Robust Binary Image Deconvolution with Positive Semidefinite Programming

Yijiang Shen, Edmund Y. Lam and Ngai Wong *

Abstract

This paper reports on a novel approach to binary image deconvolution using Positive Semidefinite (PSD) Programming. We note the combinatorial nature of this problem: binary image deconvolution requires the minimization of a global energy function over binary variables, taking into account not only both local similarity and spatial context, but more specifically the relationship between individual pixel values and the point spread function. We subsequently modify the problem to a convex relaxation of the original problem without introducing additional parameters. We first compute the optimal solution of the convex relaxation based on PSD programming, and then use the randomized-hyperplane method to find the combinatorial solution to the original problem. We apply our approach to a collection of blurred binary images, and show the advantages of this approach in binary image deconvolution.

Keywords: point spread function (PSF), convex relaxation, image restoration, Positive Semidefinite (PSD) Programming, randomized-hyperplane

1 Introduction

Image blurring is a common phenomenon in photography, and is caused by various reasons, such as moving objects in still images (motion blur), defocus, and vibration in the machinery. Image restoration is therefore important in obtaining a good estimate of the degraded images [1]. One important special case is where the true scenery is binary, appropriate for many useful scenarios such as black-illuminated opaque objects, fingerprinting recognition, automated document handling and alike. There have been numerous efforts in the area using machine vision techniques to deal with binary image restoration. Meloche and Zamar [2] developed the weighted mean square error (WMSE) method to restore noisy images. Hitchcock and Glasbey [3] identified a statistical model for digital image data, and proposed an inferential procedure to restore images of blob-like and filamentous objects. In other words, the method required some assumptions on the form of the image. Gu *et al.* [4] used pulse coupled neural network to restore noisy binary images. Chan *et al.* [5] provided a convergent method to find a minimizer of the total-variation functional to restore noisy binary images. These approaches, however, are designed to restore noisy binary images, while in most cases, images are degraded not only because noise is present, but also because they are blurred.

Most blurring processes can be approximated by the two dimensional convolution of the true image $f(x_1, x_2)$ with a linear shift-invariant blur, also known as the point spread function (PSF), $h(x_1, x_2)$. That is,

$$g(x_1, x_2) = f(x_1, x_2) * h(x_1, x_2), \qquad (1)$$

in which "$*$" denotes the two dimensional linear convolution operator and $g(x_1, x_2)$ represents the degraded image. The problem of recovering the true image $f(x_1, x_2)$ requires the deconvolution of the PSF $h(x_1, x_2)$ from the degraded image $g(x_1, x_2)$. Several deconvolution methods have been proposed such as Maximum Entropy Method [6], Wiener filter, One-Step Least Squares [7], etc. The reversal of blur is a numerically unstable procedure which can be made tractable only by including assumptions about the form of the true scene. For example, a linear deconvolution such as the Wiener filter suppresses the high frequency component of the images, and in maximum entropy restoration the constraint is used such that pixels cannot take negative values. Recently, the combinatorial nature of binary image restoration has been noted in [8, 9, 10, 11]: the optimization process to restore a noisy binary image requires the the minimization of a global energy function over binary vari-

*Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong Tel: 852-28592696 Email: yjshen@eee.hku.hk, elam@eee.hku.hk, nwong@eee.hku.hk

(Advance online publication: 1 February 2007)

ables. The optimization method was studied in detail and its application to several combinatorial problems such as binary partitioning, perceptual grouping and restoration was further discussed in [12]. In this paper, we extend this optimization process and design the energy function which combines not only both the local similarity and spatial context, but more specifically the relationship between individual pixel values and the PSF to deal with binary image deconvolution with known linear PSFs. As such, both deblurring and denoising functions are simultaneously carried out through the proposed method.

This paper is organized as follows. Section 2 describes the binary combinatorial optimization of the problem and designs the energy function. Section 3 gives an introduction to positive semidefinite programming and describes the optimization relaxation to solve the combinatorial problem in Section 2. Experimental results are given in Section 4. Finally, Section 5 draws some concluding remarks and provides some future insights.

2 Binary Combinatorial Optimization

Keuchel *et al.* [9, 12] investigated the combinatorial problem of binary image restoration or binary image denoising. Consider some scalar-valued feature (gray-value, color feature, etc.) g, suppose that, for each pixel i, the feature-value g_i is known to originate from either of two prototypical values u_1, u_2. In practice, of course, g is real-valued due to measurement errors and noise. For binary image restoration and in our case, binary image deconvolution, g_i would be the pixel values, and the two prototype values u_1 and u_2 can be represented by $u_1 : -1$ and $u_2 : +1$ without loss of generosity. To restore a discrete-valued image function represented by the vector $x \in \{-1, +1\}^n$ from the measurement g, we would like to minimize the functional:

$$z(x) = \frac{1}{4}\sum_i ((u_2 - u_1)x_i + u_2 + u_1 - 2g_i)^2 + \frac{\lambda}{2}\sum_{<i,j>}(x_i - x_j)^2, \qquad (2)$$

with λ being the smoothness term parameter, and $< i, j >$ stands for adjacent vector entry indices. Equation (2) comprises two terms, namely, a data-fitting term and a smoothness term modeling spatial context. Suppose that the true image is of size u by v, vector $x \in \{-1, +1\}^n$ is of size uv, and contains the columns of true binary image, stacked upon one another. It can be noted that the numerical result of (2) finds its local minimum when vector $x \in \{-1, +1\}^n$ represents the true binary image: by assigning every individual x_i as either of the prototype values u_1 or u_2, every summation in the first term of (2) will reach its minimum. As such the whole optimization process will minimize the functional and denoise the degraded binary image. Up to constant terms, (2) leads to the following optimization problem:

$$\inf_x \frac{1}{4}x^T Qx + \frac{1}{2}b^T x, x \in \{-1, +1\}^n, \qquad (3)$$

with $b_i = (u_2 - u_1)(u_1 + u_2 - 2g_i)$ and matrix entries $Q_{ij} = -2\lambda$ for adjacent pixels i, j and $Q_{ij} = 0$ otherwise.

When a true image f is blurred by a linear PSF h, the blurred image g is the convolution of f and h as in (1). Suppose that h is a matrix of size m by n, and the entries of h are denoted by $h(i, j), i = 1, ..., m, j = 1, ..., n$. In the computation of convolution, the columns and lines of h are flipped to form h_r as follows

$$h_r(i, j) = h(m + 1 - i, n + 1 - j). \qquad (4)$$

For a specific pixel x_i in the discrete-valued image function represented by $x \in \{-1, +1\}^n$, the blurred pixel value would be the linear combination of x_i together with its neighboring pixel values and entry values of the PSF:

$$x_i^b = \sum_{j=1}^{m}\sum_{k=1}^{n} x_{i+(j-\tilde{m})+(k-\tilde{n})u} h_r(j, k), \qquad (5)$$

with $\tilde{m}$ and $\tilde{n}$ being the “center” entry of h_r, respectively, $\tilde{m} = \lfloor \frac{m+1}{2} \rfloor$ and $\tilde{n} = \lfloor \frac{n+1}{2} \rfloor$. To restore the vector $x \in \{-1, +1\}^n$, now the functional is changed by replacing x_i in (2) with x_i^b in the data-fitting term, and we have

$$z(x) = \frac{1}{4}\sum_i ((u_2 - u_1)x_i^b + u_2 + u_1 - 2g_i)^2 + \frac{\lambda}{2}\sum_{<i,j>}(x_i - x_j)^2. \qquad (6)$$

Similarly as in (2), every single summation in the first term of the energy function designed as in (6) will take its numerical minimum when individual x_i is assigned a prototype value, either u_1 or u_2, which represents the true binary image scene. Therefore the functional in (6) will be able to deconvolve and denoise the degraded binary images at the same time. It can also be noted that in (6), u_1, u_2, g_i are constants and $x_i^2 = 1$, the computation of (6) leads to

$$\frac{1}{2}\sum_i (u_2 - u_1)(u_2 + u_1 - 2g_i)x_i^b + \frac{1}{4}\sum_i ((u_2 - u_1)x_i^b)^2 - \lambda \sum_{<i,j>} x_i x_j. \quad (7)$$

We define $g_{(\hat{h},\hat{v})}$ to represent the image shifting g by $\hat{h}$ lines and $\hat{v}$ columns, in which $\hat{h} > 0$ means shifting $\hat{h}$ lines down, otherwise $|\hat{h}|$ up; and $\hat{v} > 0$ means shifting $\hat{v}$ columns right, otherwise $|\hat{v}|$ left. To represent $\sum_i (u_2 - u_1)(u_2 + u_1 - 2g_i) \sum_{j=1}^{m} \sum_{k=1}^{n} x_{i+(j-\tilde{m})+(k-\tilde{n})u} h_r(j,k)$ in the form of $b^T x$, b_i is defined as:

$$b_i = \sum_{j=1}^{m}\sum_{k=1}^{n} (u_2 - u_1)(u_2 + u_1 - 2g_{(j-\tilde{m},k-\tilde{n})i})h_r(j,k), \quad (8)$$

where g is the blurred image. To represent the summation

$$\sum_i \frac{1}{4}(u_2 - u_1)^2 (\sum_{j=1}^{m}\sum_{k=1}^{n} x_{i+(j-\tilde{m})+(k-\tilde{n})u} h_r(j,k))^2, \quad (9)$$

it is noted that for any i, $x_i^2 = 1$ and $\sum_i x_{i+(j-\tilde{m})+(k-\tilde{n})u} x_{i+(\hat{j}-\tilde{m})+(\hat{k}-\tilde{n})u}$, in which $j = \hat{j}$ and $k = \hat{k}$ do not happen concurrently, differs from $\sum_i x_i x_{i+(j-\hat{j})+(k-\hat{k})u}$ in that for an arbitrary $\sum_i x_i x_{i+p+qu}$, if $p > 0$ then the summation does not include the last p lines in the image, otherwise the first p lines; if $q > 0$, the summation does not include the last q columns, otherwise the first q columns in the image. Q of size uv by uv is constructed as: when the summation region of $\sum_i x_{i+(j-\tilde{m})+(k-\tilde{n})u} x_{i+(\hat{j}-\tilde{m})+(\hat{k}-\tilde{n})u}$ is identified and we define $T = (u_2 - u_1)^2/2$, for line i, $Q(i, i + (j - \hat{j}) + (k - \hat{k})u)$ will be added with value $2 \times T \times h_r(j,k)h_r(\hat{j},\hat{k})$. There are altogether $\frac{mn\times(mn-1)}{2}$ summation terms in (9). The third term of (6) can also be represented in Q, viz., for every line i of Q, $Q(i, i-1)$ and $Q(i, i+1)$ will be added with value -2λ. Thus the second term and the third term can be represented in the form of $\frac{1}{4}x^T Q x$ and (7) also leads to the optimization problem in (3).

For example, if the PSF is a 2 by 2 matrix $\frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, according to (5), we have

$$x_i^b = \frac{1}{4}(x_i + x_{i+u} + x_{i+1} + x_{i+1+u}).$$

We will use this PSF as an example to illustrate how to construct matrices Q and b. Up to constant terms, the computation of (6) leads to the following equation:

$$\frac{1}{2}\sum_i \frac{1}{16}(u_2 - u_1)^2(x_i x_{i+u} + x_i x_{i+1} + x_i x_{i+1+u} + x_{i+1}x_{i+u} + x_{i+1}x_{i+1+u} + x_{i+u}x_{i+1+u}) + \frac{1}{2}\sum_i \frac{1}{4}(u_2 - u_1)(u_2 + u_1 - 2g_i)(x_i + x_{i+u} + x_{i+1} + x_{i+1+u}) - \lambda \sum_{<i,j>} x_i x_j. \quad (10)$$

Equation (10) comprises three terms. In the first term, $\sum_i x_i x_{i+1}$ and $\sum_i x_{i+u}x_{i+1+u}$ are different only because $\sum_i x_{i+u}x_{i+1+u}$ starts the summation from index $i+u$, that is from the second column in the image; $\sum_i x_i x_{i+u}$ and $\sum_i x_{i+1}x_{i+1+u}$ are different only because $\sum_i x_{i+1}x_{i+1+u}$ starts the summation from index $i+1$, that is from the second line in the image. We construct the matrix Q of size uv by uv. With T defined as $(u_2 - u_1)^2$, for line i of Q, $Q(i, i+1)$ has the value $T/8$ if $i \le u$, or otherwise $2 \times T/8$, corresponding to $\sum_i x_i x_{i+1} + \sum_i x_{i+u}x_{i+1+u}$. $Q(i, i+u)$ has the value $T/8$ if entry with index i of $x \in \{-1,+1\}^n$ denotes a pixel in the first line of the image, or otherwise $2 \times T/8$, corresponding to $\sum_i x_i x_{i+u} + \sum_i x_{i+1}x_{i+1+u}$. $Q(i, i+1+u)$ has the value $T/8$ if entry with index i of $x \in \{-1,+1\}^n$ does not denote a pixel in the last line of the image, corresponding to $\sum_i x_i x_{i+1+u}$. Finally, $Q(i, i-1+u)$ has the value $T/8$ if entry with index i of $x \in \{-1,+1\}^n$ does not denote a pixel in the first line of the image, corresponding to $\sum_i x_{i+1}x_{i+u}$.

In the second term, $\sum_i (u_2 - u_1)(u_2 + u_1 - 2g_i)x_i$ is already in the form of $b^T x$; to obtain $\sum_i (u_2 - u_1)(u_2 + u_1 - 2g_i)x_{i+1}$, the blurred image g is shifted a line down to have $g_{(1,0)}$; to obtain $\sum_i (u_2 - u_1)(u_2 + u_1 - 2g_i)x_{i+u}$, the blurred image g is shifted a column right to have $g_{(0,1)}$; to obtain $\sum_i (u_2 - u_1)(u_2 + u_1 - 2g_i)x_{i+1+u}$, the blurred image g is shifted a line down and a column right to have $g_{(1,1)}$. Now the second term of (10) is rewritten as $\frac{1}{2}b^T x$ with

$$b_i = \frac{1}{4}(u_2 - u_1)(4u_2 + 4u_1 - 2g_i - 2g_{(1,0)i} - 2g_{(0,1)i} - 2g_{(1,1)i}).$$

The third term of (10) can also be represented in Q by adding value -2λ to $Q(i, i-1)$ and $Q(i, i+1)$, for every line i of Q.

3 Positive Semidefinite Relaxation

In this section, we first introduce the optimization approach to solve the problem presented in Section 2, then we will give a brief review of positive semidefinite (PSD) programming and the randomized-hyperplane technique.

3.1 PSD Relaxation

In fact, the objective function of (10) can be homogenized in the following way:

$$x^TQx + 2b^Tx = \begin{pmatrix} x \\ 1 \end{pmatrix}^T L \begin{pmatrix} x \\ 1 \end{pmatrix}, L = \begin{pmatrix} Q & b \\ b^T & 0 \end{pmatrix}. \tag{11}$$

In order to relax (11), the integer constraint is first replaced by its quadratic equivalence $x_i^2 - 1 = 0, i = 1, ..., n$. Denoting the Lagrangian multiplier variables with $y_i, i = 1, ..., n$, the Lagrangian of (11) reads

$$x^TLx - \sum_{i=1}^{n} y_i(x_i^2 - 1) = x^T(L - D(y))x + e^Ty,$$

in which $D(y)$ denotes the diagonal matrix with y_i as the diagonal values, and e is a vector of all entries being 1. This leads to the Lagrangian relaxation

$$\sup_y \inf_x x^T(L - D(y))x + e^Ty.$$

Removing the constraint on x, the inner minimization is finite-valued if and only if $L - D(y)$ is positive semidefinite. Hence, the relaxed problem can be described as

$$z_d := \sup_y e^Ty, \qquad L - D(y) \in S_+^n, \tag{12}$$

and (12) is a convex optimization problem. The set S_+^n is a cone which is also self-dual, so that it coincides with its dual cone $(S_+^n)^* = \{Y : X \bullet Y \geq 0, X \in S_+^n\}$, where $X \bullet Y$ stands for $trace(XY)$.

To obtain the dual problem to (12), the Lagrangian dual of (12) is derived. Choosing Lagrangian multiplier $X \in S_+^n$, similar reasoning as above yields:

$$z_d = \sup_y \inf_{x \in S_+^n} e^Ty + X \bullet (L - D(y))$$

$$\leq \inf_{x \in S_+^n} \sup_y e^Ty + X \bullet (L - D(y))$$

$$= \inf_{x \in S_+^n} \sup_y L \bullet X - D(y) \bullet (X - I).$$

Here, the inner maximization of the above relaxation is finite only if $D(X) = I$. Hence, the dual problem to (12) is obtained as

$$z_d := \inf_{x \in S_+^n} L \bullet X, \qquad D(X) = I, \tag{13}$$

which again is convex. This final semidefinite relaxation (13) can also be obtained intuitively in a direct way from (11) by writing the objective function as $\inf_x x^TLx = \inf_x L \bullet xx^T$. Note that matrix xx^T is positive semidefinite and has rank one. In (13), the rank one condition is dropped by replacing xx^T by an arbitrary matrix $X \in S_+^n$ and the constraints are lifted to the higher-dimensional space respectively.

3.2 PSD programming and randomized-hyperplane technique

We consider the problem of minimizing a linear function of a variable $\bar{x} \in R^m$ subject to a matrix inequality:

$$\begin{array}{ll} \text{minimize} & c^T\bar{x} \\ \text{subject to} & F(\bar{x}) > 0, \end{array} \tag{14}$$

where

$$F(\bar{x}) = F_0 + \sum_{i}^{m} \bar{x}_i F_i.$$

The problem data are the vector $c \in R^m$ and $m + 1$ symmetric matrices $F_0, \cdots, F_m \in R_{n \times n}$. The inequality in $F(\bar{x}) \geq 0$ means that $F(\bar{x})$ is positive semidefinite, *i.e.*, $z^TF(\bar{x})x > 0$ for all $z \in R^n$. The inequality $F(\bar{x}) \geq 0$ is called a linear matrix inequality and problem in (14) is a semidefinite program. A semidefinite program is a convex optimization problem since its objective and constraint are convex: if $F(\bar{x}) \geq 0$ and $F(\bar{y}) \geq 0$, then for all θ, $0 \leq \theta \leq 1$,

$$\theta F(\bar{x}) + (1 - \theta)F(\bar{y}) \geq 0.$$

Although the semidefinite program in (14) may appear quite specialized, it includes many important optimization problems such as the linear program (LP) and quadratic programming. Semidefinite programming can be regarded as an extension of linear programming where the componentwise inequalities between vectors are replaced by matrix inequalities, or, equivalently, the first orthant is replaced by the cone of positive semidefinite matrices. There are good reasons for studying PSD programming. First, positive semidefinite programming constraints arise in may practical applications. Secondly, PSD programming provides a unified way of studying the properties and the derived algorithms of the convex optimization problems. As mentioned earlier, many optimization problems can be incorporated in the category of PSD programming. And the most important reason is that PSD programming can be solved very efficiently, both in theory and in practice.

To compute the optimal solution X^* in (13) and y^* in (12), a wide range of iterative interior-point algorithms and corresponding solvers are available, such as the popular package SP [13], SDPA [14], CSDP [15], SDPHA [16] and SDPT3 [17]. The optimization solver SeDuMi [18] is used in our experiments. SeDuMi is chosen because it possesses some beneficial features including taking full advantage of sparsity which leads to significant speed benefits, a theoretically proven worst-case iteration bounds, promoting sparsity by handling dense columns separately, *et al.* In SeDuMi, X^* and y^* are obtained by the call

$$[X^*, y^*, info] = \text{sedumi}(\bar{A}, \bar{b}, \bar{c}, K),$$

where $\bar{A}$, $\bar{b}$ and $\bar{c}$ contain problem data, and K is a cone. The call solves the optimization problem

$$\begin{aligned} \text{minimize} \quad & \bar{c}^T X^* \\ \text{subject to} \quad & \bar{A}X^* = \bar{b}, \\ & X^* \in K. \end{aligned} \tag{15}$$

Comparing (15) with (13), $\bar{A}$, $\bar{b}$, $\bar{c}$ and K can be defined, and X^* can thus be computed.

Based on the solution matrix X^* to the convex optimization problem (13), a combinatorial solution x to the original problem (11) can be found by using the randomized-hyperplane technique proposed by Goemans and Williamson [19]. Since $X^* \in S^n_+$, X^* can be decomposed as $V^T V, V = (v_1, ..., v_n)$ using Cholesky factorization. From the constraint $D(X) = I$, it follows that $\|v_i\| = 1, i = 1, ..., n$ and hence each primitive x_i is associated with a vector v_i on the unit sphere in a high-dimensional space. Accordingly, the matrix entries $(xx^T)_{ij} = x_i x_j$ are replaced by the matrix entries $X_{ij} = v_i^T v_j$. Choosing a random vector r from the unit sphere, a combinatorial solution vector x is calculated from $X^* = V^T V$ by setting $x_i = 1$ if $v_i^T r \geq 0$, and $x_i = -1$ otherwise. This is done for multiple times for different random vectors. The final solution, x_{SDP}, is the one that yields the minimum value for the objective function $x^T Lx$. This technique can be interpreted as selecting different hyperplanes through the origin, identified by their normal r, which partition the vectors $v_i, i = 1, ..., n$ into two sets.

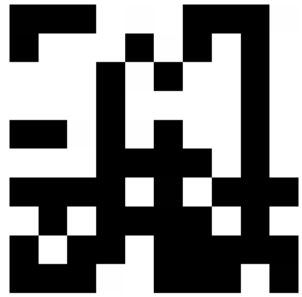

Figure 1: The true original binary image of size 10 by 10.

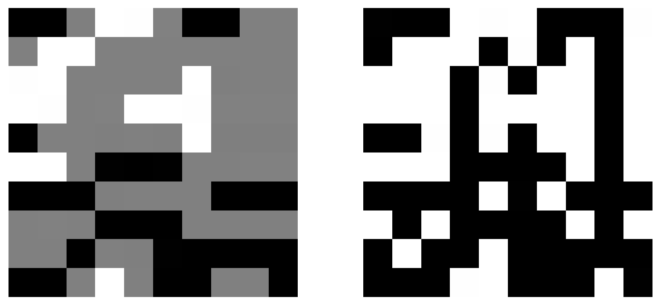

Figure 2: (Left) The image blurred by $h_{1\times2}$. (Right) The result of deconvolution.

4 Experimental Results

In Section 2, we have discussed the binary combinatorial optimization of binary image deconvolution. The functional to be minimized has been set up and the corresponding matrices constructed. In Section 3, we have described the relaxation of the original problem to PSD programming and the randomized-hyperplane technique for finding the solution. In this section, we apply the proposed optimization technique to blurred binary images, thereby validating the feasibility of the method. In our experiments, the true original binary image is an image extracted from the map of Iceland with a size of 10 by 10 shown in Figure 1.

First, we blur the original image with a PSF of $h_{1\times2} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$, and apply our method to restore the original image. The blurred image and the result of the deconvolution are shown in Figure 2. Note that in our approach, in the final optimization solution $x \in \{-1, +1\}^n$, vector entries with value 1 and -1 actually have the value 255 and 0 respectively. From this experiment, we can see that the accuracy of our method is 100%, thus demonstrating the deblurring capability of the proposed method.

Next, we add Gaussian white noise of mean 0 and vari-

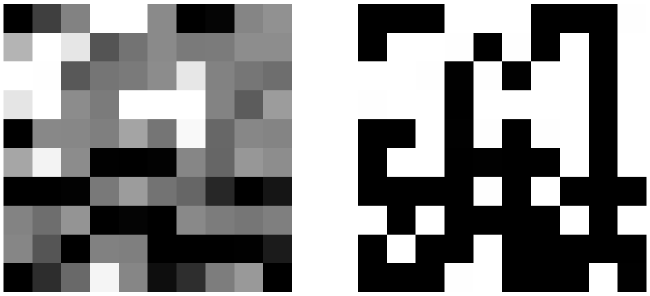

Figure 3: (Left) The image blurred by $h_{1\times 2}$ and with noise. (Right) The result of deconvolution.

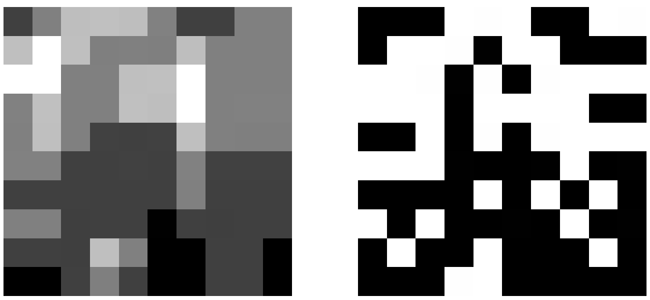

Figure 4: (Left) The image blurred by $h_{2\times 2}$. (Right) The result of deconvolution.

ance 0.01 to the blurred image in Figure 2 to check the noise fighting ability of the optimization method. From Figure 3, only 1 pixel value out of the 100 pixels is wrong, from this, we can see that the proposed method fights noise quite well. This demonstrates that in addition to deblurring, the proposed method simultaneously possesses good denoising capability.

Now, we extend the PSFs to 2 dimensional matrices. In Figure 4, the PSF is defined as $h_{2\times 2} = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. The accuracy of the proposed method is 88%. In Figure 5, the blurred image is added with the same noise as in Figure 3, and this time the proposed method shows its ability to fight noise, with the accuracy of 91%.

Figures 6 and 7 show the results of our method to deconvolve the blurred image when the PSF is defined as $h_{3\times 3} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and when the blurred image is added with the same noise as in Figure 3. The accuracy of the deconvolution when there is no noise is 77%, and the accuracy when noise is present is 75%.

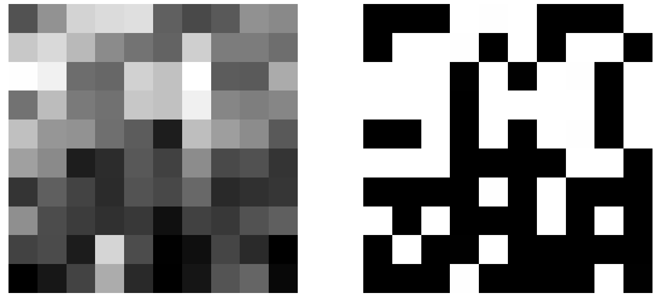

Figure 5: (Left) The image blurred by $h_{2\times 2}$ and with noise. (Right) The result of deconvolution.

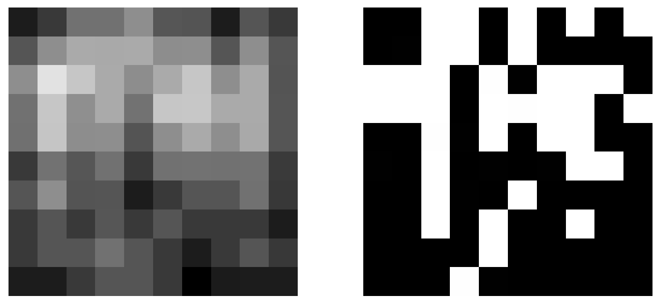

Figure 6: (Left) The image blurred by $h_{3\times 3}$. (Right) The result of deconvolution.

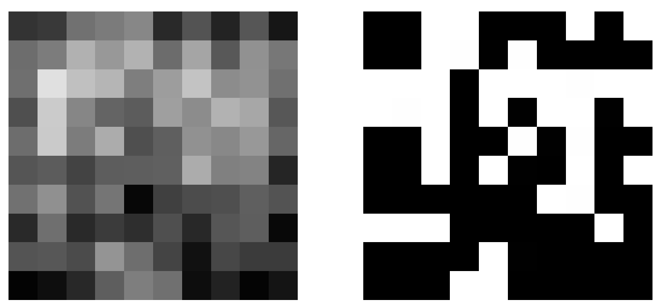

Figure 7: (Left) The image blurred by $h_{3\times 3}$ and with noise. (Right) The result of deconvolution.

Table 1: CPU Time

Deconvolution Experiments	CPU time(s)
PSF size 1 by 2, no noise	23.2
PSF size 1 by 2, with noise	20.7
PSF size 2 by 2, no noise	21.0
PSF size 2 by 2, with noise	21.0
PSF size 3 by 3, no noise	22.1
PSF size 3 by 3, with noise	22.0

The experiments were conducted on a 3.2GHz PC with 512MB RAM. The CPU times of the experiments are given in table 1. It can be shown that in our experiments, when the true image is blurred by our selected 1 dimensional PSF, the proposed method restores the original image with almost 100% accuracy. When the size of the PSF starts to increase, the accuracy drops to around 80% when the size reaches 3 by 3. It should be noted that, even when noise is present in addition to blurring, the proposed method continues to restore the true binary image with satisfactory accuracy. It should also be noted that the price paid for the outstanding performance of the optimization approach is the computation complexity. The computational time grows exponentially with the number of variables such that problems with thousands of variables cannot be solved in practical time. At present moment, this limits the application of our approach to the restoration of large binary images.

5 Conclusions and Future Work

This paper has presented a novel image processing technique for solving the combinatorial problem of binary image deconvolution. The problem is relaxed to a positive semidefinite (PSD) relaxation, the suboptimal solution of the PSD relaxation is computed and an optimal solution to the original problem is obtained using randomized-hyperplane technique. The proposed optimization method is accurate, anti-noise and simple without introducing new parameters. Numerical examples have demonstrated the excellent performance of the proposed method in simultaneously carrying out deblurring and denoising functions.

One way to speed up restoration process of large binary images is to apply image segmentation. The degraded binary images can be segmented to smaller image blocks before applying the optimization approach to each block. An obvious problem is that without taking into account the pixels on the boundary lines of columns of the block, error inevitably increases because the computation of x_i^b requires its adjacent pixels within the range of the PSF matrix as illustrated in (5). As such, a tradeoff has to be made between the computation time and quality of the restored images. A hybrid scheme with overlapping lines and columns will be considered to restore large binary images in our future research work. Furthermore, future research efforts might also lead to blind binary image deconvolution wherein the PSF is unknown.

6 Acknowledgments

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Number HKU 7164/03E).

References

[1] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine* **14**(2), pp. 24–41, 1997.

[2] J. Meloche and R.H.Zamar, "Binary image restoration," *The Canadian J. Statistics* **22**(3), pp. 335–355, 1994.

[3] D. Hitchcock and C. A. Glasbey, "Binary image restoration at subpixel resolution," *Biometrics* **53**, pp. 1010–1053, 1997.

[4] X. D. Gu, H. M. Wang, and D. H. Yu, "Binary image restoration using pulse coupled neural network," *8th Int. Conf. Neural Information Processing* , pp. 922–927, 2001.

[5] T. F. Chan, S. Esedoḡlu, and M. Nikolova, "Finding the global minimum for binary image restoration," *ICIP 2005* , pp. 121–124, 2005.

[6] J. Myrheim and H. Rue, "New algorithms for maximum entropy image restoration," *CVGIP: Graphical Models and Image Processing* **54**(3), pp. 223–238, 1992.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

[8] B. Radig and S. Florczyk, *Pattern Recognition*, Springer Berlin, 2001.

[9] J. Keuchel, C. Schellewald, D. Cremers, and C. Schnőrr, "Convex relaxations for binary image partitioning and perceptual grouping," *DAGM-Symposium 2001* , pp. 353–360, 2001.

[10] E. Y. Lam, "Blind deconvolution of bi-level images with successive filtering," in *20th Congress of the International Commission for Optics*, pp. 1228–1233, August 2005.

[11] E. Y. Lam, "Blind bi-level image restoration with iterated quadratic programming," *IEEE Transactions on Circuits and Systems II* , 2007.

[12] J. Keuchel, C. Schellewald, D. Cremers, and C. Schnőrr, "Binary partitioning, perceptual grouping, and restoration with semidefinite programming," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(11), pp. 1364–1379, 2003.

[13] L. Vandenberghe and S. Boyd, *SP: Software for semidefinite programming*, Information Systems Laboratory, Electrical Engineering Department, Stanford University, Stanford, USA, 1994.

[14] K. Fujisawa, M. Kojima, and K. Nakata, "SDPA (semidefinite programming algorithm) user's manual-version 4.10," tech. rep., Tokyo Institute of Technology, 1998.

[15] B. Borchers, "CSDP, a C library for semidefinite programming," tech. rep., New Mexico Techo Maths Faculty, 1997.

[16] N. Brixius and F. A. Potra, "SDPHA: A MATLAB implementation of homogeneous interior-point algorithms for semidefinite programming," tech. rep., Department of Computer Science, University of Iowa, 1998.

[17] K. C. Toh, M. J. Todd, and R. H. Tűtűncű, "SDPT3 - a MATLAB package for semidefinite programming," tech. rep., Cornell University, 1996.

[18] J.F.Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software* **11**(12), pp. 625–653, 1999.

[19] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J.ACM* **42**(6), pp. 1115–1145, 1995.