JOURNAL OF THE
CHUNGCHEONG MATHEMATICAL SOCIETY
Volume **24**, No. 4, December 2011

TWO DIMENSIONAL VERSION OF LEAST SQUARES METHOD FOR DEBLURRING PROBLEMS

SunJoo Kwon* and SeYoung Oh**

Abstract. A two dimensional version of LSQR iterative algorithm which takes advantages of working solely with the 2-dimensional arrays is developed and applied to the image deblurring problem. The efficiency of the method comparing to the Fourier-based LSQR method and the 2-D version CGLS algorithm methods proposed by Hanson ([4]) is analyzed.

1. Introduction

Image deblurring is to predict the original image from a recorded image corrupted by blurring and noise. The standard linear model for the description of the degraded image is

$$b(i,j) = \sum_{(k,l)\in R_s} h(i-k, j-l)x(k,l) + \eta(i,j), \tag{1.1}$$

where b is the observed (or degraded) image, x is the true image, h is the spatially invariant point spread function(PSF) with region of support R_s, and η represents unknown perturbations in the data like noise. A discrete model of (1.1) is usually formulated by a large-scale linear system

$$b = Hx + \eta \tag{1.2}$$

where the matrix H is defined by PSF. By considering the presence of unknown perturbations in the data, the image deblurring problem can be written as a functional minimization problem

$$\min_x \|Hx - b\|_2 . \tag{1.3}$$

Received October 15, 2011; Accepted November 18, 2011.
2010 Mathematics Subject Classification: Primary 65F22, 65K10.
Key words and phrases: image deblurring, Tikhonov regularization, Kronecker product, LSQR method.
Correspondence should be addressed to SeYoung Oh, soh@cnu.ac.kr.

The above problems are typically ill-posed since the noise in the data may give rise to significant errors in the computation of approximations. Thus a reguarization is needed in order to get stable solutions when the matrix H is severely ill-conditioned. The Tikhonov regularization of (1.3) drives

$$\min_x(\|Hx-b\|_2^2+\lambda^2\|Lx\|_2^2), \tag{1.4}$$

where the regularization operator L is often chosen as the identity matrix or a discretization of a differentiation operator([4, 7]). The regularization parameter λ lies in between the smallest and the largest singular value of H.

Iterative methods such as Landweber and conjugate gradient type approaches (e.g., CGLS, MINRES, LSQR) can be applied for solving (1.4) ([5, 6, 9, 10]). Notice that the LSQR method is a variation of CGLS and these two methods are mathematically equivalent ([1]).

Suppose that the PSF kernel h in (1.1) is separable, that is, the blurring can be separated into the pure horizontal and vertical components corresponding to the horizontal blurring matrix $\bar{A}$ and the vertical blurring matrix A respectively. Then the PSF matrix can be reformulated as

$$H=\bar{A}\otimes A, \tag{1.5}$$

where $\otimes$ denotes the Kronecker product of two square matrices. Now the Tikhonov regularized problem (1.4) can be written by the associated damped least squares problem

$$\min_x\left\|\begin{pmatrix}\bar{A}\otimes A\\ \lambda I\end{pmatrix}x-\begin{pmatrix}b\\ 0\end{pmatrix}\right\|_2^2=\min_x\left\|\hat{H}x-\hat{b}\right\|_2. \tag{1.6}$$

By using the properties of the Kronecker product (1.6) can be converted to the 2-D deconvolution problem which is solvable by the 2-D version of the Landweber and CGLS algorithm methods proposed by Hanson ([4]). The advantage of this method is that we can work solely with the 2-dimensional arrays that represent the underlying 2-D functions.

In this study we derive the 2-D version of LSQR algorithm and apply the algorithm for solving the image deblurring problem. The efficiency of the 2-D version of LSQR, comparing to the Fourier-based version, will be analyzed. Also, the result compared to the 2-D version CGLS algorithm will be investigated.

The brief description of the LSQR method is summarized in Section 2. Section 3 illustrates the two dimensional LSQR method (2-D LSQR) for the image deblurring problem. Numerical experiments and final remarks are described in Section 4.

The product of a block matrix $\mathcal{V}_k = [V_1, V_2, ..., V_k]$ for k matrices of $V_i, i = 1, ..., k$ and k-vector y is defined by $\mathcal{V}_k * y = \sum_{i=1}^{k} y_i V_i$. The notation of $y = vec(Y)$ is a column stacking of matrix Y.

2. LSQR method

This section briefly reviews basic properties of LSQR algorithm in [1] and [11]. The LSQR method is an iterative method for computing a sequence of approximate solutions to the linear least squares problem (1.6).

The bidiagonalization of $\hat{H}$ with starting vector b is initialized with

$$\beta_1 u_1 = b, \ \alpha_1 v_1 = \hat{H}^T u_1 \tag{2.1}$$

and for $i = 1, 2, \ldots$, the method computes

$$\begin{aligned} \beta_{i+1} u_{i+1} &= \hat{H} v_i - \alpha_i u_i, \\ \alpha_{i+1} v_{i+1} &= \hat{H}^T u_{i+1} - \beta_{i+1} v_i, \end{aligned} \tag{2.2}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|u_i\|_2 = \|v_i\|_2 = 1$. After k steps, two orthonormal basis, $\hat{\mathcal{U}}_k \equiv \begin{bmatrix} u_1 & u_2 & \ldots & u_k \end{bmatrix}$ and $\hat{\mathcal{V}}_k \equiv \begin{bmatrix} v_1 & v_2 & \ldots & v_k \end{bmatrix}$, and a lower bidiagonal matrix $\mathsf{T}_k \equiv tridiagonal(\beta_i, \alpha_i, 0)$ are produced. The recurrence formulas (2.1) and (2.2) of the bidiagonalization can be rewritten as

$$\begin{aligned} \hat{\mathcal{U}}_{k+1}(\beta_1 e_1) &= \hat{b}, \\ \hat{H}\hat{\mathcal{V}}_k &= \hat{\mathcal{U}}_{k+1}\mathsf{T}_k, \\ \hat{H}\hat{\mathcal{U}}_{k+1} &= \hat{\mathcal{V}}_k \mathsf{T}_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T, \end{aligned} \tag{2.3}$$

where e_i is the ith column of identity matrix. From (2.3) the LSQR algorithm can find an approximate solution of the form $x_k = \hat{\mathcal{V}}_k y_k$, where y_k is a solution of $\min_x \|\beta_1 e_1 - \mathsf{T}_k y_k\|_2$.

3. Two dimensional version of LSQR

In this section, we propose a two dimensional version of LSQR method (2-D LSQR) which is LSQR method using matrix representations with the property of Kronecker product.

First, consider the following relation of the Kronecker product $\otimes$,

$$(\bar{A} \otimes A)vec(X) = vec(AX\bar{A}^T), \tag{3.1}$$

where $vec(X) = x$ is a column stacking of matrix $X_{n\times n}$ ([2]). Then the 2-D problem of the equation (1.4) is represented by

$$\min_X \{ \left\| AX\bar{A}^T - B \right\|_F^2 + \lambda^2 \left\| X \right\|_F^2 \},$$

and the associated form (1.6) can be converted to

$$\min_X \left\| \begin{pmatrix} AX\bar{A}^T \\ \lambda X \end{pmatrix} - \begin{pmatrix} B \\ O \end{pmatrix} \right\|_F^2 = \min_X \left\| \begin{pmatrix} AX\bar{A}^T - B \\ \lambda X \end{pmatrix} \right\|_F^2, \tag{3.2}$$

where $b = vec(B)$. Set $v_i = vec(V_i)$, $u_i = vec(U_i)$, and $U_i = \begin{pmatrix} U_i^1 \\ U_i^2 \end{pmatrix}$, where $U_i^1 = U_i(1:n,\ :)$ and $U_i^2 = U_i(n+1:2n,\ :)$ are $n \times n$ matrices.

The bidiagonalization of the problem (3.2) similar to (2.1) and (2.2) can be written as

$$\begin{aligned}
&\beta_1 U_1 = \hat{B}, \\
&\alpha_1 V_1 = A^T U_1^1 \bar{A} + \lambda U_1^2 \\
&\text{for } i = 1, 2, \ldots \\
&\qquad \beta_{i+1} U_{i+1} = \begin{bmatrix} AV_i\bar{A}^T \\ \lambda V_i \end{bmatrix} - \alpha_i U_i, \\
&\qquad \alpha_{i+1} V_{i+1} = A^T U_{i+1}^1 \bar{A} + \lambda U_{i+1}^2 - \beta_{i+1} V_i.
\end{aligned} \tag{3.3}$$

The nonnegative scalars α_i and β_i above are the elements of the lower bidiagonal matrix $T_k \equiv tridiag(\beta_i, \alpha_i, 0)$. The bidiagonalization procedure (3.3) constructs two F-orthonormal basis, $\mathcal{U}_k \equiv \begin{bmatrix} U_1 & U_2 & \ldots & U_k \end{bmatrix}$ and $\mathcal{V}_k \equiv \begin{bmatrix} V_1 & V_2 & ... & V_k \end{bmatrix}$ such that $\|U_i\|_F = \|V_i\|_F = 1, i = 1, \cdots, k$.

Now the matrix representations of the recurrence formula (3.3) can be stated as following:

$$\begin{aligned}
&\mathcal{U}_{k+1} * (\beta_1 e_1) = \hat{B}, \\
&\begin{pmatrix} \bar{A} \otimes A \\ \lambda I \end{pmatrix} \hat{\mathcal{V}}_k = \begin{pmatrix} AV_1\bar{A}^T & AV_2\bar{A}^T & \ldots & AV_k\bar{A}^T \\ \lambda V_1 & \lambda V_2 & \ldots & \lambda V_k \end{pmatrix} \\
&\qquad\qquad = \mathcal{U}_{k+1} * \mathsf{T}_k, \\
&\begin{pmatrix} \bar{A} \otimes A \\ \lambda I \end{pmatrix}^T \hat{\mathcal{U}}_{k+1} \\
&\quad = \begin{pmatrix} A^T U_1^1 \bar{A} + \lambda U_1^2 & A^T U_2^1 \bar{A} + \lambda U_2^2 & \ldots & A^T U_k^1 \bar{A} + \lambda U_k^2 \end{pmatrix} \\
&\quad = \mathcal{V}_k * \mathsf{T}_k^T + \alpha_{k+1} v_{k+1} * e_{k+1}^T.
\end{aligned} \tag{3.4}$$

An approximate solution matrix $X_k\,(\in span(\mathcal{V}_k))$ to (3.2) is represented by

$$X_k = \mathcal{V}_k * y_k,$$

where the vector y_k minimizes the F-norm of the corresponding residual

$$\begin{aligned} Res_k &= \hat{B} - \mathcal{U}_{k+1} * \mathsf{T}_k y_k \\ &= \beta_1 U^{(1)} - (\mathcal{U}_{k+1} * \mathsf{T}_k) y_k \\ &= \beta_1 U^{(1)} - \mathcal{U}_{k+1} * (\mathsf{T}_k y_k) \\ &= \mathcal{U}_{k+1} * (\beta_1 e_1 - \mathsf{T}_k y_k). \end{aligned}$$

Consequently, the problem is reduced to the least squares problem:

$$\min \|Res_k\|_F = \min_{y_k} \|\beta_1 e_1 - \mathsf{T}_k y_k\|_2 .$$

Constructing the orthogonal matrix Q to eliminate the subdiagonal elements $\beta_2, \ldots \beta_{k+1}$ of T_k, we obtain the QR decomposition of T_k :

$$Q \begin{bmatrix} \mathsf{T}_k & \beta_1 e_1 \end{bmatrix} = \begin{bmatrix} \mathfrak{R} & g_k \\ 0 & \bar{\psi}_{k+1} \end{bmatrix},$$

where the matrix $\mathfrak{R} = upperbidiagonal(\rho_k, \theta_{k+1})$ and k-vector $g_k = [\psi_i]_{i=1}^k$. The approximate solution matrix to (3.2) is formed as in the following theorem.

THEOREM 3.1. *The iterates computed by 2-D LSQR are represented by recurrence formulas*

$$X_j = X_{j-1} + P_j \psi_j, \quad j = 1, \ldots, k$$

where the j-th block matrix P_j of $\mathcal{P}_k = \mathcal{V}_k * \mathfrak{R}^{-1}$ is obtained from $P_j = (V_j - \theta_j P_{j-1})\rho_j^{-1}$.

Proof. Updating the approximate solution matrix X_j ;

$$X_j = \mathcal{V}_j * y_j = \mathcal{V}_j * (\mathfrak{R}^{-1} g_j) = (\mathcal{V}_j * \mathfrak{R}^{-1}) * g_j = \mathcal{P}_j * g_j,$$

we get

$$\begin{aligned} X_j = \mathcal{P}_j * g_j &= \begin{bmatrix} \mathcal{P}_{j-1} & P_j \end{bmatrix} * \begin{bmatrix} g_{j-1} \\ \psi_j \end{bmatrix} \\ &= \mathcal{P}_{j-1} g_{j-1} + P_j \psi_j \\ &= X_{j-1} + P_j \psi_j, \quad j = 1, \ldots, k. \end{aligned}$$

The j-th column of the matrix $\mathfrak{R}^{-1}$ is

$$\begin{bmatrix} (-1)^{j-1} \frac{\theta_2 \theta_3 \cdots \theta_j}{\rho_1 \cdots \rho_{j-1} \rho_j} & (-1)^{j-2} \frac{\theta_3 \theta_4 \cdots \theta_j}{\rho_2 \cdots \rho_{j-1} \rho_j} & \ldots & -\frac{\theta_j}{\rho_{j-1} \rho_j} & \frac{1}{\rho_j} & 0 & \ldots & 0 \end{bmatrix}^T .$$

The j-th element P_j of $\mathcal{P}_k \equiv \mathcal{V}_k * \mathfrak{R}^{-1}$ is

$$\begin{aligned}
P_j &= (-1)^{j-1}\frac{\theta_2\theta_3\cdots\theta_j}{\rho_1\cdots\rho_{j-1}\rho_j}V_1 + (-1)^{j-2}\frac{\theta_3\theta_4\cdots\theta_j}{\rho_2\cdots\rho_{j-1}\rho_j}V_2 + \cdots \\
&\qquad\qquad \cdots - \frac{\theta_j}{\rho_{j-1}\rho_j}V_{j-1} + \frac{1}{\rho_j}V_j \\
&= \frac{1}{\rho_j}\left(V_j - \frac{\theta_j}{\rho_{j-1}}V_{j-1} + \right. \\
&\qquad \left. \cdots + (-1)^{j-2}\frac{\theta_3\theta_4\cdots\theta_j}{\rho_2\cdots\rho_{j-1}}V_2 + (-1)^{j-1}\frac{\theta_2\theta_3\cdots\theta_j}{\rho_1\cdots\rho_{j-1}}V_1\right) \\
&= \frac{1}{\rho_j}\left(V_j - \frac{\theta_j}{\rho_{j-1}}\left(V_{j-1} - \frac{\theta_{j-1}}{\rho_{j-2}}\left(\cdots - \frac{\theta_3}{\rho_2}\left(V_2 - \frac{\theta_2}{\rho_1}V_1\right)\right)\right)\right).
\end{aligned}$$

Since $P_1 = V_1{\rho_1}^{-1}$, $P_2 = (V_2 - \theta_2 P_1)\,{\rho_1}^{-1}$, and so on, we can obtain $P_j = (V_j - \theta_j P_{j-1})\,{\rho_j}^{-1}$. □

Note that the matrix residual norm $\|Res_k\|_F$ is given by

$$\|Res_k\|_F = |\bar{\psi}_{k+1}|.$$

Based on the above discussion, 2-D version of the LSQR algorithm can be stated as follows.

ALGORITHM 1. **2-D LSQR algorithm.**

1. Input X_0, B, and λ
2. Set $\hat{B} = [B; O]$
3. $\beta_1 = \left\|\hat{B}\right\|_F$, $U_1 = \hat{B}/\beta_1$, $\alpha_1 = \left\|A^T U_1 \bar{A}\right\|_F$, $V_1 = A^T U_1 \bar{A}/\alpha_1$
4. Set $W_1 = V_1$, $\bar{\psi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$
5. **For k=1, 2, . . .**
   i. $\varpi_k = \begin{bmatrix} AV_k\bar{A}^T \\ \lambda V_k \end{bmatrix} - \alpha_k U_k$
   ii. $\beta_{k+1} = \|\varpi_k\|_F$
   iii. $U_{k+1} = \varpi_k/\beta_{k+1}$
   iv. $\tau_k = A^T U_{i+1}(1:n,:)\bar{A} + \lambda U_{i+1}(n+1:2n,:) - \beta_{k+1}V_k$
   v. $\alpha_{k+1} = \|\tau_k\|_F$
   vi. $V_{k+1} = \tau_k/\alpha_{k+1}$
   vii. $\rho_k = (\bar{\rho}^2 + \beta_{k+1}^2)^{1/2}$
   viii. $c_k = \bar{\rho}_k/\rho_k$, $s_k = \beta_{k+1}/\rho_k$
   ix. $\theta_{k+1} = s_k\alpha_{k+1}$, $\bar{\rho}_{k+1} = c_k\alpha_{k+1}$
   x. $\psi_k = c_k\bar{\psi}_k$, $\bar{\psi}_{k+1} = -s_k\bar{\psi}_k$
   xi. $X_k = X_{k-1} + (\psi_i/\rho_i)W_i$
   xii. $W_{k+1} = V_{k-1} - (\theta_{i+1}/\rho_i)W_i$
   xiii. If $|\bar{\psi}_{k+1}|$ is small enough, then stop

4. Experimental results

The performance of the 2-D version of LSQR algorithm for the image deblurring problem is presented in this section. For the purpose of the comparison, 2-D CGLS method and 2-D LSQR method are implemented in Matlab and applied to the practical image deblurring problems (3.2) for the sample image in Figure 1(a). The 2-D LSQR method was also compared to the Fourier-based algorithm for LSQR method.

Under the zero boundary condition, both of the matrices $\bar{A}$ and A in (1.5) have the square and Toeplitz structures. Thus H becomes a block Toeplitz with Toeplitz blocks(BTTB) matrix. It can be determined by its first row and first column elements. By extending the BTTB into a block circulant with circulant blocks matrix(BCCB), we can use the two dimensional discrete Fourier transform to compute the matrix-vector multiplications ([12]).

For our test, we only considered atmospheric turbulence blur with a spatially invariant point spread function whose discrete function is induced by $h(i-j,\ k-l) = \frac{1}{2\pi\sigma\bar{\sigma}} \exp\left(-\frac{1}{2}\left(\frac{i-j}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{k-l}{\bar{\sigma}}\right)^2\right)$ where $-r \leq i-j,\ k-l \leq r$. This is known as a Gaussian PSF in the image processing community and can be used to model aberrations in a lens with finite aperture([2, 8]).

In computations, the parameter $\sigma = \bar{\sigma} = 7$ and $r = 16$ was fixed in the atmospheric turbulence. The *blur* function of the Regularization tool([6]) was used in the tests of the methods. We set up the regularization parameter λ as 0.01 and the stopping criteria with relative error $< 5.0 \times 10^{-2}$.

Figure 1 shows the results for a 512×512 true image ([3]) in Figure 1(a) corrupted with an additive Gaussian normal distributed. The blurred and noisy image is shown in Figure 1(b). The reconstructed image by the 2-D version LSQR method is given in Figure 1(c). Figure 1(d) shows the behavior of the PSNR as the iteration proceeds.

In Table 1, it is shown that the 2-D version of LSQR method for size of 512×512 image was 3.9 times faster than Fourier-based LSQR method to approach 30.03 for PSNR (peak signal-to-noise ratio). The CPU time needed for the 2-D version of CGLS algorithm to get the same accuracy was 59.87 which is slower by 0.81 than 2-D LSQR. Based on our computational results of the algorithms for many sample images, it is experimentally testified that the two dimensional version of LSQR

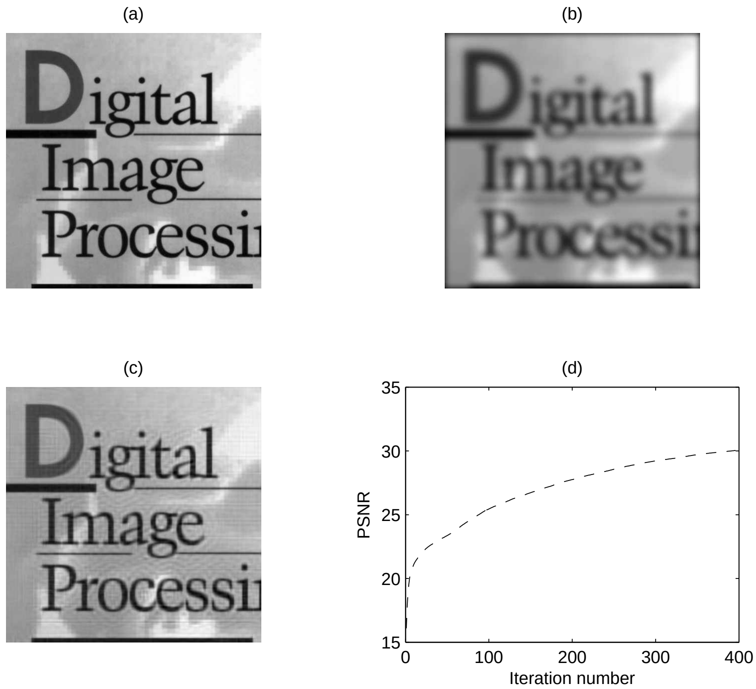

FIGURE 1. (a) Exact image, (b) blurred and noisy image, (c) restored image by the 2-D LSQR method with relative error $= 5.0 \times 10^{-2}$ and PSNR $= 30.03$, and (d) plot of the PSNR versus iteration numbers.

iterative algorithm working solely with the 2-dimensional arrays is more efficient than the 2-D CGLS method and Fourier-based LSQR algorithm.

TABLE 1. Comparison of the CPU time for (I) 2-D version of LSQR, (II) Fourier-based LSQR, and (III) 2-D version of CGLS (relative error is about 5.0×10^{-2}).

Image size	PSNR	Iter. num.	I	II	III
256×256	29.8	842	17.33	57.52	20.13
	31.3	910	19.60	61.40	22.27
512×512	30.0	396	49.47	193.22	59.87

The implementation of the image deblurring problem is a memory intensive application with insurmountable data. An adaption technique of LSQR using the properties of the Kronecker product and solving the

2-D deconvolution problem with the 2-dimensional arrays can reduce the amount of works and speed up to get approximate solutions of least squares problems. It can be a computationally effective way to many applications as well as the image deblurring problems.

Our future work is to develop 2-D version techniques of the global iterative methods such as the Gl-CGLS algorithm and the Gl-LSQR algorithms and investigate their efficiencies.

References

[1] Å. Björk, *Numerical methods for least squares problems*, SIAM, 1996.
[2] H. C. Andrews, B. R. Hunt, *Digital image restoration*, Prentice-Hall Inc, 1977.
[3] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Prentice Hall, 2002.
[4] P. C. Hansen, *Deconvolution and regularization with Toeplitz matrices*, Numerical Algorithm, **29** (2002), 323-328
[5] P. C. Hansen, J. G. Nagy, and D. P. O' Leary, *Deblurring images matrices, spectra, and filtering*, SIAM, 2006.
[6] P. C. Hansen, *Regularization tools 4.0 for Matlab 7.3,* Numerical Algorithms **46** (2007), no. 2, 189-194.
[7] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, 2010.
[8] A. K. Jain, *Fundamental of digital image processing*, Prentice-Hall, Engelwood Cliffs, NJ, 1989.
[9] R. L. Lagendijk, J. Biemond, *Iterative identification and restoration of images*, Kluwer, 1991.
[10] 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.
[11] C. C. Paige, M. A Saunders, *LSQR: An algorithm for sparse Linear equations and sparse least squares,* ACM Trans. on Math. Soft. **8** (1982), no. 1, 43-71.
[12] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, 2002.

*
Innovation Center of Engineering Education
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: sjkw@cnu.ac.kr

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