Institute for Computational Mathematics
Hong Kong Baptist University

ICM Research Report
08-01

Fast Image Restoration Methods for Impulse and Gaussian Noises Removal

Yu-Mei Huang [*] Michael K. Ng [†] You-Wei Wen [‡]

May 17, 2008

Abstract

In this paper, we study the restoration of blurred images corrupted by impulse noise or mixed impulse plus Gaussian noises. In the proposed method, we use the modified total variation minimization scheme to regularize the deblurred image and fill in suitable values for noisy image pixels where these are detected by median-type filters. An alternating minimization algorithm is employed to solve the proposed total variation minimization problem. We will show the convergence of the alternating minimization algorithm and demonstrate that the algorithm is very efficient. Our experimental results show that the quality of restored images by the proposed method is competitive with those restored by the existing variational image restoration methods.

Keywords: deblurring, denoising, impulse noise, Gaussian noise, total variation.

1 Introduction

Digital image restoration and reconstruction play an important part in various areas of applied sciences such as medical and astronomical imaging, film restoration, image and video coding. In this paper, we focus on two degradation models: an ideal image $\boldsymbol{f} \in \mathbb{R}^{n^2}$ is observed in the presence of a spatial-invariant blur matrix $\boldsymbol{H} \in \mathbb{R}^{n^2 \times n^2}$, an additive zero-mean Gaussian white noise $\boldsymbol{n} \in \mathbb{R}^{n^2}$ of standard deviation σ, and an impulse noise to the degraded image. Let $\mathbb{N}_{\text{imp}}$ denote the process of image degradation with impulse noise. Thus the observed image $\boldsymbol{g} \in \mathbb{R}^{n^2}$ is obtained by:

$$\boldsymbol{g} = \mathbb{N}_{\text{imp}}\left(\boldsymbol{H}\boldsymbol{f} + \boldsymbol{n}\right). \tag{1.1}$$

[*]Centre for Mathematical Imaging and Vision and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, and School of Information Science and Engineering, Lanzhou University, Gansu, P. R. China.

[†]The Corresponding Author. Centre for Mathematical Imaging and Vision and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: mng@math.hkbu.edu.hk. Research supported in part by RGC 7045/05P, and HKBU FRGs.

[‡]Faculty of Science, South China Agricultural University, Wushan, Guangzhou, P.R.China. Research supported in part by NSFC Grant No. 60702030.

Another noise model is that the blurred image is corrupted by an impulse noise, and then by a Gaussian noise:

$$\boldsymbol{g} = \mathbb{N}_{\text{imp}}\left(\boldsymbol{H}\boldsymbol{f}\right) + \boldsymbol{n}. \tag{1.2}$$

Salt-and-pepper noise and random-valued noise are the two common types of impulse noise. They degrade an image in a totally different way from that by Gaussian white noise. Suppose $\boldsymbol{u}_{i,j}$ ($(i,j) \in \mathcal{I} = \{1,2,\cdots,n\}\times\{1,2,\cdots,n\}$) is the gray level of an image $\boldsymbol{u}$ at location (i,j), and $[a_{\min}, a_{\max}]$ is the dynamic range of $\boldsymbol{u}$. The salt-and-pepper noisy image $\tilde{\boldsymbol{u}}$ is given as follows:

$$\widetilde{\boldsymbol{u}}_{i,j} = \begin{cases} a_{\min}, & \text{with probability } p, \\ a_{\max}, & \text{with probability } q, \\ \boldsymbol{u}_{i,j}, & \text{with probability } 1-(p+q). \end{cases} \tag{1.3}$$

where $s = p + q$ is the noise ratio which determines the noise level of salt-and-pepper noise.We note that corrupted pixel values take either the maximum value $a_{\max}$ or the minimum value $a_{\min}$. On the other hand, corrupted pixel values contaminated by random-valued noise can take any random number a which is uniformly distributed in $[a_{\min}, a_{\max}]$, i.e.,

$$\widetilde{\boldsymbol{u}}_{i,j} = \begin{cases} a_{i,j}, & \text{with probability } r, \\ \boldsymbol{u}_{i,j}, & \text{with probability } 1-r. \end{cases} \tag{1.4}$$

Here r is the noise ratio which determines the noise level of the random-valued noise.

In the literature, there are several image processing methods [14, 11, 13, 21] based on median-type filters for impulse noise removal. The main idea of these filters is that the locations of possible noisy pixels are determined and their values are replaced by median values calculated from their corresponding neighborhoods. These methods are superior in identifying the locations of noisy pixels. Since the pixels in the vicinity of edges may be replaced by median values without taking into account of local features, the edges in the recovered image are usually smeared, especially when the impulse noise level is high. Another approach is to use variational methods [1, 3, 2, 19] for impulse noise removal. The advantage of this approach is that edges can be recovered effectively. Recently, Chan *et al.* [10, 9] proposed a two-phase method for impulse noise removal. Their idea is to use median filters to identify the noisy pixels and then employ an variational method to fill in the suitable values to such noisy pixels. In [5], they further extended their scheme to restore blurred images corrupted by both impulse and Gaussian noises together.

It is well-known that restoring an image $\boldsymbol{f}$ is a very ill-conditioned problem. A regularization method should be used in the image restoration process. The total variation (TV) regularization, proposed by Rudin, Osher and Fatemi [22], has become very popular for this purpose. The main advantage of the TV formulation is the ability to preserve edges in the image due to the piecewise smooth regularization property of the TV norm. In this paper, we use median-type filters to identify the possible noisy pixels, and then we employ a fast total variation minimization method for image restoration in (1.1) or (1.2). We remark that the TV regularization is not used in [10, 9, 5]. An alternating minimization algorithm is employed to solve the proposed total variation minimization problem. We will show the convergence of the alternating minimization algorithm and demonstrate that the algorithm is very efficient. Our experimental results show that the quality of restored images by the proposed method is competitive with those restored by the existing variational restoration methods.

The outline of this paper is as follows. In Section 2, we present the proposed algorithm. In Section 3, numerical examples are given to demonstrate the effectiveness of the proposed model. Finally, some concluding remarks are given in Section 4.

2 The Proposed Model

Two effective median filters are adaptive median filter (AMF) and adaptive center-weighted median filter (ACWMF) [14, 11, 13, 21] are employed in this paper. Suppose $\boldsymbol{y} \in \mathbb{R}^{n^2}$ is the output by median-type filters. The candidates of noisy pixels can be determined as follows:

- For salt-and-pepper noise:

$$\mathcal{N} = \{(i,j) \in \mathcal{I} \ : \ \boldsymbol{y}_{i,j} \neq \boldsymbol{g}_{i,j} \ \text{ and } \ \boldsymbol{g}_{ij} \in \{d_{\min}, d_{\max}\}\},$$

  where $d_{\min}$ and $d_{\max}$ is the minimum and the maximum values of the recorded image;

- For random-valued impulse noise:

$$\mathcal{N} = \{(i,j) \in \mathcal{I} \ : \ \boldsymbol{y}_{i,j} \neq \boldsymbol{g}_{i,j}\}.$$

We note that impulse noise-free pixels can be defined as follows:

$$\boldsymbol{x}_{i,j} = \begin{cases} 0 & \text{if } (i,j) \ \in \mathcal{N} \\ 1 & \text{otherwise.} \end{cases}$$

After we identify impulse noise-free pixels, we can apply the total variation minimization for restoring an image and filling in suitable values for noisy image pixels. As it is more difficult to detect random-valued impulse noise pixels, we propose to repeat the noisy pixels detection procedure and total variation minimization image restoration procedure a few times so that a high quality of restored image can be obtained, see the numerical results in the next section.

2.1 The Minimization Model

In this paper, we propose to minimize the following objective function for image restoration after the detection of noisy pixels:

$$\min_{\boldsymbol{f},\boldsymbol{u}} \ \mathcal{J}(\boldsymbol{f},\boldsymbol{u}) \equiv \min_{\boldsymbol{f},\boldsymbol{u}} \ \|\boldsymbol{X}(\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g})\|_2^2 + \alpha_1\|\boldsymbol{f}-\boldsymbol{u}\|_2^2 + \alpha_2\|\boldsymbol{u}\|_{TV}, \tag{2.5}$$

where $\|\cdot\|_2$ is the Euclidean norm, $\boldsymbol{X}$ is a nonzero diagonal matrix with its diagonal entries given by $\boldsymbol{x}_{i,j}$, α_1 and α_2 are two positive regularization parameters, and $\|\cdot\|_{TV}$ is the discrete TV regularization term. The discrete total variation of $\boldsymbol{f}$ is defined by

$$\|\boldsymbol{f}\|_{TV} := \sum_{1\le j,k\le n} |(\nabla \boldsymbol{f})_{j,k}|_2 = \sum_{1\le j,k\le n} \sqrt{|(\nabla \boldsymbol{f})^x_{j,k}|^2 + |(\nabla \boldsymbol{f})^y_{j,k}|^2}.$$

where $|\cdot|_2$ is the Euclidean norm in $\mathbb{R}^2$. Suppose $\boldsymbol{f}$ is a n-by-n image, the discrete gradient operator $\nabla : \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$ is defined by

$$(\nabla \boldsymbol{f})_{j,k} = ((\nabla \boldsymbol{f})^x_{j,k}, (\nabla \boldsymbol{f})^y_{j,k})$$

with

$$(\nabla \boldsymbol{f})^x_{j,k} = \begin{cases} \boldsymbol{f}_{j+1,k} - \boldsymbol{f}_{j,k} & \text{if } j < n, \\ 0 & \text{if } j = n, \end{cases} \qquad (\nabla \boldsymbol{f})^y_{j,k} = \begin{cases} \boldsymbol{f}_{j,k+1} - \boldsymbol{f}_{j,k} & \text{if } k < n, \\ 0 & \text{if } k = n \end{cases} \tag{2.6}$$

for $j, k = 1, \ldots, n$. Here $\boldsymbol{f}_{j,k}$ refers to the $(jn+k)$th entry of the vector $\boldsymbol{f}$ (it is the (j,k)th pixel location of the image).

In (2.5), we can interpret the total variation minimization scheme to regularize the deblurred image pixels (for $\boldsymbol{x}_{i,j} = 1$) and to fill in suitable values to noisy image pixel values (for $\boldsymbol{x}_{i,j} = 0$). Here α_1 measures the trade off between a deblurred image $\boldsymbol{f}$ and a regularized image $\boldsymbol{u}$, and α_2 measures the amount of regularization to those noisy image pixel values. The main advantage of the proposed method is that a TV norm is used in the image restoration process. Therefore the new method has the ability to preserve edges very well in the restored image. An alternating minimization algorithm is employed to solve the proposed total variation minimization problem (2.5).

2.2 Alternating Minimization Algorithm

In (2.5), there are two unknown images. One is the deblurred image $\boldsymbol{f}$ and the other is the regularized image $\boldsymbol{u}$. We propose to use an alternating minimization algorithm to solve (2.5). Starting from an initial guess $\boldsymbol{u}^{(0)}$, this method computes a sequence of iterates

$$\boldsymbol{f}^{(1)}, \boldsymbol{u}^{(1)}, \boldsymbol{f}^{(2)}, \boldsymbol{u}^{(2)}, \ldots, \boldsymbol{f}^{(i)}, \boldsymbol{u}^{(i)}, \ldots.$$

such that

$$\begin{cases} \mathcal{S}_h(\boldsymbol{u}^{(i-1)}) := \boldsymbol{f}^{(i)} = \min_{\boldsymbol{f}} \|\boldsymbol{X}(\boldsymbol{H}\boldsymbol{f} - \boldsymbol{g})\|_2^2 + \alpha_1 \|\boldsymbol{f} - \boldsymbol{u}^{(i-1)}\|_2^2 \\ \mathcal{S}_{tv}(\boldsymbol{f}^{(i)}) := \boldsymbol{u}^{(i)} = \operatorname{argmin}_{\boldsymbol{u}} \alpha_1 \|\boldsymbol{f}^{(i)} - \boldsymbol{u}\|_2^2 + \alpha_2 ||\boldsymbol{u}||_{TV} \end{cases}$$

for $i = 1, 2, \cdots$. Therefore, we can express the following relationship between $\boldsymbol{u}^{(i)}$ and $\boldsymbol{u}^{(i-1)}$:

$$\boldsymbol{u}^{(i)} = \mathcal{S}_{tv}(\mathcal{S}_h(\boldsymbol{u}^{(i-1)})), \quad i = 1, 2, \cdots.$$

For simplicity, we denote

$$\boldsymbol{u}^{(i)} = \mathcal{T}\left(\boldsymbol{u}^{(i-1)}\right), \tag{2.7}$$

where

$$\mathcal{T}(\cdot) = \mathcal{S}_{tv}(\mathcal{S}_h(\cdot)).$$

In the next subsection, we will analyze the convergence of $\boldsymbol{u}^{(i)}$ under $\mathcal{T}$.

Let us first study the computational cost of the alternating minimization algorithm. The first step of the method is to perform the deblurring. The minimizer of the optimization problem:

$$\min_{\boldsymbol{f}} \ \|\boldsymbol{X}(\boldsymbol{H}\boldsymbol{f} - \boldsymbol{g})\|_2^2 + \alpha_1 \|\boldsymbol{f} - \boldsymbol{u}^{(i-1)}\|_2^2$$

is equivalent to solving a linear system:

$$(\boldsymbol{H}^t \boldsymbol{X} \boldsymbol{H} + \alpha_1 \boldsymbol{I})\boldsymbol{f} = \boldsymbol{H}^t \boldsymbol{X} \boldsymbol{g} + \alpha_1 \boldsymbol{u}^{(i-1)}. \tag{2.8}$$

Because of the regularization term $\alpha_1 \boldsymbol{I}$, the coefficient matrix $\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{H} + \alpha_1\boldsymbol{I}$ is always invertible as $\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{H}$ is singular. We remark in image restoration that $\boldsymbol{H}$ is usually a matrix of block Toeplitz with Toeplitz blocks (BTTB) when zero boundary conditions are applied, and block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB) when Neumann boundary conditions are used [16]. The conjugate gradient method can be used to solve (2.8) at each iteration. Convergence can be improved using preconditioning techniques. Transform-based preconditioning techniques have been proved to be very successful [15]. For instance, if $\boldsymbol{H}$ is a blurring matrix generated by a symmetric point spread function, $\boldsymbol{H}$ can be diagonalized by a fast transform matrix: then (2.8) is solved by using three fast transforms in $O(n^2 \log n)$ operations for an n-by-n restored image, see for instance [16].

The second step of the method is to apply an exact TV regularization scheme to the image generated by the previous deblurring step. The minimizer of the optimization problem

$$\alpha_1 \|\boldsymbol{f}^{(i)} - \boldsymbol{u}\|_2^2 + \alpha_2 ||\boldsymbol{u}||_{TV}$$

can be solved by many TV denoising methods like Chambolle's projection algorithm [6], semismooth Newton's method [17], multilevel optimization method [8] and graph-based optimization method [7]. In this paper, we employ the Chambolle projection algorithm in the denoising step. In the Chambolle scheme, we solve the following constrained minimization problem:

$$\min_{\boldsymbol{p}} \left\| \boldsymbol{f}^{(i)} - \frac{\alpha_2}{\alpha_1} \operatorname{div}\boldsymbol{p} \right\|_2^2 \tag{2.9}$$

subject to

$$|\boldsymbol{p}_{j,k}| \leq 1, \quad \forall 1 \leq j,k \leq n.$$

Here

$$\boldsymbol{p}_{j,k} = \begin{bmatrix} \boldsymbol{p}_{j,k}^x \\ \boldsymbol{p}_{j,k}^y \end{bmatrix}$$

is the dual variable at the (j,k)th pixel location, $\boldsymbol{p}$ is the concatenation of all $\boldsymbol{p}_{j,k}$, and the discrete divergence of $\boldsymbol{p}$ is defined such that

$$(\operatorname{div}\boldsymbol{p})_{j,k} \equiv \boldsymbol{p}_{j,k}^x - \boldsymbol{p}_{j-1,k}^x + \boldsymbol{p}_{j,k}^y - \boldsymbol{p}_{j,k-1}^y$$

with $\boldsymbol{p}_{0,k}^x = \boldsymbol{p}_{j,0}^y = 0$. The vector $\operatorname{div}\boldsymbol{p}$ is the concatenation of all $(\operatorname{div}\boldsymbol{p})_{j,k}$. For simplicity, we denote $\beta = 2\alpha_2/\alpha_1$. When the minimizer $\boldsymbol{p}^*$ of the constrained optimization problem in (2.9) is determined, the denoised image $\boldsymbol{u}^{(i)}$ can be generated as follows:

$$\boldsymbol{u}^{(i)} = \boldsymbol{f}^{(i)} - \beta \operatorname{div}\boldsymbol{p}^*.$$

In [6], the iterative scheme for computing the optimal solution $\boldsymbol{p}$ is given as follows:

$$\boldsymbol{p}_{j,k}^{(l+1,x)} = \frac{\boldsymbol{p}_{j,k}^{(l,x)} + \gamma\beta\nabla\left(\beta \operatorname{div}\boldsymbol{p}^{(l)} - \boldsymbol{f}^{(i)}\right)_{j,k}^x}{1 + \gamma\beta|\nabla\left(\beta \operatorname{div}\boldsymbol{p}^{(l)} - \boldsymbol{f}^{(i)}\right)_{j,k}|}, \quad \forall 1 \leq j,k \leq n$$

and

$$\boldsymbol{p}_{j,k}^{(l+1,y)} = \frac{\boldsymbol{p}_{j,k}^{(l,y)} + \gamma\beta\nabla\left(\beta\ \mathrm{div}\boldsymbol{p}^{(l)} - \boldsymbol{f}^{(i)}\right)_{j,k}^{y}}{1 + \gamma\beta|\nabla\left(\beta\ \mathrm{div}\boldsymbol{p}^{(l)} - \boldsymbol{f}^{(i)}\right)_{j,k}|}, \quad \forall 1 \leq j,k \leq n$$

where $\boldsymbol{p}_{j,k}^{(l,z)}$ $(z \in \{x,y\})$ is the lth iterate of the iterative method for the minimizer, $\nabla(\cdot)_{j,k}^{z}$ $(z \in \{x,y\})$ is defined as in (2.6), and γ is the step size introduced in the projection gradient method, see [6] for details.

2.3 Convergence Analysis

In this subsection, by making use of the Opial theorem [20] we would like to show that the algorithm converges to a stationary point of $\mathcal{J}$. We can show that $\mathcal{J}$ in (2.5) is coercive and $\mathcal{T}$ in (2.7) is non-expansive and asymptotically regular.

Definition 2.1. *An operator $\mathcal{P}$ is called non-expansive if for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{R}^{n^2}$, we have*

$$||\mathcal{P}(\boldsymbol{x}_1) - \mathcal{P}(\boldsymbol{x}_2)||_2 \leq ||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2.$$

If there exists some non-expansive operator $\mathcal{A}$ and $\alpha \in (0,1)$ such that $\mathcal{P} = (1-\alpha)\mathcal{I} + \alpha\mathcal{A}$, then $\mathcal{P}$ is called α–averaged non-expansive.

Lamma 2.1. *[12, Lemma 2.4] Let φ be convex and semi-continuous and $\alpha > 0$. Suppose $\widehat{\boldsymbol{x}}$ is defined as follows:*

$$\widehat{\boldsymbol{x}} = \mathrm{argmin}_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \alpha\varphi(\boldsymbol{x}). \tag{2.10}$$

Define $\mathcal{S}$ such that $\widehat{\boldsymbol{x}} = \mathcal{S}(\boldsymbol{y})$ for each $\boldsymbol{y}$. Then $\mathcal{S}$ is $\frac{1}{2}$-averaged non-expansive.

Next we show that the operator $\mathcal{T}$ defined in (2.7) is non-expansive.

Lamma 2.2. *The operator $\mathcal{T}$ in (2.7) is non-expansive.*

Proof. With Lemma 2.1, we know that $\mathcal{S}_{tv}$ is non-expansive, for any $\boldsymbol{x}$ and $\boldsymbol{y}$, we have

$$\begin{aligned}
& \|\mathcal{T}(\boldsymbol{x}) - \mathcal{T}(\boldsymbol{y})\|_2 \\
= \ & ||\mathcal{S}_{tv}(\mathcal{S}_h(\boldsymbol{x})) - \mathcal{S}_{tv}(\mathcal{S}_h(\boldsymbol{y}))||_2 \\
\leq \ & ||\mathcal{S}_h(\boldsymbol{x}) - \mathcal{S}_h(\boldsymbol{y})||_2 \\
= \ & \left\|\left(\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{H} + \alpha_1\boldsymbol{I}\right)^{-1}\left(\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{g} + \alpha_1\boldsymbol{x}\right) - \left(\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{H} + \alpha_1\boldsymbol{I}\right)^{-1}\left(\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{g} + \alpha_1\boldsymbol{y}\right)\right\|_2 \\
= \ & \left\|\alpha_1\left(\boldsymbol{H}^t\boldsymbol{X}\boldsymbol{H} + \alpha_1\boldsymbol{I}\right)^{-1}(\boldsymbol{x} - \boldsymbol{y})\right\|_2 \\
\leq \ & \|\boldsymbol{x} - \boldsymbol{y}\|_2 .
\end{aligned}$$

The result follows. ∎

Lamma 2.3. *Let $\{\boldsymbol{u}^{(i)}\}$ be generated by (2.7). Then $\sum_{i=1}^{\infty}\left\|\boldsymbol{u}^{(i-1)} - \boldsymbol{u}^{(i)}\right\|_2^2$ converges.*

Immediately, we have the following lemma, which states that the operator $\mathcal{T}$ is asymptotically regular.

Lamma 2.4. *For any initial guess $\boldsymbol{u}^{(0)} \in \mathbb{R}^{n^2}$, suppose $\{\boldsymbol{u}^{(i)}\}$ is generated by (2.7), then $\mathcal{T}$ is asymptotically regular, i.e.,*

$$\lim_{i\to\infty} \left\|\boldsymbol{u}^{(i+1)} - \boldsymbol{u}^{(i)}\right\|_2 = \lim_{i\to\infty} \left\|\mathcal{T}^{i+1}\left(\boldsymbol{u}^{(0)}\right) - \mathcal{T}^{i}\left(\boldsymbol{u}^{(0)}\right)\right\|_2 = 0.$$

To show the coerciveness of $\mathcal{J}$, we introduce the following definitions.

Definition 2.2. *A function $\phi : \mathbb{R}^{n^2} \to \mathbb{R}$ is proper over a set $X \subset \mathbb{R}^{n^2}$ if $\phi(\boldsymbol{x}) < \infty$ for at least one $\boldsymbol{x} \in X$ and $\phi(\boldsymbol{x}) > -\infty$ for all $\boldsymbol{x} \in X$. A function $\phi : \mathbb{R}^{n^2} \to \mathbb{R}$ is coercive over a set $X \subset \mathbb{R}^{n^2}$ if for every sequence $\{\boldsymbol{x}_k\} \subset X$ such that $||\boldsymbol{x}_k||_2 \to \infty$, we have*

$$\lim_{k\to\infty} \phi(\boldsymbol{x}_k) = \infty.$$

When $X = \mathbb{R}^{n^2}$, we say that ϕ is coercive on $\mathbb{R}^{n^2}$.

Lamma 2.5. *[4, Proposition 2.1.1] Let $\phi : \mathbb{R}^{n^2} \to \mathbb{R}$ be a closed, proper and coercive function. Then the set of minima of ϕ over $\mathbb{R}^{n^2}$ is nonempty and compact.*

The following lemma states the objective function $\mathcal{J}(\boldsymbol{f}, \boldsymbol{u})$ is coercive under certain conditions.

Lamma 2.6. *Let $\boldsymbol{L}_h$ and $\boldsymbol{L}_v$ be the one-side difference matrix on the horizontal direction and the vertical direction respectively, and*

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_v \end{pmatrix}.$$

The function $\mathcal{J}(\boldsymbol{f}, \boldsymbol{u})$ is coercive if Null $(\boldsymbol{XH}) \bigcap$ Null $(\boldsymbol{L}) = \emptyset$, where Null $(\cdot)$ denotes the null space of the corresponding matrix.

Proof. The lower bound of the discrete total variation is given by

$$\begin{aligned} \|\boldsymbol{u}\|_{TV} &= \sum_{1\le j,k\le n} |(\nabla \boldsymbol{u})_{j,k}| \\ &= \sum_{1\le j,k\le n} \sqrt{((\nabla \boldsymbol{u})^x_{j,k})^2 + ((\nabla \boldsymbol{u})^y_{j,k})^2} \\ &\ge \frac{1}{\sqrt{2}} \sum_{1\le j,k\le n} \left|(\nabla \boldsymbol{u})^x_{j,k}\right| + \left|(\nabla \boldsymbol{u})^y_{j,k}\right| \;=\; \frac{1}{\sqrt{2}} \|\boldsymbol{Lu}\|_1 . \end{aligned}$$

By using the above inequality, we have

$$\begin{aligned} & \mathcal{J}(\boldsymbol{f}, \boldsymbol{u}) && (2.11) \\ \ge\; & \|\boldsymbol{X}(\boldsymbol{Hf} - \boldsymbol{g})\|_2^2 + \alpha_1 ||\boldsymbol{f} - \boldsymbol{u}||_2^2 + \frac{\alpha_2}{\sqrt{2}} \|\boldsymbol{Lu}\|_1 \\ =\; & \left\| \begin{pmatrix} \boldsymbol{XH} & 0 \\ \sqrt{\alpha_1}\boldsymbol{I} & -\sqrt{\alpha_1}\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{u} \end{pmatrix} - \begin{pmatrix} \boldsymbol{Xg} \\ \boldsymbol{0} \end{pmatrix} \right\|_2^2 + \frac{\alpha_2}{\sqrt{2}} \left\| \begin{pmatrix} 0 & \boldsymbol{L} \end{pmatrix} \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{u} \end{pmatrix} \right\|_1, && (2.12) \end{aligned}$$

where $\mathbf{0}$ is the zero vector. Let

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{XH} & 0 \\ \sqrt{\alpha_1}\boldsymbol{I} & -\sqrt{\alpha_1}\boldsymbol{I} \\ 0 & \frac{\alpha_2}{\sqrt{2}}\boldsymbol{L} \end{pmatrix} \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{u} \end{pmatrix}.$$

We note that

$$\begin{pmatrix} \boldsymbol{XH} & 0 \\ \sqrt{\alpha_1}\boldsymbol{I} & -\sqrt{\alpha_1}\boldsymbol{I} \\ 0 & \boldsymbol{L} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & 0 & 0 \\ 0 & \boldsymbol{I} & 0 \\ & -\frac{\alpha_2}{\sqrt{2}\alpha_1}\boldsymbol{L} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{XH} & 0 \\ \sqrt{\alpha_1}\boldsymbol{I} & -\sqrt{\alpha_1}\boldsymbol{I} \\ \frac{\alpha_2}{\sqrt{2}}\boldsymbol{L} & 0 \end{pmatrix}$$

and the above matrix is full rank as $Null\ (\boldsymbol{XH}) \bigcap Null\ (\boldsymbol{L}) = \emptyset$. Hence when $\left\| \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{u} \end{pmatrix} \right\|_2$ tends to infinity, either $\left\| \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \right\|_2$ or $\|\boldsymbol{z}\|_2$ tends to infinity, therefore by using (2.12), $\mathcal{J}(\boldsymbol{f}, \boldsymbol{u})$ also tends to infinity. The result follows. ∎

We remark that if $\boldsymbol{f} \in Null(\boldsymbol{L})$, then $\boldsymbol{f}_{i,j} = c$ for any i and j, where c is a nonzero constant. Since $\boldsymbol{H}$ is a blurring matrix (all the nonzero entries should be positive), it is clear that $\boldsymbol{XHf}$ is a nonzero vector whenever $\boldsymbol{X}$ is not a zero matrix. It follows that the assumption $Null\ (\boldsymbol{XH}) \bigcap Null\ (\boldsymbol{L}) = \emptyset$ holds in general.

Now we show that the set of fixed points of $\mathcal{T}$ is non-empty.

Lamma 2.7. *Suppose $Null\ (\boldsymbol{XH}) \bigcap Null\ (\boldsymbol{L}) = \emptyset$. Then the set of fixed points of $\mathcal{T}$ is non-empty.*

Proof. Since the objective function $\mathcal{J}$ is coercive, the set of minimizers of $\mathcal{J}$ is non-empty. Assume $(\boldsymbol{f}', \boldsymbol{u}')$ is a minimizer of $\mathcal{J}(\boldsymbol{f}, \boldsymbol{u})$, i.e.,

$$\begin{pmatrix} \frac{\partial \mathcal{J}}{\partial \boldsymbol{f}}(\boldsymbol{f}', \boldsymbol{u}') \\ \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}}(\boldsymbol{f}', \boldsymbol{u}') \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}.$$

Therefore we have

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{f}}(\boldsymbol{f}', \boldsymbol{u}') = \mathbf{0} \quad \text{and} \quad \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}}(\boldsymbol{f}', \boldsymbol{u}') = \mathbf{0}.$$

It implies that

$$\begin{cases} \boldsymbol{f}' = \mathcal{S}_h(\boldsymbol{u}') = \text{argmin}\mathcal{J}(\cdot, \boldsymbol{u}') \\ \boldsymbol{u}' = \mathcal{S}_{tv}(\boldsymbol{f}') = \text{argmin}\mathcal{J}(\boldsymbol{f}', \cdot). \end{cases}$$

Thus we obtain $\boldsymbol{u}' = \mathcal{S}_{tv}(\mathcal{S}_h(\boldsymbol{u}')) = \mathcal{T}(\boldsymbol{u}')$ and $\boldsymbol{u}'$ is a fixed point of $\mathcal{T}$. The result follows. ∎

Theorem 2.1. *Suppose $Null\ (\boldsymbol{XH}) \bigcap Null\ (\boldsymbol{L}) = \emptyset$. For any initial guess $\boldsymbol{u}^{(0)} \in \mathbb{R}^{n^2}$, suppose $\{\boldsymbol{u}^{(i)}\}$ is generated by (2.7), $\boldsymbol{u}^{(i)}$ converges to a stationary point of $\mathcal{J}$.*

We remark that the objective function in (2.5) is convex. The alternating minimization algorithm can determine a stationary point of the objective function.

3 Experimental Results

In this section, numerical results are presented to demonstrate the performance of our proposed algorithm for image restoration involving impulse noise and/or Gaussian noise. The results are compared with those obtained by two-phase method proposed in [5]. For simplicity, we call it "Cai-TP" method. In the Cai-TP method, a modified Mumford-Shah regularization scheme is used. There are three parameters (α, β and ϵ) in the Cai-TP method. We remark that there are two regularization parameters in the proposed algorithm. In order to reduce the complexity of searching the optimal parameters for image restoration. In all the tests, we set α_2 to be 1. On the other hand, we determine α_1 such that

$$\frac{\|\boldsymbol{u}(\alpha_1) - \boldsymbol{u}\|_2}{\|\boldsymbol{u}\|_2}$$

is the smallest among all tested values of α_1. For the Cai-TP method, we report the restoration results using the optimal parameters given in [5].

Peak signal to noise ratio (PSNR) is used to measure the quality of the restoration results. It is defined as follows:

$$\text{PSNR} = 20 \log_{10} \left(\frac{\|255\|_2}{\frac{1}{n}\|\widetilde{\boldsymbol{f}} - \boldsymbol{f}\|_2} \right),$$

where $\boldsymbol{f}$ and $\widetilde{\boldsymbol{f}}$ are the original image and the restored image respectively. The stopping criterion of both methods is that the relative difference between the successive iterate of the restored image should satisfy the following inequality:

$$\frac{||\boldsymbol{f}^{(i+1)} - \boldsymbol{f}^{(i)}||_2}{||\boldsymbol{f}^{(i+1)}||_2} < \delta.$$

δ is set to be 5×10^{-4} in the Cai-TP method and the proposed minimization method. Similar to the Cai-TP method, we apply a detection procedure to identify noisy pixels and then use the proposed minimization scheme and the alternating minimization algorithm to restore images. However, for random-valued noise removal problems, both detection and minimization procedures are repeated several times so that a high quality of restored image can be obtained.

We also note in degradation of random-valued noise that many outliers still remain after the median filtering. An additional preprocessing such as ℓ_1 norm smoothing disposal [5] is introduced in this case before the restoration step. It has been demonstrated in [18] that the use of ℓ_1 norm is less sensitive to outliers.

3.1 Experiment 1

In the first four experiments, we test that the observed image is degraded as follows:

$$\boldsymbol{g} = \mathbb{N}_{\text{imp}} \left(\boldsymbol{H}\boldsymbol{f} + \boldsymbol{n} \right).$$

In the first experiment, we restore images corrupted by salt-and-pepper noise only. The "Lena" image is blurred by an out of focus blur with radius 3, and then is corrupted by different levels

of salt-and-pepper noise. Figures 1(a)–(d) are the blurred and noisy images. Figures 1(e)–(h) are the corresponding restored images by the proposed method. Figures 2(a)–(d) are the other blurred and noisy images. They are blurred by an out of focus blur with radius 3, and then are corrupted by salt-and-pepper noise of level $s = 70\%$. Figures 2(e)–(h) are the corresponding restored images by the proposed method. In Table 1, we show the summary results of the proposed method and the Cai-TP method. We report the PSNRs and the computational timed required by the two methods. The PSNRs of the restored images by two methods are about the same. However, the proposed method is more efficient than the Cai-TP method.

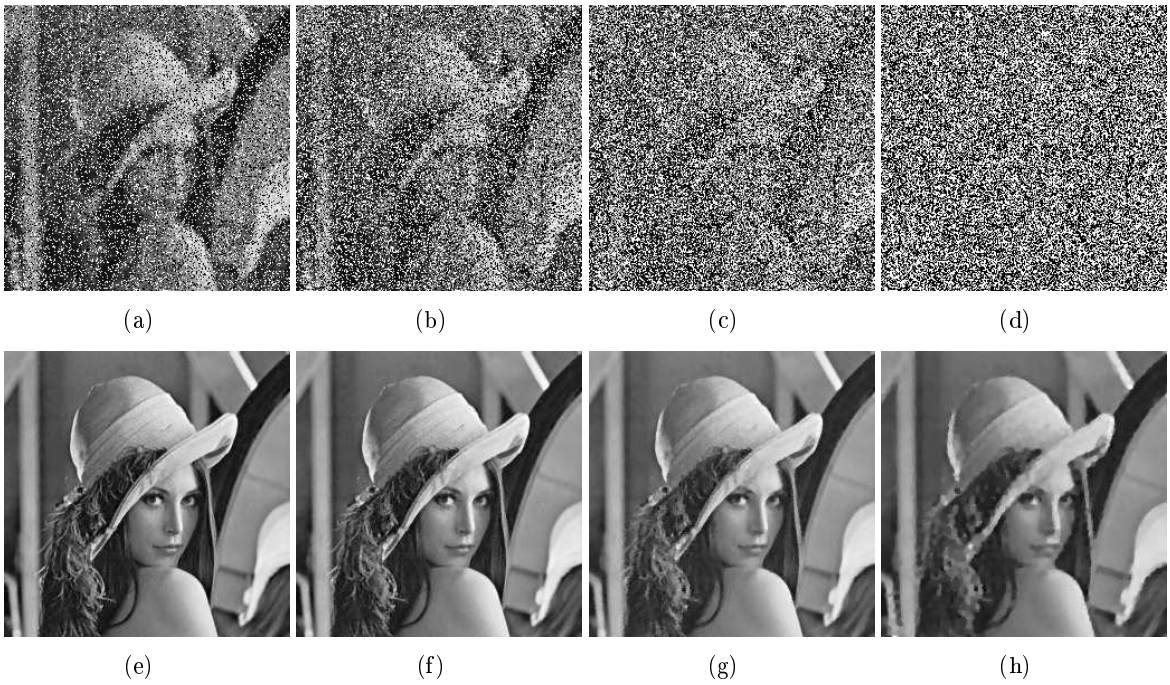


Figure 1: The blurred and noisy "Lena" images by using an out of focus kernel with radius 3 and corrupted by salt-and-pepper noise with different noise levels (the upper part); (a) $s = 30\%$; (b) $s = 50\%$; (c) $s = 70\%$; (d) $s = 90\%$. The corresponding restored images (the lower part) by the proposed method.

3.2 Experiment 2

In the second experiment, we restore images corrupted by random-valued noise only. The "Lena" image is blurred by an out of focus blur with radius 3, and then is corrupted by different levels of random-valued noise. Figures 3(a)–(d) are the blurred and noisy images. Figures 3(e)–(h) are the corresponding restored images by the proposed method. Figures 4(a)–(d) are the other blurred and noisy images. They are blurred by an out of focus blur with radius 3, and then are corrupted by random-valued noise of level $r = 40\%$. Figures 4(e)–(h) are the corresponding restored images

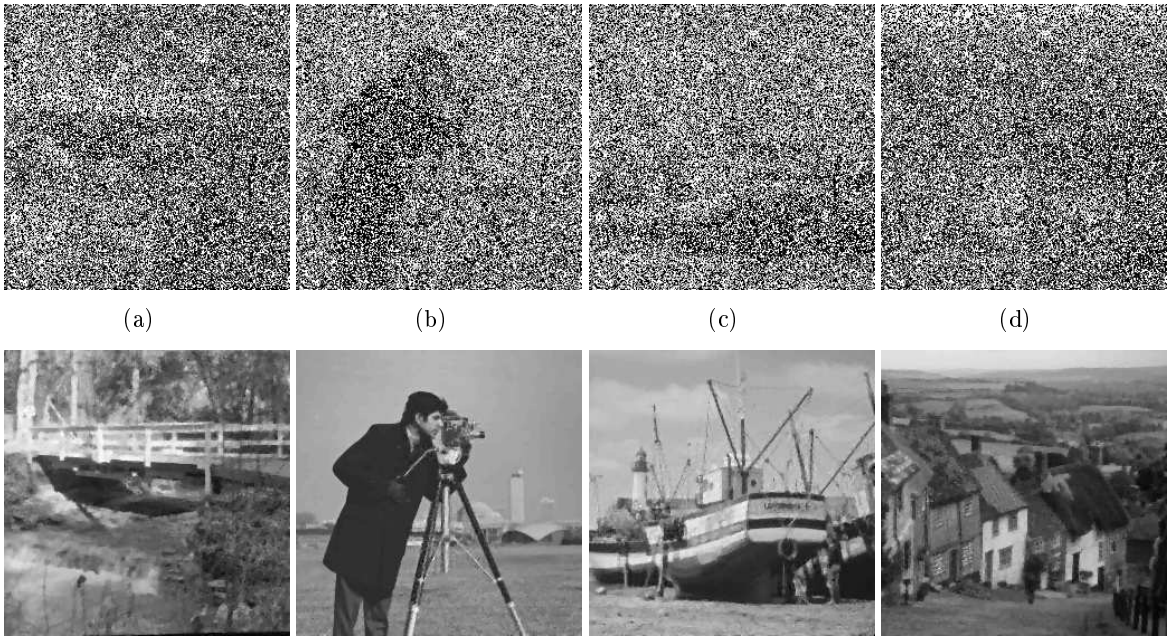


Figure 2: The blurred and noisy (a) "Bridge", (b) "Cameraman", (c) "Boat" and (d) "Goldhill" images by using an out of focus kernel with radius 3 and corrupted by salt-and-pepper noise with noise level $s = 70\%$ (the upper part). The corresponding restored images (the lower part) by the proposed method.

by the proposed method. In Table 2, we show the summary results of the proposed method and the Cai-TP method. We report the PSNRs and the computational times required by the two methods. In these tests, both PSNRs of the restored images and the computational times required by the proposed method are better those by the Cai-TP method.

3.3 Experiment 3

In the third experiment, we restore images corrupted by Gaussian noise plus impulse noise. The "Lena" image is blurred by an out of focus blur with radius 3, and then is corrupted by Gaussian noise of mean zero and standard deviation ($\sigma = 5$), and then by different levels of salt-and-pepper noise (or random-valued noise). Figures 5(a)–(d) are the blurred and noisy images. Figures 5(e)–(h) are the corresponding restored images by the proposed method. Figures 6(a)–(d) are the other blurred and noisy images. Figures 6(e)–(h) are the corresponding restored images by the proposed method. In Tables 3 and 4, we show the summary results of the proposed method and the Cai-TP method. We report the PSNRs and the computational times required by the two methods. In these tests, both PSNRs of the restored images and the computational times required by the proposed method are again better those by the Cai-TP method.

Image	Noise Level s	Proposed method		Cai-TP method	
		PSNR (dB)	CPU time (second)	PSNR (dB)	CPU time (second)
"Lena"	30%	36.7	52.0	35.9	465.5
	50%	33.2	58.5	32.7	432.0
	70%	30.3	61.0	30.1	602.9
	90%	26.6	71.5	26.7	729.2
"Bridge"	70%	26.1	61.0	26.2	585.4
"Cameraman"		27.2	65.0	26.7	621.3
"Boat"		27.2	52.1	26.7	601.5
"Goldhill"		28.6	48.7	28.4	592.0

Table 1: Summary results for the proposed method ($\alpha_1 = 0.0001$) and the Cai-TP method. The blurring kernel is an out-of-focus with radius 3 and the image is corrupted by salt-and-pepper noise only. The parameters used in Cai-TP method for "Lena" image are: $s = 30\%, [\alpha = 0.0002, \beta = 0.0002, \epsilon = 0.001]$; $s = 50\%, [\alpha = 0.0002, \beta = 0.0002, \epsilon = 0.0005]$; $s = 70\%, [\alpha = 0.0005, \beta = 0.0005, \epsilon = 0.0002]$ and $s = 90\%, [\alpha = 0.001, \beta = 0.001, \epsilon = 0.0001]$ respectively. The parameters used for other pictures are the same: $[\alpha = 0.0005, \beta = 0.0005, \epsilon = 0.0002]$.

Image	Noise Level r	Proposed method			Cai-TP method	
		α_1	PSNR (dB)	CPU time (second)	PSNR (dB)	CPU time (second)
"Lenna"	10%	0.0007	39.4	74.6	38.7	599.3
	25%	0.0009	35.1	74.3	34.4	763.4
	40%	0.0010	32.1	81.4	31.2	643.3
	55%	0.0030	29.0	78.0	27.8	779.2
"Bridge"	40%	0.0010	28.2	100.6	27.3	573.6
"Cameraman"		0.0010	27.9	91.7	27.8	532.9
"Boat"		0.0010	29.3	68.5	28.2	523.5
"Goldhill"		0.0010	30.8	65.2	29.5	487.3

Table 2: Summary results for the proposed method and the Cai-TP method. The blurring kernel is an out-of-focus with radius 3 and the image is corrupted by random-valued noise only. The parameters used in Cai-TP method for "Lena" image are: $r = 10\%, [\alpha = 0.0005, \beta = 0.0005, \epsilon = 0.001]$; $r = 25\%, [\alpha = 0.001, \beta = 0.001, \epsilon = 0.0005]$; $r = 40\%, [\alpha = 0.002, \beta = 0.002, \epsilon = 0.0005]$ and $r = 55\%; [\alpha = 0.005, \beta = 0.005, \epsilon = 0.0001]$ respectively. The parameters for other pictures are the same: $[\alpha = 0.002, \beta = 0.002, \epsilon = 0.0005]$.

Image	Noise Level s	Proposed method			Cai-TP method	
		α_1	PSNR (dB)	CPU time (second)	PSNR (dB)	CPU time (second)
"Lena"	30%	0.7	27.5	18.9	27.2	512.0
	50%	0.5	27.2	19.5	26.9	642.0
	70%	0.4	26.6	20.4	26.4	555.8
	90%	0.2	24.9	32.5	24.7	771.4

Table 3: Summary results for the proposed method and the Cai-TP method. The blurring kernel is an out-of-focus with radius 3 and the image is corrupted by Gaussian noise and then salt-and-pepper noise. The parameters used in Cai-TP method for different noises are the same $[\alpha = 0.05, \beta = 0.05, \epsilon = 0.0002]$.

Image	Noise Level r	Proposed method			Cai-TP method	
		α_1	PSNR (dB)	CPU time (second)	PSNR (dB)	CPU time (second)
"Lena"	10%	0.6	27.6	48.5	27.2	569.7
	25%	0.7	27.3	54.0	27.0	873.2
	40%	0.8	27.0	56.4	26.7	736.5
	55%	0.6	25.8	64.8	25.6	912.3

Table 4: Summary results for the proposed method and the Cai-TP method. The blurring kernel is an out-of-focus with radius 3 and the image is corrupted by Gaussian noise and then random-valued noise. The parameters used in Cai-TP method for different noises are: $r = 10\%, [\alpha = 0.05, \beta = 0.05, \epsilon = 0.0002]$; $r = 25\%, [\alpha = 0.01, \beta = 0.005, \epsilon = 0.0002]$, $[\alpha = 0.01, \beta = 0.005, \epsilon = 0.0002]$ and $[\alpha = 0.01, \beta = 0.01, \epsilon = 0.0001]$ respectively.

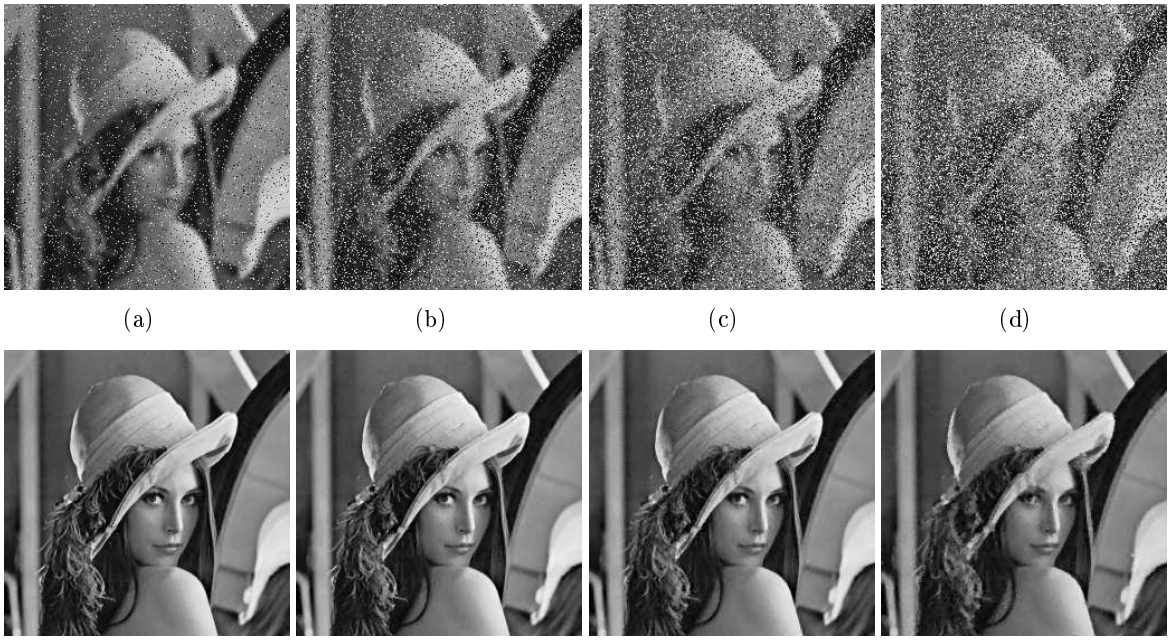


Figure 3: The blurred and noisy "Lena" images by using an out of focus kernel with radius 3 and corrupted by random-valued noise with different noise levels (the upper part); (a) $r = 10\%$; (b) $r = 25\%$; (c) $r = 40\%$; (d) $r = 55\%$. The corresponding restored images (the lower part) by the proposed method.

3.4 Experiment 4

In the fourth experiment, we restore images blurred by a Gaussian kernel (generated by MATLAB command fspecial('Gaussian', [7 7], 1)) and a motion kernel (generated by MATLAB command fspecial('motion', 9, 1)). Figures 7(a) and (b) are such blurred images. When these two blurred images are corrupted by salt-and-pepper noise only of level $s = 70$ (see Figures 7(c) and (d)), Figures 7(g) and (h) are the corresponding restored images by the proposed method. When the blurred images are corrupted by random-valued noise only of level $r = 40$ (see Figures 7(e) and (f)), Figures 7(i) and (j) are the corresponding restored images by the proposed method. The PSNRs of the restored images in (g), (h), (i) and (j) by the proposed method are 31.1 dB, 30.4 dB, 31.3 dB and 31.4 dB respectively. The computational time required for the restored images in (g), (h), (i) and (j) by the proposed method are 89.8 seconds, 98.1 seconds, 101.8 seconds and 104.3 seconds respectively. In contrast, the PSNRs of the restored images by the Cai-TP method are 30.6 dB, 28.0 dB, 31.5 dB and 29.8 dB respectively. Also the computational time required for the restored images by the Cai-TP method are 603.5 seconds, 613.5 seconds, 839.2 seconds and 551.7 seconds respectively. These results show that the proposed method can restore corrupted image quite well in an efficient manner.

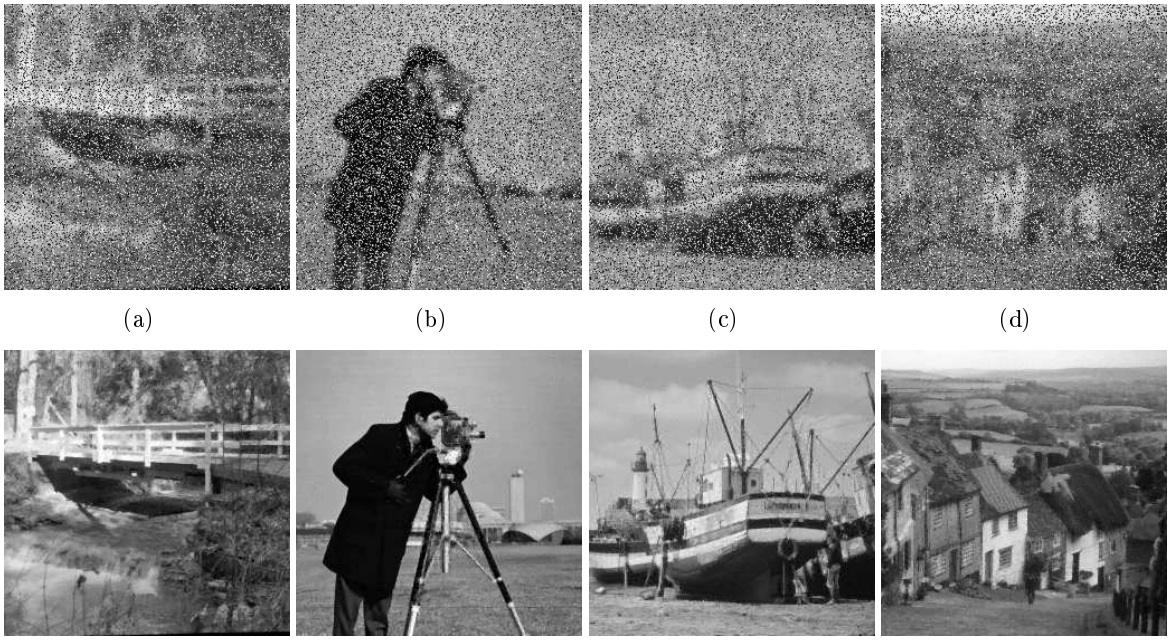


Figure 4: The blurred and noisy (a) "Bridge", (b) "Cameraman", (c) "Boat" and (d) "Goldhill" images by using an out of focus kernel with radius 3 and corrupted by random-valued noise with noise level $r = 40\%$ (the upper part). The corresponding restored images (the lower part) by the proposed method.

3.5 Experiment 5

In this subsection, we test images degraded as follows:

$$\boldsymbol{g} = \mathbb{N}_{\text{imp}}\left(\boldsymbol{H}\boldsymbol{f}\right) + \boldsymbol{n}.$$

In the experiment, we restore images corrupted by impulse noise and then by Gaussian noise. The "Lena" image is blurred by an out of focus blur with radius 3, and then is corrupted by different levels of salt-and-pepper noise plus Gaussian noise, see Figures 8(a)–(d), or of random-valued noise plus Gaussian noise, see Figures 9(a)–(d). Figures 8(e)–(h) and Figures 9(e)–(h) are the corresponding restored images by the proposed method. In Tables 5 and 6, we show the summary results of the proposed method. We report the PSNRs and the computational times required by the proposed method. The proposed method can restore image quite well. We remark that this image degradation model is not considered and studied in [5]. It is interesting to note that the PSNRs and the computational times of the proposed method are about the same as those images degraded by $\boldsymbol{g} = \mathbb{N}_{\text{imp}}\left(\boldsymbol{H}\boldsymbol{f} + \boldsymbol{n}\right)$, see Tables 3 and 5, and Tables 4 and 6.

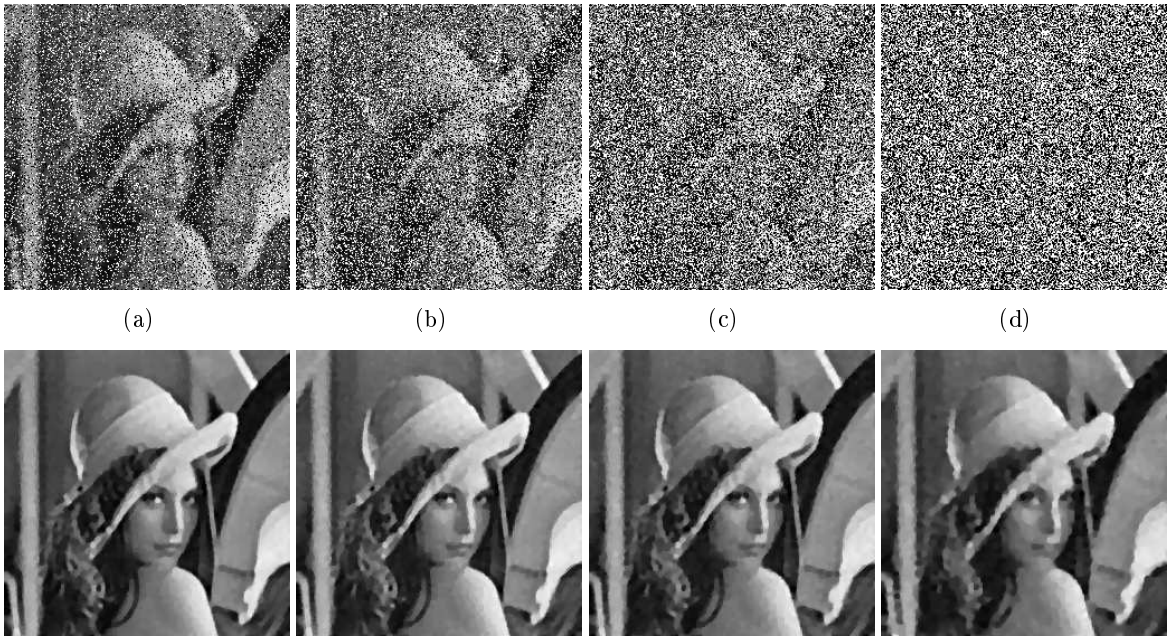
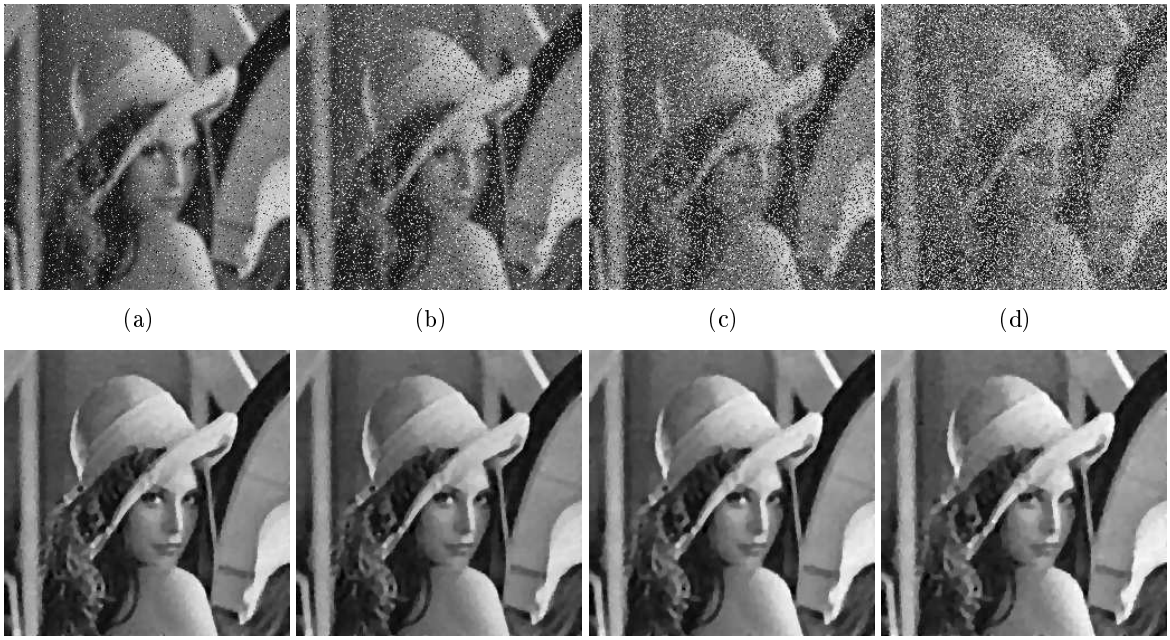


Figure 5: The blurred and noisy "Lena" images by using an out of focus kernel with radius 3 and corrupted by Gaussian noise of standard deviation 5 and then salt-and-pepper noise with different noise levels (the upper part); (a) $s = 30\%$; (b) $s = 50\%$; (c) $s = 70\%$; (d) $s = 90\%$. The corresponding restored images (the lower part) by the proposed method.

4 Concluding Remarks

In this paper, we have proposed fast restoration methods for blurred images corrupted by impulse noise or mixed impulse plus Gaussian noises. In the proposed method, we use the modified total variation minimization scheme to regularize the deblurred image and fill in suitable values for noisy image pixels where these are detected by median-type filters. An alternating minimization algorithm is employed to solve the proposed total variation minimization problem. We have shown the convergence of the alternating minimization algorithm and demonstrate that the algorithm is very efficient. Our experimental results have also shown that the quality of restored images by the proposed method is competitive with those restored by the existing variational image restoration methods.

Acknowledgement: The authors would like to thank Dr. Jian-Feng Cai for his kind offer of the source codes for the Cai-TP method compared in the paper.



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

(e) (f) (g) (h)

Figure 6: The blurred and noisy "Lena" images by using an out of focus kernel with radius 3 and corrupted by Gaussian noise of standard deviation 5 and then random-valued noise with different noise levels (the upper part); (a) $r = 10\%$; (b) $r = 25\%$; (c) $r = 40\%$; (d) $r = 55\%$. The corresponding restored images (the lower part) by the proposed method.

References

[1] L. Bar, A. Brook, N. Schen and N. Kiryati, Deblurring of color images corrupted by salt-and-pepper noise, *IEEE Trans. Image Process.*, Vol. 16 (2007), pp. 1101–1111.

[2] L. Bar, N. Schen and N. Kiryati, Image deblurring in the presence of salt-and-pepper noise, *Proceeding of 5th international conference on scale space and PDE methods in computer vision, LNCS*, Vol. 3439 (2005), pp. 107–118.

[3] L. Bar, N. Schen and N. Kiryati, Image deblurring in the presence of impulsive noise, *Int. J. Computer Vision*, Vol. 70 (2006), pp. 279–298.

[4] D. Bertsekas, A. Nedic and E. Ozdaglar, Convex analysis and optimization, *Athena Scientific*, 2003.

[5] J. Cai, R. Chan and M. Nikolova, Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise, *Inv. Prob. Imag.*, Vol. 2 (2008), pp. 187-204.

Image	Noise Level s	α_1	PSNR (dB)	CPU time (second)
"Lena"	30%	0.7	27.5	35.1
	50%	0.5	27.2	39.3
	70%	0.4	26.6	45.3
	90%	0.2	24.8	62.0

Table 5: Summary results for the proposed method. The blurring kernel is an out-of-focus with radius 3 and the image is corrupted by salt-and-pepper noise and then Gaussian noise.

Image	Noise Level r	α_1	PSNR (dB)	CPU time (second)
"Lena"	10%	0.6	27.6	40.5
	25%	0.7	27.3	44.8
	40%	0.8	27.0	44.5
	55%	0.6	25.9	50.7

Table 6: Summary results for the proposed method. The blurring kernel is an out-of-focus with radius 3 and the image is corrupted by random-valued noise and then Gaussian noise.

[6] A. Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vision.* , Vol. 20 (2004), pp. 89–97.

[7] A. Chambolle, Total variation minimization and a class of binary MRF models, *Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, Springer Berlin*, Vol. 3757 (2005), pp. 136–152.

[8] T. Chan and K. Chen, An optimization based total variation image denoising, *SIAM J. Multi. Model. Simul.*, Vol. 5 (2006), pp. 615–645.

[9] R. Chan, C. Ho and M. Nikolova, Salt-and-pepper noise removal by median-type noise detector and edge-preserving regularization, *IEEE Trans. Image Process.*, Vol. 14 (2005), pp. 1479–1485.

[10] R. Chan, C. Hu and M. Nikolova, An iterative procedure for removing random-valued impulse noise, *IEEE Sig. Process. Lett.*, Vol. 11 (2004), pp. 921–924.

[11] T. Chen and H. Wu, Space variant median filters for the restoration of impulse noise currupted images, *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, Vol. 48 (2001), pp. 784–789.

[12] P. Combettes and V. Wajs, Signal recovery by proximal forward-backward splitting, *SIAM J. Multi. Model. Simul.*, Vol. 4 (2005), pp. 1168–1200.

[13] H. Eng and K. Ma, Noise adaptive soft-switching median filter, *IEEE Trans. Image Process.*, Vol. 10 (2001), pp. 242–251.

[14] H. Hwang and R. Haddad, Adaptive median filters: new algorithms and results, *IEEE Trans. Image Process.*, Vol. 4 (1995), pp. 499–502.

[15] M. Ng, Iterative methods for toeplitz systems, *Oxford University Press*, 2004.

[16] M. Ng, R. Chan and W. Tang, A fast algorithm for deblurring models with Neumann boundary conditions. *SIAM J. Sci. Comput.*, Vol. 21 (2000), pp. 851–866.

[17] M. Ng, L. Qi, Y. Yang and Y. Huang, On semismooth Newton's methods for total variation minimization, *J. Math. Imaging Vision.*, Vol. 27 (2007), pp. 265–276.

[18] M. Nikolova, Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers, *SIAM J Numer Ana.*, Vol. 40 (2002), pp. 965–994.

[19] M. Nikolova, A variational approach to remove outliers and impulse noise, *J. Math. Imaging Vis.*, Vol. 20 (2004), pp. 99–120.

[20] Z. Opial, Weak convergence of the sequence of successive approximations for nonexpansive mappings, *Bullet. Amer. Math. Soc.*, Vol. 73 (1967), pp. 591–597.

[21] G. Pok, J. Liu and A. Nair, Selective removal of impulse noise based on homogeneity level information, *IEEE Trans. Image Process.*, Vol. 12 (2003), pp. 85–92.

[22] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D*, Vol. 60 (1992), pp. 259–268.

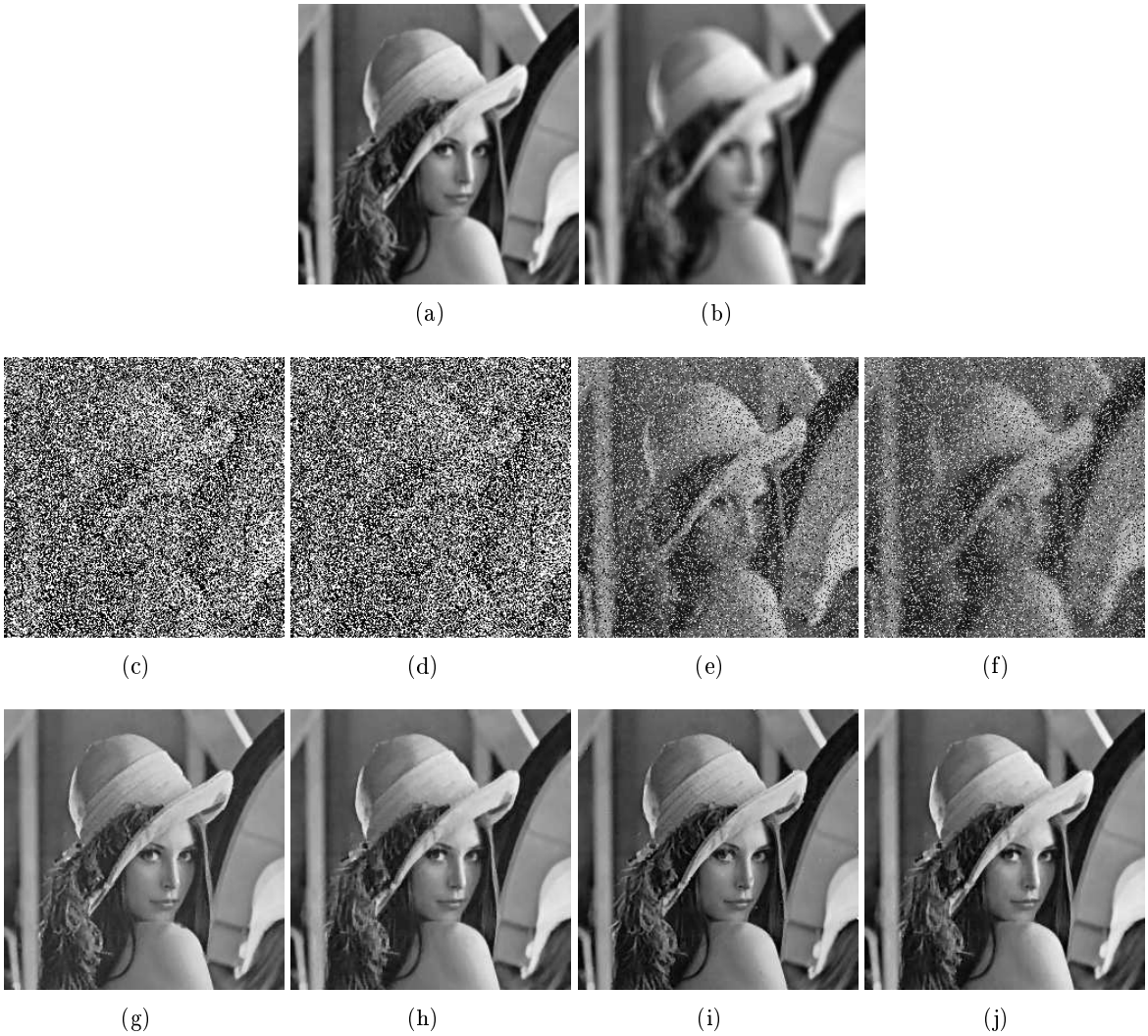


Figure 7: "Lena" image blurred with different kernels, and then corrupted by impulse noise only. (a) the blurred image with a Gaussian kernel (generated by MATLAB command fspecial('Gaussian', [7 7], 1)); (b) the blurred image with a motion kernel (generated by MATLAB command fspecial('motion', 9, 1)); (c) the blurred image in (a) corrupted by salt-and-pepper noise only with $s = 70\%$; (d) the blurred image in (b) corrupted by salt-and-pepper noise only with $s = 70\%$; (e) the blurred image in (a) corrupted by random-valued noise only with $r = 40\%$; (f) the blurred image in (b) corrupted by random-valued noise only with $r = 40\%$; (g)–(j) are the corresponding restored images for (c)–(f).

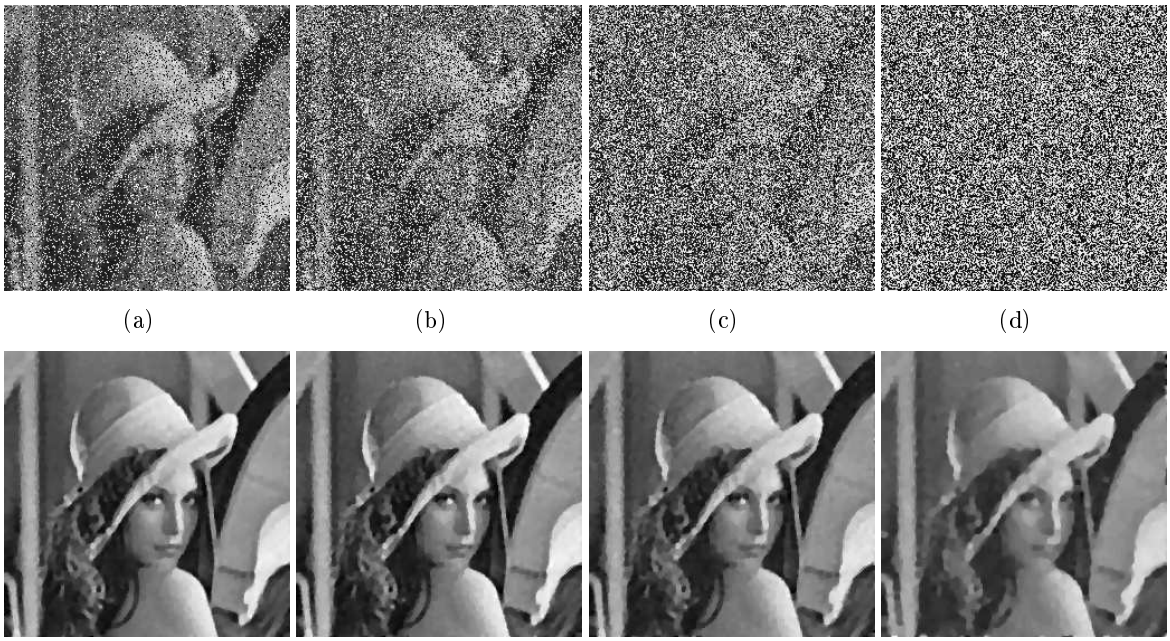
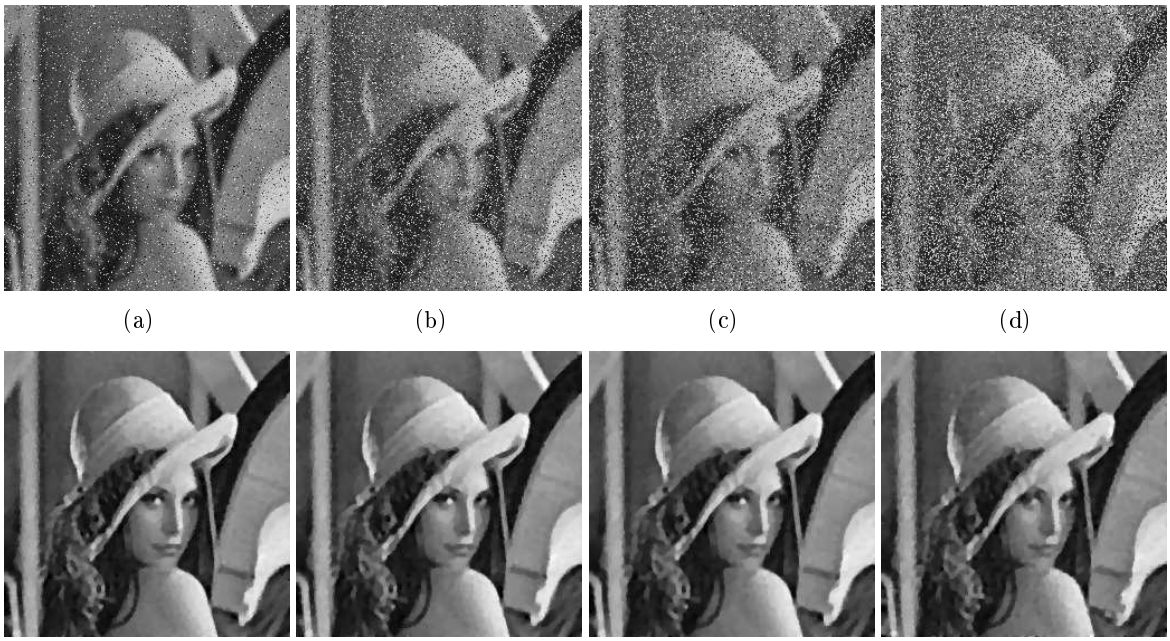


Figure 8: The blurred "Lenna" images by using an out of focus kernel with radius 3 and corrupted by salt-and-pepper noise with different noise levels and then Gaussian noise of standard deviation 5 (the upper part); (a) $r = 30\%$; (b) $r = 50\%$; (c) $r = 70\%$; (d) $r = 90\%$. The corresponding restored images (the lower part) by the proposed method.



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

(e) (f) (g) (h)

Figure 9: The blurred "Lenna" images by using an out of focus kernel with radius 3 and corrupted by random-valued noise with different noise levels and then Gaussian noise of standard deviation 5 (the upper part); (a) $r = 10\%$; (b) $r = 25\%$; (c) $r = 40\%$; (d) $r = 55\%$. The corresponding restored images (the lower part) by the proposed method.