



Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A boundary condition based deconvolution framework for image deblurring[☆]

Xu Zhou, Fugen Zhou, Xiangzhi Bai ^{*}, Bindang Xue*Image Processing Center, Beijing University of Aeronautics and Astronautics, Beijing, 100191, China*

HIGHLIGHTS

- Various BC based deconvolution methods are re-derived.
- We propose a repeated BC based deconvolution method.
- The repeated BC can outperform undetermined BC in some cases.

ARTICLE INFO

Article history:

Received 12 May 2012

Received in revised form 14 October 2013

Keywords:

Deconvolution

Image deblurring

Boundary conditions

ABSTRACT

In image deconvolution, various boundary conditions (BC) based deconvolution methods have been proposed to reduce boundary artifacts. However, most of them are not considering the accuracy of BC due to computation limitation. In this paper, we propose a BC based deconvolution framework, which considers the convolution matrix as a product of partial convolution matrix and boundary condition matrix. By computing the adjoint matrix of boundary condition matrix, we can solve this large linear system with conjugate gradient algorithm. With this framework, we can easily derive two efficient non-blind image deconvolution algorithms, which treat the borders of image as repeated instances of the edge pixel values and unknown variables, respectively. Experiments on synthetic data and real data are both presented to show the performance of various BCs. Our conclusion is that undetermined BC usually has the best performance, and repeated BC outperforms undetermined BC if the latent image has high local similarity around the boundary.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In digital image processing, the general, discrete model for a linear degradation caused by blurring and additive noise can be given by the following superposition summation [1],

$$y(i, j) = \sum_k \sum_l h(i, j, k, l)x(k, l) + n(i, j), \quad (1.1)$$

where x is an original image, h is the point spread function (PSF) of imaging system, and y represents the degraded image which is acquired by the imaging system. In this formulation, n represents an additive noise introduced by the system, and is assumed to be a zero mean Gaussian distributed white noise. Let x and y be written as vectors, arranged in column

[☆] This work was supported by the National Natural Science Foundation of China (No. 61233005, 60832011), open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-12KF-04), Innovation Foundation of AVIC (No.CXY2010BH02) and Fundamental Research Funds for the Central Universities (Grant No.YWF-13-T-RSC-028, YWF-13-JQCJ-026).

* Corresponding author. Tel.: +86 10 82338578.

E-mail addresses: xuzhou@sa.buaa.edu.cn (X. Zhou), zhfugen@buaa.edu.cn (F. Zhou), jackybxz@buaa.edu.cn (X. Bai), xuebd@buaa.edu.cn (B. Xue).



Fig. 1. An example of boundary artifacts. (a) The original image with a field of view shown in the white box. (b) Blurred image using a uniform kernel of size 9×9 , where the missing boundary values are set to zeros. (c) Deblurred image with severe boundary artifacts.

lexicographic ordering, then (1.1) can be represented in terms of a matrix–vector formulation,

$$y = Hx + n, \quad (1.2)$$

where H represents the convolution matrix defined by the PSF.

Image deconvolution aims at recovering x from y . The problem is called blind deconvolution if both PSF and image are unknown, or non-blind deconvolution if only the image is unknown. Image deconvolution has been widely applied in many fields, such as astronomical imaging, remote sensing, medical imaging, and digital photography. In this paper, we focus on non-blind deconvolution problem.

Recent research shows that, for atmospherically degraded blur [2] and motion blur [3–5], the PSF can be approximated as a spatially invariant kernel function and hence, the image degradation model has the form of (see Eq. (2.2) in [6])

$$y(i, j) = h(i, j) * x^e(i, j) + n(i, j), \quad (1.3)$$

where $*$ denotes a partial 2-D convolution which computes only the pixels inside Field of View (FOV), and x^e denotes an expanded version of x with the boundary information outside FOV. Since the outside information of FOV is unavailable, some assumptions on the outside values are needed to estimate x . These assumptions are called the boundary conditions [6].

Due to lack of boundary information that is used to produce the blurred image, in most cases, it is impossible to estimate an accurate solution from the observation data. The missing boundary values usually cause serious ringing artifacts around the boundary of restored image, and would propagate it throughout the entire image if image boundary is not well treated. Fig. 1 shows an example of boundary artifacts with zero Dirichlet BC, which assumes the missing boundary values of x are zeros. To reduce boundary artifacts, a simple and effective way is to make the BC as accurate as possible. However, previous BCs, such as zero Dirichlet, periodic, reflective [6] and anti-reflective BC [7], are designed for computation purpose without fully considering the accuracy of BC, and therefore usually introduce boundary artifacts.

As pointed out in [8], the imposed BC has a substantial impact in two directions: (a) precision of the reconstruction especially close to boundaries (presence of ringing effects); (b) cost of computation for recovering the “true” image from a blurred one with or without noise. In this work, we propose a simple BC based deconvolution framework for image deblurring, which usually satisfies requirement (b) and therefore gives us more freedom to consider requirement (a). Unlike the previous algebraic frameworks [6,9,10,8] which represent the convolution matrix H of (1.3) as a summation of several square matrices, this framework simply represents the convolution matrix H of (1.3) as a product of two non-square matrices, a partial convolution matrix and a boundary condition matrix. Given a proper boundary condition, we can easily derive an iterative deconvolution algorithm that not only introduces very little boundary artifacts but also can be implemented efficiently. Incorporated with the state-of-the-art regularization priors, the resulting algorithms can yield a visually pleased image without noticeable noise and boundary artifacts.

1.1. Existing BCs

Generally, the missing boundary information can be estimated by extrapolating the available image data. Several extrapolation methods have been proposed by defining different BCs, including zero Dirichlet, periodic, improved periodic [11], reflective (also known as Neumann [6]), anti-reflective [7,12], and synthetic BC [9].

Reflective BC treats the scene outside the FOV as a mirror reflection of the scene inside the FOV, and it therefore preserves the continuities at the boundary (rigorously, there are two kinds of Reflective BC, one is half-sample symmetric BC [6], and the other is whole-sample symmetric BC [10]). However, reflective BC may make sense only when there are significant features that overlap the edge of the viewable region. Unlike reflective BC, anti-reflective BC preserves not only the continuity of image but also the continuity of normal derivative. In the case of reflective or anti-reflective BC, if the PSF is strongly symmetric, the resulting convolution matrix H can be diagonalized by 2D discrete cosine transform or 2D discrete sine transform, respectively. This good property allows efficient implementation of direct filtering type methods, such as spectral

filtering methods. However, we argue that these “reflective” boundary structures, which are designed for computation purpose, seldom exist in reality, and could often cause boundary artifacts.

Since circulant matrix can be diagonalized by FFT, periodic BC is very popular in the community of image deconvolution, in spite of its periodic structure rarely existing. Due to the efficiency of 2-D FFT, some fast non-blind deconvolution algorithms with total variation (TV) regularization [13] or Hyper-Laplacian prior [14] can be efficiently implemented. In order to reduce introducing boundary artifacts, Liu and Jia [11] establish new periodic BCs by smoothly expanding the input image to a large tile using a Gaussian filter. The large tile has periodic boundary structures and hence, this method introduces much less boundary artifacts than periodic BC. However, the expanded blurry image is not very accurate so visible boundary artifacts still exist. This method has been used in the non-blind deconvolution algorithm [4] (available at http://www.cse.cuhk.edu.hk/~leojia/projects/motion_deblurring/).

Recently, Fan and Nagy [9] propose a synthetic BC as well as a linear algebraic framework for BC based approaches. They consider the estimation of missing boundary information as an image inpainting or texture synthesis problem to enforce the outside boundary to continue the texture and edge. However, it is not easy to estimate the true relationship between miss boundary values and image edge from a blurred observation, so the resulting BC still may introduce boundary artifacts into the recovered image. In their linear algebraic framework, the convolution matrix H is decomposed into a summation of a block Toeplitz matrix T and a boundary condition matrix B which is defined by both the BC and the PSF. The structure of B is usually very complex, and loses the neat Kronecker product decompositions if the PSF is not separable. For more details about the structure of B , see [9].

In fact, instead of building a relationship between the missing boundary pixel values and the available image data, it has been found that a better way to reduce boundary artifacts is setting the miss boundary pixel values as unknown variables (see the MATLAB code provided by [15], available at <http://groups.csail.mit.edu/graphics/CodedAperture/>). We call this boundary treatment undetermined BC, and we will discuss this approach with more details in Section 3 and provide a simpler approach. In addition, recent studies [16,17] show that undetermined BC based model also can be efficiently solved by ADMM (Alternating Direction Method of Multipliers).

1.2. Regularization priors

An overview of classical methods on non-blind deconvolution can be found in [1], which contains Wiener filter, Kalman filter, and constrained least squares. A former algorithm, Richardson–Lucy [18,19], is very sensitive to noise, and usually produces many ringing artifacts around the borders. Recent approaches, which are under the regularization framework, make a great progress in handling with both ill-posedness and noise. These approaches can be divided into two categories. One is non-iterative approach [20], which shrinkages the coefficients of transform domain to suppress the leaked noise as well as the ringing artifacts. The other is iterative approach, which is supported on some form of a priori knowledge (under the form of priors or regularizers) about the original image to be estimated. Some of these methods, including wavelet-based priors/regularizers [21,22], sparse priors [15], total variation (TV) regularization [23,13], nonlocal regularization [24,25], Hyper-Laplacian priors [14] and bilateral filter regularization [26], are state-of-the-art. In this paper, we use sparse priors [15] to suppress both noise and ringing artifacts.

1.3. Our contributions

In this paper, our contributions contain two parts. First, we propose a boundary condition based deconvolution framework for image deblurring. Using this framework, we can easily re-derive new efficient deconvolution algorithms with various BCs, such as reflective, anti-reflective, synthetic and undetermined BC, no matter whether the given kernel is symmetric or not. Second, we propose a repeated BC based deconvolution method, which can be easily implemented by our deconvolution framework. Our experimental results show that repeated BC can outperform undetermined BC if the latent image has high local similarity around the borders of image.

1.4. Outline of the paper

In Section 2, we present our deconvolution framework and its general implementation scheme. In Section 3, we first derive the repeated BC based method, and then analogously present other BC based methods. Finally, we review the undetermined BC based approach proposed by [15], and present our simplified approach. In Section 4, we briefly introduce the non-blind deconvolution approach with sparse priors. In Section 5, we show some experimental results both on synthetic and real data. Besides, a qualitative comparison of determinate and undetermined BC is also presented. In Section 6, we present the concluding remarks.

2. Boundary condition based deconvolution framework

In this section, we introduce our deconvolution framework. For simplicity, we suppose the noise is zero, and aims to recover x from y with a given blur kernel.

In this paper, we overload the notation x and y with different meanings: (1) a $N_1 \times N_2$ array of pixels of a digital image, (2) a $N = N_1 \cdot N_2$ vector arranged in column lexicographic ordering, which notation is being used at each point will be clear from the context.

Let the domain of PSF be $[-m_1, m_1] \times [-m_2, m_2]$, and $X^e = L^2(\Omega^e)$ be an expanded linear space of $X = L^2(\Omega)$, where $\Omega = [1, N_1] \times [1, N_2]$ and $\Omega^e = [1 - m_1, N_1 + m_1] \times [1 - m_2, N_2 + m_2]$.

Suppose $P \in L(X, X^e)$, and is defined as

$$P(x) = x^e. \quad (2.1)$$

Obviously, P depends on BC that we have defined. If we view x as a vector, P becomes a very large matrix of size $[(N_1 + M_1 - 1)(N_2 + M_2 - 1), N_1 N_2]$, where $M_1 = 1 + 2m_1$ and $M_2 = 1 + 2m_2$. Hence, we can rewrite (2.1) as the vector–matrix form of

$$Px = x^e. \quad (2.2)$$

We call P the boundary condition operator or boundary condition matrix. For many BCs, such as zero Dirichlet, periodic, reflective and synthetic BC, P is a row permutation matrix.

Removing the noise term n and substituting (2.2) into (1.3), we obtain

$$y = TPx, \quad (2.3)$$

where $T \in L(X^e, X)$ is a partial convolution operator defined by the following equation

$$(Tx^e)(u, v) = \sum_{s=-m_1}^{m_1} \sum_{t=-m_2}^{m_2} h(s, t)x^e(u - s, v - t), \quad (u, v) \in \Omega. \quad (2.4)$$

Obviously, the convolution matrix H of (2.3) is given by

$$H = TP. \quad (2.5)$$

Here, Eq. (2.5) is our boundary condition based algebraic framework. Unlike previous algebraic frameworks [6,9,10,8], which split the convolution matrix into a summation of a few square matrices, our framework simply splits the convolution matrix into a production of two non-square matrices.

For many iterative deconvolution algorithms [15,24,23,22,27], a key step is to compute H^*x . If we can calculate H^*x efficiently, it is easy to solve problem (2.3). For example, knowing how to compute H^*x , one can use CG based algorithm [15,23] or fast iterative shrinkage–thresholding algorithm (FISTA, [27]). In the remainder of this section, we show how to compute H^*x efficiently without calculating any elements of H^* .

Taking the conjugate transpose of (2.5), we have

$$H^* = P^*T^*. \quad (2.6)$$

We find that it is very easy to perform H^* via two linear operations of T^* and P^* . Indeed, we can use Theorem 2.1 to perform T^* efficiently. Although Theorem 2.1 is known and widely used by all the imaging community, we nevertheless present a proof of Theorem 2.1 for the readers who may not know it. Before we prove Theorem 2.1, we first introduce Lemma 2.1.

Lemma 2.1. For a linear operator H defined by $Hx = \sum_{s=1}^{M_1} \sum_{t=1}^{M_2} h(u, v, s, t)x(s, t)$, we have $H^*x = \sum_{s=1}^{M_1} \sum_{t=1}^{M_2} h(u, v)x(s, t)$.

A proof of 1-D case can be found in [28]. For the 2-D case, it can be proved in a similar way.

Theorem 2.1. For a spatially invariant kernel h , the adjoint operator T^* of a partial convolution operator T is a full convolution operator with the kernel h rotated by 180°. That is

$$(T^*x)(u, v) = \sum_{s=\max(-m_1, u-N_1)}^{\min(m_1, u-1)} \sum_{t=\max(-m_2, v-N_2)}^{\min(m_2, v-1)} hr(s, t)x(u - s, v - t), \quad (u, v) \in \Omega^e, \quad (2.7)$$

where $hr(s, t) = h(-s, -t)$ means the rotated version of h by 180°. (For the proof, see Appendix)

Given a boundary condition operator P , we can directly get P^* by transposing P or using Lemma 2.1, since most rows of P are similar to the rows of identity matrix. Given P^* and a blur kernel, we can obtain the solution of (2.3) by solving the following linear equations with CG algorithm

$$P^*T^*TPx = P^*T^*y. \quad (2.8)$$

Eq. (2.8) is our deconvolution framework. Given a BC or P , we only need to derive P^* to obtain an iterative deconvolution algorithm. Hence, the advantage of our deconvolution framework is that we have plenty of freedom to choose a reasonable BC without worrying about the implementation.

Because the computation of convolution is sensitive to the size of kernel, we need to perform T and T^* with 2-D FFT if the kernel is very large.

According to convolution theorem, we have

$$T^* = F^{-1} \text{diag}(HR)FQ, \quad (2.9)$$

where $F \in L(X^e)$ denotes the Fourier transform operator, HR is the Fourier transform of hr padded with zeros to size of x^e after the last array element along each dimension and $Q \in L(X, X^e)$ also pads x with zeros to size of x^e after the last array element along each dimension.

Taking the conjugate transpose of (2.9), we obtain

$$T = Q^* F^{-1} \text{diag}(HR^*) F, \quad (2.10)$$

where HR^* denotes the complex conjugate of HR and $Q^* \in L(X^e, X)$ removes the last $2m_1$ rows and $2m_2$ columns of x^e . Specifically, $Q^*x^e = x^e(1 - m_1 : N_1 - m_1, 1 - m_2 : N_2 - m_2)$ for any $x^e \in X^e$.

Multiplying Eq. (2.9) with (2.10), we get

$$T^* T = F^{-1} \text{diag}(HR) F U F^{-1} \text{diag}(HR^*) F, \quad (2.11)$$

where $U = QQ^*$ just sets the last $2m_1$ rows and $2m_2$ columns of x^e to zero.

3. Deconvolution algorithms with various boundary conditions

3.1. Repeated BC

Since the gradients of a nature image obey heavy-tailed distributions [29], which imply the majority of pixel values are similar to its neighborhood, the borders of x^e are treated as repeated instances of the edge pixel values of x . For a 1-D signal x of length N , we assume $x^e(j) = x(1) (j = 0, -1, \dots, -m + 1)$ and $x^e(N + j) = x(N) (j = 1, 2, \dots, m)$. The extension from 1-D signal to 2-D image is straightforward and Fig. 2 shows the scheme that x expands to x^e . In a word, repeated BC pads the outside elements with the nearest borders so that the local similarity of natural image is taken into consideration.

According to the definition of repeated BC, we can derive that P has the following form

$$P = \begin{bmatrix} & \begin{bmatrix} p \\ \vdots \\ p \end{bmatrix}_{m_2} \\ & \ddots \\ \begin{bmatrix} p & \ddots & p \end{bmatrix}_{N_2} \\ & \begin{bmatrix} p \\ \vdots \\ p \end{bmatrix}_{m_2} \end{bmatrix}_{[N_2+M_2-1] \times N_2}, \quad (3.1)$$

where p has the form of

$$p = \begin{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{m_1} \\ & \ddots \\ \begin{bmatrix} 1 & \ddots & 1 \end{bmatrix}_{N_1} \\ & \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{m_1} \end{bmatrix}_{[N_1+M_1-1] \times N_1}. \quad (3.2)$$

Taking the transpose of P , we obtain P^* which has the form of

$$P^* = \left[\begin{bmatrix} p^* & \cdots & p^* \end{bmatrix}_{m_2} \begin{bmatrix} p^* & \ddots & p^* \end{bmatrix}_{N_2} \begin{bmatrix} p^* & \cdots & p^* \end{bmatrix}_{m_2} \right]_{[N_2+M_2-1] \times N_2}, \quad (3.3)$$

$$P \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 2 & 3 & 3 & 3 \\ 1 & 1 & 1 & 2 & 3 & 3 & 3 \\ 1 & 1 & 1 & 2 & 3 & 3 & 3 \\ 4 & 4 & 4 & 5 & 6 & 6 & 6 \\ 7 & 7 & 7 & 8 & 9 & 9 & 9 \\ 7 & 7 & 7 & 8 & 9 & 9 & 9 \\ 7 & 7 & 7 & 8 & 9 & 9 & 9 \end{bmatrix}$$

Fig. 2. Illustration of repeated BC. The left 3×3 matrix is expanded to a 5×5 matrix with repeated boundary, where the kernel size is 5×5 .

where p^* has the form of

$$p^* = \left[\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{m_1} \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}_{N_1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{m_1} \right]_{N_1 \times [N_1 + M_1 - 1]} . \quad (3.4)$$

As we can see from (3.3) and (3.4), P^* adds all padding boundary values of x^e back to the original boundary of x^e , while keeps the inside values of x^e constant. Obviously, it is very simple to perform repeated boundary condition operator as well as its adjoint operator.

For the other BCs, such as reflective, anti-reflective and synthetic BC, we can derive the respective deconvolution algorithm in a similar way. Here, we present only the adjoint operator P_{HR}^* of reflective boundary operator of the half-sample version, which is shown in (3.5). As we can see from (3.5), P_{HR}^* adds all padding region of x^e back to the associated reflective boundary, while keeps the inside values of x^e constant. For the adjoint operator P_{AR}^* of anti-reflective boundary operator, we have $P_{AR}^* = 2P^* - P_{WR}^*$, since $P_{AR} = 2P - P_{WR}$, where P_{WR} denotes reflective boundary operator of the whole-sample version and P is the repeated boundary operator. This formula shows that we can easily perform P_{AR}^* by two operations of P^* and P_{WR}^* . Because P_{WR} is quite similar to P_{HR} , we use P_{HR} to replace P_{WR} in our experiment. For the synthetic boundary operator P_S^* [9], we cannot show the general form of P_S^* since P_S depends on image. Here, we just present an intuitive way to compute $P_S^*x^e$, which is adding the copying pixels in the padding region of x^e back to the associated pixel of x .

$$P_{HR}^* = \left[\begin{bmatrix} p_{HR}^* & & \\ & \ddots & & \\ & & p_{HR}^* & \\ p_{HR}^* & \ddots & & \end{bmatrix}_{m_2} \begin{bmatrix} p_{HR}^* & & & \\ & \ddots & & \\ & & p_{HR}^* & \\ & & & \ddots & \\ & & & & p_{HR}^* \end{bmatrix}_{N_2} \begin{bmatrix} p_{HR}^* & & & \\ & \ddots & & \\ & & p_{HR}^* & \\ p_{HR}^* & \ddots & & \end{bmatrix}_{m_2} \right]_{[N_2 + M_2 - 1] \times N_2} \quad (3.5)$$

where p_{HR}^* has the form of

$$p_{HR}^* = \left[\begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ 1 & & & \ddots & \\ & \ddots & & & 1 \end{bmatrix}_{m_1} \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}_{N_1} \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}_{m_1} \right]_{[N_1 + M_1 - 1] \times N_1} . \quad (3.6)$$

3.2. Undetermined BC

To our best knowledge, undetermined BC is first proposed by Levin et al. [15]. Here, we present a brief review of their method. For simplicity, some MATLAB notations are used.

In essence, Levin et al. [15] try to solve the following minimization problem

$$\min_{x^e} \|mask * (Hx^e - y^e)\|_2^2 , \quad (3.7)$$

where $Hx^e = conv2(x^e, h, 'same')$, y^e is the enlarged version of y , and $mask$ is a diagonal matrix which is defined as

$$mask(i) = \begin{cases} 0, & \text{if } i \text{ is inside the padding region} \\ 1, & \text{others,} \end{cases} \quad (3.8)$$

where i denotes the pixel index. With the help of mask, the unwanted residual introduced by the outside boundary pixels are removed, whereas the residual within the FOV are preserved. Besides, H^* can be efficiently performed by the operator of $conv2(\bullet, hr, 'same')$.

Obviously, the solution of (3.7) is equivalent to the least square solution of the following linear equations

$$maskHx^e = masky^e . \quad (3.9)$$

Actually, for undetermined BC, it is not necessary to involve a mask. Setting the boundary condition matrix P as unknown variables, and noting that $\ker(P) = 0$ and $Px = x^e$, linear equations (2.8) reduce to

$$T^*Tx^e = T^*y. \quad (3.10)$$

Removing all the zero equations of (3.9), the least square solution of the rest of linear equations of (3.9) is equivalent to the solution of (3.10). Besides, T^*Tx^e only needs to convolute $N_1N_2 + (N_1 + 2m_1)(N_2 + 2m_2)$ pixels, whereas $H^*maskHx^e$ needs to convolute $2(N_1 + 2m_1)(N_2 + 2m_2)$ pixels.

In all, we have showed that the undetermined BC based deconvolution method has a simpler form under our framework, and this method yields the same solution with the one of [15].

4. Regularization framework for non-blind deconvolution

In this section, we consider how to estimate an unknown original image x from a noisy observation data y with a given PSF.

The challenge is that this linear inverse problem (LIP) is usually ill-posed. That means the linear operator is nearly singular yielding highly noise sensitive solutions. Besides, a small error in PSF may introduce many undesirable artifacts into the recovery image if the LIP is ill-posed.

To cope with the ill-posedness of deconvolution problem, abundant techniques have been developed, many of which are in the regularization framework. In the regularization framework, the solution x is defined as a minimizer of an objective function $f : X \rightarrow R = (-\infty, +\infty)$, given by

$$f(x) = \frac{1}{2} \|y - Hx\|^2 + \lambda \Phi(x), \quad (4.1)$$

where $\Phi : x \rightarrow R$ is a nonnegative function which is designed to reduce noise, and λ is a nonnegative parameter which controls the degree of reducing noise. In this paper, we call Φ the regularizer and λ the regularization parameter. Here, we use the sparse priors (also known as Hyper-Laplacian priors [14]) proposed by [15]. Thus, the objective function defined by (4.1) becomes

$$f(x) = \frac{1}{2} \|y - Hx\|^2 + \lambda \sum_j \|G_j x\|^{0.8}, \quad (4.2)$$

where G_j denotes the j th convolution matrix which is defined by a first or second order derivative filter. The sparse prior opts to concentrate derivatives at a small number of pixels, leaving the majority of pixels constant. This produces sharper edges, reduces noise and helps remove undesirable image artifacts such as ringing. The drawback of the sparse prior is that the optimization problem is non-convex, and has no closed-form solution [15]. To solve this non-convex minimization problem, an iteratively reweighted least squares approach can be used, which leads to solve a series of linear equations

$$\left(H^*H + \lambda \sum_j G_j^* W_j^t G_j \right) x^{t+1} = H^*y, \quad t = 0, 1, 2, \dots, T, \quad (4.3)$$

where W_j^t denotes the diagonal weighted matrix, which is defined as

$$W_j^t = \max(G_j x^t, \varepsilon)^{0.8-2}, \quad W_j^0 = 1, \quad (4.4)$$

where ε is a threshold (e.g., 0.01) to avoid division by zero. Here, CG algorithm is used to solve linear equations (4.3).

In Experiments 1–5, for fair comparison, we use the same implementation as [15] so that the only difference is boundary condition. In Experiment 6, we modify the method [15] by utilizing a warm start strategy, using x^t as the initialization of the next CG algorithm. This warm start strategy makes use of previous CG iterations and therefore accelerates the convergence of CG algorithm, while [15] uses x^t for updating (4.4) only. Besides, each CG algorithm stops automatically when the relative variant of x is less than tol or the number of iterations reaches 200.

Theoretically, our CG based deconvolution algorithm has a computation complexity of $O(NM_{NZ})$, where $N = N_1N_2$ and M_{NZ} denotes the number of non-zero elements of blur kernel. Hence, it is quite suitable for large image and sparse kernel, like motion blur kernel.

5. Experiments

From Sections 5.1 to 5.2, we compare various BCs mentioned above by using synthetic data as well as real data. For results of synthetic data, we adopt PSNR to make a quantitative assessment, where PSNR is defined as

$$\text{PSNR} = 20 \log_{10} \frac{\sqrt{N}255}{\|x - x_{\text{true}}\|_2}.$$

In Section 5.3, we compare our modified CG-UBC and CG-RBC with ADMM-UBC [16] (available at http://www.lx.it.pt/~mscla/ADMM_UBC.htm) in terms of ISNR, where UBC stands for undetermined BC and RBC stands for repeated BC and



Fig. 3. Deblurred results of experiment 1. (a) Original image. (b) Blurred image. (c) Result of reflective BC. (d) Result of anti-reflective BC. (e) Result of repeated BC. (f) Result of undetermined BC. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ISNR is defined as

$$\text{ISNR} = 20 \log_{10} \frac{\|y - x_{\text{true}}\|_2}{\|x - x_{\text{true}}\|_2}.$$

5.1. Experiments on synthetic data

Experiment 1. We aim to show the effectiveness of our BC based deconvolution framework as well as the performance of various BCs. We choose the “Barbara” image, and blur it with a Gaussian kernel of size 11 with a standard deviation of 3,



Fig. 4. Deblurred results of experiment 2. (a) Original image. (b) Blurred image, PSNR = 20.7315 dB. (c) Result of [4], PSNR = 29.5999 dB. (d) Result of repeated BC, PSNR = 30.0689 dB. (e) Result of [15], PSNR = 30.4372 dB. (f) Our result of undetermined BC without mask, PSNR = 30.4372 dB.

and crop out the FOV (see Fig. 3). Table 1 shows the PSNRs of deblurred images with various BCs as well as the number of iterations. In Fig. 3, we can see that undetermined BC introduces little boundary artifacts, whereas reflective, anti-reflective and repeated BC cause visually severe ringing artifacts. The red box in Fig. 3(a) shows that repeated BC fails in this patch and causes noticeable artifacts (see the red box in Fig. 3(e)).

Experiment 2. We choose the “cameraman” image, and blur it with a uniform kernel of size 9×9 , and crop out the FOV (see Fig. 4). To make a fair comparison, we round the blurred data to the nearest integers. Hence, a very small amount

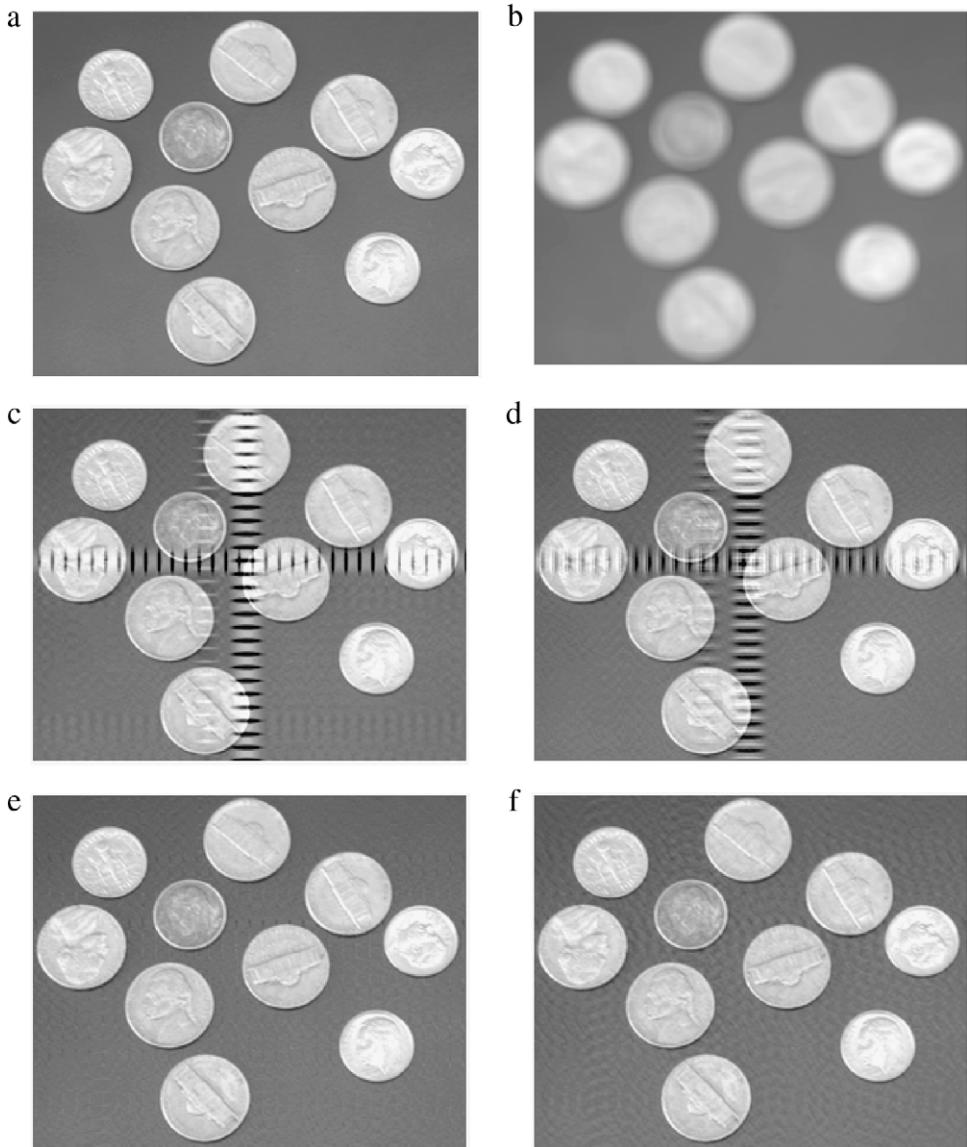


Fig. 5. Deblurred results of experiment 3. (a) Original image. (b) Blurred image. (c) Result of reflective BC. (d) Result of anti-reflective BC. (e) Result of repeated BC. (f) Result of undetermined BC.

noise is involved into the observation data. For the non-blind deconvolution method of [4], we set noiseStr = 0.0015 and localPriorWeight = 1. For the non-blind deconvolution method of [15], we set $\lambda = 0.00004$ and iteration = 200. In Fig. 4, we can see visually boundary artifacts in Fig. 4(c), whereas Fig. 4(d) just has a small amount of boundary artifacts, and Fig. 4(e)–(f) has invisible boundary artifacts (noting that Fig. 4(e) is the same as Fig. 4(f)). With the help of regularization prior, boundary artifacts in Fig. 4(d)–(f) are all reduced largely.

Experiment 3. Fig. 5 shows an example that repeated BC outperforms undetermined BC. Here, we choose an image which has “repeated” boundary, and blur it with a uniform kernel of size 9×9 , and crop out the FOV. Table 2 shows the PSNRs of deblurred images with various BCs as well as the run-times. All the results are obtained by 2000 CG iterations. This experiment demonstrates that in some cases, especially when the missing borders of the original image are similar to its neighborhood, repeated BC outperforms undetermined BC.

5.2. Experiments on real data

Experiment 4. We choose the real data provided by [30]. For a slightly blurred image in Fig. 6(b), undetermined BC rarely introduces boundary artifacts, whereas repeated BC still causes some boundary artifacts. For a severely blurred image in

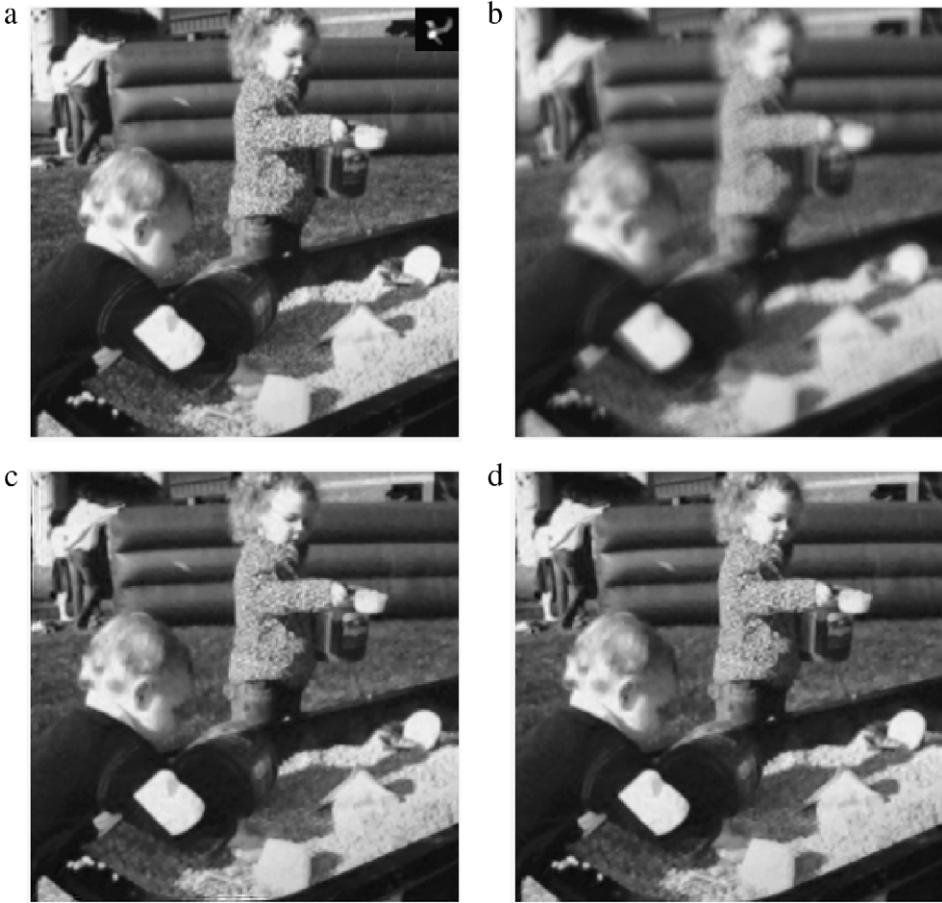


Fig. 6. (a) Ground truth. (b) Blurred image and blur kernel of size 13×13 . (c) Result of repeated BC. (d) Result of undetermined BC.

Table 1
PSNRs of deblurred images with different boundary conditions.

	Blurred image	Reflective [9]	Anti-reflective [9]	Synthetic [9]
PSNR	24.5646	27.4083	28.4664	28.7532
Iteration	–	167	255	500
	Repeated (ours)	Reflective (ours)	Anti-reflective (ours)	Undetermined (ours)
PSNR	29.0816	27.2427	28.8415	32.6487
Iteration	103	25	112	1000

Table 2
PSNRs of deblurred images with different boundary conditions.

	Blurred image	Reflective	Anti-reflective	[15]	Undetermined (ours)	Repeated
PSNR	23.8036	21.1474	24.428	36.6275	36.6275	40.131
Time (s)	–	16.61	20.5	17.46	15.29	17.13

Fig. 7(b), undetermined BC well restores the inside part of image but fails to recover the boundary pixels (only one black dot is well restored), whereas repeated BC causes many remarkable boundary artifacts propagating throughout the entire image.

Experiment 5. We choose the real data provided by [4]. As we can see from Fig. 8(e), for a large and slightly blurred image, which has high local similarity around the borders, repeated BC may outperform undetermined BC and improved periodic BC [11,4].

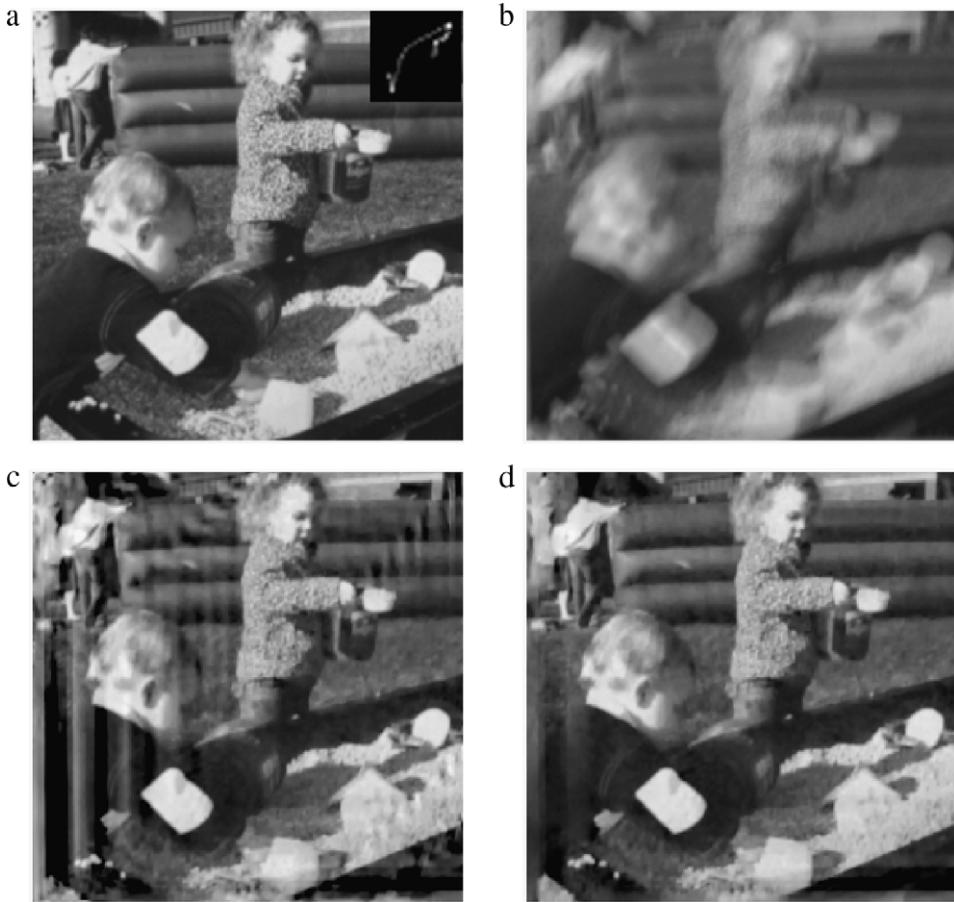


Fig. 7. (a) Ground truth. (b) Blurred image and blur kernel of size 27×27 . (c) Result of repeated BC. (d) Result of undetermined BC.

5.3. Comparison with ADMM-UBC [16]

Experiment 6. Like the experiment in [16], we use “cameraman” image with 4 different blurs (uniform, out-of-focus, linear motion and Gaussian, all of size 19×19), at 4 different BSNRs (blur signal noise ratio): 30 dB, 40 dB, 50 dB, 60 dB. We set $T = 8$ and $tol = 0.000001$ for all 16 blurred images, and manually adjust the regularization parameter λ to yield the highest ISNR for each experimental condition. Table 3 reports the ISNRs of deblurred images with CG-UBC, CG-RBC, TV-MD [16] and FA-MD [16]. As we can see from Table 3, our CG-UBC has the best performance in terms of ISNR in all scenarios. Besides, our CG-RBC is quite competitive when BSNR is less than 40 dB, thanks to the regularization term. It is interesting that CG-RBC has very good performance for Gaussian blur even if BSNR is quite high, indicating that repeated BC prefers Gaussian blur. Like we show in Experiments 1–4, CG-RBC may yield noticeable boundary artifacts when BSNR is higher than 40 dB. Fig. 9 shows one example of 16 for visual comparison. We should mention that, for the blurred image Fig. 9(b), the reported ISNRs and run-times of CG-UBC and CG-RBC are obtained by setting $tol = 0.00001$.

5.4. Results analysis

As we can see from Figs. 3–8, in general cases, undetermined BC usually outperforms determinate BCs, such as zero Dirichlet, reflective, anti-reflective, synthetic, periodic, improved periodic [11], and repeated BC. Here, we present a qualitative comparison between undetermined and determinate BCs.

Consider the following minimization problem

$$\min_x f(x) = \|TPx - y\|_2^2, \quad (5.1)$$

and substitute $y = TP_0x_0$ into (5.1), where P_0 and x_0 denote an accurate boundary condition matrix and the true scene within the FOV, respectively, we obtain

$$\min_x f(x) = \|TP_0(x - x_0) + T\delta Px\|_2^2, \quad (5.2)$$

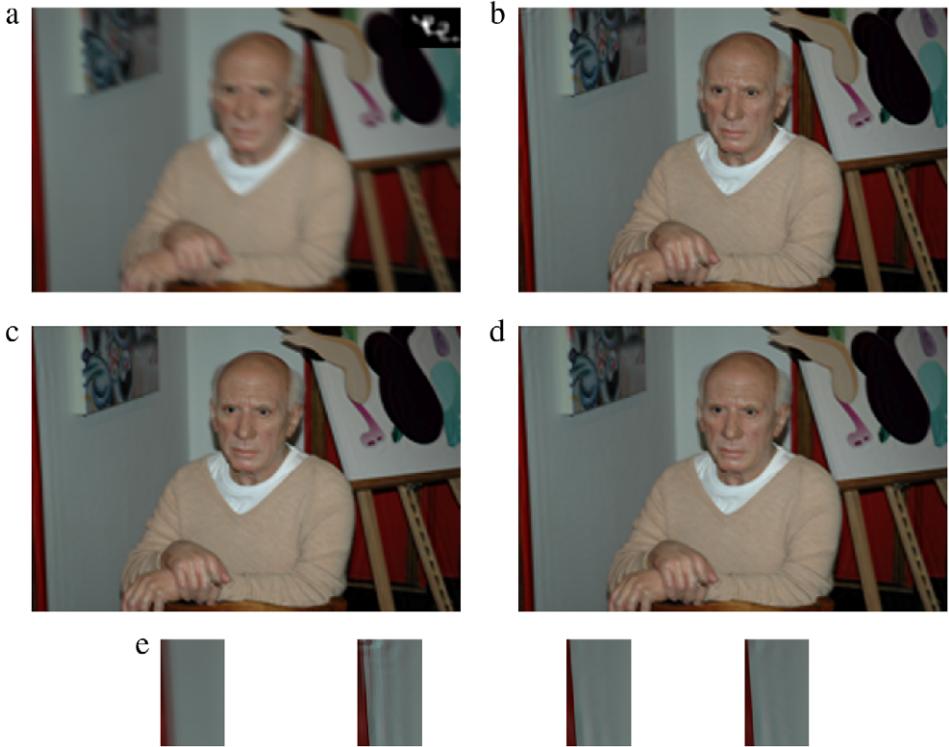


Fig. 8. (a) Blurred image and the estimated blur kernel of size 19×27 . (b) Result of [4]. (c) Result of repeated BC. (d) Result of undetermined BC. (e) From left to right: close-up views of (a)–(d), respectively.

Table 3
ISNRs of deblurred images with 4 different algorithms.

Blur	BSNR (dB)	CG-UBC	CG-RBC	TV-MD	FA-MD
Uniform	30	5.46	5.41	5.44	5.27
Out-of-focus	30	5.82	5.78	5.66	5.53
Linear motion	30	8.81	8.51	8.24	8.12
Gaussian	30	3.28	3.27	3.21	2.92
Average for	30	5.84	5.74	5.64	5.46
Uniform	40	7.26	7.14	7.02	7.17
Out-of-focus	40	8.75	8.61	8.34	8.51
Linear motion	40	13.21	11.76	12.41	12.59
Gaussian	40	4.23	4.20	4.03	3.80
Average for	40	8.36	7.93	7.95	8.02
Uniform	50	10.12	9.54	9.75	9.99
Out-of-focus	50	12.37	11.62	11.78	12.10
Linear motion	50	17.45	13.11	16.67	16.82
Gaussian	50	5.41	5.32	4.78	4.67
Average for	50	11.34	9.99	10.77	10.90
Uniform	60	12.93	11.27	11.95	12.52
Out-of-focus	60	15.79	13.10	14.89	15.77
Linear motion	60	20.72	13.30	19.88	20.34
Gaussian	60	5.85	5.74	4.97	5.01
Average for	60	13.82	10.85	12.92	13.41

where $\delta P = P - P_0$ denotes the difference between the true BC and the assumed one. Noting that the majority rows of δP are zeros, the objective function of (5.1) can be approximated by

$$\min_x f(x) \approx \|TP_0(x - x_0)\|_2^2 + \|T\delta Px\|_2^2. \quad (5.3)$$

As we can see in (5.3), for determined BC, if the assumed BC is not accurate, $T\delta P$ may keep x away from x_0 . On the contrary, for undetermined BC, $T\delta P$ consists of variables and gives more freedom to x , so x is able to get close to x_0 . Therefore, in general cases, undetermined BC usually outperforms determined BCs. Moreover, if the assumed BC is very accurate, $T\delta P$ is close to zero matrix, thus x gets very close to x_0 . Hence, in cases of Figs. 5 and 8, repeated BC outperforms undetermined BC.



Fig. 9. One instance of experiment 6. (a) Original image. (b) Motion blurred image with BSNR = 30 dB, ISNR = 8.26 dB, time = 5.03 s. (d) FA-MD [16], ISNR = 8.12 dB, time = 12.63 s. (e) CG-RBC, ISNR = 8.51 dB, time = 10.19 s. (f) CG-UBC, ISNR = 8.81 dB, time = 10.33 s.

6. Conclusion

In this work, we have presented a simple and efficient deconvolution framework for image deblurring. Using this framework, we have easily derived or re-derived a few efficient deconvolution algorithms with various BCs, such as reflective, anti-reflective, repeated, and undetermined BC. It is enough to show that the framework can derive other BC based deconvolution algorithms.

In most cases, undetermined BC has the best performance. In some cases, determinate BC can outperform undetermined BC. For instance, repeated BC is very suitable to deblur a large image with local similar borders like Fig. 8(a). Due to the help of regularization prior, different BCs may yield similar quality images, if the observation data is not severely blurred.

However, if the given image is severely blurred, even undetermined BC may fail to recover the details in image boundary. Hence, more accurate BC is required to recover the information around image boundary.

Acknowledgments

We thank Anat Levin and Qi Shan for sharing their images and source codes.

Appendix

Proof of Theorem 2.1. For any $x^e \in X^e$, we have

$$\begin{aligned} (Tx^e)(u, v) &= \sum_{s=-m_1}^{m_1} \sum_{t=-m_2}^{m_2} h(s, t)x^e(u-s, v-t), \quad (u, v) \in \Omega \\ &= \sum_{s=u-m_1}^{u+m_1} \sum_{t=v-m_2}^{v+m_2} h(u-s, v-t)x^e(s, t), \quad (u, v) \in \Omega. \end{aligned} \quad (\text{A.1})$$

According to Lemma 2.1, for any $x \in X$, by exchanging the positions of (u, v) and (s, t) , we have

$$\begin{aligned} (T^*x)(u, v) &= \sum_{s=u-m_1}^{u+m_1} \sum_{t=t-m_2}^{v+m_2} h(s-u, t-v)x(s, t), \quad (s, t) \in \Omega \text{ and } (u, v) \in \Omega^e \\ &= \sum_{s=u-m_1}^{u+m_1} \sum_{t=v-m_2}^{v+m_2} hr(u-s, v-t)x(s, t), \quad (s, t) \in \Omega \text{ and } (u, v) \in \Omega^e \\ &= \sum_{s=-m_1}^{m_1} \sum_{t=-m_2}^{m_2} hr(s, t)x(u-s, v-t), \quad (u-s, v-t) \in \Omega \text{ and } (u, v) \in \Omega^e \\ &= \sum_{s=\max(-m_1, u-N_1)}^{\min(m_1, u-1)} \sum_{t=\max(-m_2, v-N_2)}^{\min(m_2, v-1)} hr(s, t)x(u-s, v-t), \quad (u, v) \in \Omega^e. \end{aligned} \quad (\text{A.2})$$

References

- [1] M.R. Barham, A.K. Katsaggelos, Digital image restoration, *IEEE Signal Processing Magazine* 14 (1997) 24–41.
- [2] O. Shacham, O. Haik, Y. Yitzhaky, Blind restoration of atmospherically degraded images by automatic best step-edge detection, *Pattern Recognition Letters* 28 (2007) 2094–2103.
- [3] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, W.T. Freeman, Removing camera shake from a single photograph, in: ACM SIGGRAPH, 2006.
- [4] Q. Shan, J.Y. Jia, A. Agarwala, High-quality motion deblurring from a single image, in: ACM SIGGRAPH, 2008.
- [5] L. Xu, J.Y. Jia, Two-phase kernel estimation for robust motion deblurring, in: Proceedings of the 11th European conference on Computer Vision, 2010.
- [6] M. Ng, R. Chan, W. Tang, A fast algorithm for deblurring models with Neumann boundary conditions, *SIAM Journal on Scientific Computing* 21 (1999) 851–866.
- [7] S. Serra-Capizzano, A note on antireflective boundary conditions and fast deblurring models, *SIAM Journal on Scientific Computing* 25 (2003) 1307–1325.
- [8] M. Donatelli, S. Serra-Capizzano, Antireflective boundary conditions for deblurring problems, *Journal of Electrical and Computer Engineering* 2010 (2010) 18. <http://dx.doi.org/10.1155/2010/241467>. Article ID 241467.
- [9] Y.W. Fan, James G. Nagy, Synthetic boundary conditions for image deblurring, *Linear Algebra and its Applications* 434 (2011) 2244–2268.
- [10] X.G. Lv, T.Z. Huang, Z.B. Xu, X.L. Zhao, Kronecker product approximations for image restoration with whole-sample symmetric boundary conditions, *Information Sciences* 186 (2012) 150–163.
- [11] R. Liu, J.Y. Jia, Reducing boundary artifacts in image deconvolution, in: Proc. Int. Conf. Image Process, 2008.
- [12] M. Donatelli, C. Estatico, A. Martinelli, S. Serra-Capizzano, Improved image deblurring with anti-reflective boundary conditions and re-blurring, *Inverse Problems* 22 (2006) 2035–2053.
- [13] Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, *SIAM Journal on Imaging Sciences* 1 (2008) 248–272.
- [14] D. Krishnan, R. Fergus, Fast image deconvolution using hyper-Laplacian priors, in: NIPS, 2009.
- [15] A. Levin, R. Fergus, F. Durand, B. Freeman, Image and depth from a conventional camera with a coded aperture, in: ACM SIGGRAPH, 2007.
- [16] Mariana S.C. Almeida, Mário A.T. Figueiredo, Deconvolving images with unknown boundaries using the alternating direction method of multipliers, *IEEE Transactions on Image Processing* 22 (2013) 3074–3086.
- [17] Antonios Matakos, Sathish Ramani, Jeffrey A. Fessler, Accelerated edge-preserving image restoration without boundary artifacts, *IEEE Transactions on Image Processing* 22 (2013) 2019–2029.
- [18] L.B. Lucy, An iterative technique for the rectification of observed distribution, *The Astronomical Journal* 79 (6) (1974) 745–754.
- [19] W.H. Richardson, Bayesian-based iterative method of image restoration, *Journal of the Optical Society of America A* 62 (1972) 55–59.
- [20] R. Neelamani, H. Choi, R.G. Baraniuk, ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems, *IEEE Transactions on Signal Processing* 52 (2004) 418–433.
- [21] J. Bioucas-Dias, Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors, *IEEE Transactions on Image Processing* 15 (2006) 937–951.
- [22] M. Figueiredo, J. Bioucas-Dias, R.D. Nowak, Majorization–minimization algorithms for wavelet-based image restoration, *IEEE Transactions on Image Processing* 16 (2007) 2980–2991.

- [23] J. Bioucas-Dias, M. Figueiredo, J.P. Oliveira, Total variation-based image deconvolution: a majorization–minimization approach, in: IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Toulouse, France, 2006.
- [24] X. Li, Fine-granularity and spatially-adaptive regularization for projection-based image deblurring, *IEEE Transactions on Image Processing* 20 (2011) 971–983.
- [25] X.Q. Zhang, M. Burger, X. Bresson, S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, *SIAM Journal on Imaging Sciences* 3 (2010) 253–276.
- [26] L. Yuan, J. Sun, L. Quan, H.Y. Shum, Progressive inter-scale and intra-scale non-blind image deconvolution, in: ACM SIGGRAPH, 2008.
- [27] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences* 2 (2009) 183–202.
- [28] Z.J. Jiang, S.L. Sun, Functional Analysis, second ed., Higher Education Press, Beijing, 2005.
- [29] D. Field, What is the goal of sensory coding? *Neural Computation* 6 (1994) 559–601.
- [30] A. Levin, Y. Weiss, F. Durand, W.T. Freeman, Understanding and evaluating blind deconvolution algorithms, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.