Information Sciences 186 (2012) 150–163

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Kronecker product approximations for image restoration with whole-sample symmetric boundary conditions ☆

Xiao-Guang Lv [a,*], Ting-Zhu Huang [a], Zong-Ben Xu [b], Xi-Le Zhao [a]

[a] School of Mathematical Sciences/Institute of Computational Science, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, PR China
[b] Institute for Information and System Sciences, Xi'an Jiaotong University, Xi'an, Shan'xi 710049, PR China

ARTICLE INFO

Article history:
Received 10 March 2010
Received in revised form 6 September 2011
Accepted 11 September 2011
Available online 2 October 2011

Keywords:
Image restoration
Whole-sample symmetric BCs
Kronecker product approximation
TSVD
GCV

ABSTRACT

Reflexive boundary conditions (BCs) assume that the array values outside the viewable region are given by a symmetry of the array values inside. The reflection guarantees the continuity of the image. In fact, there are usually two choices for the symmetry: symmetry around the meshpoint and symmetry around the midpoint. The first is called whole-sample symmetry in signal and image processing, the second is half-sample. Many researchers have developed some fast algorithms for the problems of image restoration with the half-sample symmetric BCs over the years. However, little attention has been given to the whole-sample symmetric BCs. In this paper, we consider the use of the whole-sample symmetric boundary conditions in image restoration. The blurring matrices constructed from the point spread functions (PSFs) for the BCs have block Toeplitz-plus-PseudoHankel with Toeplitz-plus-PseudoHankel blocks structures. Recently, regardless of symmetric properties of the PSFs, a technique of Kronecker product approximations was successfully applied to restore images with the zero BCs, half-sample symmetric BCs and anti-reflexive BCs, respectively. All these results extend quite naturally to the whole-sample symmetric BCs, since the resulting matrices have similar structures. It is interesting to note that when the size of the true PSF is small, the computational complexity of the algorithm obtained for the Kronecker product approximation of the resulting matrix in this paper is very small. It is clear that in this case all calculations in the algorithm are implemented only at the upper left corner submatrices of the big matrices. Finally, detailed experimental results reporting the performance of the proposed algorithm are presented.

1. Introduction

Image restoration is a fundamental problem in digital image processing. It can be viewed as an estimation process in which operations are performed on an observed image to estimate the ideal image. Such problems arise in a variety of applications in astronomy, medicine, physics and biology [2,3,5,8,21–23,39,49]. Motivated by these applications, both mathematicians and engineers have developed some fast algorithms for the computation of image restoration over the years. Among these methods, many popular direct methods such as truncated SVD, Wiener filtering method and Tikhonov regularization method as well as other direct filtering methods are widely used to get restored images of high quality for small or medium-sized image restoration problems [29,17,22,24,30,31,35]. As an alternative to direct methods for large-scale

☆ This research is supported by 973 Program (2007CB311002), NSFC (60973015, 61170311), Sichuan Province Sci. and Tech. Research Project (2009SPT-1, 2009GZ0004, 2011JY0002).

* Corresponding author.
E-mail addresses: xiaoguanglv@126.com (X.-G. Lv), tzhuang@uestc.edu.cn, tingzhuhuang@126.com (T.-Z. Huang).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.09.026

problems, iterative methods may be more attractive; see [22,28,36,11,45] for more details. Besides, partial differential-equation-(PDE-) based methods have recently received much attention [4,6,13,18,34,38,40].

In general, a spatially invariant image degradation can be linearly modelled as

$$g = Kf + \eta, \tag{1}$$

where K is a matrix associated with the point spread function (PSF) and with the boundary conditions (BCs), g is the recorded image, f is the original image to be recovered and η is additive noise. In this work, the PSF is assumed to be known or to be estimated from the data. Many techniques are available for estimating the PSF by a variety of means [26,27,43].

The exact structure of matrix K depends on the imposed boundary conditions. Four frequently used BCs and their correspondence with K are list here [1,2,7,10,12,16,21,33,37]:

- Periodic boundary conditions. It assumes that the true infinite scene can be represented as a mosaic of a single finite dimensional image, repeated periodically in all directions. The associated matrix K with this BCs is a block circulant with circulant blocks (BCCB) matrix.
- Zero boundary conditions. In this case, all pixels outside the image domain is zero. With the BCs, the matrix K has a block Toeplitz with Toeplitz blocks (BTTB) structure.
- Half-sample symmetric boundary conditions. This is usually used reflexive boundary conditions.It assumes that the scene outside the viewable region is a mirror reflection of the scene inside the viewable region. In the case of the frequently-used reflexive BCs, the matrix K is a block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB) matrix.
- Antireflexive boundary conditions. It extends the pixel values across the boundary in such a way that continuity of the image and of the normal derivative are preserved at the boundary. The matrix arising from the BCs has a block Toeplitz-minus-Hankel-plus-Rank-2 with Toeplitz-minus-Hankel-plus-Rank-2 blocks (BTH2TH2B) structure.

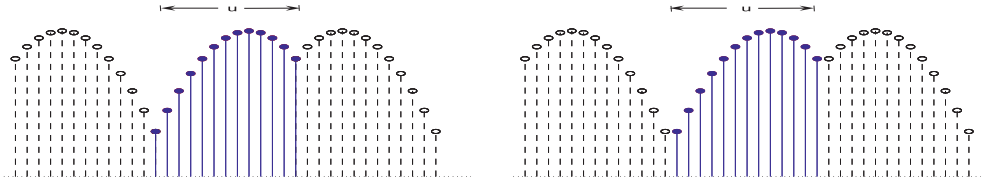
The four BCs are very attractive, since the operations of algorithms involving the resulting matrices are inexpensive. In the first case, it is well known that the computations with BCCB matrices can be done very efficiently by using fast Fourier transforms (FFTs). In the second case, for zero BCs, although direct methods cannot be implemented as efficiently as in the case of circulant structures, it is possible to efficiently use iterative methods, since matrix–vector multiplication involving BTTB matrices can be performed very efficiently using FFTs. In the last two cases, if the PSF is symmetric, the resulting BTHTHB matrices from half-sample symmetric BCs and the BTH2TH2B matrices arising from antireflexive BCs can be diagonalized by the two-dimensional discrete cosine transform matrix and sine transform matrix, respectively.

However, if there are significant features that overlap the edge of the viewable region, then it may be very effective to use reflexive boundary conditions. As we all know, reflexive BCs imply that the array values outside the viewable region are given by a symmetry of the array values inside. In fact, there are usually two choices for the symmetry: symmetry around the meshpoint and symmetry around the midpoint. The first is called whole-sample symmetry in signal and image processing, the second is half-sample [9,33,47]. More precisely, we take a signal vector u of order 13 for example; see Fig. 1.

Many authors studied the problems of the half-sample symmetric BCs in image restoration [21,35,37]. However, little attention has been given to the whole-sample symmetric BCs. In this paper we consider the use of the whole-sample symmetric BCs. By a simple discussion, we obtain that the resulting matrix K from the whole-sample symmetric BCs is a Toeplitz-plus-PseudoHankel matrix in the one-dimensional case and a block Toeplitz-plus-PseudoHankel with Toeplitz-plus-PseudoHankel blocks matrix in the two-dimensional case.

In [25,35,41], Kronecker product approximations were obtained for the resulting matrices from the zero BCs, half-sample symmetric BCs and anti-reflexive BCs, respectively. Their theoretical analysis and numerical experiments have shown that these approximation algorithms are quite efficient. These motivate us to consider the Kronecker product approximations for the whole-sample symmetric BCs. In the present paper, we propose a similar algorithm for constructing the Kronecker product approximation by using the given PSF. Using this approximation, we can obtain the solution of linear systems (1) by the truncated singular value decomposition (TSVD) method with the truncation parameter chosen by the generalized cross validation (GCV) method.

We summarize briefly the content of our paper. In Section 2, we introduce definitions and notations that will be used throughout the paper. A simple discussion about the structure of the resulting matrices from the whole-sample symmetric BCs is given in Section 3. In Section 4, we study the Kronecker product approximation to the resulting matrix K from the whole-sample symmetric BCs and provide an algorithm for constructing this approximation by using the given PSF. Detailed



(a) Whole-sample symmetry (b) Half-sample symmetry

Fig. 1. An example of symmetry types.

experimental results reporting the performance of the proposed algorithm are given in Section 5. Section 6 contains concluding remarks.

2. Notations and conventions

At first, for purposes of presentation, let us introduce some notations that will be used throughout the paper and recall some results about the Kronecker product approximations.

The matrix I denotes the identity matrix and the shift matrix Z is given by

$$Z = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}.$$

e_l represents the lth column of the identity matrix I. A zero matrix or a zero vector is represented by O, regardless of its dimension.

Some basic properties of the shift matrix are quite predictable:

1. $Z^k e_l = \begin{cases} 0, & l = 1, \ldots, k \\ e_{l-k}, & l = k+1, \ldots, n \end{cases}$,
2. $\begin{cases} (Z^k)^T Z^k = \mathrm{diag}([0, \ldots, 0, 1, \ldots, 1]) \\ \qquad\qquad\quad \uparrow \\ \qquad\qquad\quad k+1 \text{ entry} \\ Z^k (Z^k)^T = \mathrm{diag}([1, \ldots, 1, 0, \ldots, 0]) \\ \qquad\qquad\quad \uparrow \\ \qquad\qquad\quad n-k \text{ entry} \end{cases}$,
3. $\sum_{k=1}^{n-1} ((Z^k)^T Z^k + Z^k (Z^k)^T) = (n-1) I$,
4. For $0 < a, b < n$, $(Z^a)^T Z^b + Z^{n-1-a} (Z^{n-1-b})^T = \begin{cases} Z^{b-a} + E_{a+1,b+1}, & a < b \\ (Z^{a-b})^T + E_{a+1,b+1}, & a > b \end{cases}$, where $E_{a+1,b+1}$ is an n-by-n matrix whose $(a+1, b+1)$-entry is one and zero elsewhere.

To describe banded Toeplitz and PseudoHankel matrices, we use the following notations:

$$\mathrm{toep}(a, k) = \left[Z^{k-1} a, Z^{k-2} a, \ldots, Z^1 a, Z^0 a, Z^T a, \ldots, (Z^{n-k})^T a \right].$$

$\mathrm{toep}(a,k)$ is an $n \times n$ banded Toepltiz matrix whose kth column is $a \in \mathbb{R}^n$. In fact, if $a = (a_1, a_2, \ldots, a_n)^T$, then

$$\mathrm{toep}(a, k) = \begin{bmatrix} a_k & \cdots & \cdots & a_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ a_n & \ddots & \ddots & \ddots & \ddots & \ddots & a_1 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_n & \cdots & \cdots & a_k \end{bmatrix}.$$

At the same time, a PseudoHankel matrix $\mathrm{Pseudohank}(a,k)$ is defined as an $n \times n$ modified Hankel matrix given by

$$\mathrm{Pseudohank}(a, k) = \left[O, Z^k a, \ldots, Z^{n-2} a, Z^{n-1} a + (Z^{n-1})^T a, (Z^{n-2})^T a, \ldots, (Z^{n-k+1})^T a, O \right].$$

As we all know, a matrix with constant skew diagonals is called Hankel. Therefore, it is easy to see that $\mathrm{Pseudohank}(a,k)$ is formed from a Hankel matrix with relation to vector a by replacing both the first column and last column with zero vectors. For example,

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \Rightarrow \mathrm{Pseudohank}(a, 3) = \begin{bmatrix} 0 & a_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a_1 & 0 \\ 0 & a_1 & a_2 & 0 \end{bmatrix}.$$

As reported in [35], we use the notations $\mathrm{TOEP}(A,k)$ and $\mathrm{PseudoHANK}(A,k)$ for similar definitions with block matrices. With these notations,

$$A=\begin{bmatrix}A_1\\A_2\\A_3\\A_4\end{bmatrix}\Rightarrow \mathrm{TOEP}(A,2)=\begin{bmatrix}A_2 & A_1 & 0 & 0\\A_3 & A_2 & A_1 & 0\\A_4 & A_3 & A_2 & A_1\\0 & A_4 & A_3 & A_2\end{bmatrix}$$

and

$$\mathrm{Pseudohank}(A,3)=\begin{bmatrix}0 & A_4 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & A_1 & 0\\0 & A_1 & A_2 & 0\end{bmatrix}.$$

If $B\in\mathbb{R}^{m_1\times n_1}$ and $C\in\mathbb{R}^{m_2\times n_2}$, then their Kronecker product is defined as $B\otimes C$ which is an $m_1\times n_1$ block matrix whose (i,j) block is the $m_2\times n_2$ matrix $b_{ij}C$. We require the following properties of Kronecker products:

1. $(B\otimes C)^T=B^T\otimes C^T$.
2. $(B\otimes C)(E\otimes F)=(BE)\otimes(CF)$.

Let p be the true point spread function and the size of the original image be $n\times n$. It is often the case that the PSF array p has much smaller dimensions than the associated image arrays. For the convenience of presentation, we often need to pad with zeros so that it has the same dimension as the original image. Here the zero padding is done by keeping p in the upper left corner of the big PSF P. Hence both p and P have the same center; see [21] for details. Suppose the center of P is at (i,j) and let p_k^T be the kth row of P, and define

$$T=\begin{bmatrix}T_1\\T_2\\\vdots\\T_n\end{bmatrix},\quad H=\begin{bmatrix}H_1\\H_2\\\vdots\\H_n\end{bmatrix},$$

where

$$T_k=\mathrm{toep}(p_k,j),\quad H_k=\mathrm{Pseudohank}(p_k,j).$$

In Kronecker product approximations, matrices are sometimes regarded as vectors and vectors are sometimes made into matrices. To be precise about these reshaping the vec operation and tilde transformation will be used. If $X\in\mathbb{R}^{m_1\times n_1}$, then vec (X) is an n_1m_1-dimension column vector obtained by stacking columns of X. Let A be a block matrix

$$A=\begin{bmatrix}A_{11} & \cdots & A_{1n_1}\\\vdots & & \vdots\\A_{m_11} & \cdots & A_{m_1,n_1}\end{bmatrix}.$$

The tilde transform of A is a matrix which is obtained by rearranging A with the vec operation as follows

$$\widetilde{A}=\mathrm{tilde}(A)=\begin{bmatrix}\mathrm{vec}(A_{11})^T\\\vdots\\\mathrm{vec}(A_{m_11})^T\\\vdots\\\mathrm{vec}(A_{1n_1})^T\\\vdots\\\mathrm{vec}(A_{m_1,n_1})^T\end{bmatrix}.$$

3. Whole-sample symmetric BCs

In this section, we consider how to obtain the blurring matrix K in (1) for the whole-sample symmetric BCs. For convenience, we begin with the one-dimensional deblurring problem. Let the original signal

$$\hat{f}=(\ldots,f_{-m+1},\ldots,f_0,f_1,\ldots,f_n,f_{n+1},\ldots,f_{n+m},\ldots)^T$$

and the PSF be given by

$$h=(\ldots,0,0,h_{-m},h_{-m+1},\ldots,h_0,h_1,\ldots,h_{m-1},h_m,0,0\ldots)^T,$$

with center h_0 and $\sum_{j=-m}^{m} h_j = 1$. The convolution of h and $\hat{f}$ adding the noise η leads to the blurred signal g, with $g_i = \sum_{j=-\infty}^{\infty} h_{i-j} f_j + \eta_i$. In matrix form, we have

$$\begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix} = \begin{bmatrix} h_m & \cdots & h_0 & \cdots & h_{-m} & & & & \\ & h_m & \ddots & h_0 & \cdots & h_{-m} & & 0 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & 0 & & & h_m & \cdots & h_0 & \cdots & h_{-m} \\ & & & & & h_m & \cdots & h_0 & \cdots & h_{-m} \end{bmatrix} \begin{bmatrix} f_{-m+1} \\ f_{-m+2} \\ \vdots \\ f_0 \\ f_1 \\ \vdots \\ f_n \\ f_{n+1} \\ \vdots \\ f_{n+m-1} \\ f_{n+m} \end{bmatrix} + \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix}. \tag{2}$$

The purpose of the signal restoration is to recover the vector $f = (f_1, \ldots f_n)^T$ given the PSF h and a blurred signal $g = (g_1, \ldots, g_n)^T$. Thus the blurred and noisy signal g is determined not only by f, but also by $(f_{-m+1}, \ldots, f_0)^T$ and $(f_{n+1}, \ldots, f_{n+m})^T$. For the whole-sample symmetric BCs, the data outside f satisfies

$$\begin{cases} f_0 = f_2 \\ \vdots \\ f_{-m+1} = f_{m+1} \end{cases} \quad \text{and} \quad \begin{cases} f_{n+1} = f_{n-1} \\ \vdots \\ f_{n+m} = f_{n-m} \end{cases}.$$

By straightforward computation, it is not difficult to see that (2) becomes

$$Kf = g - \eta,$$

where K is an $n \times n$ Toeplitz-plus-PseudoHankel matrix given by

$$K = \begin{bmatrix} h_0 & \cdots & h_{-m} & & & \\ \vdots & \ddots & \ddots & \ddots & & \\ h_m & \ddots & \ddots & \ddots & h_{-m} & \\ & \ddots & \ddots & \ddots & \vdots & \\ & & h_m & \cdots & h_0 & \end{bmatrix} + \begin{bmatrix} 0 & h_1 & \cdots & h_m & O & & & & 0 \\ \vdots & \vdots & \iddots & & & & & & \\ 0 & h_m & & & & & O & & \vdots \\ & & & & \vdots & & & & \\ \vdots & & O & & & & & h_{-m} & 0 \\ & & & & & & & \iddots & \vdots \\ 0 & & & & O & h_{-m} & \cdots & h_{-1} & 0 \end{bmatrix}.$$

This result may be extended in a nature way to two-dimensional image restoration problems. In the case of n-by-n images, the resulting matrix K has a block Toeplitz-plus-PseudoHankel with Toeplitz-plus-PseudoHankel blocks structure. Using the notations in Section 2, we can formulate the resulting matrix $K \in \mathbb{R}^{n^2 \times n^2}$ under the whole-sample symmetric BCs as

$$K = K_{tt} + K_{th} + K_{ht} + K_{hh},$$

where K_{tt} = TOEP(T, i), K_{th} = TOEP(H, i), K_{ht} = PseudoHANK(T, i) and K_{hh} = PseudoHANK(H, i). Using the tilde transformation, we have

$$\widetilde{K} = \widetilde{K}_{tt} + \widetilde{K}_{th} + \widetilde{K}_{ht} + \widetilde{K}_{hh}.$$

4. Kronecker product approximation

Similar to [25,35,41], we hope to obtain a Kronecker product approximation of K arising from the whole-sample symmetric BCs. Given $s \in [1, \text{rank}(P)]$, all that we need is to find the $2s$ vectors a_k and b_k ($k = 1, \ldots, s$) of size n such that the matrices

$$A_{tk} = \text{toep}(a_k, i), \quad A_{hk} = \text{Pseudohank}(a_k, i),$$
$$B_{tk} = \text{toep}(b_k, j), \quad B_{hk} = \text{Pseudohank}(b_k, j)$$

minimize $\left\| K - \sum_{k=1}^{s} (A_{tk} + A_{hk}) \otimes (B_{tk} + B_{hk}) \right\|_F$.

Let C_k be an $n \times n$ Toeplitz-plus-Hankel-minus-1 rank matrix related with parameter k given by

$$C_k = C^t + C^h,$$

where C^t is an $n \times n$ symmetric Toeplitz matrix with its first row $(n,0,1,0,1,\ldots)$, C^h is an $n \times n$ Hankel-minus-1 rank matrix with $C^h(i,j) = 1$ for $i + j = 2k$ $(i \neq j)$ and zeros for others. In the following we have our main theorem.

Theorem 1. *Let P be an $n \times n$ PSF with center at (i,j), K be the blurring matrix arising from the whole-sample symmetric BCs and A_{tk}, A_{hk}, B_{tk}, B_{hk} be defined as above. Given $s \in (1, \mathrm{rank}(P))$, it has*

$$\left\| K - \sum_{k=1}^{s} (A_{tk} + A_{hk}) \otimes (B_{tk} + B_{hk}) \right\|_F = \left\| R_i P R_j - \sum_{k=1}^{s} (R_i a_k)(R_j b_k)^T \right\|_F,$$

where for $k = i$ and $j\left(i, j \leqslant \frac{n}{2}\right)$, R_k is the Cholesky factor of the $n \times n$ matrix C_k defined as above.

Proof. For convenience, we only prove the case $s = 1$. Similar techniques can be used to show other cases. Hence we only need to find four matrices A_{t1}, A_{h1}, B_{t1} and B_{h1} so that $\|K - (A_{t1} + A_{h1}) \otimes (B_{t1} + B_{h1})\|_F$ is minimized, where $A_{t1} = \mathrm{toep}(a_1, i)$, $A_{h1} = \mathrm{Pseudohank}(a_1, i)$, $B_{t1} = \mathrm{toep}(b_1, j)$, $B_{h1} = \mathrm{Pseudohank}(b_1, j)$.

Let p_k^T be the kth row of P. Then

$$K_{ht} = \mathrm{PseudoHANK}(T, i) = \begin{bmatrix} O & \mathrm{toep}(p_{i+1}, j) & \cdots & \mathrm{toep}(p_n, j) & O & \cdots & O \\ & \vdots & \iddots & & & & \\ \vdots & \mathrm{toep}(p_n, j) & & & & O & \vdots \\ \vdots & O & & & & \mathrm{toep}(p_1, j) & \vdots \\ & & & & \iddots & \vdots & \\ O & \cdots & O & \mathrm{toep}(p_1, j) & \cdots & \mathrm{toep}(p_{i-1}, j) & O \end{bmatrix}$$

Let $K_{ht}^{(k)}$ be the kth block column of K_{ht}. We may immediately obtain that

$$\widetilde{K}_{ht}^{(k)} = \begin{bmatrix} \mathrm{vec}(\mathrm{toep}(p_{i+k-1}, j))^T \\ \vdots \\ \mathrm{vec}(\mathrm{toep}(p_n, j))^T \\ O^T \\ \vdots \\ O^T \end{bmatrix}, \quad \text{for} \quad 2 \leqslant k \leqslant n - i.$$

Set $D_{t,k}$ and $D_{h,k}$ be block diagonal matrices given by

$$D_{t,k} = \mathrm{diag}\left[Z^{k-1}, Z^{k-2}, \ldots, Z^1, Z^0, Z^T, \ldots, (Z^{n-k})^T\right] \in \mathbb{R}^{n^2 \times n^2}$$

and

$$D_{h,k} = \mathrm{diag}\left[O, Z^k, \ldots, Z^{n-2}, Z^{n-1} + (Z^{n-1})^T, (Z^{n-2})^T, \ldots, (Z^{n-k+1})^T, O\right] \in \mathbb{R}^{n^2 \times n^2}.$$

Then we obtain

$$\widetilde{K}_{ht}^{(k)} = \begin{bmatrix} p_{i+k-1}^T & \cdots & p_{i+k-1}^T \\ \vdots & & \vdots \\ p_n^T & \cdots & p_n^T \\ O^T & \cdots & O^T \\ \vdots & & \vdots \\ O^T & \cdots & O^T \end{bmatrix} \begin{bmatrix} (Z^{j-1})^T & & & & & \\ & \ddots & & & & \\ & & Z^0 & & & \\ & & & Z & & \\ & & & & \ddots & \\ & & & & & Z^{n-j} \end{bmatrix} = Z^{i+k-2}[P, \cdots, P] D_{t,j}^T, \quad \text{for} \quad 2 \leqslant k \leqslant n - i.$$

Similar techniques can show that

$$\widetilde{K}_{ht}^{(k)} = (Z^{2n-i-k})^T [P, \cdots, P] D_{t,j}^T, \quad \text{for} \quad n - i + 2 \leqslant k \leqslant n - 1,$$
$$\widetilde{K}_{ht}^{(k)} = \left(Z^{i+k-2} + (Z^{2n-i-k})^T\right) [P, \cdots, P] D_{t,j}^T, \quad \text{for} \quad k = n - i + 1$$

and

$$\widetilde{K}_{ht}^{(k)} = O, \quad \text{for} \quad k = 1, n.$$

As a consequence, we further conclude that

$$\widetilde{K}_{ht} = \begin{bmatrix} \widetilde{K_{ht}^{(1)}} \\ \vdots \\ \widetilde{K_{ht}^{(k)}} \\ \vdots \\ \widetilde{K_{ht}^{(n)}} \end{bmatrix} = \begin{bmatrix} O[P,\ldots,P]D_{t,j}^T \\ Z^i[P,\ldots,P]D_{t,j}^T \\ \vdots \\ Z^{n-2}[P,\ldots,P]D_{t,j}^T \\ (Z^{n-1}+(Z^{n-1})^T)[P,\ldots,P]D_{t,j}^T \\ (Z^{n-2})^T[P,\ldots,P]D_{t,j}^T \\ \vdots \\ (Z^{n-i+1})^T[P,\ldots,P]D_{t,j}^T \\ O[P,\ldots,P]D_{t,j}^T \end{bmatrix} = D_{h,i}\widehat{P}D_{t,j}^T,$$

where

$$\widehat{P} = \begin{bmatrix} P & \cdots & P \\ \vdots & & \vdots \\ P & \cdots & P \end{bmatrix}.$$

The following statements can be obtained similarly

$$\widetilde{K}_{tt} = D_{t,i}\widehat{P}D_{t,j}^T, \quad \widetilde{K}_{th} = D_{t,i}\widehat{P}D_{h,j}^T, \quad \widetilde{K}_{hh} = D_{h,i}\widehat{P}D_{h,j}^T.$$

Therefore,

$$\widetilde{K} = \widetilde{K}_{tt} + \widetilde{K}_{th} + \widetilde{K}_{ht} + \widetilde{K}_{hh} = (D_{t,i} + D_{h,i})\widehat{P}(D_{t,j}^T + D_{h,j}^T) = (D_{t,i} + D_{h,i})\widehat{P}(D_{t,j} + D_{h,j})^T.$$

For a general block matrix, Loan and Pitsianis [32] have proved that

$$\|K - (A_{t1} + A_{h1}) \otimes (B_{t1} + B_{h1})\|_F = \left\|\widetilde{K} - \mathrm{vec}(A_{t1} + A_{h1}) \cdot \mathrm{vec}(B_{t1} + B_{h1})^T\right\|_F.$$

By a simple computation, it is not difficult to observe that

$$\mathrm{vec}(A_{t1} + A_{h1}) = (D_{t,i} + D_{h,i}) \begin{bmatrix} a_1 \\ \vdots \\ a_1 \end{bmatrix}$$

and

$$\mathrm{vec}(B_{t1} + B_{h1})^T = [b_1^T, \ldots, b_1^T](D_{t,j} + D_{h,j})^T.$$

Thus, we have

$$\widetilde{K} - \mathrm{vec}(A_{t1}+A_{h1}) \cdot \mathrm{vec}(B_{t1}+B_{h1})^T = (D_{t,i}+D_{h,i})(\widehat{P} - \begin{bmatrix} a_1 \\ \vdots \\ a_1 \end{bmatrix}[b_1^T, \cdots, b_1^T])(D_{t,j}+D_{h,j})^T = (D_{t,i}+D_{h,i}) \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix}(P - a_1 b_1^T)[I, \cdots, I](D_{t,j}+D_{h,j})^T$$

Taking into account the QR factorizations

$$(D_{t,i} + D_{h,i}) \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} = Q_i R_i \quad \text{and} \quad (D_{t,j} + D_{h,j}) \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} = Q_j R_j,$$

We have

$$\|K - (A_{t1} + A_{h1}) \otimes (B_{t1} + B_{h1})\|_F = \left\| (D_{t,i} + D_{h,i}) \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} (P - a_1 b_1^T)[I, \ldots, I](D_{t,j} + D_{h,j})^T \right\|_F = \|R_i(P - a_1 b_1^T)R_j^T\|_F.$$

In the following we consider the computation of the matrices R_i and R_j. Firstly, construct a matrix W_k as

$$W_k = (D_{t,k} + D_{h,k}) \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix}.$$

Then

$$W_k = \begin{bmatrix} Z^{k-1} \\ Z^{k-2} + Z^k \\ \vdots \\ Z^1 + Z^{2k-3} \\ Z^0 + Z^{2k-2} \\ Z^T + Z^{2k-1} \\ (Z^T)^2 + Z^{2k} \\ \vdots \\ (Z^{n-2k})^T + Z^{n-2} \\ (Z^{n-2k+1})^T + Z^{n-1} + (Z^{n-1})^T \\ (Z^{n-2k+2})^T + (Z^{n-2})^T \\ \vdots \\ (Z^{n-k-1})^T + (Z^{n-k+1})^T \\ (Z^{n-k})^T \end{bmatrix}.$$

By directly computing and carefully rearranging, it is not difficult to show that

$$R_k^T R_k = W_k^T W_k = Z_0 Z_0 + \sum_{m=1}^{n-1} \left((Z^m)^T Z^m + Z^m (Z^m)^T \right) + \sum_{m=k}^{n-k+1} \left(Z^{2k} + (Z^{2k})^T \right) + \sum_{m=1}^{k-1} \left((Z^{k-1-m})^T Z^{k-1+m} + Z^{n-k+m} (Z^{n-k-m})^T \right) + \sum_{m=1}^{k-1} \left((Z^{k-1+m})^T Z^{k-1-m} + Z^{n-k-m} (Z^{n-k+m})^T \right).$$

Taking into account $k = i,\ j(i,j \leqslant \frac{n}{2})$ and making use of some properties of the shift matrix Z, we obtain

$$R_k^T R_k = \begin{bmatrix} n & 0 & 1 & 0 & 1 & \cdots \\ 0 & n & 0 & 1 & 0 & \ddots \\ 1 & 0 & n & 0 & \ddots & \ddots \\ 0 & 1 & \ddots & \ddots & \ddots & \ddots \\ 1 & 0 & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} + \begin{bmatrix} & & & & & & k & & 2k-1 & \\ & & & & & & \downarrow & & \downarrow & \\ & & & & & & & & 1 & O \\ & & & & & & & \iddots & & \\ & & & & & 1 & & & & \\ & & & & \iddots & & & & & \\ & & & 1 & & & 0 & & & \vdots \\ & & \iddots & & & & & & & \\ 1 & & & & & & & & & \\ O & & & & \cdots & & & & O & \end{bmatrix} \leftarrow 2k-1$$

Theorem 1 is now completely proved. □

Remark. In image restoration, the dimension of the true PSF array p frequently is much smaller than that of the original image array. Since the zero padding PSF P has the same center with the true PSF p at $(i,j),\ i \leqslant \frac{n}{2}$ and $j \leqslant \frac{n}{2}$ are almost always satisfied. Actually, even if $i > \frac{n}{2}$ and $j > \frac{n}{2}$, we also obtain in C_k the symmetric Toeplitz matrix C^t whose first row is $[n,0,1,0,1\ldots,0,\ldots,0]$. By this theorem, we observe that the key to compute the nearest Kronecker product approximation of the blurring matrix K is to find $2s$ vectors a_k and $b_k\ (k = 1,\ldots,s)$ of size n and these vectors can be obtained simply by computing the SVD of the $n \times n$ matrix $R_i P R_j^T$. Especially, when the dimension of the true PSF array p is smaller than that of the zero padding PSF P, we have $R_i P R_j^T = \begin{bmatrix} R_{i1} p R_{j1}^T & O \\ O & O \end{bmatrix}$, where $P = \begin{bmatrix} p & O \\ O & O \end{bmatrix}$, $n \times n$ upper triangular matrices $R_i = \begin{bmatrix} R_{i1} & R_{i2} \\ O & R_{i3} \end{bmatrix}$ and $R_j = \begin{bmatrix} R_{j1} & R_{j2} \\ O & R_{j3} \end{bmatrix}$. Hence, for matrix multiplication $R_i P R_j^T$, it only needs R_{i1}, R_{j1} and the small PSF p. Obviously it has great advantage to only work on small matrices.

Based on Theorem 1, the algorithm for computing the Kronecker product approximation of K is given as follows.

Algorithm 1. Find $K \approx \sum_{k=1}^{s}(A_{tk}+A_{hk})\otimes(B_{tk}+B_{hk})$

1. Compute R_{i1} and R_{j1}
2. Construct $p_r \doteq R_{i1}pR_{j1}$
3. Compute the SVD of the small matrix $p_r = \sum \sigma_k u_k v_k^T$.
4. Compute the $2s$ vectors: $\hat{a}_k = \sqrt{\sigma_k}R_{i1}^{-1}u_k$ and $\hat{b}_k = \sqrt{\sigma_k}R_j^{-1}v_k$ and construct the $2s$ vectors of the approximation: $a_k = \begin{pmatrix}\hat{a}_k \\ O\end{pmatrix}$ and $b_k = \begin{pmatrix}\hat{b}_k \\ O\end{pmatrix}$
5. Construct the matrices: A_{tk} = toep(a_k,i), A_{hk} = Pseudohank(a_k,i), B_{tk} = toep(b_k,j) and B_{hk} = Pseudohank(b_k,j).

It is interesting to note that, in most cases, the computational operations of Algorithm 1 in this paper and the algorithms in [35,25,41] for the Kronecker product approximation of K are very small, since all computations in these algorithms are implemented only at the upper left corner submatrices of the big matrices. In addition, if the dimension of the true PSF array p is just $n \times n$, that is $P = p$, then we have $R_{i1} = R_i,\ R_{j1} = R_j,\ a_k = \hat{a}_k$ and $b_k = \hat{b}_k$ in Algorithm 1. In [42], Pitsianis has shown that the Kronecker product approximation will be sensitive to perturbations or errors when the perturbed matrix of K has its two largest singular values close together. When the largest singular value is isolated, the approximation factors in Algorithm 1 will be very effective.

5. Implementation and numerical examples

In this section, we will focus our attention on the implementation of the Kronecker product approximation algorithm and some applications in image restoration. All the experiments given in this paper were performed in Matlab 7.0. The results were obtained by running the Matlab codes on an Intel Core (TM)2 Duo CPU (2.93 GHz, 2.93 GHz) computer with RAM of 2048 M.

In applications where K is very large, it is generally infeasible to calculate the SVD of K explicitly. We now describe how to construct an SVD approximation to K by the Kronecker product approximation of K. We rewrite the Kronecker product approximation of K as $K \approx \sum_{k=1}^{s}A_k \times B_k$, where $A_k = A_{tk} + A_{hk}$ and $B_k = B_{tk} + B_{hk}$. Given the SVDs of A_1 and B_1, $A_1 = U_1\Sigma_A V_1^T$ and $B_1 = U_2\Sigma_B V_2^T$, the SVD of K can be expressed as

$$K \approx U\Sigma V^T,$$

where $U = U_1 \otimes U_2$, $V = V_1 \otimes V_2$ and $\Sigma = \mathrm{diag}\left(U^T\left(\sum_{k=1}^{s}A_k \times B_k\right)V\right)$.

In fact, aside from the issue of boundary conditions, in image restoration, it is also difficult to solve the problem of noise sensitivity in the computed solution. Image restoration is a highly ill-posed inverse problem. A small amount of noise in the data can lead to enormous errors in the estimates. For this reason, many kinds of image restoration techniques have been proposed such as the truncated singular value decomposition (TSVD), Tikhonov regularization, truncated iterative methods, truncated total least squares and damped total least squares; see [21,24] for details. In this work, we apply the truncated singular value decomposition method.

Using the approximate SVD as above, the TSVD solution can be simplified to

$$F_t = V_2[S \cdot *(U_2GU_1)]V_1^T,$$

where $\mathrm{vec}(G) = g$, $\mathrm{vec}(F_t) = f_t$ and $\mathrm{vec}(S) = (\sigma_1^{-1},\ldots,\sigma_t^{-1},0,\ldots,0)^T$. The parameter t, $0 \leqslant t \leqslant N$, is called the truncation parameter and it determines the number of SVD components maintained in the regularized solution. Choosing an appropriate truncation parameter is a non-trivial issue. Several methods such as the discrepancy principle, the generalized cross validation and the L-curve criterion have been used to choose an appropriate value [15,20,21,48]. For our TSVD experiments, we use the generalized cross validation:

$$t = \arg\min_k G(k) = \arg\min_k \frac{\|Kf_k - g\|_2^2}{(N-k)^2},$$

where N is the length of vector g, $f_k = \mathrm{vec}(F_k)$ and $K \approx U\Sigma V^T$.

In test experiments, we hope that the natural boundary condition will contribute to the blur. Fortunately, this can be done by performing the blurring operation on a large true image from which a central part is cut out, and adding Gaussian white noise to the pixel values.

The aim of the first test is to give evidence of the effectiveness of the deblurring approach with the Kronecker product approximation algorithm in the case of the whole-sample symmetric BCs. The first test image we used is shown in Fig. 2. In the true 412×412 image, the original 256×256 image f to be deblurred is delimited by white lines. In this case the blurring function is a nonsymmetric PSF given by

$$h_{ij} = \begin{cases} \frac{c}{i+j+1} & \text{if } 0 \leqslant i,j \leqslant 13, \\ 0 & \text{otherwise}, \end{cases}$$

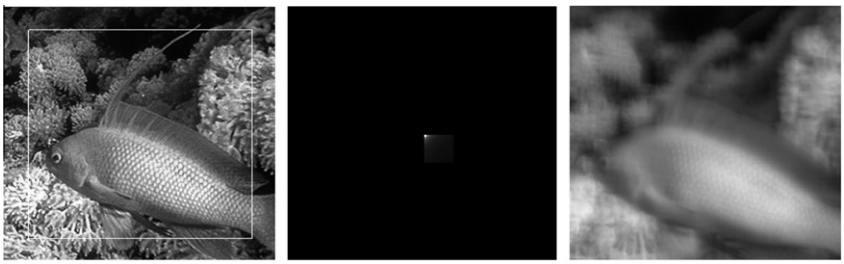
where c is the normalization constant such that $\sum_{ij} h_{ij} = 1$. Using the built-in MATLAB function randn, we add 1% Gaussian white noise to the blur data.

A simple quantitative measure of the restoration quality, applicable when the true object is known, is relative error. Relative error of the restored image is defined as $\frac{\|f - f_t\|_2}{\|f\|_2}$, where f is the original image and f_t is the TSVD restored image with truncation parameter t. In Table 1, truncation parameters and the relative errors of the corresponding computed solutions are shown for several values of s. Fig. 3 shows TSVD restorations for $s = 1, 2, 3, 4$ with the whole-sample symmetric boundary conditions. From this figure, we observe that our algorithm generates high quality restoration results and the results for the four values of s are visually indistinguishable.

In the following two tests, the same deblurring strategy based on the Kronecker product approximations will be applied with four different boundary conditions. Our aim is to compare the efficiency of employing the whole-sample symmetric BCs over the other three boundary conditions for some image restoration problems. In Test 2, we are dealing with a strongly nonsymmetric PSF arising from wavefront coding, where a cubic phase filter is used to improve depth of field resolution in light efficient wide aperture optical systems. Extending the depth of field of incoherent optical systems has been an active research topic for many years; see [14,35,50]. The data of Test 2 is shown in Fig. 4. The plot of the first 40 singular values of the PSF suggests that a Kronecker approximation of K with $s = 1$ should suffice. Our third blurring function we consider is a PSF caused by turbulence in atmosphere. This test problem is often popular in the literature concerned with astronomical image restoration [19,44,46]. The PSF we used in this test is a symmetric truncated Gaussian blur given by

$$h_{ij} = \begin{cases} ce^{-0.1(i^2+j^2)} & \text{if } |i-j| \leqslant 13, \\ 0 & \text{otherwise}, \end{cases}$$

where c is the normalization constant. The data of Test 3 is shown in Fig. 5. The separable property of the Gaussian blur shows that we only need a Kronecker approximation of K with $s = 1$. In the two tests, we add 0.5% and 0.2% Gaussian white noise to the two blurred images, respectively.

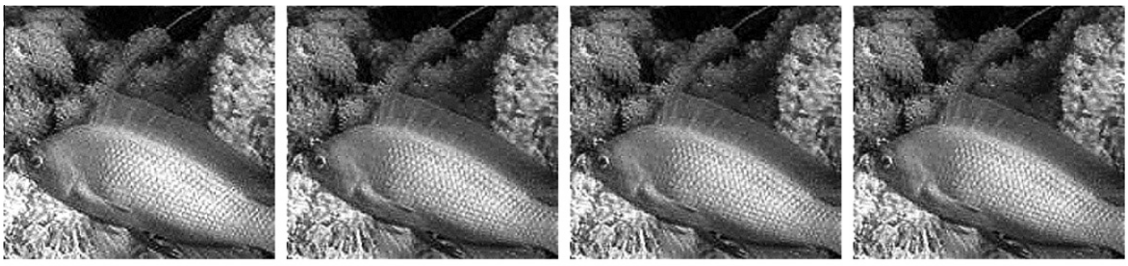


(a) True image (b) The nonsymmetric PSF (c) Blurred and noisy image

Fig. 2. True image, PSF and blurred image with noise of Test 1.

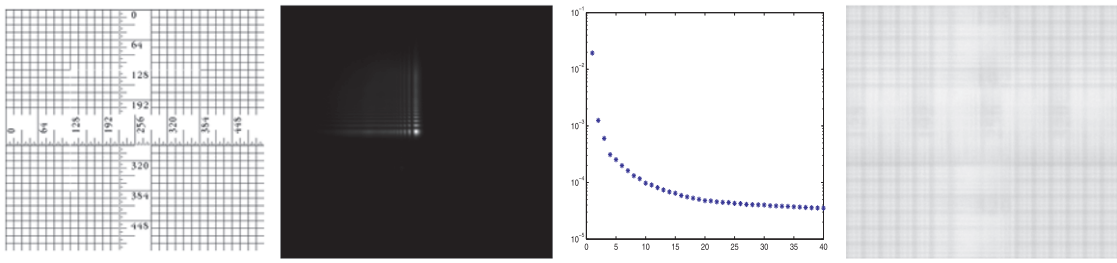
Table 1
Relative errors and truncation parameters with different parameter s.

	$s = 1$	$s = 2$	$s = 3$	$s = 4$
t (GCV)	24,461	21,889	21,835	21,899
Error	0.1611	0.1357	0.1354	0.1354



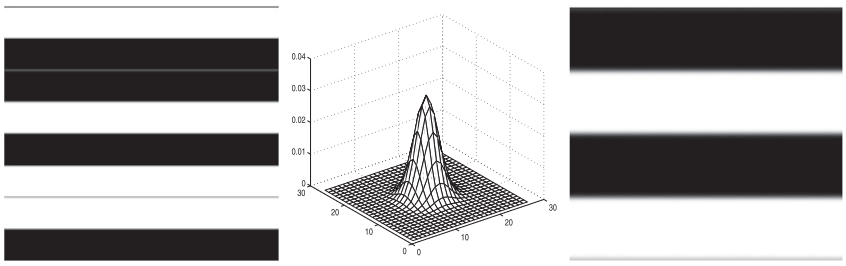
(a) s=1 (b) s=2 (c) s=3 (d) s=4

Fig. 3. TSVD restorations with the whole-sample symmetric BCs.



(a) True image (b) The nonsymmetric PSF (c) The first 40 singular values of the PSF (d) Blurred and noisy image

Fig. 4. True image, PSF and blurred image with noise of Test 2.



(a) True image (b) The truncated Guassian blur (c) Blurred and noisy image

Fig. 5. True image, PSF and blurred image with noise of Test 3.

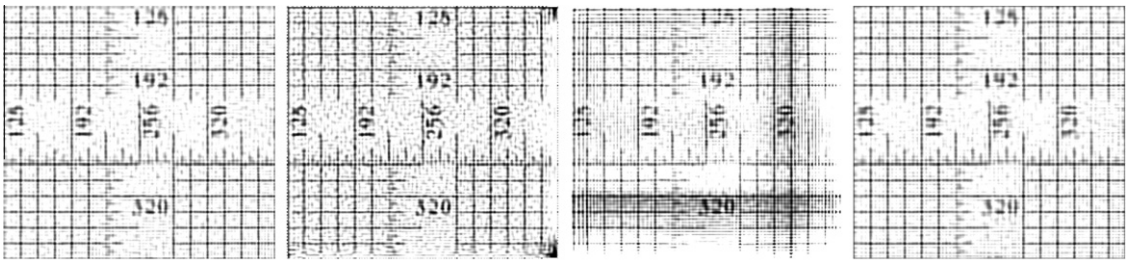
In Table 2, we give the results about relative errors and the choice of regularization parameters with the generalized cross validation. We see from Table 2 that regardless of symmetry properties of the PSF, the TSVD method based on the Kronecker product approximation algorithm under the whole-sample symmetric BCs is very efficient. In addition, it is obvious that by imposing the whole-sample symmetric BCs the relative error of the reconstructed image is the smallest. Since the operation for the deblurring strategy based on the Kronecker product approximation algorithm with four boundary conditions for image restoration problems are nearly the same, we don't show the CPU time for the two tests. Figs. 6 and 7 show the restored images with the TSVD method under four different boundary conditions for Test 2 and Test 3, respectively. As expected, the whole-sample symmetric BCs have addressed the problem of ringing effects at the image boundary. In addition, from these figures, it is obvious that reconstructions with the whole-sample symmetric BCs contain the least amount of ringing artifacts. However, It is easy to see from Fig. 6 that the digits in the restored image under the antireflexive BCs are clearer than under the other three boundary conditions. It is mainly because many entries of the restored digital image from the antireflexive BCs are become zero. In Fig. 7, we observe that the restored image of using the whole-sample symmetric BCs gives a higher quality than the other three boundary conditions. It should be noted that in all our experiments, using various test images, we observe that the whole-sample symmetric BCs perform somewhat comparably to the half-sample symmetric BCs and antireflexive BCs, in some cases comparing favorably and in other just the opposite. In almost all cases, the restored effect using the whole-sample symmetric BCs is better than the zero BCs. All results are not shown here, in this work we only report the two cases for comparison purposes.

In the fourth experiment, we compare the performance of the TSVD method based on the Kronecker product approximation algorithm with that of the CGLS method in 3D image restoration under the whole-sample symmetric BCs. The true image is $128 \times 128 \times 27$ simulated MRI of a human brain, available in the Matlab Image Processing Toolbox. Restoration of this image was used as a test problem in [36,46]. To produce the distorted image, we build an out-of-focus PSF using the function *psfDefocus* in [21], and convolve it with the MRI image, then add 0.5% Gaussian noise to the result. The test data is shown in Fig. 8.

In this test, we evaluate our method and the CGLS method using the peak signal-to-noise ratio (PSNR) which compares the restored image f with the original image f_{true}. It is defined by $\text{PSNR} = 10\log_{10}\frac{255^2 n^2}{\|f - f_{\text{true}}\|_2^2}$, where the size of the restored images is $n \times n$. Here the PSNR of the blurred and noisy image is 33.60 dB. Restored images using the whole-sample symmetric BCs appear in Fig. 9. On the left is an optimal approximate restoration using conjugate gradients on normal equations (CGLS) with the exact blurring matrix. For the CGLS method, we terminate the iterations as soon as an optimal approximate solution which satisfies the discrepancy principle has been found. The optimal approximate solution is obtained by 38

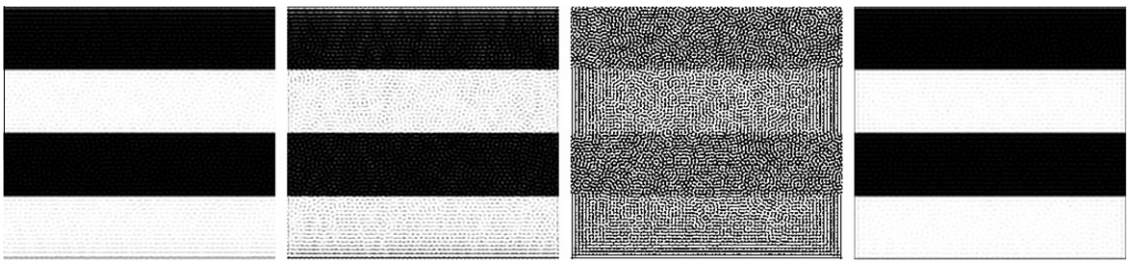
Table 2
Relative errors and the choice of regularization parameters.

	Half-sample symmetric	Antireflexive	Zero	Our method
Test two				
t (GCV)	6360	8712	6519	5226
Error	0.2401	0.2851	0.9530	0.2254
Test three				
t (GCV)	9456	13,667	10,502	19,704
Error	01362	0.3116	5.4641	0.1163



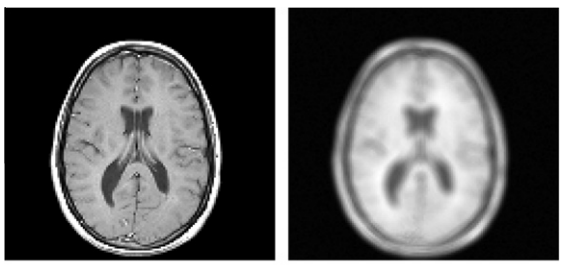
(a) Half-sample symmetric BCs (b) Antireflexive BCs (c) Zero BCs (d) Whole-sample symmetric BCs

Fig. 6. TSVD restorations with different boundary conditions.



(a) Half-sample symmetric BCs (b) Antireflexive BCs (c) Zero BCs (d) Whole-sample symmetric BCs

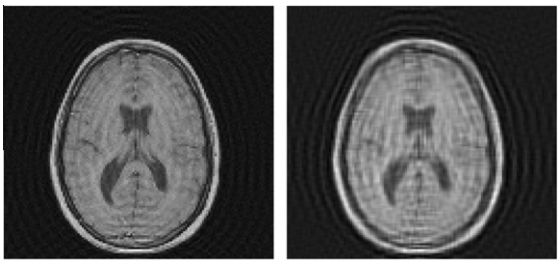
Fig. 7. TSVD restorations with different boundary conditions.



(a) True image (b) Blurred and noisy image

Fig. 8. True image and blurred image with noise of Test 4.

iterations of CGLS. This restoration has a PSNR of 40.81. On the right is the experimental restoration using the TSVD method with the Kronecker approximation for $s = 1$, with the truncation parameter chosen by the GCV method. The PSNR of this restoration is 36.54. However, there does not exist a precise and reliable stopping rule for choosing the optimal number of iterations for CGLS when the true object is unavailable. As reported in [41], the optimal status of the CGLS restoration is artificially produced, and the ability to achieve it is uncertain at best in experimental restorations. On the other hand, the GCV method predicts an appropriate truncation parameter for TSVD with the Kronecker approximation, and no subjective



(a) CGLS (b) TSVD with $s = 1$

Fig. 9. Restorations using the whole-sample symmetric BCs.

intervention is required. In addition, the CPU time for identifying the iteration count that yields an optimal approximate solution and producing the corresponding solution is 3.899 s, while a total of 0.115 s is needed to produce a Kronecker approximation and to obtain the Kronecker-based TSVD restoration.

6. Conclusion

In this paper, we investigate the use of the whole-sample symmetric BCs in image restoration. From a theoretical point of view, the whole-sample symmetric BCs are more common than the half-sample symmetric BCs in many image restoration issues. For the resulting matrix from the whole-sample symmetric BCs, a Kronecker product approximation algorithm is obtained. We also discuss the implementation of the TSVD method based on the Kronecker product approximation algorithm and some applications in image restoration. Numerical examples are given to illustrate the effectiveness of our method.

Acknowledgments

The authors would like to express their great thankfulness to the referees and Prof. Pedrycz for much constructive, detailed and helpful advice regarding revising this manuscript.

References

[1] F. Aghdasi, R. Ward, Reduction of boundary artifacts in image restoration, IEEE Trans. Image Process. 5 (4) (1996) 611–618.
[2] H. Andrew, B. Hunt, Digital Image Restoration, Prentice-Hall, Englewood Cliffs, NJ, 1977.
[3] M. Banham, A. Katsaggelos, Digital image restoration, IEEE Signal Process. Mag. 14 (2) (1997) 24–41.
[4] S. Bonettini, V. Ruggiero, An alternating extragradient method for total variation-based image restoration from Poisson data, Inverse Probl. 27 (2011) 095001. 26 pp.
[5] K. Boukerrou, L. Kurz, Suppression of "salt and pepper" noise based on Youden designs, Inform. Sci. 110 (1998) 217–235.
[6] J.F. Cai, S. Osher, Z.W. Shen, Split Bregman methods and frame based image restoration, Multiscale Model. Simul. 8 (2) (2010) 337–369.
[7] S.S. Capizzano, A note on anti-reflective boundary conditions and fast deblurring models, SIAM J. Sci. Comput. 25 (2003) 1307–1325.
[8] T.F. Chan, J. Shen, Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods, SIAM, Philadelphia, 2005.
[9] F. Chen, L. Yu, Image deblurring with odd symmetry discrete Neumann boundary condition, in: Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2007.
[10] M. Christiansen, M. Hanke, Deblurring methods using antireflective boundary conditions, SIAM J. Sci. Comput. 30 (2) (2008) 855–872.
[11] J. Chung, J.G. Nagy, An efficient iterative approach for large-scale separable nonlinear inverse problems, SIAM J. Sci. Comput. 31 (6) (2010) 4654–4674.
[12] M. Donatelli, S.S. Capizzano, Anti-reflective boundary conditions and re-blurring, Inverse Probl. 21 (2005) 169–182.
[13] Y. Dong, M. Hintermüller, M.M. Rincon-Camacho, Automated regularization parameter selection in multi-scale total variation models for image restoration, J. Math. Imaging Vis. 40 (2011) 82–104.
[14] E. Dowski, W. Cathey, Extended depth of field through wavefront coding, Appl. Optics 34 (1995) 1859–1866.
[15] H. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer Academic Publishers, The Netherlands, 1996.
[16] Y.W. Fan, J.G. Nagy, Synthetic boundary conditions for image deblurring, Linear Algebra Appl. 434 (11) (2011) 2244–2268.
[17] A.I. González, M. Graña, J.R. Cabello, A. D'Anjou, F.X. Albizuri, Experimental results of an evolution-based adaptation strategy for VQ image filtering, Inform. Sci. 176 (2006) 2988–3010.
[18] A.E. Hamidi, M. Ménard, M. Lugiez, C. Ghannam, Weighted and extended total variation for image restoration and decomposition, Pattern Recognit. 43 (2010) 1564–1576.
[19] M. Hanke, J.G. Nagy, C.R. Vogel, Quasi-Newton approach to nonnegative image restorations, Linear Algebra Appl. 316 (2000) 223–236.
[20] P.C. Hansen, Rank Defficient and Discrete Ill-posed Problems: Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1997.
[21] P.C. Hansen, J.G. Nagy, D.P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.
[22] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, SIAM, Philadelphia, 2010.
[23] G. Jeon, M. Anisetti, V. Bellandi, E. Damiani, J. Jeong, Designing of a type-2 fuzzy logic filter for improving edge-preserving restoration of interlaced-to-progressive conversion, Inform. Sci. 179 (13) (2009) 2194–2207.
[24] J. Kamm, Singular value decomposition-based methods for signal and image processing, Ph. D. Thesis, Southern Methodist University, USA, 1998.
[25] J. Kamm, J.G. Nagy, Optimal Kronecker product approximation of block Toeplitz matrices, SIAM J. Matrix Analysis Appl. 22 (2000) 155–172.
[26] R.L. Lagendijk, J. Biemond, D.E. Boekee, Identification and restoration of noisy blurred images using the expectation-maximization algorithm, IEEE Trans. Acoust. Speech Signal Process. 38 (7) (1990) 1180–1191.
[27] K.T. Lay, A.K. Katsaggelos, Identification and restoration based on the expectation-maximization algorithm, Opt. Eng. 29 (5) (1990) 436–445.
[28] X. Li, Fine-granularity and spatially-adaptive regularization for projection-based image deblurring, IEEE Trans. Image Process. 20 (4) (2011) 971–983.

[29] C.L. Lim, B. Honarvar, K.H. Thung, R. Paramesran, Fast computation of exact Zernike moments using cascaded digital filters, Inform. Sci. 181 (17) (2011) 3638–3651.
[30] T.C. Lin, A new adaptive center weighted median filter for suppressing impulsive noise in images, Inform. Sci. 177 (2007) 1073–1087.
[31] T.C. Lin, Switching-based filter based on Dempsters combination rule for image processing, Inform. Sci. 180 (2010) 4892–4908.
[32] C.F.V. Loan, N.P. Pitsianis, Approximation with Kronecker products, in: Moonen MS, Golub GH (Eds.), Linear Algebra for Large Scale and Real Time Applications, Kluwer, Dordrecht, 1993, pp. 293–314.
[33] S. Martucci, Symmetric convolution and the discrete sine and cosine transforms, IEEE Trans. Signal Process. 42 (1994) 1038–1051.
[34] O.V. Michailovich, An iterative shrinkage approach to total-variation image restoration, IEEE Trans. Image Process. 20 (5) (2011) 1281–1299.
[35] J.G. Nagy, M.K. Ng, L. Perrone, Kronecker product approximations for image restoration with reflexive boundary conditions, SIAM J. Matrix Anal. Appl. 25 (2003) 829–841.
[36] J.G. Nagy, K. Palmer, L. Perrone, Iterative methods for image deblurring: a Matlab object-oriented approach, Numer. Algor. 36 (2004) 73–93. <http://www.mathcs.emory.edu/nagy/RestoreTools>.
[37] M.K. Ng, R. Chan, W. Tang, A fast algorithm for deblurring models with Neumann boundary conditions, SIAM J. Sci. Comput. 21 (2000) 851–866.
[38] M.K. Ng, P. Weiss, X. Yuan, Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods, SIAM J. Sci. Comput. 32 (5) (2010) 2710–2736.
[39] H. Nobuhara, B. Bede, K. Hirota, On various eigen fuzzy sets and their application to image reconstruction, Inform. Sci. 176 (2006) 2988–3010.
[40] Z. Pang, Y. Yang, A projected gradient algorithm based on the augmented Lagrangian strategy for image restoration and texture extraction, Image Vision Comput. 29 (2011) 117–126.
[41] L. Perrone, Kronecker product approximations for image restoration with anti-reflective boundary conditions, Numer. Lin. Alg. Appl. 13 (2006) 1–22.
[42] N.P. Pitsianis, The Kronecker product in approximation and fast transform generation, Ph. D. Thesis, Cornell University, Ithaca, NY, 1997.
[43] S.J. Reeves, R.M. Mersereau, Blur identification by the method of generalized cross-validation, IEEE Trans. Image Process. 1 (7) (1992) 301–311.
[44] M. Rojas, T. Steihaug, An interior-point trust-region-based method for large-scale non-negative regularization, Inverse Probl. 18 (2002) 1291–1307.
[45] E. Shaked, O. Michailovich, Iterative shrinkage approach to restoration of optical imagery, IEEE Trans. Image Process. 20 (2) (2011) 405–416.
[46] V.N. Strakhov, S.V. Vorontsov, Digital image deblurring with SOR, Inverse Probl. 24 (2) (2008) p.02502, doi:10.1088/0266-5611/24/2/025024.
[47] G. Strang, The discrete cosine transform, SIAM Rev. 41 (1999) 135–147.
[48] C.R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.
[49] Z. Wang, A.C. Bovik, Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures, IEEE Signal Process. Mag. 26 (1) (2009) 98–117.
[50] W. Zhang, Z. Ye, T. Zhao, Y. Chen, F. Yu, Point spread function characteristics analysis of the wavefront coding system, Optics Exp. 15 (4) (2007) 1543–1552.