

ON ALGORITHMS FOR AUTOMATIC DEBLURRING FROM A SINGLE IMAGE*

Wei Wang

Department of Mathematics, Tongji University, Shanghai 200092, China

Email: weiwamng@163.com

Michael K. Ng

Centre for Mathematical Imaging and Vision and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong

Email: mng@math.hkbu.edu.hk

Abstract

In this paper, we study two variational blind deblurring models for a single image. The first model is to use the total variation prior in both image and blur, while the second model is to use the frame based prior in both image and blur. The main contribution of this paper is to show how to employ the generalized cross validation (GCV) method efficiently and automatically to estimate the two regularization parameters associated with the priors in these two blind motion deblurring models. Our experimental results show that the visual quality of restored images by the proposed method is very good, and they are competitive with the tested existing methods. We will also demonstrate the proposed method is also very efficient.

Mathematics subject classification: 65J22, 65K10, 68U10.

Key words: Blind deconvolution, Iterative methods, Total variation, Framelet, Generalized cross validation.

1. Introduction

The blurring of images often occurs from the motion of objects, unfocused cameras and calibration errors with imaging devices. Mathematically, the forward model of the blurring process is stated as follows:

$$f = p * g. \quad (1.1)$$

Here f is the observed image, g is the original image, p is the blur kernel which is also known as point spread function, $*$ represents the convolution operator. Recovering g from problem (1.1) with known p is called non-blind deconvolution problem which is a mathematically ill-posed problem. When p is also unknown, the problem is called blind deconvolution which is even more ill-posed. A survey and a book on blind image deconvolution can be found in [1] and [2] respectively.

In this paper, we consider blind motion deblurring problem. Motion blur appears when there is a relative motion between the camera and the scene during exposure. In the literature, there are several deblurring techniques by making use of information from multiple motion blurred images [3–8]. For single-image blind motion deblurring, some parametric models for the motion blur kernel are studied and considered in [9, 10]. In [11], Fergus et al. employed

* Received February 28, 2011 / Revised version received August 7, 2011 / Accepted August 15, 2011 /
Published online January 9, 2012 /

ensemble learning to recover a motion blur kernel with some image priors. In [12], Tai et al. introduced a new Projective Motion Blur Model that treats the blurred image as an integration of a clear scene under a sequence of projective transformations that describe the cameras path. In [13–16], researchers proposed some efficient and high-quality kernel estimation methods based on using different approaches. In [14], Jia formulated the kernel estimation as solving a Maximum a Posteriori (MAP) problem with the defined likelihood and prior on transparency. In [15], Shan et al. computed a deblurred image using a unified probabilistic model of both motion blur kernel estimation and unblurred image restoration. They developed a model of the spatial randomness of noise in the blurred image, as well a new local smoothness prior that reduces ringing artifacts by constraining contrast in the unblurred image wherever the blurred image exhibits low contrast. In [16], Xu et al. proposed an efficient and high-quality kernel estimation method based on using the spatial prior and the iterative support detection kernel refinement, which avoids hard threshold of the kernel elements to enforce sparsity. However, these algorithms are required to input some values of parameters so that motion blur kernels can be recovered and image details can be enhanced properly.

The main aim of this paper is to develop algorithms for automatic deblurring from a single image. We study two variational blind deblurring models for a single image restoration. The formulation is given as a minimization problem with some regularization terms on p and g :

$$\min_{p,g} E(p, g) \equiv \Phi(p * g - f) + \lambda_1 R_1(p) + \lambda_2 R_2(g), \quad (1.2)$$

where $\Phi(p * g - f)$ is the data fidelity term, $R_1(p)$ and $R_2(g)$ are the regularization terms for p and g respectively, and λ_1 and λ_2 are the two positive regularization parameters which are used to balance the data fidelity term and the two regularization terms. One of the useful regularization approaches is the total variation (TV) regularization method [17–20]. In this approach, the data fidelity term is usually l^2 norm for image intensity fitting; and the regularization terms are both measured by the total variation. According to the experimental results in [18, 21], we find that the use of total variation as a prior to general blur kernels may not be effective. However, it has been observed for blind motion or out-of-focus deblurring problems [18] that these blur kernels can be recovered and image details can be enhanced very well. Recently, another popular regularization method is to determine a sparse representation of image and blur under tight frame systems [22–24]. These methods are able to recover high-quality images from given blurred images.

According to (1.2), the regularization parameters must be determined properly so that deblurring algorithms can be used to provide good recovered images and blurs. In [18, 22–24], methods are not given for searching suitable regularization parameters. The main contribution of this paper is to show how to employ the generalized cross validation (GCV) method [25] efficiently and automatically to estimate the two regularization parameters λ_1 and λ_2 associated with the priors in blind motion deblurring models. In this paper, we consider two types of regularization terms and compare their performance. One type is to use the total variation for R_1 and R_2 . The other type is to employ tight frame systems for R_1 and R_2 . Our experimental results show that the visual quality of restored images by the proposed method is very good, and it is competitive with the tested existing methods. We will demonstrate the proposed method is also very efficient.

This paper is organized as follows. In Section 2, we present models using the total variation regularization and the frame-based regularization. In Section 3, we propose the two blind deblurring algorithms by employing the generalized cross validation (GCV) method to estimate

regularization parameters. In Section 4, we demonstrate the effectiveness of the proposed algorithms by experimental results. Finally, some concluding remarks are given in Section 5.

2. Regularization Models

2.1. The TV Regularization Model

Let us first discuss the total variation model:

$$\min_{p,g} E(p,g) \equiv \int_{\Omega} (p * g - f)^2 dx + \lambda_{1,1} \int_{\Omega} |Dp| + \lambda_{1,2} \int_{\Omega} |Dg|, \quad (2.1)$$

Here Ω denotes the image domain, and D represents the gradient operator. The data fidelity term is measured in L_2 -norm of $(p * g - f)$, and the two regularization terms are measured in L_1 -norm of the gradients of p and g respectively. In order to obtain a physically meaningful solution, we usually impose the following physical conditions:

$$\int p = 1, \quad p \geq 0 \quad \text{and} \quad l_g \leq g \leq u_g.$$

Here we assume the pixel value of restored image is in between l_g and u_g . For grey-level image, l_g and u_g are usually set to be 0 and 255 respectively. Now we formulate the basic procedure of our approach for solving problem (2.1) as follows: for $k = 0, 1, 2, \dots$

Algorithm 1.

Step 1: let $g^0 = f$ be the initial image, and set ϵ_g and ϵ_p to be the stopping criterion.

Step 2: the k th iteration:

- Given g^k , compute $p^{k+\frac{1}{4}}$ by solving:

$$\min_p E_p(p) \equiv \int_{\Omega} (p * g^k - f)^2 dx + \lambda_{1,1} \int_{\Omega} |Dp|, \quad (2.2)$$

update $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+\frac{1}{2}} = \max\{p^{k+\frac{1}{4}}, 0\},$$

normalize $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+1} = \frac{p^{k+\frac{1}{2}}}{\|p^{k+\frac{1}{2}}\|_1};$$

- Given p^{k+1} , compute $g^{k+\frac{1}{2}}$ by solving:

$$\min_g E_g(g) \equiv \int_{\Omega} (p^{k+1} * g - f)^2 dx + \lambda_{1,2} \int_{\Omega} |Dg|, \quad (2.3)$$

update g^{k+1} by setting:

$$g^{k+1} = \min\{\max\{g^{k+\frac{1}{2}}, l_g\}, u_g\}.$$

Step 3: Go back to Step 2 of Algorithm 1 until $\frac{\|g^{k+1} - g^k\|}{\|g^{k+1}\|} \leq \epsilon_g$ and $\frac{\|p^{k+1} - p^k\|}{\|p^{k+1}\|} \leq \epsilon_p$.

2.2. The Frame-based Regularization Model

In this subsection, we discuss the frame-based regularization model. A countable set of functions $X \in L^2(\mathbb{R})$ is called a tight frame of $L^2(\mathbb{R})$ if

$$f = \sum_{h \in X} \langle f, h \rangle h, \quad \forall f \in L^2(\mathbb{R}),$$

where $\langle \cdot, \cdot \rangle$ is the inner product of $L^2(\mathbb{R})$. The tight frame is a generalization of an orthonormal basis which allows more flexibility by sacrificing the orthonormality and the linear independence. Given a finite set of generating functions:

$$\Psi := \{\varphi_1, \dots, \varphi_r\} \subset L^2(\mathbb{R}).$$

If a countable set X defined by the dilations and the shifts of generators from Ψ is a tight frame of $L^2(\mathbb{R})$, $\varphi \in \Psi$ are called (tight) framelets. Here,

$$X := \{\varphi_{l,j,k} : 1 \leq l \leq r, j, k \in \mathbb{Z}\} \quad \text{and} \quad \varphi_{l,j,k} = 2^{\frac{j}{2}} \varphi_l(2^j x - k), \quad \varphi_l \in \Psi.$$

The construction of the generators which can generate a tight frame starts from a refinable function, see details in [31]. What we usually care about is how to convert the framelet decomposition and reconstruction to finite dimensional frames because of the finiteness of images dimensionality. Let \mathcal{A} be a $M \times N$ ($M \geq N$) matrix whose rows are vectors in \mathbb{R}^N . The set containing all the rows of \mathcal{A} is also denoted by \mathcal{A} . Then \mathcal{A} is a tight frame for \mathbb{R}^N if for any vector $x \in \mathbb{R}^N$,

$$x = \sum_{y \in \mathcal{A}} \langle x, y \rangle y.$$

Here $\langle \cdot, \cdot \rangle$ is the inner product of finite dimensional Euclidean spaces. The matrix \mathcal{A} is called the analysis (or decomposition) operator, and its adjoint \mathcal{A}^* is called the synthesis (or reconstruction) operator. The above equation is equivalent to the perfect reconstruction formula which can be written as $x = \mathcal{A}^* \mathcal{A} x$. Hence \mathcal{A} is a tight frame of \mathbb{R}^N if and only if $\mathcal{A}^* \mathcal{A} = \mathcal{I}$. We can derive \mathcal{A} from some fixed filters. Let h_0 be a low-pass filter, and h_1, \dots, h_r be high-pass filters. In this paper, we will use piece-wise linear framelets:

$$h_0 = \frac{1}{4}[1, 2, 1]; \quad h_1 = \frac{\sqrt{2}}{4}[1, 0, -1]; \quad h_2 = \frac{1}{4}[-1, 2, -1].$$

The 2D filters can be obtained by the tensor product of 1D filters:

$$\{H_l\} = \{h_{l_1} \otimes h_0, h_0 \otimes h_{l_2}, h_{l_1} \otimes h_{l_2}, 1 \leq l_1, l_2 \leq r\}.$$

For the 1D case, given a filter $\{h(j)\}_{j=-J}^{j=J}$, let $\mathcal{S}(h)$ be the convolution operator with filter h under the Neumann (symmetric) boundary condition:

$$\mathcal{S}(h) = \mathcal{T}(h) + \mathcal{H}(h),$$

where $\mathcal{T}(h)$ and $\mathcal{H}(h)$ are Toeplitz and Hankel matrices respectively, see [32] for details,

$$\mathcal{T}(h) = \begin{bmatrix} h(0) & \cdots & h(-J) & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h(J) & \ddots & \ddots & \ddots & h(-J) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & h(J) & \cdots & h(0) \end{bmatrix},$$

and

$$\mathcal{H}(h) = \begin{bmatrix} h(1) & h(2) & \cdots & h(J) & 0 \\ h(2) & \ddots & \ddots & \ddots & h(-J) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h(J) & \ddots & \ddots & \ddots & h(-2) \\ 0 & h(-J) & \cdots & h(-2) & h(-1) \end{bmatrix}.$$

In this paper, we use a multi-level tight frame system corresponding to the decomposition without down sampling. For a given filter $h = \{h(j)\}_{j=-J}^J$, the filters $h^{(l)}$ at level l corresponding to the decomposition without down sampling is given as follows:

$$h^{(l)} = \left\{ h(-J), \underbrace{0, \dots, 0}_{2^{l-1}-1}, h(-J+1), 0, \dots, 0, h(-1), \underbrace{0, \dots, 0}_{2^{l-1}-1}, h(0), \right. \\ \left. \underbrace{0, \dots, 0}_{2^{l-1}-1}, h(1), 0, \dots, 0, h(J-1), \underbrace{0, \dots, 0}_{2^{l-1}-1}, h(J) \right\}.$$

For given filters $\{h_i\}_{i=0}^r$, let $\mathcal{Z}_i^l = \mathcal{S}(h_i^l)$. Then the multi-level decomposition operator up to level L (without down sampling) is given by:

$$\mathcal{A} = \begin{bmatrix} \prod_{l=0}^{L-1} \mathcal{Z}_0^{L-l} \\ \mathcal{Z}_1^L \prod_{l=1}^{L-1} \mathcal{Z}_0^{L-l} \\ \vdots \\ \mathcal{Z}_r^L \prod_{l=1}^{L-1} \mathcal{Z}_0^{L-l} \\ \vdots \\ \mathcal{Z}_1^1 \\ \vdots \\ \mathcal{Z}_r^1 \end{bmatrix}.$$

If the filters $\{h_i\}_{i=0}^r$ satisfy the unitary extension principle [33], we have $\mathcal{A}^* \mathcal{A} = \mathcal{I}$. Then \mathcal{A} is a tight frame of \mathbb{R}^N . Based on the deblurring problem (1.2), we consider the following model:

$$\min_{p, g} E(p, g) \equiv \int_{\Omega} (p * g - f)^2 dx + \lambda_{2,1} \|\mathcal{A}p\|_1 + \lambda_{2,2} \|\mathcal{A}g\|_1. \quad (2.4)$$

We formulate the basic procedure of our approach for solving problem (2.4) as follows: for $k = 0, 1, 2, \dots$

Algorithm 2.

Step 1: let $g^0 = f$ be the initial image, and set ϵ_g and ϵ_p to be the stopping criterion.

Step 2: the k th iteration:

- Given g^k , compute $p^{k+\frac{1}{4}}$ by solving:

$$\min_p E_p(p) \equiv \int_{\Omega} (p * g^k - f)^2 dx + \lambda_{2,1} \| \mathcal{A}p \|_1, \quad (2.5)$$

update $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+\frac{1}{2}} = \max\{p^{k+\frac{1}{4}}, 0\},$$

normalize $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+1} = \frac{p^{k+\frac{1}{2}}}{\|p^{k+\frac{1}{2}}\|_1};$$

- Given p^{k+1} , compute $g^{k+\frac{1}{2}}$ by solving:

$$\min_g E_g(g) \equiv \int_{\Omega} (p^{k+1} * g - f)^2 dx + \lambda_{2,2} \| \mathcal{A}g \|_1, \quad (2.6)$$

update g^{k+1} by setting:

$$g^{k+1} = \min\{\max\{g^{k+\frac{1}{2}}, l_g\}, u_g\}.$$

Step 3: Go back to Step 2 of Algorithm 2 until $\frac{\|g^{k+1} - g^k\|}{\|g^{k+1}\|} \leq \epsilon_g$ and $\frac{\|p^{k+1} - p^k\|}{\|p^{k+1}\|} \leq \epsilon_p$.

We remark here the regularization functionals in (2.1) and (2.4) are not convex about (p, g) . However, the objective functionals in (2.2) and (2.5) are strictly convex about p , and the objective functionals in (2.3) and (2.6) are strictly convex about g because of the linearity of the blurring operator.

3. The Proposed Algorithms

3.1. The Algorithms

There are two computational issues in solving blind image restoration models in (2.1) and (2.4). The first issue is how to solve their associated subproblems in (2.2), (2.3), (2.5) and (2.6) efficiently. The second issue is how to estimate the regularization parameters $\lambda_{1,1}$, $\lambda_{1,2}$ and $\lambda_{2,1}$, $\lambda_{2,2}$ in (2.1) and (2.4). We remark that the optimization problems in (2.2), (2.3), (2.5) and (2.6) are not linear least squares problems, and the generalized cross validation cannot be applied directly to these optimization problems for estimation of regularization parameters.

Our main aim is to develop algorithms that can recover the blur and the original image, and to estimate the two regularization parameters automatically, i.e., given the observed image f , we can obtain the blur and the image without any prior knowledge.

For the optimization problems in (2.2) and (2.3), there are many effective and efficient methods for solving such total variation image deblurring problem, see [18,34,35]. In particular, we introduce two auxiliary variables \hat{p} and \hat{g} and an additional quadratic terms to minimize the

objective functionals in (2.2) and (2.3) respectively [35], i.e.,

$$\min_{p, \hat{p}} E_1(p, \hat{p}) \equiv \frac{1}{\lambda_{1,1}} \int_{\Omega} (p * g^k - f)^2 dx + \frac{\beta}{2} \int_{\Omega} (p - \hat{p})^2 dx + \int_{\Omega} |D\hat{p}|, \quad (3.1)$$

and

$$\min_{g, \hat{g}} E_2(g, \hat{g}) \equiv \frac{1}{\lambda_{1,2}} \int_{\Omega} (p^{k+1} * g - f)^2 dx + \frac{\beta}{2} \int_{\Omega} (g - \hat{g})^2 dx + \int_{\Omega} |D\hat{g}|, \quad (3.2)$$

Here note that as β increases, the solutions of problem (3.1) and (3.2) will get closer to those of problem (2.2) and (2.3). Therefore, the value of β will be increased iteration by iteration in practice so that the two minimization problems in (3.1) and (3.2) are close to those in (2.2) and (2.3) respectively. Algorithms 1A and 1B for solving problems in (3.1) and (3.2) are summarized as follows:

Algorithm 1A.

1. let \hat{p}^0 be the initial kernel, and set ϵ to be the stopping criterion.
2. the k th iteration:

- Given \hat{p}^k , compute $p^{k+\frac{1}{4}}$ by solving:

$$\min_p \left\{ \int_{\Omega} (p * g^k - f)^2 dx + \frac{\lambda_{1,1}\beta}{2} \int_{\Omega} (p - \hat{p}^k)^2 dx \right\}, \quad (3.3)$$

update $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+\frac{1}{2}} = \max\{p^{k+\frac{1}{4}}, 0\},$$

normalize $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+1} = \frac{p^{k+\frac{1}{2}}}{\|p^{k+\frac{1}{2}}\|_1};$$

- Given p^{k+1} , compute $\hat{p}^{k+\frac{1}{4}}$ by solving:

$$\min_{\hat{p}} \left\{ \frac{\beta}{2} \int_{\Omega} (\hat{p} - p^{k+1})^2 dx + \int_{\Omega} |D\hat{p}| \right\}, \quad (3.4)$$

update $\hat{p}^{k+\frac{1}{2}}$ by setting:

$$\hat{p}^{k+\frac{1}{2}} = \max\{\hat{p}^{k+\frac{1}{4}}, 0\},$$

normalize $\hat{p}^{k+\frac{1}{2}}$ by setting:

$$\hat{p}^{k+1} = \frac{\hat{p}^{k+\frac{1}{2}}}{\|\hat{p}^{k+\frac{1}{2}}\|_1},$$

increase β by using:

$$\beta = \theta \cdot \beta.$$

3. Go back to Step 2 of Algorithm 1A until $\frac{\|p^{k+1} - p^k\|}{\|p^{k+1}\|} \leq \epsilon$.

Algorithm 1B.

1. let $\hat{g}^0 = f$ be the initial image, and set ϵ to be the stopping criterion.

2. the k th iteration:

- Given \hat{g}^k , compute $g^{k+\frac{1}{2}}$ by solving:

$$\min_g \left\{ \int_{\Omega} (p^{k+1} * g - f)^2 dx + \frac{\lambda_{1,2}\beta}{2} \int_{\Omega} (g - \hat{g}^k)^2 dx \right\}, \quad (3.5)$$

update g^{k+1} by setting:

$$g^{k+1} = \min\{\max\{g^{k+\frac{1}{2}}, l_g\}, u_g\};$$

- Given g^{k+1} , compute $\hat{g}^{k+\frac{1}{2}}$ by solving:

$$\min_{\hat{g}} \left\{ \frac{\beta}{2} \int_{\Omega} (\hat{g} - g^{k+1})^2 dx + \int_{\Omega} |D\hat{g}| \right\}, \quad (3.6)$$

update \hat{g}^{k+1} by setting:

$$\hat{g}^{k+1} = \min\{\max\{\hat{g}^{k+\frac{1}{2}}, l_g\}, u_g\},$$

increase β by using:

$$\beta = \theta \cdot \beta.$$

3. Go back to Step 2 of Algorithm 1B until $\frac{\|g^{k+1} - g^k\|}{\|g^{k+1}\|} \leq \epsilon$.

For the subproblems in (3.3) and (3.5), it can be solved by using fast Fourier transforms, see [35]. For the subproblems in (3.4) and (3.6), it can be solved by the projection algorithm [36, 37] or the Split Bregman algorithm [38]. It is interesting to note that (3.3) and (3.5) are linear least squares problems with Tikhonov regularization, and therefore the associated regularization parameters $\lambda_{1,1}\beta/2$ and $\lambda_{1,2}\beta/2$ can be estimated by the generalized cross validation technique which will be discussed in next Section. We propose the following computational procedures in Algorithms 1A and 1B. We first estimate the regularization parameters and then solve the corresponding linear regularized least squares problems.

By using the same approach, we use two auxiliary variables \hat{p} and \hat{g} and additional quadratic terms to minimize the objective functionals in (2.5) and (2.6) respectively, i.e.,

$$\min_{p, \hat{p}} E_3(p, \hat{p}) \equiv \frac{1}{\lambda_{2,1}} \int_{\Omega} (p * g^k - f)^2 dx + \frac{\beta}{2} \int_{\Omega} (p - \hat{p})^2 dx + \|\mathcal{A}\hat{p}\|_1, \quad (3.7)$$

and

$$\min_{g, \hat{g}} E_4(g, \hat{g}) \equiv \frac{1}{\lambda_{2,2}} \int_{\Omega} (p^{k+1} * g - f)^2 dx + \frac{\beta}{2} \int_{\Omega} (g - \hat{g})^2 dx + \|\mathcal{A}\hat{g}\|_1, \quad (3.8)$$

Here the value of β will also be increased iteration by iteration so that the two minimization problems in (3.7) and (3.8) are close to those in (2.5) and (2.6) respectively. Algorithms 2A and 2B for solving problems in (3.7) and (3.8) are summarized as follows:

Algorithm 2A.

1. let \hat{p}^0 be the initial kernel, and set ϵ to be the stopping criterion.
2. the k th iteration:

- Given \hat{p}^k , compute $p^{k+\frac{1}{4}}$ by solving:

$$\min_p \left\{ \int_{\Omega} (p * g^k - f)^2 dx + \frac{\lambda_{2,1}\beta}{2} \int_{\Omega} (p - \hat{p}^k)^2 dx \right\}, \quad (3.9)$$

update $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+\frac{1}{2}} = \max\{p^{k+\frac{1}{4}}, 0\},$$

normalize $p^{k+\frac{1}{2}}$ by setting:

$$p^{k+1} = \frac{p^{k+\frac{1}{2}}}{\|p^{k+\frac{1}{2}}\|_1};$$

- Given p^{k+1} , compute $\hat{p}^{k+\frac{1}{4}}$ by solving:

$$\min_{\hat{p}} \left\{ \frac{\beta}{2} \int_{\Omega} (\hat{p} - p^{k+1})^2 dx + \|\mathcal{A}\hat{p}\|_1 \right\}, \quad (3.10)$$

update $\hat{p}^{k+\frac{1}{2}}$ by setting:

$$\hat{p}^{k+\frac{1}{2}} = \max\{\hat{p}^{k+\frac{1}{4}}, 0\},$$

normalize $\hat{p}^{k+\frac{1}{2}}$ by setting:

$$\hat{p}^{k+1} = \frac{\hat{p}^{k+\frac{1}{2}}}{\|\hat{p}^{k+\frac{1}{2}}\|_1},$$

increase β by using:

$$\beta = \theta \cdot \beta.$$

3. Go back to Step 2 of Algorithm 2A until $\frac{\|p^{k+1} - p^k\|}{\|p^{k+1}\|} \leq \epsilon$.

Algorithm 2B.

1. let $\hat{g}^0 = f$ be the initial image, and set ϵ to be the stopping criterion.
2. the k th iteration:

- Given \hat{g}^k , compute $g^{k+\frac{1}{2}}$ by solving:

$$\min_g \left\{ \int_{\Omega} (p^{k+1} * g - f)^2 dx + \frac{\lambda_{2,2}\beta}{2} \int_{\Omega} (g - \hat{g}^k)^2 dx \right\}, \quad (3.11)$$

update g^{k+1} by setting:

$$g^{k+1} = \min\{\max\{g^{k+\frac{1}{2}}, l_g\}, u_g\};$$

- Given g^{k+1} , compute $\hat{g}^{k+\frac{1}{2}}$ by solving:

$$\min_{\hat{g}} \left\{ \frac{\beta}{2} \int_{\Omega} (\hat{g} - g^{k+1})^2 dx + \|\mathcal{A}\hat{g}\|_1 \right\}, \quad (3.12)$$

update \hat{g}^{k+1} by setting:

$$\hat{g}^{k+1} = \min\{\max\{\hat{g}^{k+\frac{1}{2}}, l_g\}, u_g\},$$

increase β by setting:

$$\beta = \theta \cdot \beta.$$

3. Go back to Step 2 of Algorithm 2B until $\frac{\|g^{k+1} - g^k\|}{\|g^{k+1}\|} \leq \epsilon$.

Similarly, for the subproblems in (3.9) and (3.11), it can be solved by using fast Fourier transforms. For the subproblems in (3.10) and (3.12), it can be solved by the Linearized Bregman algorithm [39, 40]. Both regularization parameters are estimated by the generalized cross validation method with respect to each iteration.

3.2. Estimation of regularization parameters

The method of regularization is used to achieve stability for deblurring problems. Recall the classical Tikhonov regularization [26], a regularization operator \mathcal{D} is added to restrict the set of admissible solutions. Then the regularized solution $g(\lambda)$ is computed as follows:

$$\min_{g(\lambda)} \left\{ \lambda \|\mathcal{D}g(\lambda)\|_2^2 + \|f - \mathcal{H}g(\lambda)\|_2^2 \right\}. \quad (3.13)$$

The parameter λ controls the regularity of the solution. One can solve the problem in (3.13) by the following equation:

$$(\lambda \mathcal{D}^t \mathcal{D} + \mathcal{H}^t \mathcal{H})g(\lambda) = \mathcal{H}^t f.$$

For different choices of \mathcal{D} with some appropriate boundary conditions, one can find some fast algorithms to deal with the above equation, see [27–30] for detailed discussion.

Another computational issue in regularization is the choice of λ . Generalized cross-validation [25] is a method that estimates λ without requiring an estimate of the noise variance. It is based on the concept of prediction errors. For each $k = 1, \dots, n$, let g_{λ}^k be the vector that minimizes the error measure:

$$\sum_{i=1, i \neq k}^n ([f]_i - [\mathcal{H}g(\lambda)]_i)^2 + \lambda \|\mathcal{D}g(\lambda)\|_2^2,$$

where n is the size of the vector f , $[\mathcal{H}g]_i$ is the i th element of $\mathcal{H}g$ and $[f]_i$ is the i th element of f . If λ is such that g_{λ}^k is a good estimate of g , then $[\mathcal{H}g_{\lambda}^k]_k$ should be a good approximation of $[f]_k$ on average. For a given λ , the average squared error between the predicted value $[\mathcal{H}g_{\lambda}^k]_k$ and the actual value $[f]_k$ is given by:

$$\frac{1}{n} \sum_{k=1}^n \left([f]_k - [\mathcal{H}g_{\lambda}^k]_k \right)^2.$$

The generalized cross-validation (GCV) is a weighted version of the above error:

$$v(\lambda) = \frac{1}{n} \sum_{k=1}^n \left([f]_k - [\mathcal{H}g\lambda^k]_k \right)^2 \left[\frac{1 - m_{kk}(\lambda)}{1 - \frac{1}{n} \sum_{j=1}^n m_{jj}(\lambda)} \right]^2,$$

where $m_{jj}(\lambda)$ is the (j, j) th entry of the so-called influence matrix:

$$\mathcal{M}(\lambda) = \mathcal{H}(\mathcal{H}^t\mathcal{H} + \lambda\mathcal{D}^t\mathcal{D})^{-1}\mathcal{H}^t.$$

In [25], one can find that $v(\lambda)$ can be written as:

$$v(\lambda) = n \frac{\|[\mathcal{I} - \mathcal{M}(\lambda)]f\|_2^2}{\text{tr}[\mathcal{I} - \mathcal{M}(\lambda)]^2}.$$

Then the optimal regularization parameter is chosen to minimize $v(\lambda)$. If the periodic boundary condition or the Neumann boundary condition is used for both \mathcal{H} and $\mathcal{D}^t\mathcal{D}$, we can rewrite $v(\lambda)$ as follows:

$$v(\lambda) = n \frac{\sum_{i=1}^n \left[\frac{\lambda\beta_i}{\alpha_i^2 + \lambda\beta_i} \right]^2 [\mathcal{F}f]_i^2}{\left[\sum_{i=1}^n \frac{\lambda\beta_i}{\alpha_i^2 + \lambda\beta_i} \right]^2},$$

where \mathcal{F} represents either the discrete Fourier matrix or the discrete cosine transform matrix, and α_i and β_i represent the eigenvalues of \mathcal{H} and $\mathcal{D}^t\mathcal{D}$ respectively. We also recall that α_i and β_i can be obtained by fast Fourier transforms with periodic boundary condition or by fast cosine transforms with Neumann boundary condition. Then we use the GCV method to estimate the regularization parameters $\lambda_{1,1}\beta/2$, $\lambda_{1,2}\beta/2$, $\lambda_{2,1}\beta/2$, and $\lambda_{2,2}\beta/2$ in (3.3), (3.5), (3.9) and (3.11). For example, we rewrite (3.3) as follows:

$$\min_{\tilde{p}} \left\{ \frac{\lambda_{1,1}\beta}{2} \int_{\Omega} \tilde{p}^2 dx + \int_{\Omega} (g^k * \tilde{p} + g^k * \hat{p}^k - f)^2 dx \right\},$$

where $\tilde{p} = p - \hat{p}^k$. Compare the above problem with the initial GCV functional (3.13), one can find that $D = I$ is the identity matrix, and if we denote G^k as the matrix form of the convolution operator by the kernel g^k , then the eigenvalues of G^k can be obtained by fast Fourier transforms with periodic boundary condition or by fast cosine transforms with Neumann boundary condition. Then the the GCV function of $\mu_1 = \frac{\lambda_{1,1}\beta}{2}$ in (3.3) is given by:

$$v(\mu_1) = n \frac{\sum_{i=1}^n \left[\frac{\mu_1}{\alpha_i^2 + \mu_1} \right]^2 [\mathcal{F}(f - G^k\hat{p}^k)]_i^2}{\left[\sum_{i=1}^n \frac{\mu_1}{\alpha_i^2 + \mu_1} \right]^2},$$

where α_i is the eigenvalues of G^k . The optimal regularization parameters in (3.5), (3.9) and (3.11) can be obtained by similar approach.

In the next section, we demonstrate that the visual quality of restored images by the proposed method is very good and the two proposed methods are quite efficient.

4. Experimental Results

In this section, we test the two proposed algorithms on several images with different kinds of motion or out-of-focus blurs. We use peak signal to noise ratio (PSNR) to measure the quality of restored images. It is defined as follows:

$$\text{MSE} = \frac{1}{N} \|g_c - g\|_2^2,$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right),$$

where MSE means the mean squares error, g is the original image, g_c is the deblurred image, and N is the size of the image. For a color image, we compute MSE as $0.3R+0.59G+0.11B$, here R, G, and B represent the MSE of red, green and blue channels.

In all the experiments, we use an observed image as an initial image and an δ function as an initial point spread function. The lower and upper bounds of pixel values are set to be $l_g = 0$ and $u_g = 255$. The initial value of β is set to be 2 and the value of β at each outer iteration is increased by a factor of $\theta = 2$ so that β becomes larger and larger, and this force the optimization problems in (3.1), (3.2), (3.7) and (3.8) are close to (2.2), (2.3), (2.5), and (2.6) respectively. For the stopping criteria, we set ϵ_g and ϵ_p to be 1×10^{-3} in Algorithm 1 and Algorithm 2 and ϵ to be 1×10^{-1} in Algorithm 1A, 1B, 2A and 2B for the tolerance of the relative difference between two successive iterates. The two proposed algorithms are written in MATLAB, and the maximum iterations of Algorithm 1 and Algorithm 2 are set to 100. We remark here all the original blurred images are noise free in the experiments.

Table 4.1: The PSNRs for different θ .

θ	1.2	2	3	4	5
PSNR for Alg. 1(dB)	24.6763	24.6763	24.6763	24.6763	24.6763
PSNR for Alg. 2(dB)	26.5092	26.3900	26.3135	26.4655	26.2783
θ	6	7	8	9	10
PSNR for Alg. 1(dB)	24.6763	24.6763	24.6763	24.6763	24.6763
PSNR for Alg. 2(dB)	26.4890	26.3109	26.4698	26.3396	26.2968

Table 4.2: The PSNRs for different β .

β	1.2	2	4	8	16
PSNR for Alg. 1(dB)	24.6763	24.6763	24.6763	24.6763	24.6763
PSNR for Alg. 2(dB)	26.3187	26.3900	26.3375	26.3167	26.3335
β	32	64	128	256	512
PSNR for Alg. 1(dB)	24.6763	24.6763	24.6763	24.6763	24.6763
PSNR for Alg. 2(dB)	26.4554	26.5633	26.3418	26.4234	26.4092

4.1. Robustness to Parameters

In Table 1 and Table 2, we test the effect of two parameters β and θ in Algorithm 1 and Algorithm 2. In Table 1, we fix the value of β to be 2 and check the PSNRs of restored images for different initial values of $\theta = 1.2, 2, 3, 4, 5, 6, 7, 8, 9, 10$. Here the initial image is blurred by linear motion blurs of length 15 pixels, See Fig. 4.1(b). We also show the restored

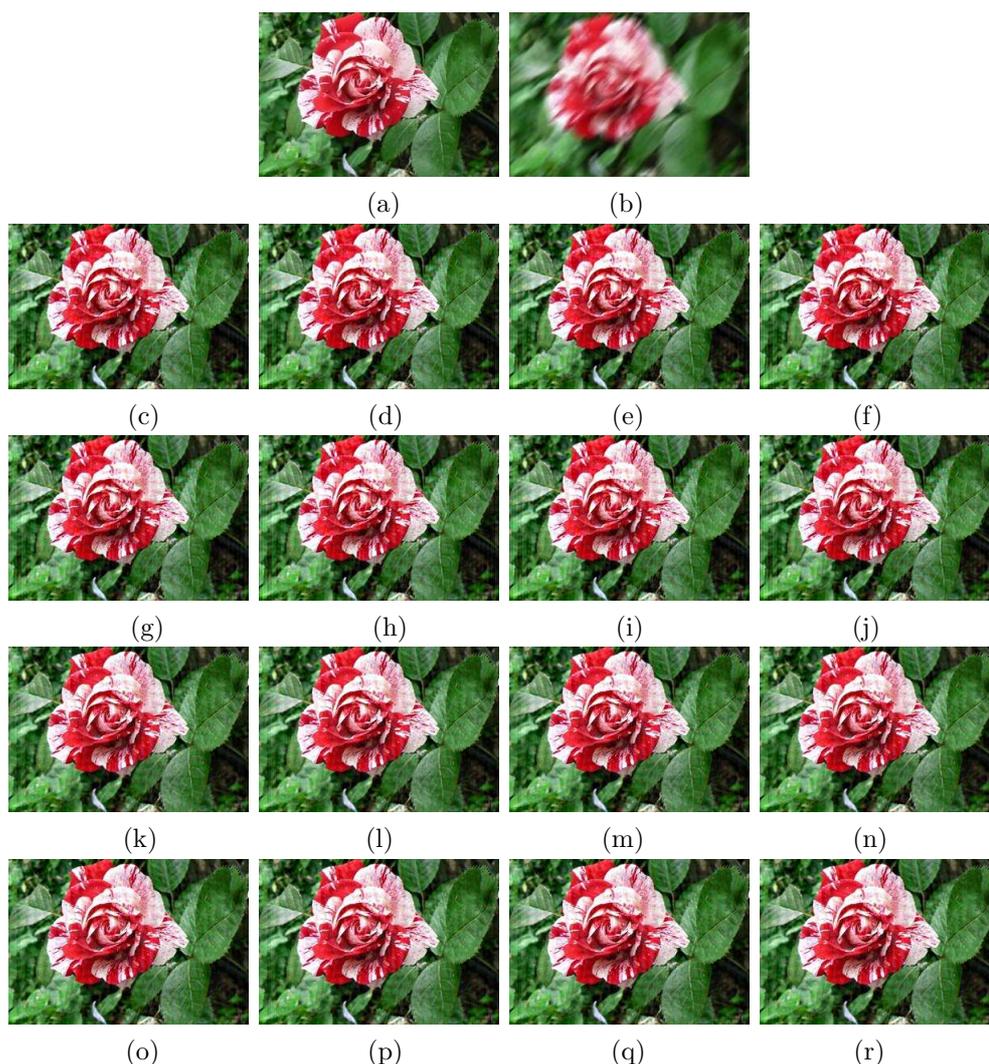


Fig. 4.1. (a) original image; (b) motion blurred image (len = 15, theta = 45); (c)-(f) deblurred images by Algorithm 1 with $\beta = 2$ and $\theta = 1.2, 4, 7, 10$; (g)-(j) deblurred image by Algorithm 1 with $\theta = 2$ and $\beta = 1.2, 8, 64, 512$; (k)-(n) deblurred images by Algorithm 2 with $\beta = 2$ and $\theta = 1.2, 4, 7, 10$; (o)-(r) deblurred image by Algorithm 1 with $\theta = 2$ and $\beta = 1.2, 8, 64, 512$.

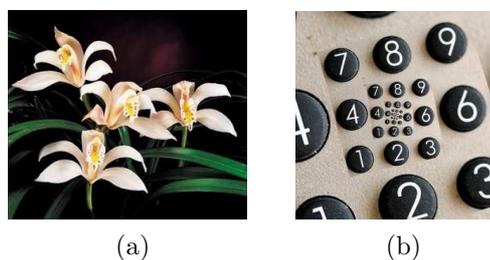


Fig. 4.2. original images.

images corresponding to $\theta = 1.2, 4, 7, 10$ in Fig. 4.1(c)-(f) for Algorithm 1 and Fig. 4.1(k)-(n) for Algorithm 2. We see that the PSNRs for different θ in Algorithm 1 are always 24.6763 dB, and the variation of PSNR for Algorithm 2 is within 0.2309 dB. Meanwhile, we see from

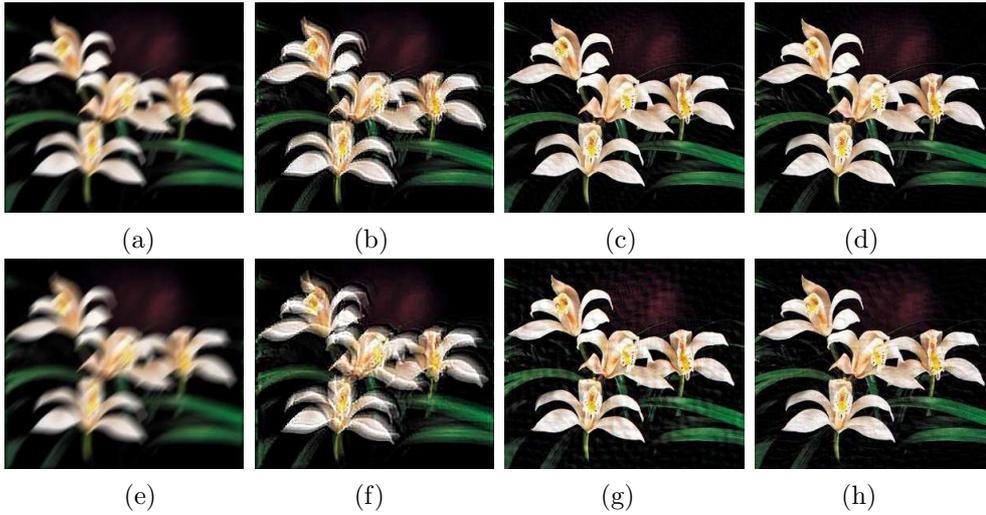


Fig. 4.3. (a) motion blurred image (len = 9, theta = 45); (b) deblurred image by the algorithm in [16] (PSNR = 18.95dB); (c) deblurred image by Algorithm 1 (PSNR = 31.93dB); (d) deblurred image by Algorithm 2 (PSNR = 31.94dB); (e) motion blurred image (len = 15, theta = 45); (f) deblurred image by the algorithm in [16] (PSNR = 17.43dB); (g) deblurred image by Algorithm 1 (PSNR = 28.06dB); (h) deblurred image by Algorithm 2 (PSNR = 29.02dB).

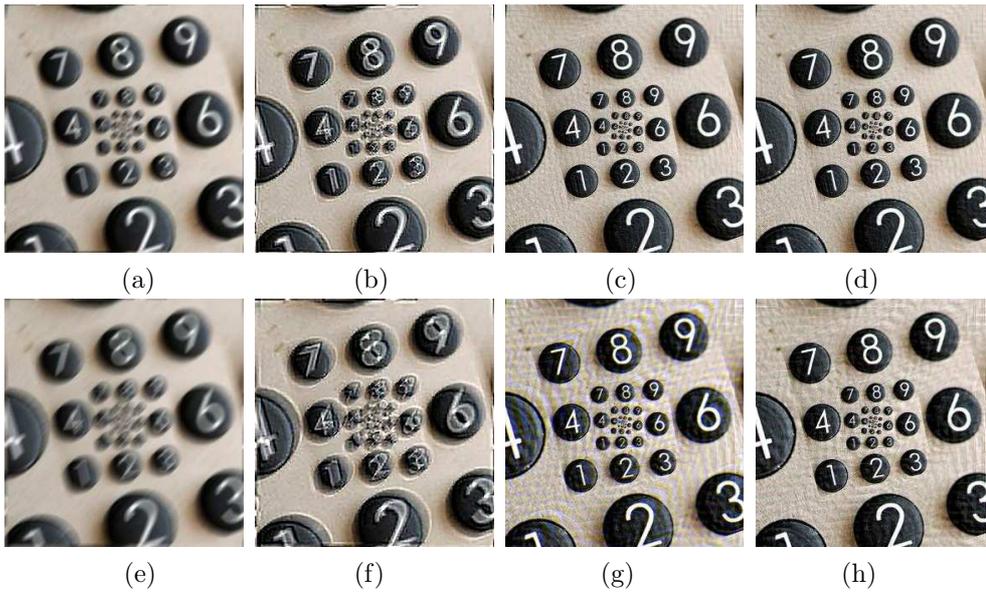


Fig. 4.4. (a) motion blurred image (len = 9, theta = 45); (b) deblurred image by the algorithm in [16] (PSNR = 16.91dB); (c) deblurred image by Algorithm 1 (PSNR = 28.31dB); (d) deblurred image by Algorithm 2 (PSNR = 28.32dB); (e) motion blurred image (len = 15, theta = 45); (f) deblurred image by the algorithm in [16] (PSNR = 15.25dB); (g) deblurred image by Algorithm 1 (PSNR = 23.33dB); (h) deblurred image by Algorithm 2 (PSNR = 25.20dB).

Fig. 4.1(c)-(j) that the restored images change very little visually for different choice of θ and β by Algorithm 1.

In Table 2, we fix the value of θ to be 2 and check the PSNRs of restored images for different initial values of $\beta = 1.2, 2, 4, 8, 16, 32, 64, 128, 256, 512$. Here the initial image is the same, see

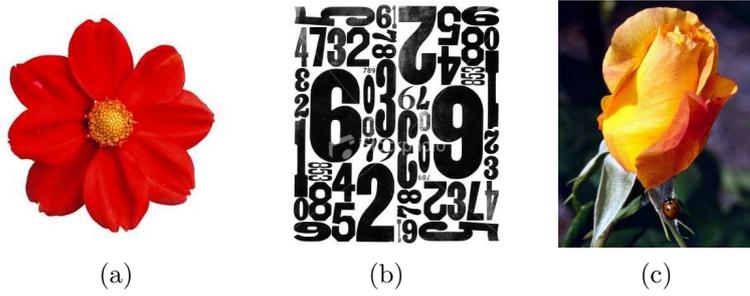


Fig. 4.5. original images.

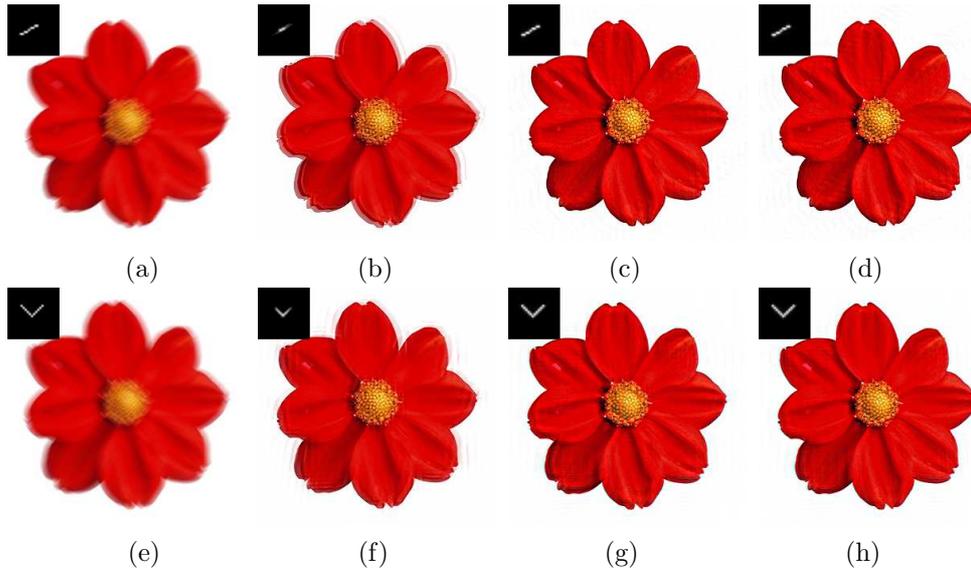


Fig. 4.6. (a) and (e) motion blurred images with the blur kernels are shown on the top left of the corresponding images; (b) and (f) deblurred images by the algorithm in [16] with the estimated kernels; (c) and (g) deblurred images by Algorithm 1 with the estimated kernels; (d) and (h) deblurred images by Algorithm 2 with the estimated kernels.

Fig. 4.1(b). We still show the restored images corresponding to $\beta = 1.2, 8, 64, 512$ in Fig. 4.1(g)-(j) for Algorithm 1 and Fig. 4.1(o)-(r) for Algorithm 2. We see that the PSNRs for different β in Algorithm 1 are always 24.6763 dB, and the variation of PSNR for Algorithm 2 is within 0.2466 dB. Meanwhile, we see from Fig. 4.1(k)-(r) that the restored images also change very little visually for different choice of θ and β by Algorithm 2. Therefore, we can say that the proposed algorithms are very robust for the two parameters θ and β .

4.2. PSNRs and motion blur kernels

In this section, we give some examples using Algorithm 1 and Algorithm 2 to deal with the motion deblurring problem. we compare the two proposed algorithms with the recent algorithm proposed in [16]. All the parameters of the algorithm in [16] use the default values. In the first experiment, two images are used to test the proposed two algorithms. These images are shown in Fig. 4.2. The original images are blurred by linear motion blurs of length 9 and 15 pixels. The resulting blurred images are shown in Figs. 4.3 - 4.4(a) and (e). We display the restored images by the the algorithm in [16], the total variation based algorithm, and the frame-based algorithm

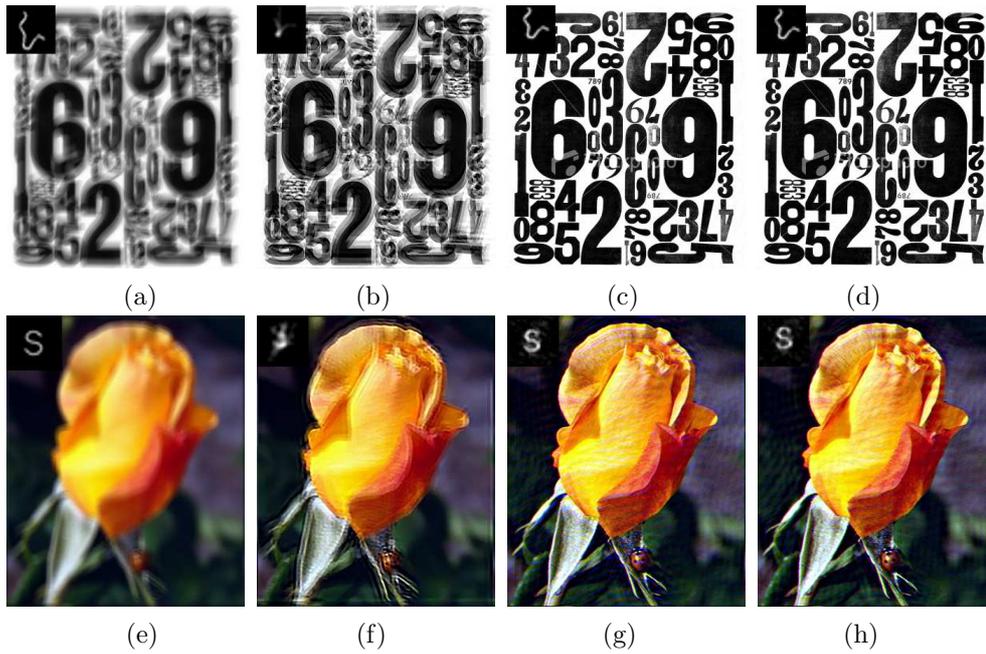


Fig. 4.7. (a) and (e) input observed images; (b) and (f) deblurred images by the algorithm in [16] with the estimated kernels; (c) and (g) deblurred images by Algorithm 1 with the estimated kernels; (d) and (h) deblurred images by Algorithm 2 with the estimated kernels.

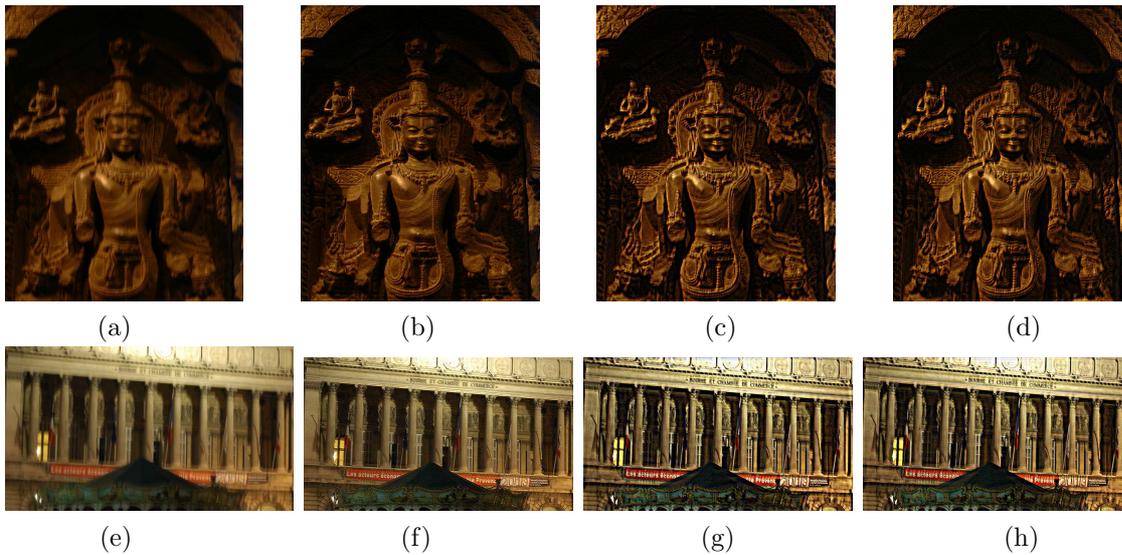


Fig. 4.8. (a) and (e) input observed images; (b) and (f) deblurred images by the algorithm in [16]; (c) and (g) deblurred images by Algorithm 1; (d) and (h) deblurred images by Algorithm 2.

in Figs. 4.3 - 4.4(b) and (f), Figs. 4.3 - 4.4(c) and (g), and Figs. 4.3 - 4.4(d) and (h) respectively. Meanwhile, we remark that the iterations required for convergence to obtain Figs. 4.3(c) and (d) by using Algorithm 1 and Algorithm 2 are 56 and 65 respectively. Each iterations in Algorithm 1 and Algorithm 2 take 1.48 and 8.71 seconds respectively on a windows PC with an Intel 2.66 GHz CPU for the blurred image in Fig. 4.2(a) of the size 300×265 . Because the frame-based algorithm requires to process different frames in the matrix \mathcal{A} , more computational times are

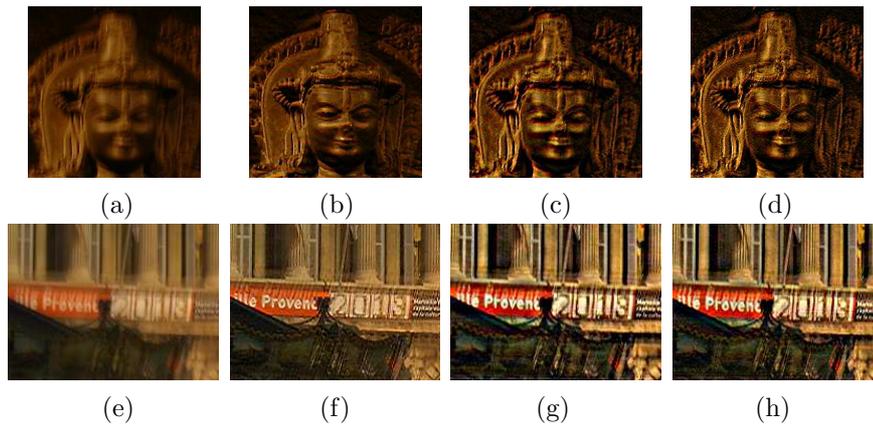


Fig. 4.9. The zooming parts of Fig. 4.8(a)-(h).

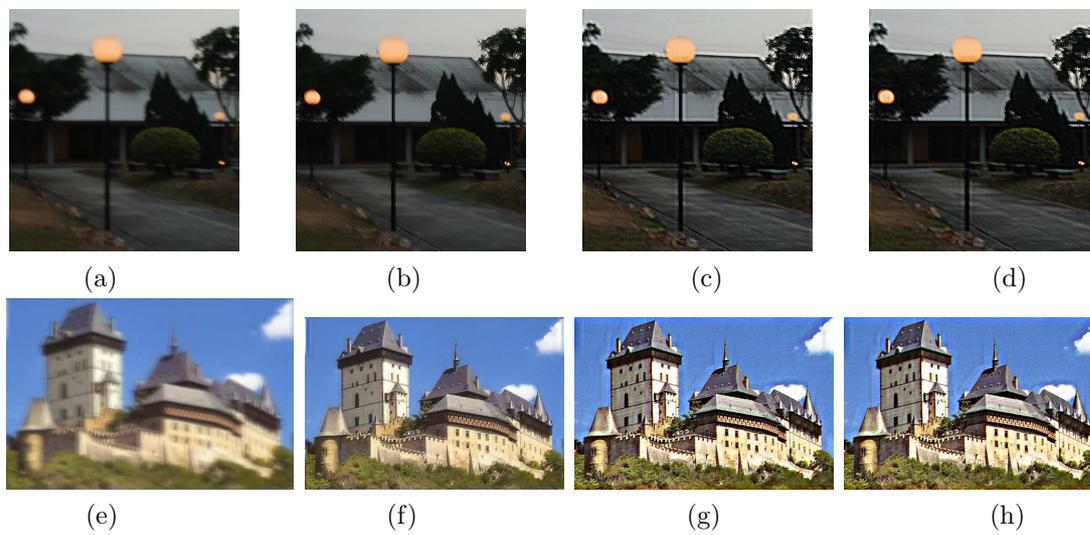


Fig. 4.10. (a) and (e) input observed images; (b) and (f) deblurred images by the algorithm in [16]; (c) and (g) deblurred images by Algorithm 1; (d) and (h) deblurred images by Algorithm 2.

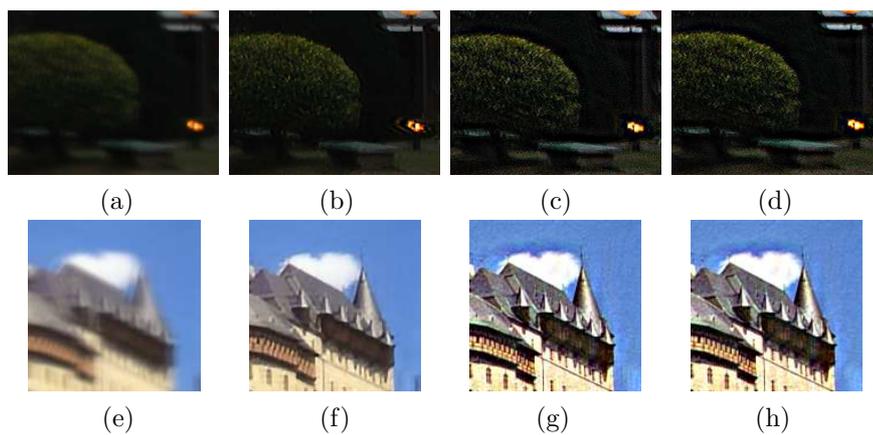


Fig. 4.11. The zooming parts of Fig. 4.10(a)-(h).



Fig. 4.12. The input out-of-focus images.

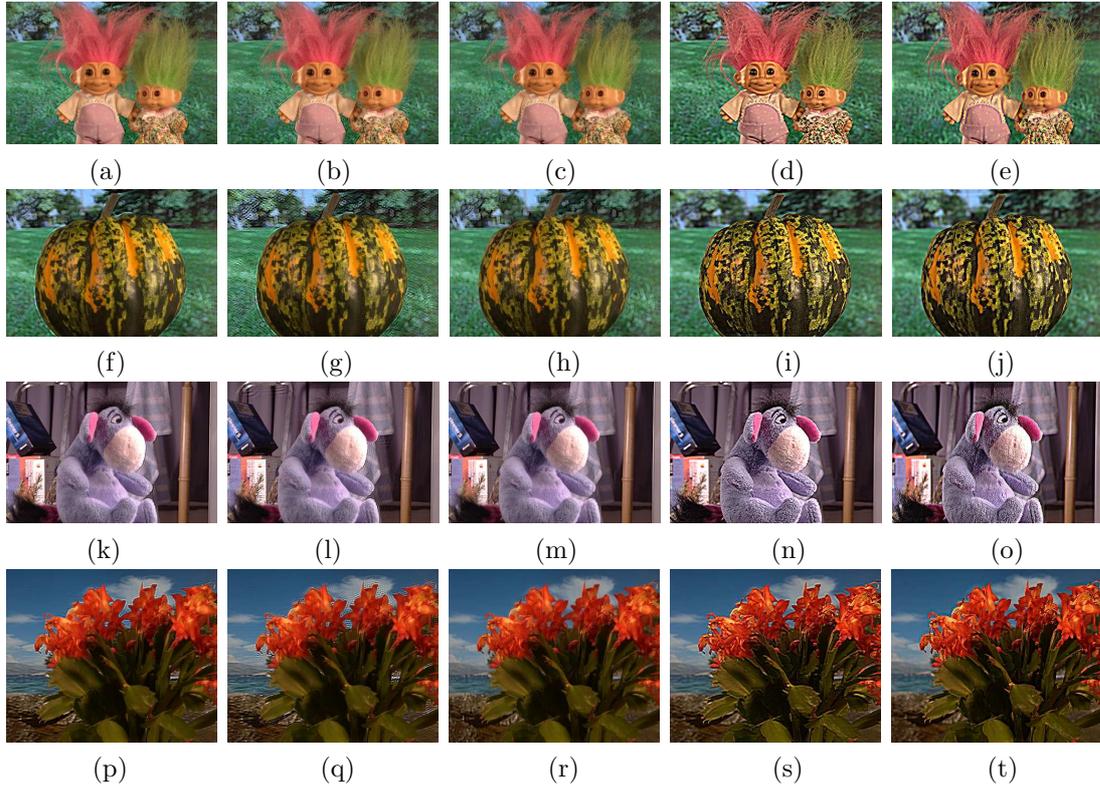


Fig. 4.13. (a), (f), (k), and (p) the restored images by the method in [14]; (b), (g), (l), (q) the restored images by the method in [20]; (c), (h), (m), (r) the restored images by the method in [41]; (d), (i), (n), (s) the restored images by Algorithm 2; (e), (j), (o), (t) the restored images by Algorithm 1.

required. We observe from the figures that the visual quality of the deblurred images by the two algorithms are quite good. However, we find that the deblurred images by Algorithm 2 are better than those by Algorithm 1, especially when the motion blur is more serious. In the second experiment, we consider the images that are blurred by a random linear motion and a V-type motion kernels. The original images are shown in Fig. 4.5(a). The resulting blurred images and motion blur kernels are shown in Figs. 4.6(a) and (e). We display the restored images and the recovered motion blurs by the the algorithm in [16], the total variation based algorithm, and the frame-based algorithm in Figs. 4.6(b) and (f), Figs. 4.6(c) and (g), and Figs. 4.6(d) and (h) respectively. We observe from the figures that the visual quality of the deblurred images and the estimated motion kernels by the two algorithms are quite good.

In the third experiment, we consider the images that are blurred by random non-linear motion kernels. The original images are shown in Fig. 4.5(b) and (c). The resulting blurred images and motion blur kernels are shown in Figs. 4.7(a) and (e). We display the restored images

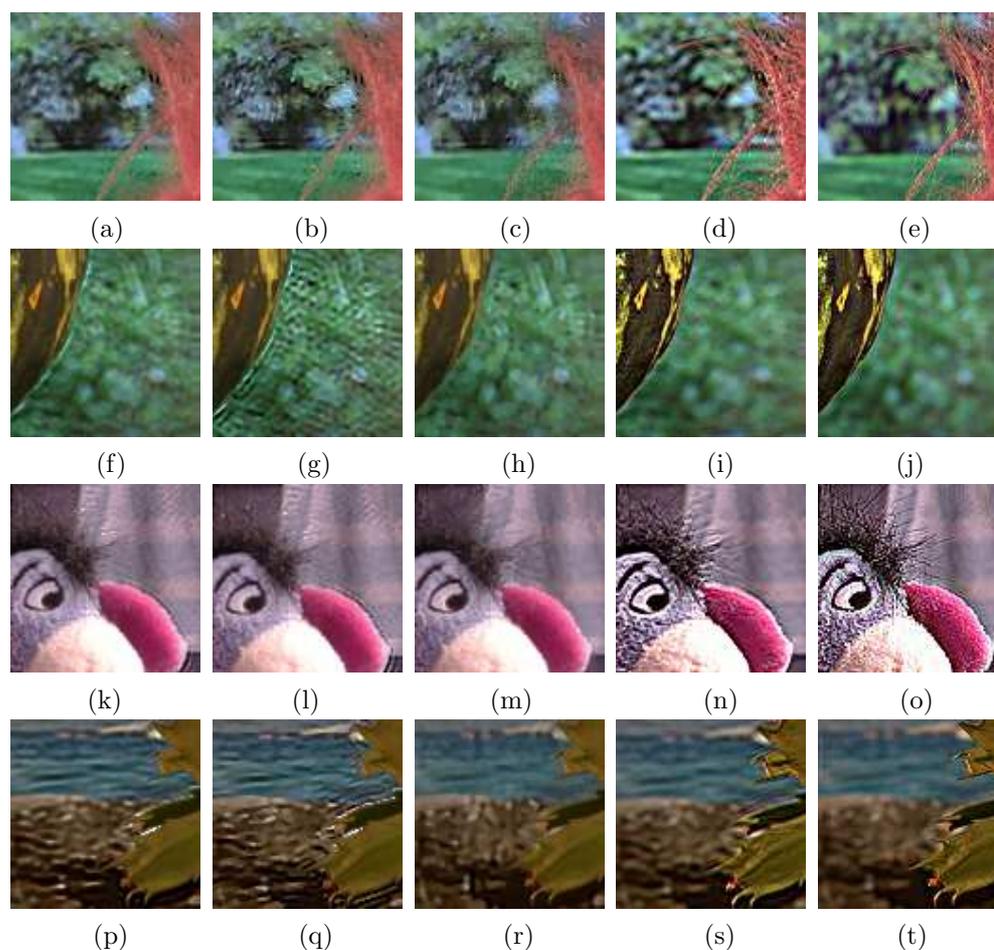


Fig. 4.14. The zooming parts of Fig. 4.12(a)-(t).

and the recovered motion blurs by the the algorithm in [16], the total variation based algorithm, and the frame-based algorithm in Figs. 4.7(b) and (f), Figs. 4.7(c) and (g) and Figs. 4.7(d) and (h) respectively. The estimated motion blurs are also shown on the top left hand side of the restored images. According to the restoration results, the proposed two algorithms can recover both the blur and the image quite well.

4.3. Comparisons

In this section, we give some other examples for comparison. The input observed images are shown in Figs. 4.8 - 4.11(a) and (e). We display the restored images by the algorithm in [16], the total variation based algorithm, and the frame-based algorithm, in Figs. 4.8 - 4.11(b), (f), Figs. 4.8 - 4.11(c), (g) and Figs. 4.8 - 4.11(d), (h) respectively. Visually, we find that the restored images are better than those given by the algorithm in [16]. In Figs. 4.9 and 4.11, we show the zoomed parts of the restored images in Figs. 4.8 and 4.10 respectively. Again it is clear that the proposed two algorithms can restore images quite well.

Then we test the effects of the proposed algorithms for out-of-focus blurs. The input images are shown in Fig. 12, where we can see that they are all out-of-focus especially the background. We compare the two proposed algorithms with some other approaches, such as the method

in [14], in [20], and in [41]. We display the restored images by the method in [14], in [20], in [41], the frame-based algorithm, and the total variation based algorithm, in Figs. 4.13(a), (f), (k), (p), Figs. 4.13(b), (g), (l), (q), Figs. 4.13(c), (h), (m), (r), Figs. 4.13(d), (i), (n), (s), Figs. 4.13(e), (j), (o), (t) respectively. In Fig. 4.14, we show the zoomed parts of the restored images in Fig. 4.13 respectively. We can see that the performance of the proposed total variation based algorithm is better than the proposed frame-based algorithm for out-of-focus blurs, and is also better than other methods.

5. Concluding Remarks

In this paper, we proposed two blind deconvolution methods to remove camera motion or out-of-focus blur from a single image by minimizing a energy functional with different regularization terms. The first model is to use the total variation prior in both image and blur, while the second model is to use the frame based prior in both image and blur. The main contribution of this paper is to show how to employ the generalized cross validation (GCV) method automatically to estimate the two regularization parameters associated with the priors in these two blind motion deblurring models. Experimental results show the effectiveness and efficiency of the proposed methods.

References

- [1] D. Kundur and D. Hatzinakos, Blind image deconvolution, *IEEE Signal Proc. Mag.*, **13:3** (1996), 43-64.
- [2] P. Campisi and K. Egiazarian, Blind image deconvolution: theory and applications, CRC Press, 2007.
- [3] B. Bascle, A. Blake, and A. Zisserman, Motion deblurring and super-resolution from an image sequence, In ECCV, (1996), 573-582.
- [4] M. Ben-Ezra and S.K. Nayar, Motion-based motion deblurring, *IEEE Trans. PAMI*, **26** (2004), 689-698.
- [5] J. Chen, L. Yuan, C.K. Tang, and L. Quan, Robust dual motion deblurring, In CVPR, 2008.
- [6] Y. Lu, J. Sun, L. Quan, and H. Shum, Blurred/non-blurred image alignment using an image sequence, In SIGGRAPH, 2007.
- [7] A. Rav-Acha, and S. Peleg, Two motion blurred images are better than one, *Pattern Recogn. Lett.*, **26** (2005), 311-317.
- [8] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, Image deblurring with blurred/noisy image pairs, In SIGGRAPH, 2007.
- [9] S.K. Kim, and J.K. Paik, Out-of-focus blur estimation and restoration for digital auto-focusing system, *Electron. Lett.*, **34** (1998), 1217-1219.
- [10] A. Likas, and N. Galatsanos, A Variational approach for bayesian blind image deconvolution, *IEEE T. Signal Proces.*, **52** (2004), 2222-2233.
- [11] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W. Freeman, Removing camera shake from a single photograph, *ACM T. Graphic.*, **25** (2006), 787-794.
- [12] Y.-W. Tai, P. Tan, M. S. Brown, Richardson-Lucy Deblurring for Scenes under Projective Motion Path, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, Accepted.
- [13] S. Dai, and Y. Wu, Motion from Blur, CVPR, 2008.
- [14] J. Jia, Single Image Motion Deblurring Using Transparency, CVPR, 2007.
- [15] Q. Shan, J. Jia, and A. Agarwala, High-quality motion deblurring from a single image, In SIGGRAPH, 2008.

- [16] L. Xu and J. Jia, Two-Phase Kernel Estimation for Robust Motion Deblurring, ECCV 2010.
- [17] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D*, **60** (1992), 259-268.
- [18] T.F. Chan and C.K. Wong, Total variation blind deconvolution, *IEEE T. Image Process.*, **7:3** (1998), 370- 375.
- [19] Y. You and M. Kaveh, Blind image restoration by anisotropic regularization, *IEEE T. Image Process.*, **8:3** (1999), 396-407.
- [20] S. Chan, P. Gill, and T. Nguyen, An Augmented Lagrangian Method for Total Variation Image Restoration, *IEEE TIP*, 2010, submitted, under review.
- [21] Y. Huang and M. Ng, Lipschitz and total-variational regularization for blind deconvolution, *Commun. Comput. Phys.*, **4** (2008), 195-206.
- [22] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, Blind motion deblurring using multiple images, UCLA Cam Report 09-59.
- [23] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, High-quality curvelet-based motion deblurring from an image pair, UCLA Cam Report 09-60.
- [24] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, Blind motion deblurring from a single image using sparse approximation, UCLA Cam Report 09-61.
- [25] G. Golub, M. Heath, and G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, *Technometrics*, **21** (1979), 215-223.
- [26] H. Engl, M. Hanke, and A. Neubauer, Regularization of inverse problems, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [27] M. Buckley, Fast computation of a discretized thin-plate smoothing spline for image data, *Biometrika*, **81** (1994), 247-258.
- [28] R. Chan, M. Ng, and R. Plemmons, Generalization of Strang's preconditioner with applications to Toeplitz least squares problems, *J. Numer. Linear Algebra Appl.*, **3** (1996), 45-64.
- [29] R. Gonzalez and R. Woods, Digital Image Processing, Addison- Wesley, Reading, MA, 1992.
- [30] A. Jain, Fundamentals of Digital Image Processing, Prentice- Hall, Englewood Cliffs, NJ, 1989.
- [31] A. Chai, and Z. Shen, Deconvolution: a wavelet frame approach, *Numer. Math.*, **106** (2007), 529-587.
- [32] J.-F. Cai, R. H. Chan, and Z. Shen, A framelet-based image inpainting algorithm, *Appl. Comput. Harmon. A.*, **24** (2008), 131-149.
- [33] A. Ron and Z. Shen, Affine system in $L_2(\mathbb{R}^d)$: the analysis of the analysis operator, *J. Funct. A.*, **148** (1996), 408-447.
- [34] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro, A variational framework for simultaneous motion estimation and restoration of motion-blurred video, In ICCV, 2007.
- [35] Y. Huang, M. Ng and Y. Wen, A fast total variation minimization method for image restoration, *SIAM J. Multiscale Model. Simul.*, **7** (2008), 774-795.
- [36] A. Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vis.*, **20** (2004), 89-97.
- [37] M. Ng, L. Qi, Y. Yang, and Y. Huang, On semismooth Newton's methods for total variation minimization, *J. Math. Imaging Vis.*, **27** (2007), 265-276.
- [38] T. Goldstein, and S. Osher, The split bregman method for L1 regularized problems, UCLA Cam Report 08-29.
- [39] J. Cai, S. Osher, and Z. Shen, Linearized bregman iterations for compressed sensing, UCLA Cam Report 08-06.
- [40] J. Cai, S. Osher, and Z. Shen, Convergence of the linearized bregman iterations for l_1 -norm minimization, UCLA CAM Report 08-52.
- [41] S. H. Chan and T. Nguyen, Single Image Spatial Variant Out-of-focus Blur Removal, submitted to IEEE International Conference on Image Processing (ICIP'11), 2011.