Blind motion deblurring using multiple images

Jian-Feng Cai[a,*], Hui Ji[b,*], Chaoqiang Liu[a], Zuowei Shen[b]

[a] *Center for Wavelets, Approx. and Info. Proc., National University of Singapore, Singapore, 117542*
[b] *Department of Mathematics, National University of Singapore, Singapore, 117542*

Abstract

Recovery of degraded images due to motion blurring is one challenging problem in digital imaging. Most existing techniques on blind deblurring are not capable of removing complex motion blurring from the blurred images of complex structures. One promising approach is to recover the clear image using multiple images captured for the scene. However, it is observed in practice that such a multi-frame approach can recover a high-quality clear image of the scene only after multiple blurred image frames are accurately aligned during pre-processing, which is a very challenging task even with user interactions. In this paper, by exploring the sparsity of the motion blur kernel and the clear image under certain domains, we proposed an alternative iteration approach to simultaneously identify the blur kernels of given blurred images and restore a clear image. Our proposed approach not only is robust to image formation noises, but also is robust to the alignment errors among multiple images. A modified version of linearized Bregman iteration is then developed to efficiently solve the resulting minimization problem. The experiments showed that our proposed algorithm is capable of accurately estimating the blur kernels of complex camera motions with minimal requirements on the accuracy of image alignment. As a result, our method is capable of automatically recovering a high-quality clear image from multiple blurred images.

Key words: blind deconvolution, tight frame, motion blur, image restoration

1. Introduction

Motion blurring caused by camera shake has been one of the prime causes of poor image quality in digital imaging, especially when using telephoto lenses or long shuttle speeds. In many imaging applications, there is simply not enough light to produce a clear image by using a short shutter speed. As a result, the image will appear blurry due to the relative motion between the camera and the

*Corresponding author. Tel.: +65-65168845, Fax: +65-67795452
Email addresses: tslcaij@nus.edu.sg (Jian-Feng Cai), matjh@nus.edu.sg (Hui Ji), tslcql@nus.edu.sg (Chaoqiang Liu), matzuows@nus.edu.sg (Zuowei Shen)

scene. Motion blurring can significantly degrade the visual quality of images. Thus, how to restore motion-blurred images has long been a fundamental problem in digital imaging. Motion blurring due to camera shake is usually modeled as a spatially invariant convolution process:

$$f = g * p + n, \tag{1}$$

where $*$ is the convolution operator, g is the clear image to recover, f is the observed blurred image, p is the blur kernel (or so-called *point spread function*), and n is the noise. How to recover the clear image g from the blurred image f is the so-called *image deconvolution* problem.

There are two cases in image deconvolution problems: non-blind deconvolution and blind deconvolution. In the non-blind case, the blur kernel p is assumed to be known or estimated somewhere else, and the task is to recover the clear image g by reversing the effect of convolution on the blurred image f. Such a deconvolution is known as an ill-conditioned problem, as a small perturbation on f may cause the direct solution from (1) being heavily distorted. In past, there have been extensive studies on robust non-blind deconvolution algorithms (e.g. [1, 22, 8, 7, 18]). In the case of blind deconvolution, both the blur kernel p and the clear image g are unknown. Then the problem becomes under-constrained and there exist infinitely many solutions. In general, blind deconvolution is much more challenging than non-blind deconvolution. Motion deblurring is a typical blind deblurring problem, because the motion between the camera and the scene always varies for different images.

Some prior assumptions on both the kernel p and the image g have to be made in order to eliminate the ambiguities between the kernel and the image. In practice, the motion-blur kernel is very different from the kernels of other types of blurring (e.g. out-of-focus blurring, Gaussian-type optical blurring), as there exist not simple parametric forms to represent motion-blur kernels. In general, the motion-blur kernel can be expressed as

$$p = v(x, y)|_C, \tag{2}$$

where C is a continuous curve of finite length in $\mathbb{R}^2$ which denotes the camera trajectory and $v(x, y)$ is the speed function which varies along C. Briefly, the motion-blur kernel p is a smooth function with the support of a continuous curve.

1.1. Previous work

Early works on motion deblurring usually use only one single blurred image. Most such methods (e.g. [10, 29, 21, 17]) require a prior parametric knowledge of the blur kernel p such that the blur kernel can be obtained by only estimating a few parameters. These methods usually are computationally efficient but only work on simple blurrings such as symmetric optical blurring or simple motion blurring of constant velocity. To remove more complicated blurring from images, an alternative approach is using a joint minimization model to

simultaneously estimate both the blur kernel and the clear image. To overcome the inherent ambiguities between the blur kernel and the clear image, certain regularization terms have to be added in the minimization, e.g., total variation (TV) regularization proposed by [9, 8, 16, 20]. These TV-based blind deconvolution techniques showed good performance on removing certain blurrings on specific types of images, such as out-of-focus blurring on medical images and satellite images. However, TV regularization is not a good choice for the case of motion-blurring, because TV regularization penalizes, e.g., the total length of the edges for piecewise constant functions (see [8]). As a result, the support of the resulting blur kernel tends to be a disk or several isolated disks, instead of a continuous curvy camera trajectory. Also, for many images of nature scenes, TV-based regularization does not preserve the details and textures very well on the regions of complex structures due to the stair-casing effects (see [13, 23]).

In recent years, the concept of *epsilon photography* ([27]) has been popular in digital imaging to optimize the digital camera by recording the scene via multiple images, captured by an epsilon variation on the camera setting. These multiple images make a more complete description on the scene, which leads to an easier configuration for many traditionally challenging image processing tasks including blind motion deblurring. Most such approaches (e.g. [3, 26, 28, 19]) actively control the capturing process using specific hardwares to obtain multiple images of the scene such that the blur kernel is easier to infer and the deblurring is also more robust to noise. Impressive results have been demonstrated by these approaches. However, the requirement on the active acquisition of input images limits the wider applications of these techniques in practice.

1.2. Our work

The goal of this paper is to develop a robust numerical algorithm to recover a high-quality clear image of the scene from multiple motion-blurred images. In our setting, the input multiple images are passively captured by a commodity digital camera without any specific hardware. It is noted that the proposed algorithm could also be applied to removing motion blurring in the video with little modifications. In other words, as the input in our algorithm, there are M available blurred images with the following relationships:

$$\{f_i = g(h_i(\cdot)) * p_i + n_i, \quad i = 1, 2, \cdots, M\},$$

where h_i is the spatial geometric transform from the clear image g to the blurred image f_i, determined by the camera pose when the i-th image is taken, p_i is the blur kernel of the i-th image, and n_i is the noise.

In this paper, we assume the geometric transforms h_is among all images are estimated using some existing image alignment technique during pre-processing. However, we emphasize that there is no available image alignment technique which is capable of accurately aligning blurred images. Therefore, we take the following f_is as the input in our mathematical formulation:

$$\{f_i = g((I + \epsilon h_i)(\cdot)) * p_i + n_i, \quad i = 1, 2, \cdots, M\}, \tag{3}$$

where I is the identical operator, p_i and g are unknowns, ϵh_i and n_i are alignment errors and image formation noises respectively. In summary, there are two types of undesired perturbations when taking a multi-image approach to remove motion blurring. One is the image formation noise n_i and the other is image alignment error ϵh_i. The goal of this paper is to develop a numerical algorithm robust to both types of perturbations.

In this paper, we begin our study on blind motion deblurring by investigating how to measure the "clearness" of the recovered image and on the "soundness" of the blur kernel. Our study shows that, given multiple images (≥ 2), the sparsity of image under tight frame systems ([12, 30]) is a good measurement on the "clearness" of the recovered image, and the sparsity of blur kernels in a weighted image space is a good measurement on the "soundness" of the blur kernel when combined with a smoothness regularization. In particular, it is shown empirically that the impressive robustness to image alignment error is achieved by using these two sparsity regularizations on restoring the clear image and on estimating the blur kernels. In our proposed approach, the sparsity of an image under a tight frame system is the ℓ_1-norm of its tight frame coefficients, and the sparsity of a blur kernel in a weighted image space is measured by its weighted ℓ_1-norm.

The rest of the paper is organized as follows. In Section 2, we formalized the minimization strategy and explained the underlying motivation. In Section 3, we presented the detailed numerical algorithm for solving the proposed minimization problem. Section 4 is devoted to the experimental evaluation and the discussion on future works.

2. Problem formulation and analysis

When taking an image by a digital camera, the image does not represent the scene in a single instant of time, instead it represents the scene over a period of time. As the camera does not keep still due to unavoidable camera shake, the image of the scene must represent an integration of all camera viewpoints over the period of exposure time, which is determined by the shuttle speed. The resulted image will look blurry along the relative motion path between the camera and the scene. As the relative motion between the scene and the camera is generally global; usually we assume that the relative motion is spatially invariant. Thus the motion blurring is simplified as a convolution process:

$$f = g * p + n, \tag{4}$$

where p is so-called blur kernel function which represents the relative motion with normalization, f is the observed blurred image and g is the desired clear image.

2.1. Benefits of using multiple images

The benefits of using multiple images to remove motion blurring are twofolded. First, it is known that non-blind deblurring is an ill-conditioned problem

as it is sensitive to noises. The noise sensitivity comes from the fact that reversing image blurring process will greatly amplify the high-frequency noise contribution to the recovered image. Let $\widehat{(\cdot)}$ denote the Fourier transform, then (1) becomes

$$\widehat{p}(\omega) \cdot \widehat{g}(\omega) = \widehat{f}(\omega) - \widehat{n}.$$

The values of $|\widehat{p}|$ are zero or very small at large ω because the blur kernel p usually is a low-pass filter. Thus, $\widehat{g}(\omega)$ is very sensitive to even small perturbations at large ω. Extensive studies have been carried out in past to reduce such noise sensitivities by imposing some extra regularizations.

However, if we have multiple perfectly aligned images f_i on the same scene with different motion-blur kernels:

$$f_i = p_i * g + n_i, \quad i = 1, 2, \cdots, M. \tag{5}$$

Because motion-blur kernels are not isotropic, the intersection of all ω with small $|\widehat{p_i}(\omega)|$ is very likely to be much smaller than the set of ω with small $|\widehat{p_i}(\omega)|$ for each individual p_i. Let us examine a simple example of two blurred images: one is blurred by a horizontal filter $p_1 = \frac{1}{N}(1, 1, \cdots, 1)$; the other is blurred by a vertical filter $p_2 = \frac{1}{N}(1, 1, \cdots, 1)^T$. Then we have

$$\widehat{p_1}(\omega_x, \omega_y) = \mathrm{sinc}(\frac{2}{N}\omega_x); \quad \widehat{p_2}(\omega_x, \omega_y) = \mathrm{sinc}(\frac{2}{N}\omega_y).$$

$\widehat{p_1}$ has the periodic lines of zero points $\{(\omega_x = \frac{kN\pi}{2}, \omega_y), k \in \mathbb{Z}\}$ and so does $\widehat{p_2}$. However, the combination of $\widehat{p_1}$ and $\widehat{p_2}$ only has periodic points $\{(\omega_x = \frac{kN\pi}{2}, \omega_y = \frac{kN\pi}{2}), k \in \mathbb{Z}\}$, which can greatly improve the condition of the deconvolution process. In practice, we may not have such an ideal configuration. But it is very likely that different blurred images have different blur kernels, as the camera motion is random during the capture. Heuristically, it is a reasonable assumption that the combination of all blur kernels

$$(\sum_{i=1}^{N} |\widehat{p_i}|^2(\omega))^{\frac{1}{2}}.$$

will have much fewer small values in its spectrum than the individual blur kernel does. Therefore, using multiple images can greatly improve the noise sensitivity of the deconvolution process.

Secondly, for blind deblurring, the estimation of blur kernels will benefit even more by using multiple images. The main difficulty in blind deblurring is that the problem is under-constrained. There exist infinitely many solutions. For example, the famous degenerate solution to (4)

$$p := \delta, \quad g := f.$$

has been a headache for many blind deblurring techniques. Although the degenerate solution can be avoided by some ad-hoc processes, the inherent ambiguities between the kernel and the image still lead to the poor performance of most available techniques on removing complex motion blurring from images.

However, such ambiguities can be significantly reduced by using multiple images. Again, let us examine one example. In the case of a single image, let the blur kernel be decomposed into two parts: $p = p_1 * p_2$. Then besides the true solution (p, g), $(p_1, p_2 * g)$ is also a solution to (4). As long as p_2 is a low-pass filter, even imposing other available physical constraints on images or blur kernels, e.g.,

$$p \geq 0; \quad \sum_{j,k} p(j,k) = 1; \quad g \geq 0, \tag{6}$$

will not eliminate such ambiguities. On the contrary, in the case of multiple images, it is very unlikely that such ambiguities will happen. In order to have another solution to the system (5) when using multiple images, all N kernels p_i need to have a common factor p_2 such that

$$p_i = p_{1_i} * p_2.$$

Considering the fact that the support of each p_i is a curve in 2D, it is unlikely that all p_i will have a non-trivial common factor.

In summary, multiple blurred images on the same scene provide much more information than a single blurred image does, which leads to a better configuration for recovering a clear image on the scene. But, some new challenging computational problems also arise when taking a multi-frame approach.

2.2. Challenges of using multiple images

When we take multiple images on the same scene, the camera pose varies during the capture. In other words, we have multiple observed blurred image f_i related to the clear picture g up to a spatial geometrical transform h_i:

$$\{f_i = g(h_i(\cdot)) * p_i + n_i, \quad i = 1, 2, \cdots, M\}.$$

The estimation on h_i is known as the image registration problem, which has been extensively studied in past (see [34] for more details). However, most image registration techniques need to assume a simple model (e.g. affine transform) on the geometrical transform, which may not approximate the true geometrical transform well when the 3D structure of the scene is complicated. Furthermore, when images are seriously blurred, the appearances of the same scene blurred by different blur kernels are very different from each other. Currently there does not exist an alignment technique which is capable of accurately aligning seriously blurred images. Therefore, we have a new perturbation source when using multiple images: image alignment error.

It is observed in [19] that alignment errors will seriously impact the performance of multi-image blind deblurring. In [19], a simple experiment is carried out to illustrate the high sensitivity of blind deblurring to alignment errors when estimating blur kernels. The experiment considered a simplified configuration by assuming knowing the clear image and the blurred image up to a small alignment perturbation. The only unknown is the blur kernel. Thus, the inputs of the experiment are a clear image g shown in Fig. 1 (a) and its blurred version

(a) clear image (b) blurred image

$(c)(-1.4°, 0.98)$ $(d)(-0.7°, 0.99)$ (e) $(0°, 1.00)$ (f) $(0.7°, 0.99)$ (g) $(1.4°, 0.98)$.

Figure 1: (b) is the synthesized blurred version of the clear image (a) by using the blur kernel shown in the top right of (b). (c)-(f) are the blur kernels estimated based on the blurred image (a) and a number of transformed clear image (b) with different rotation and scale. Two numbers in brackets under each kernel are the rotation angle and the scale value (θ, s). All images are taken from [19].

f shown in Fig. 1 (b) up to an alignment perturbation. The corresponding blur kernel p is shown in the top right corner of the blurred image. The alignment perturbations are simulated by applying a similarity transform on Fig. 1 (b) with various pairs of rotations and scales (θ, s) and with the same translation (t_x, t_y):

$$h : \begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \tag{7}$$

Then the blur kernel is estimated based on the clear image g in Fig. 1 (a) and the blurred image Fig. 1 (b) with a geometrical perturbation h defined in (7). In other words, the estimated blur kernel p' is based on the following perturbed version of the original equation:

$$p' * g(h(\cdot)) = f = p * g(\cdot). \tag{8}$$

In the experiment done by [19], p' is estimated by solving (8) using a least squares method with Tikhonov regularizations. It is noted that the multi-image blind blurring is not sensitive to small translation errors, as the translation between two images only results in a spatial shift of the estimated kernel. Thus, the translation error is fixed as a constant in the experiment.

The estimated blur kernels p' is given in Fig 1 (c)–(g) with respect to small alignment perturbations of the form (7) for various s and θ. The results clearly showed that the blur kernel is very sensitive to even a small alignment error in the simplified case where the true clear image is available. In practice, the problem is much more ill-conditioned as we do not have the clear image in hand. This experiment clearly indicated the importance of the robustness to alignment errors when developing multi-image blind motion deblurring techniques.

3. Formulation for blind motion deblurring with sparsity regularizations

3.1. Outline of our proposed algorithm

Given M blurred images f_i, $i = 1, 2, \cdots, M$ satisfying the relationship (3):

$$f_i = g((I + \epsilon h_i)(\cdot)) * p_i + n_i, \quad i = 1, 2, \cdots, M,$$

we take a regularization-based approach to solve the blind motion deblurring problem, which requires the simultaneous estimations on both the clear image g and M blur kernels $\{p_i,\, i = 1, \cdots, M\}$. It is well known that the regularization-based blind deconvolution approaches usually result in solving a challenging non-convex minimization problem. In our case, the number of the unknowns is up to the order of millions. The most commonly used approach is an alternative iteration scheme; see [9] for instance. The alternative iteration scheme can be described as following: let $g^{(0)}$ be the initial guess on the clear image.

Algorithm 1 Outline of the alternative iterations

For $k = 0, 1, \cdots,$

1. given the clear image $g^{(k)}$, compute the blur kernels $\{p_i^{(k+1)},\, i = 1, 2, \ldots, M\}$.
2. given the blur kernels $\{p_i^{(k+1)},\, i = 1, 2, \ldots, M\}$, compute the clear image $g^{(k+1)}$;

There are two steps in Algorithm 1. Step 2 is a non-blind image deblurring problem, which has been studied extensively in the literature; see, for instances, [1, 22, 8, 7, 18, 5]. However, there is one more error source in Step 2 than the traditional non-blind deblurring problem does, that is, the error in the intermediate blur kernel $p_i^{(k+1)}$ used for deblurring. Inspired by recent non-blind deblurring techniques which are based on the sparse approximation to the image under certain tight frame systems ([7, 4]), we also use the sparsity constraint on the clear image g under tight frame systems to regularize the non-blind deblurring. And we use a modified version of so-called *linearize Bregman iteration* (See [24, 32, 31, 16, 20, 11, 25, 4, 5, 6, 15]) to achieve impressive robustness to image noises, alignment errors, and, more importantly, the perturbations on the given intermediate blur kernels. In our implementation, we choose the tight framelet system constructed in [12, 30] as the choice of the tight frame system representing the clear image g.

For Step 1, it is observed in Fig. 1 that the alignment error will lead to a false estimation on the motion blur kernel. The support of the false kernel tends to be much larger than that of the true blur kernel. Based on this observation, we propose to overcome the sensitivity of estimating blur kernels to alignment errors by exploring the sparsity constraint on the motion blur kernel in its

spatial domain. Similarly, we also use a modified version of linearized Bregman iteration to solve the resulting minimization problem. Before we present the detailed algorithm, we give a brief introduction to the framelet system used in our method in the remaining of the section.

3.2. Tight framelet system and image representation

A countable set $X \subset L^2(\mathbb{R})$ is called a *tight frame* of $L^2(\mathbb{R})$ if

$$f = \sum_{h \in X} \langle f, h \rangle h, \quad \forall f \in L^2(\mathbb{R}), \tag{9}$$

where $\langle \cdot, \cdot \rangle$ is the inner product of $L^2(\mathbb{R})$. An orthonormal basis is a tight frame, hence a tight frame is a generalization of an orthonormal basis. However, tight frames sacrifice the orthonormality and the linear independence of the system in order to get more flexibilities. Tight frames can be redundant.

For given $\Psi := \{\psi_1, \ldots, \psi_r\} \subset L_2(\mathbb{R})$, the *affine (or wavelet) system* is defined by the collection of the dilations and the shifts of Ψ as

$$X(\Psi) := \{\psi_{\ell,j,k} : 1 \le \ell \le r;\ j,k \in \mathbb{Z}\} \quad \text{with} \quad \psi_{\ell,j,k} := 2^{j/2}\psi_\ell(2^j \cdot -k). \tag{10}$$

When $X(\Psi)$ forms a tight frame of $L_2(\mathbb{R})$, it is called a *tight wavelet frame*, and ψ_ℓ, $\ell = 1, \ldots, r$, are called the *(tight) framelets*.

To construct a set of framelets, usually, it starts with a compactly supported refinable function $\phi \in L^2(\mathbb{R})$ (a scaling function) with a refinement mask τ_ϕ satisfying

$$\widehat{\phi}(2\cdot) = \tau_\phi \widehat{\phi}.$$

Here $\widehat{\phi}$ is the Fourier transform of ϕ, and τ_ϕ is a trigonometric polynomial with $\tau_\phi(0) = 1$, i.e., a refinement mask of a refinable function must be a lowpass filter. For a given compactly supported refinable function, the construction of tight framelet systems is to find a finite set Ψ that can be represented in the Fourier domain as

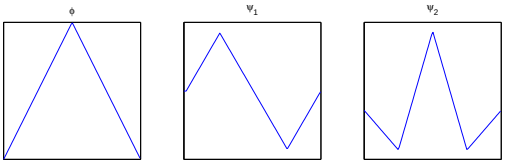
$$\widehat{\psi}(2\cdot) = \tau_\psi \widehat{\phi}$$

for some 2π-periodic τ_ψ. The unitary extension principle (UEP) of [30] says that $X(\Psi)$ in (10) generated by Ψ forms a tight frame in $L^2(\mathbb{R})$ provided that the masks τ_ϕ and $\{\tau_\psi\}_{\psi \in \Psi}$ satisfy:

$$\tau_\phi(\omega)\overline{\tau_\phi(\omega + \gamma\pi)} + \sum_{\psi \in \Psi} \tau_\psi(\omega)\overline{\tau_\psi(\omega + \gamma\pi)} = \delta_{\gamma,0}, \quad \gamma = 0, 1 \tag{11}$$

for almost all ω in $\mathbb{R}$. τ_ϕ must correspond to a low-pass filter and $\{\tau_\psi\}_{\psi \in \Psi}$ must correspond to highpass filters. The sequences of Fourier coefficients of τ_ψ, as well as τ_ψ itself, are called *framelet masks*. In our implementation, we adopt piece-wise linear B-spline framelet constructed in [12, 30]. The refinement mask is $\tau_\phi(\omega) = \cos^2(\frac{\omega}{2})$, whose corresponding lowpass filter is $h_0 = \frac{1}{4}[1, 2, 1]$.

Figure 2: Piecewise linear framelets.



Two framelets are $\tau_{\psi_1} = -\frac{\sqrt{2}i}{2}\sin(\omega)$ and $\tau_{\psi_2} = \sin^2(\frac{\omega}{2})$, whose corresponding highpass filters are

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

The associated refinable function and framelets are given in Fig. 2. With a 1D framelet system for $L_2(\mathbb{R})$, the 2D framelet system for $L_2(\mathbb{R}^2)$ can be easily constructed by the tensor product of 1D framelet.

In the discrete case, an $n \times n$ image $\mathbf{f}$ is considered as the coefficients $\{\mathbf{f}(i) = \langle f, \phi(\cdot - i)\rangle, i \in \mathbb{Z}^2\}$ up to a dilation, where ϕ is the refinable function associated with the framelet system, and $\langle\cdot,\cdot\rangle$ is the inner product in $L_2(\mathbb{R}^2)$. The L-level discrete framelet decomposition of f is then the coefficients $\{\langle f, 2^{-L/2}\phi(2^{-L}\cdot - j)\rangle, j \in \mathbb{Z}^2\}$ at a prescribed coarsest level L, and the framelet coefficients

$$\{\langle f, 2^{-l/2}\psi_i(2^{-l}\cdot - j)\rangle, j \in \mathbb{Z}^2, 1 \le i \le r^2 - 1\},$$

for $0 \le l \le L$.

If we denote f as a vector $\mathbf{f} \in \mathbb{R}^N, N = n^2$ by concatenating all columns of the image, the discrete framelet decomposition of $\mathbf{f}$ can be described by the vector $A\mathbf{f}$, where A a $K \times N$ matrix. By the UEP (11), $A^TA = I$, thus the rows of A form a tight frame system in $\mathbb{R}^N$. In other words, the framelet decomposition operator A can be viewed as a tight frame system in $\mathbb{R}^N$ as its rows form a tight frame in $\mathbb{R}^N$ such that the perfect reconstruction formula $\mathbf{x} = \sum_{\mathbf{y}\in A}\langle \mathbf{x}, \mathbf{y}\rangle\mathbf{y}$ holds for all $\mathbf{x} \in \mathbb{R}^N$. Unlike the orthonormal case, we emphasize that $AA^T \neq I$ in general. In our implementation, we use a multi-level tight framelet decomposition without down-sampling under the Neumann (symmetric) boundary condition. The detailed description can be found in [7].

4. Numerical algorithm

This section is devoted to the detailed numerical algorithm of our blind deconvolution algorithm outlined in Algorithm 1. Both steps in Algorithm 1 are based on the linearized Bregman iteration. The Bregman iteration was first introduced for non-differentiable TV-energy in [24], and then was successfully applied to ℓ_1-norm minimization in compressed sensing in [32] and to wavelet based denoising in [31]. The Bregman iteration was also used in TV-based blind

deconvolution in [16, 20]. To further improve the performance of the Bregman iteration, a linearized Bregman iteration was invented in [11]; see also [32]. More details and an improvement called "kicking" of the linearized Bregman iteration was described in [25], and a rigorous theory was given in [4, 6]. The linearized Bregman iteration for frame-based image deblurring was proposed in [5]. Recently, a new type of iteration based on Bregman distance, called split Bregman iteration, was introduced in [15], which extended the utility of Bregman iteration and linearized Bregman iteration to minimizations of more general ℓ_1-based regularizations including total variation, Besov norms and sums of such things.

Consider M blurred images $\mathbf{f}_i \in \mathbb{R}^N, i = 1, \cdots, M$. We assume that the size of the blur kernel is no larger than $n \times n$. Let $\mathbf{p}_i \in \mathbb{R}^N$ denote the blurred image p_i after column concatenation. Let $[\,]_*$ denote the convolution operator of p and f after concatenating operation:

$$p * f = [\mathbf{p}]_* \mathbf{f} = [\mathbf{f}]_* \mathbf{p}.$$

Let $\mathbf{u} = A\mathbf{g}$ denote the framelet coefficients vector of the clear image $\mathbf{g}$.

4.1. Method for Step 2 in Algorithm 1

In Step 2 of Algorithm 1, at the k-th iteration, we need to find a clear image $\mathbf{g}^{(k+1)}$ given the blur kernels $\{\mathbf{p}_i^{(k+1)}, i = 1, 2, \cdots M\}$.

In the initial stages, since the kernel is not in good shape, it is not necessary to find an accurate $\mathbf{g}^{(k+1)}$. We simply use a least squares deblurring algorithm, i.e., solve

$$\min_{\mathbf{g}} \frac{1}{2} \sum_{i=1}^{M} \|[\mathbf{p}_i^{(k+1)}]_* \mathbf{g} - \mathbf{f}_i\|_2^2 + \lambda \|\nabla \mathbf{g}\|_2^2. \tag{12}$$

This can be done efficiently by FFTs.

In the final stages, the kernel becomes in good shape, so we need an accurate solution of the clear image. For this, we solve the image deblurring problem in the tight framelet domain. Let $\mathbf{u}$ be the tight framelet coefficients of the clear image $\mathbf{g}^{(k+1)}$, i.e., $\mathbf{g}^{k+1} = A^T \mathbf{u}$. Our strategy of recovering the clear image $\mathbf{g}^{(k+1)}$ is to find a sparse solution $\mathbf{u}$ in the tight framelet domain among all solutions with reasonable constraints.

Temporarily, we ignore the mis-alignment error and assume that the blur kernels are accurate enough, such that there exist solutions for the equations

$$[\mathbf{p}_i^{(k+1)}]_* A^T \mathbf{u} = \mathbf{f}_i, \quad i = 1, 2, \cdots, M.$$

To seek a sparse set of coefficient $\mathbf{u}$, we need to minimize its ℓ_1-norm $\|\mathbf{u}\|_1$. Thus, we have to solve

$$\begin{array}{ll} \min\limits_{\mathbf{u}} & \|\mathbf{u}\|_1 \\ \text{subject to} & [\mathbf{p}_i^{(k+1)}]_* (A^T \mathbf{u}) = \mathbf{f}_i, \quad i = 1, 2, \cdots, M. \end{array} \tag{13}$$

The linearized Bregman iteration is a very efficient tool to solve the above minimization problem. Given $\mathbf{x}^{(0)} = \mathbf{v}^{(0)} = \mathbf{0}$, the linearized Bregman iteration generates a sequence of $\mathbf{x}^{(l)}$ and $\mathbf{v}^{(l)}$ as follows

$$\begin{cases} \mathbf{v}^{(l+1)} = \mathbf{v}^{(l)} - \sum_{i=1}^{M} A P_i^{(k+1)} [\mathbf{p}_i^{(k+1)}]_*^T \Big([\mathbf{p}_i^{(k+1)}]_* (A^T \mathbf{x}^{(l)}) - \mathbf{f}_i \Big), \\ \mathbf{x}^{(l+1)} = \frac{1}{\nu_2} T_{\mu_2}(\mathbf{v}^{l+1}). \end{cases} \tag{14}$$

Here T_{μ_2} is the soft-thresholding operator defined by

$$T_{\mu_2}(\mathbf{v}) = [t_{\mu_2}(v_1), t_{\mu_2}(v_1), \ldots], \quad \text{with} \quad t_{\mu_2}(v_i) = \operatorname{sign}(v_i) \max(|v_i| - \mu_2, 0),$$

and $P_i^{(k)}$ is a preconditioning matrix to accelerate the convergence of the iteration. Usually, we choose

$$P_i^{(k+1)} = \big([\mathbf{p}_i^{(k+1)}]_*^T [\mathbf{p}_i^{(k+1)}]_* + \tau_2 \Delta \big)^{-1}, \tag{15}$$

where Δ is the discrete Laplacian. Notice that the operators $P_i^{(k+1)}$ and $[\mathbf{p}_i^{(k+1)}]_*^T$ are commutable since they are all entrywise multiplications in the Fourier domain.

The basic idea of the linearized Bregman iteration (14) for finding a sparse solution is as follows. Two steps are involved in the linearized Bregman iteration. The first step is to find an approximate solution (a least squares solution in our case) to the residual equation of the constraint in (13) to update the data. However, the updated data may not be sparse. Therefore, the second step, a soft-thresholding operator, is applied to obtain a sparse framelet coefficients set. The procedure is repeated and it converges to a sparse solution in the framelet domain. The algorithm is efficient and robust to noises as analyzed by [5] and we also have the following convergence results from [5]. See [4, 5, 6, 32] for a more detailed analysis.

Proposition 1. *Assume that there exists at least one solution of* $\{[\mathbf{p}_i^{(k+1)}]_* (A^T \mathbf{u}) = \mathbf{f}_i, \ \forall \ i = 1, 2, \ldots, M\}$. *Then, there exists a constant* c *such that, for all* $\nu_2 \in (0, c)$, *the sequence* $\mathbf{x}^{(l)}$ *generated by* (14) *converges to the unique solution of*

$$\begin{aligned} \min_{\mathbf{u}} \quad & \|\mathbf{u}\|_1 + \tfrac{1}{2\mu_2\nu_2} \|\mathbf{u}\|_2^2 \\ \text{subject to} \quad & [\mathbf{p}_i^{(k+1)}]_* (A^T \mathbf{u}) = \mathbf{f}_i, \quad i = 1, 2, \cdots, M. \end{aligned} \tag{16}$$

Therefore, if we choose a sufficiently large thresholding parameter μ, then the iteration (14) converges to a solution of (13).

During the iterations of Algorithm 1, the intermediate results of the blur kernels are not accurate until the last few iterations. More importantly, there are alignment errors among the observed images. Thus, to obtain a good deblurred image, one can still use (13), but need to stop it early when the error of the constraint in (13) satisfies

$$\|[\mathbf{p}_i^{(k+1)}]_* (A^T \mathbf{u}) - \mathbf{f}_i\|_2^2 \le \delta_2^2, \quad i = 1, 2, \cdots M, \tag{17}$$

where δ_2^2 is an estimation of the variance of the image noises, the inaccuracy of the blur kernels, and the image alignment errors. The main reason is that the Bregman iteration has a good property: in the sense that the Bregman distance decreases, $\mathbf{x}^{(l)}$ approaches the tight frame coefficients of the true image until the residual in the iteration drops below the variance of the errors, as shown theoretically in [24, 32]. Furthermore, (14) is very robust to image noises and alignment errors in $\mathbf{f}_i$ ([5]).

In summary, in Step 2 of Algorithm 1, we use the linearized Bregman iteration (14) with the stopping criteria (17) to get a clear image. Usually, it takes only tens of iterations for (14) to get an image of satisfactory visual quality.

4.2. Method for Step 1 in Algorithm 1

In Step 1 of Algorithm 1, given the intermediate clear image $\mathbf{g}^{(k)}$, we want to compute the blur kernels $\{\mathbf{p}_i^{(k+1)}, i = 1, 2, \cdots M\}$. As shown in (2), a true motion blur kernel can be approximated well by a smooth function with the support of a continuous curve. It is observed that there are two essential properties of a "sound" motion blur kernel: one is the overall smoothness of the blur kernel; the other is its curvy support which implies its sparsity in spatial domain. Inspired by this observation, we model the motion blur kernel as a sparse solution in spatial domain with strong smoothness on its support. Thus, the proposed algorithm is to find the sparse blur kernels $\{\mathbf{p}_i^{(k+1)}, i = 1, 2, \ldots, M\}$ in spatial domain subject to certain smoothness constraints, which results in an ℓ_1 norm minimization problem. In order to further improve the accuracy and the efficiency of estimating the blur kernel, we use a weighted ℓ_1 norm instead the ordinary one.

Same as our previous discussions on Step 1, temporarily, we ignore the image alignment errors and assume that the clear image are accurate enough, such that there exist solutions for the equations

$$[\mathbf{g}^{(k)}]_* \mathbf{p}_i = \mathbf{f}_i, \quad i = 1, 2, \cdots, M.$$

Since a weighted ℓ_1-norm is minimized, we have to solve

$$\underset{\mathbf{p}_i}{\operatorname{argmin}} \|W_i \mathbf{p}_i\|_1, \quad \text{subject to} \quad [\mathbf{g}^{(k)}]_* \mathbf{p}_i = \mathbf{f}_i, \quad i = 1, 2, \cdots, M, \tag{18}$$

where W_i is the diagonal weighting matrix. Again, the linearized Bregman iteration can be applied to solve (18). The iteration is as follows, starting from $\mathbf{r}_i^{(0)} = \mathbf{q}_i^{(0)} = \mathbf{0}$,

$$\begin{cases} \mathbf{r}_i^{(l+1)} = \mathbf{r}_i^{(l)} - Q^{(k)} [\mathbf{g}^{(k)}]_*^T \Big([\mathbf{g}^{(k)}]_* \mathbf{q}_i^{(l)} - \mathbf{f}_i \Big), \\ \mathbf{q}_i^{(l+1)} = \frac{1}{\nu_1} W_i T_{\mu_1} \big((W_i)^{-1} \mathbf{r}_i^{l+1} \big). \end{cases} \tag{19}$$

Here T_{μ_1} is the soft-thresholding operator, $Q^{(k)}$ is a preconditioner matrix:

$$Q^{(k)} = \big([\mathbf{g}^{(k)}]_*^T [\mathbf{g}^{(k)}]_* + \tau_1 \Delta \big)^{-1},$$

where Δ is the discrete Laplacian. Again, $Q^{(k)}$ and $[\mathbf{g}^{(k)}]_*^T$ are commutable. Similar to Proposition 1, (19) gives a sparse solution and we have the convergence of (19).

Proposition 2. *Assume that there exists at least one solution of* $[\mathbf{g}^{(k)}]_*\mathbf{p}_i = \mathbf{f}_i,\ \forall\, i = 1, 2, \ldots, M$. *Then, there exists a constant* c *such that, for all* $\nu_1 \in (0, c)$, *the sequence* $\mathbf{q}_i^{(l)}$ *generated by* (19) *converges to the unique solution of*

$$\begin{array}{ll} \min\limits_{\mathbf{p}_i} & \|W_i\mathbf{p}_i\|_1 + \frac{1}{2\mu_1\nu_1}\|\mathbf{p}_i\|_2^2 \\ \text{subject to} & [\mathbf{g}^{(k)}]_*\mathbf{p}_i = \mathbf{f}_i, \quad i = 1, 2, \ldots, M. \end{array} \tag{20}$$

Therefore, if we choose a sufficiently large thresholding parameter μ_1, then the iteration (19) converges to a solution of (18).

During the iterations of Algorithm 1, the intermediate result of the clear image is not accurate and there are alignment errors among the observed images. Similar to the method for Step 2, we still use (19), but stop it early when the error of the constraint satisfies

$$\|[\mathbf{g}^{(k)}]_*\mathbf{q}_i^l - \mathbf{f}_i\|_2^2 \le \delta_1^2, \tag{21}$$

where δ_1^2 is an estimation of the variance of the errors caused by the noise, the inaccuracy of the clear image and the alignment errors. Therefore, $\mathbf{q}_i^{(l)}$ generated by (19), with a large μ_1 and stopping criteria (21), gives a good estimation of the blur kernel. From our empirical observations, only a couple of iterations already yield a satisfactory estimation on the blur kernel.

From Proposition 2, (19) minimizes the ℓ_2 norm of the blur kernel among all the minimal weighted ℓ_1 norm solutions (See [4, 5, 6] for details). We would like to explain more on why (19) is likely to yield a blur kernel which is a smooth function with a curvy support. In the first step of (19), the operation $Q^{(k)}[\mathbf{g}^{(k)}]_*^T\left([\mathbf{g}^{(k)}]_*\mathbf{q}_i^{(l)} - \mathbf{f}_i\right)$ essentially yields the solution of the following minimization:

$$\min_{\mathbf{p}_i} \frac{1}{2}\|[\mathbf{g}^{(k)}]_*\mathbf{p}_i - \mathbf{t}_i^{(k)}\|^2 + \tau_1\|\nabla\mathbf{p}_i\|^2,$$

where $\mathbf{t}_i^{(k)}$ is the residual of the i-th observed image at the k-th step satisfying

$$\mathbf{t}_i^{(k)} = [\mathbf{g}^{(k)}]_*\mathbf{q}_i^{(l)} - \mathbf{f}_i.$$

In other words, a Tikhonov regularization with the penalty $\|\nabla\mathbf{p}_i\|^2$ is applied in the preconditioning step. Thus, the smoothness of the estimated blur kernel is imposed to better constrain the smoothness of the blur kernel $\mathbf{p}_i$. The side effect is that the resulting blur kernel tends to spread out of the true support due to the smoothing. Therefore, the second step of (19) comes in to remove this side effect by finding a sparse solution with minimal weighted ℓ_1 norm from the previous one, which is done by applying a soft-thresholding operator as shown in (19). The support of the resulting sparse solution is then shrunken and likely to approximate the true support better than the previous one.

If only using the Tikhonov regularization, the resulting blur kernel will be a smooth function with the support of a large region; if only using the sparsity constraint, the resulting blur kernel will be very likely to converge to the degenerate case of only a few isolated points. By balancing the smoothness of the kernel using a Tikhonov regularization and the sparsity of the kernel in spatial domain using a weighted ℓ_1 norm, the sequence generated from (19) will converge as Proposition 2 proved. And the resulting solution will be close to the ideal motion blur kernel model, that is, a smooth function with the support of a continuous curve.

How to choose an appropriate W_i is dependent on the support of the true blur kernel $\mathbf{p}_i^*$. A good example of W_i is

$$W_i(m,m) = \begin{cases} \frac{1}{|\mathbf{p}_i^*(m)|}, & \text{if } \mathbf{p}_i^*(m) \neq 0 \\ \infty, & \text{otherwise;} \end{cases} \quad \text{and} \quad W_i(m,n) = 0, \text{ if } m \neq n,$$

for $j,k = 1, \cdots, N$. Unfortunately, It is impossible to construct such a W_i without knowing the blur kernel $\mathbf{p}_i$. In our approach, we take a simple algorithm which updates the weighting matrix W_is iteratively. That is, during the k-th iteration for the blur kernel estimation, we define the weighting matrix $W_i^{(k)}$ as such

$$W_i^{(k)}(m,n) = \begin{cases} (|([\mathbf{a}]_* \mathbf{p}_i^{(k)})(m)| + \epsilon)^{-1}, & \text{if } m = n \\ 0, & \text{otherwise,} \end{cases}$$

based on the value $\mathbf{p}_i^{(k)}$ obtained from the last iteration. Here $[\mathbf{a}]_*$ is the convolution with a local average kernel. The weighting matrix $W_i^{(k)}$ will be used in the next iteration. The parameter ϵ is to avoid numerical instability. It is observed empirically that such a weighting ℓ_1 norm can greatly speed up the algorithm.

4.3. The Complete Algorithm

Due to all types of noises and errors, the numerical algorithms do not always yield a solution which satisfies the physical rules of the image in the digital world. In order to obtain a physical solution, we also impose the following physical conditions:

$$\begin{cases} \mathbf{p}_i \geq 0, \quad \sum_j \mathbf{p}_i(j) = 1, \quad i = 1, 2, \cdots, M. \\ \mathbf{g} = A^t \mathbf{u} \geq 0. \end{cases} \tag{22}$$

The constraints (22) say that all pixel values of the recovered image has to be non-negative, and the estimation kernel should also be non-negative and its summation should be 1. It is noted that the physical constraint on the kernel $\mathbf{p}_i$ partially explains the reason why the regular ℓ_1 norm is not a good sparsity measurement on the kernel $\mathbf{p}_i$ because the ℓ_1 norm of the kernel $\mathbf{p}_i$ is always 1 if $\mathbf{p}_i$ satisfies the constraints (22). Combining all together, the complete algorithm for our blind deconvolution is described in Algorithm 2.

Algorithm 2 Complete algorithm for blind motion deblurring

1. Let $W_i^{(0)}$ and $\mathbf{g}^{(0)}$ be initial guess.
2. Iterate on k until convergence.
   (a) Fixing the clear image $\mathbf{g}^{(k)}$, iterate as (19) until (21) is satisfied. Then impose $\mathbf{p}_i^{(k+1/2)} = \mathbf{q}_i^{(\ell)}$, where ℓ is the minimal l such that (21) is satisfied.
   (b) Get the blur kernels by
   $$\mathbf{p}_i^{(k+1)} = \mathcal{P}(\mathbf{p}_i^{k+1/2}),$$
   where $\mathcal{P}$ is the projection operator, with an explicit expression, onto the set $\{\mathbf{p} : \mathbf{p}(j) \geq 0, \ \sum_j \mathbf{p}(j) = 1\}$.
   (c) Adjust the weightings W_i by
   $$W_i^{(k+1)}(j,\ell) = \begin{cases} \frac{1}{|([\mathbf{a}] * \mathbf{p}_i^{(k+1)})(j)| + \epsilon}, & \text{if } j = \ell \\ 0, & \text{otherwise.} \end{cases}$$
   (d) Fixing the blur kernels $\mathbf{p}_i^{(k+1)}$, $i = 1, 2, \ldots, M$, if $k \leq K$, get $\mathbf{g}^{k+1/2}$ by solving (12). Otherwise, iterate as (14) until (17) is satisfied; then impose $\mathbf{u}^{(k+1)} = \mathbf{x}^{(\ell)}$, where ℓ is the minimal l such that (17) is satisfied; set $\mathbf{g}^{(k+1/2)} = A^T \mathbf{u}^{(k+1)}$.
   (e) Get the clear image $\mathbf{g}^{(k+1)}$ by
   $$\mathbf{g}^{(k+1)}(j) = \begin{cases} \mathbf{g}^{(k+1/2)}(j), & \text{if } \mathbf{g}^{(k+1/2)}(j) \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$
   (f) $k = k + 1$.

(a) (b) (c)

Figure 3: (a) The original clear image. (b) One blurred image. (c) Another blurred image. The corresponding blur kernels are shown on the top left of the images respectively.

5. Experimental Evaluation and discussion

In our implementation, the initial diagonal elements of $W_i^{(0)}$ is set as 1 on those points falling in the center square of the image with size $\frac{n}{2} \times \frac{n}{2}$ and as $\frac{1}{\epsilon}$ otherwise, and the initial image $\mathbf{g}^{(0)}$ is one of the input images $\mathbf{f}_i$ for some i. Also, the maximum iteration number of Algorithm 2 is set to 100. The number K in Step 2(d) of Algorithm 2 is set about 2/3 of the maximum iteration number. The other parameters in Algorithm 2 are chosen empirically. All our experiments are done by running Matlab codes of our algorithm on a windows PC with an Intel 2G Hz CPU. Each iteration in Algorithm 1 takes roughly 18 seconds for all channels of the two input blurring color images with the resolution 1280×1024 pixels.

5.1. Simulated images

In the first experiment, we would like to see how robust the estimation of motion blur kernels in our proposed method is to the alignment error. The images used in this experiment is synthesized as follows. First, two blurred images (Fig. 3 (b) and (c)) are generated by applying two different blur kernels on the clear image (Fig. 3 (a)) respectively. Then the alignment error is added to the second blurred image (Fig. 3 (c)) by applying a spatial transform of (7) on the image with different rotations and scales (θ, s). The translation (t_x, t_y) always keeps the same (5 pixels shift along x-axis and 5 pixels shift along y-axis). Our proposed method is then applied to each pair of the blurred images in (Fig. 3(b) and Fig. 3(c)) with spatial distortions to recover the clear image and the blur kernels. Fig. 4 showed the intermediate results during the iterations of Algorithm 2 to illustrate the convergence behaviors of our algorithm. Fig. 5 (b) showed the estimated motion blur kernels from our method for different alignment errors.

It is clear that our proposed method can perfectly estimate the complicated blur kernels when the alignment is perfect. When there exist modest mis-alignments between two images, our method still can accurately estimate two blur kernels. The results shown in Fig. 5 (b) demonstrated that our method

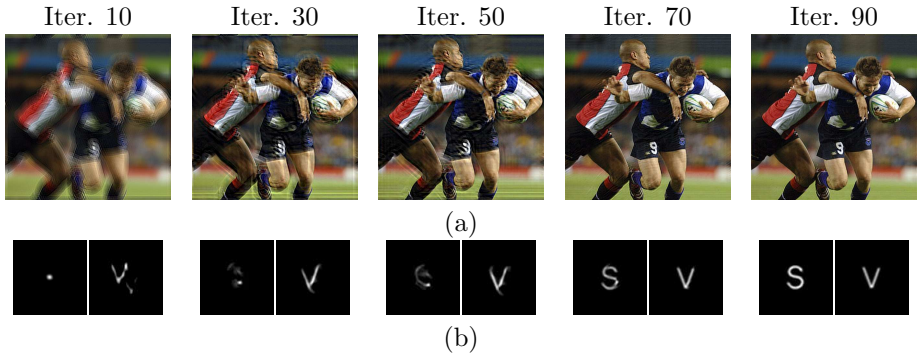


Figure 4: Five images in (a) are the recovered images during k-th iteration when applying our proposed algorithm on two blurred images in Fig. 3 (a)–(b). The corresponding estimated blur kernels are shown in (b).

is capable of finding complicated blur kernels and is robust to the modest alignment errors. For the comparison, we also estimates the blur kernels by least squares minimization with Tikhonov regularizations. Fig. 5 (a) showed that the standard approach cannot identify the motion blur kernel even there is no alignment error. Fig. 6 (a)-(c) showed a number of deblurred images using our proposed method under different alignment errors. As the comparison, Fig. 6 showed the deblurred image by least squares minimization method with Tikhonov regularization when there exist no alignment errors.

In the second experiment, we would like to evaluate the robustness of our method to image noises. All blurred images in this experiment are generated by applying two blur kernels on the original image, subsequently contaminated by zero mean white noise with different noise levels. Thirty two random samples are generated for each noise level. The noise level is measure by so-called *SNR* (signal to noise ratio) of the noised image $\widetilde{I}$ to the clean image I defined as

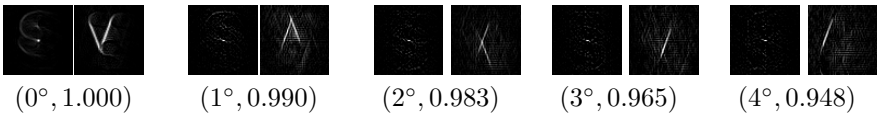
$$SNR(\widetilde{I}) = 20 \log_{10} \|I\|_2 / \|I - \widetilde{I}\|_2.$$

Fig. 7 showed that the estimation of the blur kernel by our method is also very robust to image noises.

5.2. Real images

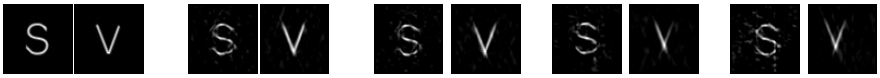
We also tested our method on real images taken by a handheld commodity digital camera with small hand trembles. All images are first automatically aligned by using the conventional image alignment method from [2] before applying our method.

Fig. 8 (a)-(b) and Fig. 9 (a)-(b) showed two blurred tele-photos on two indoor objects due to hand trembles. As the comparison, the recovered images from Algorithm 2 are compared against the results from the state-of-art blind motion deblurring technique ([14]) which utilizes the statistical prior on the



$(0°, 1.000)$ $(1°, 0.990)$ $(2°, 0.983)$ $(3°, 0.965)$ $(4°, 0.948)$

(a)



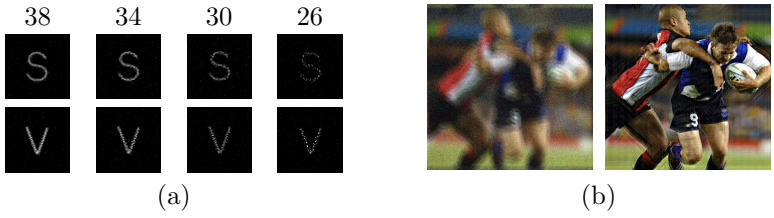
$(0°, 1.000)$ $(1°, 0.990)$ $(2°, 0.983)$ $(3°, 0.965)$ $(4°, 0.948)$

(b)

Figure 5: (a) The estimated kernels of the images in Fig. 3(b)-(c) using least square minimization with Tikhonov regularization. (b) The estimated kernels of the images in Fig. 3 (b)-(c) using our method. Two numbers in brackets under each pair of estimated kernels are the rotation angle parameter θ and the scale parameter s for the alignment error of the form (7).

$(0°, 1.000)$ (a) $(1°, 0.990)$ (b) $(2°, 0.983)$ (c) $(0°, 1.000)$ (d)

Figure 6: (a)–(c) are the deblurred images using the kernels from Algorithm 1. (d) is the deblurred image using the estimated kernel from the least squares method with Tikhonov regularization. Two numbers in brackets under each deblurred image are the rotation angle parameter and the scale parameter of (7).



(a) (b)

Figure 7: (a) The estimated kernels of Fig. 3(b) and (c) under various noise settings. The horizontal vector on the top is the SNR of the noisy images. (b) The left image is the noisy blurred image with $SNR = 26dB$ and the right image is the deblurred image.

image gradients to derive the motion blur kernel. It is seen that the restored image by Algorithm 2 shown in Fig. 8 (d) and Fig. 9 (d) are very clear with little artifacts. Obviously, they have much better visual quality than the images restored by the method from [14] which are shown in Fig. 8 (c) and Fig. 9 (c).

We also tested our method on out-door scenes. The blurred images on out-door scenes usually tend to be more difficult to deblur as there are multiple layers of blurring due to more complicated 3D structures, e.g., out-of-focus blurring and moving objects. Also, the complex image structure of typical outdoor scenes makes the deblurring process more challenging. Fig. 8 (a)-(b) and Fig. 9 (a)-(b) showed two blurred image pairs on two out-door scenes. We compared the results from Algorithm 2 to the results from more traditional cepstrum-based approach ([17]). Obviously, the results from Algorithm 2 are much better than those using the method from [17]. However, the restored images shown in Fig. 10 (d) and Fig. 11 (d) are less impressive than the previous results of in-door images. One reason is that the framelet coefficients of images with rich textures are not as sparse as those of images with less textures, which results in less robustness of our deblurring algorithm to image noises. Also, there are more noticable artifacts in Fig. 10 (d) than in Fig. 11 (d). The reason could be that the actual blurring for the case of Fig. 10 is the mixture of multiple blurring process and our model only focuses on the motion blurring. One evidence is that the estimated blur kernels shown in Fig. 10 (e) are not in the form of typical motion-blur kernels. Another possible reason could be that the blurring kernel of Fig. 10 is not spatial invariant due to the wind blowing the leaves during the camera exposure. This can be seen from the fact that the artifacts in Fig. 10 (e) have different directions for different leaves.

5.3. Conclusion and future work

Using multiple images not only improves the condition on deconvolution process, but also provides more information to help the identification of complicated motion blurring. However, the benefits of using multiple images can not be easily materialized by the standard approaches as the unavoidable image alignment errors could eliminate all the advantages of using multiple images. In this paper, we proposed an approach to recover the high-quality clear image by using multiple images to accurately identify motion blur kernels. By using the sparsity constraints on the images and on the blur kernels in suitable domains, the proposed approach is robust to the image formation noise and more importantly robust to the image alignment errors. Furthermore, based on the linearized Bregman iteration technique, we developed a fast approximate algorithm to find a good approximate solution to the resulting large-scale minimization problem very efficiently.

Our proposed method does not require a prior parametric model on the motion blur kernel, and does not require accurate image alignment among frames. These two properties greatly extend the applicability of motion deblurring on general video sequences in practice. In future, we would like to investigate the localization of our algorithm on spatial-variant motion blurring such as deblurring fast-moving objects in the image. Also, we are interested in investigating

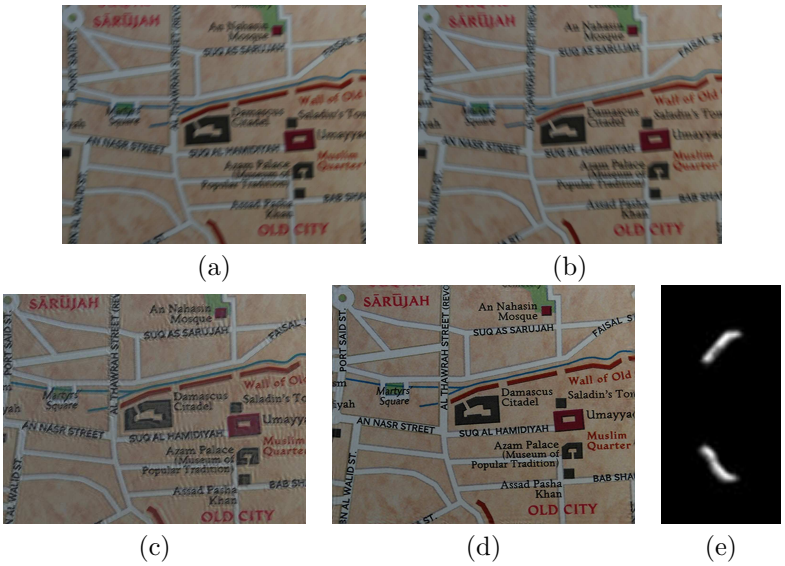


Figure 8: (a)–(b): two blurred images; (c): the recovered image using the method in [14]; (d): the deblurred image using Algorithm 2; (e): the two blur kernels estimated by Algorithm 2 w.r.t. (a) and (b).

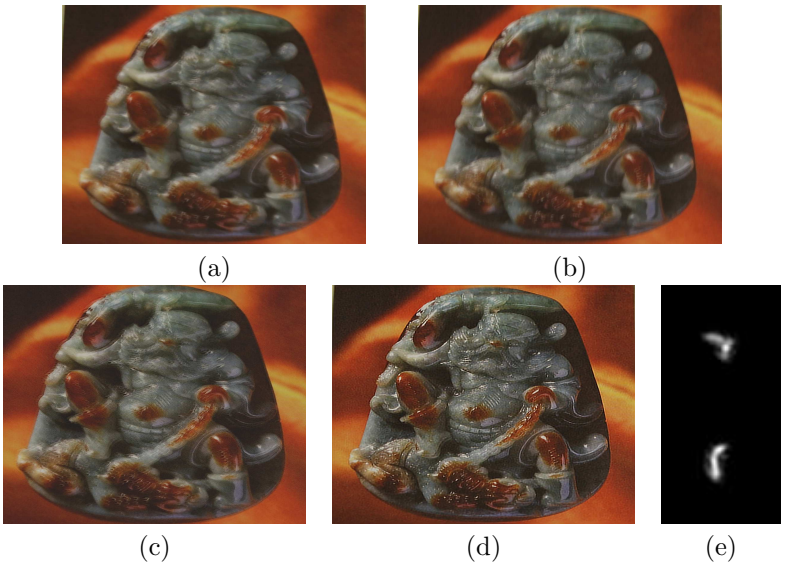


Figure 9: (a)–(b): two blurred images; (c): the recovered image using the method in [14]; (d): the deblurred image using Algorithm 2; (e): the two blur kernels estimated by Algorithm 2 w.r.t. (a) and (b).

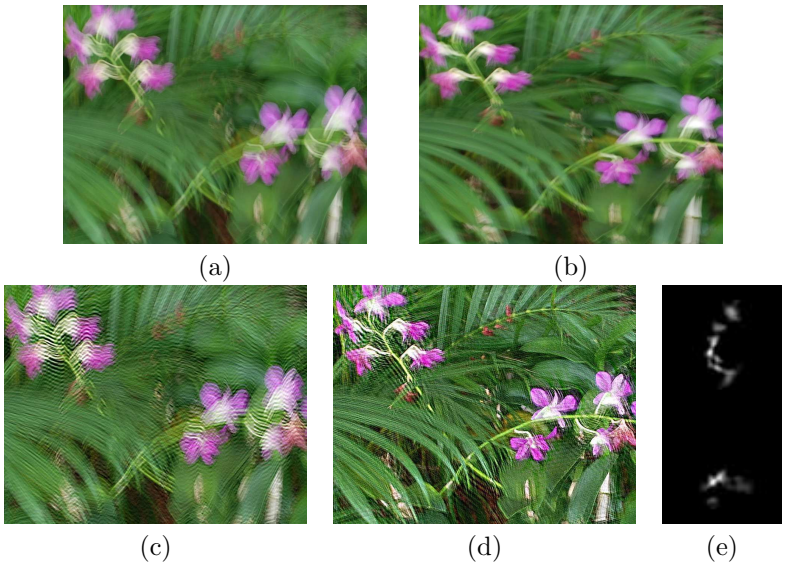


Figure 10: (a)–(b): two blurred images; (c): the recovered image using the newest cepstral method ([17]); (d): the deblurred image using Algorithm 2; (e): the two blur kernels estimated by Algorithm 2 w.r.t. (a) and (b).

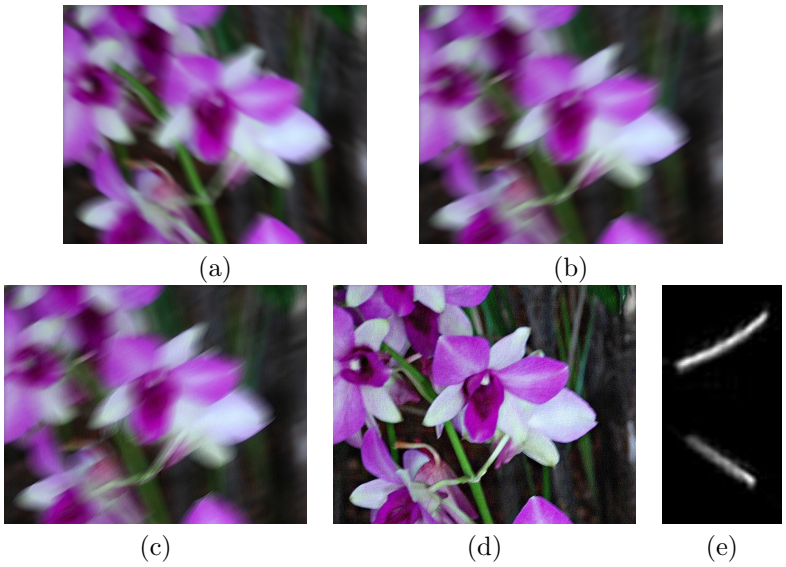


Figure 11: (a)–(b): two blurred images; (c): the recovered image using the newest cepstral method ([17]); (d): the deblurred image using Algorithm 2; (e): the two blur kernels estimated by Algorithm 2 w.r.t. (a) and (b).

how to incorporate the image alignment of blurred image into the proposed minimization to achieve even better performance.

References

[1] H. C. Andrews, B. R. Hunt, Digital image restoration, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[2] J. Bergen, P. Anandan, K. Hanna, R. Hingorani, Hierarchical model-based motion estimation, in: ECCV, 1992.

[3] M. Ben-Ezra, S. K. Nayar, Motion-based motion deblurring, IEEE Trans. Pattern Aanalysis and Machine Intelligence 26 (6) (2004) 689–698.

[4] J.-F. Cai, S. Osher, Z. Shen, Linearized Bregman iterations for compressed sensing, Mathematics of Computation 78 (2009) 1515–1536.

[5] J.-F. Cai, S. Osher, Z. Shen, Linearized Bregman iterations for frame-based image deblurring, SIAM J. Imaging Sci. 2 (1) (2009) 226–252.

[6] J.-F. Cai, S. Osher, Z. Shen, Convergence of the linearized Bregman iteration for ℓ_1-norm minimization, Mathematics of Computation 78 (2009) 2127–2136.

[7] A. Chai, Z. Shen, Deconvlolution: A wavelet frame approach, Numer. Math. 106 (2007) 529–587.

[8] T. F. Chan, J. Shen, Image processing and analysis, Variational, PDE, wavelet, and stochastic methods, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

[9] T. F. Chan, C. K. Wong, Total variation blind deconvolution, IEEE Tran. Image Processing 7 (3) (1998) 370–375.

[10] M. M. Chang, A. M. Tekalp, A. T. Erdem, Blur identification using the bispectrum, IEEE transactions on signal processing 39 (1991) 2323–2325.

[11] J. Darbon, S. Osher, Fast discrete optimization for sparse approximations and deconvolutions, in: preprint, 2007.

[12] I. Daubechies, B. Han, A. Ron, Z. Shen, Framelets: MRA-based constructions of wavelet frames, Appl. Comput. Harmon. Anal. 14 (2003) 1–46.

[13] D. C. Dobson, F. Santosa, Recovery of blocky images from noisy and blurred data, SIAM J. Appl. Math. 56 (1996) 1181–1198.

[14] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, W. T. Freeman, Removing camera shake from a single photograph, SIGGRAPH, 25 (2006) 783–794.

[15] T. Goldstein, S. Osher, The split Bregman algorithm for l1 regularized problems, UCLA CAM Reports (08-29).

[16] L. He, A. Marquina, S. J. Osher, Blind deconvolution using TV regularization and Bregman iteration, Int. J. Imaging Syst. Technol. 15 (2005) 74–83.

[17] H. Ji, C. Q. Liu, Motion blur identification from image gradients, in: Proceeding of IEEE Computer Society conference on computer vision and pattern recognition, (2008) 1–8.

[18] Y. Lou, X. Zhang, S. Osher, A. Bertozzi, Image recovery via nonlocal operators, UCLA CAM Reports (08-35).

[19] Y. Lu, J. Sun, Q. L., H. Shum, Blurred/non-blurred image alignment using kernel sparseness prior, IEEE Int. Conf. on Computer Vision (2007).

[20] A. Marquina, Inverse scale space methods for blind deconvolution, UCLA CAM Reports (06-36).

[21] C. Mayntz, T. Aach, Blur identification using a spectral inertia tensor and spectral zeros, IEEE transacions on image processing 2 (1999) 885–559.

[22] M. K. Ng, R. H. Chan, W. Tang, A fast algorithm for deblurring models with neumann boundary condition, SIAM J. Sci. Comput. 21 (3) (2000) 851–866.

[23] M. Nikolova, Local strong homogeneity of a regularized estimator, SIAM J. Appl. Math. 61 (2000) 633–658.

[24] S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin, An iterative regularization method for total variation-based image restoration, Multiscale Model. Simul. 4 (2005) 460–489.

[25] S. Osher, Y. Mao, B. Dong, W. Yin, Fast linearized Bregman iteration for compressed sensing and sparse denoising, UCLA CAM Reprots (08-37).

[26] A. Rav-Acha, S. Peleg, Two motion blurred images are better than one, Pattern Recognition Letters 26 (2005) 311–317.

[27] R. Raskar, J. Tubmlin, A. Mohan, A. Agrawal, Y. Li, Computational photography, EUROGRAPHICS (2006).

[28] R. Raskar, A. Agrawal, J. Tumblin, Coded exposure photography: Motion deblurring via fluttered shutter, SIGGRAPH 25 (2006) 795–804.

[29] I. M. Rekleitis, Steerable filters and cepstral analysis for optical flow calculation from a single blurred image, Vision Interface 1 (1996) 159–166.

[30] A. Ron, Z. Shen, Affine system in $l_2(r^d)$: the analysis of the analysis operator, Journal of Functional Analysis 148 (1997) 408–447.

[31] J. Xu, S. J. Osher, Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising, IEEE Trans. Image Process. 16 (2007) 534–544.

[32] W. Yin, S. Osher, D. Goldfarb, J. Darbon, Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing, SIAM J. Imaging Sci. 1 (2008) 143–168.

[33] Y. Yitzhaky, I. Mor, A. Lantzman, N. S. Kopeika, Direct method for restoration of motion-blurred images, J. Opt. Soc. Am. 15 (6) (1998) 1512–1519.

[34] B. Zitova, J. Flusser, Image registration methods: a survey, Image and Vision Computing 21 (11) (2003) 977–1000.