JOURNAL OF THE
CHUNGCHEONG MATHEMATICAL SOCIETY
Volume **22**, No. 2, June 2009

RESTORATION OF BLURRED IMAGES BY GLOBAL LEAST SQUARES METHOD

SEI-YOUNG CHUNG*, SEYOUNG OH**, AND SUNJOO KWON***

ABSTRACT. The global least squares method (Gl-LSQR) is a generalization of LSQR method for solving linear system with multiple right hand sides. In this paper, we present how to apply this algorithm for solving the image restoration problem and illustrate the usefulness and effectiveness of this method from numerical experiments.

1. Introduction

An image is a signal carrying information about a physical object which cannot be directly observed. The quality of the recorded images is degraded by blurring and noise. Since blurring is the degradation from the process of image formation, it is a deterministic process which has a sufficiently accurate mathematical model for its description.

The goal of the image restoration is to recover a good approximation of the original image X which is $m \times n$, for given the degraded image B, the blur matrix H, and the statistics of the noise vector R. This general linear model is

$$b + r = Hx. \tag{1.1}$$

Here three mn-length vector x, b and r are to rearrange the elements of the images X, B and R into column vectors by stacking the columns of these images.

A blur matrix H is generally ill-conditioned. This implies that round-off errors will blow up and cause the computed solution to be completely

Received February 28, 2009; Revised May 20, 2009; Accepted May 21, 2009.
2000 Mathematics Subject Classification: Primary 65F22, 65K10.
Key words and phrases: image restoration, Tikhonov regularization, Gl-LSQR.
Correspondence should be addressed to SunJoo Kwon, sjkw@cnu.ac.kr.
This study was financially supported by research fund of Chungnam National University in 2008.

inaccurate. It is well known that some regularization will give useful results. Generally a quite effective method for solving (1.1) is the Tikhonov regularization method.

Typically, the implementation of the image restoration problem is a memory intensive application with insurmountable data. One way to reduce memory use is to process the image in blocks. Block processing can produce the same effects and results as processing the image all at once. Specially the process for each block as columns can reduce the execution time([7, 8]).

For efficient storage, we can use memory efficient methods which partition the image domain into regions based on the size of the point spread function([2, 6]). In this paper, we partition the blurred and noisy image into appropriate size blocks instead of using the full image directly. Collection of image restoration problem for each block image brings linear system with multiple right hand side. This can be formulated as minimization problem with respect to the Frobenius norm.

The global least squares (Gl-LSQR) method suggested by F. Toutounian and S. Karimi ([9]) is a generalization of LSQR method for solving linear system with multiple right hand sides,

$$HX = B, \tag{1.2}$$

where B and X are $n \times s$ matrices. This paper suggests Gl-LSQR algorithm as a solver of image restoration problem and describes some advantages of numerical experiments.

The outline of this paper is as follows. The review of the Gl-LSQR method is presented in Section 2. Section 3 illustrates how this method can be implemented for the image restoration. Numerical experiments and concluding remarks are described in Section 4.

2. Global least squares algorithm

This section recalls concept and some properties of Gl-LSQR algorithm in [9].

DEFINITION 2.1. For two $n \times s$ matrices X and Y, $\langle X, Y\rangle_F$ denotes the trace of the square matrix X^TY. Then Frobenius norm is defined by $\|X\|_F = \sqrt{\langle X, X\rangle_F}$.

DEFINITION 2.2. If some $n \times s$ block vectors $V_1, V_2, \ldots$ are orthonormal with respect to $< \cdot\ ,\ \cdot >_F$, then it is F-orthonormal basis.

The global bidiagonalization with starting matrix B is initialized with

$$\beta_1 U_1 = B,\ \alpha_1 V_1 = H^T U_1$$

and for $i = 1, 2, \ldots$ compute

$$\begin{aligned} \beta_{i+1} U_{i+1} &= H V_i - \alpha_i U_i, \\ \alpha_{i+1} V_{i+1} &= H^T U_{i+1} - \beta_{i+1} V_i, \end{aligned} \tag{2.1}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|U_i\|_F = \|V_i\|_F = 1$.

Global bidiagonalization procedure (2.1) constructs the set of the $n \times s$ block vectors $V_1, V_2, \ldots$ and $U_1, U_2, \ldots$ which are two F-orthonormal basis of $R^{n \times ks}$. We can define the matrices $\mathcal{V}_k \equiv \begin{bmatrix} V_1 & V_2 & \ldots & V_k \end{bmatrix}$, $\mathcal{U}_k \equiv \begin{bmatrix} U_1 & U_2 & \ldots & U_k \end{bmatrix}$ and a lower bidiagonal matrix

$$T_k \equiv \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}. \tag{2.2}$$

Proposition 2.3. *The symbol $*$ denote the following product,*

$$\mathcal{V}_k * t = \sum_{j=1}^{k} V_j t_j, \qquad t \in \mathcal{R}^k.$$

Then below relations are satisfied :

$$\begin{aligned} \mathcal{V}_k * (t + s) &= (\mathcal{V}_k * t) + (\mathcal{V}_k * s), \quad s \in \mathcal{R}^k, \\ (\mathcal{V}_k * H_k) * t &= \mathcal{V}_k * (H_k t), \\ \|\mathcal{V}_k * t\|_F &= \|t\|_2 . \end{aligned}$$

Using above Proposition 2.3, the recurrence formula (2.1) of the global bidiagonalization can be rewritten as

$$\begin{aligned} &\mathcal{U}_{k+1} * (\beta_1 e_1) = B, \\ &H\mathcal{V}_k = \mathcal{U}_{k+1} * T_k, \\ &H^T \mathcal{U}_{k+1} = \mathcal{V}_k * T_k^T + \alpha_{k+1} V_{k+1} * e_{k+1}^T, \end{aligned} \tag{2.3}$$

where e_i is the ith column of identity matrix.

We now seek an approximate solution X_k to (1.2) such that $X_k \in span(\mathcal{V}_k)$ and write

$$X_k = \mathcal{V}_k * y_k,\ y_k \in R^k.$$

Then the corresponding residual matrix of the equation (1.2) is

$$\begin{aligned} R_k &= B - HX_k \\ &= B - H\mathcal{V}_k * y_k \\ &= \beta_1 U_1 - (\mathcal{U}_{k+1} * T_k) * y_k \\ &= \beta_1 U_1 - \mathcal{U}_{k+1} * (T_k y_k) \\ &= \mathcal{U}_{k+1} * (\beta_1 e_1 - T_k y_k). \end{aligned} \tag{2.4}$$

The global LSQR algorithm chooses the vector y_k which minimizes $\|R_k\|_F$. Thus by Proposition 2.3 and (2.4),

$$\min \|R_k\|_F = \min_{y_k \in \mathcal{R}^k} \|\beta_1 e_1 - T_k y_k\|_2 . \tag{2.5}$$

The QR factorization of T_k induces

$$Q \begin{bmatrix} T_k & \beta_1 e_1 \end{bmatrix} = \begin{bmatrix} R_k & f_k \\ 0 & \bar{\phi}_{k+1} \end{bmatrix},$$

where the matrix Q is a product $G_{k,k+1} G_{k-1,k} \cdots G_{1,2}$ chosen to eliminate the subdiagonal element $\beta_2, \ldots \beta_{k+1}$ of T_k and

$$R_k = \begin{bmatrix} \rho_1 & \theta_2 & & & \\ & \rho_2 & \theta_3 & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1} & \theta_k \\ & & & & \rho_k \end{bmatrix} \text{ and } f_k = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{k-1} \\ \phi_k \end{bmatrix}.$$

The minimizer y_k of (2.5) can then be obtained from $R_k y_k = f_k$. So an approximate solution is formed as

$$X_k = \mathcal{V}_k * y_k = \mathcal{V}_k * (R_k^{-1} * f_k) = (\mathcal{V}_k * R_k^{-1}) * f_k.$$

Setting $\mathcal{P}_k \equiv \mathcal{V}_k * R_k^{-1} \equiv \begin{bmatrix} P_1 & P_2 & \ldots & P_k \end{bmatrix}$, the approximate solution $X_k = \mathcal{P}_k * f_k$. With the initial guess $P_0 = X_0 = O$, X_k can be obtained by the relation

$$X_k = X_{k-1} + P_k \phi_k.$$

Note that the last block column P_k of $\mathcal{P}_k$ can be updated by

$$P_k = (V_k - P_{k-1}\theta_k)\rho_k^{-1}$$

and

$$f_k = \begin{bmatrix} f_{k-1} \\ \phi_k \end{bmatrix}$$

in which

$$\phi_k = c_k \bar{\phi}_k.$$

So the matrix residual norm $\|R_k\|_F$ is computed directly from

$$\|R_k\|_F = |\bar{\phi}_{k+1}|.$$

The version of the Gl-LSQR algorithm can be stated as follows.

ALGORITHM 1. Gl-LSQR algorithm.

1. Set $X_0 = O_{n\times s}$
2. $\beta_1 = \|B\|_F$, $U_1 = B/\beta_1$, $\alpha_1 = \|H^T U_1\|_F$, $V_1 = H^T U_1/\alpha_1$.
3. Set $W_1 = V_1$, $\bar{\phi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$
4. For k=1, 2, . . .
   i. $\varpi_k = HV_k - \alpha_k U_k$, $\beta_{k+1} = \|\varpi_k\|_F$, $U_{k+1} = \varpi_k/\beta_{k+1}$
   ii. $\tau_k = H^T U_{k+1} - \beta_{k+1} V_k$, $\alpha_{k+1} = \|\tau_k\|_F$, $V_{k+1} = \tau_k/\alpha_{k+1}$
   iii. $\rho_k = (\bar{\rho}^2 + \beta_{k+1}^2)^{1/2}$, $c_k = \bar{\rho}_k/\rho_k$, $s_k = \beta_{k+1}/\rho_k$
   iv. $\theta_{k+1} = s_k\alpha_{k+1}$, $\bar{\rho}_{k+1} = c_k\alpha_{k+1}$
   v. $\phi_k = c_k\bar{\phi}_k$, $\bar{\phi}_{k+1} = -s_k\bar{\phi}_k$
   vi. $X_k = X_{k-1} + (\phi_i/\rho_i)W_i$
   vii. $W_{k+1} = V_{k-1} - (\theta_{i+1}/\rho_i)W_i$
   viii. If $|\bar{\phi}_{k+1}|$ is small enough, then stop

3. Solving the image restoration problem

The Gl-LSQR method as a generalization of LSQR solves the linear systems with multiple right-hand sides simultaneously. This is more effective in the computation process than LSQR handling each right-hand sides sequently. In this section, we investgate the image restoration problem with multiple right-hand sides and suggest Gl-LSQR method as its solver.

The image restoration problem (1.1) has extremely large dimension. If storage becomes an issue, then alternative scheme can be used. One way to reduce memory use is to process the image in blocks. Block processing can produce the same result as processing the image all at once.

Using zero-padding in nonsquare image, an original image can be assumed by N^2 pixels. For an original image, divide it to d small block images, where d is $N^2/n^2 (n << N)$ and number increasing along small block's column-row order. The kth block image be denoted by $IM(k) = \begin{bmatrix}\mathbf{x}_{\cdot 1}^k & \ldots & \mathbf{x}_{\cdot n}^k\end{bmatrix}_{n\times n}$, where $\mathbf{x}_{\cdot j}^k$ means the jth column of kth block image, and X_k a vector produced by the column stacking of $IM(k)$. For each k, $B_{\cdot k}$ is a vector representation of blurred and noisy image corresponding one piece of the original image. Setting $X \equiv \begin{bmatrix}X_1 & X_2 & \cdots & X_d\end{bmatrix}_{n^2\times d}$ and $B \equiv \begin{bmatrix}B_1 & B_2 & \cdots & B_d\end{bmatrix}_{n^2\times d}$, the image restoration problem (1.1) is restated by the following form with d right-hand sides:

$$HX = \bar{B} + R = B, \tag{3.1}$$

where the coefficient matrix $H_{n^2 \times n^2}$ for separable blur is decomposed as a Kronecker product of the horizontal and vertical blurring matrices of appropriate sizes and R denotes the collection of the measurement noise associated with each subimage.

Here, the blurring matrix H has ill-determined rank. So image restoration problem will be extremely sensitive to perturbations in the right-hand side matrix. In order to find a meaningful approximate solution of (3.1), the system has to be replaced by a Tikhonov regularization

$$X_\tau = \arg\min_X \{\|HX - B\|_F^2 + \tau^2 \|LX\|_F^2\}, \tag{3.2}$$

where L and τ are regularization operator and parameter respectively. If the matrix L is taken by the identity matrix, the parameter τ acts on the size of the solution, while if L is taken by a discrete form of first or second derivative, τ acts on the smoothness of the solution.

The best way to solve (3.2) numerically is to treat it as a minimization problem

$$X_\tau = \arg\min_X \left\| \begin{pmatrix} H \\ \tau L \end{pmatrix} X - \begin{pmatrix} B \\ O \end{pmatrix} \right\|_F, \tag{3.3}$$

in certain situations the normal equations

$$(H^T H + \tau^2 L^T L)X = H^T B \tag{3.4}$$

is suited. We try to use Gl-LSQR method as its solver. The basic operation of Gl-LSQR algorithm is matrix-matrix multiplication.

The preconditioned Gl-LSQR algorithm solves (3.3) by transforming the problem with a preconditioner P,

$$\min_Y \left\| \hat{H} P^{-1} Y - \hat{B} \right\|_F \tag{3.5}$$

with $Y = PX$, where $\hat{H} = \begin{pmatrix} H \\ \tau L \end{pmatrix}$ and $\hat{B} = \begin{pmatrix} B \\ O \end{pmatrix}$.

Images are shown only in a finite region. Points near the boundary of a blurred image are affected by information outside the field of view: Zero boundary condition implies the pixels, outside the borders of the image, all zero. Periodic boundary condition implies that the image repeats itself endlessly in all directions. Neumann boundary condition is that the pixels outside image have mirror image values of the scene inside the image borders. Each boundary condition makes the PSF matrix H have different special structure ([4, 5]).

Considering the Neumann boundary condition, coefficient matrix H has a block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTH-HTB) structure which can be diagonalized by two dimensional discrete cosine transform matrix C. The matrix $\hat{H}P^{-1}$ in (3.5) is to be well conditioned when preconditioned by

$$P = C^*\Lambda C = C^*(|\Lambda_C|^2 + \tau^2|\Lambda_L|^2)^{1/2}C,$$

where $H \approx C^*\Lambda_H C$ and $L \approx C^*\Lambda_L C$.

The following Algorithm 2 summarizes our approach.

ALGORITHM 2. Structure of main program.

Input: $N \times N$ degraded image, PSF, n.

1. Set $d = N^2/n^2$.
2. Divide $N \times N$ blurred and noisy image to d block images $\{IM(k)\}_{k=1}^{d}$.
3. Construct the image restoration problem with d right-hand sides by column stacking of each block image $IM(k)$.
4. Compute approximate solution using the (preconditioned) Gl-LSQR algorithm.
5. Reshape each column of approximate solution to $n \times n$ block image.
6. Make a restored image by patching all block images.

4. Numerical experiments and final comments

This section reports the efficiency of Gl-LSQR algorithm for the image restoration problem. All computations were done by Matlab environment.

We apply Gl-LSQR and LSQR method to two practical image restoration problems with multiple right-hand side (3.1) under the Neumann boundary condition : one is *rose* image, and the other is *text* image. Let $t_{Gl-LSQR}$ and t_{LSQR} denote the CPU time obtained when applying Gl-LSQR for multiple right-hand side linear system and LSQR for sequential linear systems with single right-hand side.

To estimate the accuracy of the computed solution X_c with respect to the true solution X of system (3.5), we adopted the relative error $\frac{\|X-X_c\|_F}{\|X\|_F}$.

For a test, we only considered a spatially invariant point spread function whose discrete function was chosen from

$$h_{i-j,\ k-l} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-j)^2+(k-l)^2}{2\sigma^2}}, \quad -r \le i-j,\ k-l \le r, \tag{4.1}$$

where σ is a constant that characterize the blurring. Note that this is called a Gaussian PSF, and can be used to model aberrations in a lens with finite aperture([3]). In (4.1), the variance of the gaussian σ was taken as 0.01 and $r = 6$. Our test used only the identity matrix as a regularization matrix and $\tau = 0.8$ as regularization parameter. To minimize edge effects, each small block image has properly overlapped when the image is divided and patched.

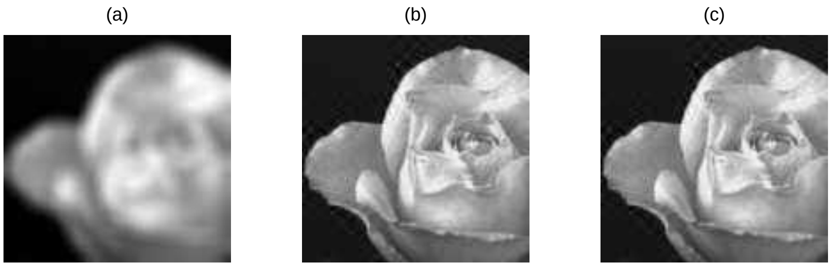

FIGURE 1. Degraded image(a) and restored images by means of Gl-LSQR(b) and LSQR(c).

The size of exact image is 128×128 in the first problem and divide it to 64 small block images. The blurred and noisy image of Figure 1(a) has been built by convolving the PSF with exact small block images and adding the error which was normally distributed with zero mean. The restored images by the preconditioned Gl-LSQR and preconditioned LSQR are shown in Figure 1(b) and Figure 1(c) respectively. The cpu time ratio $t_{LSQR}/t_{Gl-LSQR}$ was about 7. The relative errors of restored images were 0.038 in the preconditioned Gl-LSQR and 0.037 in the preconditioned LSQR (Table 1).

TABLE 1. Comparison the restoring results of the degraded image in Figure 1(a).

	cpu time(sec)	relative error
Gl-LSQR	15.59	0.038
LSQR	109.17	0.037

Second problem has a 256×256 *text* image. The restored images by the Gl-LSQR and LSQR are shown in Figure 2(b) and Figure 2(c) respectively. The cpu time ratio $t_{LSQR}/t_{Gl-LSQR}$ was about 13.7. The relative errors of restored images were 0.39 in the Gl-LSQR and 0.16 in the LSQR (Table 2).

As shown in Table 1-2, the Gl-LSQR algorithm is about $7 \sim 14$ time faster than the LSQR algorithm while both algorithms have not much

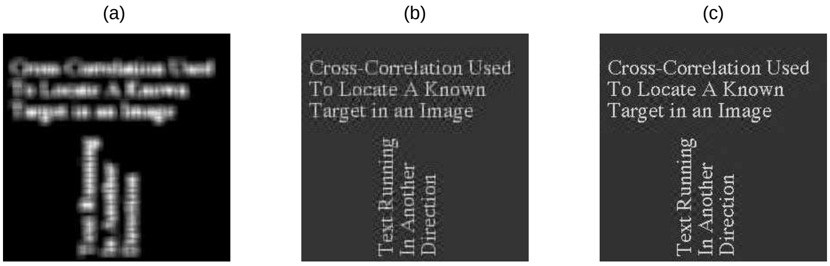

FIGURE 2. Degraded image(a) and restored images by means of Gl-LSQR(b) and LSQR(c).

TABLE 2. Comparison the restoring results of the degraded image in Figure 2(a).

	cpu time(sec)	relative error
Gl-LSQR	25.84	0.39
LSQR	354.51	0.16

relative error difference. Consequently, degraded image can be restored time efficiently using the Gl-LSQR algorithm.

In this work, we have illustrated how Gl-LSQR method can be applied to image restoration problems. Moreover, this method can be applied to mosaicing which has been developed for the detailed reconstruction of large images by successively acquiring image patches in a given row-column or column-row order. We left this research in the future.

References

[1] P. C. Hansen, *Regularization tools 4.0 for Matlab 7.3,* Numerical Algorithms **46** (2007), no. 2, 189-194.

[2] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring images matrices, spectra, and filtering*, SIAM, 2006.

[3] A. K. Jain, *Fundamental of digital image processing*, Prentice-Hall, Engelwood Cliffs, NJ 1989.

[4] K. P. Lee, J. G. Nagy, and L. Perrone, *Iterative methods for image restoration: A Matlab object oriented approach*, Numerical Algorithms, **36** (2004), no. 1, 73-93.

[5] M. K. Ng, R. H. Chan, and W. C. Tang, *A fast algorithm for deblurring models with neumann boundary conditions*, SIAM J. Sci. Comp. **21** (1999), no. 3, 851-866.

[6] S. Y. Oh, S. J. Kwon, and J. H. Yun, *A method for structured linear total least norm on blind deconvolution problem,* Journal of Applied Mathematics and Computing **22** (2006), 373-385.

[7] F. Pitas, *Digital image processing algorithm and applicaitons*, John Wiley & Sons Inc. 2000.

[8] C. M. Thompson and L. Shure, *Image processing toolbox for use with MATLAB*, The MathWorks, Inc. 1993.

[9] F. Toutounian and S. Karimi, *Global least squares method for solving general linear systems with several right-hand sides*, Appl. Math. and Comp. **178** (2006), 452-460.

[10] C. R. Vogel, *Computational methods for inverse problems*, SIAM, 2002.

*

Department of Mathematics
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: sychung@cnu.ac.kr

**

Department of Mathematics
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: soh@cnu.ac.kr

***

Division of Mechanical, Mechanical Design and Mechatronics Engineering
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: sjkw@cnu.ac.kr