IMAGE RECONSTRUCTION WITH MULTISENSORS

by

Wun-Cheung TANG

Thesis
Submitted to the Faculty of the Graduate School of
The Chinese University of Hong Kong
(Division of Mathematics)

In partial fulfillment of the requirements
for the Degree of
Master of Philosophy

June, 1998

DECLARATION

The author declares that this thesis represents his own work based on the ideas suggested by Prof. Raymond H.F. Chan, the author's supervisor. All the work is done under the supervision of Prof. Raymond H.F. Chan during the period 1996-1998 for the degree of Master of Philosophy at The Chinese University of Hong Kong. The work submitted has not been previously included in a thesis, dissertation or report submitted to any institution for a degree, diploma or other qualification.

Wun-Cheung TANG

To

My parents

ACKNOWLEDGEMENT

I wish to express my sincere and deepest gratitude to my supervisor, Prof. Raymond H.F. Chan, for his inspired guidance, constant encouragement and help throughout the period of my M.Phil. studies and in the preparation of this thesis. I am deeply grateful to Dr. Michael K.P. Ng of the Department of Mathematics, The University of Hong Kong for his beneficial discussions and suggestions. I would also like to thank Drs. X.Q. Jin, F.R. Lin, H.W. Sun, my colleagues Mr. C.F. Chan, H.C. Chan, C.P. Cheung, W.K. Ching, L.L. Heung, K.W. Mak, W.F. Ng, T.M. Tso, C.K. Wong, and H.M. Zhou for their many helpful discussions.

CONTENTS

摘　要

本畢業論文包括了兩項關於圖像改進和圖像恢復的最新工作。第一篇文章研究如何從多個低解像度的，帶有位移和小像素擾動誤差的粗糙圖像，重新構造出高解像度的圖像。這問題對應之構造算子H是一個空間變分算子。在此文章中，與通常之零邊值條件(對應於漆黑背景)不同，我們考慮了Neumann邊值條件(對應於原像在邊界上的反射)。H的離散矩陣是一個塊-Toeplitz(BTTB)矩陣。我們應用以餘弦變換爲預處理算子的預條件共軛梯度法(PCG)解其離散問題。初步結果顯示，由Neumann邊值條件所重造之圖像比零邊值條件所重造之圖像清晰，而且PCG方法的收斂速度非常快。

在第二篇文章中，我們提出了新的圖像恢復模型。在零邊值條件下，對應之歪曲矩陣在一維情況下是一個Toeplitz矩陣，而在二維情況下則是一個BTTB矩陣。求解此矩陣的逆，計算量非常大(尤其在二維情況)。使用週期邊值條件，對應之歪曲矩陣爲循環矩陣,這種矩陣能被快速離散Fourier變換所對角化。但是,這兩種邊值條件都容易引致邊值謬誤。在本文中，我們提出使用Neumann邊值條件，其對應之歪曲矩陣爲Toeplitz加Hankel矩陣。對於對稱的歪曲函數，我們證明了其歪曲矩陣能被離散餘弦變換所對角化。因此求這種矩陣的逆的計算量比零或週期邊值條件小的多。況且，實驗結果也顯示這種邊值條件所引致的邊值謬誤大大小於其他兩種邊值條件所引致的邊值謬誤。

本畢業論文由下列兩篇文章組成，在本畢業論文中稱作文章A及文章B。

文章 A: Raymond H. Chan, Tony F. Chan, Michael K. Ng, Wun-Cheung Tang and Chiu-Kwong Wong. *Preconditioned Iterative Methods for High-resolution Image Reconstruction with Multisensors*, published in the proceedings of SPIE Symposium on Advace Signal Processing Algorithms, Archietectures and Implementations, Vol. 3461, San Diego CA, July 1998, Ed: F. Luk

文章 B: Raymond H. Chan, Michael K. Ng and Wun-Cheung Tang. *Deblurring Models with Neumann Boundary Conditions*, submitted to SIAM J. Sci. Comput.

Abstract

This thesis contains two recent works on image enhancement and image deblurring. In the first paper, we study the problem of reconstructing a high-resolution image from multiple undersampled, shifted, degraded frames with subpixel displacement errors. The corresponding reconstruction operator $\mathcal{H}$ is a spatially variant operator. In this paper, instead of using the usual zero boundary condition (corresponding to a dark background outside the scene), the Neumann boundary condition (corresponding to a reflection of the original scene at the boundary) is imposed on the images. The resulting discretization matrix of $\mathcal{H}$ is a block-Toeplitz-Toeplitz-block-like matrix. We apply the preconditioned conjugate gradient (PCG) method with cosine transform preconditioners to solve the discrete problems. Preliminary results show that the image model under the Neumann boundary condition gives better reconstructed high-resolution images than that under the zero boundary condition, and the PCG method converges very fast.

In the second paper, we propose a new model in image deblurring. The blurring matrices obtained by using the zero boundary condition are Toeplitz matrices for 1-dimensional problems and block-Toeplitz-Toeplitz-block matrices for 2-dimensional cases. They are computationally intensive to invert especially in the block case. Using periodic boundary conditions, the matrices become (block) circulant and can be diagonalized by discrete Fourier transform matrices. However, both boundary conditions easily lead to boundary artifacts. In this paper, we propose to use the Neumann boundary condition. The resulting matrices are (block) Toeplitz-plus-Hankel matrices. We show that for symmetric blurring functions, these blurring matrices can always be diagonalized by discrete cosine transform matrices. Thus the cost of inversion is much cheaper than that of using zero or periodic boundary conditions. Moreover, experimental results also show that the boundary artifacts are much less prominent than that of using the other two boundary conditions.

The thesis is based on the following two papers, which will be referred in the text

as Paper A and Paper B.

Paper A Raymond H. Chan, Tony F. Chan, Michael K. Ng, Wun-Cheung Tang and Chiu-Kwong Wong, *Preconditioned Iterative Methods for High-resolution Image Reconstruction with Multisensors*, published in the Proceedings of SPIE Symposium on Advance Signal Processing Algorithms, Architectures, and Implementations, Vol 3461, San Diego CA, July 1998, Ed: F. Luk.

Paper B Raymond H. Chan, Michael K. Ng and Wun-Cheung Tang, *Deblurring Models with Neumann Boundary Conditions*, submitted to SIAM J. Sci. Comput.

Introduction

In this thesis, we present two recent works in image enhancement and image deblurring. Most image processing problems require solving a Toeplitz system, so we will first give an introduction on Toeplitz matrices. Fast algorithms for solving Toeplitz systems by the preconditioned conjugate gradient (PCG) method will then be discussed. In particular, we will introduce the optimal cosine transform preconditioner in §3. Since most problems in signal and image processing are ill-conditioned, we should apply regularization techniques to regularize the solution. We will discuss in §4 classical Tikhonov regularization and the L-curve method for choosing the regularization parameter. Finally, we will give a summary on the two papers in §5.

1 Toeplitz and Circulant Matrix

An $n \times n$ matrix $\mathbf{A}_n$ is said to be a *Toeplitz matrix* if

$$\mathbf{A}_n = \begin{pmatrix} a_0 & a_{-1} & \cdots & \cdots & \cdots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & \ddots & \vdots \\ \vdots & a_1 & a_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & \cdots & a_1 & a_0 \end{pmatrix}. \tag{1}$$

That is, a Toeplitz matrix is constant along its diagonals. Mathematically, if a_{ij} is the element of the i-th row and the j-th column of the matrix $\mathbf{A}_n$, then $a_{ij} = a_{i-j}$.

Two-dimensional case can be defined similarly. Let us consider an $n^2 \times n^2$ block

matrix with the following structure:

$$\mathbf{A}_{nn} = \begin{pmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \cdots & \mathbf{A}_{1,n} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{A}_{n-1,n-1} & \mathbf{A}_{n-1,n} \\ \mathbf{A}_{n-1} & \cdots & \cdots & \mathbf{A}_{n,n-1} & \mathbf{A}_{n,n} \end{pmatrix}, \tag{2}$$

where $\mathbf{A}_{i,j}$ are $n \times n$ matrices. $\mathbf{A}_{nn}$ is called a *block-Toeplitz-Toeplitz-block (BTTB) matrix* if $\mathbf{A}_{i,j}$ are Toeplitz matrices and $\mathbf{A}_{i,j} = \mathbf{A}_{i-j}$. A linear system $\mathbf{Ax} = \mathbf{b}$ is said to be a *Toeplitz system* if $\mathbf{A}$ is a one-dimensional or two-dimensional Toeplitz matrix.

Now, let us consider a special type of Toeplitz matrix. An $n \times n$ matrix $\mathbf{B}_n$ is said to be a *circulant matrix* if

$$\mathbf{B}_n = \begin{pmatrix} b_0 & b_{-1} & \cdots & & \cdots\cdots & b_{-(n-1)} \\ b_1 & b_0 & b_{-1} & \ddots & \ddots & \vdots \\ \vdots & b_1 & b_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & b_1 & b_0 & b_{-1} \\ b_{n-1} & \cdots & \cdots & \cdots & b_1 & b_0 \end{pmatrix},$$

where $b_i = b_{i-n}$. As in the two-dimensional case for Toeplitz matrices, an $n^2 \times n^2$ matrix is said to be a *block-circulant-circulant-block (BCCB) matrix* if it has the structure as in (2), where each block is an $n \times n$ circulant matrix and it is also circulant in the block sense.

Circulant matrices $\mathbf{B}_n$ can be diagonalized by the Fourier matrix $\mathbf{F}_n = [f_{jk}]$ where

$$f_{jk} = \frac{1}{\sqrt{n}} e^{-\frac{2\pi ijk}{n}}, \quad 0 \le j,\, k < n.$$

That is,

$$\mathbf{B}_n = \mathbf{F}_n^* \Lambda_n \mathbf{F}_n, \tag{3}$$

where Λ_n is a diagonal matrix, see Davis [10]. The two-dimensional case of circulant matrices can be factorized similarly. That is, an $n^2 \times n^2$ BCCB matrices can be

diagonalized by the two-dimensional Fourier matrices $\mathbf{F}_n \otimes \mathbf{F}_n$, where $\otimes$ denotes the tensor product.

We note that the diagonal entries of Λ_n can be obtained in $O(n \log n)$ operations by taking the fast Fourier transform on the first column of $\mathbf{B}_n$. In fact, the diagonal entries λ_k of Λ_n are given by

$$\lambda_k = \sum_{j=0}^{n-1} b_j e^{2\pi ijk/n}, \quad k = 0, \ldots, n-1.$$

Once Λ_n is obtained, the product $\mathbf{B}_n\mathbf{y}$ and $\mathbf{B}_n^{-1}\mathbf{y}$ for any vector $\mathbf{y}$ can be computed by FFTs in $O(n \log n)$ operations using (3).

Actually, multiplication of an $n \times n$ Toeplitz matrix and an n-vector can also be done in the order of $O(n \log n)$ operations, see Strang [17], by embedding the Toeplitz matrix in a circulant matrix with twice the size as shown below:

$$\begin{pmatrix} \mathbf{A}_n & \times \\ \times & \mathbf{A}_n \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_n\mathbf{x} \\ * \end{pmatrix}$$

Here $\mathbf{0}$ is the zero vector of length n. The required product is then obtained. Thus, the computational cost of the multiplication of an $n \times n$ Toeplitz matrix and an n-vector is also of the order $O(n \log n)$ operations.

For the two-dimensional case, an $n^2 \times n^2$ BCCB matrix can be multiplied by a vector in the order of $O(n^2 \log n)$ operations only. By the similar idea, an $n^2 \times n^2$ BTTB matrix can be enlarged to a $4n^2 \times 4n^2$ BCCB matrix, so it can also be multiplied by a vector in the order of $O(n^2 \log n)$ operations. By using such a nice result, a Toeplitz system can be solved fast by using an iterative method, such as the conjugate gradient method. This will be discussed in the next section.

Toeplitz systems can be found in a variety of practical applications, like signal and image processing, time series analysis, integral equations and queueing problems. In signal and image deblurring, if the blurring function is spatially invariant, then the blurring matrix corresponding to this blurring function is Toeplitz in the one-dimensional case and BTTB in the two-dimensional case. To deblur the image, we

need to solve a Toeplitz system. It is well-known that an $n \times n$ linear system can be solved by the Gaussian elimination method in $O(n^3)$ operations. Recent research on solving Toeplitz systems have reduced the complexity to only $O(n \log n)$ for a large class of Toeplitz matrices, see [4].

2 Conjugate Gradient Method

The conjugate gradient method is an iterative method for solving Hermitian positive definite matrix system. The algorithm of the method can be found in Golub and Van Loan [11, p.523]. In each iteration, it requires two inner products of n-vectors and one multiplication of the coefficient matrix with an n-vector. For Toeplitz systems, since we can perform the matrix-vector multiplication in $O(n \log n)$ iterations, the cost per iteration is reduced from $O(n^2)$ to $O(n \log n)$.

The convergence rate of the conjugate gradient method is well studied, see Axelsson and Barker [2]. It depends on the condition number of the matrix $\mathbf{A}_n$ and how clustered the spectrum of $\mathbf{A}_n$ is. It is shown in [4, P.7] that the convergence rate for Toeplitz system is usually linear. Moreover, we can see in [4, P.7] that the convergence rate depends on the ratio of the largest eigenvalue to the smallest eigenvalue. If the ratio is large, then the convergence will be very slow. To speed up the convergence rate, we can precondition the Toeplitz matrices. Thus, instead of solving $\mathbf{A}_n\mathbf{x} = \mathbf{b}$, we solve the preconditioned system

$$\mathbf{P}_n^{-1}\mathbf{A}_n\mathbf{x} = \mathbf{P}_n^{-1}\mathbf{b}.$$

The matrix $\mathbf{P}_n$ is called the preconditioner.

In 1986, Strang [17] and Olkin [15] independently proposed the use of circulant matrices to precondition Toeplitz matrices in conjugate gradient iterations. With circulant matrices as preconditioners, in each iteration, we have to solve a circulant system. From (3), we see that circulant matrices can be diagonalized by the Fourier matrix, and hence the inversion of an $n \times n$ circulant system can be done in $O(n \log n)$ operations by using FFTs of size n. Therefore, the cost per iteration using a circulant matrix as precondi-

tioner is still $O(n \log n)$. Theoretical and numerical results [7, 18, 15] suggested that the method converges very fast for a wide range of Toeplitz matrices.

There are many circulant preconditioners proposed. One of the most popular circulant preconditioners is the one proposed by T. Chan in [8]. For an $n \times n$ Toeplitz matrix $\mathbf{A}_n$, T. Chan's circulant preconditioner $\mathrm{cir}(\mathbf{A}_n)$ is defined to be the minimizer of

$$\|\mathbf{B}_n - \mathbf{A}_n\|_F \tag{4}$$

over all $n \times n$ circulant matrices $\mathbf{B}_n$. Here $\|\cdot\|_F$ denotes the Frobenius norm. More explicitly, the j-th diagonals of $\mathrm{cir}(\mathbf{A}_n)$ for an $n \times n$ Toeplitz matrix $\mathbf{A}_n$ are equal to

$$b_j = \begin{cases} \dfrac{(n-j)a_j + ja_{j-n}}{n}, & 0 \le j < n, \\ b_{n+j}, & 0 < -j < n, \end{cases}$$

which are just the average of the diagonals of $\mathbf{A}_n$ with the diagonals being extended to length n by a wrap around. Note that when $\mathbf{A}_n$ is not a Toeplitz matrix, T. Chan's circulant preconditioner can still be obtained by taking the arithmetic average of the entries of $\mathbf{A}_n$. i.e. its diagonals are given by

$$b_l = \frac{1}{n} \sum_{j-k=l(\mathrm{mod}\ n)} a_{jk}, \quad l = 0, \ldots, n-1,$$

see [22]. We observe that a matrix is circulant if and only if it can be diagonalized by the Fourier matrix. From this observation, we can construct preconditioners that can be diagonalized by the cosine transform matrix and minimize the Frobenius norm in (4). Such a preconditioner is called the cosine transform preconditioner.

3 Cosine Transform Preconditioner

3.1 Construction of One-dimensional Preconditioner

The (j,k)-th entry of the $n \times n$ discrete cosine transform matrix $\mathbf{C}_n$ is given by

$$\sqrt{\frac{2-\delta_{i1}}{n}} \cos\left(\frac{(i-1)(2j-1)\pi}{2n}\right), \quad 1 \le i, j \le n, \tag{5}$$

where δ_{ij} is the Kronecker delta. We note that for any n-vector $\mathbf{y}$, the matrix-vector multiplication $\mathbf{C}_n\mathbf{y}$ can be computed in $O(n \log n)$ real operations, see Sorensen and Burrus [16].

Let $\mathfrak{C}_{n\times n}$ be the vector space containing all $n \times n$ matrices that can be diagonalized by $\mathbf{C}_n$. For an $n \times n$ matrix $\mathbf{A}_n$, the cosine transform preconditioner $\cos(\mathbf{A}_n)$ of $\mathbf{A}_n$ is defined to be the matrix $\mathbf{C}_n^T \Lambda \mathbf{C}_n$ that minimizes

$$\|\mathbf{C}_n^T \Lambda \mathbf{C}_n - \mathbf{A}_n\|_F,$$

where Λ is a diagonal matrix. Recall that T. Chan's circulant preconditioner is the minimizer of (4) over all matrices that can be diagonalized by the Fourier matrix. The cosine transform preconditioner is defined similarly, but the preconditioner is diagonalized by the cosine transform matrix.

We now give the explicit formula for $\cos(\mathbf{A}_n)$. In the first place, we have the following result:

Theorem 1 (Boman and Koltracht [3]) *Let Q_i, $i = 1, \ldots, n$, be $n \times n$ matrices with the (h,k)-th entry given by*

$$Q_i(h,k) = \begin{cases} 1 & \text{if } |h-k| = i-1, \\ 1 & \text{if } h+k = 2n-i+2, \\ 1 & \text{if } h+k = i, \\ 0 & \text{otherwise.} \end{cases}$$

Then $\{Q_i\}_{i=1}^n$ is a basis for $\mathfrak{C}_{n\times n}$.

For any matrix $\mathbf{A}_n = (a_{ij})$, let r_n be an n-vector with the k-th component given by

$$(r_n)_k = \sum_{(Q_k)_{i,j}\neq 0} a_{ij},$$

which is the sum of those a_{ij} for which the corresponding entries of Q_k are nonzero. The first column of the cosine transform preconditioner is given by the following theorem:

Theorem 2 (Chan, Chan and Wong [5]) *Let* $\mathbf{A}_n$ *be an* $n \times n$ *matrix and* $\cos(\mathbf{A}_n)$ *be the minimizer of* (4) *over all* $\mathbf{B}_n \in \mathfrak{C}_{n\times n}$. *Denote by* $q = (q_1, \ldots, q_n)^T$ *the first column of* $\cos(\mathbf{A}_n)$. *If* s_o *and* s_e *are defined respectively to be the sum of the odd and even index entries of* r_n, *then we have, for* n *even,*

$$\begin{aligned} q_1 &= \frac{1}{2n^2}(2n(r_n)_1 + n(r_n)_2 - 2s_e), \\ q_i &= \frac{1}{2n^2}(n(r_n)_i + n(r_n)_{i+1} - 2s_e), \quad i = 2, \ldots, n-1, \\ q_n &= \frac{1}{2n^2}(-2ns_o + (2n-2)s_e + n(r_n)_n), \end{aligned}$$

and for n *odd,*

$$\begin{aligned} q_1 &= \frac{1}{2n^2}(2n(r_n)_1 + n(r_n)_2 - 2s_o), \\ q_i &= \frac{1}{2n^2}(n(r_n)_i + n(r_n)_{i+1} - 2s_o), \quad i = 2, \ldots, n-1, \\ q_n &= \frac{1}{2n^2}(-2ns_e + (2n-2)s_o + n(r_n)_n). \end{aligned}$$

Once we get the first column of $\cos(\mathbf{A}_n)$, its eigenvalues can be obtained by taking a fast cosine transform (FCT) of the first column of $\cos(\mathbf{A}_n)$. In particular, any matrix in $\mathfrak{C}_{n\times n}$ is uniquely determined by its first column. We remark that the cost of constructing $\cos(\mathbf{A}_n)$ is $O(n^2)$ if $\mathbf{A}_n$ has no special structure. However, if $\mathbf{A}_n$ is Toeplitz, then the cost of constructing $\cos(\mathbf{A}_n)$ is just $O(n)$. Since most of the problems in image processing are two-dimensional problems, we should extend the construction to two-dimensional cases.

3.2 Construction of Two-dimensional Preconditioner

We now construct cosine transform preconditioners for the block matrix $\mathbf{A}_{nn}$ with the same structure as (2). In [9], T. Chan and Olkin proposed the two-dimensional circulant preconditioners for such matrices. R. Chan, T. Chan and Wong [5] use the same approach to define the two-dimensional cosine transform preconditioner for $\mathbf{A}_{nn}$. We describe the method of construction here. First, we take the cosine transform

approximation to each block of $\mathbf{A}_{nn}$,

$$\cos_1(\mathbf{A}_{nn}) = \begin{pmatrix} \cos(\mathbf{A}_{1,1}) & \cos(\mathbf{A}_{1,2}) & \dots & \cos(\mathbf{A}_{1,n}) \\ \cos(\mathbf{A}_{2,1}) & \cos(\mathbf{A}_{2,2}) & \dots & \cos(\mathbf{A}_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\mathbf{A}_{n,1}) & \cos(\mathbf{A}_{n,2}) & \dots & \cos(\mathbf{A}_{n,n}) \end{pmatrix}.$$

Denote $(\mathbf{A}_{nn})_{i,j;k,l}$ to be the (i,j)-th entry of the (k,l)-th block of $\mathbf{A}_{nn}$. Let P be a permutation matrix that reorders $\mathbf{A}_{nn}$ in another coordinate direction. i.e. P satisfies

$$(P^T \mathbf{A}_{nn} P)_{i,j;k,l} = (\mathbf{A}_{nn})_{k,l;i,j}, \quad 1 \le i,j \le n, 1 \le k,l \le n.$$

Then the cosine transform preconditioner $\cos_2(\mathbf{A}_{nn})$ for $\mathbf{A}_{nn}$ is defined by

$$\cos_2(\mathbf{A}_{nn}) = P \cos_1(P^T \cos_1(\mathbf{A}_{nn}) P) P^T.$$

It can be shown easily that $\cos_2(\mathbf{A}_{nn})$ can be diagonalized by $\mathbf{C}_n \otimes \mathbf{C}_n$. Hence, $\cos_2(\mathbf{A}_{nn})$ can be inverted in $O(n^2 \log n)$ operations.

4 Regularization

4.1 Tikhonov regularization

In many image processing problems, we need to solve an ill-conditioned system. This means that small changes in the data can cause large changes in the solution. It is necessary to incorporate further information about the desired solution in order to stabilize the problem and to filter out the influence of the noise. This is the purpose of regularization.

Let $\mathcal{X}$ and $\mathcal{Y}$ be Hilbert spaces. Suppose $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is a linear and bounded operator and y^δ is the noisy data. If we would like to solve the system $\mathcal{A}x = y$ under the information that $\|y^\delta - y\| \le \delta$, we have to accept any $x \in \mathcal{X}$ with

$$\|\mathcal{A}x - y^\delta\| \le \delta \tag{6}$$

as an approximate solution. Since the system is ill-conditioned, the set of x satisfying (6) is unbounded. Therefore, we look for the solution of $\mathcal{A}x = y$ with minimal norm. It makes sense to use this requirement also as a selection criterion. Instead, we consider the problem

$$\min \|x\| \quad \text{subject to} \quad \|\mathcal{A}x - y^\delta\| \leq \delta. \tag{7}$$

Suppose $\|y^\delta\| \geq \delta$, then the minimum is attained on the boundary of the feasible set, and thus (7) is equivalent to

$$\min \|x\|^2 \quad \text{subject to} \quad \|\mathcal{A}x - y^\delta\|^2 = \delta^2. \tag{8}$$

Using a Lagrange multiplier, this in turn is equivalent to

$$\begin{aligned} & \min_x \{\|x\|^2 + \lambda \|\mathcal{A}x - y^\delta\|^2\} \\ \text{or} \quad & \min_x \{\alpha \|x\|^2 + \|\mathcal{A}x - y^\delta\|^2\}. \end{aligned} \tag{9}$$

The functional $\|\mathcal{A}x - y^\delta\|^2 + \alpha\|x\|^2$ in (9) is called the *Tikhonov functional.* Thus, instead of solving $\mathcal{A}x = y^\delta$, we solve for (9). The method is called *Tikhonov regularization* [19, 20]. The following theorem gives a precise method in solving for the minimizer in (9):

Theorem 3 ([10],p.117 Theorem 5.1) *Let x_α^δ be the solution of the following system:*

$$(\mathcal{A}^*\mathcal{A} + \alpha\mathcal{I})x = \mathcal{A}^* y^\delta. \tag{10}$$

Then x_α^δ is the unique minimizer of the Tikhonov functional $\|\mathcal{A}x - y^\delta\|^2 + \alpha\|x\|^2$.

It is also possible to minimize the seminorm $\|\mathcal{L}x\|$ instead of $\|x\|$ in (7). Here $\mathcal{L}$ is usually chosen to be the k-th order differential operator. In this case, the minimizer is the solution of the equation

$$(\mathcal{A}^*\mathcal{A} + \alpha\mathcal{L}^*\mathcal{L})x = \mathcal{A}^* y^\delta.$$

We remark that when $\mathcal{L}$ equals the identity operator, it returns to the classical Tikhonov regularization described above. Let $\mathbf{A}$, $\mathbf{L}$, $\mathbf{x}$ and $\mathbf{y}^\delta$ be respectively the discretization

of $\mathcal{A}$, $\mathcal{L}$, x and y^δ. The discretized problem becomes

$$(\mathbf{A}^*\mathbf{A} + \alpha\mathbf{L}^*\mathbf{L})\mathbf{x} = \mathbf{A}^*\mathbf{y}^\delta.$$

In this thesis, we choose $\mathcal{L}$ to be the identity operator or the first order differential operator. Correspondingly, the discretization matrix $\mathbf{L}^*\mathbf{L}$ will be the identity matrix or the two-dimensional Laplacian matrix.

The *regularization parameter* α controls how much weight is given to the minimization of $\|\mathbf{Lx}\|$ relative to the minimization of the residual norm. Clearly, a large α (equivalent to a large amount of regularization) favors a solution of small seminorm at the cost of a large residual norm, while a small α (i.e. a small amount of regularization) has the opposite effect. Therefore, it is important to find a regularization parameter that gives a good balance, filtering out enough noise without losing too much information in the computed solution.

4.2 The L-curve

L-curve is the most convenient graphical tool for analysis of discrete ill-conditioned problems. It was used by Lawson and Hanson [14] and further studied by Hansen [13]. Given a system $\mathbf{Ax} = \mathbf{b} + \eta$, where η is the noise vector. Let $\tilde{\mathbf{b}} = \mathbf{b} + \eta$ and $\mathbf{x}_{reg}$ be the regularized solution. The L-curve is a plot of the (semi)norm $\|\mathbf{Lx}_{reg}\|$ versus the corresponding residual norm $\|\mathbf{Ax}_{reg} - \tilde{\mathbf{b}}\|$. In this way, the L-curve displays the compromise between minimization of these two quantities.

For discrete ill-conditioned problems it turns out that the L-curve, when plotted in log-log scale, almost always has a characteristic L-shaped appearance with a distinct corner separating the vertical and horizontal parts of the curve. We now explain how this happens. Let $\mathbf{x}_{true}$ denote the exact solution corresponding to the original problem $\mathbf{Ax} = \mathbf{b}$, then the error $\mathbf{x}_{reg} - \mathbf{x}_{true}$ in the regularized solution consists of two components, namely, a perturbation error from the noise η in the given right-hand side $\tilde{\mathbf{b}}$, and a regularization error due to the regularization of the noise-free component $\mathbf{b}$ in the right-hand-side. The vertical part of the L-curve corresponds to solutions

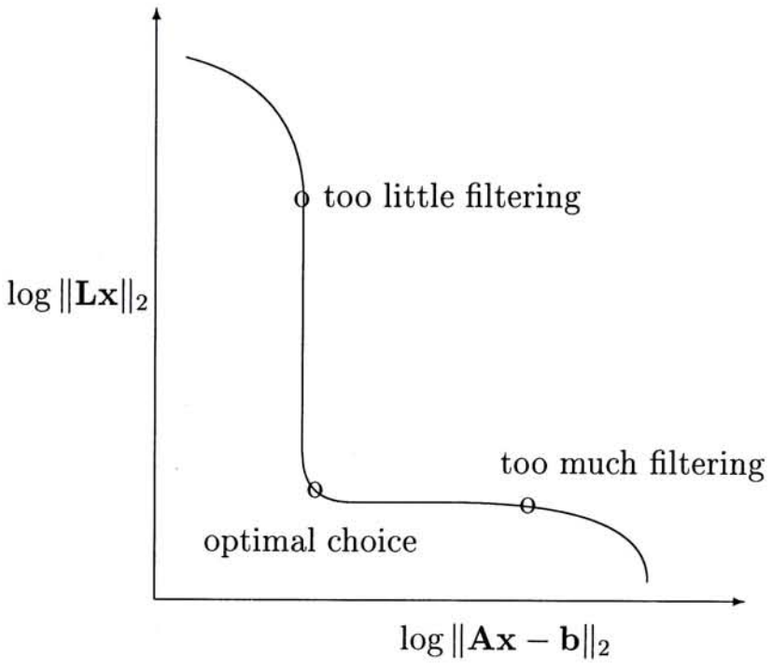

Figure 1: A typical L-curve

where $\|\mathbf{L}\mathbf{x}_{reg}\|_2$ is very sensitive to changes in regularization parameter because the noise vector η dominates $\mathbf{x}_{reg}$. The horizontal part of the L-curve corresponds to solutions where their residual norm $\|\mathbf{A}\mathbf{x}_{reg} - \tilde{\mathbf{b}}\|_2$ is most sensitive to the regularization parameter, because $\mathbf{x}_{reg}$ is dominated by the regularization error. In other words, by locating the corner of the L-curve, one can compute a regularized solution with a good balance between the perturbation error and the regularization error, and so the corresponding regularization parameter is a good one. Figure 1 shows the generic form of L-curves.

5 Summary

5.1 Paper A

In many electronic imaging applications, a high-resolution image is desired from available multiple undersampled image frames. The observed undersampled images are

often degraded by blur and noise. Thus reconstruction of a high-resolution image is of great interest and has been studied extensively, see for instance [21, 4].

Multiple undersampled images are obtained by using multiple identical image sensors shifted from each other by subpixel displacements. To obtain a high-resolution image from these undersampled frames, the simplest approach is to combine the low-resolution images and rearrange their entries alternately. However, the high-resolution image obtained in this way will be blurred and we need to deblur the image by solving a linear system.

There may be subpixel displacements errors between sensors. If the image sensors are shifted from each other by exact subpixel values, then the subpixel displacement errors equal to zero in all the sensors. In this case, the blur function will be spatially invariant. However, as perfect subpixel displacements are practically impossible, the blur function obtained should be spatially variant.

Since we do not have any information about the scene outside the frames, a common approach is to impose zero boundary condition outside the scene. However, when this assumption is not satisfied by the images, ringing effect will occur at the boundary of the reconstructed image, see for instance Bose and Boo [4]. One of the main results in this paper is that we assume Neumann boundary conditions in our problem. Imposing Neumann boundary condition means that we assume the scene immediately outside is a reflection of the original scene at the boundary. In most of the cases, this boundary assumption is more reasonable and the error of the reconstructed image under this boundary condition is less than that reconstructed by zero boundary condition, see the numerical results in Paper A.

The blurring operator in our problem is ill-conditioned, thus regularization technique should be applied to regularize the solution. In this paper, classical Tikhonov regularization [19] will be applied. The discretized problem becomes:

$$(\tilde{\mathbf{H}}^*\tilde{\mathbf{H}} + \alpha\mathbf{R})\mathbf{f} = \tilde{\mathbf{H}}^*\mathbf{g} \tag{11}$$

where $\tilde{\mathbf{H}}$ is the discretization matrix of the blurring operator under Neumann boundary condition, and $\mathbf{R}$ is the discretization matrix of the regularization functional. We

choose $\mathbf{R}$ to be the identity matrix and Laplacian matrix in this paper.

We prove that when the image is reconstructed from a 2×2 sensor array, the blurring matrix $\tilde{\mathbf{H}}$ can be diagonalized by a two-dimensional cosine transform. We thus propose to solve (19) by using PCG with the cosine transform preconditioner discussed in §3. Numerical results will be given to illustrate the sound recovery of the high-resolution images and the fast convergence of the cosine transform preconditioned system.

5.2 Paper B

Blur removal is an important problem in signal and image processing. Given a portion of the blurred signal or image g and the blurring function h, our goal is to recover the corresponding portion of the original signal or image f. Since g is the convolution of h and f, the given portion of g is not completely determined by the same portion of f. Indeed, an entry of g near the boundary is also affected by the values of f that lie outside and close to the interested portion. Thus the problem of recovering f from g is underdetermined and we need to make assumptions on the values of f outside the interested portion. These assumptions are called boundary conditions.

The most classical approach is to assume the zero boundary condition [2, p.211-220], which assumes that all the values of f outside the interested portion are equal to zero. The blurring matrix becomes a Toeplitz matrix in one-dimensional cases and a BTTB matrix in two-dimensional cases. However, the cost of solving such systems is expensive, especially in two-dimensional cases. To lower the computational costs, we may assume periodic boundary condition [2, p.126]. This means that the data outside the interested portion are exact copies of data inside. Under this boundary condition, the blurring matrix becomes a circulant matrix in one-dimensional cases and a BCCB matrix in two-dimensional cases. It is well known that these matrices can be diagonalized by the Fourier matrix, see §1. Thus we can find their inverses easily by using fast Fourier transforms (FFTs). However, practical signals and images usually do not satisfy these two assumptions and ringing effects will appear on the boundary of the recovered signals or images.

The main result in this paper is that we propose the Neumann boundary condition. It assumes that the data outside f are reflections of data inside f. The resulting blurring matrix becomes a Toeplitz-plus-Hankel matrix in one-dimensional cases and a block Toeplitz-plus-Hankel matrix with Toeplitz-plus-Hankel blocks in two-dimensional cases. We show that such matrices can be diagonalized by the discrete cosine transform matrix when the blurring function is symmetric. Therefore, their inverses can be found by using three FCTs. Since the FCT requires only real operations and is about twice as fast as FFT, solving a problem with Neumann boundary conditions is twice as fast as solving a problem with periodic boundary conditions. Also, numerical results show that such boundary assumptions produce less ringing effects than that of the zero or periodic boundary assumptions. We believe that this work will have great impact in the field of digital image processing.

References

[1] H. Andrew and B. Hunt, *Digital Image Restoration*, Prentice-Hall, New Jersey, 1977.

[2] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic Press, Orlando, Fl., 1983.

[3] E. Boman and I. Koltracht, *Fast Transform Based Preconditioners for Toeplitz Equations*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 628–645.

[4] N. K. Bose and K. J. Boo. *High-resolution image reconstruction with multisensors*, to appear in International Journal of Imaging Systems and Technology.

[5] R. Chan, T. Chan and C. K. Wong. *Cosine transform based preconditioner for total variation minimization problems in image processing*, Iterative Methods in Linear Algebra, II(3), IMACS Series in Computational and Applied Mathematics, Proceedings of the Second IMACS International Symposium on Iterative Methods in Linear Algebra, Bulgaria, pp. 311–329, 1995.

[6] R. Chan and M. Ng, *Conjugate Gradient Method for Toeplitz Systems*, SIAM Rev., 38 (1996), pp. 427–482.

[7] R. Chan and G. Strang, *Toeplitz Equations by Conjugate Gradients with Circulant Preconditioner*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 104–119.

[8] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 766–771.

[9] T. Chan and J. Olkin. *Circulant preconditioners for Toeplitz-block matrices*, Numer. Algo., 6 (1994), pp. 89–101.

[10] P. Davis, *Circulant Matrices*, John Wiley & Sons, Inc., New York, 1979.

[11] H. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, The Netherlands, 1996.

[12] G. Golub and C. Van Loan, *Matrix Computations*, 2nd Ed., The Johns Hopkins University Press, Maryland, 1989.

[13] P. Hansen, *Analysis of Discrete Ill-Posed Problems By Means of L-Curve*, SIAM Rev., 34 (1992), pp. 561–580.

[14] C. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewoods Cliffs, 1974.

[15] J. Olkin, *Linear and Nonlinear Deconvolution Problems*, Ph.D. thesis, Rice Univ., Houston, TX, 1986.

[16] H. Sorensen and C. Burrus, *Fast DFT and Convolution Algorithms*, Handbook of Signal Processing edited by S. Mitra and J. Kaiser, New York, Wiley.

[17] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[18] G. Strang and A. Edelman, *The Toeplitz-Circulant Eigenvalue Problem* $Ax = \lambda Cx$, in Oakland Conf. on PDE and Appl. Math., L. Bragg and J. Dettman, eds., Longman Sci. Tech., New York, 1987.

[19] A. N. Tikhonov, *Solution of Incorrectly Formulated Problems and the Regularization Method*, Soviet Math. Dokl., 4 (1963), pp. 1035–1038

[20] A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-Posed Problems*, Wiley, New York, 1977.

[21] R. Y. Tsai and T. S. Huang. *Multiframe image restoration and registration*, Advances in Computer Vision and Image Processing, 1 (1984), pp. 317–339.

[22] E. Tyrtyshnikov, *Optimal and Superoptimal Circulant Preconditioners*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 459–473

Preconditioned Iterative Methods for High-resolution Image Reconstruction with Multisensors

Abstract

We study the problem of reconstructing a high-resolution image from multiple undersampled, shifted, degraded frames with subpixel displacement errors. The corresponding reconstruction operator $\mathcal{H}$ is a spatially variant operator. In this paper, instead of using the usual zero boundary condition (corresponding to a dark background outside the scene), the Neumann boundary condition (corresponding to a reflection of the original scene at the boundary) is imposed on the images. The resulting discretization matrix of $\mathcal{H}$ is a block-Toeplitz-Toeplitz-block-like matrix. We apply the preconditioned conjugate gradient (PCG) method with cosine transform preconditioners to solve the discrete problems. Preliminary results show that the image model under the Neumann boundary condition gives better reconstructed high-resolution images than that under the zero boundary condition, and the PCG method converges very fast.

1 Introduction

High-resolution image reconstruction has many electronic imaging applications, including aerial or facilities surveillance, consumer, commercial, medical, forensic, and scientific imaging. The observed images often have low resolution and are degraded by blur and noise. Increasing the image resolution by using digital signal processing technique [2, 6, 7, 8, 10, 12] is therefore of great interest.

We consider the reconstruction of a high resolution image f from multiple undersampled, shifted, degraded and noisy images. Multiple undersampled images are often obtained by using multiple identical image sensors shifted from each other by subpixel displacements. If the image sensor arrays are shifted from each other by an exact subpixel displacement in the ideal case, then the task of reconstructing high resolution images reduces to solving a spatially invariant linear system, $\mathcal{H}_0 f = g$. Here g, the so-called *observed high-resolution image*, is a combination of all the low-resolution frames. However, exact subpixel displacements are not practical, and we usually obtain a spatially variant system $\mathcal{H} f = g$ instead.

Since the system is ill-conditioned and generally not positive definite, we solve it by using a minimization and regularization technique:

$$\min_f \left\{ \|\mathcal{H} f - g\|_2^2 + \alpha \mathcal{R}(f) \right\}. \tag{1}$$

Here $\mathcal{R}(f)$ is a functional which measures the regularity of f and the regularization parameter α is used to control the degree of regularity of the solution. Previous works (for instance Bose and Boo [2]) did not emphasize the boundary condition of the problem (1). Since we do not have any information about the scene outside the frames, a natural approach is to impose zero boundary condition outside the scene, i.e., assuming a dark background outside the scene [2]. However, when this assumption is not satisfied by the images, ringing effect will occur at the boundary of the reconstructed image (see the numerical results in Bose and Boo [2]). The problem is more severe if the image is reconstructed from a large sensor array since the number of pixel values of the image affected by the sensor array increases. We propose here using the Neumann boundary condition on the image, which assumes that the scene immediately outside is a reflection of the original scene at the boundary. Our numerical results show that the error of the image under the Neumann boundary condition is less than that under the zero boundary condition.

The discretization matrix of $\mathcal{H}$ is a block-Toeplitz-Toeplitz-block-like matrix. The preconditioned conjugate gradient (PCG) method is commonly used in solving the system, see Chan and Ng [4]. We observe that for a 2×2 sensor array with exact

subpixel displacement, the matrix can be diagonalized by the discrete cosine transform matrix. We thus propose using the PCG method with cosine transform preconditioners for solving the system. Numerical results show that our preconditioners perform significantly better than other preconditioners.

The main results of this paper are to propose (i) a novel approach of using Neumann boundary conditions for image reconstruction and (ii) PCG methods with cosine transform based preconditioners to solve large linear systems arising from image reconstruction. In Section 2, we give a mathematical formulation of the problem. The Neumann boundary condition and a brief introduction on cosine transform preconditioners will be given there. Finally, numerical results are given in Section 3.

2 High-resolution Image Reconstruction

2.1 Mathematical Model

Suppose we have an $L_1 \times L_2$ sensor array, each sensor has $N_1 \times N_2$ sensing elements, and the size of each sensing element is $T_1 \times T_2$. Our aim is to reconstruct an image of resolution $M_1 \times M_2$, where $M_1 = L_1 \times N_1$ and $M_2 = L_2 \times N_2$. The sampled base interval for the high-resolution image is therefore equal to $T_1/L_1 \times T_2/L_2$. To maintain the aspect ratio of the reconstructed image, we consider the case where $L_1 = L_2 = L$ only.

In order to have enough information to resolve the high resolution image, there are subpixel displacements between the sensors. In the ideal case, the sensors are shifted from each other by a value proportional to the sampled base interval $T_1/L \times T_2/L$. However, in practice there can be small perturbations around the ideal subpixel locations due to imperfection of the mechanical imaging system. Thus, for $l_1, l_2 = 0, 1, \cdots, L-1$ with $(l_1, l_2) \neq (0, 0)$, the horizontal and vertical displacements $d^x_{l_1 l_2}$ and $d^y_{l_1 l_2}$ are given by

$$d^x_{l_1 l_2} = \frac{T_1}{L}(l_1 + \epsilon^x_{l_1 l_2}) \quad \text{and} \quad d^y_{l_1 l_2} = \frac{T_2}{L}(l_2 + \epsilon^y_{l_1 l_2}),$$

where $\epsilon^x_{l_1 l_2}$ and $\epsilon^y_{l_1 l_2}$ denote respectively the normalized horizontal and vertical displacement errors. Here we assume that

$$|\epsilon^x_{l_1 l_2}| < \frac{1}{2} \quad \text{and} \quad |\epsilon^y_{l_1 l_2}| < \frac{1}{2}.$$

Since these can be set by users during camera calibration, the parameters $\epsilon^x_{l_1 l_2}$ and $\epsilon^y_{l_1 l_2}$ may be assumed to be known. We remark that the displacement errors cannot be greater than or equal to $1/2$ since the image sensor arrays are shifted from each other described by the rectangularly sampled base interval $T_1/L \times T_2/L$.

Let f be the original scene, the observed low-resolution image $g_{l_1 l_2}$ for the (l_1, l_2)-th sensor is modeled by:

$$g_{l_1 l_2}[n_1, n_2] = \int_{T_2(n_2-\frac{1}{2})+d^y_{l_1 l_2}}^{T_2(n_2+\frac{1}{2})+d^y_{l_1 l_2}} \int_{T_1(n_1-\frac{1}{2})+d^x_{l_1 l_2}}^{T_1(n_1+\frac{1}{2})+d^x_{l_1 l_2}} f(x_1, x_2) dx_1 dx_2 + \eta_{l_1 l_2}[n_1, n_2], \qquad (2)$$

for $n_1 = 1, \ldots, N_1$ and $n_2 = 1, \ldots, N_2$. Here $\eta_{l_1 l_2}$ is the noise corresponding to the (l_1, l_2)-th sensor. We intersperse the low-resolution images to form an $M_1 \times M_2$ image by assigning

$$g[L(n_1 - 1) + l_1, L(n_2 - 1) + l_2] = g_{l_1 l_2}[n_1, n_2].$$

Here g is an $M_1 \times M_2$ image and is called the *observed high-resolution image.* Figure 1 shows the method of forming a 4×4 image g with a 2×2 sensor array each having a 2×2 sensing elements ($L = 2$, $M_1 = M_2 = 4$, $N_1 = N_2 = 2$ and $T_1 = T_2 = 2$).

Using a lexicological ordering for g, we obtain

$$g = \mathcal{H} f + \eta$$

where $\mathcal{H}$ is a spatially variant operator [2] . Since $\mathcal{H}$ is ill-conditioned due to averaging of pixel values in the image model in (2), classical Tikhonov regularization is used and the minimization problem (1) is solved. In this paper, we use regularization functionals:

$$\mathcal{R}(f) = \|f\|_2^2 \quad \text{and} \quad \mathcal{R}(f) = \|\mathcal{D} f\|_2^2$$

where $\mathcal{D}$ is the first order differential operator. In these cases, the Euler-Lagrange equation of (1) becomes

$$(\mathcal{H}^*\mathcal{H} + \alpha \mathcal{I}) f = \mathcal{H}^* g \quad \text{and} \quad (\mathcal{H}^*\mathcal{H} + \alpha \mathcal{D}^*\mathcal{D}) f = \mathcal{H}^* g,$$

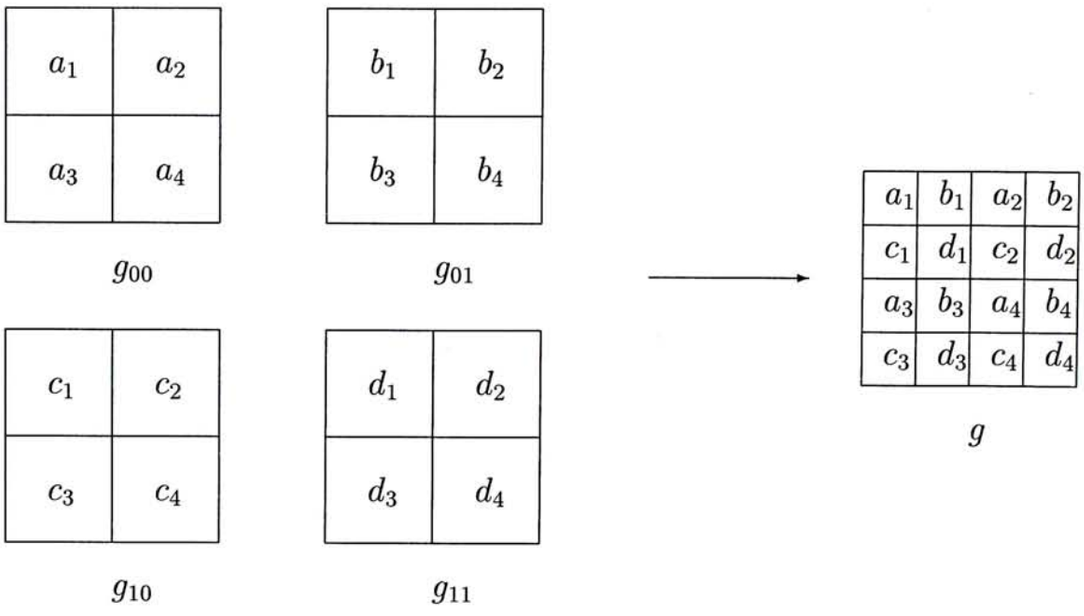

Figure 1: Construction of the observed high-resolution image

where $\mathcal{I}$ is the identity operator and $\mathcal{D}^*\mathcal{D}$ is the Laplacian operator.

2.2 Neumann Boundary Condition

The usual way of formulating the model with the zero boundary condition [2] will produce ringing effect at the boundary of the reconstructed image, see the numerical results in Bose and Boo [2] . We therefore propose to use the Neumann boundary condition, i.e., the scene immediately outside the frames are a reflection of the original frames at the boundary.

The continuous image model in (2) can be discretized by the rectangular rule and approximated by a discrete image model as follows. Let $\mathbf{g}$, $\mathbf{f}$ and $\mathbf{H}$ be respectively the discretization of g, f and $\mathcal{H}$ using a lexicological ordering. For simplicity, we discuss the case $L = 2$ here. Other cases can be derived similarly. For $L = 2$, under the zero boundary condition, the blurring matrix corresponding to the (l_1, l_2)-th sensor can be written as

$$\mathbf{H}_{l_1 l_2} = \mathbf{H}^x_{l_1 l_2} \otimes \mathbf{H}^y_{l_1 l_2}$$

where $\mathbf{H}^x_{l_1l_2}$ is the $M_1 \times M_1$ tridiagonal matrix

$$\mathbf{H}^x_{l_1l_2} = \frac{1}{2}\begin{pmatrix} 1 & h^{x+}_{l_1l_2} & & & \\ h^{x-}_{l_1l_2} & 1 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 1 & h^{x+}_{l_1l_2} \\ & & & h^{x-}_{l_1l_2} & 1 \end{pmatrix}.$$

Here $h^{x\pm}_{l_1l_2} = \frac{1}{2} \pm \epsilon^x_{l_1l_2}$. The $M_2 \times M_2$ blurring matrix $\mathbf{H}^y_{l_1l_2}$ is defined similarly. We remark that the condition numbers of $\mathbf{H}^x_{l_1l_2}$ and $\mathbf{H}^y_{l_1l_2}$ are of $O(M_1^2)$. For $L > 2$, the matrices $\mathbf{H}^x_{l_1l_2}$ and $\mathbf{H}^y_{l_1l_2}$ are band matrices with bandwidth $L + 1$.

Under the Neumann boundary condition, $\mathbf{H}^x_{l_1l_2}$ and $\mathbf{H}^y_{l_1l_2}$ are still tridiagonal matrices, but the entries on the upper left corner and the lower right corner are changed. The resulting matrix, denoted by $\tilde{\mathbf{H}}^x_{l_1l_2}$ and $\tilde{\mathbf{H}}^y_{l_1l_2}$ are given by

$$\tilde{\mathbf{H}}^x_{l_1l_2} = \frac{1}{2}\begin{pmatrix} 1 + h^{x-}_{l_1l_2} & h^{x+}_{l_1l_2} & & & \\ h^{x-}_{l_1l_2} & 1 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 1 & h^{x+}_{l_1l_2} \\ & & & h^{x-}_{l_1l_2} & 1 + h^{x+}_{l_1l_2} \end{pmatrix}.$$

The matrix $\tilde{\mathbf{H}}^y_{l_1l_2}$ can be similarly derived. The blurring matrix corresponding to the (l_1, l_2)-th sensor under the Neumann boundary condition is

$$\tilde{\mathbf{H}}_{l_1l_2} = \tilde{\mathbf{H}}^x_{l_1l_2} \otimes \tilde{\mathbf{H}}^y_{l_1l_2}.$$

Our discretization problem becomes:

$$(\tilde{\mathbf{H}}^*\tilde{\mathbf{H}} + \alpha\mathbf{R})\mathbf{f} = \tilde{\mathbf{H}}^*\mathbf{g} \tag{3}$$

where

$$\tilde{\mathbf{H}} = \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \mathbf{D}_{l_1l_2}\tilde{\mathbf{H}}_{l_1l_2}. \tag{4}$$

Here $\mathbf{D}_{l_1l_2}$ are diagonal matrices with diagonal elements equal to 1 if the corresponding component of $\mathbf{g}$ comes from the (l_1, l_2)-th sensor and zero otherwise. In (3), $\mathbf{R}$ is the discretization matrices corresponding to the regularization functional $\mathcal{R}(f)$.

2.3 Cosine Transform Preconditioners

Let $\mathbf{C}_n$ be the $n \times n$ discrete cosine transform matrix, i.e. the (i,j)-th entry of $\mathbf{C}_n$ is given by

$$\sqrt{\frac{2-\delta_{i1}}{n}} \cos\left(\frac{(i-1)(2j-1)\pi}{2n}\right), 1 \leq i,j \leq n,$$

where δ_{ij} is the Kronecker delta. Note that the matrix-vector product $\mathbf{C}_n x$ can be computed in $O(n \log n)$ operations, see Sorensen and Burrus [9, p.557]. For an $m \times m$ block matrix $\mathbf{B}$ with the size of each block equal to $n \times n$, the cosine transform preconditioner $c(\mathbf{B})$ of $\mathbf{B}$ is defined to be the matrix $(\mathbf{C}_m \otimes \mathbf{C}_n)^T \Lambda (\mathbf{C}_m \otimes \mathbf{C}_n)$ that minimizes

$$||(\mathbf{C}_m \otimes \mathbf{C}_n)^T \Lambda (\mathbf{C}_m \otimes \mathbf{C}_n) - \mathbf{B}||_F$$

in the Frobenius norm [3] . Clearly, the cost of computing $c(\mathbf{B})^{-1}y$ for any vector y is $O(mn \log mn)$ operations. For banded matrices, like the one we have in (4), the cost of constructing $c(\mathbf{B})$ is of $O(mn)$ only [3] .

When there is no subpixel displacement error, the matrix $\tilde{\mathbf{H}}_{l_1 l_2}$ are the same for all l_1 and l_2. Thus the blurring matrix $\tilde{\mathbf{H}}$ can be written as

$$\tilde{\mathbf{H}} = \tilde{\mathbf{H}}^x \otimes \tilde{\mathbf{H}}^y$$

where for $L = 2$, $\tilde{\mathbf{H}}^x$ is an $M_1 \times M_1$ tridiagonal matrix:

$$\frac{1}{4}\begin{pmatrix} 3 & 1 & & & \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ & & & 1 & 3 \end{pmatrix}$$

and $\tilde{\mathbf{H}}^y$ is an $M_2 \times M_2$ matrix with the same structure. It is easy to show that in this case, the matrices $\tilde{\mathbf{H}}^x$ and $\tilde{\mathbf{H}}^y$ can be diagonalized by $\mathbf{C}_{M_1}$ and $\mathbf{C}_{M_2}$ respectively. Thus $\tilde{\mathbf{H}}$ can be diagonalized by $\mathbf{C}_{M_1} \otimes \mathbf{C}_{M_2}$.

When there are subpixel displacement errors, the blurring matrix $\tilde{\mathbf{H}}$ is almost the same as that without errors, but with some entries slightly perturbed. We thus propose

to use the cosine transform preconditioner for the BTTB-like matrix $\tilde{\mathbf{H}}$. Our numerical results show that the cosine transform preconditioners can speed up the convergence much faster than other preconditioners. Since $\tilde{\mathbf{H}}$ is banded, the matrix-vector product $\tilde{\mathbf{H}}x$ can be done in $O((L_1+L_2)M_1M_2)$, thus the total cost per each iteration is $O((L_1+L_2)M_1M_2 + M_1M_2\log(M_1M_2))$ operations.

3 Numerical Results

In this section, we illustrate the effectiveness of the cosine transform based preconditioners by solving the high-resolution image reconstruction problem with a 2×2 sensor array and a 4×4 sensor array. In the tests, we use the zero vector as the initial guess in the preconditioned conjugate gradient method. The stopping criteria is $||\mathbf{r}^{(j)}||_2/||\mathbf{r}^{(0)}||_2 < 10^{-6}$, where $\mathbf{r}^{(j)}$ is the normal equations residual after j iterations. In the tests, the parameters $\epsilon_{l_1 l_2}^{xy}$ are set to be 0.1.

(i) 2×2 **sensor array**

Here we reconstruct a 128×128 image from four 64×64 images. The source image "Lena" is shown in Figure 9, Image A. It is a woman's face with background, and contains a high degree of contrast and detail. We first illustrate the need of regularization for this problem. In Figure 2, the left one is a low resolution image. The right one is the reconstructed image at 9 iterations, with $\|f\|_2^2$ as the regularization operator when the PCG method converges. However, when no regularization is used, the PCG method does not converge to a visually recognizable image. The middle one is the image solution we obtained at 9 iterations with no regularization. The SNR here is 40dB. We see that all the details of the original image are lost in the middle figure.

Next we test the effectiveness of using Neumann boundary conditions. Figure 3 shows the error of the first row of the image "Lena" recovered by using different boundary conditions. We used SNR=40dB and $R(f) = \|f\|_2^2$. We perform the experiment in the following way: we first generate the random noise, then reconstruct the image by imposing different boundary conditions, the pair of errors (e_z, e_n) is plotted on the

graph, where e_z is the error by imposing zero boundary condition and e_n is the error by imposing Neumann boundary condition. We repeat the same experiment 50 times. In each case the random noise we added into the observed low resolution image is different. The optimal regularization parameter is chosen such that it minimizes the relative error. Here the relative error of the reconstructed image $\mathbf{f}_c$ to the original image $\mathbf{f}$ is defined as:

$$\frac{\|\mathbf{f} - \mathbf{f}_c\|_2}{\|\mathbf{f}\|_2}.$$

From the graph, we can see that all the points are lying under the diagonal line, which means that the error by imposing the Neumann boundary condition is significantly less than that of the zero boundary condition in all of the cases. In Figure 4, we also show the reconstructed image under the zero boundary condition from the low resolution image in Figure 2 (left). We can compare Figures 2 (right) and 4. It is clear that the hair and the hat are reconstructed much better under the Neumann boundary condition than that under the zero boundary condition. We see that the boundary artifacts under the Neumann boundary condition are less prominent than that under the zero boundary condition.

In Figures 5 and 6, we show the observed and reconstructed images of the image "Lena" for SNR=40dB and 20dB respectively. Here the optimal regularization parameter α is chosen. We see from Figures 5c, 5d, 6c and 6d that the hair and the hat are restored much better than the observed high resolution image (Figures 5b, 6b). In order to compare the performance of different regularization methods, we show the relative errors of the reconstructed images. The relative errors for $\|f\|_2^2$ and $\|\mathcal{D}f\|_2^2$ are almost the same. Visually, their reconstructed images also look similar.

(ii) 4×4 **sensor array**

We perform the same test with the same scenes as in (i), but for a 4×4 sensor array. We use sixteen 32×32 images as the low-resolution images. The reconstructed image is of resolution 128×128. Figures 7–8 show the observed and reconstructed images with SNR=40dB and 20dB respectively. Again, the hair and the hat are restored much better in Figures 7c, 7d, 8c and 8d than those in Figures 7b and 8b. The optimal α

Image	$\|f\|_2^2$					$\|\mathcal{D}f\|_2^2$				
	α	cos	sin	cir	none	α	cos	sin	cir	none
A	2.7×10^{-3}	9	19	48	74	7.1×10^{-4}	9	18	47	70
B	1×10^{-2}	7	14	24	38	1.9×10^{-3}	7	14	27	42
C	2.4×10^{-2}	7	12	17	28	7.1×10^{-3}	6	11	16	26

Table 1: No. of iterations with optimal α and $L = 2$.

Image	$\|f\|_2^2$					$\|\mathcal{D}f\|_2^2$				
	α	cos	sin	cir	none	α	cos	sin	cir	none
A	1.1×10^{-2}	6	24	33	44	4.1×10^{-3}	6	27	35	48
B	8.1×10^{-3}	6	27	35	43	2.6×10^{-3}	6	32	41	49
C	5.6×10^{-2}	5	14	16	21	1.7×10^{-2}	5	16	18	26

Table 2: No. of iterations with optimal α and $L = 4$.

is chosen in the testing. Again, the functionals $\|f\|_2^2$ and $\|\mathcal{D}f\|_2^2$ perform more or less the same when an optimal α is chosen.

Finally, we test the convergence performance of the cosine transform based preconditioners. We will apply our method on 3 different 128×128 scenes, see Figure 9. Table 1 shows the performance of different preconditioners with $R(f) = \|f\|_2^2$ and $\|\mathcal{D}f\|_2^2$. The SNR here is 40dB. In the table, the α that minimizes the relative error is chosen. We test the best α up to 2 significant digit. In the tables, "cos", "sin", "cir", and "none" denote respectively the cosine transform preconditioner, the sine transform preconditioner [5], the T. Chan circulant preconditioner [11] and no preconditioner. We see from the tables that the cosine transform preconditioner converges significantly faster than the other preconditioners. Table 2 shows the performance of different preconditioners with different regularization functions. Optimal α is used in the table. The SNR is 40dB. Again, the cosine transform preconditioner is the best.

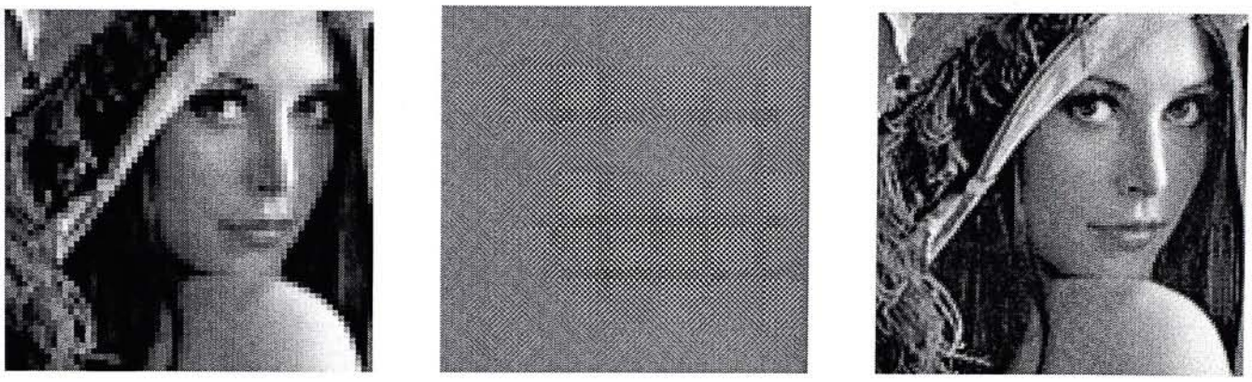

Figure 2: Low resolution 64 x 64 image (left), reconstructed 128 x 128 image (middle) without regularization (at 9 iter.) and (right) with regularization (at 9 iter under the Neumann boundary condition.)

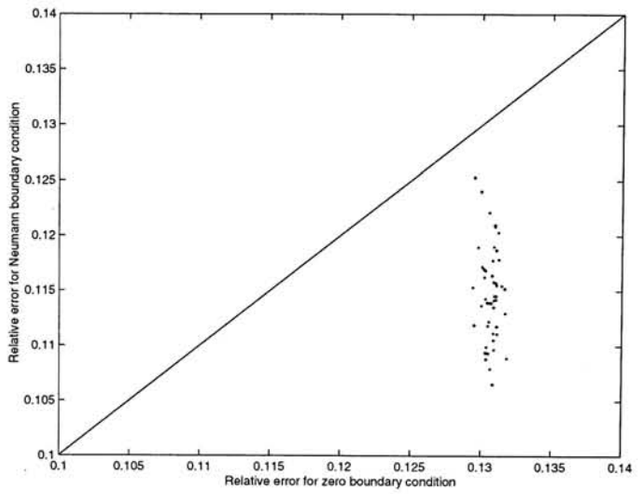

Figure 3: Errors by zero boundary conditions and Neumann boundary conditions.

Figure 4: Reconstructed 128 x 128 image under the zero boundary condition.

$L = 2$, **SNR = 40db**

Fig. 5a: Low resolution 64 x 64 image from the (0,0) sensor.

Fig. 5b: Observed high-resolution 128 x 128 image, rel. err. = 1.301×10^{-1}.

Fig. 5c: Reconstructed 128 x 128 image by αI regularization, rel. err. = 1.194×10^{-1}.

Fig. 5d: Reconstructed 128 x 128 image by $\alpha\Delta$ regularization, rel. err. = 1.156×10^{-1}.

$L = 2$, **SNR = 20db**

Fig. 6a: Low resolution 64 x 64 image from the (0,0) sensor.

Fig. 6b: Observed high-resolution 128 x 128 image, rel. err. = 1.502×10^{-1}.

Fig. 6c: Reconstructed 128 x 128 image by αI regularization, rel. err. = 1.476×10^{-1}.

Fig. 6d: Reconstructed 128 x 128 image by $\alpha\Delta$ regularization, rel. err. = 1.497×10^{-1}.

$L = 4$, **SNR = 40db**

Fig. 7a: Low resolution 32 x 32 image from the (0,0) sensor.

Fig. 7b: Observed high-resolution 128 x 128 image, rel. err. = 1.861×10^{-1}.

Fig. 7c: Reconstructed 128 x 128 image by αI regularization, rel. err. = 1.624×10^{-1}.

Fig. 7d: Reconstructed 128 x 128 image by $\alpha\Delta$ regularization, rel. err. = 1.696×10^{-1}.

$L = 4$, **SNR = 20db**

Fig. 8a: Low resolution 32 x 32 image from the (0,0) sensor.

Fig. 8b: Observed high-resolution 128 x 128 image, rel. err. = 1.932×10^{-1}.

Fig. 8c: Reconstructed 128 x 128 image by αI regularization, rel. err. = 1.765×10^{-1}.

Fig. 8d: Reconstructed 128 x 128 image by $\alpha\Delta$ regularization, rel. err. = 1.811×10^{-1}.

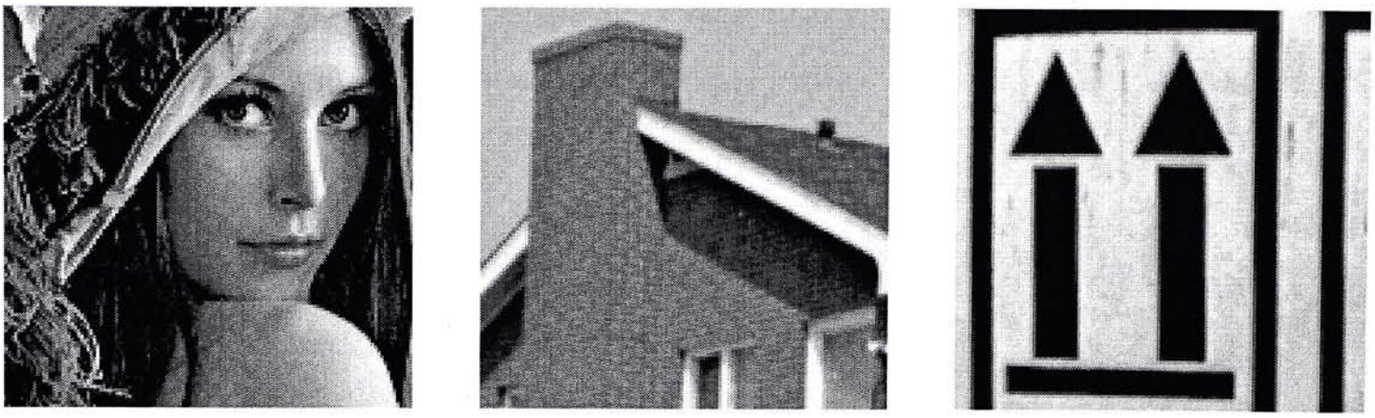

Figure 9: Original 128 × 128 images: Image A (left), Image B (middle) and Image C (right).

References

[1] H. Andrew and B. Hunt, *Digital Image Restoration*, Prentice-Hall, New Jersey, 1977.

[2] N. K. Bose and K. J. Boo., *High-resolution image reconstruction with multisensors*, to appear in International Journal of Imaging Systems and Technology.

[3] R. H. Chan, T. F. Chan and C. K. Wong., *Cosine transform based preconditioner for total variation minimization problems in image processing*, Iterative Methods in Linear Algebra, II(3), IMACS Series in Computational and Applied Mathematics, Proceedings of the Second IMACS International Symposium on Iterative Methods in Linear Algebra, Bulgaria, pp. 311–329, 1995.

[4] R. H. Chan and M. K. Ng., *Conjugate gradient method for Toeplitz system*, SIAM Review, 38 (1996) , pp. 427–482.

[5] R. H. Chan and C. K. Wong., *Sine Transform Preconditioner for Elliptic Problems*, Numer. Linear Algebra Applic., 4 (1997), pp. 351–368.

[6] E. Kaltenbacher and R. C. Hardie., *High resolution infrared image reconstruction using multiple, low resolution, aliased frames*, Proc. of IEEE 1996 National Aerospace and Electronic Conf. NAECON 2 (1996) , pp. 702–709.

[7] S. P. Kim, N. K. Bose and H. M. Valenzuela., *Recursive reconstruction of high resolution image from noisy undersampled multiframes*, IEEE Trans. on Acoust., Speech, and Signal Process, 38 (1990) , pp. 1013–1027.

[8] R. R. Schultz and R. L Stevenson., *Extraction of high-resolution frames from video sequences*, IEEE T. Image Proces., 5 (1996), pp. 996–1011.

[9] H. Sorensen and C. Burrus, *Fast DFT and convolution algorithms*, Handbook of Signal Processing, edited by S. Mitra and J. Kaiser, New York, Wiley.

[10] A. M. Tekalp, M. K. Ozkan and M. I. Sezan, *High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration*, In Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process, III, pp. 169–172, San Francisco, CA, March 1992.

[11] T. F. Chan and J. A. Olkin., *Circulant preconditioners for Toeplitz-block matrices*, Numer. Algo., 6 (1994) , pp. 89–101.

[12] R. Y. Tsai and T. S. Huang., *Multiframe image restoration and registration*, Advances in Computer Vision and Image Processing, 1 (1984), pp. 317–339.

Deblurring Models with Neumann Boundary Conditions

Abstract

Blur removal is an important problem in signal and image processing. The blurring matrices obtained by using the zero boundary condition (corresponding to assuming dark background outside the scene) are Toeplitz matrices for 1-dimensional problems and block-Toeplitz-Toeplitz-block matrices for 2-dimensional cases. They are computationally intensive to invert especially in the block case. Using periodic boundary conditions, the matrices become (block) circulant and can be diagonalized by discrete Fourier transform matrices. However, both boundary conditions easily lead to boundary artifacts. In this paper, we propose to use the Neumann boundary condition (corresponding to a reflection of the original scene at the boundary). The resulting matrices are (block) Toeplitz-plus-Hankel matrices. We show that for symmetric blurring functions, these blurring matrices can always be diagonalized by discrete cosine transform matrices. Thus the cost of inversion is much cheaper than that of using zero or periodic boundary conditions. Moreover, experimental results also show that the boundary artifacts are much less prominent than that of using the other two boundary conditions.

1 Introduction

A fundamental issue in signal and image processing is blur removal. The signal or image obtained from a point source under the blurring process is called the impulse

response function or the point spread function. The observed signal or image g is just the convolution of this blurring function h with the "true" signal or image f. The deblurring problem is to recover f from the blurred function g given the blurring function h. This basic problem appears in many forms in signal and image processing [2, 4, 12, 14].

In practice, the observed signal or image g is of finite length (and width) and we use it to recover a finite section of f. Because of the convolution, g is not completely determined by f in the same domain where g is defined. More precisely, if g is defined on the interval $[a, b]$ say, then g is not completely determined by the values of f on $[a, b]$. It is also affected by the values of f close to the boundary of $[a, b]$ because of the convolution. How far away from $[a, b]$ will these values of f affect g depends on the support of the blurring function h. Thus in solving f from a finite length g, we need some assumptions on the values of f outside the domain where g is defined. These assumptions are called boundary conditions.

The natural and classical approach is to use the zero (Dirichlet) boundary condition [2, pp.211–220]. This means that the values of f outside the domain of consideration are zero. This results in a blurring matrix which is a Toeplitz matrix in 1-dimensional cases and a block-Toeplitz-Toeplitz-block matrix in 2-dimensional cases, see [2, p.71]. However, these matrices are known to be computationally intensive to invert, especially in 2-dimensional cases, see [2, p.126]. Also ringing effect will appear at the boundary if the data are indeed not close to zeros outside the domain.

One way to alleviate the computational cost is to assume the periodic boundary condition, i.e., data outside the domain of consideration are exact copies of data inside [12, p.258]. The resulting blurring matrix is a circulant matrix in 1-dimensional cases and a block-circulant-circulant-block matrix in 2-dimensional cases. These matrices can be diagonalized by discrete Fourier matrices and hence their inverses can easily be found by using Fast Fourier Transforms, see [12, p.258]. However, ringing will also appear at the boundary unless f is close to periodic, and that is not common in practice.

An analysis on this problem can be found in [19], it proposed two methods to reduce the effect of boundary distortion, but the resulting matrix cannot be inverted in a fast way. In this paper, we propose to use the Neumann (reflective) boundary condition. It sets the data outside the domain of consideration as reflection of data inside. The resulting blurring matrix is a Toeplitz-plus-Hankel matrix in 1-dimensional cases and a block Toeplitz-plus-Hankel matrix with Toeplitz-plus-Hankel blocks in 2-dimensional cases. We show that although these matrices have more complicated structures, they can always be diagonalized by discrete cosine transform matrices provided that the blurring function h is symmetric. Thus their inverses can be obtained by using three fast cosine transforms (FCTs) (one for finding the eigenvalues of the blurring matrix and two for solving the system, see (14) below). Because an FCT requires only real multiplications and can be done at half of the cost of an FFT, see [20, pp.59–60], inversion of these matrices is faster than that of those matrices obtained from zero or periodic boundary conditions. Moreover, experimental results show that the boundary artifacts are much less prominent by using the Neumann boundary condition, see §5. We remark that blurring functions are usually symmetric, see [14, p.269] and Figure 4 in §5.

The outline of the paper is as follows. In §2, we introduce the three different boundary conditions. In §3, we show that blurring matrices obtained from the Neumann boundary condition can be diagonalized by discrete cosine transform matrices. In §4, we recall Tikhonov regularization, which is to be used in §5, the section on numerical experiments. In §5, we illustrate by numerical examples from signal and image restorations that boundary artifacts are minimized by using the Neumann boundary condition. Concluding remarks are given in §6.

2 One-Dimensional Deblurring Problem

We begin with the 1-dimensional deblurring problem. Consider the original signal

$$\bar{\mathbf{f}} = (\cdots, f_{-m+1}, \cdots, f_0, f_1, \cdots, f_n, f_{n+1}, \cdots, f_{n+m}, \cdots)^t$$

and the blurring function given by

$$\mathbf{h} = (\cdots, 0, 0, h_{-m}, h_{-m+1}, \cdots, h_0, \cdots, h_{m-1}, h_m, 0, 0, \cdots)^t. \tag{1}$$

The blurred signal is the convolution of $\mathbf{h}$ and $\bar{\mathbf{f}}$, i.e., the i-th entry g_i of the blurred signal is given by

$$g_i = \sum_{j=-\infty}^{\infty} h_{i-j} f_j. \tag{2}$$

The deblurring problem is to recover the vector $(f_1, \cdots, f_n)^t$ given the blurring function $\mathbf{h}$ and a blurred signal $(g_1, \cdots, g_n)^t$ of finite length. From (2), we have

$$\begin{pmatrix} h_m & \cdots & h_0 & \cdots & h_{-m} & & & & \\ & h_m & & h_0 & & h_{-m} & & 0 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & 0 & & & h_m & & h_0 & & h_{-m} \\ & & & & & h_m & \cdots & h_0 & \cdots & h_{-m} \end{pmatrix} \begin{pmatrix} f_{-m+1} \\ f_{-m+2} \\ \vdots \\ f_0 \\ f_1 \\ \vdots \\ f_n \\ f_{n+1} \\ \vdots \\ f_{n+m-1} \\ f_{n+m} \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}. \tag{3}$$

Thus the blurred signal $\mathbf{g} = (g_1, \cdots, g_n)^t$ is determined not only by $\mathbf{f} = (f_1, \cdots, f_n)^t$, but by $(f_{-m+1}, \cdots, f_0, f_1, \cdots, f_n, f_{n+1}, \cdots, f_{n+m})^t$. The linear system (3) is underdetermined. To deal with it, we make certain assumptions (called boundary conditions) on the unknown data $f_{-m+1}, \cdots, f_0$, and $f_{n+1}, \cdots, f_{n+m}$ so as to reduce the number of unknowns.

Before we discuss the boundary conditions, let us first rewrite (3) as

$$T_l \mathbf{f}_l + T\mathbf{f} + T_r \mathbf{f}_r = \mathbf{g}, \tag{4}$$

where

$$T_l = \begin{pmatrix} h_m & \cdots & h_1 \\ & \ddots & \ddots \\ & & h_m \\ & & \\ 0 & & \end{pmatrix}, \quad \mathbf{f}_l = \begin{pmatrix} f_{-m+1} \\ f_{-m+2} \\ \vdots \\ f_{-1} \\ f_0 \end{pmatrix}, \tag{5}$$

$$T = \begin{pmatrix} h_0 & \cdots & h_{-m} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ h_m & \ddots & \ddots & \ddots & h_{-m} \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & h_m & \cdots & h_0 \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix}, \tag{6}$$

$$T_r = \begin{pmatrix} & & 0 \\ & & \\ h_{-m} & & \\ \vdots & \ddots & \\ h_{-1} & \cdots & h_{-m} \end{pmatrix}, \quad \text{and} \quad \mathbf{f}_r = \begin{pmatrix} f_{n+1} \\ f_{n+2} \\ \vdots \\ f_{n+m-1} \\ f_{n+m} \end{pmatrix}. \tag{7}$$

Here and in the following, we assume for simplicity that $m \leq n$. The case where $m > n$ can be treated similarly. We note that T_l, T and T_r above are n-by-m, n-by-n and n-by-m Toeplitz matrices respectively. (If $m > n$, they all will be n-by-n Toeplitz matrices.)

2.1 The Zero (Dirichlet) Boundary Conditions

The zero (or Dirichlet) boundary condition assumes that the signal outside the domain of the observed vector $\mathbf{g}$ is zero [2, pp.211–220], i.e.,

$$\mathbf{f}_l = \mathbf{f}_r = \mathbf{0},$$

the zero vector. The matrix system in (4) becomes

$$T\mathbf{f} = \mathbf{g}. \tag{8}$$

We see from (6) that the coefficient matrix T is a Toeplitz matrix.

There are many iterative or direct Toeplitz solvers that can solve the Toeplitz system (8) in cost ranging from $O(n \log n)$ to $O(n^2)$ operations depending on the matrix T, see for instance [18, 16, 1, 7]. However, boundary artifacts will in general appear near the two ends of $\mathbf{f}$ as we have ignored the contribution of $\bar{\mathbf{f}}$ outside $\mathbf{f}$ by setting them to zero, see the numerical results in §5 for instance.

2.2 Periodic Boundary Conditions

Besides boundary artifacts, another drawback of the zero boundary condition is that the resulting Toeplitz system (8) is computationally intensive to solve, especially in 2-dimensional cases, see [2, p.126]. For practical applications, where we need to solve the system efficiently, one usually resorts to the periodic boundary condition, i.e.,

$$f_j = f_{n-j} \quad \text{for all } j,$$

see [12, p.258]. The matrix system in (4) becomes

$$B\mathbf{f} = [(0 \mid T_l) + T + (T_r \mid 0)]\,\mathbf{f} = \mathbf{g}, \tag{9}$$

where $(0 \mid T_l)$ and $(T_r \mid 0)$ are n-by-n Toeplitz matrices obtained by augmenting $(n-m)$ zero columns to T_l and T_r respectively. (If $m > n$, then no augmentation is required.)

One important observation is that B thus obtained is a circulant matrix. Hence B can be diagonalized by the discrete Fourier matrix and (9) can be solved by using three fast Fourier transforms (FFTs) (one for finding the eigenvalues of the matrix B and two for solving the system, cf (14) below). However practical signals and images usually do not satisfy the periodic assumptions and ringing effects will appear on the boundary, see the numerical results in §5 for instance.

2.3 Neumann Boundary Conditions

In this paper, we propose to deal with the underdetermined system (3) by reflection of the data at the boundary, i.e., we assume that the data outside $\mathbf{f}$ are a reflection of

data inside **f**. More precisely, we assume that

$$\begin{cases} f_0 &= f_1 \\ \vdots & \vdots \quad \vdots \\ f_{-m+1} &= f_m \end{cases} \quad \text{and} \quad \begin{cases} f_{n+1} &= f_n \\ \vdots & \vdots \quad \vdots \\ f_{n+m} &= f_{n-m+1} \end{cases}$$

Thus (4) becomes

$$A\mathbf{f} = [(0 \mid T_l)J + T + (T_r \mid 0)J]\,\mathbf{f} = \mathbf{g}, \tag{10}$$

where J is the n-by-n reversal matrix.

We remark that the coefficient matrix A in (10) is neither Toeplitz nor circulant. It is a Toeplitz-plus-Hankel matrix. Although these matrices have more complicated strcutures, we will show in §3 that the matrix A can always be diagonalized by discrete cosine transform matrices provided that the blurring function **h** is symmetric, i.e., $h_j = h_{-j}$ for all j in (1). It follows that (10) can be solved by using three fast cosine transforms (FCTs) in $O(n \log n)$ operations, see (14) below. This approach is computationally attractive as the FCT requires only real operations and is about twice as fast as the FFT, see [20, pp.59–60]. Thus solving a problem with Neumann boundary conditions is twice as fast as solving a problem with periodic boundary conditions. Experimental results in §5 show that it produces less ringing effects than that of the zero or periodic boundary assumptions.

3 Diagonalization of the Neumman Blurring Matrices

3.1 One-Dimensional Problems

We first review some definitions and properties of discrete cosine transform matrices. Let C be the n-by-n discrete cosine transform matrix with entries

$$[C]_{i,j} = \sqrt{\frac{2 - \delta_{i1}}{n}} \cos\left(\frac{(i-1)(2j-1)\pi}{2n}\right), \quad 1 \le i, j \le n,$$

where δ_{ij} is the Kronecker delta, see Jain [14, p.150]. We note that C is orthogonal, i.e., $C^tC = CC^t = I$. Also, for any n-vector $\mathbf{v}$, the matrix-vector multiplications $C\mathbf{v}$ and $C^t\mathbf{v}$ can be computed in $O(n \log n)$ real operations by FCTs; see [20, pp.59–60].

Let $\mathcal{C}$ be the space containing all matrices that can be diagonalized by C, i.e.

$$\mathcal{C} = \{C^t \Lambda C \mid \Lambda \text{ is an } n\text{-by-}n \text{ real diagonal matrix}\}. \tag{11}$$

Let $Q = C^t \Lambda C \in \mathcal{C}$. By multiplying $(1, 0, \ldots, 0)^t$ to both sides of $CQ = \Lambda C$, we see that the eigenvalues $[\Lambda]_{i,i}$ of Q are given by

$$[\Lambda]_{i,i} = \frac{1}{[\mathbf{q_1}]_i}[C\mathbf{q_1}]_i, \quad i = 1, \ldots, n, \tag{12}$$

where $\mathbf{q_1}$ is the first column vector of Q. Hence, the eigenvalues of Q can be obtained by taking an FCT of the first column of Q. In particular, any matrix in $\mathcal{C}$ is uniquely determined by its first column.

Next we give a characterization of this $\mathcal{C}$ class of matrices. Let us define the shift of any vector $\mathbf{v} = (v_0, \ldots, v_{n-1})^t$ as $\sigma(\mathbf{v}) \equiv (v_1, v_2, \ldots, v_{n-1}, 0)^t$. Define $T(\mathbf{v})$ to be the n-by-n symmetric Toeplitz matrix with $\mathbf{v}$ as the first column and $H(\mathbf{v})$ to be the n-by-n Hankel matrix with $\mathbf{v}$ as the first column and $J\mathbf{v}$ as the last column.

Lemma 1 (Chan, Chan, and Wong [6], and Kailath and Olshevsky [15]) *Let $\mathcal{C}$ be the class of all matrices that can be diagonalized by the discrete cosine transform matrix C. Then*

$$\mathcal{C} = \{T(\mathbf{v}) + H(\sigma(\mathbf{v})) \mid \mathbf{v} = (v_0, \ldots, v_{n-1})^t \in \mathbb{R}^n\}.$$

It follows from Lemma 1 that matrices that can be diagonalized by C are some special Toeplitz-plus-Hankel matrices.

Theorem 1 *Let the blurring function* $\mathbf{h}$ *given in (1) be symmetric, i.e., $h_j = h_{-j}$ for all $|j| \leq m$. Then the matrix A given in (10) can be written as*

$$A = T(\mathbf{u}) + H(\sigma(\mathbf{u})) \tag{13}$$

where $\mathbf{u} = (h_0, h_1, \cdots, h_m, 0, \cdots, 0)^t$. In particular, A can be diagonalized by C.

Proof: From (10) and (6), it is clear that T in (10) is equal to $T(\mathbf{u})$. From the definitions of T_l and T_r in (5) and (7), it is also obvious that

$$(0 \mid T_l) + (T_r \mid 0) = T(J\sigma(\mathbf{u})).$$

Hence

$$(0 \mid T_l)J + (T_r \mid 0)J = H(\sigma(\mathbf{u})). \qquad \square$$

By Theorem 1, the solution $\mathbf{f}$ of (10) is given by

$$\mathbf{f} = C\Lambda^{-1}C^t\mathbf{g}, \tag{14}$$

where Λ is the diagonal matrix holding the eigenvalues of A. By (12), Λ can be obtained in one FCT. Hence $\mathbf{f}$ can be obtained in three FCTs.

We remark that from (13), it is straightforward to construct the Neumann blurring matrix A from the Dirichlet blurring matrix $T = T(\mathbf{u})$ in (6). All we need is to reflect the first column of T to get the Hankel matrix $H(\sigma(\mathbf{u}))$ and add it to T. Clearly the storage requirement of both matrices A and T are the same – we need only to store the first column. However, the ringing effect at the boundary is much less than that of using T, see §5.

3.2 Two-Dimensional Problems

The results of §3.1 can be extended in a natural way to 2-dimensional image restoration problems. In this case, one is still concerned with solving a least squares problem similar to that in (3), except that the matrix is now a block matrix. For zero boundary condition, the resulting blurring matrix is a block-Toeplitz-Toeplitz-block matrix of

the form

$$T = \begin{pmatrix} T^{(0)} & \cdots & T^{(-m)} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ T^{(m)} & \ddots & \ddots & \ddots & T^{(-m)} \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & T^{(m)} & \cdots & T^{(0)} \end{pmatrix} \tag{15}$$

where each block $T^{(j)}$ is a Toeplitz matrix. The first column and row of T in (15) is completely determined by the point spread function of the blurring process.

With the Neumann boundary condition, the resulting matrix A is a block Toeplitz-plus-Hankel matrix with Toeplitz-plus-Hankel blocks. More precisely,

$$A = \begin{pmatrix} A^{(0)} & \cdots & A^{(-m)} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ A^{(m)} & \ddots & \ddots & \ddots & A^{(-m)} \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & A^{(m)} & \cdots & A^{(0)} \end{pmatrix} + \begin{pmatrix} A^{(1)} & \cdots & A^{(m-1)} & & 0 \\ \vdots & \iddots & & & \\ A^{(m-1)} & & & & A^{(-m+1)} \\ & & & \iddots & \vdots \\ 0 & & A^{(-m+1)} & \cdots & A^{(1)} \end{pmatrix} \tag{16}$$

with each block $A^{(j)}$ being an n-by-n matrix of the form given in (10). We note that the $A^{(j)}$ in (16) and the $T^{(j)}$ in (15) are related by (13). Thus again it is straightforward to construct the blurring matrix A from the matrix T or from the point spread function directly. Obviously, storage requirements of A and T are the same. We next show that for a symmetric point spread function, the blurring matrix A in (16) can be diagonalized by the 2-dimensional discrete cosine transform matrix. Hence inversion of A can be done by using only three 2-dimensional FCTs.

Theorem 2 *If the point-spread function is symmetric, i.e., A in (16) is symmetric, then A can be diagonalized by the 2-dimensional discrete cosine transform matrix $(C \otimes C)$, where $\otimes$ is the tensor product.*

Proof: We note that $C \otimes C = (C \otimes I)(I \otimes C)$. Since each block $A^{(j)}$ in (16) is of the form given by (13), by Theorem 1, $A^{(j)}$ can be diagonalized by C, i.e.,

$$CA^{(j)}C^t = \Lambda^{(j)}, \quad j = 1, \ldots, m.$$

It follows that

$$\begin{aligned}(C \otimes C)A(C^t \otimes C^t) &= (C \otimes I)(I \otimes C)A(I \otimes C^t)(C^t \otimes I) \\ &= (C \otimes I)\Lambda(C^t \otimes I)\end{aligned}$$

where

$$\Lambda = \begin{pmatrix} \Lambda^{(0)} & \cdots & \Lambda^{(m)} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ \Lambda^{(m)} & \ddots & \ddots & \ddots & \Lambda^{(m)} \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & \Lambda^{(m)} & \cdots & \Lambda^{(0)} \end{pmatrix} + \begin{pmatrix} \Lambda^{(1)} & \cdots & \Lambda^{(m-1)} & & 0 \\ \vdots & \iddots & & & \\ \Lambda^{(m-1)} & & & & A^{(m-1)} \\ & & & \iddots & \vdots \\ 0 & & \Lambda^{(m-1)} & \cdots & \Lambda^{(0)} \end{pmatrix}. \quad (17)$$

Let P be the permutation matrix that satisfies

$$[P^t \Lambda P]_{i,j;k,\ell} = [\Lambda]_{k,\ell;i,j}, \quad 1 \le i, j \le n, \ 1 \le k, \ell \le n,$$

i.e. the (i,j)th entry of the (k,ℓ)th block in Λ is permuted to the (k,ℓ)th entry of the (i,j)th block. Then we have $P^t(C \otimes I)P = (I \otimes C)$ and

$$P^t \Lambda P = \tilde{A} = \begin{pmatrix} \tilde{A}^{(1)} & & & 0 \\ & \tilde{A}^{(2)} & & \\ & & \ddots & \\ 0 & & & \tilde{A}^{(n)} \end{pmatrix}.$$

From (17), each matrix $\tilde{A}^{(j)}$ has the same form as A in (13). In particular, for all j, $C\tilde{A}^{(j)}C^t = \tilde{\Lambda}^{(j)}$, a diagonal matrix. Thus

$$\begin{aligned}(C \otimes C)A(C^t \otimes C^t) &= (C \otimes I)\Lambda(C^t \otimes I) = PP^t(C \otimes I)PP^t\Lambda PP^t(C^t \otimes I)PP^t \\ &= P(I \otimes C)\tilde{A}(I \otimes C^t)P^t = P \begin{pmatrix} \tilde{\Lambda}^{(1)} & & & 0 \\ & \tilde{\Lambda}^{(2)} & & \\ & & \ddots & \\ 0 & & & \tilde{\Lambda}^{(n)} \end{pmatrix} P^t,\end{aligned}$$

which is a permutation of a diagonal matrix and hence is still a diagonal matrix. □

In §5, we will present examples in signal and image restoration to illustrate that the Neumann boundary condition produces less boundary artifacts than the periodic and zero boundary conditions.

4 Tikhonov Regularization

Besides the issue of boundary conditions, it is well-known that blurring matrices are in general ill-conditioned and deblurring algorithms can be extremely sensitive to noise [12, p.282]. The ill-conditioning of the blurring matrices stems from the wide range of the magnitudes of their eigenvalues [10, p.31]. Therefore, excess amplification of the noise at small eigenvalues can occur. The method of regularization can be used to achieve stability for deblurring problems. In the classical Tikhonov regularization [10, p.117], stability is attained by introducing a regularization operator D which restricts the set of admissible solutions. More specifically, the regularized solution $\mathbf{f}(\mu)$ is computed as the solution to

$$\min_{\mathbf{f}(\mu)} \left\{ \mu \|D\mathbf{f}(\mu)\|_2^2 + \|\mathbf{g} - A\mathbf{f}(\mu)\|_2^2 \right\}. \tag{18}$$

The term $\|D\mathbf{f}(\mu)\|_2^2$ is added in order to regularize the solution. The regularization parameter μ controls the degree of regularity (i.e., degree of bias) of the solution.

One can find the solution $\mathbf{f}(\mu)$ in (18) by solving the normal equations

$$(\mu D^t D + A^t A)\mathbf{f}(\mu) = A^t \mathbf{g}. \tag{19}$$

In most applications [14, 8, 12], $\|D\mathbf{f}\|_2$ is chosen to be the L_2 norm $\|\mathbf{f}\|_2$ or the H_1 norm $\|L\mathbf{f}\|_2$ where L is the first order difference operator matrix. Correspondingly, the matrix $D^t D$ in (19) is the identity matrix or the 2-dimensional discrete Laplacian matrix with some boundary conditions. In the latter case, if the zero boundary condition is imposed, $D^t D$ is just the 2-dimensional discrete Laplacian with the Dirichlet boundary condition. For periodic boundary conditions, $D^t D$ is circulant and can be diagonalized by the FFTs, see for instance [12, p.283]. For the Neumann boundary condition, $D^t D$

is the 2-dimensional discrete Laplacian with the Neumann boundary condition, which can be diagonalized by the discrete cosine transform matrix, see for instance [3].

Thus if we use the Neumann boundary condition for both the blurring matrix A and the regularization operator D^tD, then the matrix in (19) can be diagonalized by the discrete cosine transform matrix and hence its inversion can still be done in three FCTs.

5 Numerical Experiments

In this section, we illustrate by numerical tests from signal and image restoration problems, the usefulness of employing Neumann boundary conditions over the other two boundary conditions. All our tests were done using Matlab. The data we used are generated from two images, see Figure 1. The first source image Lena is a woman face with background, and contains a high degree of contrast and details. The second source image, a satellite, has a dark background, so the Neumann, periodic and zero boundary conditions should all work well.

From (4), we see that to construct the right hand side vector $\mathbf{g}$ correctly, we need the vectors $\mathbf{f}_l$ and $\mathbf{f}_r$, i.e. we need to know the signal or image outside the given domain. Thus we start with the 256-by-256 images of Lena and the satellite and cut out a 128-by-128 portion from each of them. For Lena's image, we choose the portion where her face is at the center and for the satellite image, we just choose the center portion. Figure 1 gives the 128-by-128 images of both pictures.

5.1 Signal Restoration

We try restoring signals blurred by the following two blurring functions, see [14, p.269]:

(i) a truncated Gaussian blur:

$$h_i = \begin{cases} ce^{-0.1i^2}, & \text{if } |i| \leq 8, \\ 0, & \text{otherwise,} \end{cases}$$

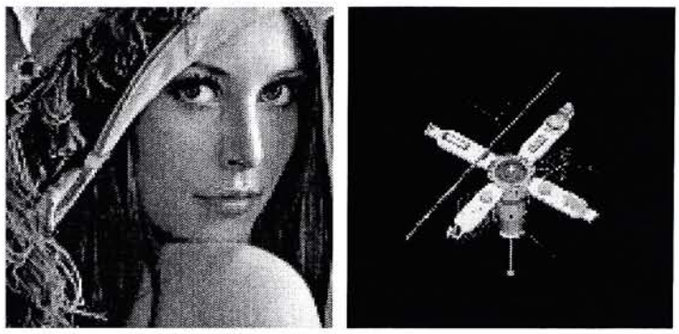

Figure 1: True image: Lena (left) and the satellite (right).

(ii) a rectangular scanning aperture:

$$h_i = \begin{cases} c, & \text{if } |i| \leq 4, \\ 0, & \text{otherwise,} \end{cases}$$

where c is the normalization constant such that $\sum_i h_i = 1$. We remark that the blurring functions are 1-D analog of the 2-D Gaussian blur and out-of-focus blur, see Figure 4. Clearly both blurring functions are symmetric.

The source signals are generated from the columns in the 128-by-128 images in Figure 1. Thus each column in the images gives a signal of 128 entries. We note that the values of the images outside the 128-by-128 range are known as the images are cut out from 256-by-256 images. Hence we can use these values to generate the correct blurred signals (right hand side vectors) $\mathbf{g}$ in (4). Measurement noise $\mathbf{n}$ is then added to $\mathbf{g}$ using a Gaussian white noise generator. For simulation purposes, the blurred signal-to-noise (SNR) ratio, $10\log_{10}(\|H\mathbf{f}\|_2^2/\|\mathbf{n}\|_2^2)$, is set to 30 dB.

We restore the signal by solving the regularized least squares problem (18) where the regularization functional is the L_2 norm. The normal equations (19) become

$$(\mu I + A^t A)\mathbf{f}(\mu) = A^t(\mathbf{g} + \mathbf{n}). \tag{20}$$

The regularization parameter μ is chosen automatically by the L-curve method using the routine `l_corner.m` in [13]. To measure the accuracy of the results, we compute the relative errors $\mathbf{e}_r$ of the restored signals, which are defined as $\mathbf{e}_r = \|\mathbf{f}_r(\mu) - \mathbf{f}\|_2/\|\mathbf{f}\|_2$

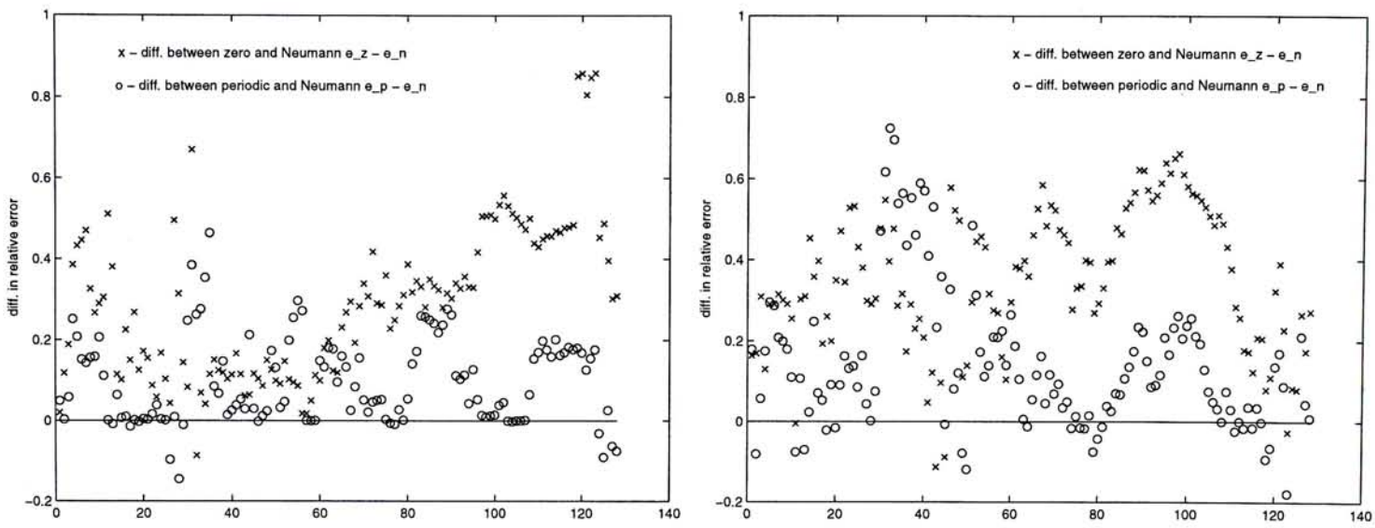

Figure 2: Lena with Gaussian blur (left) and out-of-focus blur (right).

where $\mathbf{f}$ is the original signal and $\mathbf{f}_r(\mu)$ is the restored signal of (20) with the given boundary condition ($r = z, p, n$ for zero, periodic and Neumann boundary conditions respectively). In Figures 2–3, the symbol "$\times$" represents $(\mathbf{e}_z - \mathbf{e}_n)$ whereas the symbol "$\circ$" represents $(\mathbf{e}_p - \mathbf{e}_n)$. Thus in the figures, if the point is above the line (the difference in relative error $= 0$), then using the Neumann boundary condition is a better method for restoring that signal. We clearly see from the figures that in most of the cases, the Neumann boundary condition gives the best results.

As for the computational cost, we only need to apply three FFTs and FCTs to compute the restored signals for the periodic and the Neumann boundary condition respectively, see (14). Thus the costs are both of $O(n \log n)$ operations, though FCT is about twice as fast as FFT, see [20, pp.59–60]. For the zero boundary condition, we have to solve Toeplitz systems. The fastest direct solvers require $O(n \log^2 n)$ operations, see [1]. If we use iterative methods, such as the conjugate gradient method with or without preconditioning [7], then the cost *per iteration* is more than two $2n$-length FFTs, i.e., about 4 n-length FFTs, see [7]. Thus the cost of using zero boundary conditions is larger than those of using the periodic and the Neumann boundary condition.

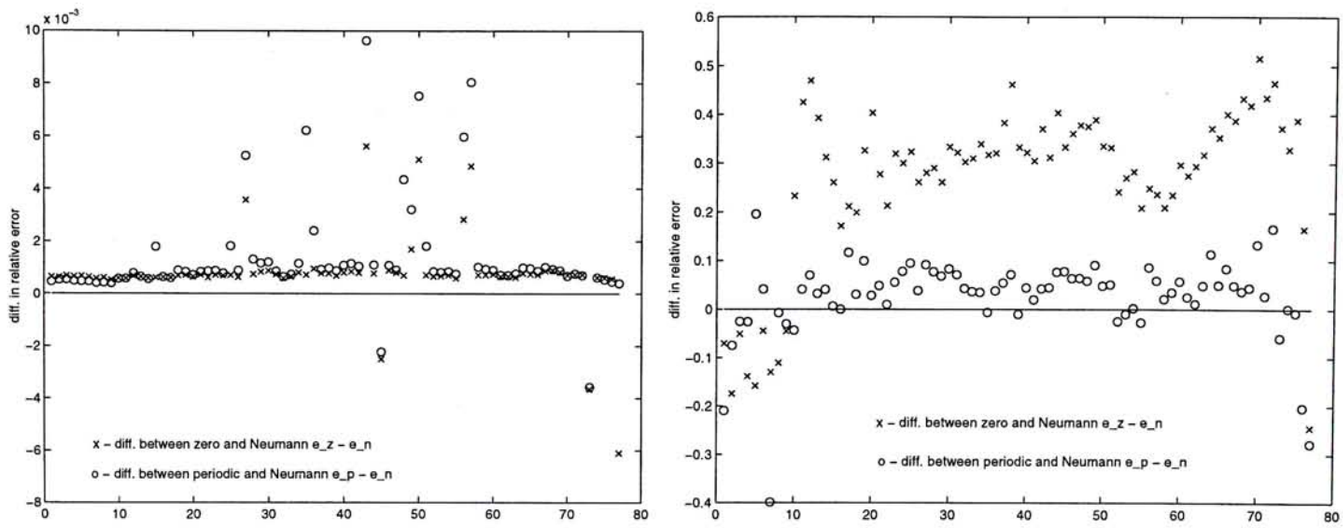

Figure 3: Satellite with Gaussian blur (left) and out-of-focus blur (right).

5.2 Image Restoration

Next we consider restoring the two images in Figure 1 blurred by the following two blurring functions, see [14, p.269]:

(i) a truncated Gaussian blur:

$$h_{i,j} = \begin{cases} ce^{-0.1(i^2+j^2)}, & \text{if } |i-j| \leq 8, \\ 0, & \text{otherwise,} \end{cases}$$

(ii) an out-of-focus blur:

$$h_{i,j} = \begin{cases} c, & \text{if } i^2+j^2 \leq 4, \\ 0, & \text{otherwise,} \end{cases}$$

where $h_{i,j}$ is the jth entry of the first column of $T^{(i)}$ in (15) and c is again the normalization constant such that $\sum_{i,j} h_{i,j} = 1$. We remark that the Gaussian blur is symmetric and separable and the out-of-focus blur is symmetric but not separable, see Figure 4.

The blurred images (right hand side vector) $\mathbf{g}$ are generated by using a block version of (4) and exact values of $\mathbf{f}_l$ and $\mathbf{f}_r$ outside the 128-by-128 region are used. (Recall

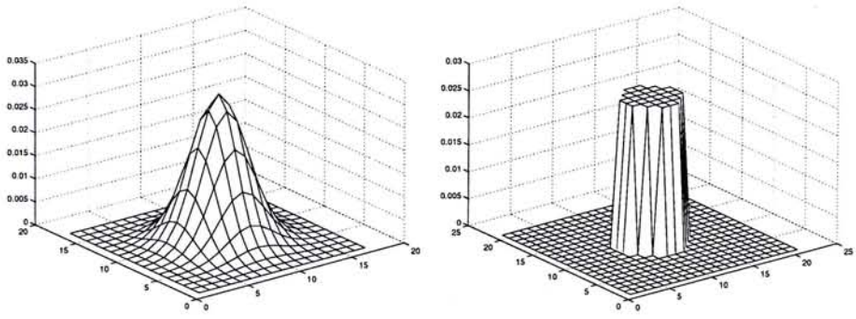

Figure 4: Gaussian (atmospheric turbulence) blur (left) and out-of-focus blur (right).

Figure 5: Noisy and blurred Lena image by Gaussian (left) and out-of-Focus blur (right).

that the images in Figure 1 are cut from the 256-by-256 images.) Gaussian white noise $\mathbf{n}$ with signal-to-noise ratio of 30dB is then added to the blurred images. The noisy blurred images are shown in Figures 5–6. Again, we use the L_2 norm as the regularization functional and we get (20), except the matrix A is now in block form.

We remark that we cannot use the routine `l_corner.m` in [13] to automatically locate the corner points (the regularization parameter μ) of the L-curves as the routine requires the SVD of the blurring matrices which is computationally infeasible. We therefore choose the corner points manually. As a comparison, we also choose the μ that gives the best restored picture visually. In Figures 7–14, we present restorations with three different boundary conditions and these two different choices of μ.

Recall that we only need to apply three 2-dimensional FFTs and FCTs to compute the restored images for the periodic and the Neumann boundary condition respectively. Thus the costs for both approaches are about $O(n^2 \log n)$ operations though the Neu-

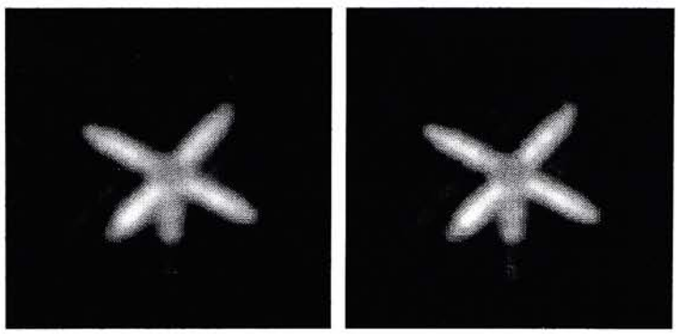

Figure 6: Noisy and Blurred satellite image by Gaussian (left) and out-of-Focus blur (right).

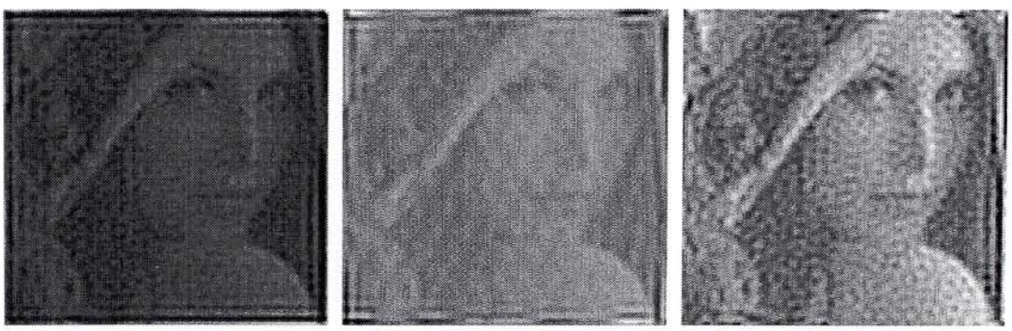

Figure 7: Restoring Gaussian blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen by L-curve.

Figure 8: Restoring Gaussian blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen visually.

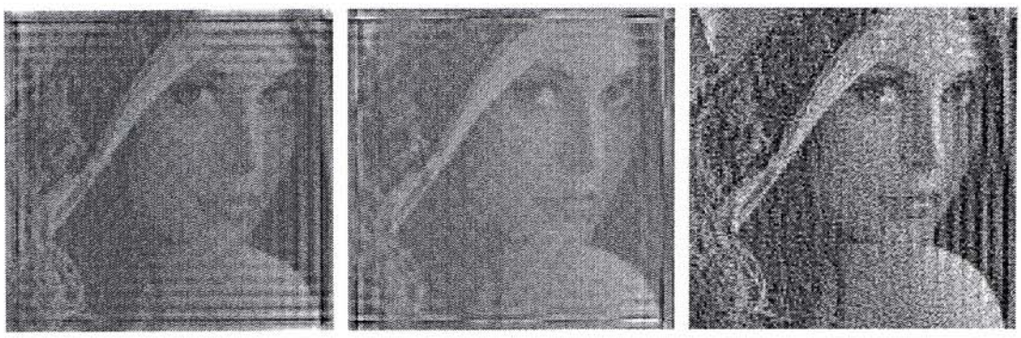

Figure 9: Restoring out-of-focus blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen by L-curve.

Figure 10: Restoring out-of-focus blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen visually.

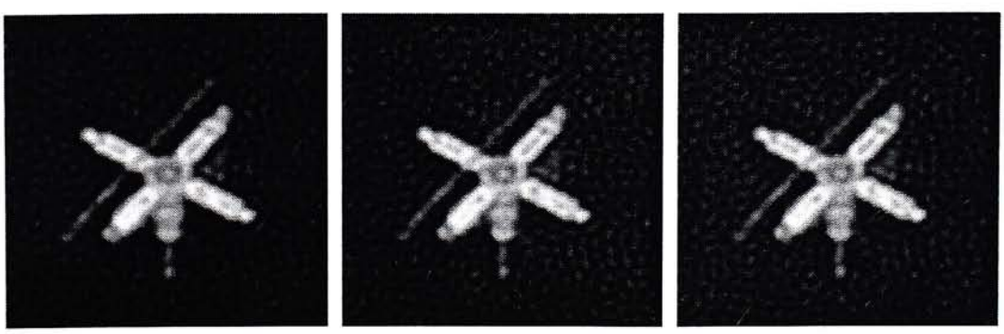

Figure 11: Restoring Gaussian blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen by L-curve.

Figure 12: Restoring Gaussian blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen visually.

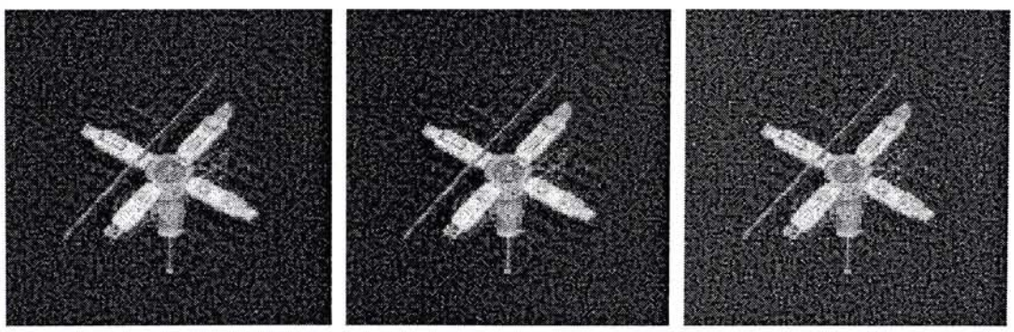

Figure 13: Restoring out-of-focus blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen by L-curve.

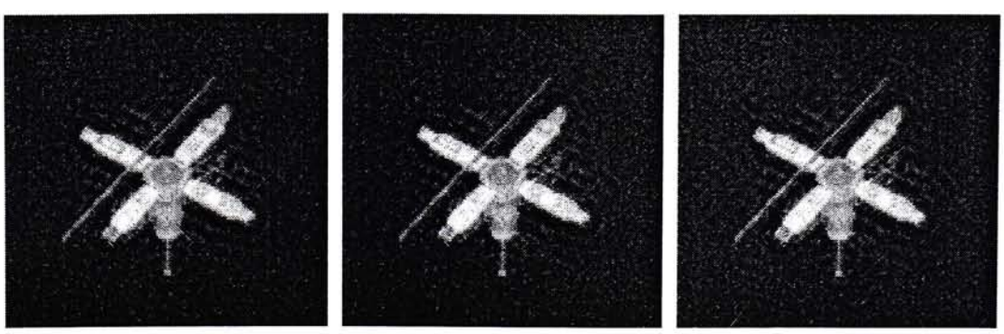

Figure 14: Restoring out-of-focus blur with zero boundary (left), periodic boundary (middle) and Neumann boundary (right) conditions, and μ chosen visually.

mann one will be twice as fast. For the zero boundary condition, we have to solve large block-Toeplitz-Toeplitz-block systems. The fastest direct Toeplitz solvers require $O(n^4)$ operations, see [17]. In our tests, the systems are solved by the preconditioned conjugate gradient method [11, p.516]. We note that the cost *per iteration* is more than four 2-dimensional FFTs [7]. To speed up the convergence, we have used the optimal cosine-transform preconditioners, see [6]. It is defined as the matrix in $\mathcal{C}$ (cf. (11)) that is closest to the blurring matrix in the Frobenius norm. Numerical results show that they perform better than the optimal circulant preconditioners proposed by T. Chan, see [7]. The restored images in Figures 7–14 (left ones) require 40, 7, 16, 20, 21, 20, 17, and 13 iterations respectively for a stopping tolerance of 10^{-6}. Thus as in the 1-dimensional cases, the cost of using zero boundary conditions is significantly larger than those of using the periodic and the Neumann boundary conditions. We see from the Figures 7–10 that the restored images using the Neumann image boundary conditions are better than the others. For the satellite image, the restored images by three different boundary conditions look similar, see Figures 11–14.

In summary, these experiments suggest that the Neumann boundary condition provides an effective model for signal and image restoration problems, both in terms of the computational cost and the quality of the restored signals and images.

6 Concluding Remarks

In this paper, we have shown that discrete cosine transform matrices can diagonalize dense (block) Toeplitz-plus-Hankel blurring matrices arising from using the Neumann (reflective) boundary condition. It is interesting to observe that discrete sine transform matrices can diagonalize Toeplitz matrices with at most 3 bands (such as the discrete Laplacian with zero boundary conditions) but not dense Toeplitz matrices in general, see [9] for instance.

Because any matrices in $\mathcal{C}$ are symmetric (see (11)), discrete cosine transform matrices can only diagonalize blurring matrices from symmetric blurring functions. For

nonsymmetric blurring functions, matrices in $\mathcal{C}$ may be used as preconditioners to speed up the convergence of iterative methods, see for instance [5].

References

[1] G. Ammar and W. Gragg, *Superfast Solution of Real Positive Definite Toeplitz Systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[2] H. Andrew and B. Hunt, *Digital Image Restoration*, Prentice-Hall, New Jersey, 1977.

[3] M. Buckley, *Fast Computation of a Discretized Thin-plate Smoothing Spline for Image Data*, Biometrika, 81 (1994), pp. 247–258.

[4] K. Castleman, *Digital Image Processing*, Prentice–Hall, NJ, 1996.

[5] R. Chan, T. Chan, M. Ng, W. Tang, and C. Wong, *Preconditioned Iterative Methods for High-resolution Image Reconstruction with Multisensors, Proceedings to the SPIE Symposium on Advanced Signal Processing: Algorithms, Architectures, and Implementations, Vol. 3461*, San Diego CA, July, 1998, Ed: F. Luk.

[6] R. Chan, T. Chan, and C. Wong, *Cosine Transform Based Preconditioners for Total Variation Minimization Problems in Image Processing*, Iterative Methods in Linear Algebra, II, V3, IMACS Series in Computational and Applied Mathematics, Proceedings of the Second IMACS International Symposium on Iterative Methods in Linear Algebra, Bulgaria, June, 1995, pp. 311–329, Ed: S. Margenov and P. Vassilevski.

[7] R. Chan and M. Ng, *Conjugate Gradient Methods for Toeplitz Systems*, SIAM Review, 38 (1996), pp. 427–482.

[8] R. Chan, M. Ng and R. Plemmons, *Generalization of Strang's Preconditioner with Applications to Toeplitz Least Squares Problems*, J. Numer. Linear Algebra Appls., 3 (1996), pp. 45–64.

[9] R. Chan, M. Ng, and C. Wong, *Sine Transform Based Preconditioners for Symmetric Toeplitz Systems*, Linear Algebra Appls., 232 (1996), 237–260.

[10] H. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, The Netherlands, 1996.

[11] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[12] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison Wesley, New York, 1992.

[13] P.C. Hansen, *Regularization Tools: a Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems*, Num. Algo., 6 (1994), 1–35.

[14] A. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[15] T. Kailath and V. Olshevsky. *Displacement Structure Approach to Discrete-trigonometric-transform Based Preconditioners of G. Strang Type and T. Chan Type*, Calcolo, 33 (1996), pp. 191–208.

[16] T. Kailath and A. Sayed, *Displacement Structure: Theory and Applications*, SIAM Review, 37 (1995), pp. 297–386.

[17] N. Kalouptsidis, G. Carayannis, and D. Manolakis, *Fast Algorithms for Block Toeplitz Matrices with Toeplitz Entries*, Signal Processing, 6 (1984), pp. 77–81.

[18] N. Levinson, *The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction*, J. Math. and Phys., 25 (1946), pp. 261–278.

[19] F. Luk and D. Vandevoorde, *Reducing boundary distortion in image restoration*, Proc. SPIE, 2296, Advanced Signal Processing Algorithms, Architectures, and Implementations VI, 1994.

[20] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, Boston, 1990.