An improved Tikhonov regularization method based on defect principle for image deblurring

M. Zarebnia[a]*, Mahsa.dejamkhooy[a]†, R. Parvaz[a]‡

[a]Department of Mathematics, University of Mohaghegh Ardabili, 56199-11367
Ardabil, Iran.

Abstract

In this paper, an approach for deblurring of noise-corrupted and blurred images has been proposed. In addition to Thikhonov regularization for robustness of restoration process and increasing the image quality ”Defect principle” is utilized. In proposed method the blurred image is divided into separated layers. These layers are defined by imposing a specific restrictions on blurred image pixels. Each layer is regularized with Tikhonov method individually, so that layers which contain useful properties will be regularized with higher accuracy. The defect principle not only increases our ability to concentrate on layers with important content but the quality of the blurred image also increases sharply. Efficiency and theorical contents of the method are evaluated and shown, moreover applications to image deblurring are also presented.

Keywords: Tikhonov regularization method; Defect principle; Image deblurring; Anti-reflective boundary.

1 Introduction

Image deblurring is an effective process for restoration of blurred and noise-corrupted images. Blurring occurs due to some mechanical or physical reasons. for instance, cameras optical system and defocusing, motions, atmospheric turbulences which happen in astronomical image, etc [1, 2, 3]. In addition to the mentioned blurring factors, more often images contain some random noise that along with blurring factors, create a blurry and noise-corrupted images.

*Corresponding author: zarebnia@uma.ac.ir

†Mahsa.Dejam@student.uma.ac.ir

‡rparvaz@uma.ac.ir

Deblurring problem can be described by mathematical models. These models have been used for restoration of contaminated and noise-corrupted images. In two-dimensional space, this can be shown as an integral equation of form:

$$g = \int_{R^2} h(z-y) f(y) dy + \nu(z), \quad z \in \Omega \subset R^2, \tag{1.1}$$

where f represents the true image, g the blurry and noisy image, h the space invariant point spread function(PSF) and ν is a random noise. In discrete form, equation (1.1) can be written as follows

$$g_i = \sum_{j \in z^2} h_{i-j} f_j + \nu_i, \tag{1.2}$$

where f_j and g_i represents the light intensity at each pixel of blurred image and the true image, respectively and $i, j \in Z^2$. The linear system of (1.2) is underdetermined, so a meaningful solution of this system is hard to find. By imposing boundary condition on image borders, the pixels out of the field of a view (FOV) assumed to be a combination of pixels inside, so that equation (1.2) can be demonstrate as linear equation system

$$g = Af + \eta, \tag{1.3}$$

where 'f' is a true image, 'g' is blurred image and 'A' models a blurring process [4]. The linear system (1.3) is an ill-posed system with high condition number therefore the solution of (1.3) cannot be computed in conventional methods, so iterative methods based on Krylove subspace with Arnoldi method or generalized minimal residual (GMRES) method like Tikhonov method or TV method has been used [24, 25, 26]. The first attempt in deblurring is obtaining an accurate blurring operator matrix 'A'. In particular, the quality of deblured image depends on deblurring matrix structure and how to model it. Modeling of matrix 'A' is influenced by imposed boundary conditions. The type of imposed boundary conditions have a critical impact on [5]:

1. Reconstructed image quality and its ability to reduce the artifacts (like ringing effects) close to the boundaries.

2. Cost of computation during the 'true' image recovery with or without the noise.

Popular boundary conditions consist of zero boundary condition (ZBC) [8], periodic boundary condition (PBC)[6], reflective boundary condition (RBC) and anti-reflective boundary condition (ARBC) (See also [1, 5, 9, 10] for more details). ARBC is defined by assuming that the pixels outside a FOV are anti-reflection of the pixels inside. Hence, the blurring matrix 'A' in (1.3) has a special structure as :

A=two-level Toeplitz + two-level Hankel + rank two matrix in 2D case.

Kronocker product approximation of 'A' is efficient for constructing and using in computations. By using this approximate structure for 'A' SVD-based solution (1.3) for image of size $n \times n$, inattentive of symmetry conditions of the psf, has in at most $O(n^3)$ arithmetic operations. It should be noted that without this approximation, the solution with explicit matrix 'A' required up to $O(n^6)$ arithmetic operation [27].

In addition to anti-reflective BC and Kronocker product approximation in constructing matrix 'A', we utulized another impressive replacement which replaced system (1.3) with

$$A'Af = A'g. \tag{1.4}$$

Discretization of system (1.1) with the same boundary condition and 180° rotated psf gives rise to matrix 'A'' . The system (1.4) will be equivalented to

$$A^T Af = A^T g. \tag{1.5}$$

The mentioned substitution is used when the assessment of matrix-vector products with the transpose 'A^T' of the matrix 'A' is complicated or time consuming. It can occur when the structure of 'A^T' is more difficult to exploit and in spite of that, the matrix 'A' is not expressly stored and has a structure that can be used when evaluating matrix-vector products efficiently. The solution of (1.4) gives the best solution when the psf is symmetric so that $A' = A$. For instance, this is remarkable when 'A' is constructed by a convolution in two-dimensions by imposing an ARBC. According to the results in next section, solution of (1.4) reduces the artifacts. For further explanation and details see also [11, 12, 13, 23]. The rest of the paper is organized as follows : Section 2 deals with description of Tikhonov regularization method, section 3 presents the proposed algorithms. In section 4 the numerical results are shown and efficiency of proposed method is evaluated.

2 Tikhonov regularization method

In this section, a method for preparing an approximate solution to a linear system of equation

$$Ax = b, \tag{2.1}$$

have been considered; where $A \in R^{m \times n}$, $b \in R^m$ and $x \in R^n$ and $m \geq n$. The matrix 'A' is a rank deficient and also its singular values of 'A' decay and cluster gently to zero. Besides, the matrix 'A' is more often assumed to be too large so that regularization techniques and iterative methods are generally employed for solving this kind of problem. These systems typically arises from the discretization of ill-posed problems such as Fredholm integral equation of the first kind with smooth kernel [14, 15]. On the other hand, if the system(2.1) related to image restoration, the element of 'b' is contaminated by some noise. In this case, 'b' has been assumed as:

$$b = \widehat{b} + e_b, \tag{2.2}$$

where e_b is random noise and $\widehat{b}$ is noise free right-hand side. Due to the presence of random noise and ill-conditioning of 'A' , the exact solution of (2.1) is not expected to be a noise-free solution. Therefore, it is essential to compute regularized solution to filter out the effect of such noise. An appropriate statement of the problem (2.1) in the case of noise-corrupted right-hand side, is the least square problem of form

$$\min_x \parallel Ax - b \parallel_2 . \tag{2.3}$$

A popular and well-known regularization method is the Tikhonov regularization method, which replaces the problem (2.3) to minimization problem of

$$\min_x \parallel Ax - b \parallel_2^2 + \lambda^2 \parallel Lx \parallel_2^2, \tag{2.4}$$

where L is a regularization operator that used to obtain a solution with eligible properties. The matrix 'L' could be the identity matrix or could represent the first or second derivates. The parameter $\lambda > 0$ is known as a regularization parameter which has influence on the size of solution. While λ plays an effective role on the smoothness of solution, the most approved techniques for computing the regularization parameter are :

1. L - Curve method [16, 17, 18, 20] ;
2. GCV method [15, 28];

The L-Curve method is based on a plot of the curves ($\parallel Ax_\lambda - b \parallel_2$, $\parallel Lx_\lambda \parallel_2$) instead of residual norm $\parallel b - Ax \parallel_2$. An appropriate parameter is lied on the corner of L-Curves. Therefore , the L-Curve method chooses the regularization parameter which consist of the point on the curve with maximum curvant[16]. in [17, 18] Per Chiristian Hansen has been adressed a solution of ill-posed problems based on L-Curve method. This recent study has been considered on GCV(Generalized cross validation) method which dose not require the details about noise. In this method, the regularization parameter is choosen by minimizing the GCV function below :

$$G(\lambda) = \frac{\left\|b - Ax_\lambda\right\|^2}{\left[tr\,(I - AA_\lambda)\right]^2}, \tag{2.5}$$

where $A_\lambda = (A^TA + \lambda^2L^2L^TL)A^T$, and x_λ is the solution of (2.4). Using the generalized singular value decomposition (GSVD) of the matrix pairs (A,L) with $A \in R^{m\times n}$, $L \in R^{p\times n}$; and letting $A = USx^{-1}$ and $L = VCx^{-1}$ where $S = diag(s_1, \cdots, s_p)$ and $C = diag(c_1, \cdots, c_p)$. therefore, the ratio $\gamma_i = \frac{S_i}{C_i}, i = 1, \cdots, p$ will define the generelized singular value of pair b(A,L) so that the expression of the GCV can be shown as :

$$G(\lambda) = \frac{\sum\limits_{i=1}^{N} \left(\frac{\lambda^2}{{\gamma_i}^2+\lambda^2} {u_i}^T b\right)^2}{(M - (N-p) - \sum\limits_{i=1}^{p} \frac{{\gamma_i}^2}{{\gamma_i}^2+\lambda^2})^2}, \tag{2.6}$$

for special case of $M = N$ and $P \leqq N$, the equation (2.6) is substituted by:

$$G(\lambda) = \frac{\sum_{i=1}^{N} \left(\frac{\lambda^2}{{\gamma_i}^2+\lambda^2} {U_i}^T b \right)^2}{\left(\sum_{i=1}^{p} \frac{\lambda^2}{{\gamma_i}^2+\lambda^2} \right)^2}. \tag{2.7}$$

Hence, the GCV method is based on choosing an appropriate ‘λ‘ which minimize the $G(\lambda)$.

In Tikhonov regularization method, solution of (2.4) is specifically computed as a linear least square problem of form :

$$\hat{X} = \arg\min \left\| \begin{bmatrix} H \\ \lambda L \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|_2^2 . \tag{2.8}$$

Tikhonov method is expressed to be in standard form, when L is the identity matrix, so the solution x_λ of (2.4) solves the problem

$$(A^T A + \lambda L^T L)x = A^T b. \tag{2.9}$$

The problem (2.4) is too large to be solved in implicit methods. Hence, it is appropriate to project the large-scale problems into Krylove subspace and use regularization in smaller dimension. As an iterative solution of (2.4), based on Krylov subspace, Arnoldi-Tikhonov method is presented in next section.

2.1 Arnoldi-Tikhonov method

Arnoldi-Tikhonov method has been considered as an iterative method to solve the minimization problem(2.4) into the smaller dimension. The main aim of this structure is to project the matrix ‘A‘ on to krylov subspace, produced by ‘A‘ and the vector ‘b‘ which $K_m(A, b) = span\{b, Ab, ..., A^{m-1}b\}$ with $m \leq n$. The Arnoldi-Tikhonov method utilizes the Arnoldi algorithm for construction of the Krylov subspace; consequently, it can be written:

$$AV_m = V_m H_m + h_{m+1} V_{m+1} e_m^T, \tag{2.10}$$

where $V_m = \{V_1, \cdots, V_m\} \in R^{n \times m}$ contains an orthonormal columns which covers the basis of Krylov subspace $K_m(A, b)$ defining by $V_1 = b/ \parallel b \parallel$. The matrix $H_m(h_{i-j}) \in R^{m \times m}$ is an upper Hessenberg matrix which represents the orthognal projection of ‘A‘ on to $K_m(A, b)$ and $e_m(0, \cdots, 0, 1)^T$. The equation (2.10) is equivalent to

$$AV_m = V_{m+1} \bar{H}_m, \tag{2.11}$$

which

$$\bar{H_m} = \left[\begin{array}{c} H_m \\ h_{m+1,m} {e_m}^T \end{array} \right] \in R^{(m+1) \times m}. \tag{2.12}$$

By considering the minimization problem (2.4) and substitution of $x = V_m y_m + x_0$ (x_0 is an approximate solution eventually is 0 if not available) where $y \in R^m$ and $r_0 = b - Ax_0$, we yeild the reduced minimization :

$$min\{\parallel \bar{H_m} y_m - \parallel b \parallel e_1 \parallel^2 + \lambda^2 \parallel LV_m y_m \parallel\}. \tag{2.13}$$

The minimizatin problem (2.13) is known as an Arnoldi- Tikhonov method. This method can be interpreted as a linear regularizatiom of the GMRES with $L = I_N$ [16, 19]. Using the suitable regularization operator $L \neq I_N$, should significantly enhance the quality of the approximate solution in comparison to choose of $L = I_N$. Generally, by choosing the $L \in Rp \times n$, it has been observed that the minimization (2.13) is equivalent to

$$\min_{y_m \in R^m} \left\| \left(\begin{array}{c} \bar{H}_m \\ \lambda L V_m \end{array} \right) y_m - \left(\begin{array}{c} r_0 e_1 \\ 0 \end{array} \right) \right\|^2. \tag{2.14}$$

Thus, with refrence to Arnoldi-Tikhonov method, the least square problem (2.14) has matrix coefficient of dimension $(m + 1 + p) \times m$, if $L \in R^{P \times N}$, instead of $(2m + 1) \times m$ in Arnoldi- Tikhonov method where in (2.14) LV_M is replaced by I_M. To simplify computing and controling the confusion with the standard Arnoldi-Tikhonov method, the reduced minimization (2.14) has been defined by Modified Arnoldi-Tikhonov method. This algorithm is represented in algorithm (1).

The minimization problem (2.14) has been used in image deblurring with specific algorithm which is described in algorithm (2),[21, 22]. In the following

Algorithm 1: Modified global Arnoldi algorithm.

```
Input  : $V, k$,
Output: $\mathcal{V}_k$, $\bar{H}_k$,
$V_1 = V/ \parallel V \parallel_F$,
for $j = 1, \cdots, K$ do
    $\tilde{V} = \mathcal{A}(V_j)$,
    for $i$= $1,...,j$ do
        $h_{i,j} = \langle V_i, \tilde{V} \rangle_F$,
        $\tilde{V} = \tilde{V} - h_{i,j} V_i$,
    end
    $h_{j+1,j} = \parallel \tilde{V} \parallel_F$,
    $V_{j+1} = \tilde{V} / h_{j+1,j}$,
end
$\mathcal{V}_k := [V_1, \ldots, V_k]$.
```

algorithm the parameter $A(X)$ is defined as follows:

$$A(X) = H_2 X H_1^T.$$

Algorithm 2: The global-GMRES method algorithm for image deblurring

```
Input  : $X_b$ =Blurred Image, $k, n$,
Output: $X_d$ =Deblurred Image,
$X_0 = X_b$,
for $j = 1, \ldots, n$ do
    $V = B - \mathcal{A}(X_0)$,
    Find $\mathcal{V}_k$ and $\bar{H}_k$ by using Algorithm 1 for $V$ and $k$,
    Determine $\lambda$ by using (2.7),
    Find $y_k$ as solution of the least squares problem (2.14),
    $X_0 = X_0 + V_k(y_k \otimes I_n)$,
end
$X_d := X_0$.
```

3 An improved algorithm for Arnoldi- Tikhonov method

In this section, an algorithm for improving of the Arnoldi- Tikhonov method by means of ”Defect principle” is proposed.

3.1 Proposed algorithm

Let A and A' in (1.4) and (1.5) defined as a kronecker product of form

$$A = H_2 \otimes H_1,$$

and

$$A' = H_2' \otimes H_1',$$

by using the kronecher product properties, the equations (1.4) and (1.5) can be written as

$$\mathcal{H}_1 X \mathcal{H}_2^T = {H_1}' B {H_2}'^T, \tag{3.1}$$

where $\mathcal{H}_1 = H_1' H_1$, $\mathcal{H}_2 = H_2' H_2$, $b = vec(B)$ and $x = vec(X)$ defined in (2.1), [21, 22]. In the proposed algorithm, as a first step, the blurred image ‘B‘ is divided into two images are named B_1 and B_2

$$B = B_1 + B_2,$$

It is important to be careful in choosing suitable techniques to determine the separated images B_1 and B_2. Here two different techniques are considered for this purpose. The first one is the application of symmetric and non-symmetric matrices. In this case, we consider

$$B_1 = \frac{1}{2}(B + B^T), \quad B_2 = \frac{1}{2}(B - B^T). \tag{3.2}$$

The second technique is based on considering pixels on the image B. An arbitrary integer value t in the range $[0, 255]$ is considered then the pixels with values less than or equal to t are named as B_1 and the remained pixels which are valued more than t are named as B_2. By considering such partitioning, the equation (3.1) can be written as

$$\mathcal{H}_1 X \mathcal{H}_2^T = H_1' B_1 (H_2')^T + H_1' B_2 ({H_2}')^T. \tag{3.3}$$

Now let X_1 and X_2 be the solution of equations

$$\mathcal{H}_1 X_1 \mathcal{H}_2^T = H_1' B_1 (H_2')^T, \tag{3.4}$$

and

$$\mathcal{H}_1 X_2 \mathcal{H}_2^T = H_2' B_1 ({H_2}')^T, \tag{3.5}$$

respectively. By Considering equations (3.3) and (3.4), the solution of (3.1) gives

$$X = X_1 + X_2. \tag{3.6}$$

The proposed algorithm is presented in Algorithm 3. The proposed algorithm can be executed using parallel computing. In the following algorithm,$A(v) = mathcal H_1 v_1 \mathcal{H}_2^T$ also X_b and X_d are considered as blurred and deblurred images, respectively.

```
Algorithm 3: Proposed algorithm
Input   : $X_b$ =Blurred Image, $k_1, k_2, n_1, n_2$,
Output: $X_d$ =Deblurred Image,
Divide $X_b$ as $X_{b1}$ and $X_{b2}$,
Determine $X_{d1}$ and $X_{d2}$ by using Algorithm 1 for $X_{b1}$, $X_{b2}$ and $A(v)$ as (25),
Consider $X_d = X_{d1} + X_{d2}$.
```

3.2 An improved algorithm based on defect principle

Let X_1^K is the solution of equation (3.4) at step k, the defect at step k is computed by

$$\mathcal{D}_1^k := {H_1}' B_1 ({H_2}')^T - \mathcal{H}_1 X_1^k \mathcal{H}_2^T, \tag{3.7}$$

By substituting of definition in equattion (3.3), we can find:

$$\mathcal{H}_1 X_2^k \mathcal{H}_2^T = H_1' B_2 (H_2')^T + \mathcal{D}_1^k. \tag{3.8}$$

We get X_2^k by solving the equation(3.8) and then at step k we obtain

$$X^k = {X_1}^k + {X_2}^k.$$

In the next step, the defect for (3.5) is defined by

$$\mathcal{D}_2^k := H_1' B_2 (H_2')^T - \mathcal{H}_1 X_2^k (\mathcal{H}_2)^T. \tag{3.9}$$

At step $k+1$, the equation (3.7) is subtitude by:

$$\mathcal{H}_1' X_1^{k+1} (\mathcal{H}_2')^T = \mathcal{H}_1' B_1 (\mathcal{H}_2')^T + \mathcal{D}_2^k. \tag{3.10}$$

The itteration of this algorithm yeilds to restored image. The proposed algorithm is summerized as Algorithm (4). In Section 4 the performace of proposed algorithm is tested and results are presented.

```
Algorithm 4: Improved algorithm based on defect principle
Input  : $X_b$ =Blurred Image, $k_1, k_2, n_1, n_2, \tau$,
Output: $X_d$ =Deblurred Image,
Divide $X_b$ as $X_{b1}$ and $X_{b2}$,
$\mathcal{D}_2 = 0$,
for $l = 1 : \tau$ do
    $X_{b1} := X_{b1} + D_2$,
    Find $X_{d1}$ by using Algorithm 2 with $k_1, n_1$,
    $\mathcal{D}_1 := X_{b1} - \mathcal{H}_1 X_{d1} (\mathcal{H}_2)^T$,
    Find $X_{d2}$ by using Algorithm 2 with $k_1, n_1$,
    $\mathcal{D}_2 := X_{b2} - \mathcal{H}_1 X_{d2} (\mathcal{H}_2)^T$,
end
Consider $X_d = X_{d1} + X_{d2}$.
```

4 Simulation results

In this section, the efficiency of the proposed algorithms have been tested on three distinct examples. The algorithms performance has been evaluated in terms of speed and accuracy. To assess the restored images quality, relative error and the peak signal-to-noise ratio (PSNR) have been utilized as follows

$$Relative\ error = \frac{||X_e - X_d||_F}{||X_e||_F},$$

$$PSNR = 10 \log_{10} \frac{n^2 d^2}{||X_e - X_d||_F^2},$$

where $d = 255$, X_e and X_d denote the exact (blur- and noise free) image and restored image, $A = H_1 \otimes H_2$ and $A' = H_1' \otimes H_2'$ are considered by using algorithm in [27]. The numerical results are obtained by executing of algorithms on MATLAB 7.11.0 .

Example 1. As a first example, the "Cameraman (256 × 256)" image is considered as a test image. In Figures 1-3, Tables 1-2, the point spread function is assumed to be a linear motion of a camera by 5 pixels along 180° direction. while, in fig. 4 the blurring process is simulated by movment of camera for 11 pixels along the 180° . For algorithms (2), (3) and (4) the values n (refers to itteration of Algorithm (1)) and k (refers to itteration of algorithm (2)) and corresponding run time, error, relative errors and PSNRs are given in Tables 1 and 2. In Table 1, for Algorithm (3), $n_1 = n_2 = n, k_1 = k_2 = k$ are considered. In Table 2 for Algorithm (2), $n = 70, k = 50$ and for Algorithm 4, $n_1 = 20, n_2 = 40, k_1 = 8, k_2 = 10, \tau = 10$ are used. Table 1 shows that Algorithm 3 is more accurate than Algorithm (2). Also Table 2 shows that at smaller run time the Algorithm (4) provided more accurate results than Algorithm (2). In this experiment for Algorithm (4), image is divided by using (3.2).

Table 1: Comparison of Relative errors and PSNR-values for Example 1.

	Algorithm 2		Algorithm 3	
	Relative error	PSNR	Relative error	PSNR
n=90, k=50	0.0890	26.8371	0.0801	27.7456
n=80, k=30	0.0898	26.7580	0.0809	27.6678
n=60, k=20	0.0922	26.5226	0.0829	27.4458

Table 2: Comparison of Algorithm (2) and Algorithm (4) for Example 1.

	Time	Error	PSNR
Algorithm 2	1.798s	0.0909	26.6523
Algorithm 4	1.609s	0.0840	27.3320

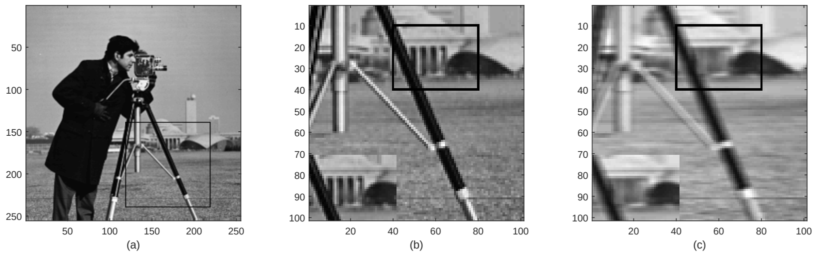


Figure 1: (a) - (b) Blur and noise free image, (c) Blurred image.

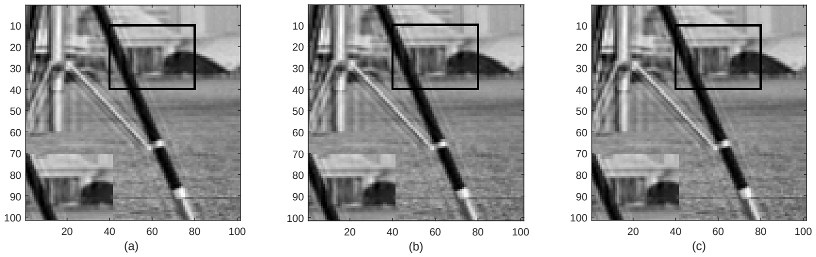


Figure 2: Restored images by using Algorithm (2): (a) $n = 60, k = 20$, (b) $n = 80, k = 30$, (c) $n = 90, k = 50$.

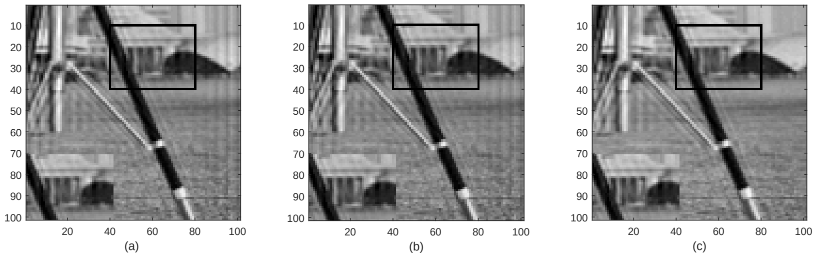
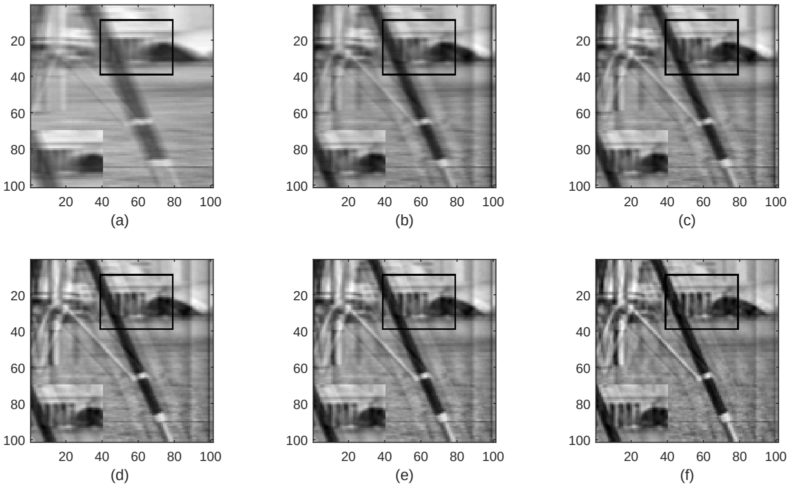


Figure 3: Restored images by using Algorithm (3): (a) $n_1 = n_2 = 60, k_1 = k_2 = 20$, (b) $n_1 - n_2 = 80, k_1 = k_2 = 30$, (c) $n_1 = n_2 = 90, k_1 = k_2 = 50$.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Results by using Algorithm (4): (a) Blurred noisy image, (b)-(f) Restored images, (b) $n_1 = n_2 = 10, k_1 = 7, k_2 = 5, \tau = 2$, $PSNR = 20.5742, e = 0.1830$, (c) $n_1 = 20, n_2 = 15, k_1 = 10, k_2 = 7, \tau = 4$, PSNR= 21.4767, Relative error= 0.1649, (d)$n_1 = n_2 = 30, k_1 = 10, k_2 = 20, \tau = 6$, PSNR= 22.3939, Relative error= 0.1484, (e)$n_1 = n_2 = 40, k_1 = k_2 = 30, \tau = 5$, PSNR= 22.8934, Relative error= 0.1401, (f) $n_1 = n_2 = 100, k_1 = 70, k_2 = 80, \tau = 8$, PSNR= 24.106, Relative error= 0.1274.

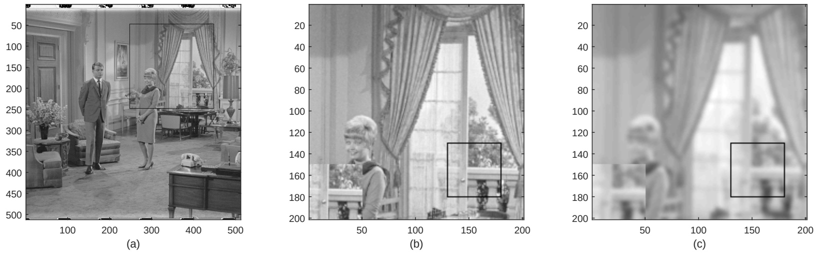


Figure 5: (a) - (b) Blur and noise free image, (c) Blurred image.

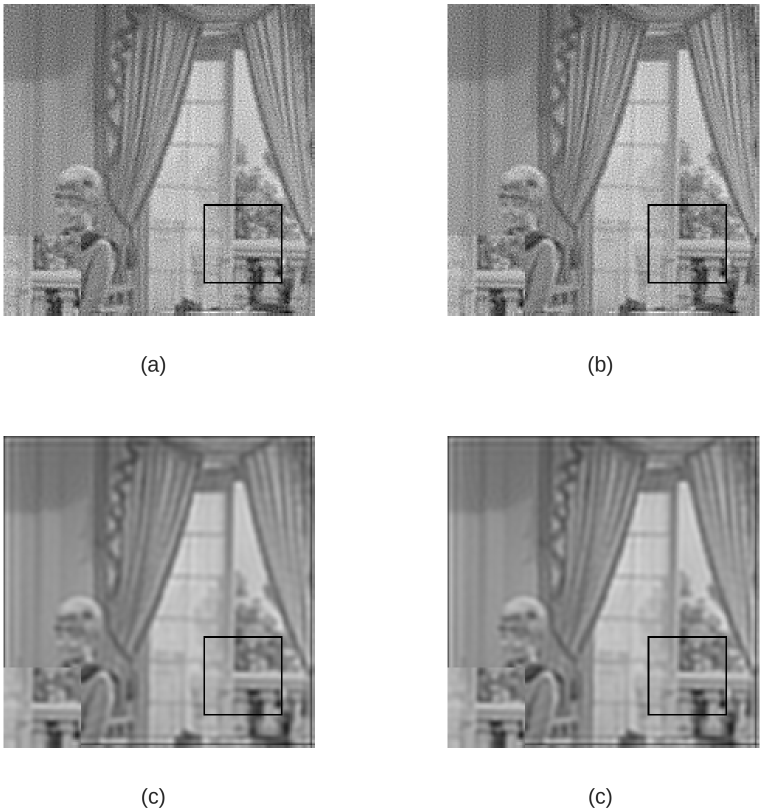
Example 2. The image "Couple (512×512)" is used as as a second test. Two dimensional Gaussian low pass filter of size 11×11 and standard deviation of 2 is used as a psf. The simulation results are given in Table.s 3-4 and Fig.s 5-7. In Table 3, for algorithm (3), we consider $n_1 = n_2 = n, k_1 = k_2 = k$. As it is shown in Table. 3, algorithm (3) is more accurate in comparison to algorithm (4). Also in Table. 4 for algorithm (2), $n = 100, k = 70$ and for algorithm (4), $n_1 = 30, n_2 = 50, k_1 = 10, k_2 = 15, \tau = 4$ are used. For Algorithm 4, image is divided by using t=109. According to Figures 6-7, the restored image by algorithm 3 is noise corrupted. It is more smoother than the image restored by algorithm 2. Based on Table 4, it is considered that in less time, the algorithm 4 is more accurate than algorithm 2.

Table 3: Comparison of Relative errors and PSNR-values for Example 2.

	Algorithm 2		Algorithm 3	
	Relative error	PSNR	Relative error	PSNR
n=120,k=80	0.1012	24.8644	0.0847	26.4122
n=90, k=30	0.1049	24.5558	0.0873	26.1557

Table 4: Comparison of Algorithm 2 and Algorithm 4 for Example 2.

	Time	Error	PSNR
Algorithm 2	17.092s	0.1012	24.8718
Algorithm 4	15.993s	0.0890	25.9883



(a)

(b)

(c)

(c)

Figure 6: (a)-(b): Restored images by using Algorithm (2), (a) $n = 90, k = 30$, (b) $n = 120, k = 80$, (c)-(d): Restored images by using Algorithm (3), (c) $n_1 = n_2 = 90, k_1 = k_2 = 30$, (d) $n_1 = n_2 = 120, k_1 = k_2 = 80$.

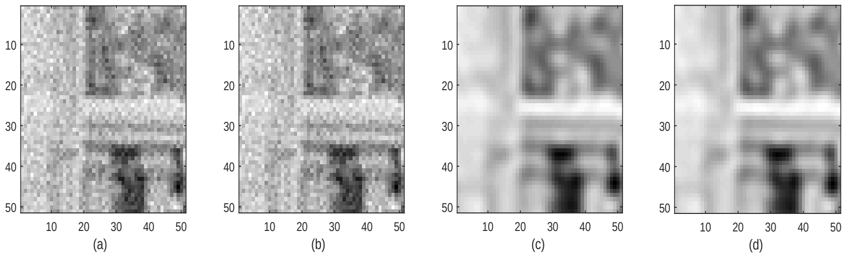
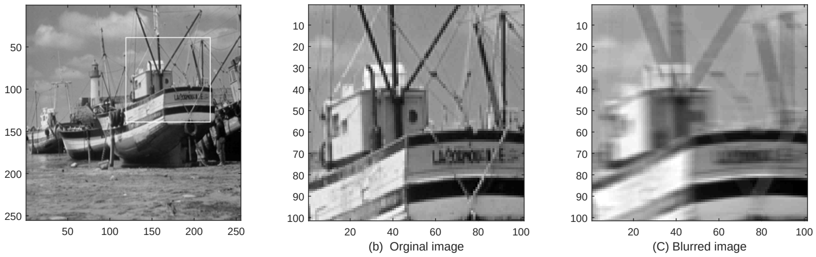


Figure 7: (a)-(b): Restored images by using Algorithm (2) , (a) $n = 90, k = 30$, (b) $n = 120, k = 80$, (c)-(d): Restored images by using Algorithm (3), (c) $n_1 = n_2 = 90, k_1 = k_2 = 30$, (d) $n_1 = n_2 = 120, k_1 = k_2 = 80$.



(b) Orginal image

(C) Blurred image

Figure 8: (a) - (b) Blur and noise free image, (c) Blurred image.

Example 3 As a last test the image of “boat 256 × 256” is utilized. Here, the psf is assumed to be a linear motion of camera by 7 pixels along the 180° . Also $n_1 = 60, n_2 = 50, k_1 = 30, k_2 = 20, \tau = 4$ are considered for Figures 9-11. In Figure 9, image is divided by using (3.2). In Figures 10-11, image is divided by using $t = 160$ and $t = 79$, respectively. According to numerical test results, it is considered that using a different partitioning can effect the simulation results.

5 Conclusion

In this paper, three algorithms based on ” defect principle ” have been proposed. Two images partitioning method for image restoration is introduced. The par-

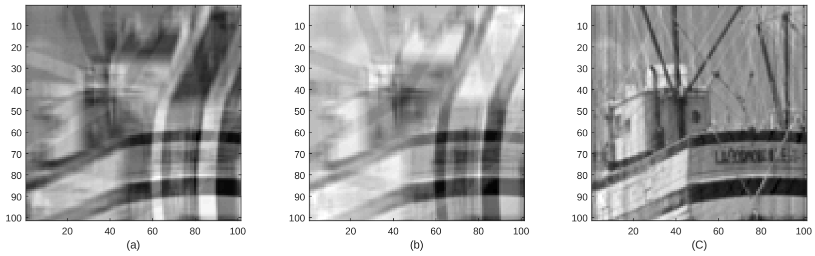


Figure 9: (a) - (b) Divided images by using (3.2), (c) Restored image, PSNR= 27.2973, Relative error= 0.0760.

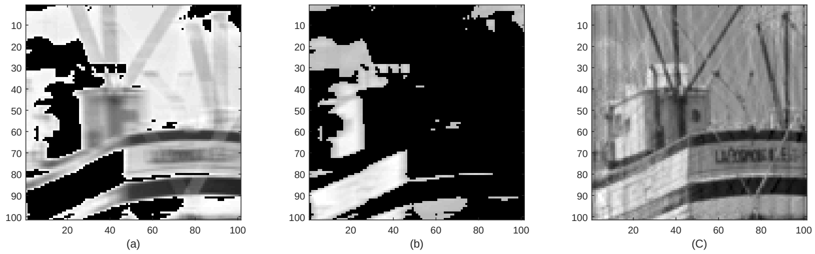


Figure 10: (a) - (b) Divided images by using $t = 160$, (c) Restored image, PSNR= 26.8061, Relative error= 0.0804.

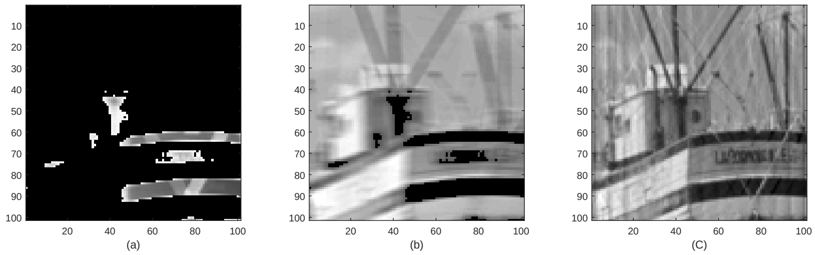


Figure 11: (a) - (b) Divided images by using $t = 79$, (c) Restored image, PSNR= 27.2260, Relative error= 0.0766.

titioning improves the restored image's quality and reduces the run times. The efficiency of the proposed algorithms are tested by using three test images which are blured and contaminated with different psfs and random noises. According to the numerical test results the proposed algorithm improved image quality segnificantly and in comparison to algorithm (2), contains less relative error. Moreover, The algorithm (4) has presented high accuracy in image deblurring with minimum run time.

References

[1] Hansen PC, Nagy JG, O'leary DP. Deblurring images: matrices, spectra, and filtering. Society for Industrial and Applied Mathematics; 2006 Jan 1.

[2] Xu J, Feng A, Hao Y, Zhang X, Han Y. Image deblurring and denoising by an improved variational model. AEU-International Journal of Electronics and Communications. 2016 Sep 30;70(9):1128-33.

[3] Ugur S, Arıkan O. SAR image reconstruction and autofocus by compressed sensing. Digital Signal Processing. 2012 Dec 31;22(6):923-32.

[4] Donatelli M, Martin D, Reichel L. Arnoldi methods for image deblurring with anti-reflective boundary conditions. Applied Mathematics and Computation. 2015 Feb 15;253:135-50.

[5] Serra-Capizzano S. A note on antireflective boundary conditions and fast deblurring models. SIAM Journal on Scientific Computing. 2004;25(4):1307-25 .

[6] Ng MK, Chan RH, Tang WC. A fast algorithm for deblurring models with Neumann boundary conditions. SIAM Journal on Scientific Computing. 1999;21(3):851-66.

[7] G. Gu, J. Ling, Donatelli M, Serra-Capizzano S. Antireflective boundary conditions for deblurring problems. Journal of Electrical and Computer Engineering. 2010 Jan 1;2010:2.

[8] Andrews HC, Hunt BR. Digital Image Restoration, 1977.

[9] Gonzalez RC, Woods R. E.(1992): Digital Image Processing. Addison-Wesley. 1992;5:11-5.

[10] Ng MK, Chan RH, Tang WC. A fast algorithm for deblurring models with Neumann boundary conditions. SIAM Journal on Scientific Computing. 1999;21(3):851-66.

[11] Donatelli M, Estatico C, Martinelli A, Serra-Capizzano S. Improved image deblurring with anti-reflective boundary conditions and re-blurring. Inverse problems. 2006 Oct 13;22(6):2035.

[12] Christiansen M, Hanke M. Deblurring methods using antireflective boundary conditions. SIAM Journal on Scientific Computing. 2008 Feb 14;30(2):855-72.

[13] Donatelli MA, Serra-Capizzano ST. Anti-reflective boundary conditions and re-blurring. Inverse Problems. 2004 Dec 6;21(1):169.

[14] Golub GH, Hansen PC, O'Leary DP. Tikhonov regularization and total least squares. SIAM Journal on Matrix Analysis and Applications. 1999;21(1):185-94.

[15] Novati P, Russo MR. A GCV based Arnoldi-Tikhonov regularization method. BIT Numerical mathematics. 2014 Jun 1;54(2):501-21.

[16] Golub GH, Von Matt U. Tikhonov regularization for large scale problems. InWorkshop on scientific computing 1997 (pp. 3-26).

[17] Hansen PC. Analysis of discrete ill-posed problems by means of the L-curve. SIAM review. 1992 Dec;34(4):561-80.

[18] Hansen PC, OLeary DP. The use of the L-curve in the regularization of discrete ill-posed problems. SIAM Journal on Scientific Computing. 1993 Nov;14(6):1487-503.

[19] Lewis B, Reichel L. ArnoldiTikhonov regularization methods. Journal of Computational and Applied Mathematics. 2009 Apr 1;226(1):92-102.

[20] Calvetti D, Morigi S, Reichel L, Sgallari F. Tikhonov regularization and the L-curve for large discrete ill-posed problems. Journal of computational and applied mathematics. 2000 Nov 1;123(1):423-46.

[21] Bouhamidi A, Jbilou K. Sylvester Tikhonov-regularization methods in image restoration. Journal of Computational and Applied Mathematics. 2007 Sep 1;206(1):86-98.

[22] Bouhamidi A, Jbilou K, Reichel L, Sadok H. A generalized global Arnoldi method for ill-posed matrix equations. Journal of Computational and Applied Mathematics. 2012 Feb 29;236(8):2078-89.

[23] Neuman A, Reichel L, Sadok H. Implementations of range restricted iterative methods for linear discrete ill-posed problems. Linear Algebra and its Applications. 2012 May 15;436(10):3974-90.

[24] Huang Y, Ng MK, Wen YW. A fast total variation minimization method for image restoration. Multiscale Modeling & Simulation. 2008 Aug 6;7(2):774-95.

[25] Novati P, Russo MR. Adaptive Arnoldi-Tikhonov regularization for image restoration. Numerical Algorithms. 2014 Apr 1;65(4):745-57.

[26] Reichel L, Sgallari F, Ye Q. Tikhonov regularization based on generalized Krylov subspace methods. Applied Numerical Mathematics. 2012 Sep 1;62(9):1215-28.

[27] Perrone L. Kronecker product approximations for image restoration with antireflective boundary conditions. Numerical linear algebra with applications. 2006 Feb 1;13(1):1-22.

[28] Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics. 1979 May 1;21(2):215-23.