

# Blind image restoration as a convex optimization problem

A. Bouhamidi<sup>1a</sup>, K. Jbilou<sup>1</sup>

<sup>1</sup>Université de Lille Nord de France, L.M.P.A, ULCO, 50 rue F. Buisson BP 699, F-62228 Calais-Cedex, France

Received 12 December 2009, Accepted 15 February 2010

**Abstract-** In this paper, we consider the blind image restoration as a convex constrained problem and we propose to solve this problem by a conditional gradient method. Such a method is based on a Thikonov regularization technique and is obtained by an approximation of the blur matrix as a Kronecker product of two matrices given as a sum of a Toeplitz and a Hankel matrices. Numerical examples are given to show the efficiency of our proposed method.

**Key words:** Image restoration; Kronecker product; Tikhonov regularization; convex optimization

## 1 Introduction

The problem of image restoration consists of the reconstruction of an original image that has been digitized and has been degraded by a blur and an additive noise. Image restoration techniques apply an inverse procedure to obtain an estimate of the original image. The background literature on image restoration has become quite large. Some treatments and overviews on image restoration are found in [1, 2, 3].

The blur and the additive noise may arise from many sources such as thermal effects, atmospheric turbulence, recording errors and imperfections in the process of digitization. The blurring process is described mathematically by a *point spread function* (PSF), which is a function that specifies how pixels in the image are distorted. We assume that the degradation process is represented by the following linear model

$$g(i, j) = (f * h)(i, j) + \nu(i, j)$$

The pair  $(i, j)$  is the discrete pixel coordinate and  $*$  denotes the discrete convolution operator. Here  $f$  represents the true image,  $h$  is the PSF,  $\nu$  is the additive noise and  $g$  is the degraded image. More explicitly,

$$g(i, j) = \sum_{l, k} f(l, k)h(i-l, j-k) + \nu(i, j) \quad (1)$$

Blind restoration refers to the image processing task of restoring the original image from a blurred version without the knowledge of the point spread function. Hence, both the PSF and the restored image must be estimated directly from the observed noisy and blurred image. The PSF is often assumed to be spatially invariant [2], which means that the blur is independent of the position of the points. The discrete model with spatially invariant PSF in the presence of an additive noise, can also be modeled in a matrix form as

$$g = H\hat{x} + \mathbf{n} \quad (2)$$

where  $\hat{x}$ ,  $\mathbf{n}$  and  $g$  are  $n^2$ -vectors representing the true image  $\hat{X}$ , the distorted image  $G$  and the additive noise  $\mathbf{N}$  of size  $n \times n$ , respectively. The vectors  $\hat{x}$ ,  $g$  and  $\mathbf{n}$  are obtained by stacking the columns of the matrices  $\hat{X}$ ,  $G$  and  $\mathbf{N}$ , respectively.

It is well-known that the blurring  $n^2 \times n^2$  matrix  $H$  is in general very ill-conditioned. The ill-conditioning results from the fact that many singular values of different orders of magnitude are close to the origin [4]. Another difficulty is due to the size of  $H$  which is in fact extremely large. We note that if the PSF is separable, then the matrix  $H$  may be decomposed into a Kronecker product of matrices with a smaller size. When the PSF is not separable, the matrix  $H$  can still be approximated by a Kronecker product [5, 6, 7].

## 2 Approximation of the blurring matrix

In practical restoration problems the PSF is unknown and in this case, the problem of the image restoration is known as a blind image restoration, see for instance [8, 9, 10, 11]. Then we need to estimate the point spread function (PSF) characterizing the blur. Namely, we need to estimate the matrix  $P$  that contains the image of the point spread function. An estimation of the matrix  $P$  may be obtained by using an iterative deconvolution scheme introduced by Ayers and Dainty [8].

The algorithm starts with a guess for the true image  $f_k$  and a guess for the PSF  $h_k$  with  $k = 0$ . So, at a step  $k$ , we pass to the Fourier domain by computing  $\hat{F}_k = FFT(f_k)$  and  $\hat{H}_k = FFT(h_k)$  and we compute the matrix  $F_{k+1} = \hat{F}_k + \Delta F_k$  and  $H_{k+1} = \hat{H}_k + \Delta H_k$  where the incremental Wiener filter  $\Delta F_k$  and  $\Delta H_k$  are given by

$$\Delta F_k = \frac{(G - \hat{F}_k \odot \hat{H}_k) \overline{\hat{H}_k}}{\|\hat{H}_k\|_F^2 + \alpha^2}$$

<sup>a</sup>Corresponding author: bouhamidi@lmpa.univ-littoral.fr

and

$$\Delta H_k = \frac{(G - \widehat{F}_k \odot \widehat{H}_{k-1}) \widehat{H}_{k-1}}{\|\widehat{H}_{k-1}\|_F^2 + \alpha^2}$$

Here the product  $\odot$  stands for the Hadamard product and  $\widehat{H}_k$  stands for the conjugate of the matrix  $\widehat{H}_k$ . The constant parameter  $\alpha^2$  represents the noise to signal ratio and it is determined as an approximation of the variance of the additive noise. Then, we compute the new approximation of the image  $f_{k+1} = IFFT(F_{k+1})$  and the PSF approximation  $h_{k+1} = IFFT(H_{k+1})$ . In each iteration, the image constraints are imposed

$$f_k(i, j) = \begin{cases} f_k(i, j) & \text{if } f_k(i, j) \in [0, 255] \\ f_k(i, j) = 0 & \text{if } f_k(i, j) < 0 \\ f_k(i, j) = 255 & \text{if } f_k(i, j) > 255 \end{cases}$$

Together with the following blur constraints of the non negativity and the normalization of the PSF,

$$\begin{cases} h_k(i, j) \geq 0, \\ \sum_{i,j} h_k(i, j) = 1 \end{cases}$$

So we increment the step  $k$  from  $k = 0$  to  $k = k_{max}$ . At the end of the algorithm we obtain an approximation denoted by  $P = h_k$  of the image containing the image of the PSF.

We recall that, the Kronecker product of a matrix  $A = (a_{ij})$  of size  $n \times p$  and a matrix  $B$  of size  $s \times q$  is defined as the matrix  $A \otimes B = (a_{ij}B)$  of size  $(ns) \times (pq)$ . The *vec* is the operator which transforms a matrix  $A$  of size  $n \times p$  to a vector  $a$  of size  $np \times 1$  by stacking the columns of  $A$ . For  $A$  and  $B$  two matrices in  $\mathbb{R}^{n \times p}$ , we define the following inner product  $\langle A, B \rangle_F = \text{trace}(A^T B)$ . It follows that the well known Frobenius norm denoted here by  $\|\cdot\|_F$  is given by  $\|A\|_F = \sqrt{\langle A, A \rangle_F}$ .

In the context of image restoration and when the point spread function (PSF) is separable the blurring matrix  $H$  given in (2) can be decomposed as a Kronecker product  $H = H_2 \otimes H_1$  of two smaller blurring matrices of appropriate sizes. In the non separable case, one can approximate the matrix  $H$  by solving the Kronecker product approximation problem (KPA) [7]

$$\min_{H_1, H_2} \|H - H_2 \otimes H_1\|_F \quad (3)$$

Recently, Kamm and Nagy [5, 6] introduced an efficient algorithm for computing a solution of the KPA problem in image restoration. Let us now give a breve description of the algorithm given in [6]. We assume that the size of the image is  $n \times n$ . For a given vector  $a = (a_1, \dots, a_n)^T \in \mathbb{R}^n$ , the matrix  $\text{toep}(a, k)$  is a banded Toeplitz matrix of size  $n \times n$  whose diagonals are constant and whose  $k$ -th column is  $a = (a_1, \dots, a_n)^T$ ; the other elements are zero. The matrix  $\text{hank}(a, k)$  is a Hankel matrix of size  $n \times n$  whose anti-diagonals are constant and

whose first row and last column are defined by the vectors  $(a_{k+1}, \dots, a_n, 0, \dots, 0)$  and  $(0, \dots, 0, a_1, \dots, a_{k-1})^T$ , respectively.

We assume that the center of the PSF (location of the point source) is at  $p_{\ell, k}$ , where  $P = (p_{ij})$  is the  $n \times n$  matrix containing the image of a the point spread function. The aim of the following algorithm is to compute vectors  $\widehat{a}$  and  $\widehat{b}$  of length  $n$  such that the matrices  $\widehat{H}_1 = \widehat{A}_t + \widehat{A}_h$  and  $\widehat{H}_2 = \widehat{B}_t + \widehat{B}_h$  where

$$\begin{aligned} \widehat{A}_t &= \text{toep}(\widehat{a}, i) & \text{and} & & \widehat{A}_h &= \text{hank}(\widehat{a}, i) \\ \widehat{B}_t &= \text{toep}(\widehat{b}, j) & \text{and} & & \widehat{B}_h &= \text{hank}(\widehat{b}, j) \end{aligned}$$

solve the Kronecker product approximation (3). Let  $R_n$  be the Cholesky factor of the  $n \times n$  symmetric Toeplitz matrix  $T_n = \text{Toeplitz}(v_n)$  with its first row  $v_n = (n, 1, 0, 1, 0, \dots)$ . The algorithm, given in [6], for constructing the matrices  $\widehat{A}_t, \widehat{A}_h, \widehat{B}_t$  and  $\widehat{B}_h$  is as follows,

#### ALGORITHM

1. Compute  $R_n$ ,
2. Construct  $P_r = R_n P R_n^T$
3. Compute the SVD:  $P_r = \sum \sigma_i u_i v_i^T$
4. Construct the vectors:
 
$$\widehat{a} = \sqrt{\sigma_1} R_n^{-1} v_1 \quad \text{and} \quad \widehat{b} = \sqrt{\sigma_1} R_n^{-1} u_1$$
5. Construct the matrices:
 
$$\begin{aligned} \widehat{A}_t &= \text{toep}(\widehat{a}, \ell), & \widehat{A}_h &= \text{hank}(\widehat{a}, \ell), \\ \widehat{B}_t &= \text{toep}(\widehat{b}, k), & \widehat{B}_h &= \text{hank}(\widehat{b}, k). \end{aligned}$$

### 3 Convex Tikhonov minimization problem

In order to determine an approximation of  $\widehat{x} = \text{vec}(\widehat{X})$ , we consider the following convex optimization problem

$$\min_{x \in \Omega} \|Hx - g\|_2 \quad (4)$$

The set  $\tilde{\Omega} \subset \mathbb{R}^n$  could be a simple convex set (e. g., a sphere or a box) or the intersection of some simple convex sets. Due to the ill-conditioning of the matrix  $H$ , we replace the original problem by a better conditioned one in order to diminish the effects of the noise in the data. One of the most popular regularization methods is due to Tikhonov. The method replaces the problem (4) by the new one

$$\min_{x \in \tilde{\Omega}} (\|Hx - g\|_2^2 + \lambda^2 \|Lx\|_2^2) \quad (5)$$

where  $L$  is a regularization operator chosen to obtain a solution with desirable properties such as small norm or good smoothness and the parameter  $\lambda$  is a scalar to be determined. The most popular methods for determining such a parameter  $\lambda$ , are the generalized cross-validation (GCV) method and the L-curve criterion, see [12, 13, 14, 15, 16, 17, 18].

Here, we assume that  $H = H_2 \otimes H_1$  and  $L = L_2 \otimes L_1$  where  $H_1, H_2, L_1$  and  $L_2$  are square matrices of dimension  $n \times n$ . Using the relations  $\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$  and  $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ , the problem (5) can be reformulated as

$$\min_{x \in \Omega} (\|H_1 X H_2^T - G\|_F^2 + \lambda^2 \|L_1 X L_2^T\|_F^2) \quad (6)$$

where the set  $\Omega$  is such that

$$x = \text{vec}(X) \in \tilde{\Omega} \subset \mathbb{R}^{n^2} \Leftrightarrow X \in \Omega \subset \mathbb{R}^{n \times n}$$

Then the problem (5) is replaced by a new one involving matrix equations with small dimensions. Now, we consider the function  $f_\lambda : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  given by

$$f_\lambda(X) = \|H_1 X H_2^T - G\|_F^2 + \lambda^2 \|L_1 X L_2^T\|_F^2$$

The convex constrained minimization problem (6) considered here is

$$\text{Minimize } f_\lambda(X) \quad \text{subject to } X \in \Omega. \quad (7)$$

The function  $f_\lambda : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  is differentiable and its gradient is given by the following formula

$$\begin{aligned} \nabla f_\lambda(X) &= 2 \left( H_1^T (A(X) - G) H_2 + \lambda^2 L_1^T \mathcal{L}(X) L_2 \right) \\ &= 2 \left( H_1^T (H_1 X H_2^T - G) H_2 + \lambda^2 L_1^T L_1 X L_2^T L_2 \right) \end{aligned}$$

The set  $\Omega$  could be a simple convex set (e. g., a sphere or a box) or the intersection of some simple convex sets. Specific cases that will be considered are

$$\Omega_1 = \{X \in \mathbb{R}^{n \times p} : L \leq X \leq U\} \quad (8)$$

and

$$\Omega_2 = \{X \in \mathbb{R}^{n \times p} : \|X\|_F \leq \delta\} \quad (9)$$

Here,  $Y \leq Z$  means  $Y_{ij} \leq Z_{ij}$  for all possible entries  $ij$ .  $L$  and  $U$  are given matrices and  $\delta > 0$  is a given scalar. Another option to be considered is  $\Omega = \Omega_1 \cap \Omega_2$ .

In this section, we describe the conditional gradient method for solving the convex constrained optimization problem (7). This method is well-known and was one of the first successful algorithms used to solve nonlinear optimization problems. It is also called Frank-Wolfe method. The algorithm can be summarized as follows

### Algorithm 1. The Conditional Gradient Algorithm

1. Choose a tolerance  $tol$ , an initial guess  $X_0 \in \Omega$ , and set  $k = 0$ .
2. Solve the minimization problem of a linear function over the set  $\Omega$ :
$$\min_{X \in \Omega} \langle \nabla f_\lambda(X_k) | X \rangle_F. \quad (*)$$
Let  $\bar{X}_k$  be a solution to problem (\*)
3. Compute the value:
$$\eta_k = \langle \nabla f_\lambda(X_k) | \bar{X}_k - X_k \rangle_F$$
4. If  $|\eta_k| < tol$  Stop else continue
5. Solve the one dimensional minimization problem
$$\min_{\alpha \in [0,1]} f_\lambda(X_k + \alpha(\bar{X}_k - X_k)). \quad (**)$$
Let  $\alpha_k^*$  be a solution to problem (\*\*)
6. Update  $X_{k+1} = X_k + \alpha_k^*(\bar{X}_k - X_k)$ , set  $k = k + 1$  and go to Step 2.

If the convex set  $\Omega$  consists of the set  $\Omega_1$  given by (8), then, a solution of the problem (\*) in Step 2 of Algorithm 1, is given by

$$[\bar{X}_k]_{ij} = \begin{cases} L_{ij} & \text{if } [\nabla f_\lambda(X_k)]_{ij} \geq 0 \\ U_{ij} & \text{if } [\nabla f_\lambda(X_k)]_{ij} < 0 \end{cases} \quad (10)$$

$M_{ij}$  denote the components of a matrix  $M$ . Indeed, from (10) we have

$$[\nabla f_\lambda(X_k)]_{ij} [\bar{X}_k]_{ij} \leq [\nabla f_\lambda(X_k)]_{ij} X_{ij}$$

for all  $X \in \Omega_1$ . Then for  $\bar{X}_k$  given by (10), we have

$$\langle \nabla f_\lambda(X_k) | \bar{X}_k \rangle_F \leq \langle \nabla f_\lambda(X_k) | X \rangle_F, \quad \forall X \in \Omega_1$$

If  $\Omega$  is chosen to be  $\Omega_2$  given by (9), then, a solution of the problem (\*) in Step 2 of Algorithm 1, is given by

$$\bar{X}_k = - \frac{\nabla f_\lambda(X_k) \delta}{\|\nabla f_\lambda(X_k)\|_F} \quad (11)$$

Indeed, for all  $X \in \Omega_2$ , we have

$$\begin{aligned} \langle \nabla f_\lambda(X_k) | X \rangle_F &\geq -\|\nabla f_\lambda(X_k)\|_F \|X\|_F \\ &\geq -\|\nabla f_\lambda(X_k)\|_F \delta \end{aligned}$$

and

$$\|\nabla f_\lambda(X_k)\|_F \delta = \langle \nabla f_\lambda(X_k) | \frac{\nabla f_\lambda(X_k) \delta}{\|\nabla f_\lambda(X_k)\|_F} \rangle_F$$

It follows that, for all  $X \in \Omega_2$ , we have

$$\langle \nabla f_\lambda(X_k) | X \rangle_F \geq \langle \nabla f_\lambda(X_k) | \bar{X}_k \rangle_F$$

where  $\bar{X}_k$  is given by (11).

Now, let  $H_k = \bar{X}_k - X_k$ , then it is easy to obtain

$$f_\lambda(X_k + \alpha H_k) = a_k \alpha^2 + b_k \alpha + c_k$$

where

$$\begin{aligned} a_k &= \|\mathcal{A}(H_k)\|_F^2 + \lambda^2 \|\mathcal{L}(H_k)\|_F^2 \\ b_k &= \langle \nabla f_\lambda(X_k) | H_k \rangle_F \\ c_k &= \|\mathcal{A}(X_k) - G\|_F^2 + \lambda^2 \|\mathcal{L}(X_k)\|_F^2 \end{aligned}$$

Then, it follows that the minimum of the quadratic one dimensional problem

$$\min_{\alpha} f_{\lambda}(X_k + \alpha H_k)$$

is analytically given by

$$\alpha_k = -\frac{b_k}{2a_k} = -\frac{\langle \nabla f_{\lambda}(X_k) | H_k \rangle_F}{2\|\mathcal{A}(H_k)\|_F^2 + \lambda^2\|\mathcal{L}(H_k)\|_F^2} \quad (12)$$

which may be written in the following form

$$\alpha_k = -\frac{\langle \mathcal{A}(X_k) - G | \mathcal{A}(H_k) \rangle_F + \lambda^2 \langle \mathcal{L}(X_k) | \mathcal{L}(H_k) \rangle_F}{\|\mathcal{A}(H_k)\|_F^2 + \lambda^2\|\mathcal{L}(H_k)\|_F^2} \quad (13)$$

Then, the solution of the problem (\*\*) in Step 5 of Algorithm 1, is given by

$$\alpha_k^* = \begin{cases} \alpha_k & \text{if } 0 \leq \alpha_k \leq 1 \\ 1 & \text{if } \alpha_k > 1 \\ 0 & \text{if } \alpha_k < 0 \end{cases}$$

The following algorithm combines the conditional gradient method together with the Tikhonov regularization. The convex set  $\Omega$  is the one given by (8) or (9).

#### Algorithm 2.

##### The Conditional Gradient-Tikhonov Algorithm

1. Choose a tolerance  $tol$ , an initial guess  $X_0 \in \Omega$ , and set  $k = 0$ .
2. Determine  $\lambda$  by the L-curve method
3. While  $k < k_{max}$ 
  - 3.1- Compute the matrix  $\bar{X}_k$  by using the relation (10),
  - 3.2- Compute the value:
$$\eta_k = \langle \nabla f_{\lambda}(X_k) | \bar{X}_k - X_k \rangle_F$$
  - 3.3- If  $|\eta_k| < tol$  Stop else continue,
  - 3.4- Compute  $\alpha_k$  by using (12) or (13),
  - 3.5- If  $\alpha_k > 1$  then  $\alpha_k^* = 1$ ,  
ElseIf  $\alpha_k < 0$  then  $\alpha_k^* = 0$ ,  
Else  $\alpha_k^* = \alpha_k$ ,  
EndIF.
  - 3.6- Update  $X_{k+1} = X_k + \alpha_k^*(\bar{X}_k - X_k)$ ,
  - 3.7- Set  $k = k + 1$ ,
4. EndWhile.

## 4 Numerical examples

In this section we give a numerical example to illustrate our proposed method. The original  $512 \times 512$  "fruit" image was degraded by a 'speckle' multiplicative noise with different values of the variance  $\sigma_m$  plus an additive white Gaussian noise with zero mean and different values of the variance  $\sigma_a$ . Figure 1 shows the original image. The degraded image was corrupted with a multiplicative noise with variance  $\sigma_m = 0.01$  plus an additive white Gaussian noise with the variance  $\sigma_a = 0.02$  is presented on Figure 2. In order to define local smoothing

constraints, we determine the bound matrices  $L_b$  and  $U_b$  from the parameters that describe the local properties of an image. For the degraded image  $G$ , the local mean matrix  $\bar{G}$  and the local variance  $\sigma_G^2$  are measured over a  $3 \times 3$  window and are given by

$$\bar{G}(i, j) = \frac{1}{9} \sum_{l=i-3}^{i+3} \sum_{k=j-3}^{j+3} G(l, k)$$

and

$$\sigma_G^2(i, j) = \frac{1}{9} \sum_{l=i-3}^{i+3} \sum_{k=j-3}^{j+3} [G(l, k) - \bar{G}(l, k)]^2$$

The maximum local variance over the entire image  $G$ , denoted by  $\sigma_{max}^2$  is given by

$$\sigma_{max}^2 = \max_{1 \leq i, j \leq 256} \sigma_G^2(i, j)$$

Let  $\beta > 0$  be a positive constant, the matrices  $L_b$  and  $U_b$  defining the domain  $\Omega_1$  are given by

$$L_b(i, j) = \max(\bar{G}(i, j) - \beta \frac{\sigma_G^2(i, j)}{\sigma_{max}^2}, 0)$$

and

$$U_b(i, j) = \bar{G}(i, j) + \beta \frac{\sigma_G^2(i, j)}{\sigma_{max}^2}$$

The constant  $\beta$  controls the tightness of the bounds. In the following numerical tests, the domain was chosen with  $\beta = 50$ .

Here we mainly compare the visual quality of the restored images and the values of the PSNR. We recall that the PSNR is the peak signal-to-noise ratio (PSNR) and it measures the distortion between the original image  $I_0$  and the degraded image  $I = G$  or the restored image  $I = X$  and is defined by

$$PSNR(I) = 10 \log_{10} \left( \frac{d^2}{\frac{1}{n^2} \|I_0 - I\|_F^2} \right)$$

where  $d = 255$  in the case of gray images and  $n \times m$  is the size of the images; in our case we have  $n = m = 500$ . We recall that

$$\|I_0 - I\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n |I_0(i, j) - I(i, j)|^2$$

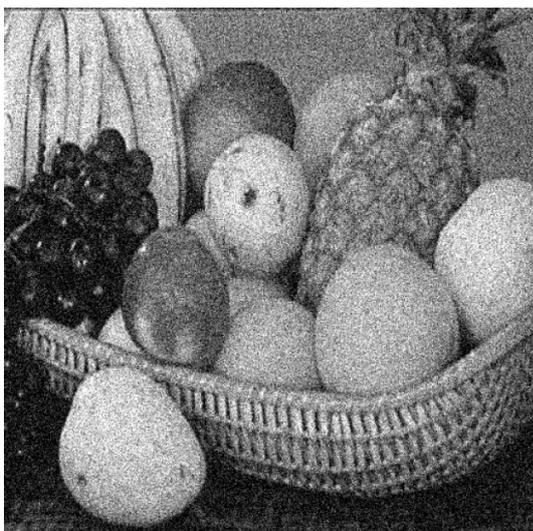
The value of the  $PSNR0 = PSNR(G)$  for the degraded image was 16.94dB. The restored image is presented on Figure 3 and the value of the  $PSNR1 = PSNR(X)$  was improved to 26.42dB.

Table 1: PSNR for different values of the variance of the multiplicative and the additive noises

$\sigma_m$	$\sigma_a$	PSNR0	PSNR1
0.1	0.001	16.02	23.53
0.25	0.001	13.88	20.69
0.50	0.001	11.83	18.99
0.001	0.1	14.54	18.36
0.001	0.25	12.00	15.03



**Fig. 1:** Original image



**Fig. 2:** Degraded image



**Fig. 3:** Restored image

## References

1. H. Andrews, B. Hunt, *Digital image restoration*, Prentice-Hall, Engelwood Cliffs, NJ, (1977).
2. A.K. Jain, *Fundamentals of digital image processing*, Prentice-Hall, Engelwood Cliffs, NJ, (1989).
3. R.L. Lagendijk, J. Biemond, *Iterative identification and restoration of images*, Norwell, MA: Kluwer Academic Publishers, (1991).
4. H.W. Engl, M. Hanke, A. Neubauer, *Regularization of inverse problems*, Kluwer, Dordrecht, The Netherlands, 71–143, (1996).
5. J. Kamm, J.G. Nagy, *Kronecker product and SVD approximations in image restoration*, *Linear Algebra and its Applications* **284**, 177–192, (1998).
6. J. Kamm, J.G. Nagy, *Kronecker product approximations for restoration image with reflexive boundary conditions*, *SIAM J. Matrix Anal. Appl.*, **25**(3), 829–841, (2004).
7. C.F. Van Loan, N.P. Pitsianis, *Approximation with Kronecker products*, M.S. Moonen, G.H. Golub (Eds.), *Linear Algebra for large scale and real time applications*, Kluwer Academic Publishers, Dordrecht, 293–314, (1993).
8. G. R. Ayers, J. C. Dainty, *Iterative blind deconvolution method and its applications*, *Optics Letters*, **13** (7), 547–549, (1988).
9. L. B. Lucy, *An iterative technique for the rectification of observed distributions*, *Astronomical Journal*, **79**, 745–754, (1974).

10. W. H. Richardson, *Bayesian-based iterative method of image restoration*, J. Optic. Soc. Amer. A, **62**, 55–59, (1972).
11. A. Pruessner, D. P. O’Leary, *Blind deconvolution using a regularized structured total least norm algorithm*, SIAM J. Matrix Anal. Appl. **24** (4), 1018–1037, (2003).
12. G.H. Golub, U. von Matt, *Tikhonov regularization for large scale problems*, in: G.H. Golub, S.H. Lui, F. Luk, R. Plemmons (Eds.), Workshop on Scientific Computing, Springer, New York, 3–26, (1997).
13. A. Bouhamidi, K. Jbilou, *Sylvester Tikhonov-regularization methods in image restoration*, J. Comput. Appl. Math., **206**, 1, 86–98, (2007).
14. M. Hanke, P.C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Indust., **3**, 253–315, (1993).
15. P. C. Hansen *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., **34**, 561–580, (1992).
16. D. Calvetti, G.H. Golub, L. Reichel, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, **39**, 603–619, (1999).
17. D. Calvetti, B. Lewis, L. Reichel, *GMRES, L-curves and discrete ill-posed problems*, BIT, **42**, 44–65, (2002).
18. G.H. Golub, M. Heath, G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics **21**, 215–223, (1979).