”, IEEE Trans. Circuits Syst. Video Technol., Vol. 4, No. 4, pp.438-442.

[25] LIU Hao, ZHANG Wen-jun, CAI Jun, 2006: “A fast block-matching algorithm based on variable shape search”, Journal of Zhejiang university science vol.7, No.2, pp.194-198.

[26] Shan Zhu and Kai-Kuang Ma, February 2000: “A New Diamond Search Algorithm for Fast Block Matching Motion Estimation”, IEEE Trans. Image Processing, Vol. 9, No. 2, pp. 287 –290.

[27] Mak, C.M. and Cham, W.K, 20- 22 October 2004: “Global motion estimation using modified least median of square error” Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, pp.591- 594.

[28] Erickson, J., Her Peled, S. and Mount, D.M, 2006: “On the Least Median square problem” Discrete and Computational Geometry, 36, pp. 593 – 607.