Binary Matching Pursuit for Impulse Noise Removal

Ming Yan
Department of Mathematics, University of California, Los Angeles, CA, USA
yanm@math.ucla.edu

Abstract

This article studies the problem of image restoration of observed images corrupted by impulse noise and other types of noise (e.g. zero-mean Gaussian white noise). Since the pixels damaged by impulse noise contain no information about the true image, the damaged pixels can also be considered as missing information. If the pixels corrupted by impulse noise are known, then the image restoration becomes a standard image inpainting problem. However, the set of damaged pixels is usually unknown, thus how to find this set correctly is a very important problem. We proposed a method using binary matching pursuit that can simultaneously find the damaged pixels and restore the image. This method can also be applied to situations where the damaged pixels are not randomly chosen, but follow some unknown procedure. By iteratively restoring the image and updating the set of damaged pixels, this method has better performance than other methods, as shown in the experiments.

1. Introduction

Observed images are often corrupted by impulse noise during image acquisition and transmission, caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or bit errors in transmission [5]. There are two common types of impulse noise: salt-and-pepper impulse noise and random-valued impulse noise. Assume that the dynamic range of an image is $[d_{\min}, d_{\max}]$. For images corrupted by salt-and-pepper impulse noise, the noisy pixels can take only two values $d_{\min}$ and $d_{\max}$, while for images corrupted by random-valued impulse noise, the noisy pixels can take any random value between $d_{\min}$ and $d_{\max}$. If the original unknown image u is defined on a domain Ω, then the observed image f is modeled as

$$f = \begin{cases} Hu + e_1 & x \in \Lambda \\ e_2 & x \in \Lambda^c := \Omega \backslash \Lambda, \end{cases} \tag{1}$$

where Λ^c is the set of pixels corrupted by impulse noise. Since the damaged pixels contain no information about the true image, the damaged pixels can also be considered as missing information. If Λ^c is known, then the image restoration problem becomes a standard image inpainting (and deblurring) problem [3], and there are many methods for solving this problem [4, 13]. When the set Λ^c is not empty and unknown, the challenge of the problem is to detect the damaged pixels (remove the impulse noise) and restore the lost details simultaneously. There are already several types of approaches for solving this problem.

One group of approaches [20, 2, 1, 15] treat e_2 as outliers and use the L_1 norm in the fidelity term to increase the robustness of inpainting to outliers, and the problem is to solve

$$\min_u \int_\Omega |Hu - f| + J(u), \tag{2}$$

where $J(u)$ is a regularization on the true image u. There are many candidates for the regularization $J(u)$ such as Rudin, Osher and Fatemi's total variation models [21, 22] and framelet based models [8, 17]. This approach does not need to find the damaged pixels and performs well when there is only impulse noise. For the case of images corrupted by multiple types of noise (e.g. Gaussian noise plus impulse noise), other types of noise are not treated properly.

Second group of approaches are the two-stage approaches [9, 10, 6, 7, 17, 23], which estimate the inpainting region Λ^c before restoring the true image u. In these approaches, the second stage after estimating the set Λ^c becomes a regular image inpainting (and deblurring) problem:

$$\min_u S_\Lambda(Hu, f) + J(u), \tag{3}$$

where S_Λ is a fidelity term performing only on Λ and depends on the type of noise e_1. If the noise e_1 is zero-mean additive Gaussian white noise, we can choose $S_\Lambda(Hu, f) = \int_\Lambda \frac{1}{2}(Hu - f)^2$. The success of these two-stage approaches relies on the accurate detection of Λ^c, e.g. adaptive median filter (AMF) [16] is used to detect salt-and-pepper impulse noise, while adaptive center-weighted median filter (ACWMF) [14] is utilized to detect random-valued impulse noise.

Though adaptive median filter can detect most pixels damaged by salt-and-pepper impulse noise, it is more difficult to detect pixels corrupted by random-valued impulse noise than salt-and-pepper impulse noise. We proposed a method which can simultaneously detect the damaged pixels and restore the image. Instead of keeping the estimation of the set Λ^c fixed, the estimation of the set is updated together with restoring the image.

The work is organized as follows. The review of adaptive center-weighted median filter is given in section 2. In section 3, we introduce our general method for removing impulse noise, and one example based on framelet is provided. Some experiments are given in section 4 to show the efficiency of proposed method. We will end this work by a short conclusion section.

2. The Adaptive Center-Weighted Median Filter

In order to remove random-valued impulse noise, adaptive center-weighted median filter (ACWMF) [14] is a good method when the noise level is not high. So the result of ACWMF is often utilized in two-stage methods [6, 7, 17] to estimate the set Λ^c.

If u is a noisy M-by-N grayscale image, and $u_{i,j}$ is the gray level at pixel $(i,j) \in \{1,\cdots,M\} \times \{1,\cdots,N\}$, the expression of the ACWMF filter is as follows:

$$y_{i,j}^{2k} = \text{median}\{u_{i-s,j-t}, (2k) \diamond u_{i,j} | -h \le s,t \le h\},$$

where $(2h+1) \times (2h+1)$ is the window size, and $\diamond$ represents the repetition operation. For $k = 0,1,\cdots,J-1$, where $J = 2h(h+1)$, we can determine the differences $d_k = |y_{i,j}^{2k} - u_{i,j}|$. They satisfy the condition $d_k \le d_{k-1}$ for $k \ge 1$. To determine if the considered pixel (i,j) is noisy, a set of thresholds T_k is utilized, where $T_{k-1} > T_k$ for $k = 1,\cdots,J-1$. The output of the filter is defined in the following manner:

$$u_{\text{ACWMF}} = \begin{cases} y_{i,j}^0, & \text{if } d_k > T_k \text{ for some } k, \\ u_{i,j}, & \text{otherwise.} \end{cases} \tag{4}$$

Usually, if the window size is chosen as 3×3 (i.e., $h = 1$ and $J = 4$), four thresholds $T_k (k = 0,\cdots,3)$ are needed, and they are calculated as follows:

$$T_k = s \cdot \text{MAD} + \delta_k, \tag{5}$$

$$\text{MAD} = \text{median}\{|u_{i-s,j-t} - y_{i,j}^1| - h \le s,t \le h\}, \tag{6}$$

where $[\delta_0, \delta_1, \delta_2, \delta_3] = [40, 25, 10, 5]$ and $0 \le s \le 0.6$.

The performance of ACWMF is demonstrated in Figure 1 on a 256x256 blurry cameraman image when 25% of the pixels are corrupted by random-valued impulse noise. For the first case (top row), the set having the corrupted pixels are chosen randomly, and from the result obtained from ACWMF, we can still see some features of the cameraman image. For the other case (bottom row), we specify a set containing the damaged pixels, and ACWMF misses quite a lot of real noise and false-hits some noise-free pixels. The success of two-stage methods depends on the accuracy of detecting damaged pixels. Therefore, we propose a method to update the set together with the restoration of the image, instead of estimating the set before restoration as in two-stage methods.

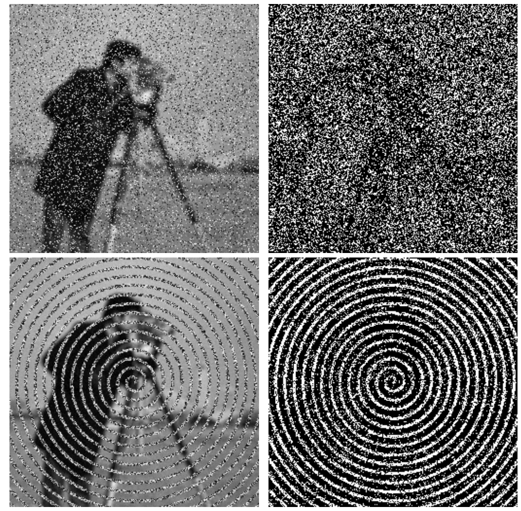

Figure 1: Noisy images and the sets detected by ACWMF. Left column: noisy images corrupted by random-valued noise; Right column: the sets of damaged pixels detected by ACWMF. White point means that the corresponding pixel is corrupted by noise.

3. Blind Inpainting using Binary Matching Pursuit

3.1. Formulation

For a $M \times N$ image, $\Lambda \in \{0,1\}^{M \times N}$ is a binary matrix denoting the undamaged pixels (pixels not corrupted by impulse noise):

$$\Lambda_{i,j} = \begin{cases} 1, & \text{if pixel } (i,j) \in \Lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

We use $\mathcal{P}_\Lambda(\cdot)$ to represent the projection of an image onto a matrix supported by Λ:

$$\mathcal{P}_\Lambda(u)_{i,j} = \begin{cases} 0, & \text{if } \Lambda_{i,j} = 0, \\ u_{i,j}, & \text{if } \Lambda_{i,j} = 1. \end{cases} \tag{8}$$

Given a degraded image f, our objective is to estimate the damaged (or missing) pixels and restore them. We propose the following model to solve this problem:

$$\begin{aligned} \min_{u,v} \quad & S(Hu+v,f) + \lambda_1 J(u) \\ \text{subject to:} \quad & \|v\|_0 \le K, \end{aligned} \tag{9}$$

where $S(Hu+v,f)$ is the fidelity term, $J(u)$ is the regularization term on the true image, The parameter λ_1 is dependent on the noise level of e_1. The higher the noise level, the larger the parameter should be. The parameter K is dependent on the number of damaged pixels.

Assuming that (u^*, v^*) is the optimal solution to the problem (9), when $v^*_{i,j} \neq 0$, we have $v^*_{i,j} = f_{i,j} - (Hu^*)_{i,j}$. Therefore if we denote

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } v_{i,j} \neq 0, \\ 1, & \text{if } v_{i,j} = 0. \end{cases} \tag{10}$$

then problem (9) is equivalent to

$$\begin{array}{rl} \min\limits_{u,\Lambda} & S_\Lambda(Hu,f) + \lambda_1 J(u), \\ \text{subject to:} & \sum_{i,j}(1-\Lambda_{i,j}) \leq K \end{array} \tag{11}$$

where $S_\Lambda(Hu,f)$ is the fidelity term only on the set Λ assumed to be undamaged. Problem (11) can be solved by alternating minimization method, and the algorithm for solving (11) is described below.

3.2. Algorithm

The optimization problem defined in (11) is non-convex and has both continuous and discrete variables. It is still difficult to solve it in the pair (u, Λ), but we can use alternating minimization method, which separates the energy minimization over u and Λ into two steps. For solving the problem of u with Λ fixed, it is a convex optimization problem and finding Λ with u fixed is a combinatorial optimization problem.

1) Finding u: Given an estimate of the support matrix Λ, the minimization over u is just an image inpainting (and deblurring) problem [12]:

$$\min_u S_\Lambda(Hu,f) + \lambda_1 J(u), \tag{12}$$

and there are many existing methods for solving this problem.

2) Finding Λ: Given an estimate of the image u, the minimization over Λ becomes:

$$\begin{array}{rl} \min\limits_{\Lambda} & S_\Lambda(Hu,f), \\ \text{subject to:} & \sum_{i,j}(1-\Lambda_{i,j}) \leq K. \end{array} \tag{13}$$

Since $\Lambda_{ij} \in \{0,1\}$, if the fidelity term is a summation over all pixels assumed to be undamaged, the energy can be rewritten as follows:

$$S_\Lambda(Hu,f) = \sum_{i,j} \Lambda_{i,j}\phi\left((Hu)_{i,j}, f_{i,j}\right) \tag{14}$$

where ϕ is the fidelity term for each pixel.

It can be solved exactly in one step:

$$\Lambda_{i,j} = \begin{cases} 0 & \text{if } \phi\left((Hu)_{i,j}, f_{i,j}\right) > \tau, \\ 1 & \text{if } \phi\left((Hu)_{i,j}, f_{i,j}\right) \leq \tau, \end{cases} \tag{15}$$

where τ is the K^{th} largest value of $\phi\left((Hu)_{i,j}, f_{i,j}\right)$. Therefore, the proposed algorithm for blind inpainting is iteratively finding u and Λ. Since $\Lambda_{i,j}$ can only take values 0 and 1, this can be considered a binary version of matching pursuit [18], named binary matching pursuit (BMP).

The algorithm can be applied to many methods for better performance and one example using framelet for image deblurring when the blurry image is corrupted by Gaussian white noise and impulse noise is described below.

3.2.1 Framelet-Based Deblurring

Though total variation (TV) is popular for regularization term in recent years as it preserves edges, its limitation is that TV-based regularization can not preserve the details and textures very well on the regions of complex structures due to the stair-casing effects [19]. Framelet-based algorithms are introduced in [8, 17] for impulse noise removal. In [17], iterative framelet-based approximation/sparsity deblurring algorithm (IFASDA) and accelerated algorithm of IFASDA (Fast_IFASDA) are proposed to deblur images corrupted by impulse plus Gaussian noise. They both have two steps, the first step is to apply AMF or ACWMF on f to estimate the set Λ^c; the second step is deblurring the image using framelet from $\mathcal{P}_\Lambda(f)$.

The energy functional to be minimized is

$$E(u) := \|\phi(\mathcal{P}_\Lambda(Hu-f))\|_1 + \lambda\|\mathcal{O}_{wH}u\|_1, \tag{16}$$

where $\mathcal{O}_{wH} := [w_1\mathcal{O}_1^2 \; w_2\mathcal{O}_2^T \; \cdots \; w_{17}\mathcal{O}_{17}^T]^T$ and $\phi(x) = \frac{\eta x^2}{\eta+|x|}$. $\{\mathcal{O}_k\}_{k=0}^{17}$ are the matrix representations of the tight framelet filters $\{\tau_k\}_{k=0}^{17}$ under a proper boundary condition [11] (see Section III in [17] for the tight framelet filters $\mathcal{O}$ and weight w). In addition a matrix $\mathcal{O}_H$ is formed by stacking the matrices $\mathcal{O}_k, 1 \leq k \leq 17$ together, that is

$$\mathcal{O}_H = [\mathcal{O}_1^2 \; \mathcal{O}_2^T \; \cdots \; \mathcal{O}_{17}^T]^T \tag{17}$$

Associated with the matrix $\mathcal{O}_H$ and a diagonal matrix

$$\Gamma := \text{diag}(\cdots, \gamma_l, \cdots), \qquad \gamma_l \geq 0, \tag{18}$$

the shrinkage function $Sh_\Gamma : \mathbf{R}^{M\times N} \rightarrow \mathbf{R}^{M\times N}$ is defined as follow:

$$Sh_\Gamma(f) := \mathcal{O}_0^T\mathcal{O}_0 f + \mathcal{O}_H^T \cdot \text{shrink}(\mathcal{O}_H f, \Gamma) \tag{19}$$

where shrink is the componentwise thresholding operator

$$\text{shrink}(x,\Gamma)[l] = \text{sign}(x[l])\max\{|x[l]| - \gamma_l, 0\}. \tag{20}$$

The algorithm is

$$u^{n+1} = Sh_{\Gamma_n}(u^n - \beta_n \nabla J_n(u^n)) \tag{21}$$

where

$$J_n(y) := \|\phi_n(\mathcal{P}_\Lambda(Hy - f))\|_1 + \lambda_n \|\mathcal{O}_{wH} y\|_1, \quad (22)$$

with

$$\phi_n(x) = \frac{\eta_n x^2}{\eta_n + |x|}. \quad (23)$$

The ACWMF can not estimate the set Λ^c perfectly, especially when noise level is high and two types of impulse noise are present, thus frequently updating the set is necessary. The new constraint optimization problem using BMP is

$$\begin{array}{rl} \min\limits_{u,\Lambda} & \|\phi(\mathcal{P}_\Lambda(Hu - f))\|_1 + \lambda_1 \|\mathcal{O}_{wH} u\|_1 \\ \text{subject to:} & \sum_{i,j}(1 - \Lambda_{i,j}) \leq K. \end{array}$$

If alternating minimizing method is utilized, the first step for finding u is the same as IFASDA, and the second step for updating Λ is very easy to find from (15). The modified IFASDA algorithm Ada_IFASDA is shown below:

Input: Given blurred noisy image f, set $n = 1$;
Apply AMF/ACWMF on f to estimate the set Λ^c;
for $n = 1, 2, \cdots$, **do**
  Estimate η_n, β_n, λ_n, Γ_n;
  Compute u^{n+1};
  Stop if u^{n+1} meets a stopping criteria;
  if `mod(n,5)=0` **then**
    Update Λ^c by (15);
  end
  $n = n + 1$;
end

Algorithm 1: Proposed framelet-based deblurring algorithm (Ada_IFASDA).

In fact, we do not need to wait until the step for finding u converges, several iterations in IFASDA are sufficient for the full algorithm to converge. We choose to update the set Λ^c for every five iterations in IFASDA.

Similarly, Fast_IFASDA can also be modified into Ada_Fast_IFASDA by adding the steps for updating Λ^c every five iterations in Fast_IFASDA.

4. Experiments

Test images are blurred by the kernel `fspecial('disk',3)`, and corrupted by Gaussian noise of mean zero and standard deviation $\sigma = 5$. Several noise levels ($s = 25\%, 40\%, 55\%$ for random set, and $s = 25.32\%, 31.40\%, 36.83\%$ for specified set) are added into those blurry and noisy images. To evaluate the quality of the restoration results, peak signal to noise ratio

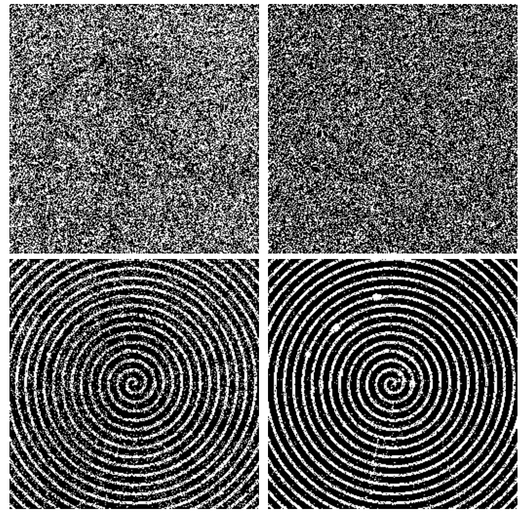

Figure 4: The damaged pixels detected by ACWMF and Ada_IFASDA. Left column: the set obtained by ACWMF; Right column: the set obtained by Ada_IFASDA

(PSNR) is employed. Given an image $u \in [0, 255]^{m \times n}$, the PSNR of the restoration result $\hat{u}$ is defined as follows:

$$\text{PSNR}(\hat{u}, u) = 10 \log_{10} \frac{255^2}{\frac{1}{mn} \sum\limits_{i,j} (\hat{u}_{ij} - u_{ij})^2}. \quad (24)$$

The quantitative qualities (PSNR values) of restored images are listed in Table 1. From Table 1, the performances of Ada_IFASDA and Ada_Fast_IFASDA are better than those of IFASDA and Fast_IFASDA respectively. The restored images of Fast_IFASDA and Ada_Fast_IFASDA for noise levels $s = 55\%$ and $s = 36.83\%$ are shown in Figures 2 and 3 respectively. From the results, a better estimate for the damaged pixels is very crucial to two-stage methods (e.g. IFASDA), especially when the noise level is high. The results show the advantage of simultaneously detecting damaged pixels and restoring images.

Furthermore, we compared the damaged pixels detected by ACWMF and obtained from Ada_IFASDA in Figure 4 for the cameraman image. For the first case where the damaged pixels are chosen randomly ($s = 40\%$), the set obtained from our method is also random and does not contain any information from the image, while the set detected by ACWMF still has some information. For the second case where the set of damaged pixels is not random ($s = 31.40\%$), the set obtained by our method is still better than the set from ACWMF.

In the modified algorithms, parameter K plays an important role in the results of restored images. If we know the number of damaged pixels (pixels in Λ^c) $|\Lambda^c|$, then K can be chosen as the exact number of damaged pixels. However

Figure 2: The restored results of images blurred by `fspecial('disk',3)` and corrupted by random-valued noise (level $s = 55\%$) at random set and Gaussian noise (STD $\sigma = 5$). Top row: blurry and noisy images; Middle row: the results restored by Fast_IFASDA; Bottom row: the results restored Ada_Fast_IFASDA.

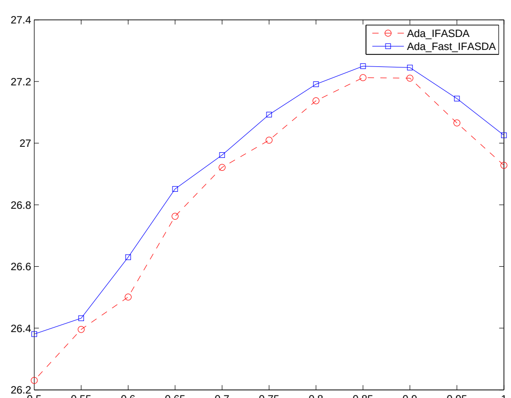

Figure 5: PSNR values for different $K/|\Lambda^c|$ for cameraman image when the level of random-valued noise is 40%.

the difference between Hu and f may be very small at some damaged pixels and these pixels may not be considered as damaged. Thus, a number slightly less than $|\Lambda^c|$ is better. To find a best rate for $K/|\Lambda^c|$, we test on the cameraman image when the level of noise is 40%, and the results are shown in Figure 5. For both methods, we can obtain image with highest PSNR when the rate is 0.85, and this number is chosen for all previous experiments.

5. Conclusion

This paper present a general algorithm using binary matching pursuit for blind image inpainting and removing impulse noise by iteratively restoring the image and identifying the damaged pixels. One method based on framelet is introduced and it is shown in the experiments that the proposed method outperforms other methods for image deblurring in the presence of Gaussian white noise and random-valued impulse noise. It can easily be applied to other methods and more difficult case where there are multiple types of impulse noise.

Acknowledgment

The author would like to thank prof. L Shen for providing the codes of IFASDA and Fast_IFASDA. This work was supported by NSF Grant DMS-0714945 and Center for

Figure 3: The restored results of images blurred by `fspecial('disk',3)` and corrupted by random-valued noise (level $s = 36.83\%$) at specific set and Gaussian noise (STD $\sigma = 5$). Top row: blurry and noisy images; Middle row: the results restored by Fast_IFASDA; Bottom row: the results restored Ada_Fast_IFASDA.

Domain-Specific Computing (CDSC) under the NSF Expeditions in Computing Award CCF-0926127.

References

[1] L. Bar, N. Kiryati, and N. Sochen. Image deblurring in the presence of impulsive noise. *International Journal of Computer Vision*, 70:279–298, 2006. 1

[2] L. Bar, N. A. Sochen, and N. Kiryati. Image deblurring in the presence of salt-and-pepper noise. In R. Kimmel, N. A. Sochen, and J. Weickert, editors, *Scale-Space*, volume 3459 of *Lecture Notes in Computer Science*, pages 107–118. Springer, 2005. 1

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 417–424, 2000. 1

[4] M. Bertalmío, L. A. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003. 1

[5] A. C. Bovik. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, Inc., Orlando, FL, USA, 2005. 1

[6] J. Cai, R. H. Chan, and M. Nikolova. Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise. *Inverse Problem and Imaging*, 2:187–204, 2008. 1, 2

[7] J. Cai, R. H. Chan, and M. Nikolova. Fast two-phase image deblurring under impulse noise. *Journal of Mathematical Imaging and Vision*, 36:46–53, 2010. 1, 2

[8] J. Cai, R. H. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24:131–149, 2008. 1, 3

[9] R. Chan, C. Hu, and M. Nikolova. An iterative procedure for removing random-valued impulse noise. *IEEE Signal Processing Letters*, 11(12):921–924, 2004. 1

[10] R. H. Chan, C.-W. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*, 14(10):1479–1485, 2005. 1

[11] R. H. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen. Tight frame: an efficient way for high-resolution image reconstruction. *Applied and Computational Harmonic Analysis*, 17(1):91–115, 2004. 3

[12] T. Chan and J. Shen. Variational image inpainting. *Communications on Pure and Applied Mathematics*, 58:579–619, 2005. 3

Algorithn	"Lena" PSNR	"Lena" Times	"Goldhill" PSNR	"Goldhill" Times	"Cameraman" PSNR	"Cameraman" Times	"Boat" PSNR	"Boat" Times	Case
	Random-Valued Noise at Random Set								
IFASDA	27.20	49	26.23	85	**24.71**	51	26.59	44	STD $\sigma = 5$ $s = 25\%$
Ada_IFASDA	**27.59**	83	**26.34**	52	24.70	45	**26.99**	84	
Fast_IFASDA	27.43	49	26.23	44	25.03	54	26.83	44	
Ada_Fast_IFASDA	**27.61**	54	**26.37**	48	**25.21**	82	**27.03**	44	
IFASDA	27.02	82	26.00	51	24.27	45	26.38	87	STD $\sigma = 5$ $s = 40\%$
Ada_IFASDA	**27.30**	64	**26.17**	49	**24.47**	45	**26.62**	68	
Fast_IFASDA	27.04	52	26.03	46	24.63	54	26.41	44	
Ada_Fast_IFASDA	**27.37**	68	**26.21**	55	**24.90**	81	**26.66**	48	
IFASDA	24.44	59	25.01	43	22.65	43	25.24	48	STD $\sigma = 5$ $s = 55\%$
Ada_IFASDA	**26.33**	76	**25.69**	47	**23.71**	48	**26.01**	55	
Fast_IFASDA	24.15	80	24.99	81	22.70	80	25.32	69	
Ada_Fast_IFASDA	**26.15**	80	**25.85**	81	**24.52**	82	**26.20**	79	
	Random-Valued Noise at Specific Set								
IFASDA	26.27	40	25.44	39	23.99	40	26.10	41	STD $\sigma = 5$ $s = 25.32\%$
Ada_IFASDA	**26.97**	43	**25.90**	39	**24.45**	42	**26.46**	42	
Fast_IFASDA	26.57	59	25.62	58	24.55	61	26.36	59	
Ada_Fast_IFASDA	**27.30**	59	**26.25**	60	**25.03**	60	**26.81**	54	
IFASDA	24.91	42	24.66	40	23.19	42	25.27	40	STD $\sigma = 5$ $s = 31.40\%$
Ada_IFASDA	**26.32**	43	**25.70**	44	**24.36**	46	**26.00**	40	
Fast_IFASDA	24.92	58	24.93	59	23.54	59	25.68	59	
Ada_Fast_IFASDA	**27.00**	59	**26.01**	59	**24.76**	61	**26.50**	59	
IFASDA	22.77	44	23.13	43	21.50	42	23.82	42	STD $\sigma = 5$ $s = 36.83\%$
Ada_IFASDA	**25.69**	47	**24.95**	44	**23.79**	47	**25.65**	46	
Fast_IFASDA	17.73	58	22.82	58	19.11	58	23.77	58	
Ada_Fast_IFASDA	**26.47**	59	**25.60**	58	**24.44**	59	**26.03**	59	

Table 1: PSNR(dB) and CPU computing time (seconds) for deblurred results of different algorithms for blurred images corrupted by random-valued noise plus Gaussian noise. The images are blurred by the blurring kernel `fspecial('disk',3)`.

[13] T. F. Chan, J. Shen, and H. Zhou. Total variation wavelet inpainting. *Journal of Mathematical Imaging and Vision*, 25:107–125, 2006. 1

[14] T. Chen and H. R. Wu. Adaptive impulse dectection using center-weighted median filters. *IEEE Signal Processing Letters*, 8(1):1–3, 2001. 1, 2

[15] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008. 1

[16] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. 1

[17] Y. Li, L. Shen, D. Dai, and B. Suter. Framelet algorithms for de-blurring images corrupted by impulse plus Gaussian noise. *IEEE Transactions on Image Processing*, 20(7):1822–1837, 2011. 1, 2, 3

[18] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397 –3415, 1993. 3

[19] Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations*, volume 22 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2001. The fifteenth Dean Jacqueline B. Lewis memorial lectures. 3

[20] M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20:99–120, 2004. 1

[21] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physics D*, 60:259–268, 1992. 1

[22] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Proceedings of 1st International Conference on Image Processing*, volume 1, pages 31–35. IEEE Comput. Soc. Press, 1994. 1

[23] Y. Xiao, T. Zeng, J. Yu, and M. Ng. Restoration of images corrupted by mixed Gaussian-impulse noise via l_1-l_0 minimization. *Pattern Recognition*, 44:1708–1720, 2011. 1