Preconditioned Conjugate Gradient Method for Boundary Artifact-Free Image Deblurring

Nam-Yong Lee*

Department of Applied Mathematics

Institute of Basic Sciences

Inje University, Gimhae, Gyeongnam 621-749, Korea

and

Bradley J. Lucier

Department of Mathematics

Purdue University, West Lafayette, IN47906, USA

*To whom correspondence should be addressed at: Department of Applied Mathematics, Inje University, Gimhae, Gyeongnam 621-749, Korea, nylee@inje.ac.kr

Abstract

Several methods have been proposed to reduce boundary artifacts in image deblurring. Some of those methods impose certain assumptions on image pixels outside the field-of-view; the most important of these assume reflective or anti-reflective boundary conditions. Boundary condition methods, including reflective and anti-reflective ones, however, often fail to reduce boundary artifacts, and, in some cases, generate their own artifacts, especially when the image to be deblurred does not accurately satisfy the imposed condition. To overcome these difficulties, we suggest using *free boundary conditions*, which do not impose any restrictions on image pixels outside the field-of-view, and *preconditioned conjugate gradient methods*, where preconditioners are designed to compensate for the non-uniformity in contributions from image pixels to the observation. Our simulation studies show that the proposed method outperforms reflective and anti-reflective boundary condition methods in removing boundary artifacts. The simulation studies also show that the proposed method can be applicable to arbitrarily shaped images and has the benefit of recovering damaged parts in blurred images.

Keywords: Boundary artifacts, Image deblurring, Convolution, Preconditioner

1 Introduction

1.1 The problem of deblurring

For an image $\mathbf{f} = (f_{j_1,j_2})$ defined for (j_1, j_2) in some rectangular domain Ω, we assume that we can observe only a noisy, blurred image

$$\mathbf{g} = \mathcal{T}\mathbf{f} + \mathbf{n}. \tag{1}$$

Here $\mathcal{T}$, sometimes called a *projector*, is a linear transform that determines the blurring process acting on the image $\mathbf{f}$. We assume that $\mathcal{T}$ can be expressed as a *truncated convolution* with a point spread function (PSF) $\mathbf{k} = (k_{j_1,j_2})$,

$$(\mathcal{T}\mathbf{f})_{i_1,i_2} = \sum_{(j_1,j_2)\in \operatorname{supp}\mathbf{k}} k_{j_1,j_2} f_{i_1-j_1,i_2-j_2}. \tag{2}$$

Here $\operatorname{supp}\mathbf{k}$, the *support* of $\mathbf{k}$, is $\{(j_1, j_2) \mid k_{j_1,j_2} > 0\}$. The PSF $\mathbf{k}$ is non-negative, its components have sum 1, and the point $(0, 0) \in \operatorname{supp}\mathbf{k}$; $\mathcal{T}\mathbf{f}$ is defined on Λ, where $(i_1, i_2) \in \Lambda$ if and only if $(i_1, i_2) - \operatorname{supp}\mathbf{k} \subseteq \Omega$.

So $\mathcal{T}$: $L^2(\Omega) \to L^2(\Lambda)$; the notation $L^2(\Omega)$ is the inner product space equipped with the inner product defined by

$$\langle \mathbf{p}, \tilde{\mathbf{p}} \rangle = \sum_{(j_1,j_2)\in\Omega} p_{j_1,j_2}\tilde{p}_{j_1,j_2}$$

for any images $\mathbf{p}$ and $\tilde{\mathbf{p}}$ that are defined on Ω. We sometimes use a *weighted* inner product on Ω; the notation $L^2(\Omega, \mathbf{w})$ means the space equipped with the weighted inner product defined by

$$\langle \mathbf{p}, \tilde{\mathbf{p}} \rangle_{\mathbf{w}} = \sum_{(j_1,j_2)\in\Omega} p_{j_1,j_2}\tilde{p}_{j_1,j_2}w_{j_1,j_2}, \tag{3}$$

where the *weight* $\mathbf{w} = (w_{j_1,j_2})$ is defined on Ω.

The result $\mathcal{T}\mathbf{f}$ is further contaminated by noise $\mathbf{n}$, which we assume to be independent and identically distributed mean zero Gaussian.

Note that $\mathcal{T}$ is not an invertible operator—there are more pixels in $\mathbf{f}$ than there are in $\mathcal{T}\mathbf{f}$.

The deblurring problem is: Assuming that we have data $\mathbf{g}$, observed from a true image $\mathbf{f}$ by the observation model (1), determine an approximation $\hat{\mathbf{f}}$ to $\mathbf{f}$. Because $\mathcal{T}$ is not invertible, this problem is *ill posed*.

1.2 Previous work: minimizing boundary artifacts

Much attention has been given to the general deblurring problem, and researchers have developed many techniques to approach this problem because there are a number of obstacles to obtaining satisfactory solutions, see [15]. From among these obstacles, we focus here mainly on the problem of *boundary artifacts* [9].

Some authors have used so-called *boundary condition* methods [11, 14, 5, 6]. In these methods, it is assumed for computational purposes that f_{j_1,j_2} in $\Omega - \Lambda$ is related to $\mathbf{f}$ in Λ via a fixed formula. Among boundary condition methods, we shall compare our method with *reflective* and *anti-reflective* boundary condition methods. To do so, we assume that the *extension operator* $\mathbb{E}\colon L^2(\Lambda) \to L^2(\Omega)$ satisfies $(\mathbb{E}\mathbf{f})_{j_1,j_2} = f_{j_1,j_2}$ for $(j_1, j_2) \in \Lambda$. Outside Λ, $\mathbf{f}$ is extended either *symmetrically* (these are reflective boundary conditions) or *anti-symmetrically* (these are anti-reflective boundary conditions); precise

definitions will be given later.

For boundary condition methods, the Tikhonov regularization approach is to find $\hat{\mathbf{f}}$ that minimizes over all $\mathbf{q} \in L^2(\Lambda)$

$$\|\mathbf{g} - \mathcal{T}\mathbb{E}\mathbf{q}\|^2_{L^2(\Lambda)} + \lambda\|\mathbf{q}\|^2_{L^2(\Lambda)}, \tag{4}$$

where λ is a positive *regularization* parameter.

Some authors have suggested that not imposing a boundary condition may lead to a better reconstruction $\hat{\mathbf{f}}$ [3, 4]. We call this a *free boundary* method; Calvetti *et al.* calls this an *Aristotelian* approach [4]. When combined with Tikhonov regularization, we find $\hat{\mathbf{f}}$ that minimizes over all $\mathbf{p} \in L^2(\Omega)$

$$\|\mathbf{g} - \mathcal{T}\mathbf{p}\|^2_{L^2(\Lambda)} + \lambda\|\mathbf{p}\|^2_{L^2(\Omega)}. \tag{5}$$

Later, we shall propose to put a weight $\mathbf{w}$ on the domain Ω, and so consider the space $L^2(\Omega, \mathbf{w})$.

In all these cases, Tikhonov regularization leads to a linear problem, which can be written in general in the form of *normal equations*

$$\mathcal{A}\mathbf{x} = \mathbf{y}, \tag{6}$$

where $\mathcal{A}$ is a positive definite operator and $\mathbf{x}$ is the minimizer of either (4) or (5) (with either $L^2(\Omega)$ or $L^2(\Omega, \mathbf{w})$). In practice, it is important that these normal equations are *not* solved exactly; most practitioners use iterative methods, often Conjugate Gradient, with a small number of iterations.

In this paper we consider an incompletely-iterated Conjugate Gradient (CG) method to find an approximate solution to these equations. Mathe-

matically, if there are N iterations of the CG method, then the image that approximately solves (6) is

$$\hat{\mathbf{x}} = \Pi^{\mathcal{A}}_{\text{span}\{\mathbf{y},\mathcal{A}\mathbf{y},\ldots,\mathcal{A}^N\mathbf{y}\}}\mathcal{A}^{-1}\mathbf{y}, \tag{7}$$

where $\Pi^{\mathcal{A}}_{\mathcal{X}}\mathbf{h}$ is the projection of $\mathbf{h}$ onto the space $\mathcal{X}$ with the inner product defined as $\langle \mathbf{u}, \mathbf{v}\rangle_{\mathcal{A}} = \langle \mathbf{u}, \mathcal{A}\mathbf{v}\rangle$.

1.3 Our approach

We combine a number of previous approaches to this problem. In particular, we propose combining Tikhonov regularization, free boundary conditions, and incomplete CG iterations. To be specific, we propose the Tikhonov regularization to find $\hat{\mathbf{f}}$ that minimizes over all $\mathbf{p} \in L^2(\Omega, \mathbf{w})$

$$\|\mathbf{g} - \mathcal{T}\mathbf{p}\|^2_{L^2(\Lambda)} + \lambda\|\mathbf{p}\|^2_{L^2(\Omega,\mathbf{w})}; \tag{8}$$

here $\mathbf{w} = \mathcal{T}^*\mathbf{I}_\Lambda$, where $\mathcal{T}^*$ is the adjoint operator of $\mathcal{T}$ and $\mathbf{I}_\mathcal{X}$ is the image of all 1s on $\mathcal{X}$. Thus, the suggested normal equation is

$$(\mathcal{T}^*\mathcal{T} + \lambda\mathcal{W})\,\mathbf{p} = \mathcal{T}^*\mathbf{g}, \tag{9}$$

or

$$\mathcal{A}\mathbf{x} = \mathbf{y}, \quad \text{where } \mathcal{A} = \mathcal{T}^*\mathcal{T} + \lambda\mathcal{W},\ \mathbf{y} = \mathcal{T}^*\mathbf{g}, \text{ and } \mathbf{x} = \mathbf{p}; \tag{10}$$

here $\mathcal{W}$ is the diagonal operator defined by the the weight $\mathbf{w}$, that is, $\mathcal{W}\mathbf{p} = \mathbf{w} .* \mathbf{p}$ for $\mathbf{p} \in L^2(\Omega, \mathbf{w})$, where $.*$ is the pixel-by-pixel multiplication.

Additionally, we shall find that a better reconstruction occurs when we *precondition* the CG iteration with $\mathcal{W}$. Mathematically, this is equivalent to

applying *un-preconditioned* CG iterations to the modified problem

$$[\mathcal{W}^{-1/2}\left(\mathcal{T}^*\mathcal{T}+\lambda\mathcal{W}\right)\mathcal{W}^{-1/2}](\mathcal{W}^{1/2}\mathbf{p}) = \mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g} \tag{11}$$

or

$$\tilde{\mathcal{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{y}}, \tag{12}$$

where $\tilde{\mathbf{y}} = \mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}$, $\tilde{\mathcal{A}} = \mathcal{W}^{-1/2}\left(\mathcal{T}^*\mathcal{T}+\lambda\mathcal{W}\right)\mathcal{W}^{-1/2}$, and $\tilde{\mathbf{x}} = \mathcal{W}^{1/2}\mathbf{p}$.

We have the relationship $\mathbf{x} = \mathcal{W}^{-1/2}\tilde{\mathbf{x}}$. However, after N iterations, preconditioned CG computes the image

$$\mathcal{W}^{-1/2}\Pi^{\tilde{\mathcal{A}}}_{\text{span}\{\mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g},\tilde{\mathcal{A}}\mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g},\ldots,\tilde{\mathcal{A}}^N\mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}\}}\tilde{\mathcal{A}}^{-1}\mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}, \tag{13}$$

which is in general different from the image computed without preconditioning

$$\Pi^{\mathcal{A}}_{\text{span}\{\mathcal{T}^*\mathbf{g},\mathcal{A}\mathcal{T}^*\mathbf{g},\ldots,\mathcal{A}^N\mathbf{y}\}}\mathcal{A}^{-1}\mathcal{T}^*\mathbf{g}. \tag{14}$$

We shall show through experiments that this combination of techniques results in reduced boundary artifacts compared to the other methods mentioned.

1.4 Outline

This paper is outlined as follows. In Section 2 we review notation, terminology, and background material, including reflective and anti-reflective boundary conditions. In Section 3 we suggest free boundary conditions and preconditioned CG iteration as a method for boundary artifact removal. In Section 4 we present simulation results of the proposed method with comparison to methods using reflective and anti-reflective boundary conditions, and

applications of the proposed method. Finally, we present some discussion and conclusions in Section 5.

2 Background

2.1 Notation and terminology

In this paper, we shall use following notations, conventions, and terminology.

- $\mathbf{g} = \mathcal{T}\mathbf{f} + \mathbf{n}$: the observation model, where $\mathbf{f}$ is the true image to be recovered, $\mathcal{T}$ is the projector that represents the space-invariant blurring of a given problem, $\mathbf{g}$ is the observed image, and $\mathbf{n}$ is Gaussian noise.
- $\mathbf{f} = (f_{j_1,j_2})$: the image, $\mathbf{f}$, will be denoted in a bold-faced alphabet, while its image pixel value, f_{j_1,j_2} at (j_1, j_2), will be denoted in a normal alphabet with subscripted indices. The same rule will hold for other images and PSFs throughout this paper.
- Λ: the set of image pixels where $\mathbf{g}$ is defined.
- Ω: the set of image pixels where $\mathbf{f}$ is defined.
- $\Omega - \Lambda$: the set of *"unseen" image pixels* across the boundary of Λ.
- $L^2(\Lambda)$: the set of images defined on Λ; for instance, $\mathbf{g} \in L^2(\Lambda)$.
- $L^2(\Omega)$: the set of images defined on Ω; for instance, $\mathbf{f} \in L^2(\Omega)$.

- $\mathcal{I}_\Lambda$: the identity map on the image space defined on Λ.
- $\mathcal{I}_\Omega$: the identity map on the image space defined on Ω.
- $\mathbf{I}_\Theta$: the all-one image on a pixel set $\Theta \subseteq \Lambda$ (or Ω),

$$(\mathbf{I}_\Theta)_{i_1,i_2} = \begin{cases} 1, & \text{if } (i_1,i_2) \in \Theta, \\ 0, & \text{if } (i_1,i_2) \notin \Theta. \end{cases} \tag{15}$$

- $\mathbf{k}$: the PSF that defines the projector $\mathcal{T}$ by (2).
- $\bar{\mathbf{k}}$: the reverse ordered PSF of $\mathbf{k} = (k_{i_1,i_2})$,

$$\bar{\mathbf{k}} = (\bar{k}_{i_1,i_2}) \quad \text{and} \quad \bar{k}_{i_1,i_2} = k_{-i_1,-i_2}. \tag{16}$$

- $\operatorname{supp}\mathbf{k}$: the support of $\mathbf{k}$, the set of (i_1,i_2) such that $k_{i_1,i_2} \neq 0$.
- Standard backprojector $\mathcal{T}^*$: the adjoint transform $\mathcal{T}^*\colon L^2(\Lambda) \to L^2(\Omega)$ associated with the projector $\mathcal{T}\colon L^2(\Omega) \to L^2(\Lambda)$, which is uniquely determined by the relation

$$(\mathbf{q}, \mathcal{T}\mathbf{p}) = \langle \mathcal{T}^*\mathbf{q}, \mathbf{p} \rangle, \tag{17}$$

  for all $\mathbf{q} \in L^2(\Lambda)$ and $\mathbf{p} \in L^2(\Omega)$. Here $(\cdot,\cdot)$ and $\langle\cdot,\cdot\rangle$ are the inner products of two images in $L^2(\Lambda)$ and $L^2(\Omega)$, respectively.

Throughout this paper, we assume that the PSF $\mathbf{k}$ is non-negative with positive origin,

$$k_{i_1,i_2} \geq 0 \quad \text{and} \quad k_{0,0} > 0, \tag{18}$$

and normalized to 1, i.e.,

$$\sum_{(i_1,i_2)\in\operatorname{supp}\mathbf{k}} k_{i_1,i_2} = 1. \tag{19}$$

2.2 Projector and backprojector

The relation between the PSF $\mathbf{k}$ and the projector $\mathcal{T}$ is defined by (2). In this paper, the right hand side of (2) will be denoted by $\mathbf{k} *_V \mathbf{f}$, i.e.,

$$\mathcal{T}\mathbf{p} = \mathbf{k} *_V \mathbf{p} \quad \text{for all } \mathbf{p} \in L^2(\Omega), \tag{20}$$

and called the *valid convolution* of $\mathbf{k}$ and $\mathbf{f}$. For future use, let the *full convolution* of $\bar{\mathbf{k}}$ and $\mathbf{q}$ in $L^2(\Lambda)$ be defined by

$$(\bar{\mathbf{k}} *_F \mathbf{q})_{j_1,j_2} = \sum_{(i_1,i_2)\in \operatorname{supp}\mathbf{k}\cap((j_1,j_2)+\Lambda)} k_{i_1,i_2} q_{i_1-j_1,i_2-j_2}. \tag{21}$$

Here $(\bar{\mathbf{k}} *_F \mathbf{q})_{j_1,j_2}$ is defined for (j_1, j_2) if and only if $\operatorname{supp}\mathbf{k}\cap((j_1,j_2)+\Lambda) \neq \emptyset$.

In the previous section, we defined Ω to be the set of image pixels where the true image $\mathbf{f}$ is defined. Since we cannot recover $\mathbf{f}$ on image pixels that do not give any contribution to the observed image $\mathbf{g}$, we can redefine Ω to be the set of image pixels that actually contribute to the observed image $\mathbf{g}$ through the blurring by the PSF $\mathbf{k}$. Following this rule, we have

$$\Omega = \{(j_1, j_2) \mid (\bar{\mathbf{k}} *_F \mathbf{I}_\Lambda)_{j_1,j_2} > 0\}. \tag{22}$$

Here we note that this result implies $\Lambda \subset \Omega$, since $k_{0,0} > 0$. The proof of (22) immediately follows from the definition of the full convolution.

Previously, we defined the standard backprojector $\mathcal{T}^*$ abstractly by the basic theory of linear algebra. When $\mathcal{T}$ is defined by the valid convolution by the PSF $\mathbf{k}$, a more practical definition of the standard backprojector $\mathcal{T}^*$ is based on following result:

$$(\forall \mathbf{p} \in L^2(\Omega)) \quad \mathcal{T}\mathbf{p} = \mathbf{k} *_V \mathbf{p} \Longleftrightarrow (\forall \mathbf{q} \in L^2(\Lambda)) \quad \mathcal{T}^*\mathbf{q} = \bar{\mathbf{k}} *_F \mathbf{q}. \tag{23}$$

The proof of (23) also immediately follows from the definition of valid and full convolutions.

The computation of $\mathcal{T}\mathbf{p}$ (or $\mathbf{k} *_V \mathbf{p}$) and $\mathcal{T}^*\mathbf{q}$ (or $\bar{\mathbf{k}} *_F \mathbf{q}$) can be performed either by pixel-wise definitions (2) and (21) or by the fast Fourier transform (FFT) with zero-paddings. Pixel-wise computations are preferred for PSFs with small support, while FFT-based computations are needed for PSFs with large support.

2.3 Boundary condition methods

In this section, we shall explain three boundary conditions (reflective, anti-reflective, and free), which have been studied by many researchers [3, 11, 14, 5, 6, 4].

Reflective and anti-reflective boundary condition methods are virtually limited to rectangular shaped images. To present them, we assume that

$$\Lambda = \{(i_1, i_2) \mid 0 \le i_\nu < N_\nu \ (\nu = 1, 2)\} \tag{24}$$

for some positive integers N_1 and N_2. To avoid some technical difficulty, we also assume that

$$\operatorname{supp} \mathbf{k} = \{(i_1, i_2) \mid -L_\nu \le i_\nu \le M_\nu \ (\nu = 1, 2)\} \tag{25}$$

for some positive integers L_1, L_2, M_1, and M_2. In this case,

$$\Omega = \{(j_1, j_2) \mid -M_\nu \le j_\nu \le N_\nu + L_\nu - 1 \ (\nu = 1, 2)\}. \tag{26}$$

We divide the true image $\mathbf{f}$ into 9 parts as follows:

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_{nw} & \mathbf{f}_{n} & \mathbf{f}_{ne} \\ \mathbf{f}_{w} & \mathbf{f}_{c} & \mathbf{f}_{e} \\ \mathbf{f}_{sw} & \mathbf{f}_{s} & \mathbf{f}_{se} \end{pmatrix}, \tag{27}$$

where $\mathbf{f}_c = (f_{i_1,i_2})$, $0 \le i_\nu < N_\nu$ $(\nu = 1, 2)$ represents the image part defined on Λ; each of the other eight parts, $\mathbf{f}_{nw}$, $\mathbf{f}_n$, $\mathbf{f}_{ne}$, $\mathbf{f}_w$, $\mathbf{f}_e$, $\mathbf{f}_{sw}$, $\mathbf{f}_s$, and $\mathbf{f}_{se}$ in (27), represents part of the image defined on $\Omega - \Lambda$, the set of unseen image pixels across the boundary of Λ.

Boundary condition methods impose certain restrictions on $\mathbf{f}_{nw}$, $\mathbf{f}_n$, $\mathbf{f}_{ne}$, $\mathbf{f}_w$, $\mathbf{f}_e$, $\mathbf{f}_{sw}$, $\mathbf{f}_s$, and $\mathbf{f}_{se}$ in (27). For example, the i-th row of the reflective boundary condition imposed image is

$$\begin{aligned} &(f_{i,M_2-1}, \cdots, f_{i,1}, f_{i,0}, \\ &\quad f_{i,0}, f_{i,1}, \cdots, f_{i,N_2-2}, f_{i,N_2-1}, \\ &\quad\quad f_{i,N_2-1}, f_{i,N_2-2}, \cdots, f_{i,N_2-L_2}), \end{aligned} \tag{28}$$

and similarly for the column, while the i-th row of the anti-reflective boundary condition imposed image is

$$\begin{aligned} &(2f_{i,0} - f_{i,M_2}, \cdots, 2f_{i,0} - f_{i,2}, 2f_{i,0} - f_{i,1}, \\ &\quad f_{i,0}, f_{i,1}, \cdots, f_{i,N_2-1}, f_{i,N_2-1}, \\ &\quad\quad 2f_{i,N_2-1} - f_{i,N_2-2}, \cdots, 2f_{i,N_2-1} - f_{i,N_2-L_2-1}), \end{aligned} \tag{29}$$

and similarly for the column.

Any set of boundary conditions introduces an *extension operator* $\mathbb{E}\colon L^2(\Lambda) \to L^2(\Omega)$ such that

$$\mathbb{E}\mathbf{f}_c = \begin{pmatrix} \tilde{\mathbf{f}}_{nw} & \tilde{\mathbf{f}}_{n} & \tilde{\mathbf{f}}_{ne} \\ \tilde{\mathbf{f}}_{w} & \mathbf{f}_{c} & \tilde{\mathbf{f}}_{e} \\ \tilde{\mathbf{f}}_{sw} & \tilde{\mathbf{f}}_{s} & \tilde{\mathbf{f}}_{se} \end{pmatrix}, \tag{30}$$

where $\tilde{\mathbf{f}}_{nw}$, $\tilde{\mathbf{f}}_{n}$, $\tilde{\mathbf{f}}_{ne}$, $\tilde{\mathbf{f}}_{w}$, $\tilde{\mathbf{f}}_{e}$, $\tilde{\mathbf{f}}_{sw}$, $\tilde{\mathbf{f}}_{s}$, and $\tilde{\mathbf{f}}_{se}$ represent parts of the image imposed by the boundary condition. The operators $\mathbb{E}$ associated with reflective and anti-reflective boundary conditions can be defined by (28) and (29), respectively.

In [11, 14, 5], reflective and anti-reflective boundary condition methods take the form (4) as Tikhonov regularization [7]. In this work, to test the performance of reflective and anti-reflective boundary condition methods, we shall apply CG iterations to the normal equation

$$(\mathbb{E}^{*}\mathcal{T}^{*}\mathcal{T}\mathbb{E} + \lambda\mathcal{I}_{\Lambda})\,\mathbf{q} = \mathbb{E}^{*}\mathcal{T}^{*}\mathbf{g}, \tag{31}$$

derived from (4). We denote by RBC reflective boundary condition–based CG iterations, and we denote by ABC anti-reflective boundary condition–based CG iterations.

For explanatory purpose, we use the term *free boundary condition* to refer to the method suggested in [3, 4], even though the suggested method does not impose any boundary conditions whatsoever.

In [4], Calvetti et al. claim that, "In an Aristotelian approach to knowledge, when it is not known a priori which boundary conditions should be chosen, by admitting our lack of information it is possible to let the data itself determine them." Thus, free boundary conditions do not need an extension operator to impose boundary conditions, and hence, this approach can be applied to arbitrarily shaped images. The proposed method in this paper will show that such flexibility in dealing with boundary artifacts gives several advantages to free boundary conditions over reflective and anti-reflective boundary conditions, which can be applied only to rectangular shaped im-

ages.

To test the performance of free boundary conditions, we shall apply the CG method to the linear system

$$(\mathcal{T}^*\mathcal{T} + \lambda\mathcal{I}_\Omega)\,\mathbf{p} = \mathcal{T}^*\mathbf{g}, \tag{32}$$

which is derived from the Tikhonov regularization (5). Form now on, we shall denote the standard CG iterations applied to (32) by FBC.

3 Proposed Method

3.1 Free boundary condition

The use of the free boundary conditions begins with determining Ω from (22). We use Figure 1 to explain this process: Notice that the observed image $\mathbf{g}$ in Figure 1(a) has a non-rectangular boundary, where the dark background indicates the region where no observation is available. In Figure 1(a), the blurring is performed by 17×17 Gaussian PSF with standard deviation the width of 3 pixels. Figure 1(b) shows the pixel set Ω, which consists of two parts: one is the observed region Λ, indicated in white, and the unseen region $\Omega - \Lambda$, indicated in gray. Note that the border line between the white and the gray colored regions in Figure 1(b) is the boundary of the observed image $\mathbf{g}$. This process clearly shows that free boundary conditions can be applied to arbitrarily shaped images. Note also that this kind of flexibility of free boundary conditions to boundary shapes is not available to reflective and anti-reflective boundary conditions, which are limited to rectangular shaped

images only.

The free boundary condition alone, or equivalently FBC, does not remove boundary artifacts, as we can see in Figure 2(a), which is the deblurred image by FBC (the image is obtained by 100 CG iterations). Despite FBC failing to remove boundary artifacts, we suggest using FBC as the first step to avoid boundary artifacts caused by the use of inappropriate boundary conditions, by noting in the next section that boundary artifacts in FBC can be suppressed by using CG preconditioning.

3.2 Preconditioned CG iterations

To deal with boundary artifacts, we propose a weight $\mathbf{w}$ defined by

$$\mathbf{w} = \mathcal{T}^* \mathbf{I}_\Lambda = \bar{\mathbf{k}} *_F \mathbf{I}_\Lambda, \tag{33}$$

on the domain Ω. The inner product defined by this weight as in (3) determines $L^2(\Omega, \mathbf{w})$. FBCW will denote CG iterations applied to the normal equation (9) derived from the $L^2(\Omega, \mathbf{w})$–based Tikhonov regularization (8).

Notice that the pixel value w_{j_1,j_2} of $\mathbf{w}$ represents the degree of the contribution of the image pixel at (j_1, j_2) to the observed image on Λ, through the blurring transform $\mathcal{T}$. For example, $w_{j_1,j_2} = 1$ (the maximum value w_{j_1,j_2} can have, by (19)) implies that the information at the image pixel (j_1, j_2) is spread out (by the blurring transform $\mathcal{T}$) to other image image pixels (i_1, i_2), all of which belong to Λ. In other words, none of the information at (j_1, j_2) is lost by the blurring. On the other hand, $0 < w_{j_1,j_2} < 1$ implies that $(1 - w_{j_1,j_2}) \times 100\%$ of information at the image pixel (j_1, j_2) is missing, or

equivalently, not observed in the observed image on Λ, due to the truncation in observation.

The suggestion of the $\mathbf{w}$ based inner product in FBCW is motivated by the principle that the image pixel that gives less contribution to the observation should be treated less importantly. FBCW, however, does not remove boundary artifacts. See Figure 14.

The failure of FBCW to remove boundary artifacts is expected, as the regularization parameter λ in FBCW must be set very small (in our simulation, $\lambda = 0.0001$), in order not to have over-smoothed results. In other words, the use of the $\mathbf{w}$-based inner product in FBCW cannot be effective since λ is very small in FBCW. In fact, differences between the results of FBC and FBCW are hardly noticeable for any practical choice of λ.

To make the use of the $\mathbf{w}$-based inner product effective, we consider the preconditioned CG iteration to (9) by a diagonal operator $\mathcal{W}$ defined by

$$(\mathcal{W}\mathbf{p})_{j_1,j_2} = w_{j_1,j_2} p_{j_1,j_2} \quad \text{for any image } \mathbf{p} \text{ defined on } \Omega. \tag{34}$$

Preconditioned conjugate gradient iterations of FBCW are equivalent to standard conjugate gradient iterations applied to

$$[\mathcal{W}^{-1/2} (\mathcal{T}^*\mathcal{T} + \lambda\mathcal{W}) \mathcal{W}^{-1/2}](\mathcal{W}^{1/2}\mathbf{p}) = \mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}, . \tag{11$'$}$$

We shall denote by FBCWP conjugate gradient iterations applied to (11).

Figure 2(b) shows the deblurred image by FBCWP (the image is obtained by 100 CG iterations). The boundary artifacts in Figure 2(a) are completely removed in Figure 2(b).

To make clear, the exact solutions of the two linear systems

$$\mathcal{W}^{-1/2}\left(\mathcal{T}^*\mathcal{T}+\lambda\mathcal{W}\right)\mathcal{W}^{-1/2}\mathcal{W}^{1/2}\mathbf{p}=\mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}$$

and

$$\left(\mathcal{T}^*\mathcal{T}+\lambda\mathcal{W}\right)\mathbf{p}=\mathcal{T}^*\mathbf{g}$$

are the *same*; nonetheless, the sequence of conjugate-gradient iterates for both linear systems generally *differ* until they converge to the exact solution (assuming exact arithmetic) at the final, $\dim(L^2(\Omega))$th, iterate. Because the best results are obtained after a relatively small number of iterates, our result images differ.

4 Simulation Results

We conducted simulation studies to compare incomplete CG iterations applied to the following normal equations in boundary artifact removal in image deblurring:

- **RBC**: Reflective boundary conditions and incomplete CG iteration applied to

  $$\left(\mathbb{E}^*\mathcal{T}^*\mathcal{T}\mathbb{E}+\lambda\mathcal{I}_\Lambda\right)\mathbf{q}=\mathbb{E}^*\mathcal{T}^*\mathbf{g}, \tag{31$'$}$$

  where $\mathbb{E}$ is the extension operator associated with the reflective boundary condition.

- **ABC**: Anti-reflective boundary conditions and incomplete CG iteration applied to (31) with the extension operator $\mathbb{E}$ that is associated with the anti-reflective boundary condition.

- **FBC**: Free boundary conditions and incomplete CG iteration applied to

$$(\mathcal{T}^*\mathcal{T} + \lambda\mathcal{I}_\Omega)\,\mathbf{p} = \mathcal{T}^*\mathbf{g}. \tag{32$'$}$$

- **FBCW**: Free boundary conditions, $L^2(\Omega, \mathbf{w})$–weighted norm, and incomplete CG iteration applied to

$$(\mathcal{T}^*\mathcal{T} + \lambda\mathcal{W})\,\mathbf{p} = \mathcal{T}^*\mathbf{g}. \tag{9$'$}$$

- **FBCWP**: Free boundary conditions, $L^2(\Omega, \mathbf{w})$–weighted norm, and incomplete CG iteration applied to

$$[\mathcal{W}^{-1/2}\,(\mathcal{T}^*\mathcal{T} + \lambda\mathcal{W})\,\mathcal{W}^{-1/2}](\mathcal{W}^{1/2}\mathbf{p}) = \mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}. \tag{11$'$}$$

- **FBCP**: Free boundary conditions and incomplete CG iteration applied to

$$[\mathcal{W}^{-1/2}\,(\mathcal{T}^*\mathcal{T} + \lambda\mathcal{I}_\Omega)\,\mathcal{W}^{-1/2}](\mathcal{W}^{1/2}\mathbf{p}) = \mathcal{W}^{-1/2}\mathcal{T}^*\mathbf{g}. \tag{35}$$

In most case, we shall omit results by FBCW and FBCP, since they are almost identical to results by FBC and FBCWP, respectively. Here we note that the "incomplete CG iteration" is a necessary requirement to avoid noise amplification that would be generated by "complete CG iteration".

In simulation studies, we used the "Airfield"' image in Figure 3(a), as the true image $\mathbf{f}$, and three different PSFs (uniform, Gaussian, and diagonal gradient) as our image blurring models.

For noise model, we used a noise $\mathbf{n}$ such that such as

$$\mathbf{n} \sim \text{Normal}(\mathbf{0}, \sigma^2 \mathrm{I}_{|\Lambda|}), \tag{36}$$

where $\mathbf{0}$ is the all zero image defined on Λ and $\mathrm{I}_{|\Lambda|}$ is the $|\Lambda| \times |\Lambda|$ identity matrix. Throughout his paper, we assume that the standard deviation σ in (36) is set to be 0.5% of the average value of $\mathcal{T}\mathbf{f}$. We use such little noise so that changes in boundary artifacts will be visually noticeable.

We chose the deblurred image that had the smallest RSE (Relative Square Error) in 200 iterations for each simulation. Here the RSE is defined by

$$\mathrm{RSE} = \frac{\sum_{i_1,i_2} |\tilde{f}_{i_1,i_2} - f_{i_1,i_2}|^2}{\sum_{i_1,i_2} |f_{i_1,i_2}|^2}, \tag{37}$$

where $\tilde{f}_{i_1,i_2}$ and f_{i_1,i_2} are pixel values of the deblurred image and the true image, respectively, at the pixel $(i_1, i_2) \in \Lambda$. This restriction is made for fair comparison. Notice that RBC and ABC recover images that are the same size as the observed image, while FBC, FBCW, and FBCWP recover images with all pixels that give any contributions to the observed image. For instance, in the simulation with the 11×11 uniform PSF and the true image of size 500×500, the size of deblurred images by FBC, FBCW, and FBCWP is 500×500, while the size of deblurred images by RBC and ABC is 490×490. For fair visual comparison, however, our figures present image results from FBC, FBCW, and FBCWP after removing unseen image pixels.

4.1 Summary of claims and supporting figures

In this section we give a list of our claims with the figures that support them.

- The proposed method FBCWP removes boundary artifacts better than the other methods (RBC, ABC, FBC) when images are blurred with a uniform point spread function. Figures 4 and 5 support this claim.

- While FBCWP removes boundary artifacts better than the other methods when images are blurred with a Gaussian point spread function, the differences, while noticeable, are not as large as with uniform blurring. Figures 7 and 8 support this claim.

- Figures 10 and 11 show that FBCWP works significantly better than the other methods when the point spread function is a one-dimensional diagonal gradient blur.

- Boundary artifacts can occur even when there are no "unseen" pixels across the boundary (pixels on $\Omega - \Lambda$). The bottom and right boundaries of the images in Figures 10 and 11, which are blurred with a diagonal gradient point spread function, illustrate this claim.

- Our proposed method FBCWP removes boundary artifacts better than the other methods over a wide range of regularization parameters λ. Figures 4, 12, and 13 illustrate the truth of this claim for uniform blurring.

- Preconditioning by $\mathcal{W}$ is essential—using a weighted norm is not enough to achieve good boundary artifact removal. Figure 14 supports this claim.

- Our FBCWP method recovers certain blurred, damaged images better than previous methods that use median filtering to recover missing data pixels before deblurring. Figures 15 and 16 illustrate this point.

- The FBCWP method can recover blurred images with salt-and-pepper noise. This is shown in Figures 17 and 18.

The following sections discuss these claims in more detail.

4.2 Reduction of boundary artifacts

In this simulation, we used $\lambda = 0.001$ as the regularization parameter for all PSFs. We will discuss simulation results with other regularization parameters in Section 4.3.

4.2.1 Uniform blurring

Figure 3(b) shows a noisy blurred image $\mathbf{g} = \mathcal{T}\mathbf{f} + \mathbf{n}$, where $\mathcal{T}$ is the blurring transform defined by the 11×11 uniform PSF, $\mathbf{f}$ is the true image in Figure 3(a), and $\mathbf{n}$ is the Gaussian noise in (36). Figure 4 shows deblurred images by RBC, ABC, FBC, and FBCWP, using the image in Figure 3(b) as input. In this simulation, RBC, ABC, FBC, and FBCWP attain their RSE minimums, 1.16%(43), 1.13%(89), 1.06%(57), and 0.98%(53), respectively. Here the number in the parenthesis is the iteration number that attains the smallest RSE for each method.

Figure 4 shows that all methods (RBC, ABC, FBC, and FBCWP) produce almost identical results in the center part in deblurred images, but they are very different in boundary artifact removal. The boundary artifacts in RBC (Figure 4(a)) are less annoying than those in ABC (Figure 4(b)); they appear only in some parts, while the boundary artifacts in ABC appear all over image pixels near boundaries. A similar phenomenon also holds for FBC (Figure 4(c)). On the other hand, FBCWP does not show any noticeable

boundary artifacts. See zoomed images in Figure 5 for detailed comparison.

In our simulation, the trends described in this section also held for other test images; FBCWP outperformed RBC, ABC, and FBC objectively, by having smaller RSE than RBC, ABC, and FBC, and subjectively, by not showing boundary artifacts for all test images.

4.2.2 Gaussian blurring

Figure 6 shows a noisy blurred image $\mathbf{g}$, where the blurring is computed by a 17×17 Gaussian PSF with standard deviation equal to the width of 3 pixels.

Figure 7 shows deblurred images by RBC, ABC, FBC, and FBCWP, from the image in Figure 6. In this simulation, RBC, ABC, FBC, and FBCWP attain their RSE minimums, 1.29%(82), 1.49%(99), 1.29%(90), and 1.22%(77), respectively. Here the number in the parenthesis is, again, the iteration number that attains the smallest RSE for each method.

In this simulation, all methods produce almost identical results in the center part in deblurred images, and they show some difference in boundary artifact removal. Again, FBCWP does not show any noticeable boundary artifacts, while RBC, ABC, and FBC show mild boundary artifacts. Unlike the simulation with the uniform blurring, however, the boundary artifacts in RBC, ABC, and FBC do not propagate into center parts of deblurred images. Zoomed images in Figure 8 show detailed comparison.

In our simulation, the trends described in this section also held for other test images; RBC, ABC, and FBC do not severely suffer from boundary

artifacts under Gaussian blurring in the sense that those non-propagating boundary artifacts can easily be removed by cropping out a few rows and columns of image pixels near the boundary, while FBCWP does not show any sign of boundary artifacts.

We speculate that boundary artifacts do not propagate to the interior of the image when using RBC and ABC in this example because of some special property of Gaussian blurring, perhaps because the Gaussian PSF we use decays quickly away from its center. We also note that FBC does not propagate boundary artifacts under Gaussian blurring, even though it does not impose any boundary conditions.

4.2.3 Diagonal gradient blurring

Figure 9 shows a noisy blurred image $\mathbf{g} = \mathcal{T}\mathbf{f} + \mathbf{n}$, where the blurring transform $\mathcal{T}$ is computed by the 11×11 diagonal gradient PSF represented by a matrix whose main diagonal elements are $\{\frac{30}{275}, \frac{29}{275}, \ldots, \frac{20}{275}\}$, where the first diagonal element $\frac{30}{275}$ is $k_{0,0}$.

Figure 10 shows deblurred images by RBC, ABC, FBC, and FBCWP, from the image in Figure 9. In this simulation, RBC, ABC, FBC, and FBCWP attain their RSE minimums, 0.78%(20), 0.46%(155), 0.45%(38), and 0.31%(34), respectively. Here the number in parentheses is, again, the iteration number that attains the smallest RSE for each method.

RBC (Figure 10(a)) shows boundary artifacts in some regions near the left boundary and at the whole region along the lower and right boundaries. Boundary artifacts near the lower and right boundaries in Figure 10(a) form

straight lines that are parallel to the boundary line. Similar pattern are also shown near the lower and right boundaries in ABC (Figure 10(b)) and FBC (Figure 10-c), but they are not as severe as those in RBC (Figure 10(a)). On the other hand, FBCWP does not show any sign of boundary artifacts. Zoomed images in Figure 11 show detailed comparison.

The assumption that the first diagonal element of $\mathbf{k}$ is $k_{0,0}$ indicates that $\Omega = \{(j_1, j_2) \mid -10 \leq j_\nu < 490,\ \nu = 1, 2\}$ and $\Lambda = \{(i_1, i_2) \mid 0 \leq i_\nu < 490,\ \nu = 1, 2\}$, and hence there are no unseen image pixels across the lower and right boundaries in Figure 9, while ten rows and ten columns of image pixels are across the upper and left boundaries, respectively. Based on this observation, one might expect that RBC, ABC, and FBC would not suffer from artifacts near the lower and right boundaries, since there are no unseen image pixels across the lower and right boundaries. The simulation results in this section, however, clearly show that such an expectation would be wrong. In other words, the existence of unseen image pixels is not the source of boundary artifacts.

Again, the trends described in this section held for other test images in our simulation. In all test images, RBC, ABC, and FBC suffered from propagating boundary artifacts, while FBCWP showed no sign of boundary artifacts. Moreover, in all test images, FBCWP outperformed RBC, ABC, and FBC in the RSE criterion.

4.3 Selection of regularization parameter

The simulation results in Section 4.2 showed that, when the fixed regularization parameter $\lambda = 0.001$ was used, FBCWP outperformed RBC, ABC, and FBC in deblurring of uniform, Gaussian, and diagonal gradient PSFs in the presence of Gaussian noise $\mathbf{n}$. In the previous section, we used $\lambda = 0.001$ as the single regularization parameter for all methods (RBC, ABC, FBC, and FBCWP), despite the difference in PSFs, test images, and methods, by noting that $\lambda = 0.001$ produced the smallest RSE result among various λ's (0.0, 0.1, 0.01, 0.001, 0.0001, 0.00001) for RBC, ABC, FBC, and FBCWP, in the simulation with the true image, "Airfield"(Figure 3(a)), the 11×11 uniform PSF, and the Gaussian noise $\mathbf{n}$ in (36). Since the optimal regularization parameter depends on the true image $\mathbf{f}$, the PSF $\mathbf{k}$, the method, and the noise $\mathbf{n}$, one might suspect that different λ's would give a chance for RBC, ABC, or FBC to outperform FBCWP. The simulation results, however, show that this does not happen.

In our simulation studies, for any reasonable choice for the regularization parameter λ, FBCWP outperformed RBC, ABC, and FBC without exceptions in test images or blurring PSFs, by having the smallest RSE and not showing any sign of boundary artifacts. Moreover, the regularization parameter λ did not give noticeable differences in RBC, ABC, FBC, and FBCWP. These are well expected results, since RBC, ABC, FBC, and FBCWP are virtually identical CG iterations; the only differences are made on ways of treating image pixels near boundaries. Based on this argument, we can conclude that the superior performance of FBCWP over RBC, ABC, and FBC,

which was shown in simulations with $\lambda = 0.001$ as the regularization parameter, also holds for any other reasonable regularization parameters.

To support this claim, we present Figures 12 and 13, which show deblurred images from Figure 3(b), with regularization parameters $\lambda = 0.0001$ and $\lambda = 0.01$, respectively. Again, we selected deblurred images which attains the smallest RSE for given methods. The results in Figures 12 and 13 show that FBCWP outperforms RBC, ABC, and FBC with both rather small and large regularization parameters, and the regularization parameter give almost identical effect on each method.

4.4 Effect of preconditioning

Up to now, we did not present results by FBCW and FBCP. The excuse for doing so is that results by FBCW and FBCP were almost identical to results by FBC and FBCWP, respectively. To verify this, we conducted a deblurring simulation by FBCW and FBCP.

Figure 14(b) shows the deblurred image by FBCW with $\lambda = 0.001$, from Figure 9 (blurred by the 11×11 diagonal gradient PSF). The result is almost identical to Figure 14(a), which is obtained by FBC with $\lambda = 0.001$. Figure 14(c) shows the deblurred image by FBCP with $\lambda = 0.001$ from Figure 9. The result is almost identical to Figure 14(d), which is obtained by FBCWP with $\lambda = 0.001$.

The comparison between Figures 14(a) and 14(b) shows that the use of the weighted norm $L^2(\Omega, \mathbf{w})$ is not sufficient to remove boundary artifacts. The

comparison between Figures 14(b) and 14(d) confirms our earlier mention that FBCW and FBCWP produce different results, even though they are based on two equivalent linear systems (9) and (11), due to incomplete CG iterations in FBCW and FBCWP.

Notice that FBCP and FBCWP are preconditioned FBC and FBCW, respectively, by $\mathcal{W}$. The comparison between Figures 14(a) and 14(c), and 14(b) and 14(d) shows that the use of the preconditioning by $\mathcal{W}$ removes boundary artifacts. The difference between FBCP and FBCWP is hardly noticeable.

The results in this section show that in removing boundary artifacts, the use of the preconditioner $\mathcal{W}$ is the most essential factor.

4.5 Applications of free boundary conditions

As mentioned in Section 3.1, the proposed method, FBCWP, can be used for arbitrarily-shaped images and it can recover unseen image pixels across the boundary. Combining these two advantages, we can apply FBCWP to other interesting applications in image deblurring.

In this paper, we consider the recovery of damaged parts in noisy blurred images as an application of FBCWP. Let $\mathbf{g}$ be a damaged version, as seen in Figure 15(a), of the noisy blurred image in Figure 3(b), i.e., the true image is sequentially corrupted by 11×11 uniform blurring, Gaussian noise $\mathbf{n}$ in (36), and damage. With the assumption that the damaged parts in Figure 15(a) are relatively small so that every image pixel in the damaged parts gives

some contribution to our observation, we can use FBCWP to recover image pixels in the damaged parts by treating them as unseen image pixels across the boundary. Figure 15(b) shows the recovered image by FBCWP.

This approach is different from methods that are commonly used in so called *inpainting* applications, in which image pixels in damaged parts are filled up possibly by applying a smoothing transform that resembles a diffusion process[2]. On the other hand, FBCWP treats damaged parts as unseen image parts and recovers them, as a result of deblurring, by using the hidden information (of damaged parts) in undamaged image pixels through blurring.

RBC and ABC cannot be applied to this problem, since RBC and ABC are virtually limited to rectangular images only, and they do not attempt to recover unseen image pixels across the boundary. Another possible approach might take the process of recovering damaged image pixels first and the processing of deblurring after that. Figure 15(c) shows the result of recovering damaged image pixels by applying three-round 3×3 median filtering on Figure 15(a), and Figure 15(d) shows the result of deblurring by FBCWP from Figure 15(c). Even though Figure 15(c) is almost identical to Figure 3(b), Figure 15(d), deblurred from Figure 15(c), suffers from artifacts all over the image, while Figure 15(b), directly deblurred from Figure 15(a), does not show any noticeable artifacts. Zoomed images in Figure 16 show detailed comparisons.

The success of FBCWP in the recovery of damaged pixels can be extended to image recovery in the presence of 'salt and pepper' noise. Figure 17(a) shows the observed image, which is blurred by 11×11 uniform PSF, and

corrupted by salt and pepper noise, where a randomly selected subset of 60% of the image pixels change to black or white pixels. In other words, the Gaussian noise in Figure 3(b) is replaced by salt and pepper noise in Figure 17(a). Figure 17(b) shows the deblurred image by FBCWP from Figure 17(a), by regarding salt and pepper noised image pixels, which can be easily detectable by checking intensities at image pixels, as unseen image pixels across the boundary. Figure 17(b) was obtained at the iteration 99 with RSE = 1.0%. This result is compatible with the result in Figure 4(d), which is obtained at iteration 53 with RSE = 0.98%, even though Figure 17(b) used only 40% of of the pixels in the observed image. Zoomed images in Figure 5, Figure 8, and Figure 18 show detailed comparisons.

5 Conclusion and Discussion

In this paper we propose using free boundary conditions, which do not impose any restrictions on unseen image pixels, and the preconditioned CG method, where the preconditioner is designed to compensate for the non-uniformity in contributions from image pixels to the observation, in image deblurring. In simulation studies with uniform, Gaussian, and diagonal gradient PSFs, the proposed method, FBCWP, outperforms RBC (the reflective boundary condition–based CG method) and ABC (the anti-reflective boundary condition–based CG method) in all test images objectively, by having smaller RSE, and subjectively, by not showing boundary artifacts. Simulation results in Section 4.5 show that FBCWP can be used for the recovery of damaged regions in noisy blurred images by treating damaged regions as

unseen image pixels across the boundary. Based on these simulation results, we can conclude that FBCWP is more efficient in removing boundary artifacts, more flexible in dealing with boundaries, and more applicable in image deblurring than RBC and ABC.

Simulation results in this paper show that the non-uniformity in contributions from image pixels to the observation, instead of the existence of unseen image pixels across the boundary, is the main source of boundary artifacts in image deblurring. Therefore, the use of the preconditioner $\mathcal{W}$ (34) exactly as suggested in FBCWP is an essential step in removing boundary artifacts.

The non-negativity of the PSF is an essential requirement for the success of FBPWP. For example, if the PSF is not non-negative, then the preconditioner $\mathcal{W}$ (34) may not be defined, since the weight $(\mathcal{T}^*\mathbf{I}_\Lambda)_{i_1,i_2}$ in (33) could be zero for some image pixel (i_1, i_2). This would violate the invertibility of the preconditioner $\mathcal{W}$ in (34).

Acknowledgments

The work of the first author was supported by the Inje Research and Scholarship Foundation in 2012 and conducted while the first author was a visiting scholar at the Center for Nonlinear Analysis, Department of Mathematical Sciences, Carnegie Mellon University. The work of the second author was supported in part by grants from the Simons Foundation (Awards #209418 and #229816).

References

[1] Aghdasi F and Ward R K 1996 Reduction of boundary artifacts in image restoration *IEEE Trans. Image Processing* **5**(4) 611–618

[2] Bertalmio M, Sapiro G, Caselles V, and Ballester C 2000 Image Inpainting *SIGGRAPH*, 417–424

[3] Bertero M and Boccacci P 2005 A simple method for the reduction of boundary effects in the Ricardson-Lucy approach to image deconvolution *Astron. Astrophys.* **437** 369–374

[4] Calvetti D, Kaipio J P, and Someralo E 2006 Aristotelian prior boundary conditions *Inter. J. Mathematics and Computer Science* 63–81

[5] Donatelli M, Estatico C, Martinelli A, and Serra-Capizzano S 2006 Improved image deblurring with anti-reflective boundary conditions and re-blurring *Inverse Problems* **22**(6) 2035–2053

[6] Fan Y W and Nagy J G 2009 Synthetic boundary conditions for image deblurring *Linear Algebra Appl.* doi:10.1016/j.laa.2009.12.021

[7] Groetsch C W 1984 *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind* (Pitman, Boston)

[8] Hestenes M R and Stiefel E 1952 Methods of conjugate gradients for solving linear systems *Journal of Research of the National Bureau of Standards*, **49**(6) 409–436

[9] Jain A K 1989 *Fundamentals of digital image processing* (Prentice-Hall, Englewood Cliffs, NJ)

[10] Lucy L B 1974 An iterative techniques for the rectification of observed distributions *Astronomical Journal* **79**(6) 745–754

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

[12] Richardson W H 1972 Bayesian-based iterative method of image restoration *J. Opt. Soc. Am.* **62**(1) 55–59

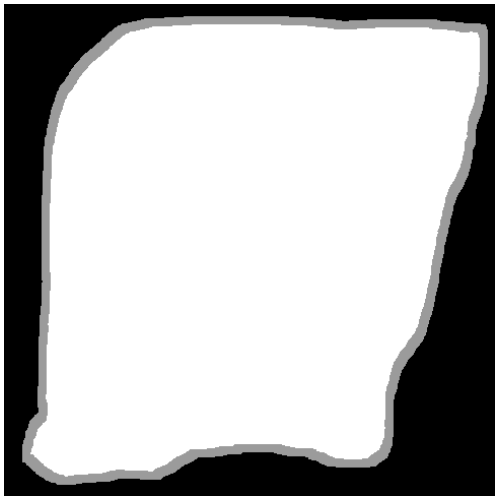
[13] Saad Y 2003 *Iterative methods for sparse linear systems* 2nd ed. (SIAM Publications, Philadelphia)

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

[15] Tekalp A M and Sezan M I 1990 Quantitative analysis of artifacts in linear space-invariant image restoration *Multidimensional Syst. Signal Processing* **1** 143–177

[16] Woods J W, Biemond J, and Kekalp A M 1985 Boundary value problem in image restoration in *Proc. Sixth Int. Conf. Acoust. Speech Signal Processing* 18.11.1–18.11.4

(a) Blurred image with a non-rectangular boundary



(b) Images pixels on Λ (white) and $\Omega - \Lambda$ (gray)

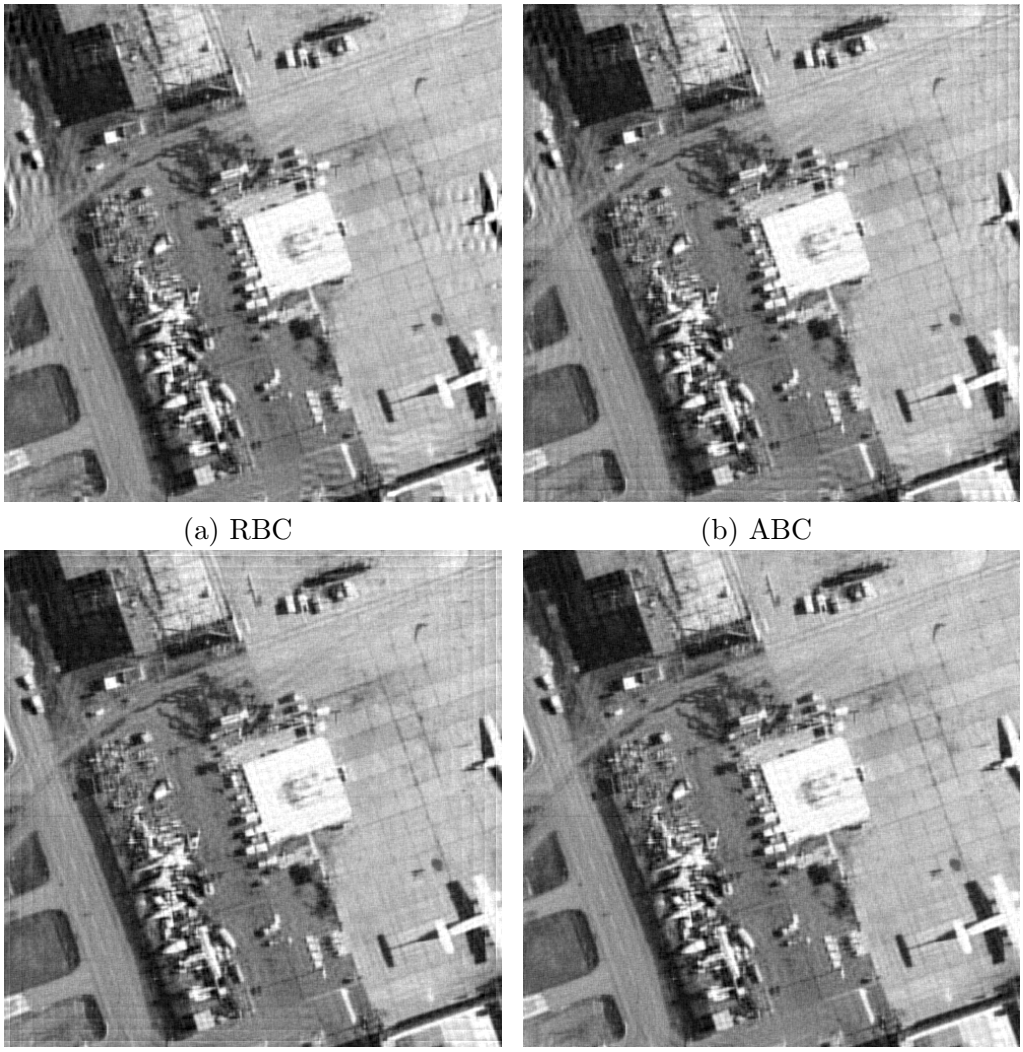
Figure 1. (a) A blurred image with a non-rectangular boundary. Here the blurring is generated by a 17×17 Gaussian PSF with standard deviation equal to the width of 3 pixels. The black background represents the region where no observation is available. (b) The set Ω of image pixels that can contribute to the observation through the 17×17 Gaussian PSF. The set Ω consists of two regions; observed image pixels, Λ, represented in white, and the unseen image pixels across the boundary, $\Omega - \Lambda$, represented in gray. The border line between white and gray colored regions is the boundary of the observed image.

(a) FBC (b) FBCWP

Figure 2. (a) The deblurred image by FBC from Figure 1(a). Those ripples are propagating into the center area as the iteration proceeds. (b) The deblurred image by FBCWP from Figure 1(a). The deblurred image shows no boundary artifacts at all. This result shows that FBCWP can deblur arbitrarily shaped images, without causing boundary artifacts.

(a) True image (b) Blurred by uniform PSF

Figure 3. (a) The true "Airfield" image $\mathbf{f}$ of size 500×500. (b) The observed image $\mathbf{g} = \mathcal{T}\mathbf{f} + \mathbf{n}$, where $\mathcal{T}$ is defined by the 11×11 uniform PSF $\mathbf{k}$ as in (2) and $\mathbf{n}$ is Gaussian noise defined in (36). The size of $\mathbf{g}$ is of 490×490 pixels.

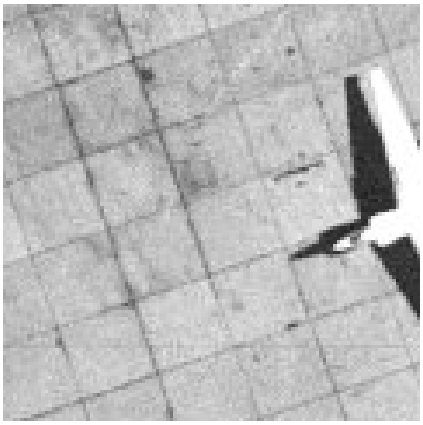


(a) RBC (b) ABC

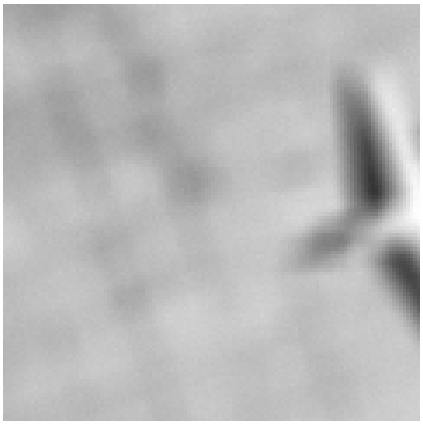
(c) FBC (d) FBCWP

Figure 4. Deblurred images from Figure 3(b). All images are of size 490×490. (a) The deblurred image by RBC, with RSE = 1.16% at the 43rd iteration. RBC suffered from boundary artifacts in the region where the reflected boundary did not provide a sufficient similarity in image pixels across the boundary. (b) The deblurred image by ABC, with RSE = 1.13% at the 89th iteration. ABC suffered from boundary artifacts at all regions

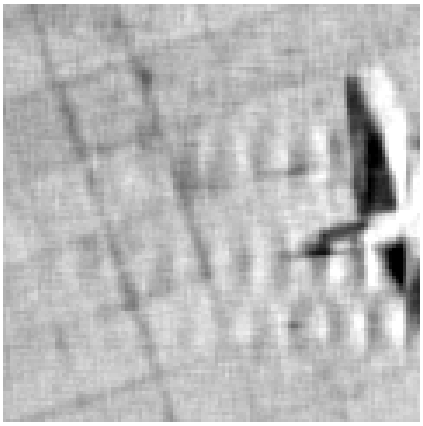
near boundary. (c) The deblurred image by FBC, with RSE = 1.06% at the 57th iteration. FBC suffered from boundary artifacts at all regions near boundary. (d) The deblurred image by FBCWP, with RSE = 0.98% at the 53rd iteration. FBCWP did not show any noticeable boundary artifacts.



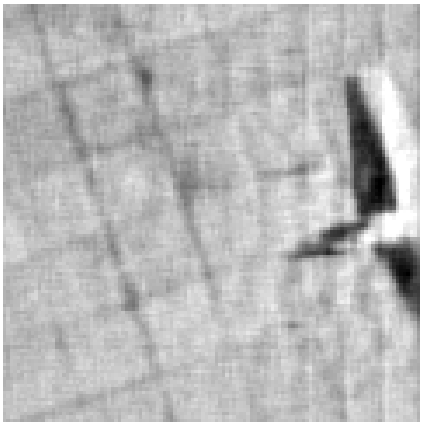
(a) True image



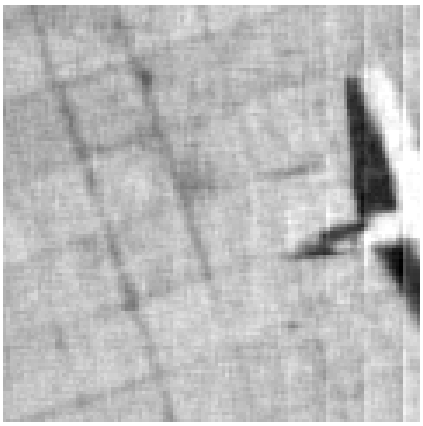
(b) Blurred by uniform PSF



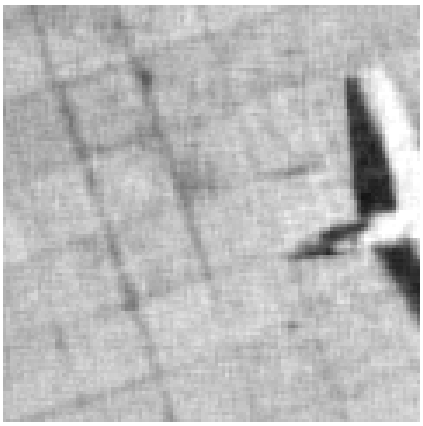
(c) RBC



(d) ABC



(e) FBC

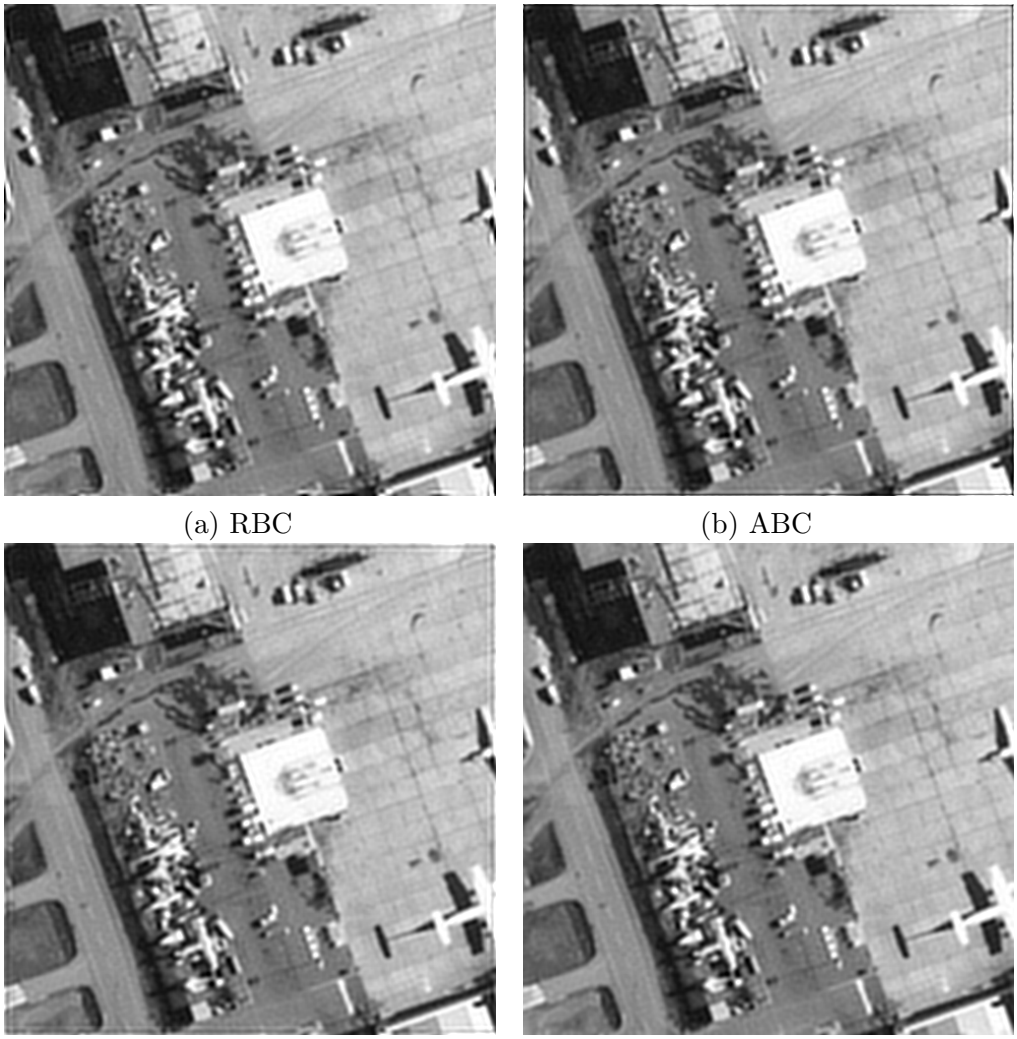


(f) FBCWP

Figure 5. The zoomed parts of the "Airfield" images. All images are of size 120×120. In deblurred images by RBC, ABC, and FBC, propagating boundary artifacts appear, while no boundary artifacts appear in FBCWP. (a) A part of the true image in Figure 3(a). (b) A part of the observed image in Figure 3(b). (c) A part of the deblurred image by RBC in Figure 4(a). (d) A part of the deblurred image by ABC in Figure 4(b). (e) A part of the deblurred image by FBC in Figure 4(c). (f) A part of the deblurred image by FBCWP in Figure 4(d).

Blurred by Gaussian PSF

Figure 6. The observed image $\mathbf{g}$, which is blurred by a 17×17 Gaussian PSF with standard deviation equal to the width of 3 pixels and corrupted by Gaussian noise $\mathbf{n}$ defined in (36). The size of the observed image $\mathbf{g}$ is of 484×484.

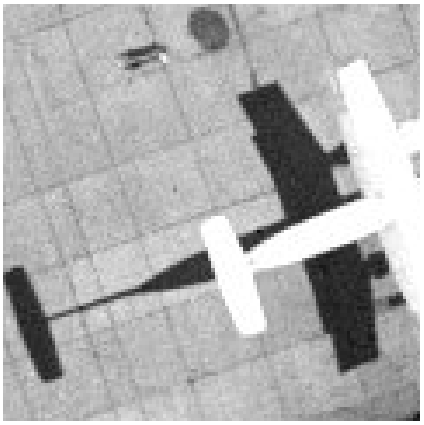


(a) RBC (b) ABC

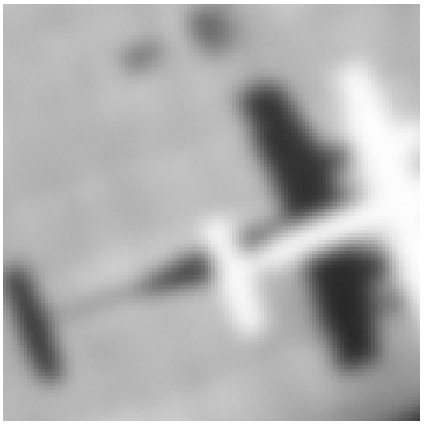
(c) FBC (d) FBCWP

Figure 7. Deblurred images from Figure 6. All images are of size 484×484. (a) The deblurred image by RBC, with RSE = 1.29% at the 82nd iteration. RBC generated mild non-propagating boundary artifacts in the region where the reflected boundary was not similar to the true image pixels across the boundary. (b) The deblurred image by ABC, with RSE = 1.49% at the 99th iteration. ABC generated mild non-propagating boundary artifacts in all

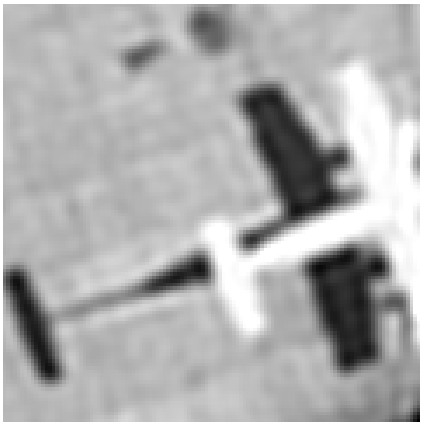
regions near boundary. (c) The deblurred image by FBC, with RSE = 1.29% at the 90th iteration. FBC showed a similar result as ABC. (d) The deblurred image by FBCWP, with RSE = 1.22% at the 77th iteration. FBCWP did not show boundary artifacts.



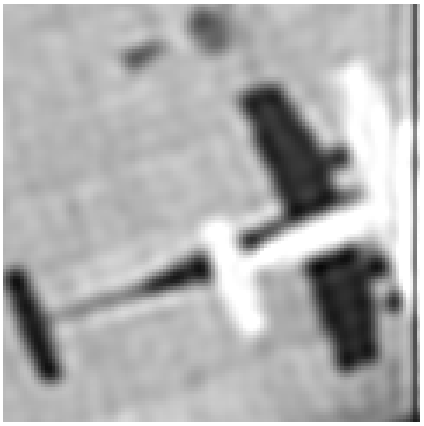
(a) True image



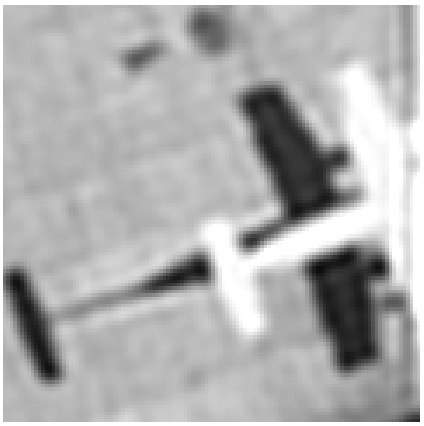
(b) Blurred by Gaussian PSF



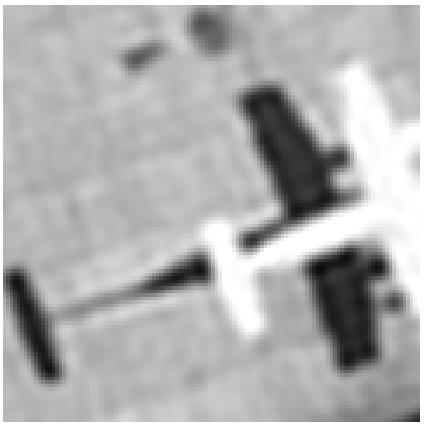
(c) RBC



(d) ABC



(e) FBC

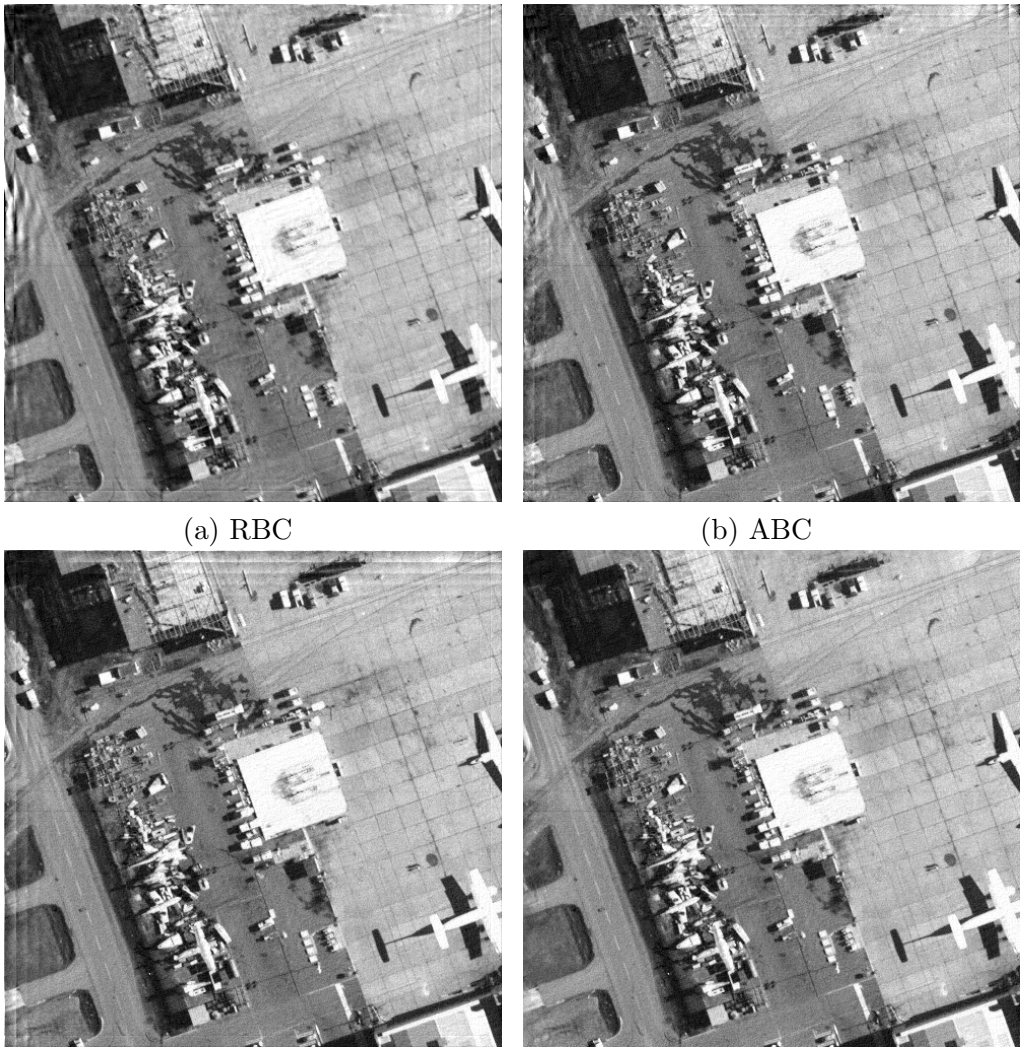


(f) FBCWP

Figure 8. The zoomed parts of the "Airfield" images. All images are of size 120×120. In deblurred images by RBC, ABC, and FBC, non-propagating boundary artifacts appear, while no boundary artifacts appear in FBCWP. (a) A part of the true image in Figure 3(a). (b) A part of the observed image in Figure 6. (c) A part of the deblurred image by RBC in Figure 7(a). (d) A part of the deblurred image by ABC in Figure 7(b). (e) A part of the deblurred image by FBC in Figure 7(c). (f) A part of the deblurred image by FBCWP in Figure 7(d).

Blurred by diagonal gradient PSF

Figure 9. The observed image $\mathbf{g} = \mathcal{T}\mathbf{f} + \mathbf{n}$, where $\mathcal{T}$ is defined by an 11×11 diagonal gradient PSF with diagonal elements $\{\frac{30}{275}, \frac{29}{275}, \ldots, \frac{20}{275}\}$ as in (2), where the first diagonal element $\frac{30}{275}$ is $k_{0,0}$, and $\mathbf{n}$ is Gaussian noise defined in (36). The size of the observed image $\mathbf{g}$ is of 490×490.

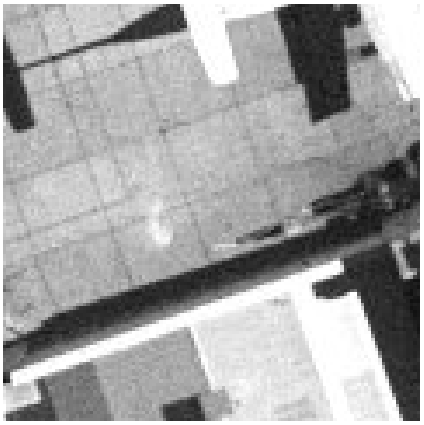


(a) RBC (b) ABC

(c) FBC (d) FBCWP

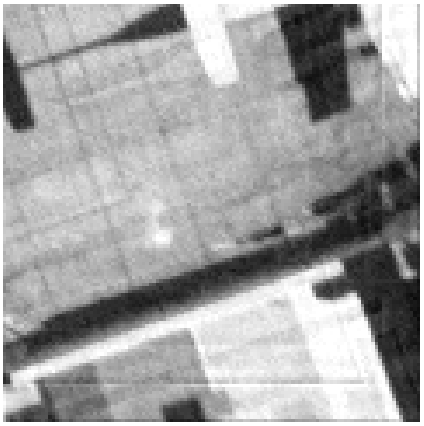
Figure 10. Deblurred images from Figure 9. All images are of size 490×490. (a) The deblurred image by RBC, with RSE = 0.78% at the 20th iteration. RBC generated propagating boundary artifacts at some part near the left boundary an propagating boundary artifacts in all regions near the lower and right boundaries. (b) The deblurred image by ABC, with RSE = 0.46% at the 155th iteration. ABC generated propagating boundary artifacts

at some part near the left boundary and propagating boundary artifacts in all regions near the lower and right boundaries, as RBC did. The severity of boundary artifacts in ABC is weaker than that of RBC, however. (c) The deblurred image by FBC, with RSE = 0.45% at the 38th iteration. FBC generated propagating boundary artifacts in all regions near all boundaries. Boundary artifacts near the upper and left boundaries look more severe than those near the lower and right boundaries. (d) The deblurred image by FBCWP, with RSE = 0.31% at the 34th iteration. FBCWP did not show boundary artifacts.

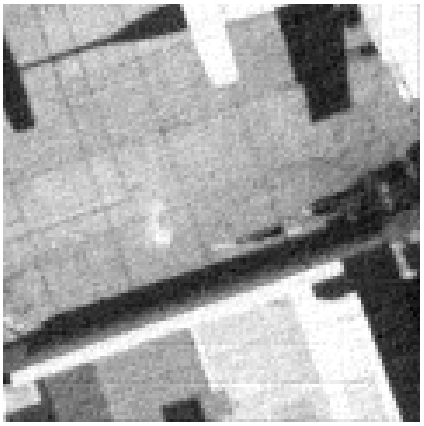


(a) True image

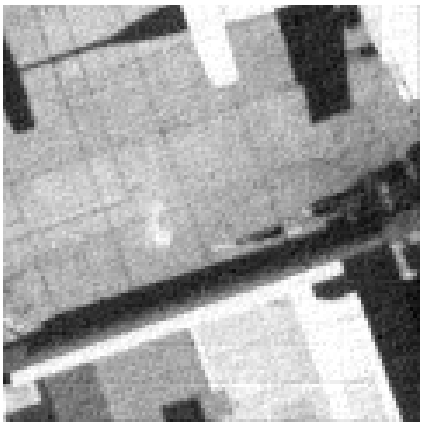
(b) Blurred by diagonal PSF



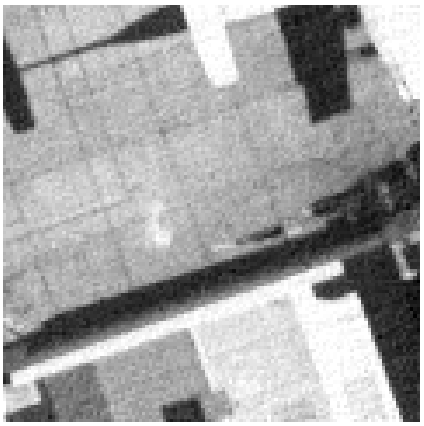
(c) RBC



(d) ABC

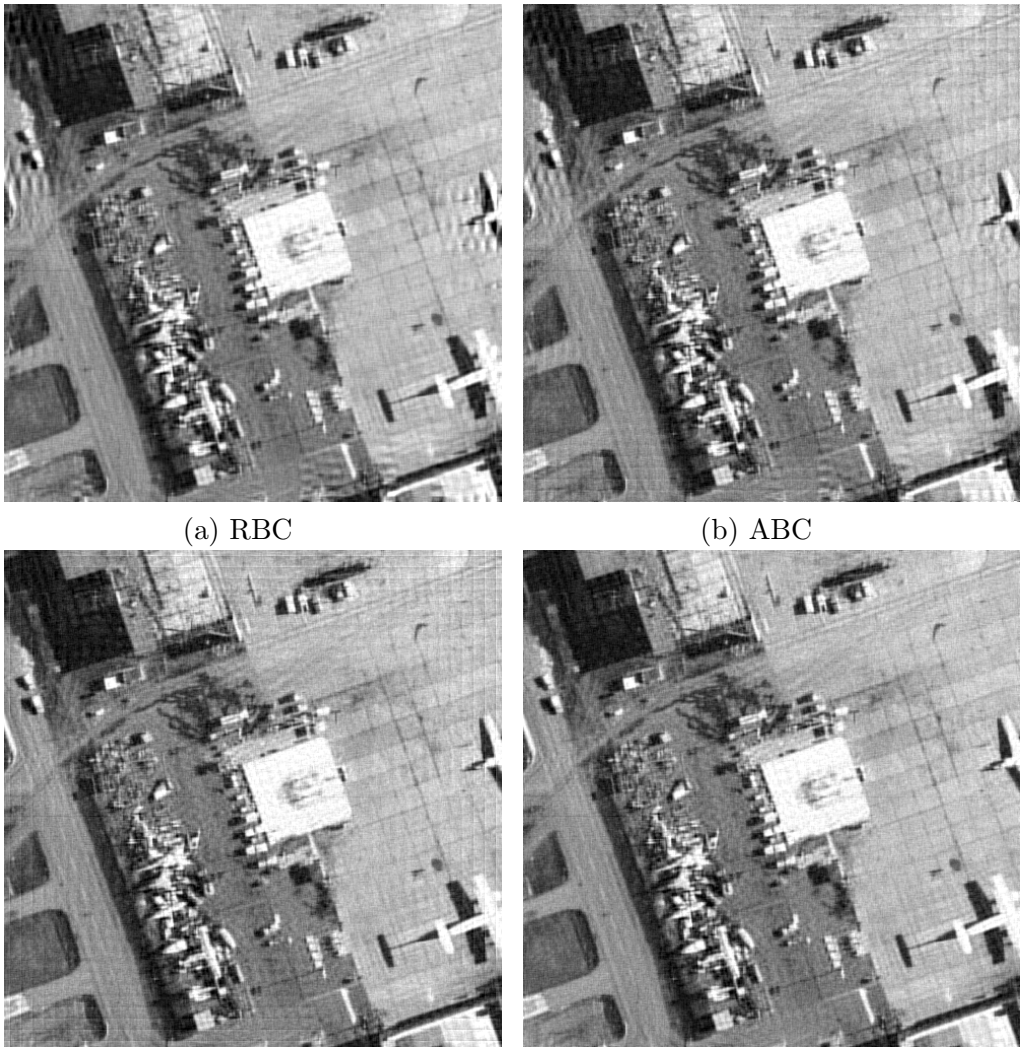


(e) FBC



(f) FBCWP

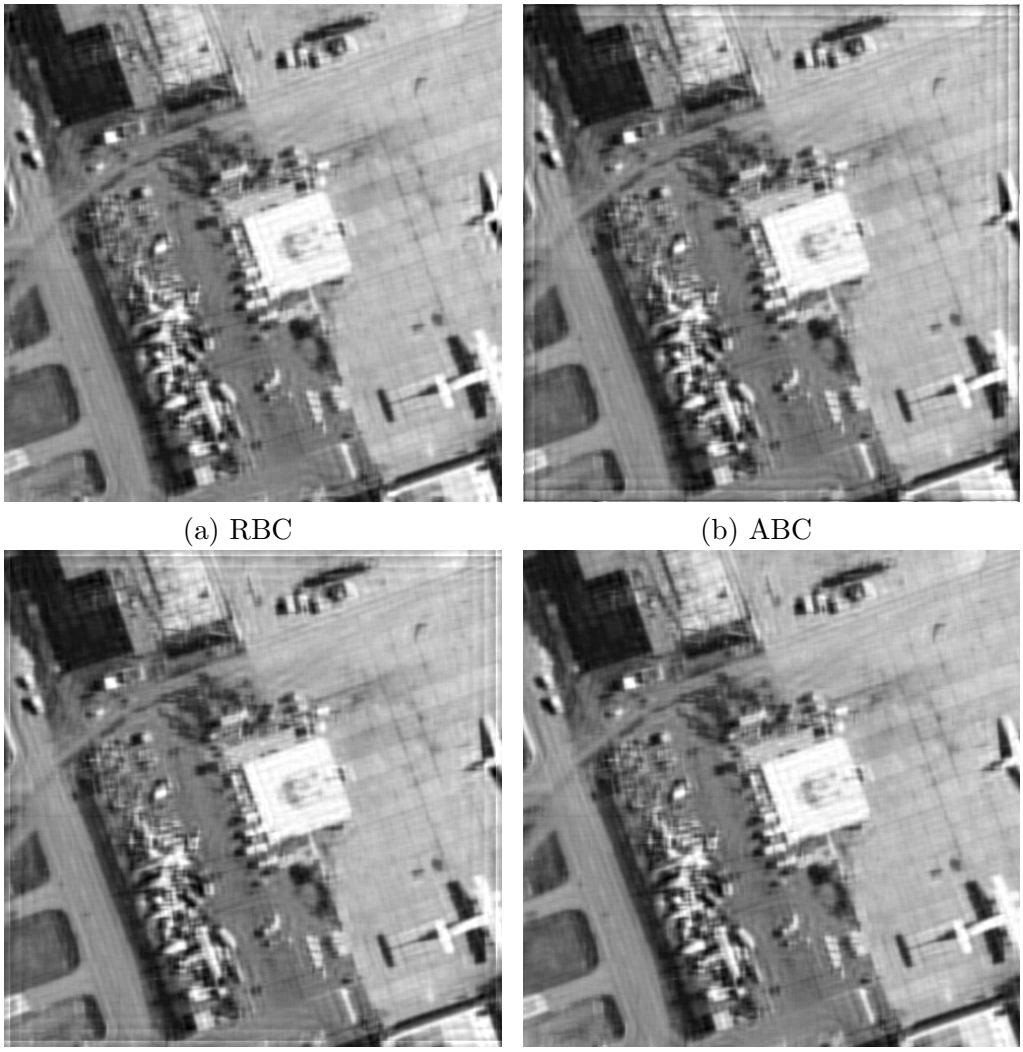
Figure 11. The zoomed parts of 'Airfield' images. All images are of size 120×120. (a) A part of the true image in Figure 3(a). (b) A part of the observed image in Figure 9. (c) A part of the deblurred image by RBC in Figure 10(a). (d) A part of the deblurred image by ABC in Figure 10(b). (e) A part of the deblurred image by FBC in Figure 10(c). (f) A part of the deblurred image by FBCWP in Figure 10(d).



(a) RBC (b) ABC

(c) FBC (d) FBCWP

Figure 12. Deblurred images from Figure 3(b) with $\lambda = 0.0001$ as the regularization parameter. (a) The deblurred image by RBC, with RSE = 1.2% at the 29th iteration. (b) The deblurred image by ABC, with RSE = 1.14% at the 64th iteration. (c) The deblurred image by FBC, with RSE = 1.07% at the 41th iteration. (d) The deblurred image by FBCWP, with RSE = 0.99% at the 38th iteration.

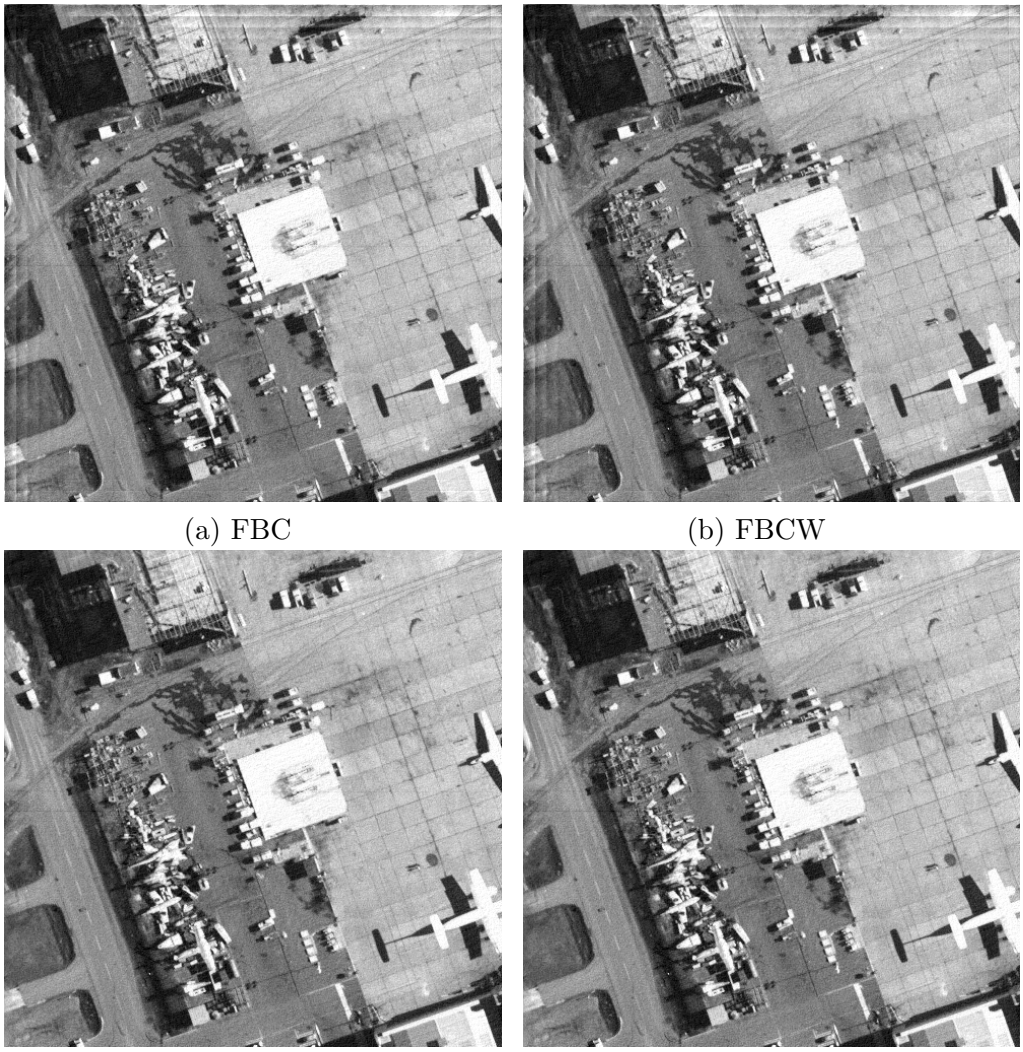


(a) RBC (b) ABC

(c) FBC (d) FBCWP

Figure 13. Deblurred images from Figure 3(b) with $\lambda = 0.01$ as the regularization parameter. (a) The deblurred image by RBC, with RSE = 1.46% at the 34th iteration. (b) The deblurred image by ABC, with RSE = 1.69% at the 73th iteration. (c) The deblurred image by FBC, with RSE = 1.54% at the 36th iteration. (d) The deblurred image by FBCWP, with RSE

$= 1.42\%$ at the 43th iteration.

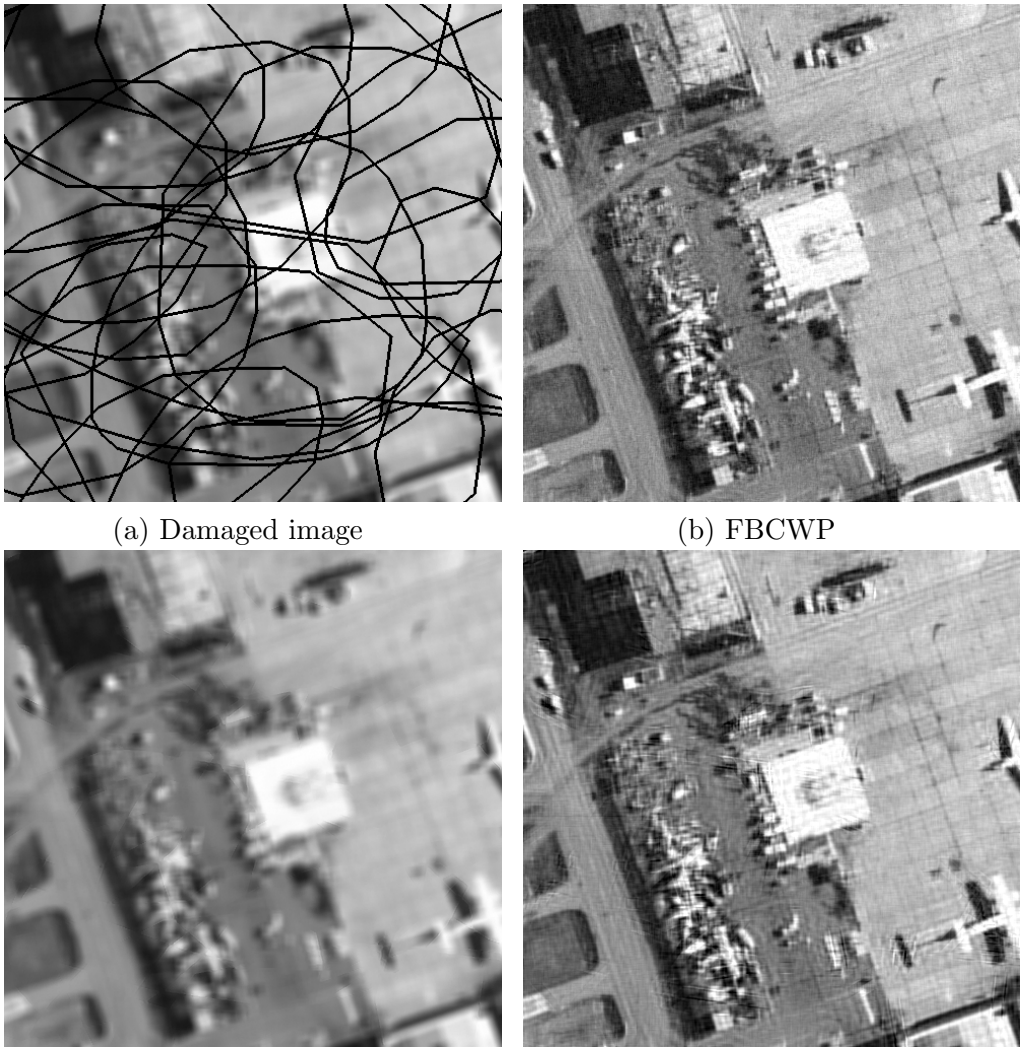


(a) FBC (b) FBCW

(c) FBCP (d) FBCWP

Figure 14. Deblurred images by FBC, FBCW, FBCP, and FBCWP from the image in Figure 9 (blurred by diagonal gradient PSF). Here results by FBC (Figure 10(c)) and FBCWP (Figure 10(d)) are presented again for better visual comparison. (a) Deblurred image by FBC. (b) Deblurred image by FBCW. This result is almost identical to that by FBC. (c) Deblurred image

by FBCP. This result is almost identical to that by FBCWP. (d) Deblurred image by FBCWP.

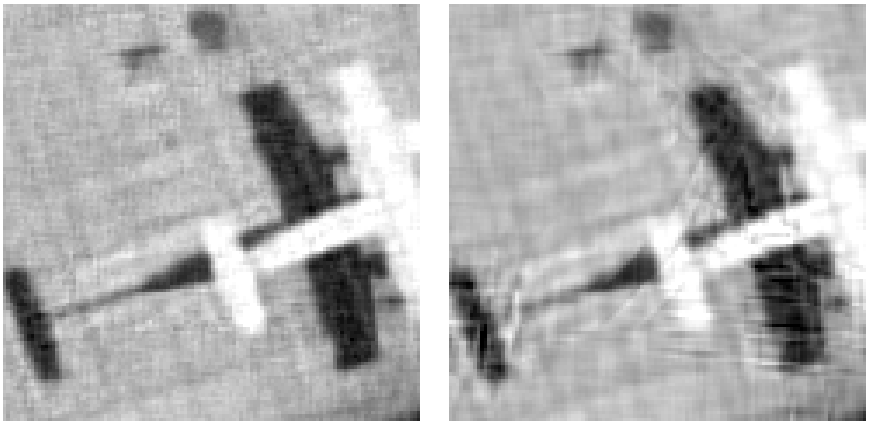


(a) Damaged image (b) FBCWP

(c) Median filtering (d) Median filtering + FBCWP

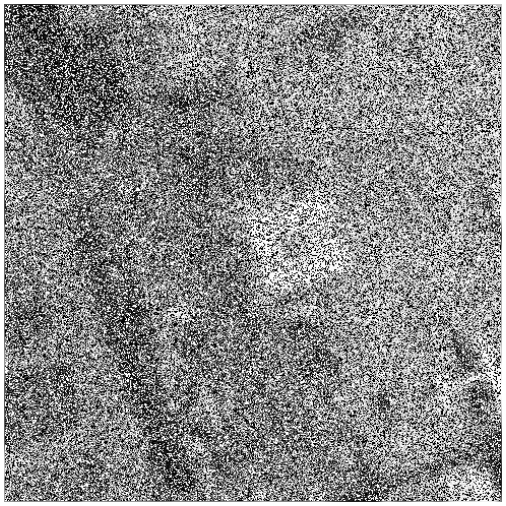
Figure 15. (a) The damaged version of the observed image in Figure 3(b). (b) The deblurred image by FBCWP from the damaged image, by regarding the damaged part, represented by black colored pixels in (a), as part of unseen image pixels across the boundary, i.e., the damaged image in (a) has non-rectangular inner boundaries. (c) The image obtained by applying three-round 3×3 median filtering on the damaged image in (a). Here the median

filtering is applied to the image pixel that is not determined in the previous round due to the absence of determined image pixels in 3×3 neighborhood. The median filtered image looks almost identical to the image in Figure 3(b). (d) The deblurred image by FBCWP from the median filtered image in (c). The comparison with the image in (b) shows that a small difference made by mis-filling in median filtering causes severe artifacts in deblurring.



(a) FBCWP (b) Median filtering + FBCWP

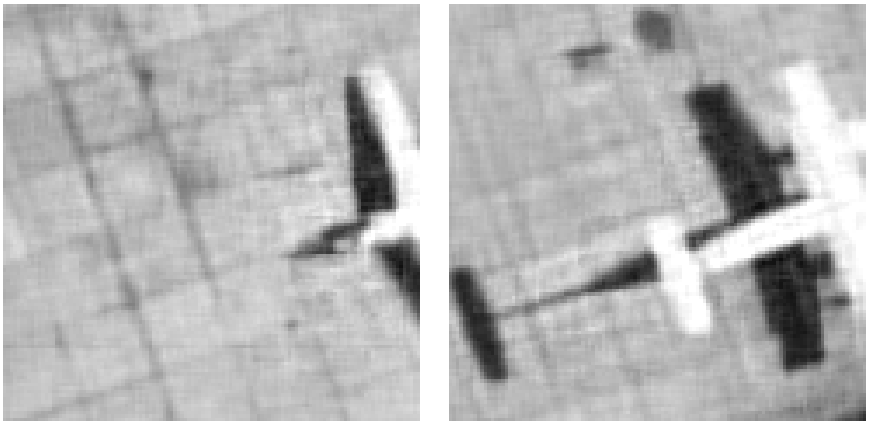
Figure 16. (a) A part of Figure 15(b). (b) A part of Figure 15(d).



(a) Corrupted by salt-and-pepper noise

(b) FBCWP

Figure 17. (a) The observed image $\mathbf{g}$, which is blurred by a 11×11 uniform PSF, and corrupted by "salt and pepper" noise, where 60% of pixels change to the darkest or brightest pixels. In other words, the Gaussian noise in Figure 3(b) is replaced by salt and pepper noise here. (b) The deblurred image by FBCWP from the heavily noised image in (a), by ignoring heavily noised image pixels. FBCWP deblurred the image with RSE = 1.0% at the 99-th iteration. The deblurred image is compatible with the image in Figure 4(d), even though the former only use 40% of the pixel data that is used in the latter.



(a) FBCWP (to be compared with Figure 5)

(b) FBCWP (to be compared with Figure 8)

Figure 18. Parts of Figure 17(b), which is deblurred by FBCWP from the "salt-and-pepper" image in Figure 17(a). (a) The comparison with images in Figure 5 shows that FBCWP can deblur the images using only with 40% of the observed pixels. (b) The comparison with images in Figure 8 shows that FBCWP can deblur the images using only with 40% of the observed pixels.