A Detection Statistic for Random-Valued Impulse Noise

Yiqiu Dong, Raymond H. Chan, and Shufang Xu

***Abstract*—This paper proposes an image statistic for detecting random-valued impulse noise. By this statistic, we can identify most of the noisy pixels in the corrupted images. Combining it with an edge-preserving regularization, we obtain a powerful two-stage method for denoising random-valued impulse noise, even for noise levels as high as 60%. Simulation results show that our method is significantly better than a number of existing techniques in terms of image restoration and noise detection.**

***Index Terms*—Edge-preserving regularization, image denoising, noise detector, random-valued impulse noise.**

I. Introduction

DIGITAL images are often corrupted by impulse noise due to transmission errors, malfunctioning pixel elements in the camera sensors, faulty memory locations, and timing errors in analog-to-digital conversion [1]. An important characteristic of this type of noise is that only part of the pixels are corrupted and the rest are noise free. In most applications, denoising the image is fundamental to subsequent image processing operations, such as edge detection, image segmentation, object recognition, etc. The goal of noise removal is to suppress the noise while preserving image details. To this end, a variety of techniques have been proposed to remove impulse noise.

One of the most popular methods is the median filter [2], which can suppress noise with high computational efficiency. However, since every pixel in the image is replaced by the median value in its neighborhood, the median filter often removes desirable details in the image and blurs it, too. The weighted median filter [3] and the center-weighted median filter [4] were proposed as remedy to improve the median filter by giving more weight to some selected pixels in the filtering window. Although these two filters can preserve more details than the median filter, they are still implemented uniformly across the image without considering whether the current pixel is noise free or not.

Over the years, better noise removal methods with different kinds of noise detectors have been proposed, for example, switching median (SM) filter [5], multistate median (MSM) filter [6], adaptive center weighted median (ACWM) filter [7], the peak-and-valley filter [8], [9], signal-dependent rank-order mean (SD-ROM) filter [10], conditional signal-adaptive median (CSAM) filter [11], the pixel-wise MAD (PWMAD) filter [12], modified threshold Boolean filter (TBF) [13], Jarque–Bera test based median (JM) filter [14], two-output nonlinear filter [15], iterative median filter [16], etc. With the noise detector, these filters will check each pixel if it is corrupted or not. Then, only noisy pixels are changed so as to avoid undue distortion. The main drawback of these filters is that they just use median values or their variations to restore the noisy pixels, and hence they usually cannot preserve the image details even when the images are mildly corrupted, say with noise ratio less than 30%; see (1) for the definition of noise ratio.

In the last few years, many methods based on fuzzy techniques have been developed for noise removal, too [17], [18]. The Fuzzy system is well suited to model the uncertainty that occurs when both noise removal and detail preservation are required. However, when the images are highly corrupted, discovering the rulebase structure becomes quite difficult. In order to overcome this difficult, many methods based on neuro-fuzzy system are proposed [19]–[23], which make full use of the ability of neural networks to learn from examples. With suitable and sufficient training, they can preserve the image details during noise removal.

Recently, an edge-preserving regularization method has been proposed to remove impulse noise [24]. It uses a nonsmooth data-fitting term along with edge-preserving regularization. In order to improve this variational method in removing impulse noise, a two-stage method was proposed in [25] and [26]. In the first phase, it uses either the adaptive median (AM) filter [27] or the ACWM filter [7] to identify pixels which are likely to be noise (noise candidates). In the second phase, these noise candidates are restored by the edge-preserving regularization which allows edges and noise-free pixels to be preserved. This method, denoted by AM-EPR and ACWM-EPR, can restore large patches of noisy pixels because it introduces pertinent prior information via the regularization term. It is most efficient in dealing with high noise ratio, e.g., ratio as high as 90% for salt-and-pepper impulse noise and 50% for random-valued impulse noise can be removed by AM-EPR and ACWM-EPR, respectively; see [25] and [26]. Its capability is mainly limited by the accuracy of the noise detector in the first phase. The aim of this paper is to find a better noise detector for the two-stage method, especially for the random-valued impulse noise.

In [28], Garnett *et al.* introduced a new local image statistic called ROAD to identify the impulse noisy pixels and incorporated it into a filter designed to remove additive Gaussian

Manuscript received February 28, 2006; revised October 21, 2006. This work was supported in part by RFDP (20030001103), in part by the NSFC (10571007) of China, and in part by HKRGC Grants CUHK 400503 and CUHK DAG 2060257. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley J. Reeves.

Y. Dong and S. Xu are with the School of Mathematical Sciences, Peking University, Beijing 100871, China (e-mail: dyiqiu@math.pku.edu.cn; xsf@math.pku.edu.cn).

R. H. Chan is with the Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: rchan@math.cuhk.edu.hk).

Digital Object Identifier 10.1109/TIP.2006.891348

noise. The result is a trilateral filter capable of removing mixed Gaussian and impulse noise. This method also performs well for removing impulse noise. However, when the noise level is high, it blurs the images seriously.

Our new local image statistic, and hence a new noise detector is based on ROAD. Our idea is to amplify the differences between noisy pixels and noise-free pixels in ROAD so that the noise detection can be more accurate. We use this new statistic in phase one of the two-stage method [25], [26] to detect the noisy candidate pixels and utilize again the edge-preserving regularization method in [24] and [29] in the second phase. We have compared our method, denoted by ROLD-EPR, with a number of methods. It outperforms the others in both image restoration and noise detection. In particular, when the random-valued impulse noise ratio is as high as 60%, it still can remove most of the noise while preserving image details.

The outline of this paper is as follows. In Section II, we define the new statistic. Section III describes our noise removal method in detail. Section IV gives simulation results to demonstrate the performance of the new method. Finally conclusions are drawn in Section V.

II. ROLD Statistic for Detecting Impulse Noise

When an image is corrupted by impulse noise, only part of the pixels are changed. To be precise, let $x_{i,j}$ and $y_{i,j}$ be the pixel values at location (i,j) in the original image and the noisy image, respectively. Let the dynamic range of the image be $[s_{\min}, s_{\max}]$. If the noise ratio is p, then

$$y_{i,j} = \begin{cases} x_{i,j}, & \text{with probability } 1-p \\ n_{i,j}, & \text{with probability } p \end{cases} \tag{1}$$

where $n_{i,j}$ is the gray-level value of the noisy pixel. There are two models of impulse noise: the easier-to-restore *salt-and-pepper noise* where $n_{i,j} = s_{\min}$ or $s_{\max}$, and the more difficult *random-valued impulse noise* where $n_{i,j}$ are uniformly-distributed random numbers in $[s_{\min}, s_{\max}]$. Here, for simplicity we let the dynamic range of the images be [0, 1]. Since there are excellent noise detectors such as the peak-and-valley [8] and AM [27] filters for detecting salt-and-pepper noise even for noise ratio as high as 90%, in this paper, we only focus on the detection and denoising of random-valued impulse noise.

A. Review on ROAD

First, we briefly review the statistic, called ROAD, proposed in [28]. Let Ω_N denote the set of coordinates in a $(2N+1) \times (2N+1)$ window centered at (0,0), i.e.

$$\Omega_N = \{(s,t) | -N \le s,t \le N\}$$

and let $\Omega_N^0 = \Omega_N \backslash (0,0)$. Define d_{st} as the absolute difference between the gray-level values $y_{i+s,j+t}$ and $y_{i,j}$, i.e.

$$d_{st}(y_{i,j}) = |y_{i+s,j+t} - y_{i,j}|, \quad \forall (s,t) \in \Omega_N^0. \tag{2}$$

Sort the d_{st} values in increasing order, and let r_k be the kth smallest one amongst them. Then define

$$\mathrm{ROAD}_m(y_{i,j}) = \sum_{k=1}^{m} r_k(y_{i,j}) \tag{3}$$

where $2 \le m \le (2N+1)^2 - 2$.

As an example, if we set N to 1 and m to 4, ROAD provides a measure of how close the current pixel value is to its four closest neighbors in a 3×3 window. We note that noisy pixels should have intensities vary greatly from those of its neighbors (i.e., their ROAD values will be large), whereas noise-free pixels should have at least half of the neighbors having similar intensity (i.e., their ROAD values will be small), even for pixels on the edges; see [28]. Thus, one can use ROAD to detect impulse noise, i.e., if the ROAD value of a pixel is greater than a certain fixed threshold, then we consider it as a noisy pixel; otherwise the pixel is considered noise free.

In [28], it is suggested that we use the 5×5 windows and $m = 12$ if the noise ratio is higher than 25%, and use the 3×3 windows and $m = 4$ if otherwise.

B. Definition of ROLD

The ROAD is already a good statistic. However, for random-valued impulse noise, some noise values may be close to their neighbors' values, in which case, the ROAD value of the pixel may not be large enough for it to be distinguished from the noise-free pixels. Thus, one way to improve the ROAD statistic is to find a way to increase these ROAD values, and yet keep the small ROAD values from increasing much. Here, we use a logarithmic function to realize this goal.

Using the logarithmic function on the absolute difference d_{st} defined in (2), we get

$$\tilde{D}_{st}(y_{i,j}) = \log_a |y_{i+s,j+t} - y_{i,j}|, \quad \forall (s,t) \in \Omega_N^0.$$

Clearly for any $a > 1$, the number $\tilde{D}_{st}$ is always in $(-\infty, 0]$. In order to keep it in the dynamic range [0,1], we use a truncation and a linear transformation

$$D_{st}(y_{i,j}) \equiv 1 + \max\{\log_a |y_{i+s,j+t} - y_{i,j}|, -b\}/b \quad \forall (s,t) \in \Omega_N^0$$

where a, b are positive numbers to be chosen. Note that the value of a controls the shape of the curve of the logarithmic function and the value of b decides the truncation position.

The selections of the numbers a and b have great effects on the accuracy of our detection. To choose them properly, consider the function $h_{a,b}(x)$ defined as

$$h_{a,b}(x) = 1 + \max\{\log_a x, -b\}/b, \quad (x \ge 0) \tag{4}$$

we can prove the following property.

Theorem 1: Let $h_{a,b}$ be defined as in (4), and let (a_1, b_1) and (a_2, b_2) be two pairs of numbers with $a_1^{b_1} = a_2^{b_2}$, then we have $h_{a_1,b_1}(x) = h_{a_2,b_2}(x)$ for all $x \ge 0$.

Proof: It follows from $a_1^{b_1} = a_2^{b_2}$ that

$$\frac{1}{b_1}\log_{a_1} x = \frac{\ln x}{b_1 \ln a_1} = \frac{\ln x}{b_2 \ln a_2} = \frac{1}{b_2}\log_{a_2} x.$$

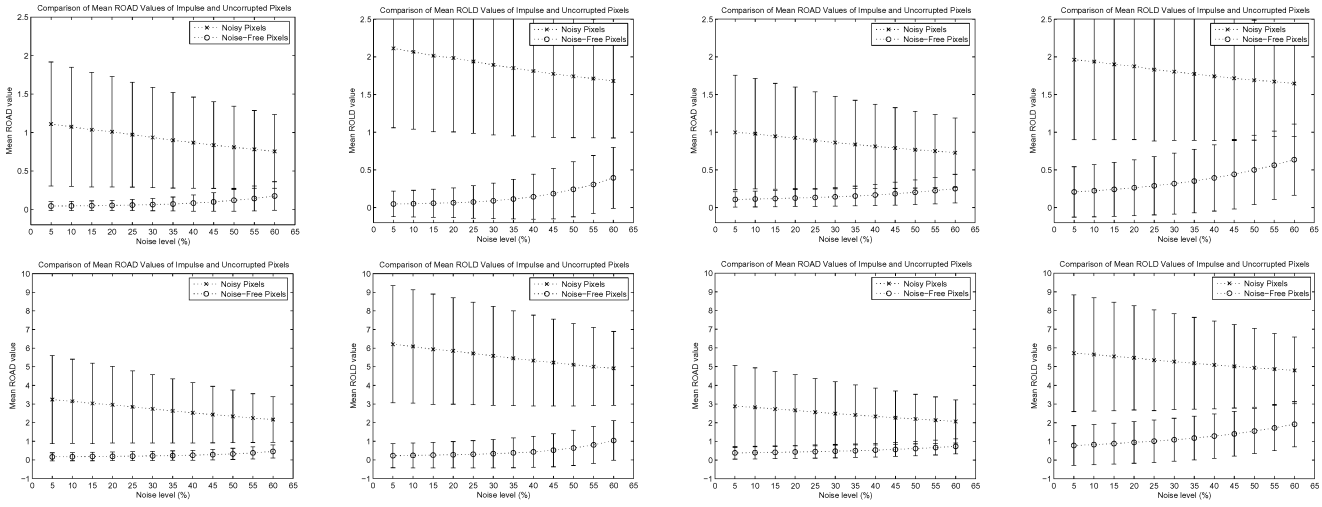


Fig. 1. Error-bar charts for the statistics on (a), (b) "Lena" and (c), (d) "Bridge" with different choices of N and m: $N = 1$, $m = 4$ in the first row and $N = 2$, $m = 12$ in the second row. (a) ROAD; (b) ROLD; (c) ROAD; (d) ROLD.

Hence

$$1 + \max\left\{\frac{1}{b_1}\log_{a_1} x, -1\right\} = 1 + \max\left\{\frac{1}{b_2}\log_{a_2} x, -1\right\}.$$

■

From the definition of $h_{a,b}(x)$, we find that the range of x for $h_{a,b}(x)$ to be zero is $[0, 1/a^b]$. Combining with Theorem 1, the main problem of choosing a and b (or more precisely a^b) is to decide this range. This is just another good reason for using the logarithmic function: it converts the problem about the shape of the transformation function into one about the truncation point. Recall that x denotes the absolute difference between a pixel value and that of its neighbors. In general, for an 8-bit gray-level image, if this absolute difference is less than 8, it is not noticeable [30]. So, we set $h_{a,b}(x) = 0$ in [0,(8/255)]. Since $8/256 = (1/2)^5$, we choose $a = 2$ and $b = 5$. Accordingly, we define $D_{st}(y_{i,j})$ as

$$D_{st}(y_{i,j}) = 1 + \max\left\{\log_2 |y_{i+s,j+t} - y_{i,j}|, -5\right\}/5 \qquad \forall (s,t) \in \Omega_N^0.$$

Arrange all D_{st} in an increasing order, and let R_k be the kth smallest D_{st} for all $(s,t) \in \Omega_N^0$. Like ROAD, we define our local image statistic as

$$\mathrm{ROLD}_m(y_{i,j}) = \sum_{k=1}^{m} R_k(y_{i,j}).$$

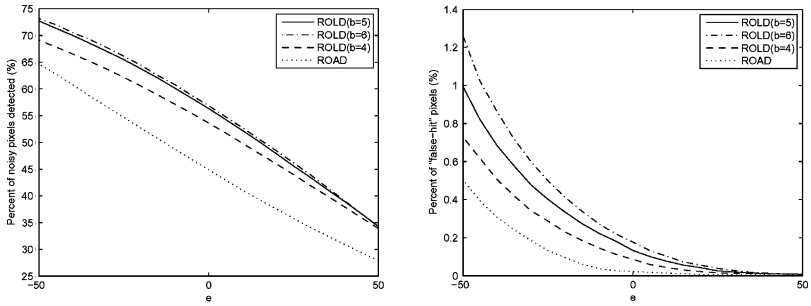
We name this statistic Rank-Ordered Logarithmic Difference (ROLD for short). With ROLD, we can define a noise detector by employing a threshold T: a pixel $y_{i,j}$ is detected as noisy if $ROLD_m(y_{i,j}) > T$, and noise free if otherwise.

C. Comparison of ROAD and ROLD

To demonstrate that ROLD is a better statistic than ROAD, we test them on two 512×512 images. One is a homogeneous image "Lena," and the other is the image "Bridge" which has many fine details. Here, we suppose the locations of all noisy pixels are known in advance, then all pixels can be grouped into two sets: the noisy pixel set and the noise-free pixel set.

In Fig. 1, we show the error-bar charts for the statistics on the images with different choices of N and m given in [28]. The four charts in the first row are with $N = 1$ (3×3 windows) and $m = 4$, the ones in the second row are with $N = 2$ (5×5 windows) and $m = 12$. In all these figures, the upper dashed lines represent the mean ROAD [Fig. 1(a) and (c)] and ROLD [Fig. 1(b) and (d)] values for the noisy pixels, and the lower dashed lines represent those of the noise-free pixels. The error bars represent the standard deviations of ROAD and ROLD values for the two sets. The height of the bars tells how tightly the values are clustered around the means. We can see from the figures that the distance between the means and the separation between the error bars all increase in our statistic ROLD, and hence it will be easier to separate noisy pixels from noise-free pixels. Thus, the noise detection by our statistic will be more accurate. In addition, we see that for highly corrupted images, the statistics using 5×5 windows perform better than the ones using 3×3 windows, a fact pointed out in [28] already.

In order to illustrate more clearly that it is easier to separate noisy pixels from noise-free pixels by ROLD, we perform another experiment. Consider the "Lena" image corrupted by 50% random-valued impulse noise. We test the detection capability of the two statistics using 5×5 windows with thresholds T_e of the form: $T_e = \mu + e \cdot \sigma$, where μ and σ are the mean and the standard deviation of statistic values for all noisy pixels, and $e \in [-0.5, 0.5]$. In Fig. 2, we plot the percentage of the noisy pixels identified correctly, and the percentage of the noise-free pixels identified as noise ("false-hit" pixels). Here, we tried three different values of b for ROLD (as noted in Theorem 1, the ROLD is dependent only on a^b). It is easy to see that ROLD performs better than ROAD. For example, ROLD with $b = 5$ can identify roughly 6% more noisy pixels than ROAD can, yet the "false-hit" percentages differ by no more than 0.5%. Among ROLD with different b, the ROLD with $b = 5$ is the best,



(a) (b)

Fig. 2. Detection results by two statistics for the image "Lena" corrupted by 50% random-valued impulse noise. (a) Detection rate; (b) false-hit rate.

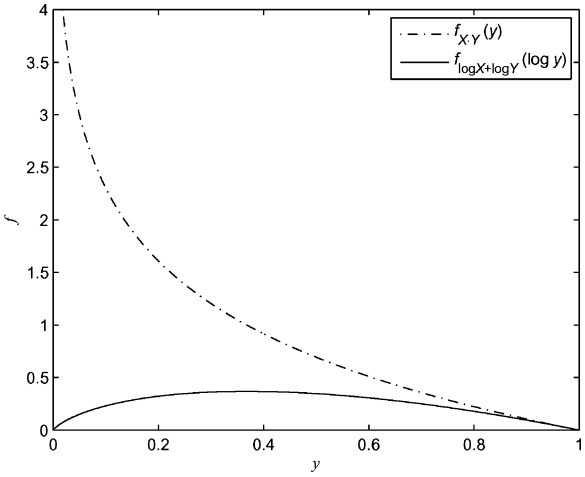


Fig. 3. Probability density function graphs.

which can identify much more noise than ROLD ($b = 4$) with less "false-hit" pixels than ROLD ($b = 6$).

D. Multiplication Versus Logarithm

Although $\log X \cdot Y = \log X + \log Y$, we cannot simply replace the summation in (3) by $\prod_{k=1}^{m} r_k(y_{i,j})$ to get a better statistic. Here, we use a simple model to explain this point. Suppose X and Y are independent and both uniformly distributed in [0, 1], it is easy to calculate the probability densities of $X \cdot Y$ and $\log X + \log Y$

$$f_{X\cdot Y}(y) = -\log y, \quad y \in [0,1],$$
$$f_{\log X+\log Y}(z) = -ze^{z}, \quad z \in (-\infty, 0].$$

Since these two densities are defined on different domains: [0,1] and $(-\infty, 0]$, in order to compare them, we use a transformation $z = \log y$ for $(\log X + \log Y)$ so that y is defined on (0,1]. In Fig. 3, we plot the graphs of $f_{X\cdot Y}(y)$ and $f_{\log X+\log Y}(\log y)$. It is obvious that multiplication will lead to the statistic values very concentrated, especially near 0. Hence, the impulse noise will be difficult to distinguish. However, the logarithmic function makes the distribution much more dispersed, so that the noisy pixels can be distinguished from the noise-free ones more easily.

III. Our Method

After the noise detection, most nonlinear methods replace the noisy pixels by the median filter or its variants without considering features in the images such as the possible presence of edges. In [24] and [29], Nikolova used edge-preserving regularization to remove impulse noise. This method is not good because the regularization is applied to all pixels in the image, even if they are noise free.

Recently, a two-stage iterative method for removing random-valued impulse noise is proposed in [26]. In the first phase, they use the ACWM filter [7] to identify pixels which are likely to be corrupted (noise candidates). In the second phase, these noise candidates are restored by using the edge-preserving regularization in [24] and [29]. The method, denoted by ACWM-EPR, can restore noise as high as 50%, and its capability is mainly limited by the accuracy of the noise detector ACWM in the first phase.

Since now we have a good noise detector ROLD, we can combine it with the edge-preserving regularization in the two-stage method. We denote the new method ROLD-EPR method. To ensure high accuracy of detection, it is executed iteratively with decreasing thresholds, see [26]. At early iterations, with large thresholds, ROLD will identify pixels that are most likely to be noisy. In the subsequent iterations, we decrease the threshold to include more noise candidates. Suppose the noisy image is $\mathbf{y}$. Our algorithm is as follows.

Algorithm ROLD-EPR:

Step 1) Set $k = 0$ and $\mathbf{u}^{(0)} = \mathbf{y}$.

Step 2) (Noise detection) If $\mathrm{ROLD}(u_{i,j}^{(k)}) > T_k$, then $u_{i,j}^{(k)}$ is noise, and $(i,j) \in \mathcal{N}_k$, the noise candidate set; otherwise, $u_{i,j}^{(k)}$ is noise free.

Step 3) (Noise restoration) Restore all pixels in $\mathcal{N}_k$ by minimizing the following functional [31]:

$$\sum_{(i,j)\in\mathcal{N}_k}\left(\sum_{(m,n)\in\mathcal{V}_{i,j}\cap\mathcal{N}_k}\varphi\left(u_{i,j}^{(k+1)}-u_{m,n}^{(k+1)}\right)+2\sum_{(m,n)\in\mathcal{V}_{i,j}\setminus\mathcal{N}_k}\varphi\left(u_{i,j}^{(k+1)}-u_{m,n}^{(k)}\right)\right) \quad (5)$$

where $\mathcal{V}_{i,j}$ is the set of the four closest neighbors of (i,j), and φ is an edge-preserving potential function [32]. For all $(i,j) \notin \mathcal{N}_k$, take $u_{i,j}^{(k+1)} = u_{i,j}^{(k)}$.

Step 4) Stop the iteration as soon as k is larger than $K_{\max}$, the maximum number of iterations. Otherwise, set $k = k + 1$, and go back to Step 2).

TABLE I
COMPARISON OF RESTORATION RESULTS IN PSNR (dB) FOR IMAGES CORRUPTED WITH RANDOM-VALUED IMPULSE NOISE

Method	"Baboon" image			"Bridge" image			"Lena" image			"Pentagon" image		
	20%	40%	60%	20%	40%	60%	20%	40%	60%	20%	40%	60%
Median Filter [2]	22.52	20.65	19.36	25.04	22.17	19.36	32.37	27.64	21.58	28.29	25.16	23.41
Switching Scheme I [5]	23.67	20.85	19.27	26.26	22.66	19.13	32.93	27.90	20.61	29.34	26.26	23.90
Switching Scheme II [5]	22.46	21.35	19.42	25.90	22.85	19.04	33.43	27.75	20.61	28.28	26.43	23.85
SD-ROM Filter (M=2, without training) [10]	23.70	21.33	19.33	27.04	23.33	19.43	35.29	28.59	21.64	30.25	26.54	22.82
SD-ROM Filter (M=1296, with training) [10]	23.81	21.49	19.45	26.56	23.80	20.66	35.71	29.85	23.41	30.38	27.27	24.33
PSM Filter [35]	23.43	21.07	19.56	26.33	22.75	19.73	35.09	28.92	22.06	29.18	26.19	23.87
TSM Filter [36]	23.73	21.38	19.44	26.52	22.89	19.60	34.21	28.30	21.67	29.29	26.29	23.59
MSM Filter [6]	24.02	21.52	19.63	27.27	23.55	20.07	35.44	29.26	22.14	30.34	27.04	24.22
ACWM Filter [7]	24.17	21.58	19.56	27.08	23.23	19.27	36.07	28.79	21.19	30.23	26.84	23.50
PWMAD Filter [12]	23.78	21.56	19.68	26.90	23.83	20.83	36.50	31.41	24.30	30.11	27.33	24.46
Luo, Iterative Median Filter [16]	24.18	21.41	19.08	27.05	23.88	19.74	36.90	30.25	22.96	30.42	26.93	23.72
ROAD-Trilateral Filter [28]	24.18	21.60	19.52	27.60	24.01	20.84	36.70	31.12	26.08	30.33	27.14	24.60
ACWM-EPR [26]	23.97	21.62	19.87	27.31	24.60	20.89	36.57	32.21	24.62	30.03	27.35	24.59
ROAD-EPR	24.24	21.53	19.96	27.42	24.52	22.04	36.79	32.32	28.37	30.35	27.06	25.00
ROLD-EPR	*24.49*	*21.92*	*20.38*	*27.86*	*24.79*	*22.59*	*37.45*	*32.76*	*29.03*	*30.73*	*27.73*	*25.70*

We note that, in (5), we only have the regularization term and no data-fitting term. It is because the data are fitted exactly for uncorrupted pixels; see [31].

When computing ROLD values, we follow the rules in [28]: if the noise ratio is higher than 25%, we use 5×5 windows and $m = 12$; otherwise, we use 3×3 windows and $m = 4$. For the edge-preserving potential function, we select $\varphi(t) = |t|^{1.3}$ as in [26]. To find the minimizer of (5), we use the global BB method proposed in [33] so as to improve the computational efficiency.

The remaining problem is to select the thresholds T_k. In order to identify the noise as much as possible with high accuracy, T_0 should be close to the mean ROLD value for the noisy pixels, which gives the central tendency of the impulse noise. From Fig. 1, we see that with 3×3 windows and noise ratio $p \leq 25\%$, the mean ROLD values for the noisy pixels are always in the range [1.7, 2.1]; and with 5×5 windows and $p > 25\%$, the mean ROLD values for the noisy pixels are in [4.9, 5.9]. We, therefore, define the initial threshold as $T_0 = s \cdot q$, where $s \in [1.7, 2.1]$ if $p \leq 25\%$ or $s \in [4.9, 5.9]$ if $p > 25\%$, and q is the proportion of the pixels in the images whose ROLD values are less than s. In the subsequent iterations, we let $T_k = T_{k-1} \cdot q$. Extensive simulations conducted on standard test images [34] indicate that with s in the specified range, our method is robust against the choice of s; see Section IV-C. Hence, in our simulations, we use $s = 1.9$, the midpoint of [1.7, 2.1] for $p \leq 25\%$, and $s = 5.4$, the midpoint of [4.9, 5.9] for $p > 25\%$. In both cases, $K_{\max} = 7$ is enough.

IV. SIMULATIONS

In this section, we compare the image restoration and noise detection capability of our method with a number of methods that are capable of removing random-valued impulse noise. We have tried many commonly used images, and our method has out-performed all other methods we tested. For illustrations, the results for 512×512, 8-bit gray-level images "Baboon," "Bridge," "Lena," and "Pentagon" are presented here. Complete results can be found in http://www.math.cuhk.edu.hk/~rchan/paper/dcx/.

A. Comparison of Image Restoration

The performance of all methods are compared quantitatively by using the peak signal-to-noise ratio (PSNR) [30] and the 2-D correlation (COR) [22] which are defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (u_{i,j} - x_{i,j})^2} \ (\text{dB})$$

$$\text{COR} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (u_{i,j} - \bar{u})(x_{i,j} - \bar{x})}{\sqrt{\left(\sum_{i=1}^{M} \sum_{j=1}^{N} (u_{i,j} - \bar{u})^2\right)\left(\sum_{i=1}^{M} \sum_{j=1}^{N} (x_{i,j} - \bar{x})^2\right)}}$$

where $u_{i,j}$ and $x_{i,j}$ denote the pixel values of the restored image and the original image, respectively, $\bar{u}$ and $\bar{x}$ denote the averages of all pixel values in the images, and the image size is $M \times N$. For both criteria, larger values signify better image restoration.

In Tables I and II, we list the best PSNR and COR values from all methods for the four images with different noise ratios. The best values for other methods are underlined so that they can be compared easily with the values from our method. Although the trilateral filter together with ROAD statistic [28] is for removing mixed Gaussian and impulse noise, it is already an excellent method for impulse noise removal. Therefore, we also list its results here for comparison, and use the parameter values suggested in [28]. For fair comparison, we also tried the ROAD statistic as the noise detector in the first phase of the two-stage method (denoted by ROAD-EPR in the tables). For this, we need to define s as in Section III. From Fig. 1, we see that the mean ROAD values are near 1 for 3×3 windows when $p \leq 25\%$ and near 2.4 for 5×5 windows when $p > 25\%$. Therefore, we set $s = 1$ for $p \leq 25\%$ and $s = 2.4$ for $p > 25\%$ when using ROAD-EPR method. It is clear from the tables that for all the images and noise ratios tested, our method gives the best PSNR and COR values amongst all the methods.

TABLE II
COMPARISON OF RESTORATION RESULTS IN COR FOR IMAGES CORRUPTED WITH RANDOM-VALUED IMPULSE NOISE

Method	"Baboon" image			"Bridge" image			"Lena" image			"Pentagon" image		
	20%	40%	60%	20%	40%	60%	20%	40%	60%	20%	40%	60%
Median Filter [2]	0.894	0.831	0.771	0.966	0.935	0.889	0.993	0.982	0.939	0.924	0.839	0.756
Switching Scheme I [5]	0.921	0.841	0.758	0.974	0.940	0.866	0.994	0.981	0.907	0.941	0.876	0.777
Switching Scheme II [5]	0.896	0.858	0.768	0.972	0.943	0.866	0.995	0.981	0.907	0.924	0.881	0.774
SD-ROM Filter (M=2, without training) [10]	0.921	0.857	0.770	0.978	0.948	0.868	0.996	0.983	0.917	0.952	0.887	0.737
SD-ROM Filter (M=1296, with training) [10]	0.922	0.864	0.781	0.976	0.954	0.903	0.997	0.988	0.945	0.953	0.902	0.796
PSM Filter [35]	0.918	0.852	0.777	0.975	0.941	0.879	0.996	0.984	0.927	0.939	0.876	0.780
TSM Filter [36]	0.923	0.859	0.786	0.976	0.943	0.894	0.995	0.982	0.935	0.940	0.878	0.787
MSM Filter [6]	0.927	0.864	0.782	0.979	0.951	0.893	0.997	0.986	0.933	0.951	0.897	0.796
ACWM Filter [7]	0.930	0.869	0.776	0.979	0.947	0.866	0.997	0.984	0.911	0.952	0.893	0.761
PWMAD Filter [12]	0.922	0.867	0.783	0.978	0.954	0.910	0.997	0.991	0.958	0.951	0.904	0.809
Luo, Iterative Median Filter [16]	0.931	0.860	0.745	0.979	0.954	0.881	0.997	0.986	0.938	0.954	0.899	0.769
ROAD-Trilateral Filter [28]	0.929	0.870	0.792	0.980	0.956	0.908	0.997	0.991	0.971	0.953	0.899	0.814
ACWM-EPR [26]	0.926	0.870	0.796	0.980	0.960	0.910	0.997	0.992	0.960	0.950	0.903	0.816
ROAD-EPR	0.932	0.871	0.809	0.981	0.961	0.929	0.997	0.993	0.982	0.953	0.901	0.836
ROLD-EPR	*0.935*	*0.880*	*0.821*	*0.982*	*0.963*	*0.938*	*0.998*	*0.994*	*0.985*	*0.957*	*0.913*	*0.857*

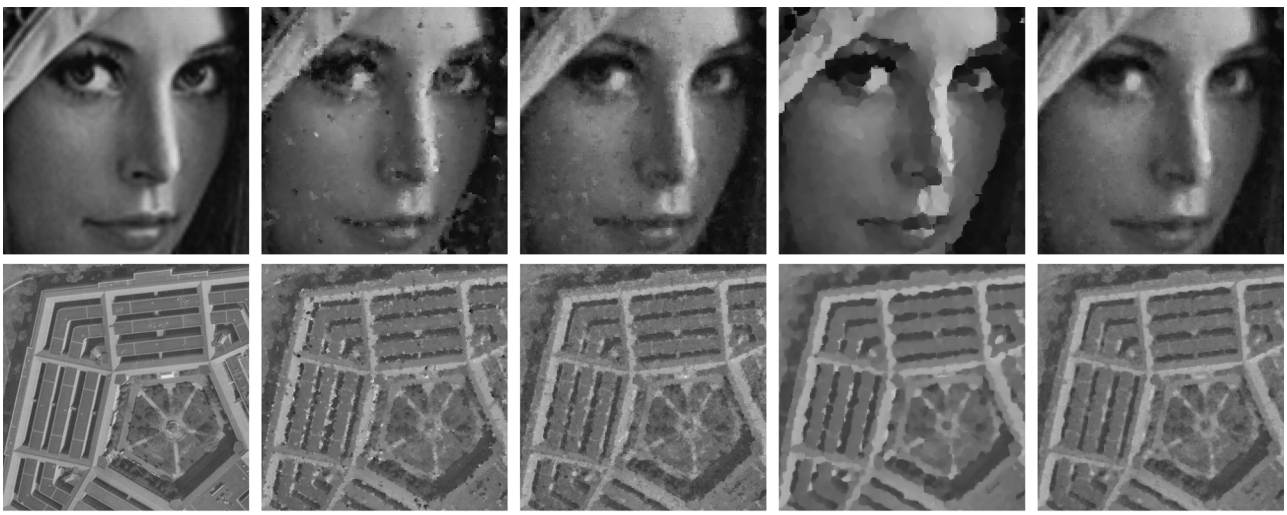
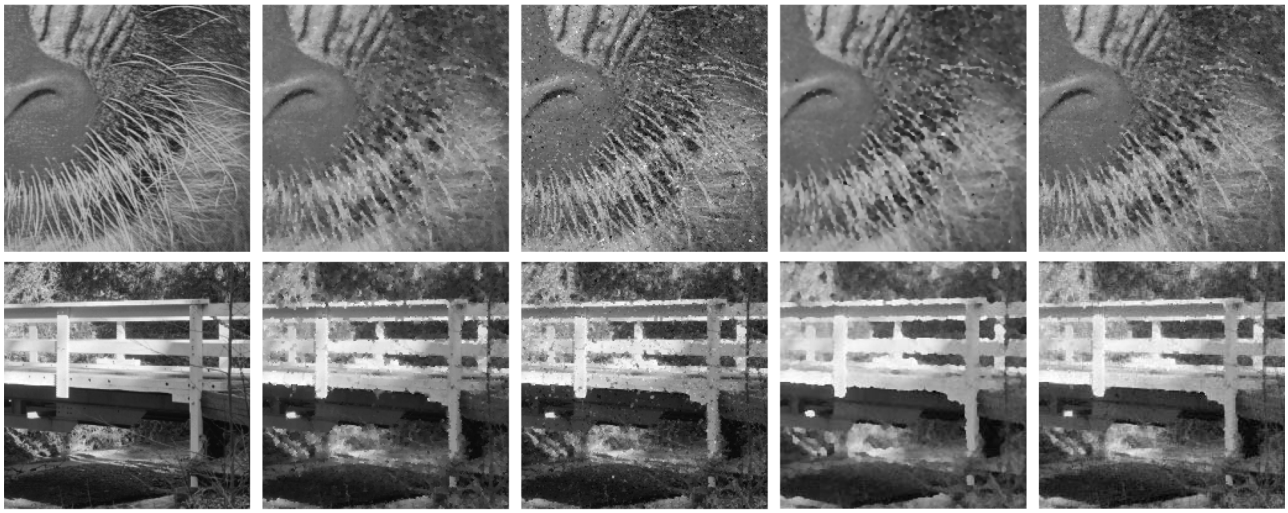


Fig. 4. Results of different methods in restoring 60% corrupted images "Lena" and "Pentagon": (a) original images; (b) results after ACWM-EPR method [26]; (c) results after ROAD-EPR method; (d) results after ROAD-trilateral filter [28]; (e) results after our ROLD-EPR method.

To compare the results subjectively, we enlarge portion of the images restored by some methods listed in Table I. Fig. 4 shows the results in restoring 60% corrupted images of "Lena" and "Pentagon." In the images restored by ACWM-EPR [26] or ROAD-EPR methods, there are still many noticeable noise patches. Although no noticeable noise is observed by the ROAD-trilateral filter, the images are blurred seriously. In contrast, our method performs better, and can suppress the noise successfully while preserving more details. To further compare the capability of preserving image details, in Fig. 5, we give the restored results for two images with rich details but corrupted by 40% impulse noise. We see that for the other methods, there are still some noticeable noise unremoved and there exist some loss and discontinuity of the details, such as the hair around the mouth of the baboon and the edges of the bridge. In contrast, the visual qualities of our restored images are quite good, even with the abundance of image details and the high noise level present in the images.

B. Comparison of Noise Detection

For good performance, the capability of noise detection is very important. Here, we compare our method with all the methods in Table I that have noise detectors. Table III lists the number of missed noisy pixels ("miss" term) and the number of "false-hit" pixels. For random-valued impulse noise, the noisy pixel values may not be so different from those of their neighbors; therefore, it is easier for a noise detector to miss a noisy pixel or wrongly detect a noise-free pixel. A good noise detector should be able to identify most of the noisy pixels, and yet its "false-hit" rate should be as small as possible. In Table III, although some methods, such as the SD-ROM and ACWM filters, produce less "false-hit" than ours, there are too many missed noisy pixels. These pixels will lead to the presence of noticeable noise patches. Comparing with other methods, our method can distinguish more noise pixels with fewer mistakes. Even when the noise level is as high as 60%,



(a) (b) (c) (d) (e)

Fig. 5. Results of different methods in restoring 40% corrupted images "Baboon" and "Bridge": (a) original images; (b) results after the ACWM-EPR method [26]; (c) results after the ROAD-EPR method; (d) results after the ROAD-trilateral filter [28]; (e) results after our ROLD-EPR method.

TABLE III
COMPARISON OF NOISE DETECTION RESULTS FOR IMAGE "LENA" CORRUPTED WITH RANDOM-VALUED IMPULSE NOISE

Method	40%		50%		60%	
	miss	false-hit	miss	false-hit	miss	false-hit
Switching Scheme I [5]	16706	9058	16011	19052	14512	32876
Switching Scheme II [5]	14342	9098	16011	19052	14122	33006
SD-ROM Filter (M=2, without training) [10]	22842	411	32566	998	45365	2651
PSM Filter [35]	23461	4773	29396	8550	35617	15152
TSM Filter [36]	18703	3974	12035	26806	15492	33029
MSM Filter [6]	16582	7258	20857	10288	26169	15778
ACWM Filter [7]	16052	1759	23683	2895	32712	7644
PWMAD Filter [12]	11817	9928	14490	15003	17760	19577
Luo, Iterative Median Filter [16]	16706	3678	23707	6410	38512	12724
ACWM-EPR [26]	13657	6192	13868	12693	23793	17573
ROAD-EPR	13476	8079	13771	10055	17212	9330
ROLD-EPR	12010	7404	13329	7761	14967	9109

our method can still identify most of the noisy pixels. In fact, if one computes the sum of the "miss" term and the "false-hit" term, our method has the lowest sum amongst all methods except for ACWM at 40% noise level.

We note that although the number of missed pixels seems to be large for our method, they are in fact pixels with corrupted values that are close to the true value and, hence, are difficult to detect correctly. To illustrate this, in Table IV, we give the mean errors of the set of all missed pixels and the set of all noisy pixels, i.e.

$$\mathrm{E}(\mathcal{A}) = \frac{1}{\sharp\mathcal{A}} \sum_{(i,j)\in\mathcal{A}} |y_{i,j} - x_{i,j}|$$

where $\mathcal{A}$ denotes either the set of all missed pixels or the set of all noisy pixels, $\sharp$ stands for cardinality, and $y_{i,j}$ and $x_{i,j}$ are the noisy pixel and true pixel values respectively. It is clear from the table that most of the missed noisy pixels in fact have intensities very close to their true pixel values.

C. Robustness With Respect to s

Here, we show that our method is robust against the choice of s, see the definition of s at the end of Section III. In the first row of Fig. 6, we plot the PSNR values of the restored images by our method for s varying from 1.7 to 2.1 and for the four test images with noise ratios 10% and 20%. In the second row of Fig. 6, we plot the PSNR values by our method for s from 4.9 to 5.9 with noise ratios 40% and 60%. From the plots, we can see that the PSNR is very stable, and the maximum difference in each case is less than 0.5 dB. Hence, we can set $s = 1.9$ for $p \leq 25\%$, and $s = 5.4$ for $p > 25\%$.

TABLE IV
COMPARISON OF MEAN ERROR VALUES FOR MISSED NOISY PIXELS AND ALL NOISY PIXELS

	40%	50%	60%
Mean error of all missed noisy pixels	11.21	10.09	10.22
Mean error of all noisy pixels	78.01	78.08	78.09

V. CONCLUSION

In this paper, we propose a new local image statistic ROLD, by which we can identify more noisy pixels with less false hits. We combine it with the edge-preserving regularization in the two-stage method [26] to get a powerful method for removing random-valued impulse noise. Simulation results show that our

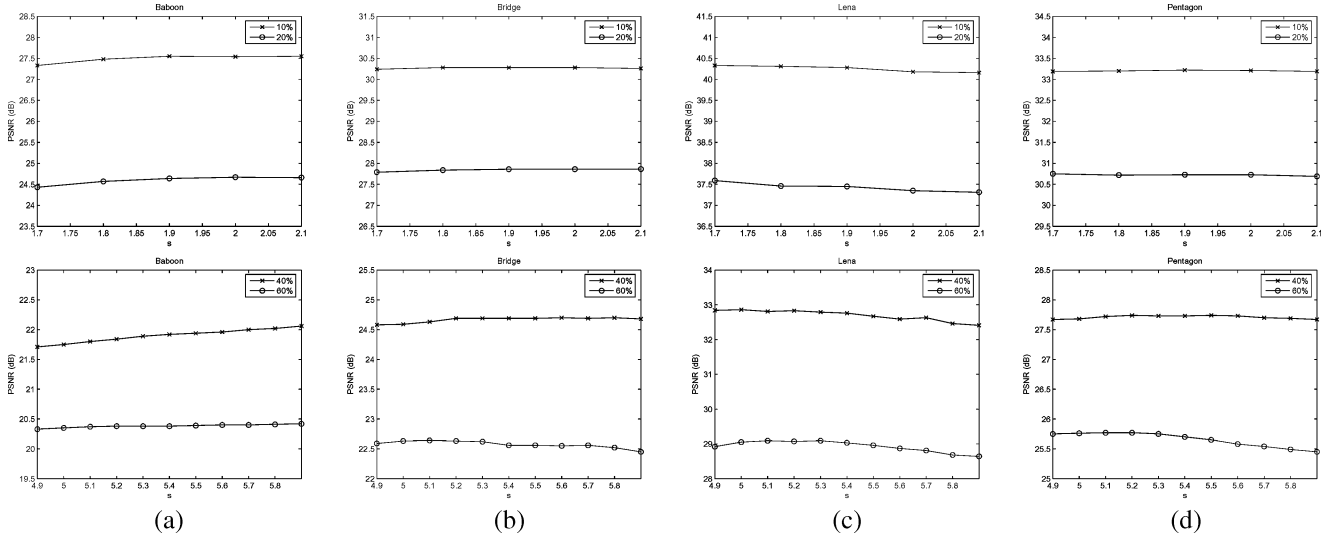


Fig. 6. PSNR of restored images by our method for different s: (a) "Baboon," (b) "Bridge," (c) "Lena," and (d) "Pentagon."

method outperforms a number of existing methods both visually and quantitatively.

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[2] W. K. Pratt, "Median filtering," Tech. Rep., Image Proc. Inst., Univ. Southern California, Los Angeles, Sep. 1975.

[3] D. Brownrigg, "The weighted median filter," *Commun. Assoc. Compu.*, pp. 807–818, Mar. 1984.

[4] S.-J. Ko and S.-J. Lee, "Center weighted median fitlers and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 15, no. 4, pp. 984–993, Sep. 1991.

[5] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognit. Lett.*, vol. 15, pp. 341–347, 1994.

[6] T. Chen and H. R. Wu, "Space variant median filters for the restoration of impulse noise corrupted images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 8, pp. 784–789, Aug. 2001.

[7] ——, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 1–3, Jan. 2001.

[8] P. S. Windyga, "Fast impulsive noise removal," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 173–179, Jan. 2001.

[9] N. Alajlan, M. Kamel, and E. Jernigan, "Detail preserving impulsive noise removal," *Signal Process.: Image Commun.*, vol. 19, pp. 993–1003, 2004.

[10] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 1012–1025, Jun. 1996.

[11] G. Pok, J.-C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 85–92, Jan. 2003.

[12] V. Crnojević, V. Šenk, and Ž Trpovski, "Advanced impulse detection based on pixel-wise MAD," *IEEE Signal Process. Lett.*, vol. 11, no. 7, pp. 589–592, Jul. 2004.

[13] I. Aizenberg, C. Butakoff, and D. Paliy, "Impulsive noise removal using threshold Boolean filtering based on the impulse detecting functions," *IEEE Signal Process. Lett.*, vol. 12, no. 1, pp. 63–66, Jan. 2005.

[14] E. Beşdok and M. E. Yüksel, "Impulsive noise suppression from images with Jarque-Bera test based median filter," *Int. J. Electron. Commun.*, vol. 59, pp. 105–110, 2005.

[15] F. Russo, "Impulse noise cancellation in image data using a two-output nonlinear filter," *Measurement*, vol. 36, pp. 205–213, 2004.

[16] W. Luo, "A new efficient impulse detection algorithm for the removal of impulse noise," *IEICE Trans. Fundam.*, vol. E88-A, no. 10, pp. 2579–2586, Oct. 2005.

[17] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. E. Kerre, W. Philips, and I. Lemahieu, "Noise reduction by fuzzy image filtering," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 429–436, Aug. 2003.

[18] H. Xu, G. Zhu, H. Peng, and D. Wang, "Adaptive fuzzy switching filter for images corrupted by impulse noise," *Pattern Recognit. Lett.*, vol. 25, pp. 1657–1663, 2004.

[19] F. Russo, "Hybrid neuro-fuzzy filter for impulse noise removal," *Pattern Recognit.*, vol. 32, pp. 1843–1855, 1999.

[20] ——, "Noise removal from image data using recursive neurofuzzy filters," *IEEE Trans. Instrum. Meas.*, vol. 49, no. 2, pp. 307–314, Apr. 2000.

[21] M. E. Yüksel and A. Baştürk, "Efficient removal of impulse noise from highly corrupted digital images by a simple neuro-fuzzy operator," *Int. J. Electron. Commun.*, vol. 57, pp. 214–219, 2003.

[22] M. E. Yüksel and E. Beşdok, "A simple neuro-fuzzy impulse detector for efficient blur reduction of imuplse noise removal operators for digital images," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 6, pp. 854–865, Dec. 2004.

[23] M. E. Yüksel, "A hybrid neuro-fuzzy filter for edge preserving restoration of images corrupted by impulse noise," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 928–936, Apr. 2006.

[24] M. Nikolova, "A variational approach to remove outliers and impulse noise," *J. Math. Imag. Vis.*, vol. 20, pp. 99–120, 2004.

[25] R. H. Chan, C.-W. Ho, and M. Nikolova, "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, Oct. 2005.

[26] ——, "An iterative procedure for removing random-valued impulse noise," *IEEE Signal Process. Lett.*, vol. 11, no. 12, pp. 921–924, Dec. 2004.

[27] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, Apr. 1995.

[28] R. Garnett, T. Huegerich, C. Chui, and W.-J. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1747–1754, Nov. 2005.

[29] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers," *SIAM J. Numer. Anal.*, vol. 40, pp. 965–994, 2002.

[30] A. Bovik, *Handbook of Image and Video Processing*. New York: Academic, 2000.

[31] R. H. Chan, C.-W. Ho, C.-Y. Leung, and M. Nikolova, "Minimization of detail-preserving regularization functional by Newton's method with continuation," in *Proc. Int. Conf. Image Processing*, 2005, pp. 125–128.

[32] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, Feb. 1997.

[33] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM J. Optim.*, vol. 7, pp. 26–33, 1997.

[34] The USC-SIPI Image Database [Online]. Available: http://sipi.usc.edu/services/database/Database.html

[35] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 78–80, Jan. 1999.

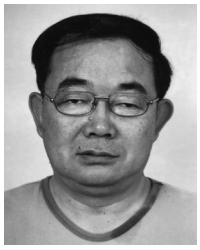
[36] T. Chen, K.-K. Ma, and L.-H. Chen, "Tri-state median filter for image denoising," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1834–1838, Dec. 1999.

Yiqiu Dong was born in 1980 in Shandong, China. She received the B.Sc. degree in mathematics from Yantai University, China, in 2002. She is currently pursuing the Ph.D. degree at the School of Mathematical Sciences, Peking University, Beijing, China.

Her research areas include image processing, matrix computation, and applications.

Raymond H. Chan was born in 1958 in Hong Kong. He received the B.Sc. degree in mathematics from the Chinese University of Hong Kong and the M.Sc. and Ph.D. degrees in applied mathematics from New York University.

He is now a Professor in the Department of Mathematics, The Chinese University of Hong Kong. His research interests include numerical linear algebra and image processing problems.



Shufang Xu was born in 1953 in Inner Mongolia, China. He received the B.Sc. and M.Sc. degrees in mathematics from Inner Mongolia University in 1982 and 1987, respectively, and the Ph.D. Degree in mathematics from the Institute of Computational Mathematics, Chinese Academy of Sciences, in 1990.

He is a Professor with the School of Mathematical Sciences, Peking University, Beijing, China. His research interests are in inverse eigenvalue problems, matrix perturbation theory, and matrix computation with applications in control, data mining, and image processing.