Suppression of Missing Data Artifacts for Deblurring Images Corrupted by Random Valued Noise

Nam-Yong Lee[*]

Department of Applied Mathematics

Institute of Basic Sciences, Inje University

Gimhae, Gyeongnam 621-749, Korea

[*] nylee@inje.ac.kr

Abstract

For deblurring images corrupted by random valued noise, two-phase methods first select likely-to-be *reliables* (data that are not corrupted by random valued noise) and then deblur images only with selected data. The selective use of data in two-phase methods, however, often causes *missing data artifacts.* In this paper, to suppress these missing data artifacts, we propose a blurring model based reliable-selection technique to select sufficiently many reliables so that all of to-be-recovered pixel values can contribute to selected data, while excluding random value noised data accurately. We also propose a normalization technique to compensate for non-uniform rates in recovering pixel values. Simulation studies show that proposed techniques effectively suppress missing data artifacts and, as a result, improve the performance of two-phase methods.

1 Introduction

The image deblurring is an ill-conditioned inverse problem, and the presence of random valued noise makes the problem more difficult. The random value noise corrupts some observed data with random values and leaves others unaffected. This type of noise is often generated by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or erroneous transmission [1].

For the simplicity of the presentation, we use the term *outliers* to refer data that are corrupted by random valued noise, and *reliables* to refer data that are not corrupted by random valued noise.

Many methods have been proposed for deblurring images corrupted by random valued noise. Depending on what kind of data are used in deblurring, those methods can be categorized into following three groups: (a) 'simultaneous outlier-smoothing and deblurring' (data as observed) [2, 3, 4], (b) 'outlier-smoothing followed by deblurring' (smoothed data), and (c) 'reliable-selection followed by deblurring' (selected data as reliables) [5, 6, 7, 8]. In particular, methods in the 'reliable-selection followed by deblurring' group are often called *two-phase* methods; they select likely-to-be reliables in the first phase, and deblur images only with selected data in the second phase.

It is well-known that methods in the 'simultaneous outlier-smoothing and deblurring' group often produce very poor results, even in the presence of very small amount of outliers [5]. Preprocessing outlier-smoothings might reduce such artifacts. Improvements made by outlier-smoothings are, however, often diminished out by errors made by outlier-smoothings themselves. For details, see [5]

Recent research works [5, 6, 7, 8] show superior performance of two-phase methods over 'simultaneous outlier-smoothing and deblurring' and 'outlier-smoothing followed by deblurring' methods. Two-phase methods can use various deblurring algorithms in the second phase. For instance, the Mumford-Shah regularization functional is used in [5] to achieve edge-preserving deblurring. In [6, 7], the L^1-norm of wavelet or framelet transforms is used to utilize sparse representations of images in wavelet or framelet transform domains. The method in [8] uses iterative two-phase approaches to improve the accuracy in reliable-selection.

Two-phase methods, however, often produce so-called *missing data artifacts*; the lack of data (caused by not filling gaps made by unselected data) makes it difficult to recover some pixel values. Missing data artifacts can be suppressed by various *inpainting* algorithms [9, 10, 11]. In fact, two-phase methods in [5, 6, 7, 8] can also do inpainting. Inpainting algorithms, however, do not suppress missing data artifacts effectively in case when the lack of data is serious. We will explain such phenomenon in Section 3.1 of this paper.

The objective of this paper is to improve the performance of two-phase methods, by effectively suppressing missing data artifacts. For this purpose, we propose *blurring model based reliable-selection* and *normalization* techniques.

Previous reliable-selection techniques (e.g., median-type approaches [12] and iterative projection comparison based reliable-selection [8]) tend to select reliables more frequently from smooth region than near-edge region. Such phenomenon often leads to the situation that no data are selected from a wide region. This often causes missing data artifacts. To avoid this kind of difficulty, the proposed blurring model based reliable-selection technique selects sufficiently many reliables so that all of to-be-recovered pixel values contribute to selected data, while excluding outliers accurately.

The proposed normalization technique is designed to compensate for the non-uniformity in deblurring images; the selective use of data in two-phase methods causes some pixel values to be recovered less accurately or more slowly than others in iterative deblurring. Such phenomenon often occurs on pixel values that give less contribution to selected data. To suppress this type of missing data artifacts, we suggest to use a weighted inner product which treats pixel values accordingly to the degree of their contribution to selected data. In this paper, the resulting algorithm derived from the use of such weighted inner product will be called the normalization technique.

In this paper, we will explain the application of proposed techniques to a total variation (TV) [13] based two-phase method. Extensions to other two-phase methods in [5, 6, 7, 8] will be straightforward.

The outline of this paper is as follows. In Section 2, definitions, notations, and previous works are reviewed. In Section 3, proposed techniques are explained. In Section 4, simulation studies are conducted to test the performance of proposed techniques in Gaussian and diagonal deblurrings. Conclusions and discussions are presented in Section 5.

2 Definitions, Notations, and Backgrounds

2.1 Observation model

In this paper, we model the problem as

$$\mathbf{g} = \mathcal{R}(\mathbf{H}\mathbf{f} + \mathbf{n}), \tag{1}$$

where $\mathbf{H}$ is the linear transform that represents the blurring process of the problem, $\mathcal{R}$ is the nonlinear transform that represents the random value noise process, $\mathbf{g}$ is the observed image on a rectangular pixel set

$$\mathbb{D} = \{(b_1, b_2) \mid 0 \leq b_1 < N_1, 0 \leq b_2 < N_2\}, \tag{2}$$

$\mathbf{f}$ is the true image to be recovered, and $\mathbf{n}$ is the mean zero Gaussian noise. We use the notation $\bar{\mathbb{D}}$ to denote the domain where $\mathbf{f}$ is defined on.

The random value noise model in (1) means that $\mathcal{R}(x) =$ one of $\{0, 1, \ldots, 255\}$ with the probability r, while $\mathcal{R}(x) = x$ with the probability $1 - r$, for some $0 < r < 1$. For future use, we consider the 'salt-and-pepper' noise model. The salt-and-pepper noise with the probability s gives $\mathcal{R}(x) = x$ with the probability $1-s$, while $\mathcal{R}(x) = 0$ or 255 with the probability s. In explaining random value and salt-and-pepper noise models, we assumed that images are gray scaled, so that their pixel values range over $\{0, 1, \ldots, 255\}$.

To denote the pixel, we use the single index in some cases or double indices with the parenthesis in other cases. For instance, $f_v = f_{(v_1,v_2)}$ is the value of the image $\mathbf{f}$ at the pixel $v = (v_1, v_2)$. With the single pixel index, we treat images $\mathbf{g}$ and $\mathbf{f}$ as one dimensional vectors. Therefore, $\mathbf{H}$ in (1) is a $|\mathbb{D}| \times |\bar{\mathbb{D}}|$ matrix. We also use following convention throughout this paper: boldface alphabets for images or matrices and normal alphabets for corresponding pixel values or entries, as in $\mathbf{f} = (f_v)$, $\mathbf{g} = (g_b)$, and $\mathbf{H} = (H_{b,v})$.

We use the notation $L^2(\mathcal{A})$ to denote the image space defined on a pixel set $\mathcal{A}$. For instance, $\mathbf{f} \in L^2(\bar{\mathbb{D}})$ and $\mathbf{g} \in L^2(\mathbb{D})$. Here the notation L^2 indicates that the usual dot product of two images (sums of pixel-by-pixel multiplications) is used as the inner product.

We assume that the blurring transform represented by $\mathbf{H}$ is a truncated convolution by a known finitely supported point spread function (PSF) $\mathbf{k} = (k_v)$. Thus, for any image $\mathbf{p} \in L^2(\bar{\mathbb{D}})$,

$$(\mathbf{H}\mathbf{p})_b = \sum_{v \in \bar{\mathbb{D}}} H_{b,v} p_v = \sum_{v \in (b - \mathrm{S}_{\mathbf{k}})} k_{b-v} p_v. \tag{3}$$

Here $\mathrm{S}_{\mathbf{k}}$, the *support* of $\mathbf{k}$, is $\{v \mid k_v > 0\}$. Throughout this paper, we assume that the PSF $\mathbf{k}$ is nonnegative, its components have sum 1, and the point $(0,0) \in \mathrm{S}_{\mathbf{k}}$. In (3), the summation is well-defined only if $b - \mathrm{S}_{\mathbf{k}} \subseteq \bar{\mathbb{D}}$. Thus, $\mathbb{D} \subseteq \bar{\mathbb{D}}$.

2.2 Probabilistic interpretation

Assumptions imposed on the PSF $\mathbf{k}$ (non-negativity and total sum 1) imply that

$$H_{b,v} \geq 0, \qquad \sum_{v \in \bar{\mathbb{D}}} H_{b,v} = 1, \quad \text{and} \quad \sum_{b \in \mathbb{D}} H_{b,v} \leq 1 \tag{4}$$

for all $b \in \mathbb{D}$ and $v \in \bar{\mathbb{D}}$. Thus, $H_{b,v}$ can be regarded as a conditional probability:

$$H_{b,v} = \text{Prob}(\text{observed at } b \text{ through } \mathbf{H} \mid \text{supposed to be at } v). \tag{5}$$

This interpretation is based on the following observation: The pixel value f_v (supposed to be at v) contributes to $(\mathbf{Hf})_b = \sum_{v \in \bar{\mathbb{D}}} H_{b,v} f_v$ (observed at b through $\mathbf{H}$) by $H_{b,v} f_v$.

This kind of probabilistic interpretation is not new. In emission tomography, which has similar obstacles and models as image deblurring has, entries $H_{b,v}$ of the matrix $\mathbf{H}$ are regarded as conditional probabilities as in (5). In emission tomography, $\sum_{b \in \mathbb{D}} H_{b,v}$ are called *normalization coefficients* [14]. For a fixed pixel v, $\sum_{b \in \mathbb{D}} H_{b,v}$ is the probability for the photon emitted from v to be detected by the tomograph modeled by the matrix $\mathbf{H}$. Notice that the same interpretation also holds for the image deblurring problem; $\sum_{b \in \mathbb{D}} H_{b,v}$ represents the *degree of contribution* of the pixel v to the observation on $\mathbb{D}$

through the blurring transform $\mathbf{H}$.

2.3 Two-phase methods

Let $\Lambda \subseteq \mathbb{D}$ be the set of selected pixels in the first phase, i.e.,

$$\Lambda = \{b \in \mathbb{D} \mid g_b \text{ is selected as reliable}\}. \tag{6}$$

Then, the problem to be solved in the second phase is

$$\tilde{\mathbf{g}} = \tilde{\mathbf{H}}\mathbf{f} + \tilde{\mathbf{n}}, \tag{7}$$

where

$$\tilde{\mathbf{g}} = \mathbf{g} \mid_\Lambda \quad \text{(restriction of } \mathbf{g} \text{ on } \Lambda), \tag{8}$$

and

$$\tilde{\mathbf{H}} = (H_{b,v})_{b\in\Lambda} \quad \text{(row-restricted matrix of } \mathbf{H} \text{ on } \Lambda), \tag{9}$$

Here note that the new noise term $\tilde{\mathbf{n}}$ in (7) is a mixed result of the original noise $\mathbf{n}$ in (1) and the error in selecting reliables in the first phase.

For the deblurring method in the second phase, we consider the following form of the TV-based method:

$$\min_{\mathbf{f}} \left(\|\tilde{\mathbf{g}} - \tilde{\mathbf{H}}\mathbf{f}\|^p_{L^p(\Lambda)} + \lambda \| |\mathscr{L}\mathbf{f}| \|_1 \right), \quad p = 1 \text{ or } 2, \tag{10}$$

where $\mathscr{L}\mathbf{f}$, defined by $(\mathscr{L}\mathbf{f})_v = (f_{v+(0,1)} - f_v, f_{v+(1,0)} - f_v)$, is used as a discretization of the smoothness quantity $\nabla\mathbf{f}$ and

$$|||\mathscr{L}\mathbf{f}|||_1 = \sum_{v\in\bar{\mathbb{D}}} \sqrt{|f_{v+(0,1)} - f_v|^2 + |f_{v+(1,0)} - f_v|^2}. \tag{11}$$

The *regularization term* $|||\mathscr{L}\mathbf{f}|||_1$ in (10) encourages the solution to be piecewise constant, and makes the resulting algorithm more robust to random valued noise. The *regularization parameter* $\lambda > 0$ balances the *data fidelity term*, $\|\tilde{\mathbf{g}} - \tilde{\mathbf{H}}\mathbf{f}\|^p_{L^p(\Lambda)}$, and the regularization term in the formulation. We use notations L1TV and L2TV to denote the TV-based regularization (10) corresponding to $p = 1$ and $p = 2$, respectively.

The minimization of (10) can be carried out by *split Bregman iteration*. The split Bregman iteration approximates the L^1 minimization by a series of L^2 minimizations. This iterative algorithm is outlined here. For details, see [6, 7].

2.3.1 L2TV

Starting from $\hat{\mathbf{f}}^0 = \mathbf{0}_{\bar{\mathbb{D}}}$ (all-zero image defined on $\bar{\mathbb{D}}$) and $\mathbf{d}^0 = \mathbf{c}^0 = \vec{\mathbf{0}}_{\bar{\mathbb{D}}}$ (two-dimensional all-zero image defined on $\bar{\mathbb{D}}$), the split Bregman iteration for (10) with $p = 2$ takes

$$\hat{\mathbf{f}}^{n+1} = \arg\min_{\mathbf{f}} \left(\|\tilde{\mathbf{g}} - \tilde{\mathbf{H}}\mathbf{f}\|^2_{L^2(\Lambda)} + \mu\|\mathscr{L}\mathbf{f} - \mathbf{d}^n + \mathbf{c}^n\|_2^2 \right), \tag{12}$$

where $\|\mathscr{L}\mathbf{f}-\mathbf{d}^n+\mathbf{c}^n\|_2^2=\sum_{v\in\bar{\mathbb{D}}}|(\mathscr{L}\mathbf{f}-\mathbf{d}^n+\mathbf{c}^n)_v|^2$ and

$$\begin{aligned}\mathbf{d}^{n+1} &= \mathcal{T}_{\lambda/\mu}(\mathscr{L}\hat{\mathbf{f}}^{n+1}+\mathbf{c}^n),\\ \mathbf{c}^{n+1} &= \mathbf{c}^n+\mathscr{L}\hat{\mathbf{f}}^{n+1}-\mathbf{d}^{n+1},\end{aligned} \tag{13}$$

where $\mu>0$ is a parameter of the algorithm, and $\mathcal{T}_{\lambda/\mu}$ is the soft-shrinkage operator defined by $(\mathcal{T}_\theta(\vec{\mathbf{x}}))_v=(t_\theta(x_v^1),t_\theta(x_v^2))$ for any two-dimensional image $\vec{\mathbf{x}}$ defined on $\bar{\mathbb{D}}$, (i.e., $\vec{\mathbf{x}}_v=(x_v^1,x_v^2)$), where $t_\theta(\xi)=\operatorname{sign}(\xi)\max(0,|\xi|-\theta)$.

The first step (12) is to find the solution of

$$\tilde{\mathbf{H}}^t\tilde{\mathbf{g}}+\mu\mathscr{L}^t(\mathbf{d}^n-\mathbf{c}^n)=\left(\tilde{\mathbf{H}}^t\tilde{\mathbf{H}}+\mu\mathscr{L}^t\mathscr{L}\right)\mathbf{f}. \tag{14}$$

It was noted in [7] that a very accurate solution of (14) is unnecessary at each split Bregman iteration. In our simulations, we used 5 conjugate gradient (CG) iterations for (14).

2.3.2 L1TV

For the case $p=1$, both data fidelity and regularization terms in (10) are based on L^1-norm. Thus the split Bregman iteration in (12) and (13) should be modified accordingly. Here we only state the result. For details, see [6, 7].

Starting from $\hat{\mathbf{f}}^0=\mathbf{0}_{\bar{\mathbb{D}}}$, $\mathbf{u}^0=\mathbf{w}^0=\mathbf{0}_\Lambda$, and $\mathbf{d}^0=\mathbf{c}^0=\vec{\mathbf{0}}_{\bar{\mathbb{D}}}$, the corresponding split Bregman iteration for the case $p=1$ takes

$$\hat{\mathbf{f}}^{n+1}=\arg\min_{\mathbf{f}}\left(\mu_1\|\tilde{\mathbf{g}}-\tilde{\mathbf{H}}\mathbf{f}+\mathbf{u}^n-\mathbf{w}^n\|_{L^2(\Lambda)}^2+\mu_2\|\mathscr{L}\mathbf{f}-\mathbf{d}^n+\mathbf{c}^n\|_2^2\right), \tag{15}$$

where

$$\begin{aligned}\mathbf{d}^{n+1} &= \mathcal{T}_{\lambda/\mu_2}(\mathscr{L}\hat{\mathbf{f}}^{n+1}+\mathbf{c}^n),\\ \mathbf{c}^{n+1} &= \mathbf{c}^n+\mathscr{L}\hat{\mathbf{f}}^{n+1}-\mathbf{d}^{n+1},\\ \mathbf{u}^{n+1} &= \mathcal{T}_{1/\mu_1}(\tilde{\mathbf{H}}\hat{\mathbf{f}}^{n+1}-\tilde{\mathbf{g}}+\mathbf{w}^n),\\ \mathbf{w}^{n+1} &= \mathbf{w}^n+\tilde{\mathbf{H}}\hat{\mathbf{f}}^{n+1}-\tilde{\mathbf{g}}-\mathbf{u}^{n+1},\end{aligned} \tag{16}$$

where $\mu_1, \mu_2 > 0$ are parameters of the algorithm.

The first step (15) is to find the solution of

$$\mu_1\tilde{\mathbf{H}}^t(\tilde{\mathbf{g}}+\mathbf{u}^n-\mathbf{w}^n)+\mu_2\mathscr{L}^t(\mathbf{d}^n-\mathbf{c}^n)=\left(\mu_1\tilde{\mathbf{H}}^t\tilde{\mathbf{H}}+\mu_2\mathscr{L}^t\mathscr{L}\right)\mathbf{f}. \tag{17}$$

Similar to (14), a very accurate solution, i.e., $\hat{\mathbf{f}}^{n+1}$, of (17) is unnecessary at each split Bregman iteration. In our simulations, we used 5 CG iterations for (17).

2.3.3 Wavelet frame based deblurring

The method in [7] uses the L^1-norm of a wavelet frame transform as the regularization term in (10). Thus, replacing $\mathscr{L}$ in (14) and (17) with the wavelet frame transform leads to the wavelet frame based two-phase method in [7].

2.3.4 Selection of reliables

A large number of median filter based methods have been proposed for selecting reliables from images corrupted by random valued noise. See [15] and references therein for details. Variational approaches based reliable-selection methods can be found in [16] and references therein.

In [8], iterative projection comparison based reliable-selection method was proposed. To be specific, the method in [8] computes $\mathbf{f}$ with a given pixel set Λ. Then the computed image $\mathbf{f}$ is projected (i.e., $\mathbf{Hf}$ is computed) and compared with the observed data $\mathbf{g}$ to update Λ by

$$\Lambda = \{b \in \mathbb{D} \mid |g_b - \sum_{v \in \bar{\mathbb{D}}} H_{b,v} f_v| < \epsilon\} \tag{18}$$

for a threshold parameter $\epsilon > 0$. This process continues to the next round by computing the next $\mathbf{f}$ with the newly updated Λ by (18). For details, see [8].

3 Proposed Method

3.1 Missing data artifacts

The selective use of data in (10) (recall that Λ is the set of selected pixels from the whole pixel set $\mathbb{D}$ and data on $\mathbb{D} - \Lambda$ are not used in deblurring) often generates so called missing data artifacts. To explain these artifacts, we conducted a following simulation.

Figure 1(a) shows the original true image that consists of *four blocks*, where each block is an identical 'clock' image. Figure 1(b) shows the observed image. The original true image was first blurred by 11 × 11 Gaussian PSF and then corrupted by some damage and salt-and-pepper noise. The circular region of the upper-left block in Figure 1(b) was damaged, and the circular region of the upper-right block was also damaged except its center pixel. See the zoomed image in Figure 1(d). Observed data in lower-left and lower-right blocks were corrupted by salt-and-pepper noise with probability $s = 60\%$ and $s = 90\%$, respectively.

Figure 1(c) shows

$$\sum_{b \in \Lambda} H_{b,v} \quad \text{normalization coefficients associated with } \Lambda. \tag{19}$$

Following the argument in Section 2.2, we can regard $\sum_{b \in \Lambda} H_{b,v}$ as the degree of contribution of the pixel v to selected data on Λ through the blurring transform $\mathbf{H}$.

In this simulation, for better visual comparison, we assumed that $\tilde{\mathbf{n}} = \mathbf{0}_\Lambda$ in (7). In other words, the original Gaussian noise $\mathbf{n} = \mathbf{0}_\mathbb{D}$ in (1) and the first phase selected reliables with the 'perfect accuracy' (all reliables were selected, while all outliers were excluded).

Figure 2 shows deblurred images by L2TV with selected data from Figure 1(b), with regularization parameters $\lambda = 0.001, 0.01, 0.1$ and iteration numbers $n = 20, 100, 500$. Deblurred images in Figure 2 show that the lack of data in the lower-right block and circular damaged regions in upper-left and upper-right blocks causes missing data artifacts.

Results in Figure 2 show that missing data artifacts can be removed by L2TV using long iterations and large regularization parameters. They also show that having one single pixel (the undamaged pixel at the center of the damaged circular region in the upper-right block) in Λ greatly help L2TV in suppressing missing data artifacts; the comparison of the upper-left block with the upper-right block in deblurred images supports this claim.

Figure 3 shows deblurred image by L1TV with selected data from Figure 1(b), with regularization parameters $\lambda = 0.001, 0.01, 0.1$ and iteration numbers $n = 20, 100, 500$. Results in Figure 3 are almost identical to those in Figure 2.

It is obvious that suppressing missing data artifacts by long iterations is not a good approach. Deblurred images in Figure 2 and 3 had better image quality as iteration numbers increased. This was because selected data did not contain any noise; recall that that the original Gaussian noise $\mathbf{n} = \mathbf{0}_{\mathbb{D}}$ in (1) and reliables were selected with the perfect accuracy. In practical examples, such perfect accuracy is impossible, and hence long iterations often produce noisy deblurred images.

Suppressing missing data artifacts by large regularization parameters is not a good approach, either. Large regularization parameters often give over-smoothing in deblurred images. Over-smoothing by large regularization parameters is very visible in deblurred images in the third row ((g), (h), (i)) of Figure 2 and 3.

3.2 Normalization technique

Let

$$\Omega = \{v \in \bar{\mathbb{D}} \mid \sum_{b \in \Lambda} H_{b,v} > 0\}. \tag{20}$$

Notice that Ω is determined by Λ and the pixel $v \notin \Omega$ cannot give any contribution to selected data on Λ. In Section 3.3 of this paper, we will propose a reliable-selection method which chooses Λ satisfying $\Omega = \bar{\mathbb{D}}$. Having this in mind, we assume that $\Omega = \bar{\mathbb{D}}$ in this section.

To deal with missing data artifacts, we consider a weighted inner product

$< \cdot, \cdot >_{\mathbf{w}}$ defined by

$$< \mathbf{p}, \tilde{\mathbf{p}} >_{\mathbf{w}} = \sum_{v \in \bar{\mathbb{D}}} p_v \tilde{p}_v w_v, \tag{21}$$

for all images $\mathbf{p}$ and $\tilde{\mathbf{p}}$ defined on $\bar{\mathbb{D}}$, where $\mathbf{w} = (w_v)$ is the image formed by normalization coefficients associated with Λ, i.e.,

$$w_v = \sum_{b \in \Lambda} H_{b,v}. \tag{22}$$

The use of normalization coefficients based inner product $< \cdot, \cdot >_{\mathbf{w}}$ is motivated by the principle that the pixel that gives less contribution to selected data is to be treated less importantly.

Note that the equation (14) is the result of setting the derivative of the functional (12) with respect to the standard inner product to the zero vector. Similarly, setting the derivative of the functional (12) with respect to the weighted inner product $< \cdot, \cdot >_{\mathbf{w}}$ to the zero vector leads to

$$\mathbf{W}^{-1} \left(\tilde{\mathbf{H}}^t \tilde{\mathbf{g}} + \mu \mathscr{L}^t (\mathbf{d}^n - \mathbf{c}^n) \right) = \mathbf{W}^{-1} \left(\tilde{\mathbf{H}}^t \tilde{\mathbf{H}} + \mu \mathscr{L}^t \mathscr{L} \right) \mathbf{f}, \tag{23}$$

where $\mathbf{W}$ is the matrix defined by pixel-by-pixel multiplications

$$(\mathbf{W}\mathbf{p})_v = w_v p_v \tag{24}$$

for all images $\mathbf{p}$ defined on $\bar{\mathbb{D}}$, with $\mathbf{w} = (w_v)$ in (22).

To verify (23), we generalize the definition of the derivative of a real-valued differentiable function φ defined on a m-dimensional vector space

$\mathbb{R}^m$. Suppose $\mathbb{R}^m$ is equipped with the inner product $< \cdot, \cdot >_{\mathbf{Q}}$ defined by a positive definite matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$, i.e., $< \mathbf{x}, \mathbf{y} >_{\mathbf{Q}} = \mathbf{x}^t \mathbf{Q} \mathbf{y}$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$. The derivative of φ at $\mathbf{x} \in \mathbb{R}^m$ with respect to the inner product $< \cdot, \cdot >_{\mathbf{Q}}$ is the unique vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ satisfying

$$\lim_{\mathbf{h} \to \mathbf{0}} \frac{|\varphi(\mathbf{x} + \mathbf{h}) - \varphi(\mathbf{x}) - < \boldsymbol{\alpha}, \mathbf{h} >_{\mathbf{Q}} |}{< \mathbf{h}, \mathbf{h} >_{\mathbf{Q}}^{1/2}} = 0, \quad \mathbf{h} \in \mathbb{R}^m. \tag{25}$$

The definition (25) implies that the derivative vector $\boldsymbol{\alpha}$ depends on how vectors are measured. A simple consideration shows that

$$\text{derivative vector } \boldsymbol{\alpha} \text{ of } \varphi \text{ at } \mathbf{x} \text{ with respect to } < \cdot, \cdot >_{\mathbf{Q}} = \mathbf{Q}^{-1} \nabla \varphi(\mathbf{x}). \tag{26}$$

This verifies (23).

To suppress missing data artifacts, we suggest to use the *normalized* split Bregman iteration that uses (23) and (13). By noting that this method is derived from L2TV and normalization coefficients (22), we call it *normalized L2TV* and denote it by NL2TV.

With a similar argument, setting the derivative of the functional (15) with respect to $< \cdot, \cdot >_{\mathbf{w}}$ to the zero vector leads to

$$\begin{aligned} &\mathbf{W}^{-1} \left(\mu_1 \tilde{\mathbf{H}}^t (\tilde{\mathbf{g}} + \mathbf{u}^n - \mathbf{w}^n) + \mu_2 \mathscr{L}^t (\mathbf{d}^n - \mathbf{c}^n) \right) \\ &\qquad = \mathbf{W}^{-1} \left(\mu_1 \tilde{\mathbf{H}}^t \tilde{\mathbf{H}} + \mu_2 \mathscr{L}^t \mathscr{L} \right) \mathbf{f}. \end{aligned} \tag{27}$$

We call the normalized split Bregman iteration that uses (27) and (16) *normalized L1TV* and denote it by NL1TV.

For the computation of (23) and (27), CG iterations can be used after making main matrices in (23) and (27) to be symmetric. For example, after multiplying $\mathbf{W}^{1/2}$ to the left, the right hand side of (23) can be rewritten as

$$[\mathbf{W}^{-1/2}\left(\tilde{\mathbf{H}}^t\tilde{\mathbf{H}}+\mu\mathscr{L}^t\mathscr{L}\right)\mathbf{W}^{-1/2}]\mathbf{W}^{1/2}\mathbf{f} \tag{28}$$

to make the matrix to be symmetric. The same approach also holds for (27). In [17], similar approach to (28) was used for suppressing *boundary artifacts* in image deblurring.

Algebraically, (14) and (23) are identical. However, unless extremely long iterations are used for (14) and (23), they produce noticeably different results. Thus L2TV and NL2TV produce different results. The same argument proves that L1TV and NL1TV also produce different results.

Figure 4 shows deblurred images by NL2TV and NL1TV from the observed image in Figure 1(b). In this simulation, the regularization parameter $\lambda = 0.001$ was used both for NL2TV and NL1TV. The comparison of Figure 4 with Figure 2 and 3 shows that the *normalization technique* (the use of normalization coefficients based inner product $< \cdot, \cdot >_{\mathbf{w}}$ in analyzing minimization problems (12) and (15) as in (23) and (27)) can suppress missing data artifacts without resorting to long iterations or large regularization parameters. In Figure 4, missing data artifacts in the lower-right block were clearly removed even from early iterations. Damaged regions were also recovered well by NL2TV and NL1TV; much smaller region remained as un-recovered in the upper-left block, so that it could be easily recovered by post-inpainting methods. The damaged region in the upper-right block was more easily recovered by NL2TV and NL1TV than L2TV and L1TV.

Before we close this section, it is worth to point out that vertical or horizontal lines near boundaries in Figure 2(a), 2(b), 2(c), 2(d), 2(e), 3(a), 3(b), 3(c), 3(d), and 3(e) are boundary artifacts. Various methods have been suggested for the suppression of boundary artifacts [17, 18, 19, 20]. Based on simulation results in this section, we can conclude that the proposed normalization technique suppresses boundary artifacts efficiently.

3.3 Blurring model based reliable-selection technique

In simulations in Figure 2, 3, and 4, reliables were assumed to be selected from the observed image (Figure 1(b)), which is corrupted by some damage and salt-and-pepper noise, with the perfect accuracy. In the random valued noise model, however, it is impossible to select reliables with the perfect accuracy.

Any reliable-selection methods are subject to make following two types of errors; 'selecting outliers as reliables' (the first type error) and 'excluding reliables as outliers' (the second type error).

Let

$$P_1 = \text{Prob(selected as reliables} \mid \text{outliers)} \tag{29}$$

and

$$P_2 = \text{Prob(excluded as outliers} \mid \text{reliables).} \tag{30}$$

Obviously, it is desirable for every reliable-selection method to have P_1 (the probability of the first type error) and P_2 (the probability of the second type error) as small as possible simultaneously. In all methods, however, the attempt to make P_1 smaller usually makes P_2 bigger, vice versa.

The performance of two-phase methods greatly depends on the accuracy of the reliable-selection in the first phase. Iterative projection comparison based reliable-selection method in [8] generally produces better results than median-type methods. As mentioned in Section 1, however, this iterative projection comparison based method tends to select reliables more frequently from smooth region than near-edge region, and hence often leads to missing data artifacts; no data from certain region of the image could be selected as reliables in some cases (see Figure 6). Such phenomenon can be reduced by performing the reliable-selection criterion (18) to blocks of the observed image with block-specific threshold parameters. See (31). This block-wise approach, however, often selects more unwanted data (i.e., outliers) than the original approach in [8]. It is also true that, occasionally, the block-wise approach still exhibits missing data artifacts by selecting no data as reliables from certain region. For details, see Figure 6.

In this paper, we denote iterative projection comparison based reliable-selection method in [8] by IPC and its block-wise version by BIPC.

The proposed reliable-selection method of this paper is based on BIPC. Let $\mathbb{D}_i, i = 1, 2, \ldots, B$ be disjoint block partitions of $\mathbb{D}$, i.e., $\mathbb{D} = \cup_{i=1}^{B} \mathbb{D}_i$. For given Λ, we use NL2TV or NL1TV to compute the image $\mathbf{f}$ with selected data on Λ. Then, we update $\Lambda = \cup_{i=1}^{B} \Lambda_i$ by combining

$$\Lambda_i = \{b \in \mathbb{D}_i | |g_b - \sum_{v \in \bar{\mathbb{D}}} H_{b,v} f_v| < \epsilon_i\}, \tag{31}$$

where $\epsilon_i > 0$ are block dependent threshold parameters. At this point, we check whether $\Omega = \bar{\mathbb{D}}$ after computing Ω by (20) with newly updated Λ. In case when $\Omega \neq \bar{\mathbb{D}}$, we select more pixels for Λ to make $\Omega = \bar{\mathbb{D}}$ as follows:

```
SelectMore: Given $\nu > 0$, let $S_v = \{b \in \mathbb{D} \mid H_{b,v} > 0\}$.
S1    for $v \in \bar{\mathbb{D}} - \Omega$
S2        if $\Lambda \cap S_v = \emptyset$
```

```
S3            Add $\hat{b}$ to $\Lambda$, where $\hat{b} = \arg\min_{b \in S_v} |g_b - \sum_{v \in \bar{\mathbb{D}}} H_{b,v} f_v|$;
S4            if $|g_{\hat{b}} - \sum_{v \in \bar{\mathbb{D}}} H_{\hat{b},v} f_v| > \nu$
S5                Use $H_{\hat{b},v} f_v$ for $g_{\hat{b}}$;
S6            end
S8            Update $\Omega$ by adding $\{v \in \bar{\mathbb{D}} \mid H_{\hat{b},v} > 0\}$ to $\Omega$;
S7        end
S9    end
```

Notice that S_v is the set of pixels that are under the influence of the pixel v through the blurring transform $\mathbf{H}$. Thus, the step **S2** represents the case that the pixel value at v does not contribute to selected data on Λ. In such case, we find $\hat{b}$ that is the pixel at which the projected value is the closest to the observed value and add it to Λ. Steps from **S4** to **S6** use $\sum_{v \in \bar{\mathbb{D}}} H_{\hat{b},v} f_v$ for $g_{\hat{b}}$ in case when none of projected data are close enough to observed data by ν. These steps are designed for extreme cases. In our simulations, we set $\nu = 20$. This choice made steps from **S4** to **S6** to be used only for very extreme cases. The **SelectMore** routine adds more pixels to Λ until $\Omega = \bar{\mathbb{D}}$. This process continues to the next round by computing the next $\mathbf{f}$ with newly updated Λ.

In this paper we denote the proposed blurring model based reliable-selection technique by BMRS.

The main objective of BMRS to make Λ to satisfy $\Omega = \bar{\mathbb{D}}$. By doing so, BMRS followed by the TV-based deblurring method (10) makes all pixel values on $\bar{\mathbb{D}}$ to be mainly recovered by the data fidelity term $\|\tilde{\mathbf{g}} - \tilde{\mathbf{H}}\mathbf{f}\|^p_{L^p(\Lambda)}$. This approach is motivated by simulation results in Figure 4; the damaged region in the upper-right block was recovered well since all pixels in the damaged region are in Ω by having the undamaged center pixel in Λ (see Figure 1(d)), while pixel values on $\bar{\mathbb{D}} - \Omega$ (un-recovered part in the upper-left block of Figure 4) were not recovered well. Simulation results in Figure 2 and 3 indicate that the recovery of those pixel values requires large regularization parameters, since those pixel values can be recovered only by the regularization term in (10).

As IPC in [8] can be used both for the reliable-selection and the deblurring, the proposed method, BMRS, can be also used both for the reliable-selection and the deblurring.

4 Simulation

In previous section, we proposed normalization (NL2TV and NL1TV) and BMRS techniques to improve two-phase methods in deblurring images corrupted by random valued noise. Results in Figure 2, 3, and 4 already showed that the use of the normalization technique significantly improves results of two-phase methods, by suppressing missing data artifacts effectively. In this section, we conducted several simulations to support the claim that BMRS also improves results of two-phase methods. Considering the superior performance of normalized methods (NL2TV and NL1TV) over un-normalized methods (L2TV and L1TV) and the fact that deblurred images by NL2TV are similar to those by NL1TV, we used only NL2TV in simulations.

Simulation studies were conducted for Gaussian and diagonal deblurring problems, where matrices $\mathbf{H}$ were modeled by 11×11 Gaussian PSF and 11×11 diagonal PSF, respectively, by (3). Here diagonal elements of the diagonal PSF were $66^{-1} \times \{11, 10, \dots, 1\}$. Original true images, 'girl' and 'house' were first blurred and then noised by mean zero Gaussian noise $\mathbf{n}$ with the standard deviation $= 0.5\%$ of their means. Finally, random valued noise with the probability $r = 40\%$ was applied to blurred and Gaussian noised images, $\mathbf{Hf} + \mathbf{n}$, in Figure 5(a) and 5(c). Figure 5(b) and 5(d) show final outputs, $\mathbf{g} = \mathcal{R}(\mathbf{Hf} + \mathbf{n})$.

In simulation, we chose the iterates that had the smallest RSE (Relative Square Error) in 300 iterations as the deblurred image by the tested method.

Here the RSE is defined by

$$\mathrm{RSE} = \frac{\sum_v |\hat{f}_v - f_v|^2}{\sum_v |f_v|^2}, \tag{32}$$

where $\hat{\mathbf{f}} = (\hat{f}_v)$ and $\mathbf{f} = (f_v)$ are the deblurred image and the original true image.

Figure 6 shows indicator images of selected pixels Λ by the center-weighted median filter (denoted by CWM) in (a), (e), IPC in (b), (f), BIPC in (c), (g), and BMRS in (d), (h), from observed images in Figure 5(b) and 5(d). In Figure 6, we presented unselected pixels by black color and selected pixels by white color.

As being iterative reliable-selection methods, IPC, BIPC, and BMRS need initial guess for Λ. We used CWM for this purpose.

Figure 7 shows deblurred images by NL2TV with data on selected pixels in Figure 6(a), 6(b), 6(c), and 6(d) from the observed image in Figure 5(b). In this simulation, the regularization parameter $\lambda = 0.001$ was used. Figure 8 shows deblurred images by NL2TV with data on selected pixels in Figure 6(e), 6(f), 6(g), and 6(h) from the observed image in Figure 5(d). In this simulation, the regularization parameter $\lambda = 0.1$ was used.

Table 1 summarizes the accuracy of tested reliable-selection methods and their RSE results.

Table 1. Accuracy of reliable-selection methods and RSE results. Here $P = \text{Prob(selected)} = 0.6 \times (1 - P_2) + 0.4 \times P_1$.

Gaussian deblurring (Figure 5(b))

	CWM	IPC	BIPC	BMRS
$P_1 \times 100$	2.04	1.12	1.39	1.28
$P_2 \times 100$	54.39	30.97	31.02	31.02
$P \times 100$	28.18	41.87	41.94	41.90
RSE	0.01083 (Fig. 7(a))	0.00883 (Fig. 7(b))	0.01047 (Fig. 7(c))	0.00712 (Fig. 7(d))

Diagonal deblurring (Figure 5(d))

	CWM	IPC	BIPC	BMRS
$P_1 \times 100$	1.66	3.96	4.88	2.75
$P_2 \times 100$	61.04	22.77	22.58	20.87
$P \times 100$	24.04	47.92	48.40	48.58
RSE	0.01736 (Fig. 8(a))	0.05736 (Fig. 8(b))	0.05386 (Fig. 8(c))	0.00493 (Fig. 8(d))

4.1 Results by CWM

For CWM, we used a filter which takes the median of 5×5 pixel values with the weight 7 on the value at the center pixel.

Table 1 shows that CWM selected much smaller number of reliables than iterative selection methods (IPC, BIPC, and BMRS). Figure 6(a) and 6(e) show that CWM selected pixels non-uniformly; pixels from smooth region tend to be selected more frequently than near-edge pixels. As a result, CWM based NL2TV produced poor results as shown in Figure 7(a) and 8(a); in Gaussian deblurring (Figure 7(a)), CWM based NL2TV suffered from some artifacts at near-edge regions (e.g., regions near to left cheek and right hand). In diagonal deblurring (Figure 8(a)), a wide part of the deblurred image was over-smoothed, apparently, as a result of selecting no data from a wide region (Figure 6(e)).

4.2 Results by IPC

Our experience in simulations showed that the diagonal deblurring seemed to need more selected data than the Gaussian deblurring. In Figure 6(b) and 6(f) (IPC), we used ϵ (in (18)) that can make $|\Lambda| \approx 0.42 \times |\mathbb{D}|$ for the Gaussian deblurring and $|\Lambda| \approx 0.48 \times |\mathbb{D}|$ for the diagonal deblurring.

Figure 6(b) and 6(f) show that IPC selected data non-uniformly, but not as much as CWM. IPC based NL2TV produced poor results as shown in Figure 7(b) and 8(b); in Gaussian deblurring (Figure 7(b)), IPC based NL2TV suffered from similar artifacts to what CWM based NL2TV suffered from at same near-edge regions. In diagonal deblurring (Figure 8(b)), a wide part of the deblurred image was recovered poorly as a result of selecting no data from a wide region (Figure 6(f)).

4.3 Results by BIPC

For the block-wise approach (31) in BIPC and BMRS, we used 13×13 sized blocks of observed images and ϵ_i that can make $|\Lambda_i| \approx 0.42 \times |\mathbb{D}_i|$ for the Gaussian deblurring and $|\Lambda_i| \approx 0.48 \times |\mathbb{D}_i|$ for the diagonal deblurring.

The comparison of Figure 6(c) and 6(g) with Figure 6(b) and 6(f) clearly shows that the block-wise approach (31) helped BIPC to select data more uniformly than IPC. Table 1, however, shows that the block-wise approach also increased the chance of selecting outliers as reliable. This explains why Figure 7(c) (deblurred by BIPC based NL2TV) had more artifacts and larger RSE than Figure 7(b) (deblurred by IPC based NL2TV).

Figure 7(c) and 8(c) show missing data artifacts. This implies that the block-wise approach (31) in selecting data is still not sufficient for avoiding missing data artifacts.

4.4 Results by BMRS

The only difference between BIPC and BMRS is the use of **SelectMore** (see Section 3.3) in BMRS. For BMRS, we used same sized blocks and parameters ϵ_i as BIPC.

The comparison of results by BMRS with those by BIPC in Table 1 shows that **SelectMore** helped BMRS to reduce the chance of selecting outliers as reliable. Figure 6(d) and 6(h) shows that BMRS selects data more uniformly than IPC.

As shown in Figure 7(d) and 8(d), BMRS based NL2TV produced deblurred images in better image quality than other methods. In fact, Table 1 shows that deblurred images by BMRS based NL2TV had the smallest RMS. These results show that BMRS improves results by two-phase methods by selecting data more efficiently than other methods such as CWM, IPC, and BIPC.

5 Conclusion and Discussion

In this paper, we studied the problem of deblurring images corrupted by random valued noise. We showed that the selective use of data in two-phase methods causes missing data artifacts. To suppress those missing data artifacts, we proposed blurring model based reliable-selection and normalization techniques. Simulation results showed that the proposed normalization technique greatly improves results of two-phase method based on total variational approaches in Gaussian and diagonal deblurring problems. Simulation results also showed that the proposed blurring model based reliable-selection technique excludes outliers accurately, while selecting sufficiently many reliables, and, as a result, significantly improves results of two-phase methods.

Acknowledgments

This work was supported by Inje Research and Scholarship Foundation in 2012 and conducted while the author was a visiting scholar at the Center for Nonlinear Analysis, Department of Mathematical Sciences, Carnegie Mellon University.

References

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

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

[3] L. Bar, A. Brook, N. Sochen, and N. Kiryati 2007 Deblurring of color images corrupted by impulsive noise *IEEE Trans. on Image Processing* **16**(4) 1101–1111

[4] J. Yang, Y. Zhang, and W. Yin 2009 An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise *SIAM J. Sci. Comput.* 2842–2865

[5] J.-F. Cai, R.H. Chan, and M. Nikolova 2008 Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise *Inverse Problems and Imaging* **2**(2) 187–204

[6] J.-F. Cai, S. Osher, and Z. Shen 2009 Split Bregman methods and frame based image restoration *Multiscale Model. Simul.* **8**(2) 337–369

[7] H. Ji, Z. Shen, and Y. Xu 2011 Wavelet frame based image restoration with missing/damaged pixels *East Asia Journal on Applied Mathematics* **1** 108–131

[8] M. Yan 2011 Restoration of images corrupted by impulse noise using blind inpainting and ℓ^0 norm *preprint*

[9] M. Bertalmo, L. A. Vese, G. Sapiro, and S. Osher 2003 Simultaneous structure and texture image inpainting. *IEEE Trans. Image Processing* **12**(8) 882–889

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

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

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

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

[14] D. L. Bailey, D. W. Townsend, P. E. Valk and M. N. Maisey (Eds) 2005 *Positron Emission Tomography* Springer-Verlag London Limited

[15] H.S. Yadiz and F. Homayouni 2010 Impulsive noise suppression of images using adaptive median filter *International Journal of Signal Processing, Image Processing and Pattern Recognition* **3**(1) 1–12

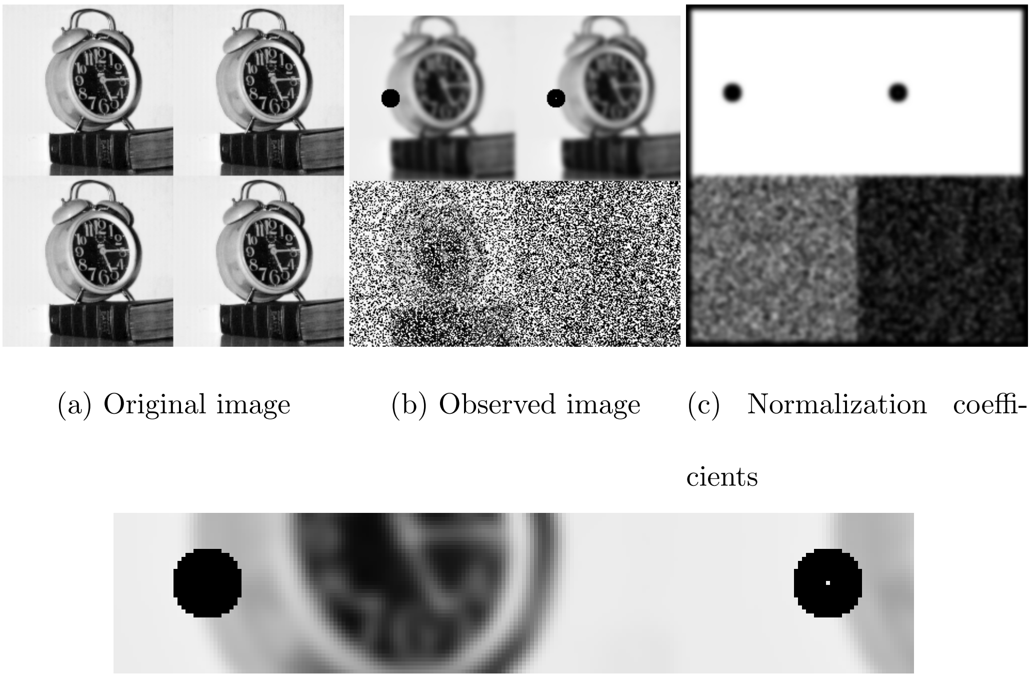
[16] M. Nikolova 2004 A variational approach to remove outliers and impulse noise *Journal of Mathematical Imaging and Vision* **20** 99–120

[17] N.-Y. Lee and B. J. Lucier 2013 Preconditioned conjugate gradient method for boundary artifact-free image deblurring *preprint*

[18] Ng M K, Chan R H, Tang W-C 1999 A fast algorithm for de-blurring models with Neumann boundary conditions *SIAM J. Sci. Comput.* **21**(3) 851–866

[19] Serra-Capizzano S 2003 A note on anti-reflective boundary conditions and fast de-blurring models *SIAM J. Sci. Comput.* **25**(4) 1307–1325

[20] Fan Y W and Nagy J G 2011 Synthetic boundary conditions for image de-blurring *Linear Algebra Appl.* **434**(11) 2244–2268



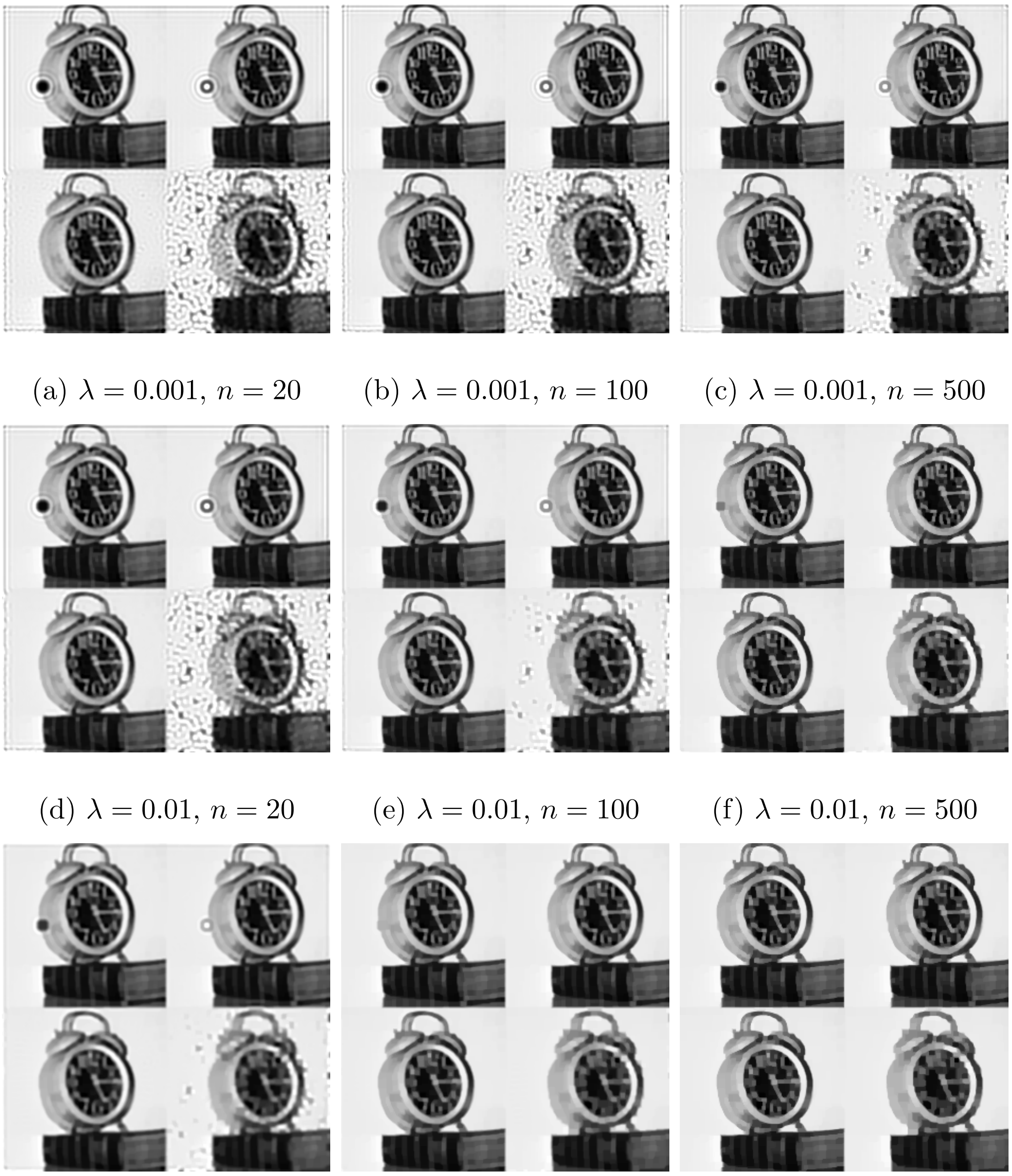
(a) Original image

(b) Observed image

(c) Normalization coefficients

(d) Zoomed part of observed image

Figure 1. (a) The original true image that consists of four identical 'clock' image blocks. (b) The observed image of size 310 × 310. The original true image of size 320 × 320 was first blurred by 11 × 11 Gaussian PSF and then corrupted by damages and salt-and-pepper noise. The circular region of the upper-left block was damaged, while the circular region of the upper-right was damaged except its center pixel. Observed data in lower-left and lower-right blocks were corrupted by salt-and-pepper noise with probability $s = 60\%$ and 90%, respectively. (c) Normalization coefficients (of size 320×320) associated with the selected pixel set Λ with the perfect accuracy. (d) A zoomed part of the observed image.



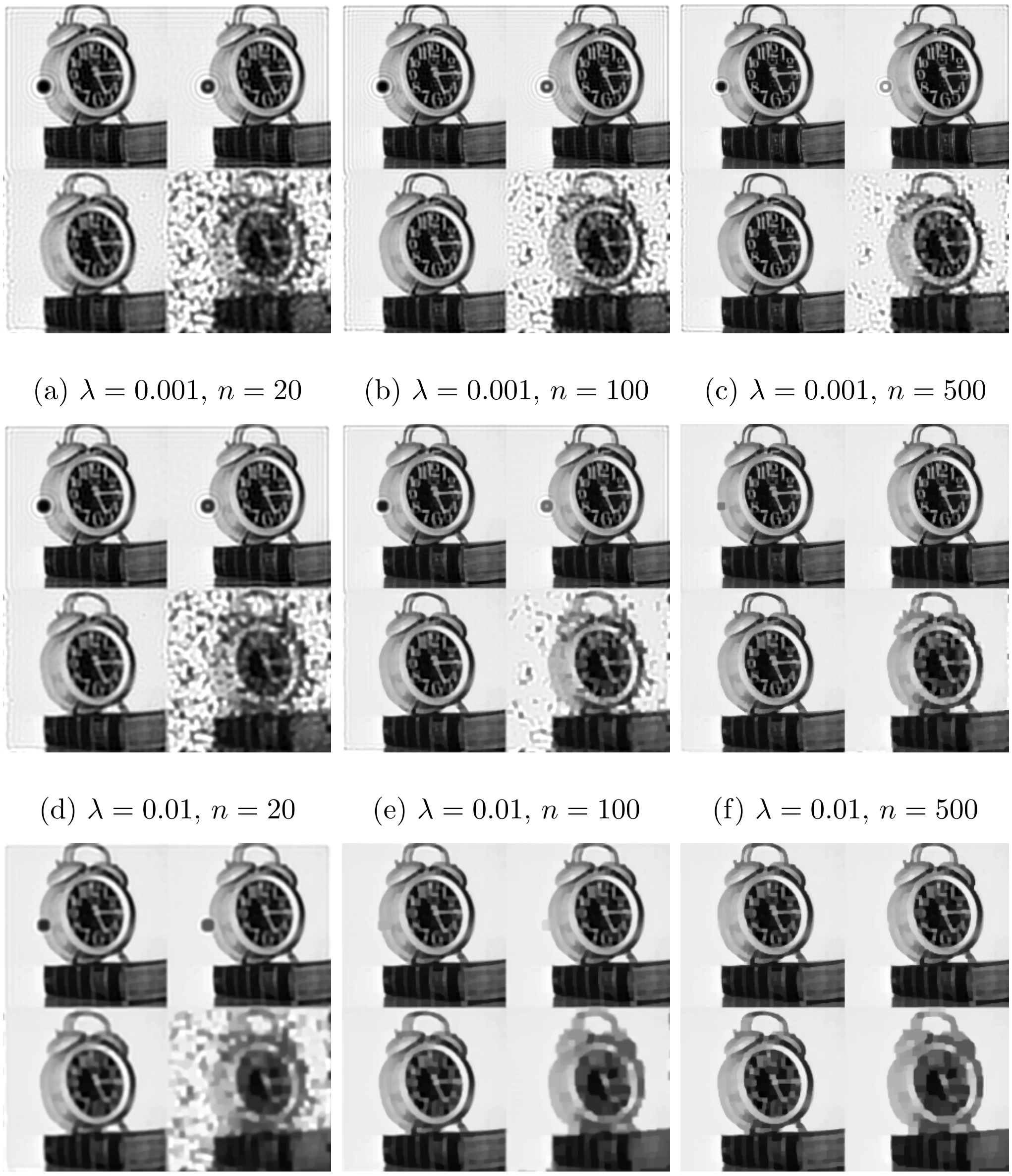
(a) $\lambda = 0.001$, $n = 20$ (b) $\lambda = 0.001$, $n = 100$ (c) $\lambda = 0.001$, $n = 500$

(d) $\lambda = 0.01$, $n = 20$ (e) $\lambda = 0.01$, $n = 100$ (f) $\lambda = 0.01$, $n = 500$

(g) $\lambda = 0.1$, $n = 20$ (h) $\lambda = 0.1$, $n = 100$ (i) $\lambda = 0.1$, $n = 500$

Figure 2. Deblurred images by L2TV with accurately selected data from the observed image in Figure 1(b), with regularization parameters $\lambda =$

$0.001, 0.01, 0.1$ and iteration numbers $n = 20, 100, 500$.



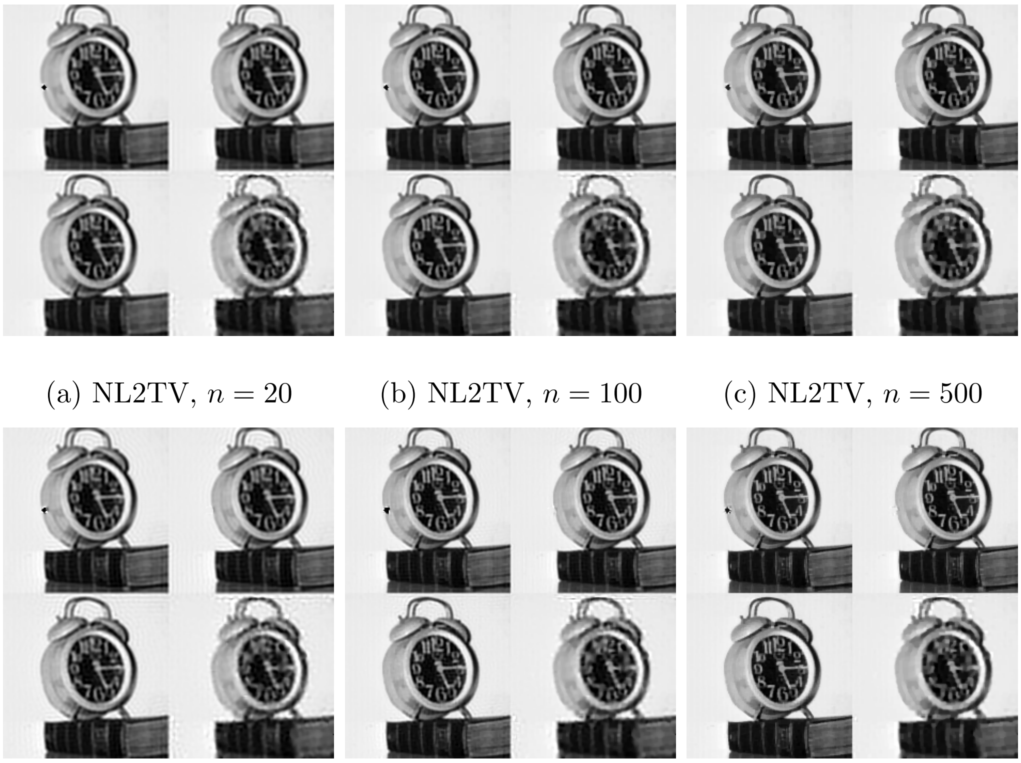
(a) $\lambda = 0.001$, $n = 20$ (b) $\lambda = 0.001$, $n = 100$ (c) $\lambda = 0.001$, $n = 500$

(d) $\lambda = 0.01$, $n = 20$ (e) $\lambda = 0.01$, $n = 100$ (f) $\lambda = 0.01$, $n = 500$

(g) $\lambda = 0.1$, $n = 20$ (h) $\lambda = 0.1$, $n = 100$ (i) $\lambda = 0.1$, $n = 500$

Figure 3. Deblurred images by L1TV with accurately selected data from the observed image in Figure 1(b), with regularization parameters $\lambda =$

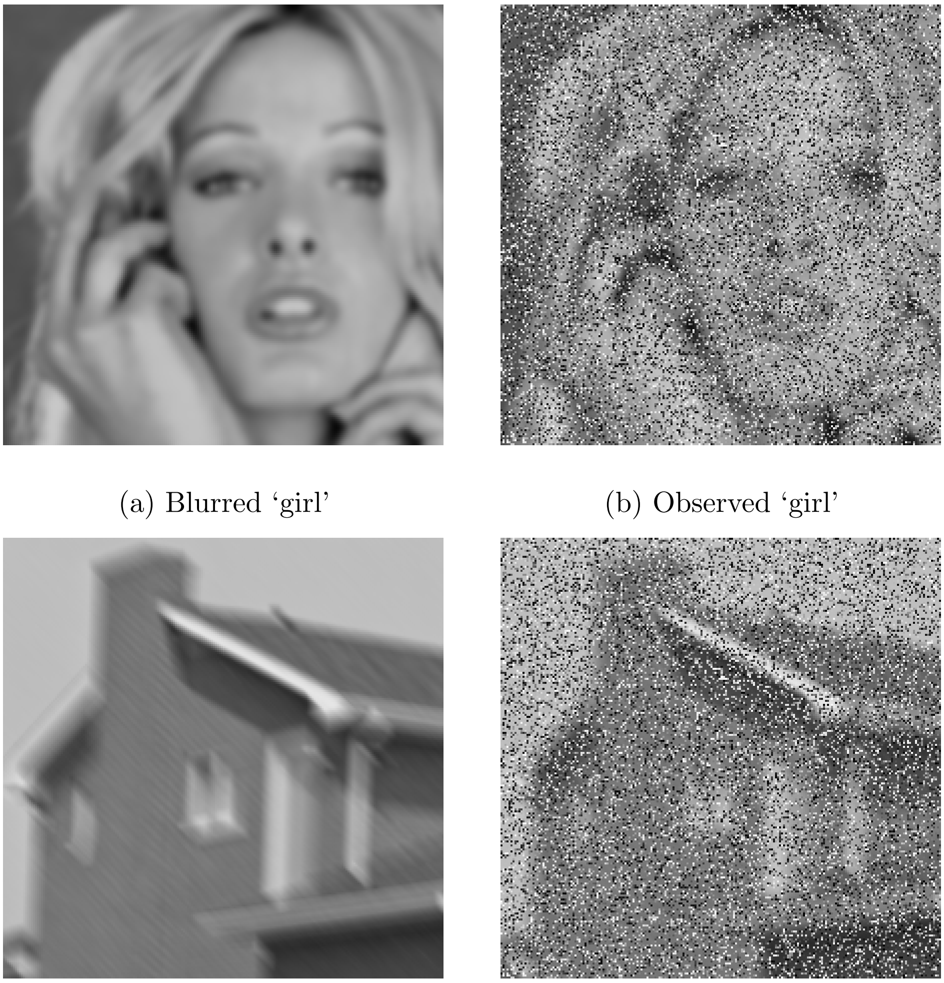
$0.1, 0.01, 0.001$ and iteration numbers $n = 20, 100, 500$.



(a) NL2TV, $n = 20$ (b) NL2TV, $n = 100$ (c) NL2TV, $n = 500$

(d) NL1TV, $n = 20$ (e) NL1TV, $n = 100$ (f) NL1TV, $n = 500$

Figure 4. Deblurred images by NL2TV ((a), (b), (c)) and NL1TV ((d), (e), (f)) with accurately selected data from the observed image in Figure 1(b), with $\lambda = 0.001$ as the regularization parameter. Here n is the iteration number.



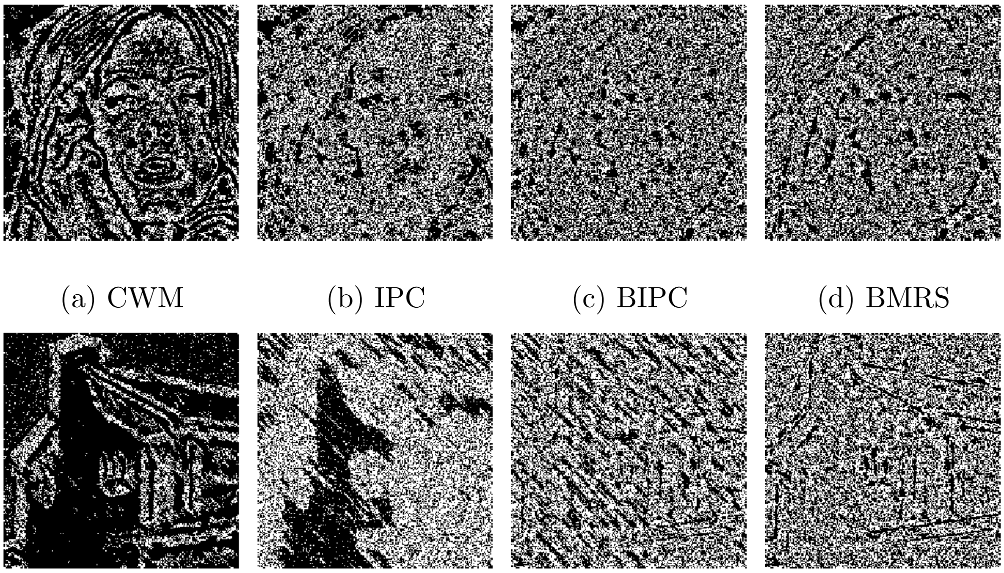
(a) Blurred ‘girl’

(b) Observed ‘girl’

(c) Blurred ‘house’

(d) Observed ‘house’

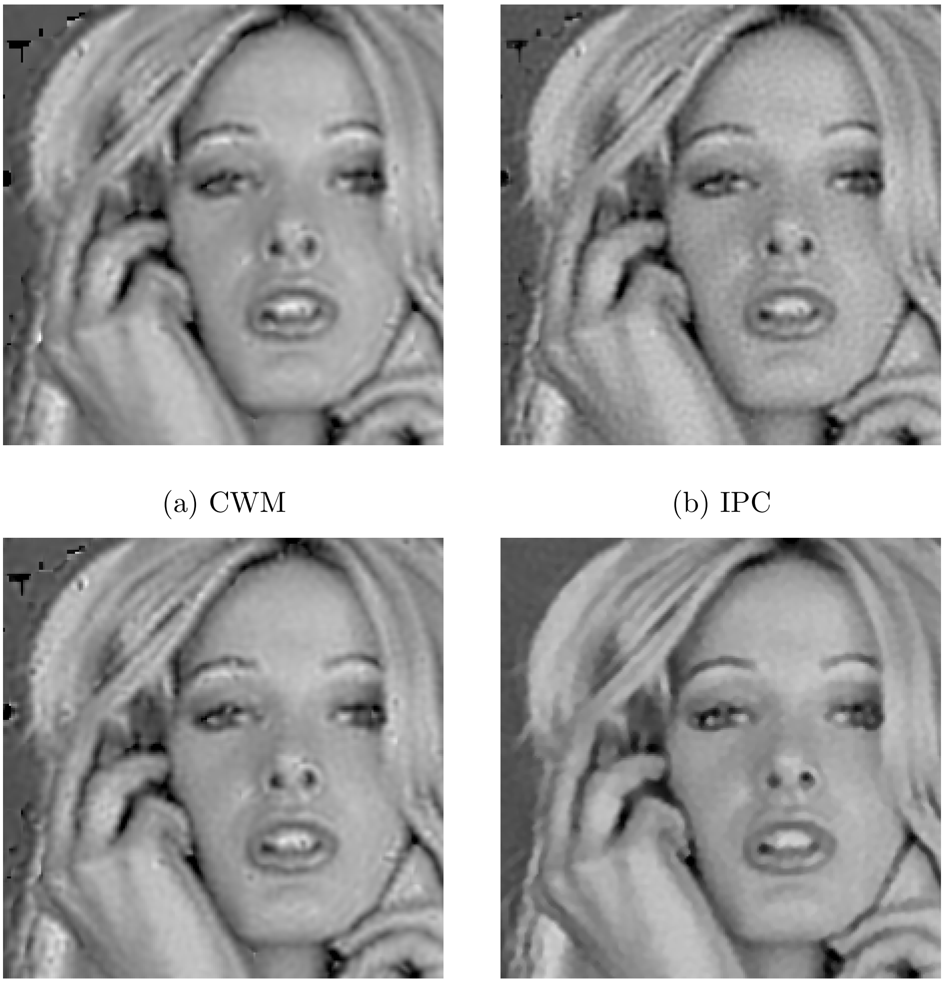
Figure 5. Blurred images and observed images. Original true images, ‘girl’ and ‘house’, were first blurred by Gaussian and diagonal PSFs, respectively, added by mean zero Gaussian noise with the standard deviation = 0.5% of mean of blurred images, and corrupted by random valued noise with the probability $r = 40\%$. Images in (a) and (c) are blurred images, i.e., $\mathbf{Hf} + \mathbf{n}$, and images in (b) and (d) are observed images, i.e., $\mathbf{g} = \mathcal{R}(\mathbf{Hf} + \mathbf{n})$ in (1).



(a) CWM (b) IPC (c) BIPC (d) BMRS

(e) CWM (f) IPC (g) BIPC (h) BMRS

Figure 6. Indicator images of selected pixels Λ by CWM in (a), (e), IPC in (b), (f), BIPC in (c), (g), and BMRS in (d), (h), where selected and unselected pixels are represented by white and black colors, respectively. Pixels in (a), (b), (c), and (d) were selected from 'girl' image in Figure 5(b), while pixels in (e), (f), (g), and (h) from 'house' image in Figure 5(d). In CWM, the 7 times center-weighted 5×5 median filter was used. In IPC, BIPC, and BMRS, NL2TV in (23) and (13) was used for computing projections in the reliable-selection. In BIPC and BMRS, the block size was set to be 13×13.



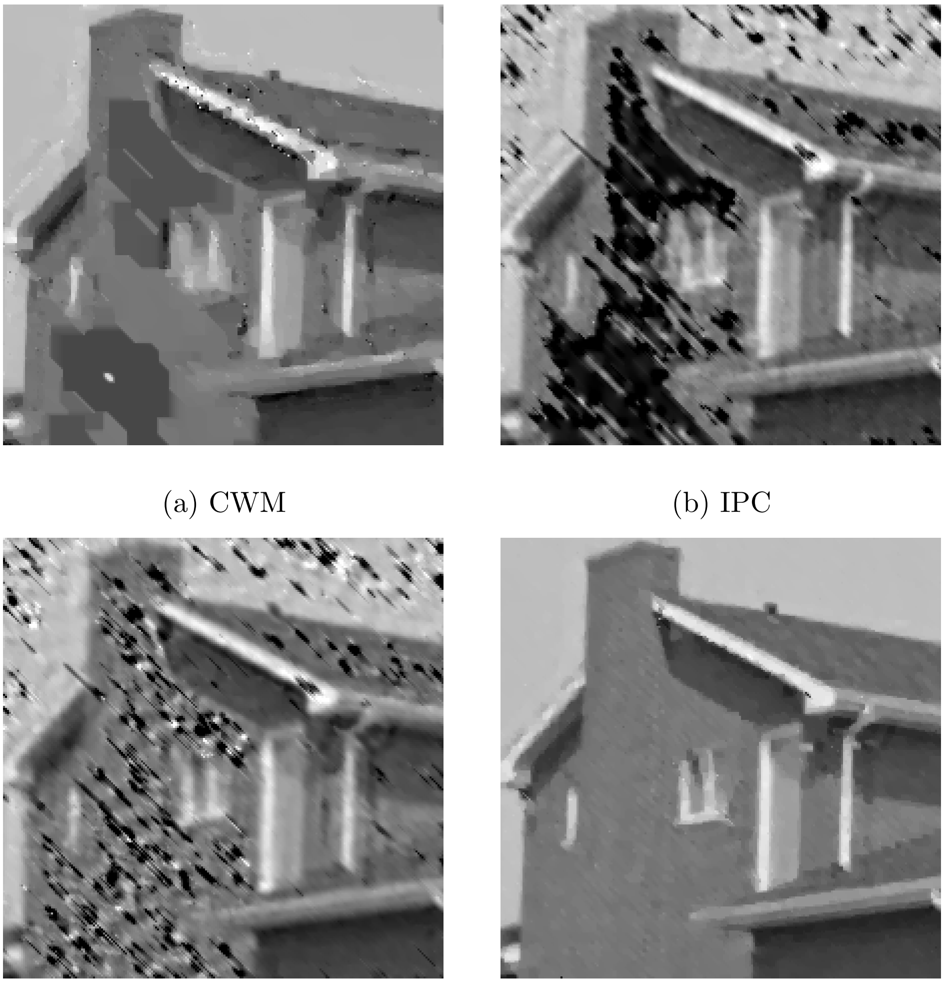
(a) CWM

(b) IPC

(a) BIPC

(b) BMRS

Figure 7. Deblurred images by NL2TV with selected data by (a) CWM, (b) IPC, (c) BIPC, and (d) BMRS from the observed image in Figure 5(b). In this simulation, the regularization parameter $\lambda = 0.001$ was used for NL2TV.



(a) CWM

(b) IPC

(c) BIPC

(d) BMRS

Figure 8. Deblurred images by NL2TV with selected data by (a) CWM, (b) IPC, (c) BIPC, and (d) BMRS from the observed image in Figure 5(d). In this simulation, the regularization parameter $\lambda = 0.1$ was used for NL2TV.