# Framelet based blind motion deblurring from a single image

Jian-Feng Cai, Hui Ji, Chaoqiang Liu and Zuowei Shen

***Abstract*—How to recover a clear image from a single motion-blurred image has long been a challenging open problem in digital imaging. In this paper, we focus on how to recover a motion-blurred image due to camera shake. A regularization-based approach is proposed to remove motion blurring from the image by regularizing the sparsity of both the original image and the motion-blur kernel under tight wavelet frame systems. Furthermore, an adapted version of the split Bregman method ( [1], [2]) is proposed to efficiently solve the the resulting minimization problem. The experiments on both synthesized images and real images show that our algorithm can effectively remove complex motion blurring from natural images without requiring any prior information of the motion-blur kernel.**

***Index Terms*—tight frame, split Bregman method, motion blur, blind deconvolution**

***EDICS Category*— TEC-MRS, TEC-RST**

## I. Introduction

Motion blurring is one of the prime causes of poor image quality in digital imaging. When an image is captured by a digital camera, the image represents not just the scene at a single instant of time, but the scene over a period of time. If objects in a scene are moving fast or the camera is moving over the period of *exposure* time, the objects or the whole scene will look blurry along the direction of relative motion between the object/scene and the camera. Camera shake is one main cause of motion blurring, especially when taking images using telephoto lens or using long shutter speed under low lighting condition.

In the past, many researchers have been working on motion deblurring which recovers clear images from motion-blurred images. In most works, the motion blur caused by camera shake is modeled by a spatial-invariant convolution process:

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

where "$*$" is the discrete convolution operator, $g$ is the original image to recover, $f$ is the observed blurry image, $p$ is the blur kernel (or *point spread function*), and $\eta$ is the noise. How to recover the original image $g$ from the observed image $f$ is the so-called *image deconvolution* problem. Based on the availability of $p$, there are two categories of image deconvolution problems. If the blur kernel $p$ is given as a prior, recovering the original image becomes a *non-blind deconvolution* problem. Non-blind deconvolution is known as an ill-conditioned inverse problem as a small perturbation of $f$ may cause the direct solution from (1) being heavily distorted. In the past, there have been extensive research literatures on robust non-blind deconvolution algorithm (e.g. [1], [3]–[7]). If the blur kernel $p$ is also unknown, how to reverse the effect of convolution by $p$ on the blurred image $f$ is then a *blind deconvolution* problem. In general, blind deconvolution is a very challenging ill-conditioned and ill-posed inverse problem because it is not only sensitive to image noise but also under-constrained with infinitely many solutions. Removing motion blurring from images is a typical blind deconvolution problem, as the relative motion between the camera and the scene varies for individual images.

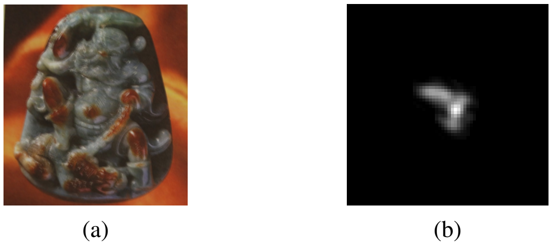


Fig. 1. (a): a motion-blurred image of size $1280 \times 1024$; (b) the image of blur kernel of (a) with image size $64 \times 64$. Both images are from [8].

Certain priors on both the blur kernel $p$ and the original image $g$ have to be assumed to overcome the ill-posedness of motion deblurring. The motion-blur kernel is quite different from some other types of blur kernels, e.g., out-of-focus blur kernel and Gaussian optical blur kernel, as motion-blur kernel can not be represented by some simple parametric form. In this paper, we assume that only significant motion of the camera is a translation along image plane and that the scene being photographed is static. Let $f$ denote the true sharp image and let $g$ denote the observed blurry image due to camera shake. Then, the relationship between $f$ and $g$ is a convolution process (1) with the blur kernel $p$ that vanishes out of the camera motion trajectory during exposure time. See Fig. 1 (b) for the illustration of a real motion-blur kernel $p$. Briefly, the motion-blur kernel $p$ is approximately a smooth function with its support on a continuous curve in the image plane. From this perspective, motion deblurring is a challenging blind deconvolution problem as the motion-blur kernels can not be characterized easily by some parametric functional. As a result, it requires a significant number of unknowns to represent motion-blur kernels.

J.-F. Cai is with Department of Mathematics, University of California, Los Angeles, CA 90095, USA (email: cai@math.ucla.edu).

H. Ji, C. Liu and Z. Shen are with Department of Mathematics, National University of Singapore, Singapore 117543, Singapore (email: matjh, matliucq, matzuows@nus.edu.sg).

### A. Previous work on blind deconvolution

In the past, there have been extensive research works on single-image blind deconvolution. Early works on blind deblurring usually use a single image and assume a prior parametric form of the blur kernel $p$, such as linear motion blur kernel model (e.g. [9]). These parametric motion-blur kernel models can be obtained by estimating only a few parameters, but they are often overly simplified for practical motion blurring. To remove more general motion blurring from images, some probabilistic priors on natural images' edge distributions have been proposed to derive the blur kernel (e.g., [10]–[13]). One weakness of these methods is either that the assumed probabilistic priors do not always hold true for natural images or that it needs certain user interactions to obtain an accurate estimation. It is noted that there also have been active researches on multi-image based blind motion deblurring methods as multiple images provide more information of the scene and could lead to an easier configuration for accurately estimating blur kernels. Interested readers are referred to [8], [14]–[19] for more details.

An alternative approach is to formulate the blind deconvolution as a joint minimization problem to simultaneously estimate both the blur kernel and the clear image. To overcome the inherent ambiguities between the blur kernel $p$ and the clear image $g$, certain regularization terms on both $p$ and $g$ have to be added in the minimization, which results in the following minimization formulation:

$$E(p, q) = \min_{p,g} \Phi(g * p - f) + \lambda_1 \Theta_1(g) + \lambda_2 \Theta_2(p), \quad (2)$$

where $\Phi(p * g - f)$ is the fidelity term, $\Theta_1(g)$ and $\Theta_2(p)$ are the regularization terms on the clear image and on the blur kernel respectively. Early regularization-based methods assume the smooth constraints on images and kernels. One such regularization (e.g. [20]) is to use the square $\ell_2$ norm of image/kernel derivatives as the regularization term on the image/kernel, which is also the so-called Tikhonov regularization method. The variational approach is proposed in [21]) which also assumes the smooth prior of both images and kernels by considering Gaussian distribution priors. Moreover, the parameters involved in the regularization are also automatically inferred in [21] by using the conjugate hyperpriors on parameters.

In recent years, TV (Total Variation) and its variations have been popular choices of the regularization term in recent years to solve various blind deblurring problems (e.g., [5], [22]–[26]). These TV-based blind deconvolution techniques showed good performance on removing certain types of blurrings on specific types of images, such as out-of-focus blurring on medical images and satellite images. However, TV regularization is not the optimal choice for removing motion-blurring, because TV regularization penalizes, e.g., the total length of the edges for piecewise constant functions (see [5]). As a result, the support of the resulting blur kernel tends to be a disk or several isolated disks. A more sophisticated TV-norm related model is presented in [27] with good performances on removing modest motion blurring from images without rich textures. Also, it is dependent on the accurate input of some prior information of the blur kernel. The main limitation of TV-based regularization for nature images is TV-based regularizations do not preserve the details and textures very well on the regions of complex structures due to the stair-casing effects (see e.g. [28], [29]).

Another type of regularization techniques for blind deconvolution is using various sparsity-based priors to regularize images, kernels or both of them. Considering a smooth blur kernel, a quasi maximum-likelyhood approach is proposed in [30] for de-convolute both sparse images and nature images which are sparsified in [30] through a sparsifying kernel learned from training data. Based on a Bayesian approach, a sparsity-based prior on kernel is proposed in [31] that assumes the kernel can be represented by a weighted combination of Gaussian-type basis functions with weights satisfying a heavy tailed student's-t distribution. The regularization on images is also based on the assumption that the image differences satisfy a heavy tailed student's-t distribution.

### B. Our approach and most related works

In this paper, we propose a new optimization approach to remove complex motion blurring from a single image by introducing new sparsity-based regularization terms on both images and motion-blur kernels. Our approach is closely related to recent works on both non–blind image deconvolution ( [1], [32]) and blind motion deblurring ( [33]).

Two non-blind image deconvolution algorithms in [1], [32] are both based on the observation that images usually have sparse representations or sparse approximations in some redundant transformed domains, e.g., wavelet ( [34]) and framelet ( [35], [36]) transforms. Given the blur kernel $p$, (1) is solved in [1], [32] by seeking a sparse solution in the corresponding transformed domain. The main difference between two methods lies in the different approaches to enforce the sparsity prior: one is using the so-called *synthesis-based* sparsity prior ( [32]) and the other is using the so-called *analysis-based* sparsity prior ( [1]). Here we give a brief explanation of two sparsity priors in terms of deconvolution. Interested readers are referred to [1], [37] for more detailed discussions.

For simplicity, we denote images as vectors in $\mathbb{R}^n$ by concatenating their columns. Let $\mathcal{D} \in \mathbb{R}^{m\times n}$ be an *analysis* operator which decomposes the data $\mathbf{g} \in \mathbb{R}^n$ to some transform coefficients $\mathcal{D}\mathbf{g} \in \mathbb{R}^m$. Let $\mathcal{R} \in \mathbb{R}^{m\times n}$ be an *synthesis* operator which synthesizes the data $\mathbf{g} \in \mathbb{R}^n$ from some transform coefficients $\mathbf{u} \in \mathbb{R}^m$. If $R$ is the left inverse of $\mathcal{D}$, $\mathcal{R}\mathcal{D} = I$, we have the perfect reconstruction formula

$$\mathbf{g} = \mathcal{R}\mathbf{u} = \mathcal{R}(\mathcal{D}\mathbf{g}).$$

Then, the approach using the synthesis-based sparsity prior gives the solution $\bar{\mathbf{g}}$ as following:

$$\bar{\mathbf{g}} := \mathcal{R}\bar{\mathbf{u}}; \quad \bar{\mathbf{u}} := \mathrm{argmin}_{\mathbf{u}\in\mathbb{R}^m} \Phi(\mathbf{p} * \mathbf{g} - \mathbf{f}) + \lambda\|\mathbf{u}\|_1, \quad (3)$$

where $\Phi$ is the fidelity term, $\lambda$ is the regularization parameter and $\mathbf{p}$ is the blur kernel. The approach using the analysis-based sparsity prior yields the following solution:

$$\bar{\mathbf{g}} := \mathrm{argmin}_{\mathbf{g}\in\mathbb{R}^n} \Phi(\mathbf{p} * \mathbf{g} - \mathbf{f}) + \lambda\|\mathcal{D}\mathbf{g}\|_1. \quad (4)$$

or equivalently,

$$\mathbf{\bar{g}} := \mathcal{R}\mathbf{\bar{u}}; \quad \mathbf{\bar{u}} := \mathrm{argmin}_{\mathbf{u}\in\mathrm{range}(\mathcal{D})}\, \Phi(\mathbf{p}*\mathbf{g}-\mathbf{f})+\lambda\|\mathbf{u}\|_1 \quad (5)$$

as $\mathcal{R}\mathcal{D} = I$. Two minimizations (3) and (5) are equivalent only if $\mathrm{range}(\mathcal{D}) = \mathbb{R}^m$, i.e., $\mathcal{D}$ (or $\mathcal{R}$) is an invertible square matrix with $m = n$. When the analysis operator $\mathcal{D}$ is a redundant transform (e.g., tight frame transform) whose row dimension $m$ is larger than its column dimension $n$, the two minimizations gives different solutions. As the domain of unknowns in (5) is only a strict subset of the domain of unknowns in (3), the synthesis-based approach (3) is seeking for the most sparse solution among all transform coefficient vectors while the analysis sparsity based minimization (5) (or (4)) is seeking for the most sparse solution only among the canonical framelet coefficient vector (transform coefficient vector decomposed from some images). Since the weighted norm of canonical framelet coefficients are closely related to the smoothness of the underlying function ( [38]), the result from the analysis-based approach (5) will give a less sparse but smoother solution than that of the synthesis-based approach (3). It is observed in extensive experiments that the deblurred result (5) with certain smoothness tends to have better visual quality than that from (3) which often has visible artifacts along image edges.

In [33], the sparsity prior under redundant tight frame system is used to remove motion blurring from natural images for the first time. In [33], the sparsity prior of images under framelet ( [35], [36]) domain and the sparsity prior of motion-blur kernels under curvelet ( [39]) domain are used to regularize both images and kernels. Both sparsity priors are enforced by the synthesis-based approach. Although impressive results have been demonstrated in [33], there are still rooms for further improvements, especially on reducing the artifacts of the estimated original images and better estimation of more complex motion-blur kernels. Motivated by the benefits gained in the application of non-blind deconvolution by using the analysis-based sparsity prior ( [1]), we also adopt the analysis based sparsity prior of images under suitable tight frame system to regularize images.

Regarding motion-blur kernel, we take a different approach from [33] to regularize it. Based on the assumption that the motion-blur kernel can be approximated by a smooth function with the support close to a continuous "thin" curve, we propose a mixed regularization strategy for blur kernel which includes both analysis based sparsity prior and $\ell_2$ norm regularization of blur kernels under a certain tight frame system. In our implementation, framelet system ( [35], [36]) is chosen to represent both original images and motion-blur kernels for its implementation simplicity and computational efficiency. Also, the minimization (5) is a more challenging problem to solve than (3). The linearized Bregman iteration ( [25], [26]) used in [33] for solving (3) is not applicable to (5). Thus, in this paper, we introduce another efficient solver for (5), i.e., the split Bregman method (See [1], [2]).

The rest of the paper is organized as follows. In Section 2, we formulate the minimization model and the corresponding algorithms. Section 3 is devoted to the experimental evaluation and the discussion on future work.

## II. Formulations and algorithms

It is known that non-blind deconvolution is an ill-conditioned problem as it is sensitive to noise, that is, a small perturbation of $f$ may lead to a large distortion on the direct solution of (1). Extensive studies have been done along the line of developing algorithms robust to noise. Imposing some regularization terms is proven to be an effective approach. However, blind deblurring is a much more challenging problem as it is also an under-constrained problem. Mathematically, there exists infinitely many solutions to (1).

There is one type of degenerate solutions $(\widetilde{g}, \widetilde{p})$ of (1) which bothers many existing blind deconvolution methods:

$$\widetilde{g} = g * h; \quad \widetilde{p} = p * h^{-1}, \tag{6}$$

where $h$ is some low-pass/high-pass filter. In such a case, the de-blurred image will be either over-deblurred when $h$ being a high-pass filter or less-deblurred when $h$ being a low-pass filter. The extreme case of less-deblurring is

$$\widetilde{g} := f; \quad \widetilde{p} := \delta, \tag{7}$$

where the image is not deblurred at all. To overcome such ill-posedness of blind deconvolution, certain priors on both images and kernels should be enforced by adding corresponding regularization terms in the minimization. And one main role of these regularization terms is to guarantee that the solution generated by the algorithm does not fall into the degenerate case.

In the remaining of this section, we will introduce a new approach to solve (1) with analysis-based sparsity priors on both images and kernels under some suitable tight frame systems. In our approach, we choose framelet system ( [35], [36]) as the frame system to represent both original images and blur kernels. Before presenting our formulation on blind motion deblurring, we first give a brief introduction to framelet system and interested readers are referred to [6], [40], [41] for more implementation details.

### A. Wavelet tight frame and image representation

The wavelet tight frames used here are mainly in two variable setting, however, for simplicity, we only present wavelet tight frames in the univariate setting, since we use tensor product wavelet frames in the implementation. A countable subset of $X \subset L_2(\mathbb{R})$ is called a *tight frame* of $L_2(\mathbb{R})$ if

$$f = \sum_{x\in X} \langle f, x\rangle x, \quad \forall\ f \in L_2(\mathbb{R}). \tag{8}$$

This is equivalent to

$$\|f\|^2 = \sum_{x\in X} |\langle f, x\rangle|^2, \quad \forall\ f \in L_2(\mathbb{R}),$$

where $\langle f, g\rangle$ and $\|f\|$ denote the inner product and the norm of $L_2(\mathbb{R})$ for any two functions $f, g \in L_2(\mathbb{R})$ respectively. Tight frame, as a generalization to orthonormal basis, relaxes the requirement of $X$ being a basis of $L_2(\mathbb{R})$ and brings in redundancy that has been proved useful in many applications in signal and image processing (see e.g. [34], [41]). Since tight frame is redundant, there are an infinite number of possible expansions of $f$ in the system $X$. The particular expansion given in (8) is called the canonical expansion, and $\{\langle f, g\rangle\}$ is the canonical frame coefficient sequence.

A wavelet system $X(\Psi)$ is defined to be a collection of dilations and shifts of a finite set $\Psi = \{\psi^1, \ldots, \psi^r\} \subset L_2(\mathbb{R})$,

$$X(\Psi) := \{\psi_{j,k} := 2^{j/2}\psi(2^j x - k),\ \ j \in \mathbb{Z},\ k \in \mathbb{Z}, \psi \in \Psi\}.$$

When $X(\Psi)$ forms a tight frame, it is called a *wavelet tight frame* and each $\psi \in \Psi$ is called a *framelet*. To construct compactly supported wavelet tight frames, one usually starts from a compactly supported refinable function $\phi$ (called a scaling function) with a refinement mask $h_0$ satisfying

$$\widehat{\phi}(2\omega) = h_0 \widehat{\phi}(\omega),$$

where $\widehat{\phi}$ is the Fourier transform ( [34]) of $\phi$, and $h_0$ is a $2\pi$-periodic trigonometric polynomial with $h_0(0) = 1$. For a given compactly supported refinable function $\phi$, the construction of a wavelet tight frame is to find an appropriate set of framelets $\Psi = \{\psi^1, \ldots, \psi^r\}$ defined in the Fourier domain by

$$\widehat{\psi^i}(2\omega) = h_i \widehat{\phi}(\omega), \quad i = 1, 2, \ldots, r,$$

where the framelet masks $h_i$'s are $2\pi$-periodic trigonometric polynomials. The Unitary Extension Principle (UEP) of [35] says that $X(\Psi)$ forms a tight frame provided that

$$h_0(\omega)\overline{h_0(\omega + \gamma\pi)} + \sum_{i=1}^{r} h_i(\omega)\overline{h_i(\omega + \gamma\pi)} = \delta_{\gamma,0}, \quad \gamma = 0, 1.$$

As an application of UEP, a family of wavelet tight frame systems is derived in [35] by using uniform B-splines ( [42]) as the refinable function $\phi$. The simplest system in this family is piecewise linear B-spline tight frame which uses piecewise linear B-spline function as $\phi$. This $\phi$ has the refinement mask $h_0 = \cos^2(\frac{\omega}{2})$, and the corresponding low-pass filter is

$$\mathbf{h}_0 = \frac{1}{4}[1, 2, 1],$$

Two framelets $\psi_1, \psi_2$ are defined by the framelet masks $h_1 = -\frac{\sqrt{2}i}{2}\sin(\omega)$ and $h_2 = \sin^2(\frac{\omega}{2})$, whose corresponding high-pass filters are

$$\mathbf{h}_1 = \frac{\sqrt{2}}{4}[-1, 0, 1], \quad \mathbf{h}_2 = \frac{1}{4}[-1, 2, -1]. \tag{9}$$

Numerical computation of the wavelet frame transform is done by using the wavelet frame decomposition algorithm given in [36]. In fact, we use the decomposition algorithm without down-sampling. This can be easily implemented by using the refinement and framelet masks. The transform can be represented by a matrix $W$ whose construction depends on the boundary conditions. In this paper, we use the Neumann (symmetric) boundary condition. Since [6], [40] have given details of how to generate such matrices, we omit the detailed discussions here and the interested reader should consult [6], [40] for details.

With the matrix $W$, it is easy to describe the transformation process. Let $\mathbf{g}$ be a vector of the image after column concatenation, the frame coefficient vector $\mathbf{u}$ can be computed via

$$\mathbf{u} = W\mathbf{g}.$$

Once we have $W$, the inverse transform, i.e. the reconstruction algorithm, is $W^T$, i.e.

$$\mathbf{g} = W^T\mathbf{u}.$$

It is very important to note that $W$ can be constructed from the refinement and framelet masks such that $W^T W = I$. The rows of the matrix $W$ form a tight frame in a finite dimensional space which connects well to the wavelet frame system in function spaces (see e.g. [36] for details). In general, since there are more rows than columns in $W$, $WW^T \neq I$. When $WW^T = I$, then the rows of $W$ form an orthonormal basis. Finally we remark that in practical computation, we are working in the bivariate setting. We employ the tensor product of 1D wavelet tight frame, and the corresponding matrix $W$ can be constructed easily via the Kronecker product of the matrix constructed by univariate wavelet frame transform (see [6] for details). In the rest of this paper, we still use $W$ to denote the discrete transform generated by bivariate wavelet tight frame. We further note that $W$ is basically used for notational convenience. In the real computation, we do not use matrix multiplication. Instead, we use a wavelet decomposition and reconstruction algorithm directly modified from [36].

### B. Formulation of our minimization model

Given a blurred image $f$ satisfying the relationship (1):

$$f = p * g + \eta.$$

we take a regularization-based approach to solve the blind motion deblurring problem, which requires the simultaneous estimations of both the original image $g$ and the blur kernels $p$. It is well known that the regularization-based blind deconvolution approach usually results in solving a challenging non-convex minimization problem. The most commonly used approach is an alternative iteration scheme; see [22] for instance. Let $p^{(0)}$ be the initial guess on the blur kernel, the alternative iteration scheme is described in Algorithm 1.

**Algorithm 1** Outline of the alternative iterations

```
For $k = 0, 1, \ldots$,
  1) given the blur kernels $p^{(k)}$, compute the clear image $g^{(k+1)}$:
       $g^{(k+1)} := \mathrm{argmin}_g \frac{1}{2}\|p^{(k)} * g - f\|_2^2 + \lambda_1\Theta_1(g),$   (10)
     where $\Theta_1(\cdot)$ is the regularization term on images and $\lambda_1$ is the corresponding regularization parameter.
  2) given the clear image $g^{(k+1)}$, compute the blur kernels $p^{(k+1)}$;
       $p^{(k+1)} := \mathrm{argmin}_p \frac{1}{2}\|g^{(k+1)} * p - f\|_2^2 + \lambda_2\Theta_2(p),$   (11)
     where $\Theta_2(\cdot)$ is regularization term on kernels and $\lambda_2$ is the corresponding regularization parameter.
```

There are two steps in Algorithm 1 and both steps are about using regularization-based approach for non-blind deconvolution. Step 1 is a non-blind image deblurring problem, which has been studied extensively in the literature; see, for instances, [3]–[7], [32]. However, there are subtle differences between Step 1 and the classic non-blind deconvolution problems, that is, the intermediate estimated blur kernel $p^{(k+1)}$ used for deblurring in Step 1 is not perfect and it is far way from the truth during the initial iterations. Inspired by the strong noise robustness of the recent non-blind deblurring technique ( [1]), we also use the analysis sparsity prior on the original image $g$ under framelet system to regularize the non-blind image deblurring to alleviate the distortion caused by erroneous intermediate estimate of the blur kernel. Thus, we propose the following regularization term in (10) of Step 1:

$$\Theta_1(\mathbf{g}) = \|W\mathbf{g}\|_1, \tag{12}$$

where $\mathbf{g}$ is the vectorized form of image $g$ and $W$ is the framelet transform given in last section.

Step 2 is also a non-blind deconvolution problem but the data to recover is the motion-blur kernel $p$. Motion-blur kernel can also be viewed as an image but with unique image content. See Fig. 1 for one sample motion-blur kernel visualized as an image. It is observed that there are two essential constraints of a function being a "sound" motion-blur kernel. One is its curvy support which implies its sparsity in spatial domain; the other is the continuity of its support and the smoothness of the kernel along its support. We propose to translate these two constraints to the following regularization term in (11) of Step 2:

$$\Theta_2(\mathbf{p}) = \|W\mathbf{p}\|_1 + \frac{\tau}{2}\|\mathbf{p}\|_2^2, \tag{13}$$

where $\mathbf{p}$ is the vectorized form of kernel $p$, $W$ is the same framelet transform as that in (12) and $\tau$ is the parameter which balances the sparsity of the blur kernel and the continuity of the support of the blur kernel.

The motivation of the proposed regularization term (13) on motion blur kernels is explained as follows. There are two components in the regularization term (13): one is the analysis based sparsity regularization under framelet domain $\|W\mathbf{p}\|_1$ and the other is the $\ell_2$ norm regularization $\|\mathbf{p}\|_2^2$. The regularization term $\|W\mathbf{p}\|_1$ penalizes the number of large framelet coefficients, which could be viewed as penalizing the number of pixels with large discontinuities. Thus, the resulting minimizer $p$ tends to be a function with the area of the support being small. However, by only using the analysis based sparsity regularization, the support of the resulting kernel is biased to sparse isolated points, especially when there are large oscillations on the speed of camera motion. The second regularization term $\|\mathbf{p}\|_2^2$ in (13) comes to correct such a bias by also regularizing the kernel $\mathbf{p}$ with $\ell_2$ norm $\|\cdot\|_2^2$, as the resulting minimizer tends to favor the blur kernels of larger connected support. By balancing the sparsity prior $\|W\mathbf{p}\|_1$ and the support continuity prior $\|\mathbf{p}\|_2^2$ using parameter $\tau$, the proposed regularization term (13) will yield a sound motion-blur kernel.

In general, the minimization problem resulting from the regularization-based formulation for blind deconvolution is not a convex problem due to the non-convexity of the fidelity term $\Phi(g*p-f)$ ($\Phi(\cdot) = \|\cdot\|_2^2$ in our approach). Thus, the method may converge to a local minimum instead of the global one, depending on how it is initialized. As a result, many existing blind deconvolution algorithms require some accurate physical measurement of the blur kernel, e.g., the kernel size ( [10]). Algorithm 1 can not guarantee the convergence to the global minimum either. However, in the experiments, it has been consistently converging to the result of high visual quality without requiring knowing the kernel size. In the remaining of this section, we give a heuristic argument on why Algorithm 1 with proposed regularization terms (12)–(13) is very likely to converge to a sound solution, instead of some degenerate solution (6).

The two regularization terms (12) and (13) play important roles on avoiding the convergence to the degenerate solution (6). For the case of over-deblurring where $h$ is a high-pass filter, the de-blured image $\widetilde{g} = g * h$ is a wrongly sharpened version of natural image by high-pass filtering, which will significantly increase the high-frequency content of the image. Thus, $\widetilde{g}$ will have much more large framelet coefficients than $g$ does, or equivalently $\|W\widetilde{g}\|_1$ is much larger than $\|Wg\|_1$. On the other hand, the resulting blur kernel $\widetilde{p} = p * h^{-1}$ is a smoothed version of $p$ by low-pass filtering. Notice that the blur kernel $p$ itself is a smooth kernel with very few high-frequency components, compared to nature images. Thus, the number of large framelet coefficients of $\widetilde{p}$ is nearly the same as that of $p$, which implies that $\|W\widetilde{p}\|_1$ is still very close to $\|Wp\|_1$. With an appropriate parameter $\tau$, the overall cost (2) for the over-deblurred solution $(\widetilde{g}, \widetilde{p})$ will be larger than that of the true solution. For the case of less-deblurring where $h$ is a low-pass filter, the deblurred image $\widetilde{g} = g * h$ is a still a smoothed version of $g$ but with less blurring effect. In such a case, the resulted blur kernel $\widetilde{p}$ is usually a blur kernel with a smaller support than the truth. Although we can see a small decrease on the costs of both $\|W\widetilde{g}\|_1$, both $\|W\widetilde{p}\|_1$ and the penalty term $\|\widetilde{p}\|_2^2$ in (13) will see an increase of their values. By assigning an appropriate value to the parameter $\tau$, the overall cost for such $(\widetilde{g}, \widetilde{p})$ will still be larger than that of the true solution. In summary, the balance between the sparsity-based regularization $\|Wg\|_1$ on the image $g$, the sparsity-based regularization $\|Wp\|_1$ and the $\ell_2$ norm-based regularization $\|p\|_2^2$ on the kernel $p$ is likely to avoid those degenerated solutions and generate a sound solution of high visual quality.

### C. Numerical algorithms

This section is devoted to the detailed numerical algorithm of our blind motion deblurring algorithm outlined in Algorithm 1. Both steps in Algorithm 1 are solving the same type of large scale minimization problems. The difficulties lie in the non-separable $\ell_1$ norm terms $\|W\mathbf{g}\|_1$ and $\|W\mathbf{p}\|_1$. One efficient solver for minimizations involving such terms is the *split Bregman iteration* [1], [2], which will be used in our solver. The split Bregman iteration is based on the Bregman iteration. The Bregman iteration was first introduced for non-differentiable TV-energy in [43] and then was successfully applied to wavelet based denoising in [44]. The Bregman iteration was also used in TV-based blind deconvolution in [25], [26]. To further improve the performance of the Bregman iteration, a linearized

Bregman iteration was invented in [45]. More details and an improvement called "kicking" of the linearized Bregman iteration is described in [46], and a rigorous theory was given in [47]. The linearized Bregman iteration for frame-based image deblurring was proposed in [32]. Recently, a new type of iteration based on Bregman distance, called split Bregman iteration, was introduced in [2] which extended the utility of Bregman iteration and linearized Bregman iteration to more general $\ell_1$ norm minimization problems. The split Bregman iteration for frame-based image deblurring was first proposed in [1]. The basic idea of split Bregman iteration is to convert the unconstrained minimization problem (10) and (12) ((11) and (13) respectively) into a constrained one by introducing an auxiliary variable $\mathbf{d}_1 = W\mathbf{g}$ ($\mathbf{d}_2 = W\mathbf{p}$ respectively) and then invoke the Bregman iteration to solve the constrained minimization problem. Numerical simulations in [1], [2] show that it converges fast and only uses a small memory footprint which make it very attractive for large-scale problems.

Let $\mathbf{f}, \mathbf{g} \in \mathbb{R}^n$ denote the given blurred image $f$ and the original image $g$ after column concatenation. We assume that the size of the blur kernel is not larger than that of the images, and let $\mathbf{p} \in \mathbb{R}^n$ denote its vectorized version. Let $[\cdot]_*$ denote the matrix form of the convolution operator after concatenating operations as follows,

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

In Step 1, we need to solve the following $\ell_1$-norm based minimization:

$$\min_{\mathbf{g}} \frac{1}{2}\|[\mathbf{p}^{(k)}]_*\mathbf{g} - \mathbf{f}\|_2^2 + \lambda_1\|W\mathbf{g}\|_1. \tag{14}$$

By letting $\mathbf{d}_1 = W\mathbf{g}$, the minimization (14) is equivalent to

$$\min_{\mathbf{g},\mathbf{d}_1} \frac{1}{2}\|[\mathbf{p}^{(k)}]_*\mathbf{g} - \mathbf{f}\|_2^2 + \lambda_1\|\mathbf{d}_1\|_1 \quad \text{s.t.} \quad \mathbf{d}_1 = W\mathbf{g}. \tag{15}$$

Then, the problem (15) is further transferred into a non-constrained minimization

$$\min_{\mathbf{g},\mathbf{d}_1} \frac{1}{2}\|[\mathbf{p}^{(k)}]_*\mathbf{g} - \mathbf{f}\|_2^2 + \lambda_1\|\mathbf{d}_1\|_1 + \frac{\lambda_1\mu}{2}\|(W\mathbf{g} - \mathbf{d}_1) + \mathbf{b}_1\|_2^2, \tag{16}$$

where $\mu$ is a constant number. This is the so-called split Bregman iteration. The iterative numerical algorithm for solving (16) (14) is described as following:

$$\left\{\begin{array}{lcl} \mathbf{g}^{(\ell+1,k)} & := & \text{argmin}_{\mathbf{g}} \frac{1}{2}\|[\mathbf{p}^{(k)}]_*\mathbf{g} - \mathbf{f}\|_2^2 \\ & & + \frac{\lambda_1\mu}{2}\|W\mathbf{g} - \mathbf{d}_1^{(\ell,k)} + \mathbf{b}_1^{(\ell,k)}\|_2^2, \\ \mathbf{d}_1^{(\ell+1,k)} & := & T_{1/\mu}(W\mathbf{g}^{(\ell+1,k)} + \mathbf{b}_1^{(\ell,k)}), \\ \mathbf{b}_1^{(\ell+1,k)} & := & \mathbf{b}_1^{(\ell,k)} + (W\mathbf{g}^{(\ell+1,k)} - \mathbf{d}_1^{(\ell+1,k)}), \end{array}\right. \tag{17}$$

Interesting readers are referred to [1] for more detailed derivation of (17). The following theorem, which follows directly from [1, Theorem 3.2], says that iteration (17) is the right one to use.

*Theorem 1:* The iteration (17) satisfies the following property:

$$\begin{array}{l} \lim_{\ell\to+\infty} \frac{1}{2}\|[\mathbf{p}^{(k)}]_*\mathbf{g}^{(\ell,k)} - \mathbf{f}\|_2^2 + \lambda_1\|W\mathbf{g}^{(\ell,k)}\|_1 \\ = \frac{1}{2}\|[\mathbf{p}^{(k)}]_*\mathbf{g}^{(*,k)} - \mathbf{f}\|_2^2 + \lambda_1\|W\mathbf{g}^{(*,k)}\|_1, \end{array}$$

where $\mathbf{g}^{(*,k)}$ is a minimizer of (14). Furthermore, assume that (14) has a unique minimizer, then iteration (17) converges, i.e.,

$$\lim_{\ell\to+\infty} \|\mathbf{g}^{(\ell,k)} - \mathbf{g}^{(*,k)}\|_2 = 0.$$

*Proof:* To apply [1, Theorem 3.2], we need to prove that there exists a minimizer satisfying (14). The existence of such a minimizer follows immediately from the definition of the cost functional. ■

In Step 2 of Algorithm 2, we need to solve the minimization problem similar to (14):

$$\frac{1}{2}\|[\mathbf{g}^{(k+1)}]_*\mathbf{p} - \mathbf{f}\|_2^2 + \lambda_2(\|W\mathbf{p}\|_1 + \frac{\tau}{2}\|\mathbf{p}\|_2^2). \tag{18}$$

The split Bregman iteration can also be used to solve the above minimization efficiently with some modifications. The sequences of $\mathbf{v}^{(\ell,k)}, \mathbf{d}_2^{(\ell,k)}, \mathbf{b}_2^{(\ell,k)}$ is generated as following:

$$\left\{\begin{array}{lcl} \mathbf{p}^{(\ell+1,k)} & := & \text{argmin}_{\mathbf{p}} \frac{1}{2}\|[\mathbf{g}^{(k+1)}]_*\mathbf{p} - \mathbf{f}\|_2^2 + \frac{\lambda_2\tau}{2}\|\mathbf{p}\|_2^2 \\ & & + \frac{\lambda_2\mu}{2}\|W\mathbf{p} - \mathbf{d}_2^{(\ell,k)} + \mathbf{b}_2^{(\ell,k)}\|_2^2, \\ \mathbf{d}_2^{(\ell+1,k)} & := & T_{1/\mu}(W\mathbf{p}^{(\ell+1,k)} + \mathbf{b}_2^{(\ell,k)}), \\ \mathbf{b}_2^{(\ell+1,k)} & := & \mathbf{b}_2^{(\ell,k)} + (W\mathbf{p}^{(\ell+1,k)} - \mathbf{d}^{(\ell+1,k)}), \end{array}\right. \tag{19}$$

where $\mu > 0$ is some parameter of the iteration, $T_\theta$ is the soft-thresholding operator. The first step of each iteration in (19) is done by solving the following positive definite linear system:

$$\begin{array}{l} ([\mathbf{g}^{(k+1)}]_*^T[\mathbf{g}^{(k+1)}]_* + \lambda_2(\mu + \tau)I)\mathbf{v} \\ = [\mathbf{g}^{(k+1)}]_*^T\mathbf{f} + \lambda_2\mu W^T(\mathbf{d}^{(\ell,k)} - \mathbf{b}^{(\ell,k)}). \end{array}$$

We have the same convergence result for (19) as that of (17).

In Step 1 (respectively Step 2), to get an exact solution of (14) (respectively (18)), we need to choose $\mathbf{g}^{(k+1)} = \mathbf{g}^{(+\infty,k)}$ (respectively $\mathbf{p}^{(k+1)} = \mathbf{p}^{(+\infty,k)}$) as stated in Theorem 1. However, it is too conservative to use infinite steps of iterations. The reason is that the image $\mathbf{g}^{(k)}$ or the kernel $\mathbf{p}^{(k)}$ might not be accurate enough, and accuracy obtained by the infinite loop will be wasted. This is typical in split Bregman iterations; see [1], [2]. Therefore, for computational efficiency, we only need to perform one iteration in (17) and (19), i.e., we choose set $\mathbf{g}^{(k+1)} = \mathbf{g}^{(1,k)}$ and $\mathbf{p}^{(k+1)} = \mathbf{p}^{(1,k)}$.

Moreover, in practice, the iterates $\mathbf{g}^{(k)}$ and $\mathbf{p}^{(k)}$ may not always be a physically sound solution. Thus, we chose to impose the following physical conditions:

$$\mathbf{p} \geq 0, \quad \sum_j \mathbf{p}(j) = 1; \quad \text{and} \quad 0 \leq \mathbf{g} \leq 1. \tag{20}$$

which says that the blur kernel is non-negative and is normalized, and the range of the original image is between $[0, 1]$. In each step, we project $\mathbf{g}^{(k)}$ and $\mathbf{p}^{(k)}$ such that they satisfies the physical conditions in (20). Taking all above into account, we get the complete iteration that will be used in our algorithm for blind motion-deblurring; see Algorithm 2 for a detailed description.

## III. Experiments and conclusions

In our implementation, the maximum iteration number of Algorithm 2 is set to 100. For blurred color images of size 866 × 1280, each iteration in Algorithm 2 (implemented in

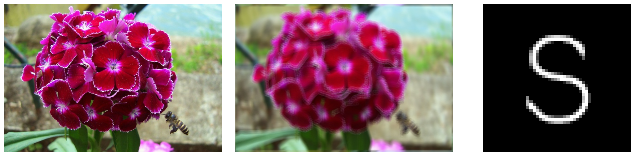
**Algorithm 2** Numerical algorithm for blind motion deblurring

(1) Set $k := 0$, $\mathbf{p}^{(0)} := \delta, \mathbf{d}_1 = \mathbf{b}_1 := 0$ and $\mathbf{d}_2 = \mathbf{b}_2 := 0$, where $\delta$ is the Delta function;

(2) DO

$$\left\{\begin{array}{lcl}
\mathbf{g}^{(k+\frac{1}{2})} & := & \mathrm{argmin}_{\mathbf{g}} \frac{1}{2}\|[\mathbf{p}^{(k)}]_{*}\mathbf{g} - \mathbf{f}\|_2^2 \\
 & & + \frac{\lambda_1\mu_1}{2}\|W\mathbf{g} - \mathbf{d}_1^{(k)} + \mathbf{b}_1^{(k)}\|_2^2, \\
\mathbf{g}^{(k+1)}(j) & := & \begin{cases} 1, & \text{if } \mathbf{g}^{(k+\frac{1}{2})}(j) > 1; \\ 0, & \text{if } \mathbf{g}^{(k+\frac{1}{2})}(j) < 0; \\ \mathbf{g}^{(k+\frac{1}{2})}(j), & \text{otherwise}, \end{cases} \\
 & & j = 1, 2, \ldots, N, \\
\mathbf{d}_1^{(k+1)} & := & T_{1/\mu_1}(W\mathbf{g}^{(k+1)} + \mathbf{b}_1^{(k)}), \\
\mathbf{b}_1^{(k+1)} & := & \mathbf{b}_1^{(k)} + (W\mathbf{g}^{(k+1)} - \mathbf{d}_1^{(k+1)}), \\
\mathbf{p}^{(k+\frac{1}{2})} & := & \mathrm{argmin}_{\mathbf{p}} \frac{1}{2}\|[\mathbf{g}^{(k+1)}]_{*}\mathbf{p} - \mathbf{f}\|_2^2 + \frac{\lambda_2\tau}{2}\|\mathbf{p}\|_2^2 \\
 & & + \frac{\lambda_2\mu_2}{2}\|W\mathbf{p} - \mathbf{d}_2^{(k)} + \mathbf{b}_2^{(k)}\|_2^2, \\
\widetilde{\mathbf{p}}^{(k+1)}(j) & := & \max(\mathbf{p}^{(k+\frac{1}{2})}(j), 0), \quad j = 1, 2, \ldots, N, \\
\mathbf{p}^{(k+1)} & := & \frac{\widetilde{\mathbf{p}}^{(k+1)}}{\|\widetilde{\mathbf{p}}^{(k+1)}\|_1}, \\
\mathbf{d}_2^{(k+1)} & := & T_{1/\mu_2}(W\mathbf{p}^{(k+1)} + \mathbf{b}_2^{(k)}), \\
\mathbf{b}_2^{(k+1)} & := & \mathbf{b}_2^{(k)} + (W\mathbf{p}^{(k+1)} - \mathbf{d}_2^{(k+1)}), \\
k & := & k + 1,
\end{array}\right. \tag{21}$$

UNTIL ($k \geq K$ or $\|p^{(k)} - p^{(k-1)}\|_2^2 \leq \epsilon$)



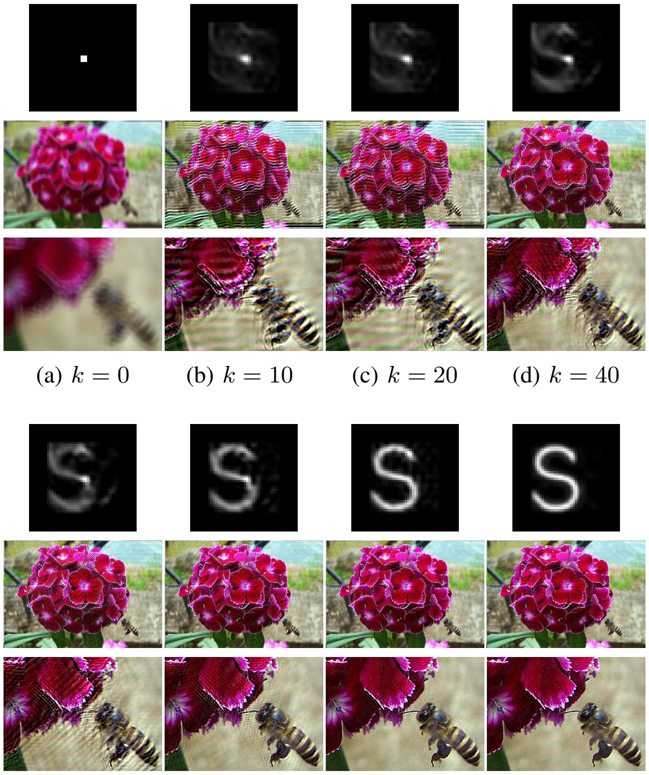
(a) One clear image (b) the blurred image (c) the blur kernel

Fig. 2. (a): an clear image; (b): the blurred image (c): the corresponding blur kernel.

Matlab 7.8) takes roughly 10 seconds on a windows PC with 4GB memory and one single Intel Core 2 CPU of 2 GHz Frequency. In Section III. B, when running Algorithm 2 on real image data, maximum iteration number $K = 200$ and the stopping threshold $\epsilon = 10^{-5}$. The parameters used inside Algorithm 2 are set as the following:

$$\left\{\begin{array}{l}
\lambda_2 = 10^{-1}\lambda_1(\sum_{i,j} f(i,j)); \\
\mu_1 = 2 \cdot 10^3; \\
\mu_2 = 10^{-1}\mu_1; \\
\tau = 10^{-1}\mu_2.
\end{array}\right.$$

In these parameters, $\mu_1$ and $\mu_2$ are only related to the convergence rate of the Bregman iteration. Setting different values of $\mu_1$ and $\mu_2$ does not change the outcome of Algorithm 2. The parameter $\lambda_1$ depends on the complexity of the image constant. The image with simple structure will prefer a smaller $\lambda_1$. It is noted that Algorithm 2 is rather insensitive to the value of $\lambda_1$. In our experiments, we tried Algorithm 2 on image data using three candidate values of $\lambda_1$: $\{10^{-4}, 10^{-3}, 10^{-2}\}$ and set $\lambda_1 = 10^{-3}$ uniformly for all experiments.



(a) $k = 0$ (b) $k = 10$ (c) $k = 20$ (d) $k = 40$

(e) $k = 80$ (f) $k = 160$ (g) $k = 320$ (h) $k = 640$

Fig. 3. (a)–(h) are the results on deblurring the blurred image shown in Fig. 2 (b) in $k$-th iteration of Algorithm 2, for $k = 0, 10, 20, 40, 80, 160, 320, 640$ respectively. Estimated blur kernels are shown in the first row; intermediate deblurred images in the second row; and one region after zooming in in the third row.

## A. *Simulated images*

In the first part of the experiments, we synthesized one sample motion-blurred image of size 866 × 1280 to illustrate the convergence behavior of Algorithm 2. The motion blur-kernel is simulated by letting the camera move in a constant speed along an ”$S$” curve in the image plane. The sample original image is shown in Fig. 2 (a) and the synthesized blurred image are shown in Fig. 2 (b) with the corresponding blur kernels shown in Fig. 2 (c). The intermediate results obtained during the iterations of Algorithm 2 are shown in Fig. 3. The estimated motion-blur kernels are shown in the first row, the intermediate recovered images are shown in the second row, and the same region of the recovered images are shown in the third row after zooming in for better visual inspection. It is clear that Algorithm 2 is empirically convergent to the ground truth of the original image and blur kernel. And the convergence rate is fairly fast as it only took 160 iterations to obtain a satisfactory result. With more iterations, it is seen from Fig. 3 (h) that Algorithm 2 estimated the motion-blur kernel very accurately and yielded a high-quality deblurring image with little artifacts.

## B. *Real images*

In the second part of the experiments, Algorithm 2 is applied on real image data from various sources ( [10], [33], [48]). We compared our results against the other five single-image based methods: You & Kaveh's method ( [20]), Fergus et al.'s method ( [10]), Shan et al.'s method ( [27]), Tzikas et al.'s method ( [31]) and Cai et al.'s method ( [33]). As we discussed in Section I, the approach proposed in [20] is based on the Tikhonov regularization on both images and kernels. Fergus et al.'s method use the statistical properties of image derivatives to infer motion-blur kernel. In order to get a good result, it need fairly accurate information regarding the size the motion-blur kernel, especially when the size of motion-blur kernel is large ($\geq$ 30 pixels). Shan et al.'s method is based on a sophisticated TV-norm based minimization model on both image intensity and image gradients, the regularization term on the motion-blur kernel is the $\ell_1$ norm of the kernel intensity. Similar to [10], it also requires the input of the kernel size. Cai et al.'s method ( [33]) is based on synthesis-based sparsity constraint of motion kernels in curvelet domain and synthesis-based sparsity constraint of images in wavelet domain. For [10], [27] and [33], the results are from the authors' implementations. For [20] and [31], the results are from our own implementations. The parameters of the above methods above are tuned up to find visually pleasant results on tested images. Tzikas et al.'s method estimates the blur kernel using a weighted combination of Gaussian-type basis functions whose weights satisfying a heavy tailed Students-t distribution. It is noted that one advantage of Tzikas et al.'s method is that many parameters are automatically estimated and we used the default values of remained few parameters proposed in their paper.

Fig. 4 – 7 showed the recovered results from all six methods on four real blurred images, which differ from each other on the type of image content, the blurring degree and the type of camera motion. The motion-blur kernels estimated by Algorithm 2 are shown in Fig. 4(h) – 7 (h) for all tested images. It is seen that there are camera motion along straight line segmentations but with varies speed, camera motion along curve and camera motion along trajectory with sharp corners. It is seen that the visual quality of recovered images from Algorithm 2 degraded a little bit compared against that from the previous simulated experiment. The main reason is that the blurring happening in real life is hardly a perfect spatial-invariant motion blurring, either there exist other image blurring effects, e.g., out of focus blurring; or the motion-blurring is not completely uniform over the whole image.

Overall, compared to other evaluated methods, Algorithm 2 performed consistently over these images and the results are of good quality with few noticeable image artifacts. The results from five other existing methods varied in terms of visual quality and are in general visually inferior than that from Algorithm 2. You & Kaveh's method ( [20]) only worked well on the image shown in Fig. 6 and did poorly on three other images. Such a result is not surprising as the Tikhonov regularization used in You & Kaveh's method emphasizes too much on the smoothness of blur kernels such that it tends to yield Gaussian-type blur kernels. As a result, the kernel is often over-sized which will lead to over-deblurred results. Fergus et al.'s method ( [10]) performance relies on how well the distribution of image derivatives fit the assumption, which certainly has its limitation as the image content could vary significantly in the experiments. It is seen from Fig. 6 and Fig. 7 that it tends to under-estimate the blur degree such that the results look still blurry. The kernel estimation of Shan et al.'s method ( [27]) is based on the regularization $\ell_1$ norm of blur kernel, which seems not adequate enough to yield accurate estimation of motion-blur kernels tested in the experiments. Thus, the results on tested images from Shan et al.'s method is not satisfactory. Tzikas et al.'s method ( [31]) has its advantages over other methods on the autonomous estimation of regularization parameters. However, the kernel model considered in Tzikas is the linear combination of Gaussian-type blur kernel. As a result, it did not perform well on images blurred by "thin" kernels, such as Fig. 5. Cai et al.'s method ( [33]) is quite stable compared to other methods. However, the results from Cai et al.'s method tend to show noticeable artifacts along edges. The main reason is the usage of synthesis-based approach to enforce the sparsity constraints of image/kernel. Such a drawback is well addressed in Algorithm 2. The results from Algorithm 2 are consistent on tested images and in general are more visually pleasant with fewer artifacts than that of the other methods.

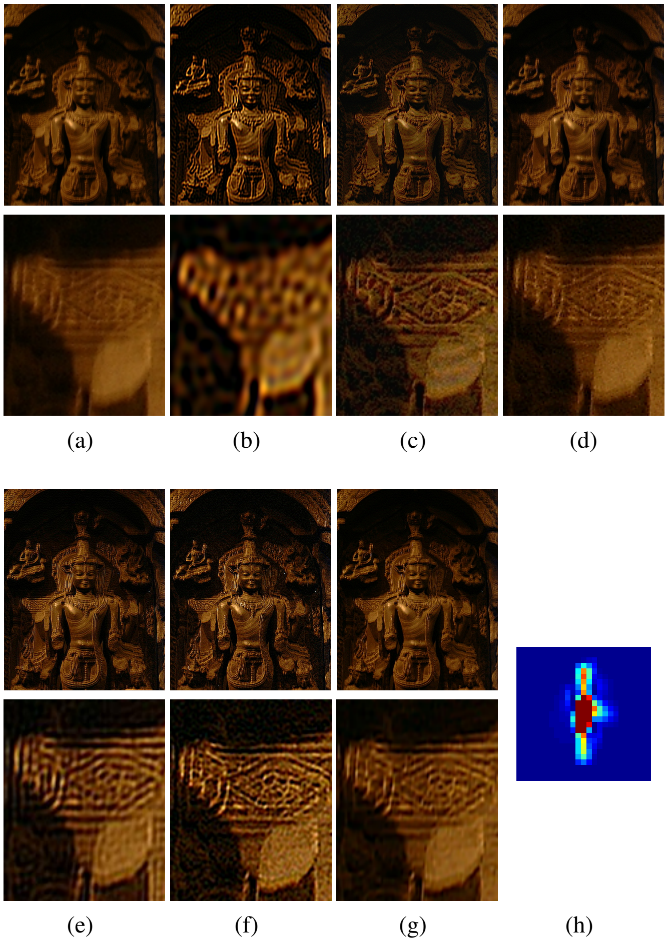
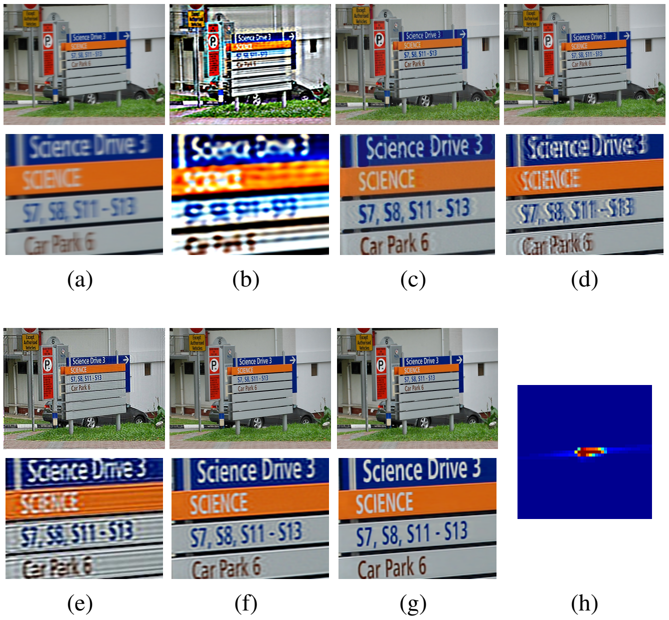


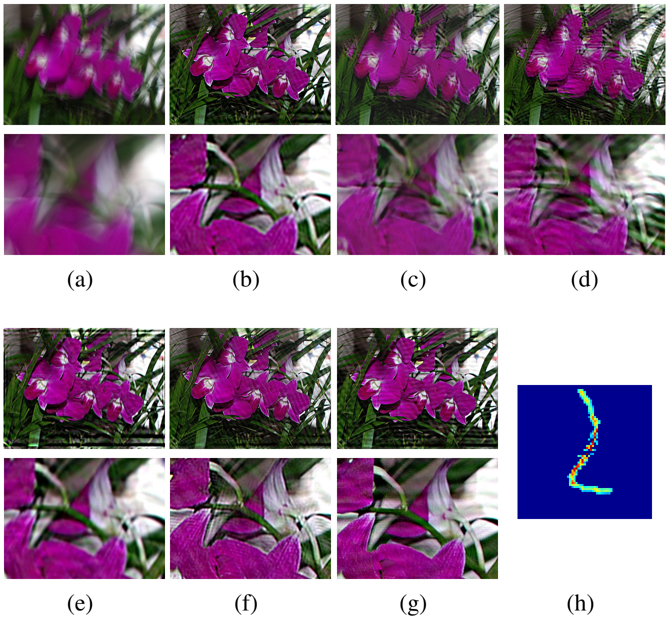
Fig. 4. The blurred image of size $728 * 905$ and its deblurred results with zoomed regions. (a) Blurred image; (b) You & Koveh [20]; (c) Fergus et al. [10]; (d) Shan et al. [27]; (e) Tzikas et al. [31]; (f) Cai et al. [33]; (g) Algorithm 2. (h) Color-visualization of the motion-blur kernel estimated by Alg. 2 with image size $25 * 25$.



(a) (b) (c) (d)

(e) (f) (g) (h)

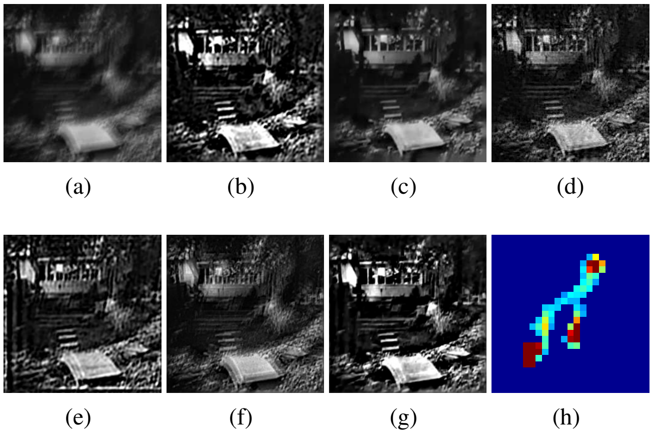
Fig. 5. The blurred image of size 640 ∗ 480 and its deblurred results with zoomed regions. (a) Blurred image; (b) You & Koveh [20]; (c) Fergus et al. [10]; (d) Shan et al. [27]; (e) Tzikas et al. [31]; (f) Cai et al. [33]; (g) Algorithm 2. (h) Color-visualization of the motion-blur kernel estimated by Alg. 2 with image size 45 ∗ 45.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. The blurred image of size 1024 ∗ 768 and its deblurred results with zoomed regions. (a) Blurred image; (b) You & Koveh [20]; (c) Fergus et al. [10]; (d) Shan et al. [27]; (e) Tzikas et al. [31]; (f) Cai et al. [33]; (g) Algorithm 2. (h) Color-visualization of the motion-blur kernel estimated by Alg. 2 with image size 65 ∗ 65.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. The blurred image of size 255 ∗ 255 and its deblurred results with zoomed regions. (a) Blurred image; (b) You & Koveh [20]; (c) Fergus et al. [10]; (d) Shan et al. [27]; (e) Tzikas et al. [31]; (f) Cai et al. [33]; (g) Algorithm 2. (h) Color-visualization of the motion-blur kernel estimated by Alg. 2 with image size 25 ∗ 25.

### C. *Conclusions and future work*

In this paper, a new algorithm is presented to remove camera shake from a single image. Based on analysis-based sparsity prior of images in framelet domain and a mixed regularization on motion-blur kernels, which includes both analysis-based sparsity prior of kernels in framelet domain and smoothness prior on kernels, our new formulation on motion deblurring leads to a powerful algorithm which can recover a clear image from a given motion-blurred image. The analysis-based sparsity prior used in our approach empirically yields more visually pleasant results than the synthesis-based sparse prior used in other methods do ( [33]). Also, our formulation does not suffer from converging to some well-known degenerate cases as many other approaches might do. As a result, our method does not require any prior information on the kernel while many existing techniques are dependently on some accurate information of motion blurring which usually needs user interactions. The resulting minimization problem from our formulation could be solved efficiently by the split Bregman method. The experiments on both synthesized and real images show that our proposed algorithm is very efficient and very effective on removing complicated blurring from nature images of complex structures.

Blind motion deblurring is a challenging blind deconvolution problem and there are still many open questions remained. Our proposed algorithm performs quite well in the case of motion-blurring being the uniform blurring over the image, but can not deal with non-uniform motion-blurring, including image blurring caused by camera rotation and the partial image blurring caused by fast moving objects in the scene. In future, we would like to extend our proposed algorithm to remove non-uniform motion blurring from images. Also, although the parameter setting in our algorithm is rather simple without requiring rigorous tuning up, it is still more preferred for some applications to have a completely automatic parameter setting process. Recently, there have been some blind deconvolution techniques that use either variational Bayesian approach (*e.g.* [49]) or cross-validation techniques to automatically determine the optimal parameter values (*e.g.* [50]). In future, we also would like to investigate how to incorporate these techniques into our method to automatically infer optimal parameter setting of the de-blurring algorithm.

## References

[1] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman method and frame based image restoration," *Multiscale model. Simul.*, vol. 8, no. 2, pp. 337–369, 2009.

[2] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

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

[4] M. K. Ng, R. H. Chan, and W. Tang, "A fast algorithm for deblurring models with neumann boundary condition," *SIAM J. Sci. Comput.*, vol. 21, no. 3, pp. 851–866, 2000.

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

[6] A. Chai and Z. Shen, "Deconvlolution: A wavelet frame approach," *Numer. Math.*, vol. 106, pp. 529–587, 2007.

[7] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi, "Image recovery via nonlocal operators," *Journal of Scientific Computing*, vol. 42, no. 2, pp. 185–197, 2010.

[8] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring using multiple images," *Journal of Computational Physics*, vol. 228, no. 14, pp. 5057–5071, 2009.

[9] G. Pavlovic and A. M. Tekalp, "Maximum likelihood parametric blur identification based on a continuous spatial domain model," *IEEE Trans. Image Processing*, vol. 1, no. 4, Oct. 1992.

[10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *SIGGRAPH*, vol. 25, 2006, pp. 783–794.

[11] A. Levin, "Blind motion deblurring using image statistics." in *NIPS*, Dec. 2006, pp. 841–848.

[12] J. Jia, "Single image motion deblurring using transparency," in *CVPR*, 2007, pp. 1–8.

[13] N. Joshi, R. Szeliski, and D. Kriegman, "PSF estimation using sharp edge prediction," in *CVPR*, 2008.

[14] B. Bascle, A. Blake, and A. Zisserman, "Motion deblurring and super-resolution from an image sequence," in *ECCV*, 1996, pp. 573–582.

[15] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. PAMI*, vol. 26, no. 6, pp. 689–698, 2004.

[16] F. Sroubek and J. Flusser, "Multichannel blind deconvolution of spatially misaligned images," *IEEE Tran. Image Processing*, vol. 14, no. 7, pp. 874–883, 2005.

[17] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: Motion deblurring via fluttered shutter," in *SIGGRAPH*, vol. 25, 2006, pp. 795–804.

[18] J. Chen, L. Yuan, C. K. Tang, and L. Quan, "Robust dual motion deblurring," in *CVPR*, 2008.

[19] Y. Lu, J. Sun, L. Quan, and H. Shum, "Blurred/non-blurred image alignment using an image sequence," in *SIGGRAPH*, 2007.

[20] Y. You and M. Kaveh, "A regularization approach to joint blur identification and image restoration," *IEEE Tran. Image Processing*, vol. 5, 1996.

[21] R. Molina, J. Mateos, and A. K. Katsaggelos, "Blind deconvolution using a variational approach to parameter, image, and blur estimation," *IEEE transacion on image processing*, vol. 15, no. 12, pp. 3715–3727, 2006.

[22] T. F. Chan and C. K. Wong, "Total variation blind deconvolution," *IEEE Tran. Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.

[23] 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.

[24] P. D. Romero and V. F. Candela, "Blind deconvolution models regularized by fractional powers of the laplacian," *Journal of Mathematical Imaging and Vision*, vol. 32, 2008.

[25] L. He, A. Marquina, and S. Osher, "Blind deconvolution using TV regularization and Bregman iteration," *Int. J. Imaging Syst. Technol.*, vol. 15, pp. 74–83, 2005.

[26] A. Marquina, "Inverse scale space methods for blind deconvolution," *UCLA CAM Reports*, vol. 06-36, 2006.

[27] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *SIGGRAPH*, 2008.

[28] D. C. Dobson and F. Santosa, "Recovery of blocky images from noisy and blurred data," *SIAM J. Appl. Math.*, vol. 56, pp. 1181–1198, 1996.

[29] M. Nikolova, "Local strong homogeneity of a regularized estimator," *SIAM J. Appl. Math.*, vol. 61, pp. 633–658, 2000.

[30] M. M. Bronstein, A. M. Bronstein, M. Zibulevsky, and Y. Y. Zeevi, "Blind deconvolution of images using optimal sparse representations," *IEEE Trans. Image Processing*, vol. 14, no. 6, pp. 726–736, 2005.

[31] D. G. Tzikas, A. C. Likas, and N. P. Galasanos, "Variational bayesian sparse kernel-based blind image deconvolution with student's-t priors," *IEEE Trans. Image Processing*, vol. 18, no. 4, pp. 753–764, 2009.

[32] J.-F. Cai, S. Osher, and Z. Shen, "Linearized bregman iterations for frame-based image deblurring," *SIAM Journal on Imaging Sciences*, pp. 226–252, 2009.

[33] J.-F. Cai, H.Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *CVPR*, 2009.

[34] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[35] A. Ron and Z. Shen, "Affine system in $L_2(R^d)$: the analysis of the analysis operator," *J. of Func. Anal.*, vol. 148, 1997.

[36] I. Daubechies, B. Han, A. Ron, and Z. Shen, "Framelets: MRA-based constructions of wavelet frames," *Appl. Comput. Harmon. Anal.*, vol. 14, pp. 1–46, 2003.

[37] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," in *EUSIPCO*, 2006.

[38] L. Borup, R. Gribonval, and M. Nielsen, "Bi-framelet systems with few vanishing moments characterize Besov spaces," *Appl. Comput. Harmon. Anal.*, vol. 14, no. 1, 2004.

[39] E. Candes, L. Demanet, D. L. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale Model. Simul.*, vol. 5, pp. 861–899, 2005.

[40] J.-F. Cai, R. Chan, and Z. Shen, "A framelet-based image inpainting algorithm," *Appl. Comput. Harmon. Anal.*, vol. 24, pp. 131–149, 2008.

[41] Z. Shen, "Wavelet frames and image restorations," in *Proceedings of the International Congress of Mathematicians*, Hyderabad, India, 2010.

[42] C. de Boor, *A Practical Guide to Splines*. Spring, 1978.

[43] S. Osher, B. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Model. Simul*, vol. 4, pp. 460–489, 2005.

[44] J. Xu and S. Osher, "Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoisin," *IEEE Trans. Image Processing.*, vol. 16, pp. 534–544, 2007.

[45] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for $\ell_1$-minimization with applications to compressed sensing," *SIAM J. Imaging Sci.*, vol. 1, pp. 143–168, 2008.

[46] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized bregman iteration for compressive sensing and sparse denoising," *Communications in Mathematical Sciences.*, vol. 8, no. 2, pp. 93–111, 2010.

[47] J.-F. Cai, S. Osher, and Z. Shen, "Linerarized bregman iterations for compressed sensing," *Mathematics of Computation*, vol. 78, no. 267, pp. 1515–1536, 2009.

[48] W. Freeman, F. Durang, Y. Fredo, and A. Levin, "Understanding and evaluating blind deconvolution algorithms," in *CVPR*, 2009.

[49] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational bayesian blind deconvolution using a total variation prior," *IEEE Tran. Image Processing*, vol. 18, p. 1, 2009.

[50] L. H. and M. K. Ng, "Blind deconvolution using generalized cross-validation approach to regularization parameter estimati," *IEEE Trans. Image Processing*, vol. 20, no. 3, pp. 670–680, 2011.

**Hui Ji** received the BSc degree from Nanjing University in China, the MSc degree in mathematics from National University of Singapore in Singapore and the PhD degree in computer science from University of Maryland, College Park, USA. Since 2006, he has been an assistant professor in Department of Mathematics at National University of Singapore in Singapore. His research interests are in human and computer vision, image and video processing, computational harmonic analysis and optimization.

**Zuowei Shen** Biography text here.