International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:9, 2007

A Technique for Improving the Performance of Median Smoothers at the Corners Characterized by Low Order Polynomials

E.Srinivasan and D.Ebenezer

***Abstract*—**Median filters with larger windows offer greater smoothing and are more robust than the median filters of smaller windows. However, the larger median smoothers (the median filters with the larger windows) fail to track low order polynomial trends in the signals. Due to this, constant regions are produced at the signal corners, leading to the loss of fine details. In this paper, an algorithm, which combines the ability of the 3-point median smoother in preserving the low order polynomial trends and the superior noise filtering characteristics of the larger median smoother, is introduced. The proposed algorithm (called the combiner algorithm in this paper) is evaluated for its performance on a test image corrupted with different types of noise and the results obtained are included.

***Keywords*—**Image filtering, detail preservation, median filters, nonlinear filters, order statistics filtering, Rank order filtering.

I. Introduction

MEDIAN smoother is a simple and an efficient point estimator that has been extensively used in signal and image processing applications since it was first described by Tukey in 1971. It is well known for being able to filter out impulses and preserve edges [1]. Although several robust estimators exist in the literature, running medians have the virtue of being simple and easy to implement [2-5]. Median smoothers have been applied in several areas of digital signal processing, which include speech processing [6,7] and image enhancement [8,9], where the signals of interest often contain edges immersed in high frequency noise. The window length of the median smoother determines the degree of smoothing of non-impulsive noise components and robustness. Greater smoothing and better robustness can be achieved by using larger windows [8]. However, when the window length is enlarged, the smoother loses its ability to follow low order polynomial trends [6] (see Fig.1).

It can be observed from Fig.1 that the 3-point smoother is superior to the larger smoother in preserving signal corners. The 3-point smoother clips only one sample at the most (exactly the corner samples A, B, C, D) to constitute a constant region having a maximum number of three samples.

Manuscript received June 09, 2007.

E. Srinivasan is with Pondicherry Engineering Collge,Pondicherry-605014, India (phone: 91-413-2655281; fax: 91-413-2655101; e-mail: esrinivasan2004@yahoo.co.in)

D. Ebenezer is with Sri Krishna Engineering College, Coimbatore, India : email: ebenez27@mailcity.com).

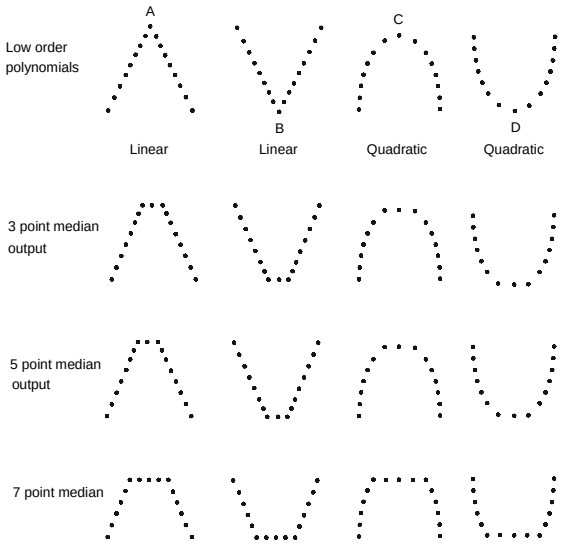

Fig. 1 Effects of various median smoothers at corners (A, B, C, D) characterized by low order polynomials.

The 7-point smoother clips a maximum of three samples and forms a constant region consisting of a maximum number of five samples. To generalize, let us define N=(ws-1)/2, where ws is the window length of the median smoother. Then, the maximum number of samples that form the constant region is N+2, out of which the maximum number of samples that are clipped (from their actual values) to become members of this constant region is N. It is now evident that when ws is increased, more and more samples are clipped about the corners and therefore, the length of the constant region increases. Consequences of smoothing of the corners are two fold: i) the loss of fine details and ii) the manifestation of constant regions as streaks or as amorphous blotches depending upon the geometry of the filter window.

These side effects of the median filters are highly undesirable, because they are perceived as lines or contours that do not exist in the original image [10]. Rabiner et al. [6] used a series combination of linear and median filters for smoothing speech signals and demonstrated the improved performance of the combination smoother in preserving the signal corners. However, the combination smoother is not able to preserve signal boundaries (edges) as faithfully as median

filters, because the linear filters lower the high frequency content of edges. Computationally, it is more complex, since it contains two median and two linear smoothers. Lee and Kassam [11] suggested the use of the Double Window Modified Trimmed Mean (DW MTM) filter, which slides two windows over the signal samples. This filter chooses the median from the small signal window to retain fine details and arrives at the average of those data samples inside the large signal window whose values are close to the median for suppressing non-impulsive noise components. The averaging operation tends to reduce the high frequency content of edges and therefore, edge blurring is quite possible.

Several variations of median filters such as max/median [12], FIR-median hybrid [13,14], multistage median [13,15] and adaptive median [16,17] filters have been proposed in the literature for recovering images without blurring fine details; but all of these filters retain the details only at the expense of noise suppression. Weighted median filters [18] and, particularly, center weighted median filters [19] are important extensions of median smoothers suitable for detail preservation and noise filtering. However, there exists a clear trade-off between detail preservation and noise removal properties of these filters. Moreover, the weights to be assigned to the samples inside the window should be carefully selected depending on the characteristics of both the input image and noise, which is not an easy task. A variety of nonlinear filters reported in the literature recently are effective in removing impulse noise [20-24]; but they do not exhibit satisfactory performance in the presence of multiple noise.

The 3-point median smoother follows polynomial trends quite well; but it is insufficient in attenuating non-impulsive noise components and also fails to remove more than single point outliers. In this paper, an algorithm called the combiner algorithm is introduced. The combiner algorithm combines the detail preserving ability of the 3-point smoother and the better noise filtering characteristics of the larger smoother. The proposed scheme, referred to as the combination smoother, consists of a 3-point median smoother and a larger median smoother of desired window length in parallel; besides, it has a combiner for appropriately combining the outputs of these two smoothers. The combination smoother will be shown to suppress non-impulsive noise, eliminate impulses and preserve fine details satisfactorily.

II. Approach

The scheme of the proposed combination smoother is shown in Fig.2. Median filtering is a discrete-time signal smoothing technique, in which a (2N+1) point wide window is slid over the input signal sequence {x(.,.)}. At each point, the samples inside the window are sorted out (arranged in ascending order) and the middle value is used as the filter output, and associated with the time sample at the center of the window. The filtering procedure is denoted as:

$$m(i,j) = \text{median}(x(i,j+k)),\ -N \le k \le N \text{ and } k \in \mathbf{Z} \qquad (1)$$

where x(i,j) and m(i,j) are the $(i,j)^{th}$ samples of the input and output sequences of the median filter, respectively. The equation (1) describes median filtering of an image sequence as an one-dimensional window moves along the rows. The deterministic, as well as the statistical properties of median filters have been analyzed in [1,10,25].

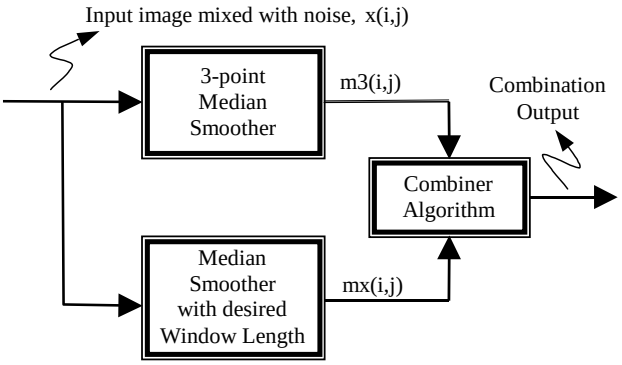

Fig. 2 Combination smoother for improving the median smoother performance on low order polynomials.

The combiner algorithm slides a time-ordered window of length 5 over the larger smoother output samples and identifies the constant regions, which precede/follow the rising/falling low order polynomials. It compares the amplitudes of constant region samples of larger smoother output with the corresponding samples of the 3-point smoother output and finds out whether these constant region samples are the clipped ones. If so, the algorithm replaces the clipped N samples of the larger smoother output with the corresponding samples of the 3-point smoother output. Therefore, the output of the combination smoother resembles the output of the 3-point smoother for N samples at the signal corners and the output of the larger smoother elsewhere.

III. Algorithm

Step 1: Let {x(i,j)}, {m3(i,j)} and {mx(i,j)}, respectively, represent the input image, the 3-point median filtered image and the larger median filtered image of u rows and v columns (see Fig.2). The combiner algorithm, in step 2, slides a 5-point wide time-ordered window over {mx(i,j)}, horizontally (i.e., along the rows), for identifying constant regions. In order for the window to reach the front and rear ends of {mx(i,j)}, two samples each are appended in the beginning and the end of each row. The front endpoints appended to each row take the value of the first sample of that row, while the rear endpoints assume the value of the last sample. Due to this appending strategy, the output samples of the larger smoother are actually stored from the locations{mx(i,3)} to {mx(i,v+2)}.

Step 2: Low order polynomials in the input, when passed through the larger smoother, are transformed as shown in Fig.3. These transformed polynomials are categorized into four types. They are i) constant region following low order polynomial rise: Type-1 (see Fig.3(a)), ii) constant region following polynomial fall: Type-2 (see Fig.3(b)), iii) constant region preceding polynomial fall: Type-3 (see Fig.3(c)) and iv) constant region preceding polynomial rise: Type-4 (see Fig.3(d)). The combiner algorithm, by sliding a 5-point time

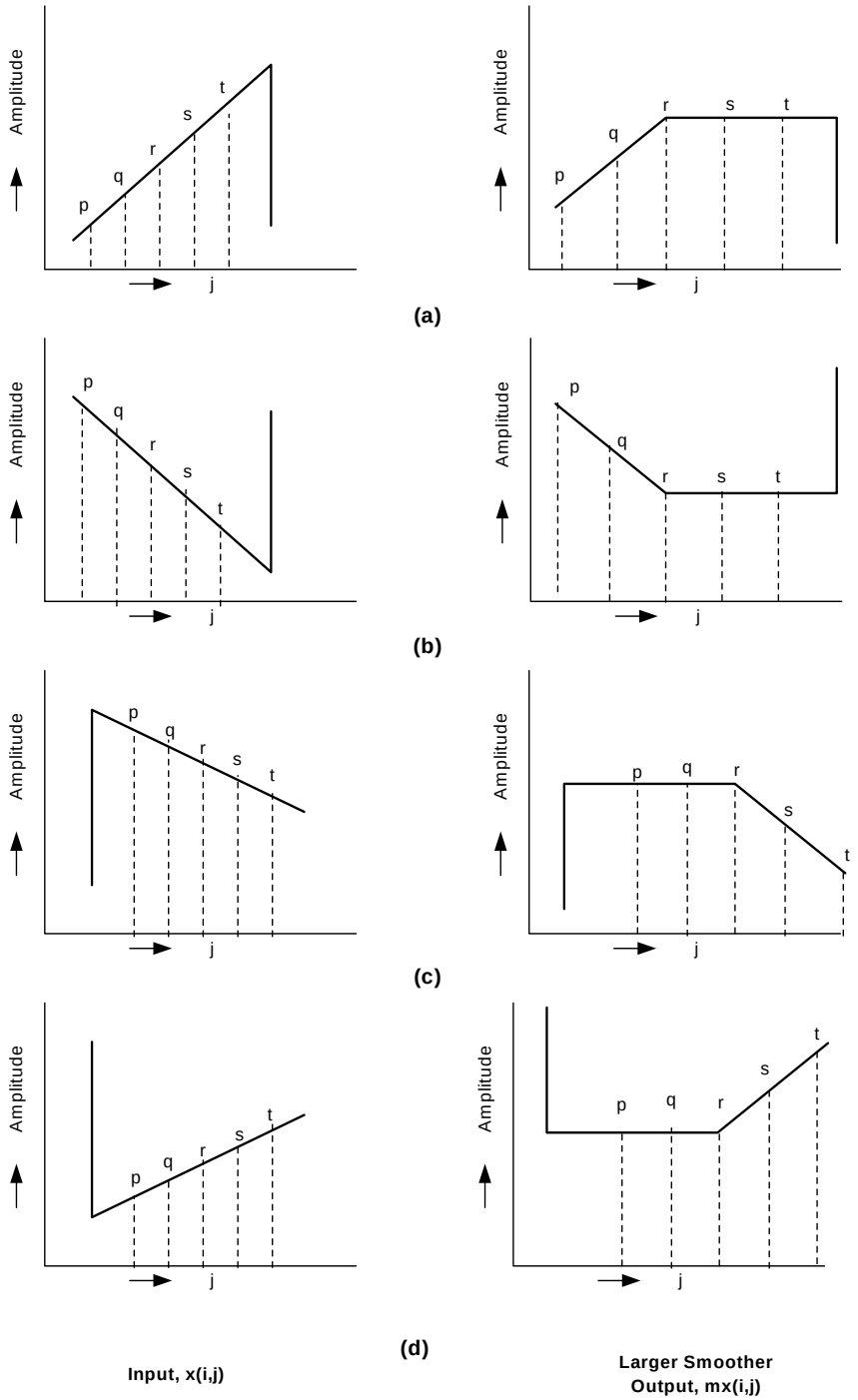

Fig. 3 Transformation of low order polynomial corners, when passed through larger median smoother. a) Type- 1 b) Type-2, c) Type-3 d) Type-4

ordered window, horizontally, over {mx(i,j)}, identifies and distinguishes each of the four types of transformed polynomials for appropriately combining the outputs of the median filters. Arrays {t1(i,j)} and {t2(i,j)}, respectively, store the first samples of all constant regions of Type-1 and Type-2. Arrays {t3(i,j)} and {t4(i,j)} contain the last samples of all constant regions of Type-3 and Type-4, respectively. The pseudo-code description of the segment of the algorithm, which performs this task, is given below:

```
/* for the given value of column index j, 5 samples are taken from the larger smoother output along the i^th row and placed inside the time-ordered window*/
for i = 1 to u
    for j = 1 to v
        p ← mx(i,j)
        q ← mx(i,j+1)
        r ← mx(i, j+2)
        s ← mx(i,j+3)
        t ← mx(i,j+4)
/*constant region which follows low order polynomial is detected (see Figs.3(a),(b))*/
        if ((s-r)=0 AND (t-s)=0)
/*Type-1 transformed polynomial is detected and the first sample of constant region is stored in t1(i,j) (see Fig.3(a))*/
            if ((q-p)>0 AND (r-q)>0)
            t1(i,j) ← mx(i,j+2)
            end
/*Type-2 transformed polynomial is detected and the constant region's first sample is stored in t2(i,j) (see Fig.3(b))*/
            if((q-p)<0 AND (r-q)<0)
            t2(i,j) ← mx(i,j+2)
            end
        end
/*constant region that precedes low order polynomial is detected (see Figs.3(c),(d))*/
        if ((q-p)=0 AND (r-q)=0)
/*Type-3 transformed polynomial is detected and the last sample of constant region is stored in t3(i,j) (see Fig.3(c))*/
            if ((s-r)<0 AND (t-s)<0)
            t3(i,j) ← mx(i,j+2)
            end
/*Type-4 transformed polynomial is detected and the constant region's last sample is stored in t4(i,j) (see Fig.3(d))*/
            if ((s-r) >0 AND (t-s) >0)
            t4(i,j) ← mx(i,j+2)
            end
        end
    end
end
```

The locations of arrays {t1(i,j)}, {t2(i,j)}, {t3(i,j)} and {t4(i,j)} that do not store the first/last samples of constant regions are initialized with zero.

Step 3: The algorithm scans the array {t1(i,j)} row by row and in the increasing order of column index j, starting from 1 to v. When it encounters a non-zero element, it indicates the beginning of Type-1 constant region (see Fig.3(a)). The non-zero element, which is the same as mx(i,j+2), is compared with its corresponding sample of the output of the 3-point smoother i.e., m3(i,j). If both of them are not equal, then mx(i,j+2) is a clipped sample and therefore, starting with the present value of j, N horizontal samples (where N=(ws-1)/2 and ws is the window length of the larger smoother) are removed from {m3(i,j)}; these N samples replace the corresponding N samples of {mx(i,j)} in the increasing order of j (starting from the position r as shown in Fig.3(a)). On the other hand, if both of them are equal, then mx(i,j+2) is not a clipped sample and therefore, no replacement is required for that sample. Hence, j is incremented and then N samples are removed from {m3(i,j)} and these N samples are used to replace
the corresponding samples of {mx(i,j)} in the increasing order of j. This procedure is repeated for all non-zero elements of {t1(i,j)} as illustrated below using the pseudo-code:

```
for i=1 to u
    for j = 1 to v
        if (t1(i,j) ≠ 0)
          if (t1(i,j) = m3(i,j))
          j ←  j+1
          end
          for k = j to (j+N-1)
          mx(i,k+2) ← m3(i,k)
          end
          j ← j+N-1
        end
    end
end
```

The array {t2(i,j)} contains the first samples of Type-2 constant regions (see Fig.3(b)). Therefore, for the array {t2(i,j)}, the algorithm repeats the procedure it followed for {t1(i,j)}.

Next, the algorithm scans {t3(i,j)} row by row in the decreasing order of column index j, starting from v to 1. A non-zero element, encountered during the scan, marks the end of Type-3 constant region (see Fig.3(c)). This non-zero element of {t3(i,j)}, which is same as mx(i,j+2), is compared with the corresponding sample of {m3(i,j)}. If they are not equal, it indicates that mx(i,j+2) is a clipped sample and therefore, from the current value of j, N samples are removed from {m3(i,j)}, and the corresponding samples of {mx(i,j)} are replaced with these samples in the decreasing order of j (starting from the position r in Fig.3(c)). On the other hand, if they are equal, then mx(i,j+2) is not a clipped sample and therefore, it is left undisturbed; then j is decremented, and N samples of {m3(i,j)} are removed and these N samples replace the corresponding samples of {mx(i,j)}, in the decreasing order of j. This procedure is repeated for all non-zero elements of {t3(i,j)} as described below:

```
for j = 1 to u
    for j = v to 1
        if (t3(i,j) ≠ 0)
          if (t3(i,j) = m3(i,j))
          j ← j-1
        end
          for k = j to (j-N+1)
          mx(i,k+2) ← m3(i,k)
          end
          j ← j-N+1
        end
    end
end
```

The array {t4(i,j)} contains the last samples of Type-4 constant regions (see Fig.3(d)). Therefore, for the array

{t4(i,j)}, the algorithm repeats the procedure it followed for {t3(i,j)}.

IV. Results and Discussion

The performance of the combination smoother is evaluated by applying it to a test image degraded by i) mixed impulses (both positive and negative impulses), ii) non-impulsive noise (e.g., Gaussian noise) and iii) impulsive and non-impulsive noise. The test image used is a picture of Bacteria (178x178 pixels, 8 bits/pixel). The window length of the larger smoother is chosen to be 9.

A quantitative comparison is performed on the basis of an objective quality measure, called enhancement factor F_e, which is defined as the ratio of mean square error before, and after filtering all the image pixels. This yields:

$$F_e = \frac{\sum_{i=1}^{178}\sum_{j=1}^{178}[s(i,j)-x(i,j)]^2}{\sum_{i=1}^{178}\sum_{j=1}^{178}[s(i,j)-f(i,j)]^2} \quad (2)$$

where s(i,j), x(i,j) and f(i,j), respectively, are the $(i,j)^{th}$ samples of original (noise-free) image, noisy input image and the filtered output image. The enhancement factors achieved by applying the median, and combination smoothers on the test image contaminated by different levels of mixed impulse noise and Gaussian noise, respectively, are plotted in Fig.4 and Fig.5.It can be observed that when the noise level is very low, the 3-point smoother enhances the images better than the combination smoother. However, its performance deteriorates rapidly with increasing noise level and becomes much inferior to that of the combination smoother (see Fig.4).

The 9-point smoother is consistently outperformed by the combination smoother (see Fig.4 and Fig.5). Fig.6 depicts the image enhancement factors obtained by applying the median, and combination smoothers on the images corrupted by multiple noise (5% mixed impulses and different levels of zero mean Gaussian noise). The combination smoother outperforms both the 3-point and the 9-point smoothers, due to its greater smoothing capability, better robustness and detail preserving characteristics.

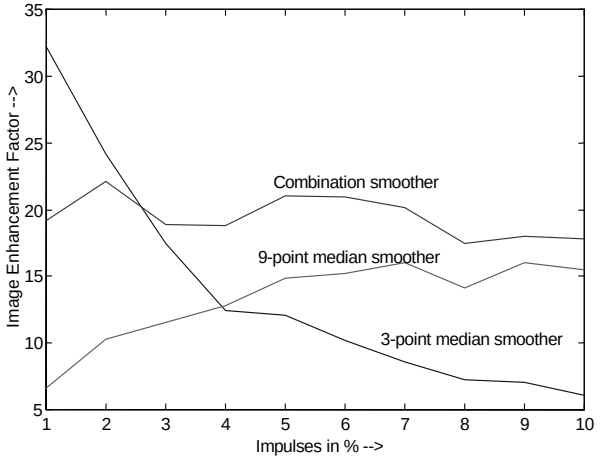

Fig. 4 Comparison of image enhancement factor when applied to Bacteria image corrupted by different levels of mixed impulses.

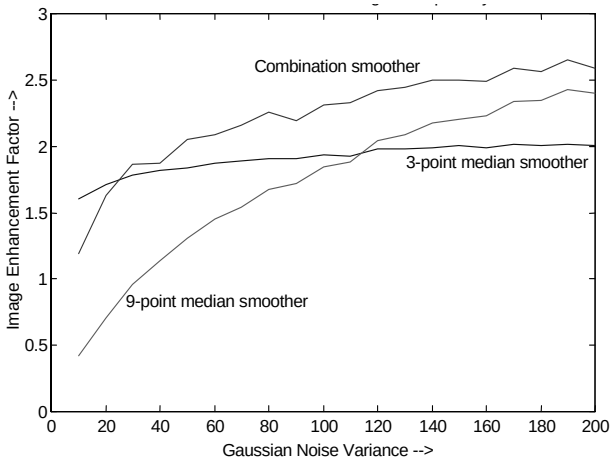

Fig. 5 Comparison of image enhancement factor when applied to Bacteria image corrupted by different levels of zero mean Gaussian noise.

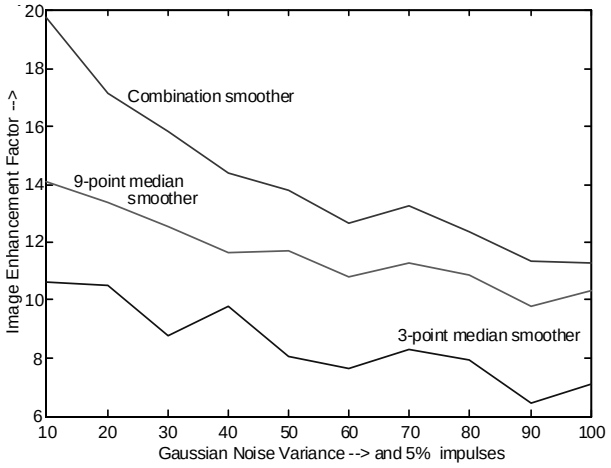

Fig. 6 Comparison of image enhancement factor when applied to Bacteria image corrupted by 5% mixed impulses and different

The performance of the combination smoother is also examined and compared with that of conventional smoothers on the basis of subjective criterion, that is, human visual perception. Fig.7, Fig.8 and Fig.9, respectively, show the results of filtering of test image corrupted by i) 5% mixed impulses, ii) Gaussian noise (mean=0 and variance=100) and iii) 5% mixed impulses and Gaussian noise (mean=0 and variance=100). The 3-point smoother fails to remove the impulses completely (see Fig.7(c) and Fig.9(c)) and provides insufficient attenuation in the presence of Gaussian (non-impulsive) noise (see Fig.8(c) and Fig.9(c)). The 9-point smoother, as can be seen from Fig.7(d), Fig.8(d) and Fig.9(d),exhibits good noise cleaning properties; but the filter smears image corners resulting in the loss of fine details. The combination smoother is seen to be superior to the 3-point

smoother in noise filtering, because it discards the impulses completely and attenuates Gaussian noise satisfactorily (see Fig.7(e), Fig.8(e) and Fig.9(e)). Further, the better space

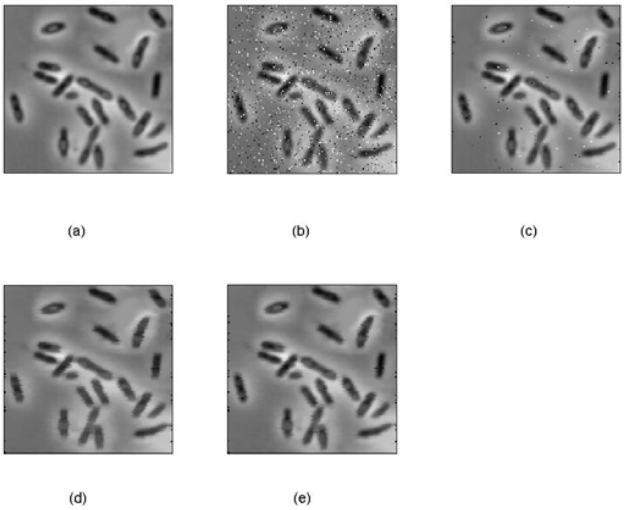

Fig. 7 Performance illustration using Bacteria image in the presence of impulsive noise: (a) Test image Bacteria; (b) Test image corrupted with 5% mixed impulses; (c) 3-point median smoother output;(d) 9-point median smoother output; (e) Combination smoother output.

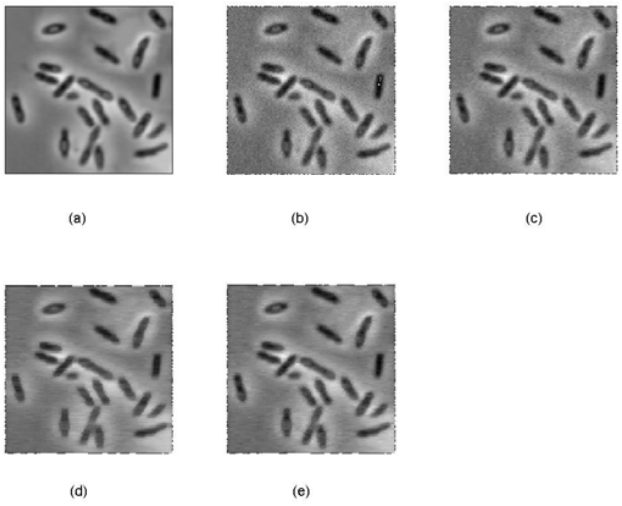

Fig. 8 Performance illustration using Bacteria image corrupted by non-impulsive (Gaussian) noise: (a) Test image Bacteria; (b) Test image corrupted by zero mean Gaussian (non-impulsive) noise of variance 100; (c) 3-point median filtered image; d) 9-point median filtered image; (e) Combination smoother output.

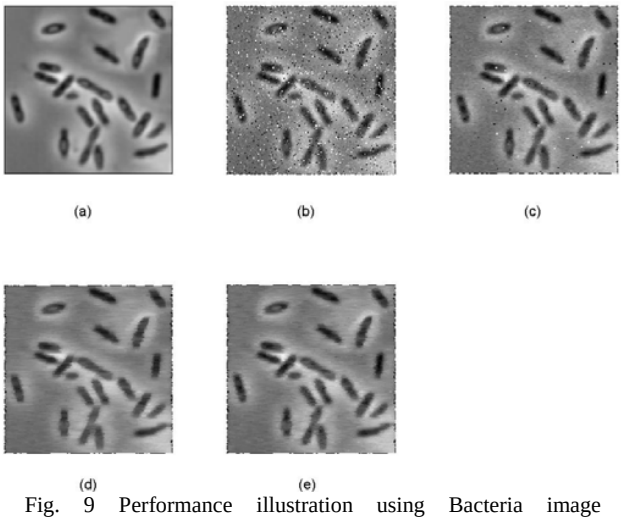

Fig. 9 Performance illustration using Bacteria image contaminated by impulsive and non-impulsive noise: a) Test image Bacteria; (b) Test image corrupted with 5% mixed impulses and zero mean Gaussian (non-impulsive) noise of variance 100; (c) 3-point median filtered image; (d) 9-point median filtered image; (e) Combination smoother output.

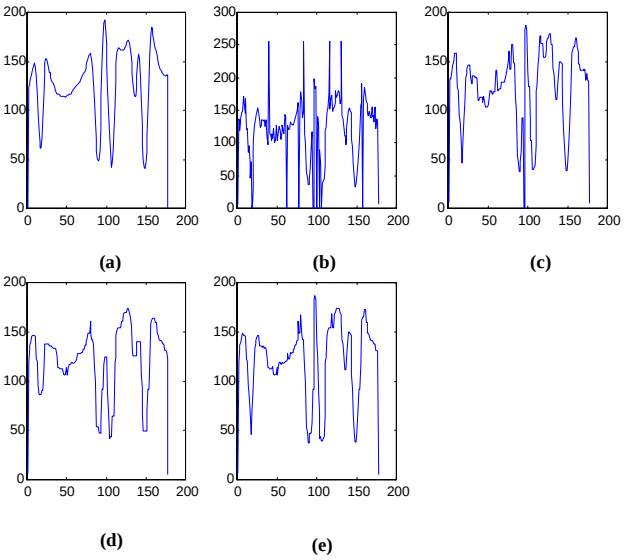

Fig. 10 Performance illustration of combination smoother on the row 122 of Bacteria image corrupted by impulsive noise: (a) Plot of row 122 of Bacteria image; (b) Plot of row 122 of Bacteria image corrupted by 5% mixed impulses; (c) 3-point median smoother output – fails to remove more than single point impulses, while preserving the signal corners; (d) 9-point median smoother output – discards impulses completely, but smears the signal corners resulting in loss of fine-details; (e) Combination smoother output – removes impulses as well as 9-point smoother and preserves signal corners as effectively as the 3-point median.

resolution of the combination smoother output indicates that the combination smoother is better than the 9-point smoother in preserving the image details. Fig.10 and Fig.11, respectively, present the results of filtering of row 122 of Bacteria image corrupted by i) 5% mixed impulses and ii) 5% mixed impulses and Gaussian noise (mean=0 and variance=100). These results are shown for a close examination of the effectiveness of combination smoother in noise filtering and detail preservation from the point of view of visual perception. It can be noted that the combination smoother output resembles the output of the 3-point smoother at the signal corners and the output of the 9-point smoother elsewhere. The output of the combination smoother, as observed from these illustrations, is seen to inherit the merits of both the 3-point smoother (detail preservation) and the 9-point smoother (greater smoothing and better robustness).

However, the proposed combination smoother suffers from a drawback. The combiner algorithm searches for constant regions preceding/following the polynomials in the output of the larger smoother and replaces N samples of those constant regions with the corresponding N samples of the 3-point smoother output. The signal may, actually, have some constant regions (not flattened due to filtering) preceding/following polynomials, such as ramp edges; N samples of these actual constant regions are also replaced as described earlier. The combination output retains the robustness and smoothing ability of larger smoother in all the regions except where the larger smoother output samples are replaced. Therefore, the robustness and noise suppression

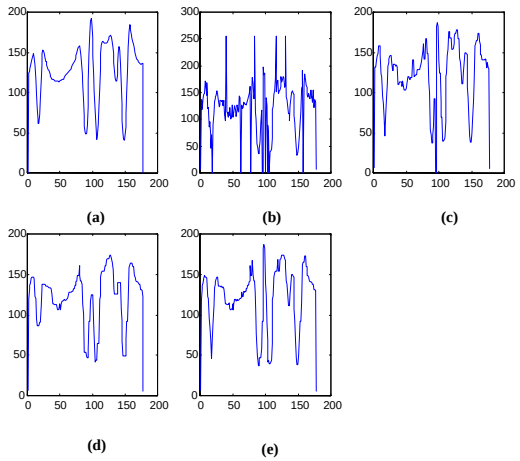

Fig. 11 Performance illustration of combination smoother on the row 122 of Bacteria image confounded by impulsive, and non-impulsive noise: (a) Plot of row 122 of Bacteria image; (b) Plot of row 122 of Bacteria image corrupted by 5% mixed impulses and zero mean Gaussian noise of variance 100; (c) 3-point median smoother output – fails to remove impulses completely and does not attenuate Gaussian noise adequately; (d) 9-point median smoother output – eliminates impulses completely and suppresses Gaussian noise quite well, but blurs the signal corners; (e) Combination smoother output – removes impulses completely, reduces Gaussian noise satisfactorily and preserves signal corners effectively.

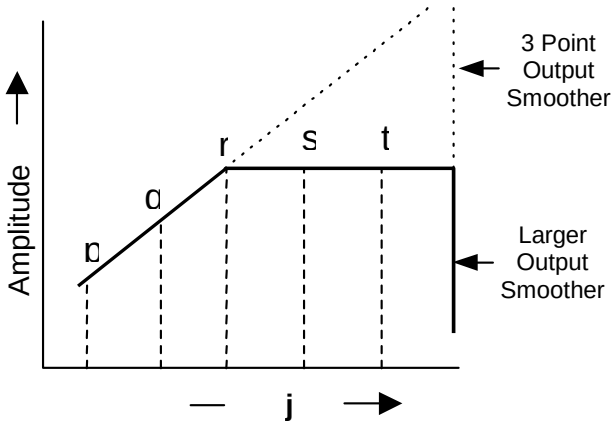

Fig. 12 Output of 3-point and larger median smoothers at a sharp corner.

characteristics of the combination output tend to deteriorate as those of the 3-point smoother in the regions of the replaced samples.

We have attempted to reduce the severity of this drawback in the regions of the replaced samples by imposing signal dependent conditions. Consider the transformed low order polynomial of Type-1 shown in Fig.12. When {t1(i,j)} is scanned, the algorithm may come across a non-zero element indicating the beginning of a constant region as shown in Fig.12. Let us define:

$$u = r - q = mx(i,j+2) - mx(i,j+1) \quad (3)$$

It can be observed from Fig.12 that u denotes the slope of the polynomial computed just prior to the beginning of a constant region. The combiner algorithm will replace N samples one after another subject to the satisfaction of the following condition:

$$m3(i,j) > mx(i,j+2) \text{ AND } m3(i,j) < (mx(i,j+2) + N*u) \quad (4)$$

If the constant region is a smoothed corner as shown in Fig.12, then the above condition will be satisfied for the entire N samples and therefore, the replacement will be carried out. If negative impulses are present among those N samples of the output of the 3-point smoother output, then the first part of the condition, i.e., $m3(i,j) > mx(i,j+2)$ cannot be satisfied and therefore, negative impulses will not be allowed to replace the corresponding samples of the larger smoother output. Thus the first part of Equation (4) improves the robustness of the combination output against negative impulses in the regions of the replaced samples. For the positive impulses, whose signal magnitude is greater than N*u, the second part of the condition fails and therefore, the robustness of the combination output in the regions of replaced samples will be as good as that of the larger smoother. However, if the amplitude of positive impulses, which appear within the replacing N samples, is less than N*u, then such impulses will also appear at the combination output. But, it should be noted that the probability of occurrence of an impulse of small magnitude is quite low. For the other three types of transformed polynomials also, such signal dependent

conditions are imposed to improve the robustness of the combination smoother in the regions of the replaced samples. Table I compares the 3-point, 9-point and combination smoothers in terms of robustness, noise filtering and detail preserving characteristics.

TABLE I
PERFORMANCE COMPARISON OF 3-POINT MEDIAN, 9-POINT MEDIAN AND COMBINATION SMOOTHERS

Type of smoother / *Characteristics*	3 Point Median	9 Point Median	Combination
Robustness	Fair	Better	Good
Noise-filtering	Fair	Better	Good
Detail preservation	Better	Poor	Better

V Conclusion

In this paper, a technique for improving the performance of median smoothers at the corners characterized by low order polynomials is proposed. An algorithm called the combiner algorithm, which imbibes the positive features of both the 3-point median smoother (detail preservation) and the larger median smoother (greater smoothing and better robustness) is described. The combination smoother, built using a 3-point median smoother, a larger median smoother and the combiner algorithm, is shown to be effective in eliminating impulses and suppressing non-impulsive noise; besides, it has good detail preserving characteristics. It is an all-median scheme and therefore, edge preservation properties and the computational simplicity of the median filter are retained.

References

[1] E.Ataman, V.K.Aatre and K.M.Wong, "Some Statistical Properties of Median Filters," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-29, pp.1073-1075, Oct.1981.
[2] K.Oflazer, "Design and Implementation of a Single-Chip 1-D Median Filter," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-31, no.5, pp.1164-1168, Oct.1983.
[3] N.Demassieux, F.Jutand, M.Saint-Paul and M.Dana, "VLSI Architecture for a One-Chip Video Median Filter," Proc. Int. Conf. on Acoust., Speech and Signal Processing, pp.1001-1004, 1985.
[4] E.Ataman, V.K.Aatre and K.M.Wong, "A Fast Method for Real Time Median Filtering," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-28, no.4, pp.415-421, Aug.1980.
[5] D.J.Delman, "Digital Pipelined Median Filter Design for Real-Time Image Processing," Proc. SPIE, vol.298, pp.184-188, Aug.1981.
[6] L.R.Rabiner, M.R.Sambur and C.E.Schmidt, "Applications of a Nonlinear Smoothing Algorithm to Speech Processing," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-23, pp.552-557, Dec.1975.
[7] N.S.Jayant, "Average- and Median-Based Smoothing Techniques for Improving Digital Speech Quality in the Presence of Transmission Errors," IEEE Trans. Commun., vol.COM-24, pp.1043-1045, Sept.1976.
[8] I.Pitas and A.N.Venetsanopoulos, Nonlinear Digital Filters: Principles and Applications. Boston, MA: Kluwer Academic, 1990.
[9] W.K.Pratt, Digital Image Processing. New York: Wiley-Interscience, 1991.
[10] A.C.Bovik, "Streaking in Median Filtered Images," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-35, no.4, pp.493-503, April 1987.
[11] Y.H.Lee and S.A.Kassam(1985), "Generalized Median Filtering and Related Nonlinear Filtering Techniques," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-33, pp.672-683, June 1985.
[12] G.R.Arce and M.P.McLoughlin, "Theoretical Analysis of Max/Median Filters," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-35, no.1, pp.60-69, Jan. 1987.
[13] A.Nieminen and P.Heinonen and Y.Neuvo, "A New Class of Detail Preserving Filters for Image Processing," IEEE Trans. Pattern Anal. and Mach. Intell., vol.PAMI-9, pp.74-90, Jan. 1987.
[14] P.Heinonen and Y.Neuvo, "FIR-Median Hybrid Filters," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-35, pp.832-838, June 1987.
[15] G.R.Arce and R.E.Foster, "Detail Preserving Ranked-Order Based Filters for Image Processing," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-37, pp.83-98, Jan. 1989.
[16] R.Bernstein, "Adaptive Nonlinear Filters for Simultaneous Removal of Different Kinds of Noise in Images," IEEE Trans. Circuits and Systems, vol.CAS-34, no.11, pp.1275-1291, Nov. 1987.
[17] H.M.Lin and A.N.Willson,Jr., "Median Filters with Adaptive Length," IEEE Trans. Circuits and Systems, vol.CAS-35, no.6, pp.675-690, June 1988.
[18] L.Yin, R.Yang, M.Gabbouj and Y.Neuvo, "Weighted Median Filters: A Tutorial," IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing, vol.43, no.3, pp.157-192, Mar. 1996.
[19] S.J.Ko and Y.H.Lee, "Center Weighted Median Filters and Their Applications to Image Enhancement," IEEE Trans. Circuits and Systems, vol.CAS-38, no.9, pp.984-993, Sept. 1991.
[20] Z.Wang and D.Zhang, "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," IEEE Trans. Circuits and Systems-II:Analog and Digital Signal Processing, Vol.46, No.1, pp.78-80, Jan.1999.
[21] T.Chen, K.K.Ma and L.H.Chen, "Tri-State Median Filter for Image Denoising," IEEE Trans. Image Processing, Vol.8, No.12, pp.1834-1838, Dec.1999.
[22] H.L.Eng and K.K.Ma, "Noise Adaptive Soft Switching Median Filter," IEEE Trans. Image Processing, Vol.10, No.2, pp.242-251, Feb.2001.
[23] S.Zhang and M.A.Karim, "A New Impulse Detector for Switching Median Filters,"IEEE Signal Processing Letters, Vol.9, no.11, pp.360-363, Nov.2002.
[24] T.C.Lin and P.T.Yu, "Salt-Pepper Impulse Noise Detection and Removal Using Multiple Thresholds for Image Restoration," Journal of Information Science and Engineering 22, 189-198, (2006).
[25] N.C.Gallagher,Jr. and G.L.Wise, "A Theoretical Analysis of the Properties of Median Filters," IEEE Trans. Acoust., Speech and Signal Processing, vol.ASSP-29, pp.1136-1141, Dec. 1981.

E.Srinivasan was born at Govindanagaram, Tamil Nadu, India. He received Bachelor's degree (B.E) in Electrical and Electronics Engineering from PSG College of Technology, Coimbatore, India in 1984 and Master's degree (M.E) in Instrument Technology from Madras Institute of Technology, Anna University, Chennai, India in 1987. He obtained Ph.D. degree in Electrical Engineering from College of Engineering, Anna University, Chennai, India in the year 2003 for his research work in nonlinear filters. He has twenty national/international publications to his credit.

He is an Assistant Professor in the department of Electronics and Communication Engineering, Pondicherry Engineering College, Pondicherry, India where he has been serving since 1987. His areas of interests are digital signal processing and nonlinear order-statistic filters for image processing applications.Dr.Srinivasan is a reviewer for AMSE Journal for Signal Processing, France and a life member of Indian Society for Technical Education. He is a recipient of outstanding student award during his undergraduate study program.

D.Ebenezer obtained his Bachelors Degree (BE) in Electronics and communication Engineering (1980) and Masters degree(ME) in Applied Electronics (1983) from PSG College of Technology, Coimbatore, India and Ph.D. (1994) in Electrical Engineering from Anna University, Chennai, India.

He served as a member of the faculty of the department of Electronics and Communication Engineering, College Of Engineering, Anna University, Chennai, India for twenty one years since 1985. At present, he is a Professor in the department of Electronics and Communication Engineering, Sri Krishna Engineering College, Coimbatore, India. He has published numerous research papers in the area of digital signal processing. His areas of interests are digital filters and telecommunication switching.

Dr.Ebenezer is a referee for Journal of Medical Engineering and Physics, UK. He is an associate member of Institution of Engineers (India) and a member of Indian Society for Technical Education.