# A Fast and reliable switching median filter for highly corrupted images by impulse noise

Wei Ping[*], Li Junli[†], Lu Dongming[*], Chen Gang[+]
[*]Department of Computer Science and Engineering, Zhejiang University, Hangzhou, China
Email: weiping@zj139.com
[†]School of Information Science and Engineering of Ningbo University, Ningbo, China
[+]Department of Mathematics of Zhejiang University, Hangzhou, China

***Abstract*- In this paper, we propose a fast and reliable impulse noise filter for highly corrupted images. The median filter was once the most popular nonlinear filter for removing impulse noise because of its good denoising power and computational efficiency, but the performance are unsatisfactory when noise ratio is high. So many algorithms have been proposed to improve the filter result. Recently, A switching median filter with boundary discriminative noise detection(BDND) was proposed, it is very effective and outperforms all previously proposed median-based filters, but it's very time-consuming in calculation. The proposed filter use similar scheme as in BDND, it is also very effective and much more efficient than BDND. The proposed method use a new powerful and efficient noise detection method to determine if current pixel has been corrupted by noise, if it's corrupted, a variable window median filter is used to attenuate impulse noise, while those uncorrupted pixels are remain unchanged. Results from computer simulations are used to demonstrate pleasing performance of our proposed method.**

## I. INTRODUCTION

Digital images may be corrupted by impulse noise in some applications. Attenuate noise is an essential task in digital image processing. The challenge of this task is how to reduce noise while keeping the image details. There are many works on the restoration of images corrupted by impulse noise. The most popular impulse noise filter is median filter[1], but it smears some details and edges of original images especially when the noise level is high. Different remedies of the median filter have been proposed, such as Weighted median filter[2], center weighted median filter[3]. These filters apply median operation to each pixel regardless if the current pixel is contaminated or not, while this operation could produce serious image blurring.

Switch median filters identifies each pixel as "corrupted" or "original", only those corrupted pixels would under go the filtering process, while other pixels remain unchanged. So SM filter can get more satisfactory results for restraining impulse noise. The question is how to detect and separate the corrupted pixels form those incorrupt ones. Earlier research works [4] achieve this objective by a fixed decision-making threshold which is obtain at a pre-assumed noise density level. These noise detection processes often lead to incorrect discrimination between pixel and noise. The noise adaptive soft-switching median (NASM) filter was proposed in [5] to address this issue, The NASM achieves robust performance in removing impulse noise while noise ranging from 10% to 50%. However, for those corrupted images with noise density greater than 50%, the quality of the recovered images become significantly degraded. Recently, Pei-Eng Ng and Kai-Kuang proposed a highly-accurate noise detection algorithm BDND[6]. The BDND can achieve pleasing result even noise density up to 90%. But it is too time-consuming to be used in real applications.

In this paper, we propose an accurate and fast switch median filter, which can effectively remove impulse noise while noise densities ranging from 10% to 90%. The quality of the filtered images is as good as BDND, but the computation cost is very low.

This paper is organized as follows: Section 2 introduces our proposed noise detection method for impulse noise identification. Section 3 describes the filtering scheme in response to the noise detection results. Sections 4 analyze computation complexity of our proposed method compare with the BDND. Section 5 present experiment results, the last section conclude the paper.

This research was supported by the National Science Foundation of China under Grant No.60672072 and No.60432030; Zhejiang Provincial Natural Science Foundation of China under Grant No Y106505; The Special Science and Technology Foundation of Ningbo of China under The Grant No.2005B100016.

## II. IMPULSE NOISE DETECTION

Impulse noise is distributed at both polar of image gray levels, either very high or very low. It seriously damage image details. In this paper, noise detection method is applied to every pixel to decide if the pixel is corrupted by impulse noise. A binary matrix is used to indicate identified results. With '0' indicate uncorrupted, '1' indicate corrupted.

The objective is achieved by two iterations. During each iteration pixels in a square window center around the current pixel are grouped into three regions, low-intensity cluster, median-intensity cluster and high-intensity cluster. The current pixel is identified as "uncorrupted" only if it falls into the median-intensity cluster; otherwise it is assigned as corrupted.

The first iteration employs a window of size 21×21, Instead of sorting the pixels in the local window as in

1-4244-0921-7/07 $25.00 © 2007 IEEE.

BDND, we calculate histogram of the local window, then we can get two boundaries $b_1$ and $b_2$ according to the histogram. The steps of the first iteration are as follows:

Step1) Impose a local window of size 21×21, which is centered around the current pixel.

Step2) Calculate the histogram of the local window and the bin indices are the gray levels. Find the maximum and minimum gray level of the local window, noted as Min and Max respectively.

Step4) For the indices between Min and $(Min+Max)/2$, compute differences of nonzero adjacent bin indices. Find the maximum difference and mark the corresponding index as boundary $b_1$.

Step5) $b_2$ is similarly computed between $(Min+Max)/2$ and Max, three clusters are formed now.

Step6) "uncorrupted" is assigned to the pixel if it belongs to the median cluster; otherwise, "corrupted" is assigned.

Suppose considering image has L gray levels. We use an L size vector of one dimension noted $\{Hist[k]\}_{k=0}^{L-1}$ to record the histogram of the local window. In 8-bits gray images, L=256. The local window sliding from top to bottom, left to right of the considering image. Pixels can be extended symmetrically when local window across the image boundaries.

Calculation of histogram of local window can be speed up by using inherent adjacent information. For better understanding, we use a 3×3 window instead of 21×21 window as an example to illustrate the above steps.

| | | | |
|---|---|---|---|
| 250 | 200 | 100 | 180 |
| 200 | 101 | 60 | 200 |
| 200 | 185 | 50 | 0 |

- Current pixel is 101, Gary level histogram is calculated and only nonzero elements of the vector are listed: { $Hist[50]=1$, $Hist[60]=1$, $Hist[100]=1$, $Hist[101]=1$, $Hist[185]=1$, $Hist[200]=3$, $Hist[250]=1$ }, the maximum Max and minimum Min are 250 and 50 respectively. $(Min+Max)/2$ is 150.
- Check the histogram vector elements $\{Hist[k]\}_{k=Min}^{(Min+Max)/2}$, if $Hist[i]$ and $Hist[j]$ are adjacent nonzero elements, $150 \geq j > i \geq 50$, then the difference of nonzero adjacent bin indices is $j-i$ .the difference vector is {10,40,1}, the maximum difference is 40, which is difference between 60 and 100, and the corresponding index(gray level) is 60.
- Check elements $\{Hist[k]\}_{k=(Min+Max)/2}^{Mzx}$, the difference vector is {15,50}, the maximum difference is 50, which is difference between 250 and 200, and the corresponding index(gray level) is 250.
- Hence b1 = 60, and b2 = 250, pixel x classed into median cluster if $50 < x < 250$, the center pixel 101 is identified as "uncorrupted".
- Local window slides to right and the current pixel becomes 60. Pixels {250,200,200} move out of the window and {180,200,0} move into the window. Histogram can be updated simply by minus those just move out and add those just move into the window. That is $Hist[250]-1$, $Hist[200]-1$, $Hist[200]-1$, and $Hist[180]+1$, $Hist[200]+1$, $Hist[0]+1$ .then the above process repeat to decide if the current pixel is corrupted.

If a pixel is classified as "corrupted", second iteration will be invoked. Histogram of pixel intensity in a 3×3 window are calculated, process to find the boundaries $b_1$ and $b_2$ is just like that in the first iteration we have describe above. The only difference is histogram need to be recalculated when the 3×3 window jump to next pixel identified as "corrupted" in the first iteration, because of no adjacent information can be used in this case. If the pixel still belongs to the middle cluster, it is classified as "corrupted", otherwise, "uncorrupted".

## III. Adaptive Filtering Scheme

Now, we have got a binary matrix to indicate every pixel is corrupted or not, based on the binary decision matrix, those "uncorrupted" pixels are remain unchanged, while switching median filter with adaptive determined window size is applied to those "corrupted".

Starting with window size W = 3, the filtering window extends one pixel in all the four sides of the window provided that the number of uncorrupted ones is less than 3. This is a little different from filtering scheme applied in BDND, where the window size is extended if the number of uncorrupted pixels is less than half of the pixels in the window, denoted by $S_{in} = W \times W / 2$, the extension process stops when uncorrupted pixels exceed $S_{in}$ or window size equal to predefined maximum windows size $W_{Max}$, which is decided through estimation of noise-density level. At high noise-density (>50%), the number of uncorrupted pixels in a square is often less than $S_{in}$ and window size will almost increase until reach the maximum windows size $W_{Max}$, this will blurring image detail severely. Further more, estimation of noise-density level need more calculation than our simple filtering scheme.

Suppose exploiting switch median filter to a noise pixel $x_{i,j}$ and finally decided window size is $W_F$, the output pixel $y_{i,j}$ is:

$$y_{i,j} = median\{x_{i+s,j+k} \mid -(W_F-1)/2 \leq s,k \leq (W_F-1)/2\}$$

Attention that in the median filter process, those corrupted pixels are excluded; only those uncorrupted ones are considered to get median value.

## IV. Computation Complexity

The proposed fast and reliable filter method has noise detection process and noise filter process as described in the above sections. In filter process, if the number of uncorrupted pixels in filter window exceeds 3, extension stopped. So most of the finally decided window size $W_F$ will not larger than 7, even noise density level up to 90%. Since operation of median filter on a small window size is very fast, we will ignore the filter process for a while and pay our attention to the noise detection process.

Consider a M×N size and L gray level image to be processed, the first iteration of our noise detection process use a 21×21 sliding window pass the image pixel by pixel from top to bottom, left to right. At the beginning of each row, histogram of gray levels in the sliding window is computed. This needs 21×21 = 441 times integer plus operations. When the filter window slides to the next right pixel, the histogram is updated by 21 times minus operations and 21 times plus operations, the process repeated until the center pixel of the sliding window reaches right boundary of the image. Then the sliding widow starts from the first pixel of the next row and histogram is recomputed. If we don't distinguish minus operation and plus operation, then the total computation of histogram pass the whole image is 441×M+42×(N-1)×M times plus operations.

The histogram vector $Hist$ has L elements. For each pixel, at most $L-1$ minus operations are needed to compute differences of nonzero adjacent elements. Then we can get two boundaries $b_1$ and $b_2$ to decide if the pixel is corrupted or not. So the computation of the first iteration is at most 441×M+42×(N-1)×M+(L-1)×M×N times integer plus operations, when L = 256 approximated as K ×M×N times integer operations, K≈300.

The second iteration of our noise detection process use 3×3 sliding window, so only 9 times integer plus operation and 8 times minus operation are needed to calculate histogram and intensity differences respectively for each identified "corrupted" pixel in the first iteration. The second iteration needs 17×M×N times integer operations in the worst-case scenario. Together, the detection process is less than 317 times integer plus operations for each pixel.

The BDND noise detection method also employs a sliding window of size 21×21, pixels in the window are sorted according to ascending order and find the median. Then compute the intensity difference between each pair of adjacent pixels, find the maximum difference in the range between 0 and median, the corresponding intensity is the boundary $b_1$, similarly the boundary $b_2$ is identified for pixel intensity between median and 255; if the pixel intensity grater than $b_1$ and less than $b_2$, it's identified as "uncorrupted". If the pixel is classified as "corrupted", then the second iteration will be invoked by the same procedure as in the first iteration except using a 3×3 window. There are 442 pixels in a window of size 21×21 in first iteration, so for each pixel, 441 minus operation needed to compute difference of each pair of adjacent pixels across sorted vector. This is an already larger number of plus operations than our proposed method not to mention sorting operation and the second iteration.

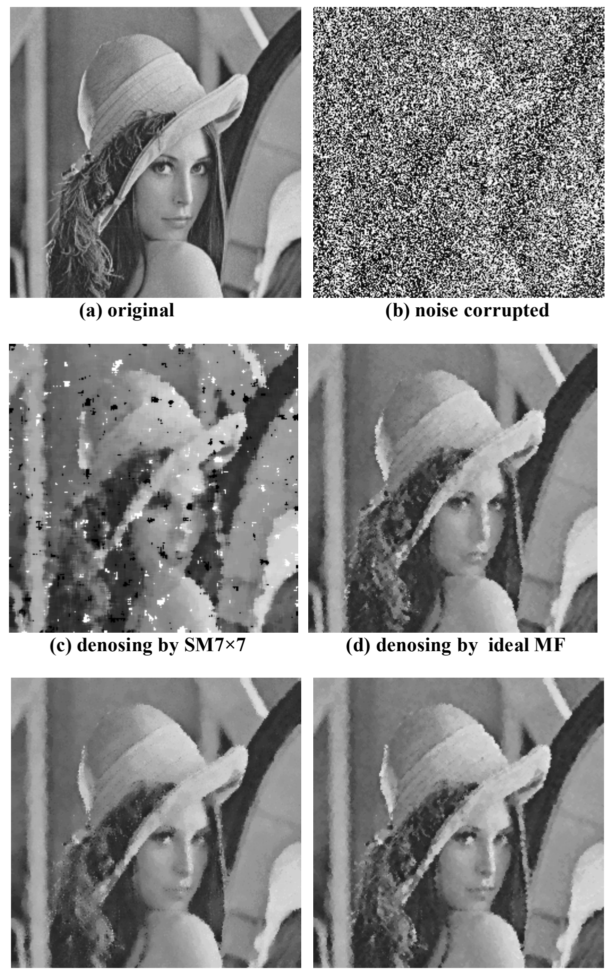
Compare with the BDND, our proposed method has two merits in computation complexity:

- No sorting operation
- In filter scheme, our proposed method don't need to estimate noise-density level, and finally decided filter window size is likely to be smaller than used in the BDND.

## V. Experimental results

We compare performance of our proposed method with the BDND method by evaluate PSNR of the filtered images and runtime consumed.

Among the commonly used 256×256 8-bit gray-scale test images, the image "Lena" is selected for simulations. In our simulation experiments, original images are corrupted by salt-and-pepper noise with equal probability; simulations are carried out under a wide range of noise-density levels (from 10% to 90%).



**(a) original** **(b) noise corrupted**

**(c) denosing by SM7×7** **(d) denosing by ideal MF**

**(e) denosing by BDND** **(f) denosing by proposed method**

**Fig.1 The first row shows original image and corresponding noisy images with 70% impulse noise. The second row presents results of denoising corrupted image by 7×7 standard median filter and ideal median filter which using filtering scheme as stated in section III at the exact positions of impulse noise. The third row are results of BDND and our proposed method.**

In Fig1, we present restoration results for the 70% corrupted Lena image, it can be see that our proposed method can get better performance in terms of noise suppression and detail preservation. The filtering results almost coincided with that of the ideal median filter.

Fig.2 summarizes the PSNR performance comparison of using the BDND switching filter and our proposed method under different noise levels. The two methods can both remove impulse effectively and the PSNR performances are very similar. To clear demonstrate differences of the two methods, difference of the PSNR performance by ours minus BDND is also illustrated as Fig.3, we can see that BDND is better than Our Method above 0.1dB when noise-density less than 0.3, while our proposed method is better than the BDND about 0.2dB when noise density larger than 0.4.

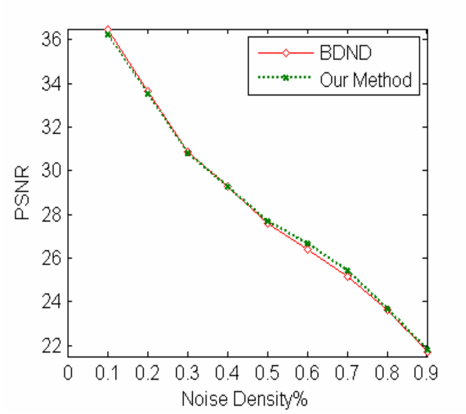



**Fig.2 PSNR performance summarization of the proposed method compare with BDND**

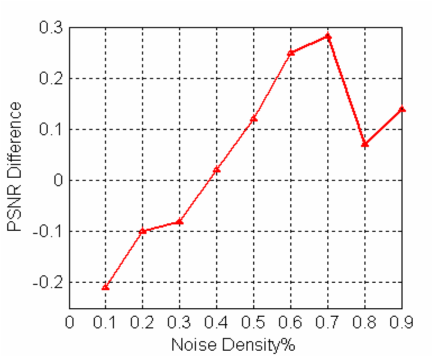



**Fig.3 PSNR difference between BDND and proposed method**

The majority of computation of the BDND consists in the first iteration, where sorting operation is very time-consuming. Quick sort algorithm is widely used and the performance is really good, so we use quick sort algorithm to fulfill the BDND method and compare with our proposed method. The time complexity of quick sort algorithm is $O(N \log N)$.

The runtime analysis of the proposed method, BDND, and 7×7 standard median filter were conducted for "Lena" image using AMD Athlon 1.81GHz Personal Computer and documented in Table I. Results reveal that proposed method is much faster than BDND, it's runtime is comparable the 7×7 SM filter, or even more faster than the 7×7 SM filter when noise density is less than 70%. BDND is the most time-consuming; much more processing time is need than the other two methods.

Fig.4 shows Runtimes (in seconds) of the three methods when denosing corrupted "Lena" image by different impulse noise density levels.

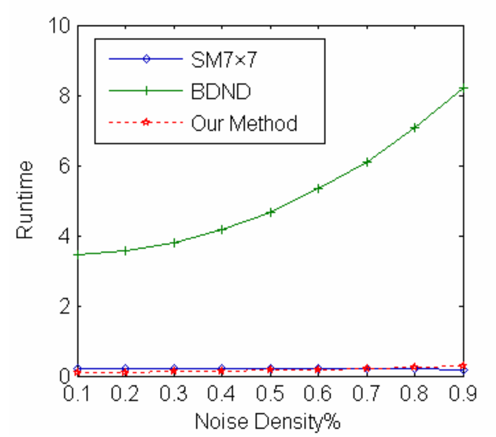



**Fig.4 Runtime of 7×7 standard median filter, BDND filter, Our proposed filter**

**TABLE I**
**Runtime of different filters**

| Filters | p=30% | p=50% | p=70% | p=90% |
|---|---|---|---|---|
| 7*7filter | 0.201 | 0.195 | 0.190 | 0.188 |
| BDND | 3.80 | 4.66 | 6.11 | 8.22 |
| Our proposed | 0.11 | 0.157 | 0.20 | 0.296 |

## VI. Conclusions

In this paper, we propose a fast and reliable switch median filter for impulse noise, experimental results show that our method performs better than the BDND filter both in PSNR and runtime evaluations, especially at high noise level. The tremendous advantage of the proposed method is that it is really simple and can be realized even faster than 7×7 standard median filter, but its PSNR performance is fairly good, which is overwhelming existing nonlinear impulse noise filters.

## References


[1] T. S. Huang, G. J. Yang, and G. Y. Tang, "Fast two-dimensional median filtering algorithm," IEEE Trans. Acoustics, Speech, Signal Process.,vol. ASSP-1, no. 1, pp. 13–18, Jan. 1979.

[2] R. Yang, L. Lin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filters under structural constraints," IEEE Trans.Signal Processing, vol. 43, pp. 591–604, Mar. 1995.

[3] T. Song, M. Gabbouj, and Y. Neuvo, "Center weighted median filters: some properties and applications in image processing," Signal Processing, vol. 35, no. 3, pp. 213–229, 1994.

[4] D. A. F. Florencio and R.W. Shafer, "Decision-based median filter using local signal statistics," in Proc. SPIE Symp. Visual Communication and Image Processing, vol. 2308, Sept. 1994, pp. 268–275

[5] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," IEEE Trans. Image Processing, vol. 10, pp. 242–251, Feb. 2001.

[6] PE Ng, KK Ma, "A Switching Median Filter With Boundary Discriminative Noise Detection for Extremely Corrupted Images", IEEE Trans. Image Processing, VOL. 15, NO. 6, June. 2006