

# Fast Two-Phase Image Deblurring Under Impulse Noise

Jian-Feng Cai · Raymond H. Chan · Mila Nikolova

Published online: 18 August 2009  
© Springer Science+Business Media, LLC 2009

**Abstract** In this paper, we propose a two-phase approach to restore images corrupted by blur and impulse noise. In the first phase, we identify the outlier candidates—the pixels that are likely to be corrupted by impulse noise. We consider that the remaining data pixels are essentially free of outliers. Then in the second phase, the image is deblurred and denoised simultaneously by a variational method by using the essentially outlier-free data. The experiments show several dB's improvement in PSNR with respect to the typical variational methods.

**Keywords** Deblurring · Impulse noise · Two-phase methods

## 1 Background

We consider how to recover a digital image  $x \in \mathbb{R}^{m \times m}$  when the observed image  $y$  is blurred and corrupted with impulse

noise. Degradation by blur is almost unavoidable in imaging systems while corruption with impulse noise comes from bit errors in transmission, wrong pixels and faulty memory locations in hardware. A review of the importance of this kind of degradations can be found e.g. in [6, 12]. Under this degradation model, our observation  $y$  is of the form

$$y = N_p(\tilde{y}) \quad \text{where } \tilde{y} = Hx. \quad (1)$$

Here,  $N_p$  represents an impulse noise while  $H$  models the blurring effect. We assume that the blurring kernel of  $H$  is known. Two main models for the impulse noise are used in a wide variety of applications: salt-and-pepper and random-valued impulse noise. Denote the dynamic range of  $\tilde{y}$  to be  $[d_{\min}, d_{\max}]$ , i.e.  $d_{\min} \leq \tilde{y}_{ij} \leq d_{\max}$  for all  $(i, j)$ . Then the models are defined by

- *Salt-and-pepper noise*: the gray level of  $y$  at pixel location  $(i, j)$  is

$$y_{ij} = \begin{cases} d_{\min}, & \text{with probability } s/2, \\ d_{\max}, & \text{with probability } s/2, \\ \tilde{y}_{ij}, & \text{with probability } 1 - s, \end{cases} \quad (2)$$

where  $s$  determines the level of the salt-and-pepper noise.

- *Random-valued impulse noise*: the gray level of  $y$  at pixel location  $(i, j)$  is

$$y_{ij} = \begin{cases} d_{ij}, & \text{with probability } r, \\ \tilde{y}_{ij}, & \text{with probability } 1 - r, \end{cases}$$

where  $d_{ij}$  are identically and uniformly distributed random numbers in  $[d_{\min}, d_{\max}]$  and  $r$  defines the level of the random-valued noise.

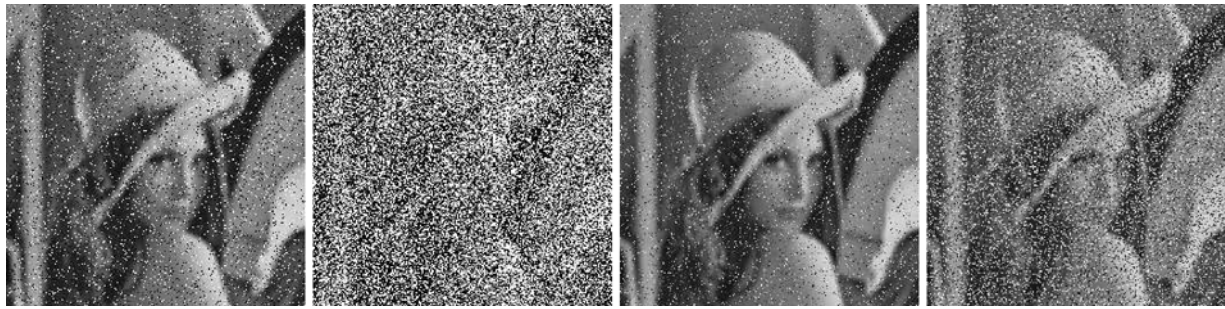
---

R.H. Chan research was supported by HKRGC Grant CUHK 400505 and CUHK DAG 2060257.

J.-F. Cai  
Department of Mathematics, UCLA, 520 Portola Plaza, Los Angeles, CA 90095, USA  
e-mail: [cai@math.ucla.edu](mailto:cai@math.ucla.edu)

R.H. Chan (✉)  
Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong  
e-mail: [rchan@math.cuhk.edu.hk](mailto:rchan@math.cuhk.edu.hk)

M. Nikolova  
Centre de Mathématiques et de Leurs Applications, ENS de Cachan, CNRS, PRES UniverSud, 61 av. du Président Wilson, 94235 Cachan Cedex, France  
e-mail: [nikolova@cmla.ens-cachan.fr](mailto:nikolova@cmla.ens-cachan.fr)



**Fig. 1** Lena image blurred by the out-of-focus kernel of radius 3 and contaminated by different noise patterns, where  $s$  and  $r$  are the levels of the salt-and-pepper (SP) and the random-valued noise, respectively.

The figures correspond to (from left to right):  $s = 10\%$ ;  $s = 70\%$ ;  $r = 10\%$ ;  $r = 40\%$



**Fig. 2** Lena image blurred with out-of-focus kernel of radius 3, and then corrupted by impulse noise. Both images are restored by minimizing (3) with  $\beta = 0.01$ . From left to right: The blurred image corrupted

by  $s = 1\%$  salt-and-pepper noise and its restoration; the blurred image corrupted with  $r = 1\%$  random-value impulse noise and its restoration

Examples of images blurred with an out-of-focus kernel of radius 3 and corrupted with different noise patterns are shown in Fig. 1. Blurring oversmooths images and thus it entails a loss of high-frequency information. It is well-known that the inverse problem—the inversion of the blurring operator  $H$ —is ill-posed [10, 17, 19]. Since [18], a large variety of regularization methods have been conceived in order to cope with perturbations due to numerical errors and noise. Usually they are based on an  $\ell_2$  data-fitting term which from a statistical point of view means that they are adapted to deal with Gaussian noise.

One can try to deblur images corrupted by impulse noise by applying the classical methods developed for Gaussian noise. These usually amount to defining the restored image as a minimizer of a functional of the form

$$F_y(x) = \|Hx - y\|_2^2 + \beta \sum_{(i,j) \in \mathcal{A}} \sum_{(k,l) \in \mathcal{V}_{ij}} \varphi(|x_{ij} - x_{kl}|), \quad (3)$$

where  $\beta$  is the regularization parameter,  $\mathcal{V}_{ij}$  is the set of the four closet neighbors of pixel location  $(i, j)$  and  $\varphi$  is a function that models the priors on the sought-after image. As seen in Fig. 2, the result is hopeless: even for very small noise ratio say 1% of impulse noise, the method gives very poor results that contain numerous spurious concentric rings.

A natural alternative is to preprocess the data using some classical de-spiking tools such as rank-order (i.e. median-based) filters [6] and then to restore the image using a variational method of the form (3). This approach is illustrated in Fig. 3. The salt-and-pepper noise and the random impulse noise were smoothed by the adaptive median filter (AMF) [13] and the adaptive center-weighted median filter (ACWMF) respectively [14]. They are chosen according to some previous experiments in [8, 9] and provide a good compromise between simplicity and robustness. As in Fig. 2, spurious circles occur in Fig. 3, especially near the edges.

Since the  $\ell_1$ -data fitting in regularized energies is known for its robustness to removing outliers [15, 16], another possible method for deblurring under impulse noise is to solve

$$F_y(x) = \|Hx - y\|_1 + \beta \sum_{(i,j) \in \mathcal{A}} \sum_{(k,l) \in \mathcal{V}_{ij}} \varphi(|x_{ij} - x_{kl}|), \quad (4)$$

where  $\varphi$  is chosen to be edge-preserving. This approach provides meaningful results and its various aspects were explored in [2–5], where usually  $\varphi$  corresponds to the Mumford-Shah functional. Let us notice that the resultant energy is nonconvex and may have numerous local minima. In our previous paper [7], a two-phase approach was used for impulsive noise deblurring. The second phase of that



**Fig. 3** Lena image blurred with out-of-focus kernel of radius 3, and then corrupted by impulse noise. The image is restored by first applying a median-type filter and then minimizing (3) with  $\varphi(t) = \sqrt{t^2 + 10^{-4}}$  and  $\beta = 0.01$ . From left to right: The output of AMF when  $s = 30\%$ ,

and the deblurred image based on the output of AMF; the output of ACWMF when  $r = 25\%$ , and the deblurred image based on the output of ACWMF

two-phase approach can be seen as a special variant of the method in [2–5], and uses nonconvex energies which may leads to numerous local minima. In this paper, we propose a new two-phase approach—described in the section below—which is much simpler. The new two-phase approach uses convex energies. Therefore, it is relatively easier to find a global minimum than the method in [7]. The experiments show that the new two-phase approach considerably improves the results.

## 2 Two-Phase Deblurring Approach

The essence of our approach is to first detect those possible corrupted pixels and then, at a second phase, to restore the image by using *only* those pixels that are surely not corrupted. The 2-phase idea comes from our two-phase *denoising* methods in [8, 9]. However, in [8, 9], the restoration is done only for corrupted pixels using the restored values of the outliers. Here the inherent ill-posedness of *deblurring* makes it impossible to use any restored values for the outliers since they are likely to be fake. A specialized minimization functional is hence necessary for deblurring. Therefore, in [7], a two-phase *deblurring* approach was proposed. It consists of the following two phases:

1. Accurate detection of the location of outliers (the noise candidates) using a median-type filter.
2. Edge-preserving restoration that deblur using only those data samples that are *not* noise candidates.

These phases are explained in details below.

### 2.1 Noise Detection

Since  $H$  is a smoothing operator, edges and other high frequency features are not that prominent in the blurred image  $Hx$ , as given in (1). This suggests that median-type filtering can efficiently detect the locations of the data pixels corrupted by impulse noise. Which median-type filter is chosen

as detector depends on the kind of the impulse noise. An overview of median-type filters can be found e.g. in [1, 6, 12]. Based on the experiments in [7–9], we use the adaptive median filter (AMF) [13] to detect salt-and-pepper noise and the adaptive center-weighted median filter (ACWMF) [14] for random-valued impulse noise. Even though other impulse noise filters—e.g., ROAD statistic [11]—can be used, our choices provide a good compromise between simplicity and robustness. Let us emphasize that any other filter (usually a median-type filter) that provides a good detection of outliers can also be employed in this phase.

Denote by  $z \in \mathbb{R}^{m \times m}$  the result obtained by applying the median-type filter to the blurred and noisy image  $y$ . The filtered data  $z$  will only be used to determine the *noise candidate set*  $\mathcal{N}$ —the data samples that are likely to be contaminated with impulse noise.

- For salt-and-pepper noise:

$$\mathcal{N} = \{(i, j) \in \mathcal{A} : z_{ij} \neq y_{ij} \text{ and } y_{ij} \in \{d_{\min}, d_{\max}\}\}, \quad (5)$$

- For random-valued impulse noise:

$$\mathcal{N} = \{(i, j) \in \mathcal{A} : z_{ij} \neq y_{ij}\}. \quad (6)$$

Accordingly, the set of data samples that are likely to be uncorrupted is defined as

$$\mathcal{U} = \mathcal{A} \setminus \mathcal{N}.$$

Clearly random-valued impulse noise are more difficult to detect than salt-and-pepper noise. We can hence expect more difficulties with random-valued noise.

### 2.2 Restoration Using a Variational Method

The data samples  $y_{ij}$  with  $(i, j) \in \mathcal{N}$  do not carry proper information of the true image. Their estimates  $z_{ij}$  provided by any median-type filter combine in some way the values of the neighboring pixels, so they inevitably contain errors

that do not fit the model for Gaussian noise in  $\tilde{y}$ , assumed in (1). Their use in the deblurring stage can only be harmful. Indeed, the harmful effect they produce on the solution can be observed in Figs. 3(b) and (d). The best we can do is to ignore all  $y_{i,j}$  with  $(i, j) \in \mathcal{N}$ . The restoration is then done using only the incomplete data set  $y_{ij}$  with  $(i, j) \in \mathcal{U}$ .

These data samples may still contain a few outliers of small amplitude as no median-type filter is a perfect noise detector. The resultant inverse problem is heavily ill-posed, but it is based on the most reliable data samples that we could find. We solve it by a variational method inspired by (4). The functional we minimize is convex and reads

$$\sum_{(i,j) \in \mathcal{U}} |[Hx - y]_{ij}| + \beta \sum_{(i,j) \in \mathcal{A}} \sum_{(k,l) \in \mathcal{V}_{ij}} |x_{ij} - x_{kl}|. \tag{7}$$

It corresponds to choosing  $\varphi(t) = t$  in (4) with the data-fitting term being restricted only to the samples belonging to  $\mathcal{U}$ —which is the crucial difference with (4) and also the functionals in [8, 9]. A more convenient and equivalent expression for our functional is

$$\sum_{(i,j) \in \mathcal{A}} |\chi_{ij}[Hx - y]_{ij}| + \beta \sum_{(i,j) \in \mathcal{A}} \sum_{(k,l) \in \mathcal{V}_{ij}} |x_{ij} - x_{kl}|, \tag{8}$$

where  $\chi$  is the characteristic function of the set  $\mathcal{U}$ , namely

$$\chi_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{U}, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that in [7], in the second phase, we minimize a functional consisting of an  $\ell_1$  fidelity and a Mumford-Shah regularization term as follows

$$\sum_{(i,j) \in \mathcal{U}} |[Hx - y]_{ij}| + \beta \int_{\Omega \setminus \Gamma} |\nabla x|^2 + \alpha \int_{\Gamma} d\sigma. \tag{9}$$

By comparing (7) and (9), we see the advantages of the former over the latter. First, (7) is convex while (9) is non-convex. Thus, there exist numerous local minimums of (9), while any local minimum of (7) is its global minimum. Secondly, it is well known that the Mumford-Shah functional in (9) is difficult to handle. One may use Gamma-convergence as we have done in [7]. Therefore, it is difficult to implement the algorithm for (9). However, for (7), as we will see in the next section, we can use a fixed point iteration that is very easy to program. Finally, as illustrated in Sect. 4, the computational time for minimizing (7) is much smaller than that for (9).

### 3 Numerical Implementation

In order to approximate the nonsmooth optimization problem in (8), we introduce a weak smooth regularization, as it

is customarily done in the literature:

$$\mathcal{F}(x) = \sum_{(i,j) \in \mathcal{A}} \sqrt{\chi_{ij}[Hx - y]_{ij}^2 + \eta} + \beta \sum_{(i,j) \in \mathcal{A}} \sum_{(k,l) \in \mathcal{V}_{ij}} \sqrt{|x_{ij} - x_{kl}|^2 + \eta}, \tag{10}$$

where  $\eta \gtrsim 0$ . Let  $G$  be the difference matrix such that  $(Gx)_{ij,kl} = x_{ij} - x_{kl}$  for  $(i, j) \in \mathcal{A}$  and  $(k, l) \in \mathcal{V}_{ij}$ . Then the gradient of  $\mathcal{F}$  is given by

$$\nabla \mathcal{F}(x) = H^* \frac{\chi \circ (Hx - y)}{\sqrt{[Hx - y]^2 + \eta}} + \beta G^* \frac{Gx}{\sqrt{[Gx]^2 + \eta}}, \tag{11}$$

where  $\circ, [\cdot]^2$  and  $\div$  are entrywise multiplication, square, and division respectively, and  $H^*$  and  $G^*$  are the adjoint of  $H$  and  $G$  respectively. Since  $\mathcal{F}$  is convex, minimizing  $\mathcal{F}(x)$  is equivalent to solving  $\nabla \mathcal{F}(x) = 0$ . Following [20] and many other authors, we minimize (10) by a fixed-point iteration method. The basic idea is to linearize the gradient of  $\mathcal{F}$  at each iteration. Given  $x^p$ , we get  $x^{p+1}$  by solving  $x$  in the equation:

$$H^* \frac{\chi \circ (Hx - y)}{\sqrt{[Hx^p - y]^2 + \eta}} + \beta G^* \frac{Gx}{\sqrt{[Gx^p]^2 + \eta}} = 0. \tag{12}$$

Since (12) is a linear equation, it can be solved efficiently by linear solvers.

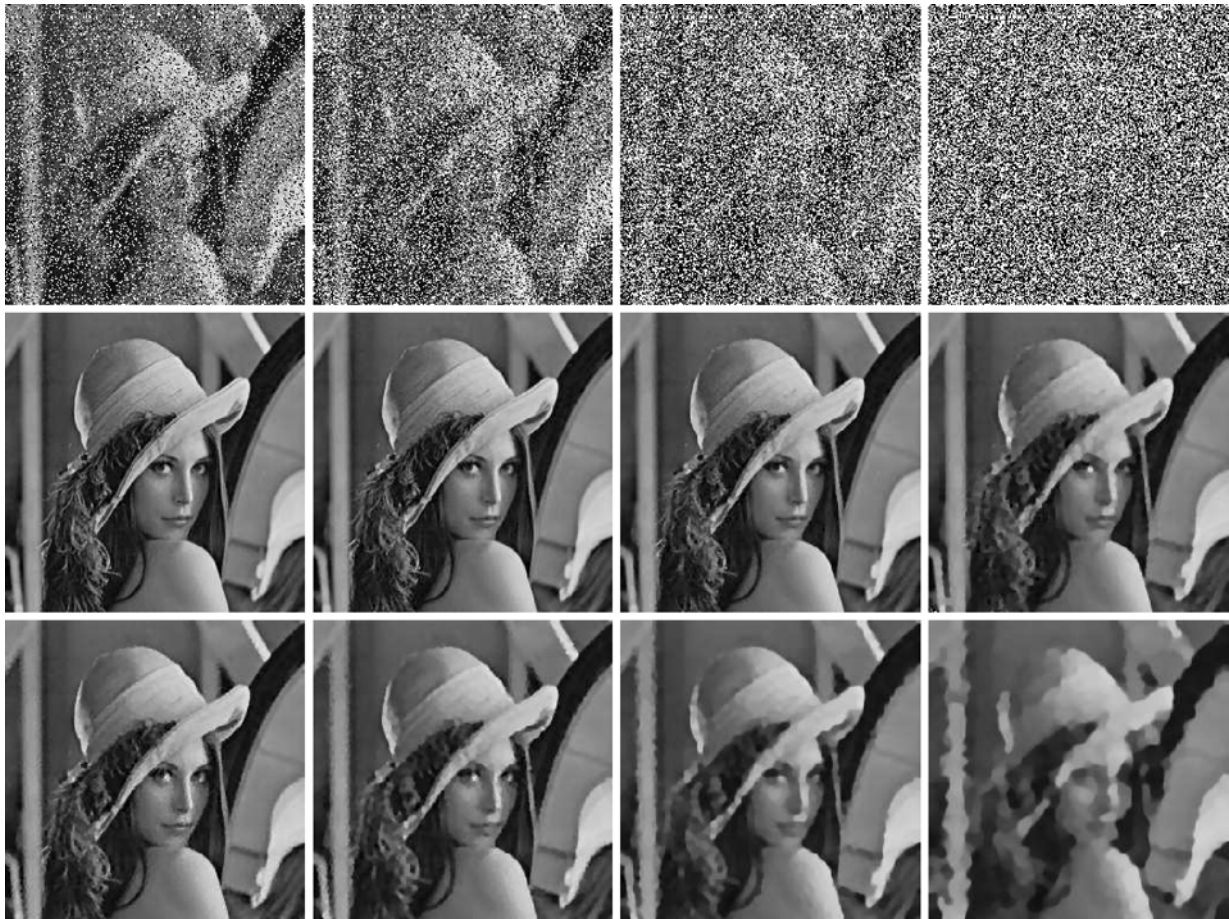
### 4 Experiments and Comparisons

In this section, numerical examples are presented to illustrate the effectiveness of our two-phase deblurring method by comparing it with the full variational method (4) with  $\varphi(t) = t$ , and with the two-phase deblurring method in [7], i.e., minimizing (9) in the second phase.

The simulations are performed in Matlab 7.01 (R14) on a PC. To assess the restoration performance quantitatively, we evaluate the peak signal to noise ratio (PSNR, see 6) defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{n^2} \sum_{(i,j) \in \mathcal{A}} (\hat{x}_{ij} - x_{ij})^2}, \tag{13}$$

where  $\hat{x}_{ij}$  and  $x_{ij}$  are the pixel values of the restored image and of the original image, respectively. The test images are all 256-by-256 gray level images. We fix  $\eta = 1$ . The remaining parameter  $\beta$  is chosen empirically such that it gives the best restoration measured in PSNR.



**Fig. 4** Lena image blurred with out-of-focus kernel of radius 3, and then corrupted by salt-and-pepper noise with noise levels 30%, 50%, 70%, and 90% respectively. *Top*: The noisy blurred image. *Middle*: The restored image by the two-phase approach, and the parameters we used

are all  $\beta = 2 \times 10^{-5}$ . *Bottom*: The restored image by the full variational method, and the parameters we used are  $\beta = 0.02, 0.05, 0.2, 0.5$  respectively

**Table 1** The PSNR(dB) and computing time (in seconds) comparisons of the two-phase method and the full variational method. The blurring kernel is the out-of-focus of radius 3

| Image           | $s$ | Two-phase method |         |         |       | Two-phase method in [7] |         |         | Full variational method |      |       |
|-----------------|-----|------------------|---------|---------|-------|-------------------------|---------|---------|-------------------------|------|-------|
|                 |     | PSNR             | Time    |         | #Iter | PSNR                    | Time    |         | PSNR                    | Time | #Iter |
|                 |     |                  | Phase 1 | Phase 2 |       |                         | Phase 1 | Phase 2 |                         |      |       |
| <i>Lena</i>     | 30% | 37.5             | 0.2     | 187     | 6     | 35.9                    | 0.2     | 504     | 30.0                    | 71.6 | 18    |
|                 | 50% | 33.7             | 0.3     | 212     | 7     | 32.7                    | 0.3     | 496     | 27.8                    | 81.0 | 25    |
|                 | 70% | 30.7             | 0.5     | 239     | 9     | 30.1                    | 0.5     | 488     | 25.6                    | 124  | 33    |
|                 | 90% | 27.1             | 10.5    | 335     | 14    | 26.7                    | 10.5    | 623     | 21.6                    | 192  | 40    |
| <i>Bridge</i>   |     | 26.4             | 0.6     | 241     | 10    | 26.2                    | 0.6     | 514     | 22.6                    | 129  | 34    |
| <i>Baboon</i>   |     | 24.7             | 0.7     | 223     | 10    | 24.7                    | 0.7     | 452     | 22.5                    | 85.0 | 26    |
| <i>Boat</i>     | 70% | 27.7             | 0.6     | 231     | 9     | 26.7                    | 0.6     | 488     | 23.4                    | 103  | 30    |
| <i>Goldhill</i> |     | 28.8             | 0.5     | 208     | 9     | 28.4                    | 0.5     | 402     | 25.3                    | 99.6 | 29    |

First we discuss the case with salt-and-pepper noise. The comparisons of our method and the full variational deblurring method (4) are shown in Fig. 4 and Table 1. In the first phase of our method, the noise candidate set  $\mathcal{N}$ , defined in (5), is detected by the AMF algorithm [13]. The

maximum window size we used in AMF is 19 throughout the test. Obviously, our two-phase deblurring method is better than the variational method. In general, the PSNR of the restoration by our method is about 2 to 7 dB higher than that by the variational method, and our two-phase method



**Fig. 5** Restoration of our method for images blurred with out-of-focus kernel of radius 3, and then corrupted by salt-and-pepper noise  $s = 70\%$ . The parameters used are the same as in Fig. 4 for  $s = 70\%$



**Fig. 6** Lena image blurred with out-of-focus kernel of radius 3, and then corrupted by random-valued noise with noise levels are 10%, 25%, 40%, and 55% respectively. *Top*: The noisy blurred image. *Middle*: The restored image by the two-phase approach, and the parameters

we used are  $\beta = 0.005, 0.01, 0.02, 0.1$  respectively. *Bottom*: The restored image by the full variational method, and the parameters we used are  $\beta = 0.01, 0.02, 0.02, 0.1$  respectively

can handle noise level as high as 90%, while the variational method fails. Comparing the two-phase methods in this paper with in [7], we see that: the computational time for the method in this paper is less than *half* of that for the method in [7], while the PSNRs of the restored images for the method in this paper is greater than that for the method in [7]; see Table 1.

Next we discuss the case of random-valued impulse noise. The noise is detected by ACWMF [14], which is successively performed 4 times with different parameters for one image. The parameters are chosen to be the same as those in [8]. Once the 4 steps of ACWMF are performed, we define the noise set  $\mathcal{N}$  by (6), and then perform the second phase. Again we compare our two-phase method with



**Fig. 7** The restorations of our method for images blurred with out-of-focus kernel of radius 3, and then corrupted by random-valued impulse noise with  $r = 40\%$ . The parameters used are the same as in Fig. 6 when  $s = 40\%$

**Table 2** The PSNR(dB) and computing time (in seconds) comparisons of the two-phase method and the full variational method. The blurring kernel is the out-of-focus of radius 3

| Image           | $s$ | Two-phase method |         |         | Two-phase method in [7] |      |         | Full variational method |      |      |       |
|-----------------|-----|------------------|---------|---------|-------------------------|------|---------|-------------------------|------|------|-------|
|                 |     | PSNR             | Time    |         | #Iter                   | PSNR | Time    |                         | PSNR | Time | #Iter |
|                 |     |                  | Phase 1 | Phase 2 |                         |      | Phase 1 | Phase 2                 |      |      |       |
| <i>Lena</i>     | 10% | 35.6             | 7.1     | 65.3    | 12                      | 38.7 | 7.1     | 584                     | 33.5 | 67.1 | 14    |
|                 | 25% | 32.8             | 7.1     | 65.6    | 15                      | 34.4 | 7.1     | 606                     | 30.6 | 72.1 | 18    |
|                 | 40% | 30.5             | 7.1     | 68.3    | 19                      | 31.2 | 7.1     | 739                     | 27.3 | 107  | 24    |
|                 | 55% | 27.2             | 7.1     | 104     | 30                      | 27.8 | 7.1     | 784                     | 24.7 | 127  | 35    |
| <i>Bridge</i>   |     | 26.4             | 7.0     | 80.5    | 23                      | 27.3 | 7.0     | 726                     | 24.4 | 101  | 25    |
| <i>Baboon</i>   |     | 24.7             | 7.1     | 63.6    | 21                      | 25.3 | 7.1     | 635                     | 23.9 | 80.7 | 22    |
| <i>Boat</i>     | 40% | 27.7             | 7.1     | 90.6    | 22                      | 28.2 | 7.1     | 709                     | 25.6 | 92.6 | 22    |
| <i>Goldhill</i> |     | 28.8             | 7.0     | 63.5    | 19                      | 29.5 | 7.0     | 705                     | 26.8 | 108  | 26    |

the variational method. The results are shown in Fig. 6 and Table 2. We can see from the figures that our method is again much better than the variational method. The PSNR of the restoration by our method is about 1 to 3 dB higher than that by the variational method. Even for blurred images corrupted by 55% random-valued noise, our method can give a very good restoration, while the variational method fails. Comparing the two-phase methods in this paper with in [7], we see that our method is more computationally efficient. Our method takes only about 1/8 CPU time of that by the method in [7]. However, the PSNRs of the restored image by our method are not as good as those given by [7].

We note that in all the cases tested, there are no circles appearing in our restored images which are common in other approaches (see Figs. 2 and 3). We can also see that in general the two-phase method for salt-and-pepper noise performs better than for random-valued noise: it can handle salt-and-pepper noise as high as 90% but random-valued noise for about 55%. The main reason is that the former is more easy to detect than the latter in the first phase. In fact, AMF is a very good detector for salt-and-pepper noise, and almost all the noise positions can be detected even when the noise ratio is very high. In addition, with salt-and-pepper noise, most of the noisy pixels are much more dissimilar to regular pixels, hence are easier to detect. However, there is

no good detector for random-valued noise when the noise ratio is high. The performance for random-valued noise can be improved if a better noise detector can be found in the first phase.

## References

1. Astola, J., Kuosmanen, P.: Fundamentals of Nonlinear Digital Filtering. CRC, Boca Raton (1997)
2. Bar, L., Sochen, N., Kiryati, N.: Image deblurring in the presence of salt-and-pepper noise. In: Proceeding of 5th International Conference on Scale Space and PDE methods in Computer Vision. LNCS, vol. 3439, pp. 107–118. Springer, Berlin (2005)
3. Bar, L., Sochen, N., Kiryati, N.: Image deblurring in the presence of impulsive noise. *Int. J. Comput. Vis.* **70**, 279–298 (2006)
4. Bar, L., Brook, A., Sochen, N., Kiryati, N.: Deblurring of color images corrupted by salt-and-pepper noise. *IEEE Trans. Image Process.* **16**, 1101–1111 (2007)
5. Bar, L., Sochen, N., Kiryati, N.: Convergence of an iterative method for variational deconvolution and impulsive noise removal. *SIAM J. Multiscale Model. Simul.* **6**, 983–994 (2007)
6. Bovik, A.: Handbook of Image and Video Processing. Academic Press, New York (2000)
7. Cai, J.-F., Chan, R.H., Nikolova, M.: Two-phase methods for deblurring images corrupted by impulse plus Gaussian noise. *Inverse Probl. Imaging* **2**, 187–204 (2008)
8. Chan, R.H., Hu, C., Nikolova, M.: An iterative procedure for removing random-valued impulse noise. *IEEE Signal Process. Lett.* **11**, 921–924 (2004)

9. Chan, R.H., Ho, C.W., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detector and edge-preserving regularization. *IEEE Trans. Image Process.* **14**, 1479–1485 (2005)
10. Demoment, G.: Image reconstruction and restoration: overview of common estimation structure and problems. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP **37**(12), 2024–2036 (1989)
11. Garnett, R., Huegerich, T., Chui, C., He, W.: A universal noise removal algorithm with an impulse detector. *IEEE Trans. Image Process.* **14**, 1747–1754 (2005)
12. Gonzalez, R., Woods, R.: *Digital Image Processing*. Addison-Wesley, Reading (1993)
13. Hwang, H., Haddad, R.A.: Adaptive median filters: new algorithms and results. *IEEE Trans. Image Process.* **4**, 499–502 (1995)
14. Ko, S.-J., Lee, Y.H.: Center weighted median filters and their applications to image enhancement. *IEEE Trans. Circuits Syst.* **38**, 984–993 (1991)
15. Nikolova, M.: Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM J. Numer. Anal.* **40**, 965–994 (2002)
16. Nikolova, M.: A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.* **20**, 99–120 (2004)
17. Tarantola, A.: *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier, Amsterdam (1987)
18. Tikhonov, A., Arsenin, V.: *Solutions of Ill-Posed Problems*. Winston (1977)
19. Vogel, C.: *Computational Methods for Inverse Problems*. *Frontiers in Applied Mathematics Series*, vol. 23. SIAM, New York (2002)
20. Vogel, C.R., Oman, M.E.: Iterative method for total variation denoising. *SIAM J. Sci. Comput.* **17**(1), 227–238 (1996)



**Jian-Feng Cai** was born in 1978 in China. He received his B.Sc. and M.Sc. degrees in computational mathematics from Fudan University in 2000 and 2004 respectively, and his Ph.D. degree in mathematics from the Chinese University of Hong Kong in 2007. He is currently an Assistant Adjunct Professor in Department of Mathematics, University of California, Los Angeles. His research interests are image processing, computational harmonic analysis, compressed sensing and matrix completion.



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



**Mila Nikolova** received the Ph.D. degree from the Universit de Paris-Sud, France, in 1995. She got a Habilitation to direct research in 2006. Currently, she is Senior Research Fellow, 1st class, with the National Center for Scientific Research (CNRS), France, and performs her research at the Centre de Mathematiques et de Leurs Applications (CMLA), ENS de Cachan, France. Her research interests are in mathematical Image and signal reconstruction, Inverse problems, Regularization and variational

methods and the properties of their solutions, Scientific computing.