

Image detail-preserving filter for impulsive noise attenuation

X. D. Jiang

Abstract: A new nonlinear filter is proposed for attenuating impulsive noise while preserving image details. The filter truncates the grey value of a pixel to the maximal or minimal value of its enclosed surrounding band. Impulsive noise inside the band is thus attenuated while image details are preserved as long as they stretch to the band. The recursive form of the proposed filter leads to a simple architecture for fast implementation. Theoretical analysis and experimental results demonstrate the effectiveness of this new filter for both noise attenuation and detail preservation. For moderately contaminated images, as shown in the experiments, the proposed filter outperforms the standard median filter, the centre-weighted median filter and the unidirectional multi-stage median filter in terms of mean absolute error and filtering speed.

1 Introduction

Filtering a digital image to attenuate noise while keeping the image details preserved is an essential part of image processing. For impulsive noise attenuation, nonlinear filters, such as the median filter [1] and its variants have shown superior performance to linear filters. However, the standard median (SM) filter suffers the drawback of removing important image details. This undesirable effect is not acceptable in many applications where the preservation of image structure is important. Finding a method that is efficient in both noise reduction and detail preservation is an active area of research.

To trade off detail preservation against noise reduction, some solutions have been proposed in the literature. The weighted median filter [2] that uses weights to control the filtering behaviour preserves features of given shapes and sizes [3]. The centre-weighted median (CWM) filter [4] only weights the centre pixel of the filtering window. The tristate median filter [5] and the soft-switching median filter [6] incorporate SM and CWM filters into a noise detection framework to enhance the noise attenuation while preserving the detail. Other approaches based on the noise detection procedure include the min-max filter [7] and switching-based median filters [8, 9]. The unidirectional multistage median (UMM) filter [10, 11] uses several subwindows inside the processing mask. Thin lines oriented along these subwindows are preserved but those along other orientations are often distorted. Two-stage architectures based on the median filter were also proposed in [12, 13]. The multistage median filters were extended to the library-median-stack filters [14] to preserve more image details by using more patterns that can be selected

from a library. Recently, a peak-and-valley filter that is faster than the median filter and can preserve image details was proposed in [15]. The peak-and-valley filter is based on min-max operators [16, 17] that were also employed in the noise peak eliminator filter [18].

This work proposes a novel nonlinear filter, called the truncation filter, which attenuates impulsive noise while preserving image details. Image details will be preserved as long as they stretch to or beyond the enclosed surrounding bands, while the isolated impulsive noise inside the bands will be attenuated. A simple filter architecture is proposed based on the recursive form of the truncation filter, which leads to a fast filtering implementation.

2 Truncation filter and its detail preservation properties

For a pixel (i, j) with grey value $x(i, j)$, we can find N square windows of size $M \times M$ that cover this pixel ($N = M^2$). Such windows are called inner windows and are denoted by WI_k . For each inner window WI_k , an outer window WO_k of size $(M + 2r) \times (M + 2r)$, $r \geq 1$, is defined that has the same centre as WI_k . So for each pixel (i, j) , N enclosed square surrounding bands B_k , $k = 1, 2, \dots, N$, with thickness r are defined as $B_k = WO_k - WI_k$. For $r = 1$, each band consists of $L = 4(M + 1)$ pixels. Fig. 1 shows a pixel (i, j) and three of its nine square surrounding bands B_1 , B_5 and B_9 for $M = 3$ and $r = 1$.

Let u_k and v_k denote the maximal and minimal grey values of the pixels in the surrounding band B_k , respectively, i.e.

$$u_k = \max_{m,n} (x(m, n)) | (m, n) \in B_k \quad (1)$$

$$v_k = \min_{m,n} (x(m, n)) | (m, n) \in B_k \quad (2)$$

For an input pixel $x(i, j)$, the output of the proposed truncation filter $y(i, j)$ is defined by

$$y(i, j) = \begin{cases} u_{\min}, & \text{iff } x(i, j) > u_{\min} \\ v_{\max}, & \text{iff } x(i, j) < v_{\max} \\ x(i, j), & \text{otherwise} \end{cases} \quad (3)$$

© IEE, 2003

IEE Proceedings online no. 20030404

DOI: 10.1049/ip-vis:20030404

Paper first received 9th September 2002 and in revised form 6th February 2003

The author is with the Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613

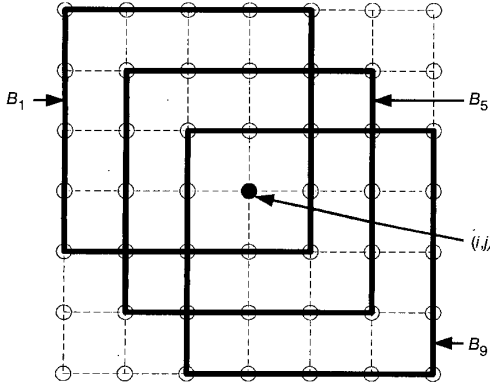


Fig. 1 Three sample square surrounding bands B_1 , B_5 and B_9 of a pixel (i, j) for $M=3$ and $r=1$

where

$$u_{\min} = \min_k(u_k | k = 1, 2, \dots, N) \quad (4)$$

$$v_{\max} = \max_k(v_k | k = 1, 2, \dots, N) \quad (5)$$

All pixels in the N surrounding bands together with the input pixel $x(i, j)$ compose a square window of size $W \times W = (2M + 2r - 1) \times (2M + 2r - 1)$. Therefore, the filter window size of the truncation filter is $W \times W$. It should be mentioned that the surrounding band of the truncation filter is not limited to the square shape. It could be any shape as long as it is four-connected and enclosed. For simplicity, in this paper, we only explore the properties of the truncation filter for the square surrounding band with a width of one pixel ($r=1$).

A simple extension of (3) is the recursive form that produces the output image by updating the grey values of the input image instead of creating a new output image. From (3), the recursive form of the truncation filter can be expressed as

$$x(i, j) \leftarrow \begin{cases} u_{\min}, & \text{iff } x(i, j) > u_{\min} \\ v_{\max}, & \text{iff } x(i, j) < v_{\max} \\ x(i, j), & \text{otherwise} \end{cases} \quad (6)$$

An advantage of the recursive truncation filter is that it can lead to a simple and fast implementation. In cases of $x(i, j) > u_{\min}$, (6) can be rewritten as

$$x(i, j) \leftarrow u_{\min} = \min(u_N, u_{N-1}, \dots, u_1, x(i, j)) \\ = \min(u_N, \min(u_{N-1}, \dots, \min(u_1, x(i, j)) \dots)) \quad (7)$$

It is easy to see that (7) can be iteratively implemented using $x(i, j) \leftarrow \min(u_k, x(i, j))$, where $k=1, 2, \dots, N$. Similarly, in cases of $x(i, j) < v_{\max}$, (6) can be iteratively implemented using $x(i, j) \leftarrow \max(v_k, x(i, j))$, with $k=1, 2, \dots, N$. Since the N surrounding bands of a pixel have different locations, (6) could be implemented by placing the surrounding band $B=WO-WI$ at the N different possible locations and performing

$$x(i, j) \leftarrow \begin{cases} \min(u, x(i, j)), & \text{iff } x(i, j) < u \\ \max(v, x(i, j)), & \text{iff } x(i, j) < v \\ x(i, j), & \text{otherwise} \end{cases} \\ = \begin{cases} u, & \text{iff } x(i, j) > u \\ v, & \text{iff } x(i, j) < v \\ x(i, j), & \text{otherwise} \end{cases} \quad (8)$$

for each location of the band, where $u = \max_{(m, n) \in B} x(m, n)$ and $v = \min_{(m, n) \in B} x(m, n)$. Therefore, the recursive truncation filter could be simply implemented by applying

$$x(i, j) \leftarrow \begin{cases} u, & \text{iff } x(i, j) > u \\ v, & \text{iff } x(i, j) < v, \text{ for } \forall x(i, j) | (i, j) \in WI \\ x(i, j), & \text{otherwise} \end{cases} \quad (9)$$

and shifting the surrounding band pixel-wise throughout the image.

Equation (9) truncates the grey values of all N pixels enclosed by a surrounding band to the maximal or minimal grey value of the band. This is why the proposed filter is called a truncation filter. Equation (9) leads to a fast implementation of the recursive truncation filter by updating N pixels with (9) through one maximum/minimum search in one band instead of updating one pixel with (6) through N times of a maximum/minimum search in N different bands.

An important property of the proposed truncation filter is that it preserves image details. Consider an eight-connected image structure that forms a set of pixels S with constant grey values h in an arbitrary background, i.e.

$$x(i, j) = \begin{cases} h, & \text{if } (i, j) \in S \\ a(i, j), & \text{if } (i, j) \notin S \end{cases} \quad (10)$$

Suppose two pixels $p_1 = (i_1, j_1)$, $p_2 = (i_2, j_2)$ can be found in this image structure that satisfies

$$\exists p_1 \exists p_2 \{ p_1 \in S \wedge p_2 \in S \wedge (|i_1 - i_2| > M \vee |j_1 - j_2| > M) \} \quad (11)$$

If any pixel of the structure S is enclosed by a surrounding band, at least one pixel of S will be located at the band since the band is four-connected and enclosed. We therefore have $v \leq h \leq u$ in (9). As a result, the updating formula (9) for each pixel of the structure S becomes

$$x(i, j) \leftarrow x(i, j) = h \quad \text{for } \forall (i, j) | (i, j) \in S \quad (12)$$

Structure S is therefore invariant to the truncation filter. More general for image structures of nonhomogenous grey level, a pixel is invariant to the filter if it is connected to two eight-connected pixel sets with its grey value not larger than the minimum of one set and not smaller than the maximum of the other, where the two sets satisfy the geometrical condition (11). From this analysis, we see that the proposed truncation filter preserves edges, corners, ramps and lines of any width and any orientation as long as they are longer than $M = (W - 1)/2$ pixels. A curve will even be preserved if it stretches to or beyond the surrounding band whereby condition (11) is satisfied.

Compared to the truncation filter, the SM filter cannot preserve any one-pixel-wide line even with the smallest 3×3 window. To preserve the line ending, i.e. there are only $(W + 1)/2$ pixels of the line in the window, the weight c of the CWM filter should be larger than $W^2 - W$ since there are $W^2 - (W + 1)/2$ elements of 'background value' and $(W + 1)/2 + c - 1$ elements of 'line value' in the median operation. The UMM filter preserves straight lines that are longer than $(W - 1)/2$ pixels only if the line orientation is one of the filter's four orientations. Interestingly, for the window size 3×3 , it is easy to see that the UMM filter and the CWM filter with the central weight of 7 ($3^2 - 3 + 1$) are equivalent to the proposed truncation filter. This is not surprising as the centre pixel together with any other pixel in the 3×3 window will form a two-pixel-long line along one of the UMM filter's four

orientations. However, this interesting equivalence among the UMM filter, the CWM filter and the proposed truncation filter does not hold if the window size is larger than 3×3 . Grey-scale morphological filters [19] also use min and max operators but they process the entire structuring element and operate sequentially (erosion and dilation). These filters will therefore remove image objects that are smaller than the structuring element. Our truncation filter is essentially different since the min and max operate simultaneously and only on the surrounding band.

3 Statistical properties of the truncation filter

Let the grey value of the input image $x(i, j)$ be independent, identically distributed with the probability distribution function $F_x(t) = P\{x \leq t\}$, where $P\{A\}$ is the probability of event A . Let $G_x(t) = 1 - F_x(t)$ and U_k and V_k denote the events $u_k \leq t$ and $v_k \leq t$, respectively. From (1) and (2) we have

$$\begin{aligned} P(U_k) &= P\left\{\bigcap_{(i,j) \in B_k} (x(i, j) \leq t)\right\} \\ &= \prod_{(i,j) \in B_k} P\{x(i, j) \leq t\} = F_x(t)^L \end{aligned} \quad (13)$$

$$\begin{aligned} P(\bar{V}_k) &= P\left\{\bigcap_{(i,j) \in B_k} (x(i, j) > t)\right\} \\ &= \prod_{(i,j) \in B_k} P\{x(i, j) > t\} = G_x(t)^L \end{aligned} \quad (14)$$

From (4) and (13) the distribution function of u_{\min} is obtained by

$$\begin{aligned} F_{u_{\min}}(t) &= P\{(\exists u_k)(u_k \leq t)\} = P\left\{\bigcup_{k=1}^N U_k\right\} \\ &= \sum_{k=1}^N P\{U_k\} - \sum_{\substack{k,m \\ k < m}} P\{U_k \cap U_m\} \\ &\quad + \sum_{\substack{k,m,n \\ k < m < n}} P\{U_k \cap U_m \cap U_n\} - \dots \\ &\quad + (-1)^{N-1} P\{U_1 \cap U_2 \cap \dots \cap U_N\} \\ &= N \cdot F_x(t)^L - \sum_{\substack{k,m \\ k < m}} F_x(t)^{L(k,m)} \\ &\quad + \sum_{\substack{k,m,n \\ k < m < n}} F_x(t)^{L(k,m,n)} - \dots \\ &\quad + (-1)^{N-1} F_x(t)^{(2M+1)^2-1} \end{aligned} \quad (15)$$

where $L(k_1, k_2, \dots, k_i)$ is the number of the pixels in the union of surrounding bands $B_{k_1}, B_{k_2}, \dots, B_{k_i}$. For example, in Fig. 1, we have $N = M^2 = 9$, $L(1) = L(2) = \dots = L(9) = L = 4(M+1) = 16$, $L(1, 5) = 30$, $L(1, 5, 9) = 42$ and $L(1, 2, \dots, 9) = (2M+1)^2 - 1 = 48$. From (5) and (14) the distribution function of v_{\max} is obtained by

$$\begin{aligned} F_{v_{\max}}(t) &= P\{(\forall v_k)(v_k \leq t)\} = P\left\{\bigcap_{k=1}^N V_k\right\} = 1 - P\left\{\bigcup_{k=1}^N \bar{V}_k\right\} \\ &= 1 - \left(\sum_{k=1}^N P\{\bar{V}_k\} - \sum_{\substack{k,m \\ k < m}} P\{\bar{V}_k \cap \bar{V}_m\} \right. \\ &\quad \left. + \sum_{\substack{k,m,n \\ k < m < n}} P\{\bar{V}_k \cap \bar{V}_m \cap \bar{V}_n\} - \dots \right. \\ &\quad \left. + (-1)^{N-1} P\{\bar{V}_1 \cap \bar{V}_2 \cap \dots \cap \bar{V}_N\}\right) \end{aligned}$$

$$\begin{aligned} &= 1 - \left(N \cdot G_x(t)^L - \sum_{\substack{k,m \\ k < m}} G_x(t)^{L(k,m)} \right. \\ &\quad \left. + \sum_{\substack{k,m,n \\ k < m < n}} G_x(t)^{L(k,m,n)} - \dots \right. \\ &\quad \left. + (-1)^{N-1} G_x(t)^{(2M+1)^2-1}\right) \end{aligned} \quad (16)$$

Since $u_{\min} \geq v_{\max}$, the output probability distribution function $F_y(t)$ of the proposed truncation filter is given from (3) by

$$\begin{aligned} F_y(t) &= P\{(x \leq t) \cap (v_{\max} \leq t) \cup (x > t) \cap (u_{\min} \leq t)\} \\ &= P\{x \leq t\}P\{v_{\max} \leq t\} + P\{x > t\}P\{u_{\min} \leq t\} \\ &= F_x(t)F_{v_{\max}}(t) + G_x(t)F_{u_{\min}}(t) \end{aligned} \quad (17)$$

Given the probability p of the positive or negative impulsive noise occurring at the input image, the probability of the impulsive noise occurring at the output of the truncation filter $PB_t(p)$, called breakdown probability [20], can be obtained from (15), (16) and (17) by

$$\begin{aligned} PB_t(p) &= p \left(1 - \left(N \cdot q^L - \sum_{\substack{k_1, k_2 \\ k_1 < k_2}} q^{L(k_1, k_2)} \right. \right. \\ &\quad \left. \left. + \sum_{\substack{k_1, k_2, k_3 \\ k_1 < k_2 < k_3}} q^{L(k_1, k_2, k_3)} - \dots + (-1)^{N-1} q^{(2M+1)^2-1}\right)\right) \\ &\quad + q \left(N \cdot p^L - \sum_{\substack{k_1, k_2 \\ k_1 < k_2}} p^{L(k_1, k_2)} + \sum_{\substack{k_1, k_2, k_3 \\ k_1 < k_2 < k_3}} p^{L(k_1, k_2, k_3)} \right. \\ &\quad \left. - \dots + (1)^{N-1} p^{(2M+1)^2-1}\right) \\ &= p - p \sum_{t=1}^N (-1)^{t-1} \sum_{\substack{k_1, k_2, \dots, k_t \\ k_1 < k_2 < \dots < k_t}} q^{L(k_1, k_2, \dots, k_t)} \\ &\quad + q \sum_{t=1}^N (-1)^{t-1} \sum_{\substack{k_1, k_2, \dots, k_t \\ k_1 < k_2 < \dots < k_t}} p^{L(k_1, k_2, \dots, k_t)} \end{aligned} \quad (18)$$

where $q = 1 - p$. Although it is not easy to calculate the breakdown probability (18) by hand for $W > 7$, it is not difficult to write a computer program to calculate (18) for

Table 1: Probability of the salt-and-pepper noise occurring at the filter's output

p	W	TF	CWM	UMM
0.25	3	0.1641	0.1641	0.1641
	5	0.1293	0.2051	0.0700
	7	0.1179	0.2373	0.0279
	9	0.1188	0.2482	0.0111
0.125	11	0.1269	0.2499	0.0043
	3	0.0504	0.0504	0.0504
	5	0.0247	0.0560	0.0104
	7	0.0144	0.0730	0.0021
0.0625	9	0.0098	0.0930	0.0004
	11	0.0077	0.1095	0.0001
	3	1.40×10^{-2}	1.40×10^{-2}	1.40×10^{-2}
	5	3.77×10^{-3}	1.07×10^{-2}	1.39×10^{-3}
	7	1.17×10^{-3}	1.18×10^{-2}	1.42×10^{-4}
	9	4.23×10^{-4}	1.50×10^{-2}	1.51×10^{-5}
	11	1.79×10^{-4}	2.01×10^{-2}	1.65×10^{-6}

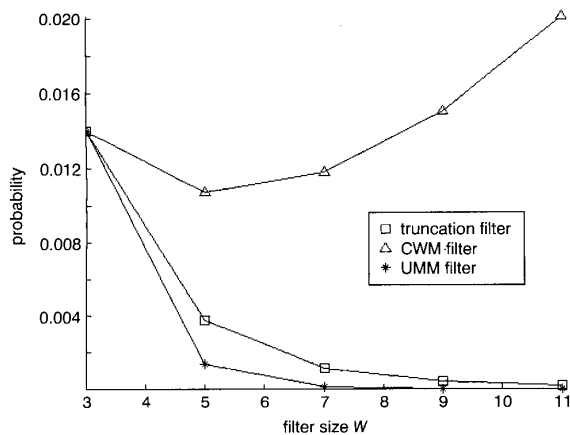


Fig. 2 Probability of salt-and-pepper noise occurring at the output of the truncation filter, the CWM filter and the UMM filter against filter size for $p=0.0625$

various values of p and M and compare them with the breakdown probabilities of other filters.

For comparison, the breakdown probabilities of the CWM filter $PB_c(p)$ and the UMM filter $PB_u(p)$ given in [4, 10] are rewritten here with consistent notation (window size $W \times W$, probability of impulse occurrence p , centre weight c , and $K = W^2$, $q = 1 - p$):

$$PB_c(p) = \sum_{n=(K-c)/2}^{K-1} \binom{K-1}{n} p^{n+1} q^{K-1-n} + \sum_{n=(K+c)/2}^{K-1} \binom{K-1}{n} p^n q^{K-n} \quad (19)$$

$$PB_u(p) = p \left(1 - \left(\sum_{n=(W+1)/2}^{W-1} \binom{W-1}{n} q^n p^{W-1-n} \right)^4 \right) + q \left(\sum_{n=(W+1)/2}^{W-1} \binom{W-1}{n} q^{W-1-n} p^n \right)^4 \quad (20)$$

The breakdown probability $PB(p)$ given above is the probability of an impulse occurring at the output of the filter if the input impulse is either positive or negative with the probability of p . For salt-and-pepper noise, the probability of an impulse occurring at the output of the filter should be $PB(p_1) + PB(p_2)$, where p_1 and p_2 are the occurring probabilities of the input positive and negative impulses, respectively. Table 1 compares the numerical values of the probability of salt-and-pepper noise occurring

at the output of the truncation filter (TF), the CWM filter and the UMM filter for different values of p ($p_1 = p_2 = p/2$) and window size W . By choosing $c = W^2 - W + 1$ for the CWM filter, Table 1 compares the noise attenuating capability of filters that can preserve lines of the same length.

It is clear to see that the truncation filter gives a much better performance in attenuating impulse noise than the CWM filter with the same values of p and W . Fig. 2 illustrates the output noise probability against filter size for $p=0.0625$. Although the truncation filter gives a higher output noise probability than the UMM filter, it should be noted that the truncation filter preserves lines along an arbitrary orientation and even preserves curves, but the UMM filter can only preserve lines along four orientations. An experimental study is therefore necessary to compare their overall filtering performances for both noise attenuation and detail preservation.

4 Experimental studies

Demonstrative experiments were undertaken to test the performance of the proposed truncation filter. Comparisons were made with the CWM filter, the UMM filter and the SM filter. Let $SM(w)$ represent a SM filter with size $W = w$. For all detail-preserving filters, however, a series connection of recursive filters with different sizes was used in the experiments to achieve a better noise attenuation performance. Let $CWM(w)$, $UMM(w)$ and $TF(w)$ represent the series connections of the recursive CWM filters, UMM filters and truncation filters with sizes $W=3, 5, \dots, w$, respectively. The overall filtering performances (including detail-preserving and noise-attenuating performance) of $SM(w)$, $CWM(w)$, $UMM(w)$ and $TF(w)$ were compared by using mean absolute error (MAE) [21]. Two commonly used test images 'Lena' and 'Baboon' of size 512×512 and another image 'Fingerprint' as shown in Fig. 3 were chosen. These three original images were corrupted by salt-and-pepper noise with equal occurrence probabilities of a positive impulse of value 255 and a negative impulse of value 0. Images corrupted by noise with values uniformly distributed between 0 and 225 were tested as well. The various noisy images with different noise densities were generated, and then passed through the filters $TF(w)$, $CWM(w)$, $UMM(w)$ and $SM(w)$ with $w=3, 5, \dots, 31$. Fig. 4 plots MAEs against the value of w for the three test images.

In the first row of Fig. 4, the smallest MAEs are achieved by the truncation filter and the CWM filter. This indicates that for a low noise density, the detail preservation is more



Fig. 3 Original test images 'Lena', 'Baboon' and 'Fingerprint'

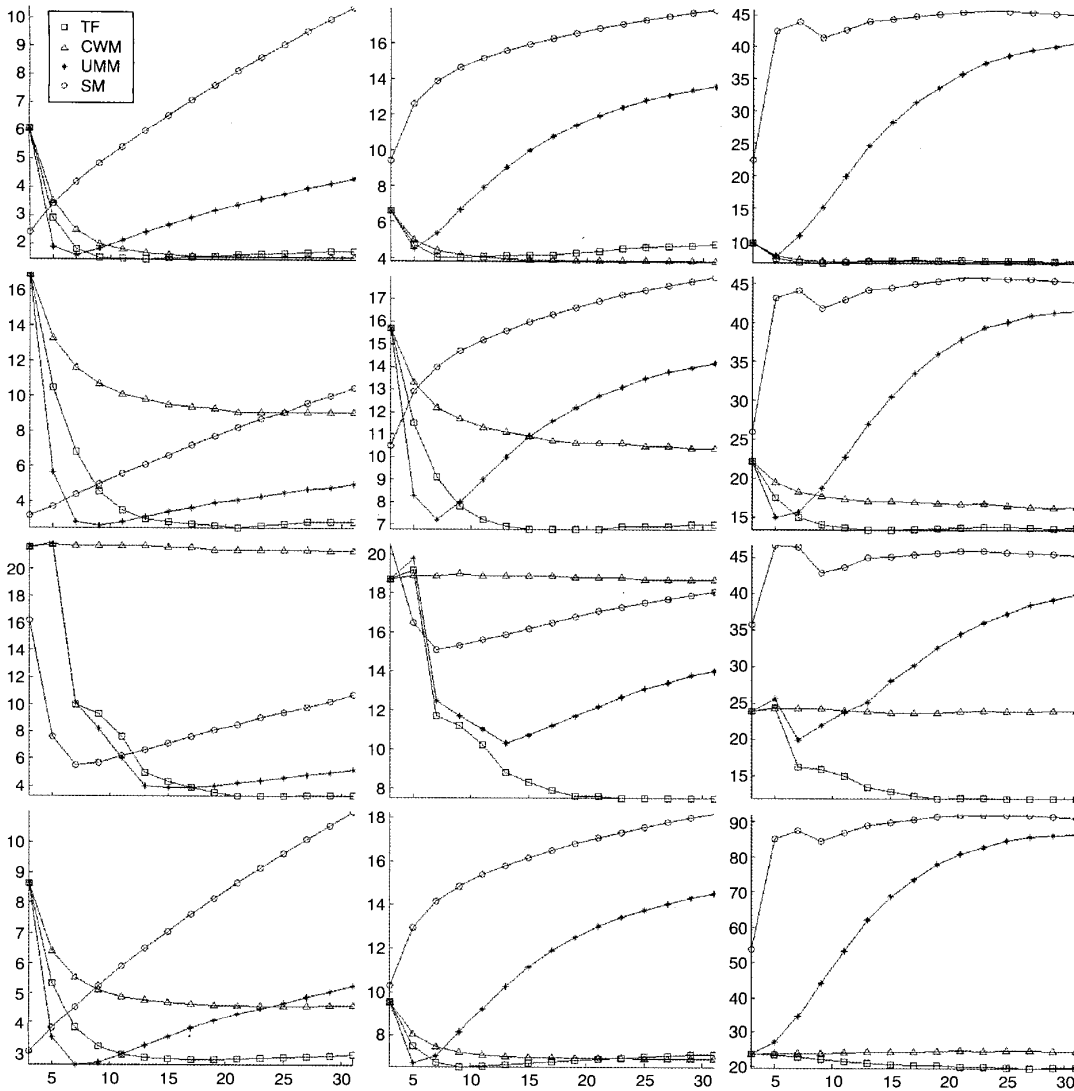


Fig. 4 MAE (y-co-ordinate) against the value of w (x-co-ordinate) of filters TF, CWM, UMM and SM for test images 'Lena' (first column), 'Baboon' (second column) and 'Fingerprint' (third column) corrupted by salt-and-pepper noise with density of 12.5% (first row) and 25% (second row) and corrupted by impulsive blob (3×3) noise with density 20% (third row), and corrupted by noise of 25% density with values uniformly distributed between 0 and 255 (fourth row)

crucial than the noise attenuation for the overall filtering performance. In the second row of Fig. 4, the smallest MAEs are achieved by the proposed truncation filter. As illustrated in these three Figures, for the noise density of 25%, the CWM filter has difficulty in attenuating noise. If several noisy samples are clustered to form a blob, they are more difficult to remove from the image than in the case of an isolated impulse. In some environments, however, the acquisition or transmission of images through sensors or communication channels is affected by impulsive blob noise. In the third row of Fig. 4, 20% pixels of the test images are corrupted by the noise of an impulsive blob of size 3×3 . The fourth row of Fig. 4 plots the results for noise of 25% density with values uniformly distributed between 0 and 255. Compared to the other three approaches, the proposed truncation filter demonstrates its powerful capability in both attenuating impulsive blob noise and preserving image details.

In the runtime test of the filters TF(w), CWM(w), UMM(w) and SM(w), the fast algorithm Quicksort

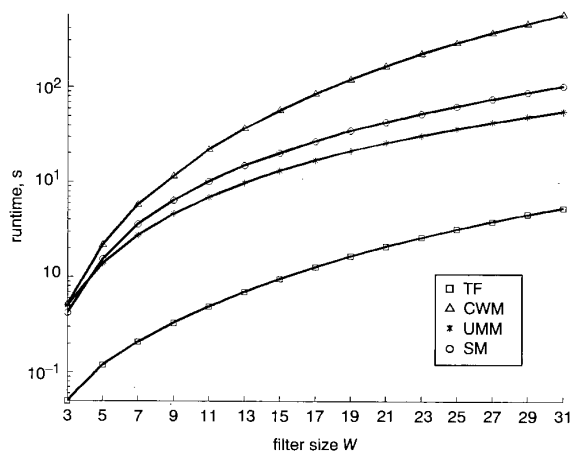


Fig. 5 Runtime against the value of w of filters TF, UMM, SM and CWM for filtering the image 'Lena' of size 512×512

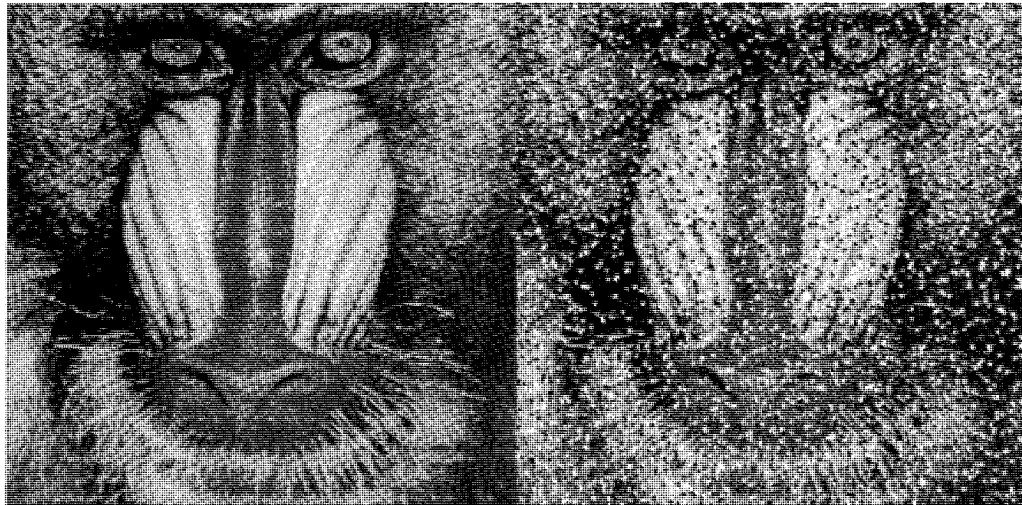


Fig. 6 Original image 'Baboon' (left) and a sample corrupted image by impulsive blob (3×3) noise with density of 20% (right)

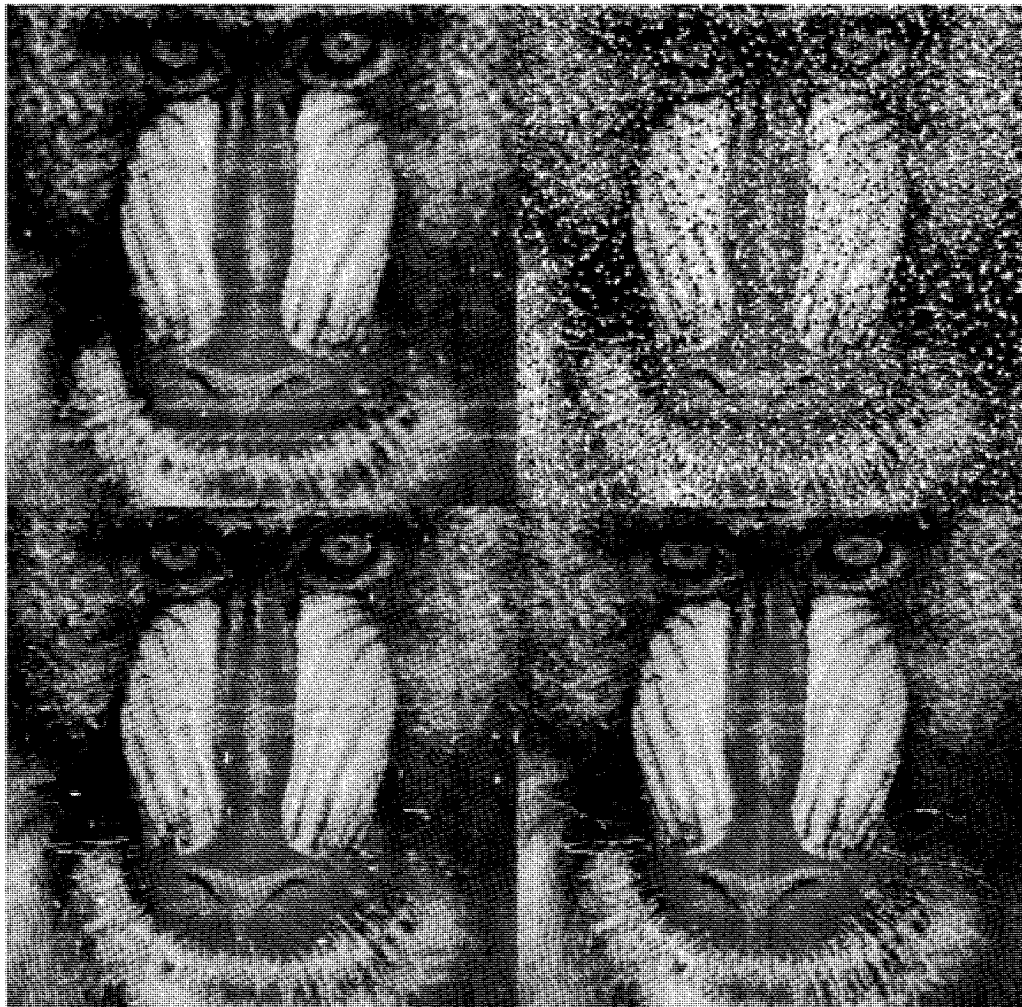


Fig. 7 Output images at the minimal MAE (according to Fig. 4) of filters SM(7) (top left), CWM(31) (top right), UMM(13) (bottom left) and TF(23) (bottom right)

[22] was used for all median calculations. The fast median algorithm [23] was not included in the comparison as it cannot be applied to structured approaches such as multi-stage median filters. In addition, the underlying mechanism of this algorithm can also be incorporated into our truncation filter by throwing away W pixels and adding W new pixels for each window movement. Thus, this algorithm cannot be strictly considered as a competing alternative. All filters were implemented with the C-program running in a Pentium III 733 MHz PC. To measure the runtime accurately, the runtimes for filtering the image 'Lena' corrupted by salt-and-pepper noise with a density of 25% 100 times were recorded and then divided by 100. Fig. 5 illustrates the runtimes against the value of w . It demonstrates that the proposed truncation filter is much faster than all other filters. This is not surprising since the truncation filter uses the maximum and minimum operations instead of the median operation used in other filters. Although the UMM filter only calculates medians of reduced subsets in the window, the proposed truncation filter is still about ten times faster than this second fastest filter. Fig. 6 shows the original image 'Baboon' and a sample image corrupted by 3×3 impulsive blob noise with density 20% and Fig. 7 gives us the corresponding output images at the minimal MAEs (according Fig. 4) of the filters considered in this paper. We see that the proposed truncation filter provides the best visual appearance of the output image.

5 Conclusions

When a filter is applied to an image, there are always two effects with noise attenuation as one and image structure distortion as the other. This work has proposed a novel nonlinear approach, called a truncation filter, to attenuate impulsive noise of images and video sequences while preserving small details. It truncates the grey value of a pixel to the maximal or minimal value of its enclosed surrounding band. Isolated impulsive noise and noise blob inside the band are thus attenuated while image details are preserved as long as they stretch to or beyond the band.

In essence, the proposed method is a cautious way to alter the pixel's grey value while preserving small details such as thin lines and curves, as proven by the filter's deterministic properties. This detail-preserving modification of the grey level can also achieve good noise attenuation as a pixel is compared with multiple bands or equivalently, truncated multiple times with the band at different locations. The study of the filter's statistical properties demonstrates the noise attenuation capability of a single stage, nonrecursive truncation filter. The high processing speed and good detail preservation of the recursive truncation filter makes multiple employments of

the filter with different sizes feasible. Experimental results demonstrate the proposed truncation filter providing a more effective and efficient method for detail-preserving image denoising compared to other median based filters considered.

6 References

- PITAS, I., and VENETSANOPOU, A.: 'Nonlinear digital filters: principles and application' (Kluwer, Norwell, MA, 1990)
- YIN, L., YANG, R., GABBOUJ, M., and NEUVO, Y.: 'Weighted median filters: a tutorial', *IEEE Trans. Circuits Syst.*, 1996, **43**, (3), pp. 157–192
- YANG, R., LIN, T., GABBOUJ, M., ASTOLA, J., and NEUVO, Y.: 'Optimal weighted median filter under structural constraints', *IEEE Trans. Signal Process.*, 1995, **43**, (3), pp. 591–604
- KO, S.J., and LEE, Y.H.: 'Center weighted median filters and their application to image enhancement', *IEEE Trans. Circuits Syst.*, 1991, **38**, (9), pp. 984–993
- CHEN, T., MA, K.K., and CHEN, L.H.: 'Tri-state median filter for image denoising', *IEEE Trans. Image Process.*, 1999, **8**, (12), pp. 1834–1838
- ENG, H.L., and MA, K.K.: 'Noise adaptive soft-switching median filter', *IEEE Trans. Image Process.*, 2001, **10**, (2), pp. 242–252
- XU, Y., and LAL, E.M.: 'Restoration of images contaminated by mixed Gaussian and impulse noise using a recursive minimum-maximum method', *IEE Proc., Vis. Image Signal Process.*, 1998, **145**, (4), pp. 264–270
- SUN, T., and NEUVO, Y.: 'Detail-preserving median based filters in image processing', *Pattern Recognit. Lett.*, 1994, **15**, pp. 341–347
- FLORENCIO, D., and SCHAFFER, R.: 'Decision-based median filter using local signal statistics', *Proc. SPIE-Int. Soc. Opt. Eng.*, 1994, **2308**, pp. 268–275
- ARCE, G.R., and FOSTER, R.E.: 'Detail preserving rank-order based filters for image processing', *IEEE Trans. Acoust. Speech Signal Process.*, 1989, **37**, (1), pp. 83–98
- NIEMINEN, A., HEINONEN, P., and NEUVO, Y.: 'A new class of detail-preserving filters for image processing', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1987, **9**, (1), pp. 74–90
- WANG, X.: 'Generalized multistage median filter', *IEEE Trans. Image Process.*, 1992, **1**, (10), pp. 543–545
- YANG, X., and TOH, P.S.: 'Adaptive fuzzy multilevel median filter', *IEEE Trans. Image Process.*, 1995, **4**, (5), pp. 680–682
- PETRESCU, D., TABUS, I., and GABBOUJ, M.: 'L-M-S filters for image restoration applications', *IEEE Trans. Image Process.*, 1999, **8**, (9), pp. 1299–1305
- WINDYGA, P.S.: 'Fast impulsive noise removal', *IEEE Trans. Image Process.*, 2001, **10**, (1), pp. 173–179
- WERMAN, M., and PELEG, S.: 'Min-max operators in texture analysis', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1985, **7**, (6), pp. 730–733
- SERRA, J.: 'Image analysis and mathematical morphology' (Academic Press, London, England, 1982)
- IMME, M.: 'A noise peak elimination filter', *CVGIP: Graph. Models Image Process.*, 1991, **53**, (2), pp. 204–211
- VINCENT, L.: 'Morphological area opening and closing for greyscale images'. Proceedings of NATO Shape in Picture Workshop, Driebergen, Sept. 1992, (Springer-Verlag, The Netherlands), pp. 197–208
- MALLOW, C.L.: 'Some theory of non-linear smoothers', *Ann. Stat.*, 1980, **8**, pp. 695–715
- COYLE, E.J., LIN, J.H., and GABBOUJ, M.: 'Optimal stack filtering and the estimation and structural approach to image processing', *IEEE Trans. Acoust. Speech Signal Process.*, 1989, **37**, pp. 2037–2066
- HOROWITZ, E., and SAHNI, S.: 'Fundamentals of computer algorithms' (Computer Science Press, New York, 1978)
- HUANG, T.S., YANG, G.J., and TANG, G.Y.: 'A fast two-dimensional median filtering algorithm', *IEEE Trans. Acoust. Speech Signal Process.*, 1979, **27**, pp. 13–18