Blind motion deblurring from a single image using sparse approximation

Jian-Feng Cai[†], Hui Ji[‡], Chaoqiang Liu[†] and Zuowei Shen[‡]
National University of Singapore, Singapore 117542
Center for Wavelets, Approx. and Info. Proc.[†] and Department of Mathematics[‡]
{tslcaij, matjh, tslliucq, matzuows@nus.edu.sg}

Abstract

Restoring a clear image from a single motion-blurred image due to camera shake has long been a challenging problem in digital imaging. Existing blind deblurring techniques either only remove simple motion blurring, or need user interactions to work on more complex cases. In this paper, we present an approach to remove motion blurring from a single image by formulating the blind blurring as a new joint optimization problem, which simultaneously maximizes the sparsity of the blur kernel and the sparsity of the clear image under certain suitable redundant tight frame systems (curvelet system for kernels and framelet system for images). Without requiring any prior information of the blur kernel as the input, our proposed approach is able to recover high-quality images from given blurred images. Furthermore, the new sparsity constraints under tight frame systems enable the application of a fast algorithm called linearized Bregman iteration to efficiently solve the proposed minimization problem. The experiments on both simulated images and real images showed that our algorithm can effectively removing complex motion blurring from nature images.

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 lens or using long shuttle speed.g In past, many researchers have been working on recovering clear images from motion-blurred images. The motion blur caused by camera shake usually is modeled by a spatial-invariant blurring 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 *point spread function*) and n is the noise. If the blur kernel is given as a prior, recovering clear image is called a *non-blind deconvolution* problem; otherwise called a *blind deconvolution* problem. It is known that the non-blind deconvolution problem is an ill-conditioned problem for its sensitivity to noise. Blind deconvolution is even more ill-posed. Because both the blur kernel and the clear image are unknown, the problem becomes under-constrained as there are more unknowns than available measurements. Motion deblurring is a typical blind deconvolution problem as the motion between the camera and the scene can be arbitrary,

1.1. Previous work

Early works on blind deblurring usually use a single image and assume a prior parametric form of the blur kernel p such that the blur kernel can be obtained by only estimating a few parameters (*e.g.*, Pavlovic and Tekalp [22]). Linear motion blur kernel model used in these works often is overly simplified for true motion blurring in practice. To solve more complex motion blurring, multi-image based approaches have been proposed to obtain more information of the blur kernel by either actively or passively capturing multiple images on the scene (*e.g.*, Bascle *et al.* [2], Ben-Ezra and Nayar [3], Chen *et al.* [10], Lu *et al.* [19], Raskar [23], Tai *et al.* [26]).

Recently, there have been steady progresses on removing complex motion blurring from a single image. There are two typical approaches. One is to use some probabilistic priors on images' edge distribution to derive the blur kernel (*e.g.*, Fergus *et al.* [15], Levin [18], Joshi [17]) or manually selecting blurred edges to obtain the local blur kernel (Jia [16]). The main weakness of this type of methods is that the assumed probabilistic priors do not always hold true for general images. The other popular approach is to formulate the blind deconvolution as a joint minimization problem with some regularizations on both the blur kernel p and the clear image g:

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

where $\Phi(p * g - f)$ is the fidelity term, $\Theta_1(p)$ and $\Theta_2(g)$ are the regularization terms on the kernel and the clear image respectively. In this paper, we focus on the regularization-based approach.

Among existing regularization-based methods, TV (Total Variation) norm and its variations have been the dominant choice of the regularization term to solve various blind deblurring problems (*e.g.*, Bar *et al.* [1], Chan and Wong [9], Cho *et al.* [11]). In these approaches, the fidelity term in (2) is usual ℓ_2 norm on image intensity similarity; and the regularization terms Θ_1 and Θ_2 in (2) are both TV-norms of the image g and the kernel p. Shan *et al.* [25] presented a more sophisticated minimization model where the fidelity term is a weighted ℓ_2 norm on the similarity of both image intensity and image gradients. The regularization term on the latent image is a combination of a weighted TV norm of image and the global probabilistic constraint on the edge distribution as ([15]). The regularization term on the motion-blur kernel is the ℓ_1 norm of the kernel intensity. These minimization methods showed good performance on removing many types of blurring. In particular, Shan *et al.*'s method demonstrated impressive performance on removing modest motion blurring from images without rich textures.

However, solving the resulting optimization problem (2) usually requires quite sophisticated iterative numerical algorithms, which often fail to converge to the true global minimum if the initial input of the kernel is not well set. One well-known degenerate case is that the kernel converges to a delta-type function and the recovered image remain blurred. Thus, these methods need some prior information on the blur kernel as the input, such as the size of the kernel (*e.g.*, Fergus *et al.* [15], Shan *et al.* [25]). Also, the classic optimization techniques (*e.g.* interior point method) are highly inefficient for solving (2) as they usually requires the computation of the gradients in each iteration, which could be very expensive on both computation amount and memory consumption as the number of the unknowns could be up to millions (the number of image pixels).

1.2. Our approach

In this paper, we propose a new optimization approach to remove motion blurring from a single image. The contribution of the proposed approach is twofold. First, we propose new sparsity-based regularization terms for both images and motion kernels using redundant tight frame theory. Secondly, the new sparsity regularization terms enable the application of a new numerical algorithm, namely *linearized Bregman iteration*, to efficiently solve the resulting ℓ_1 norm related minimization problem.

Most of nature images have sparse approximation under some redundant tight frame systems, *e.g.* translation-invariant wavelet, Gabor transform, Local cosine transform ([20]), framelets ([24]) and curvelets ([8]). The sparsity of nature images under these tight frames has been successfully used to solve many image restoration tasks including image denoising, non-blind image deblurring, image inpainting, etc (*e.g.* [12, 6, 4]). Therefore, we believe that the high sparsity of images under certain suitable tight frame system is also a good regularization on the latent image in our blind deblurring problem. In this paper, we chose *framelet* system ([24, 13]) as the redundant tight frame system used in our approach for representing images. The motion-blur kernel is different from typical images. It can be viewed as a piece-wise smooth function in 2D image domain, but with some important geometrical property: the support of the kernel (the camera trajectory during exposure) is approximately a thin smooth curve. Thus, the best tight frame system for representing motion-blur kernel is *Curvelet* system, which is known for its optimal sparse representation for this type of functions ([8]).

The sparsity-based regularization on images or kernels is not a completely new idea. Actually, the widely used TV-norm based regularization can also be viewed as a sparsity regularization on image gradients. However, tight frames provide a wide range of transforms to sparsely represent all types of images. And the redundancy of tight frames will increase the robustness to the noise. More importantly, as Donoho pointed out in [14] that the minimal ℓ_1 norm solution is the sparsest solution for most large under-determined systems of linear equations, using tight frames to represent images/kernels is very attractive because the tight frame coefficients of images/kernels are indeed heavily redundant.

Furthermore, TV-norm or its variations are not very accurate regularization terms to regularize motion-blur kernels, as they do not impose the support of the kernel being an approximately smooth curve. However, by representing the blur kernel in the curvelet system, the geometrical property of the support of motion kernels are appropriately imposed, because a sparse solution in curvelet domain tends to be smooth curves instead of isolated points.

Another benefit using the sparse constraint under tight frame system is the availability of more efficient numerical methods to use the resulting ℓ_1 norm minimization. As we will show later, the perfect reconstruction property of tight framelet system allows the application of the so-called linearized Bregmen iteration (Osher *et al.* [21]), which has been shown in recent literatures (Cai *et al.* [6, 5]) that it is more efficient than classic numerical algorithms (*e.g.* interior point method) do when solving this particular type of ℓ_1 norm minimization problems.

The rest of the paper is organized as follows. In Section 2, we formulate the minimization model and the underlying motivation. Section 3 presented the numerical algorithm and its convergence analysis. Section 4 is devoted to the experimental evaluation and the discussion on future work.

2. Formulation of our minimization model

As we see from (1), blind deblurring is an under-constrained problem with many possible solutions. Extra constraints on both the image and the kernel are needed to overcome the ambiguity and the noise sensitivity. In this paper, we present a new formulation to solve (1) with sparsity constraints on the image and the blur kernel under suitable tight frame systems.

We propose to use framelet system (Ron and Shen *et al.* [24]) to find the sparse approximation to the image under framelet domain The blur kernel is a very special function with its support being an approximate smooth 2D curve. We use the curvelet system (Candes and Donoho [8]) to find the sparse approximation to the blur kernel under curvelet domain. Before we present our formulation on the blind deblurring, we first give a brief introduction to framelet system and curvelet system, which are used in our method. See the references (*e.g.* [24, 8])for more details.

2.1. Tight framelet system and curvelet system

A countable set of functions $X \in L_2(\mathbb{R})$ is called a *tight frame* of $L_2(\mathbb{R})$ if

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

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

$$\Xi := \{\psi_1, \cdots, \psi_r\} \in L_2(\mathbb{R}),$$

a *tight wavelet frame* X is a tight frame $\in L^2(\mathbb{R})$ defined by the dilations and the shifts of generators from Ξ:

$$X := \{\psi_{\ell,j,k} : 1 \leq \ell \leq r;\ j, k \in \mathbb{Z}\} \tag{3}$$

with $\psi_{\ell,j,k} := 2^{j/2} \psi_\ell(2^j \cdot - k)$, $\psi_\ell \in \Xi$. The construction of a set of framelets starts with a refinable function $\phi \in L^2(\mathbb{R})$ such that

$$\phi(x) = \sum_k h_0(k) \phi(2x - k)$$

for some filter h_0. Then a set of framelets is defined as

$$\psi_\ell(x) = \sum_k h_\ell(k) \phi(2x - k)$$

which satisfies so-called *unitary extension principle* ([24]):

$$\tau_{h_0}(\omega)\overline{\tau_{h_0}(\omega + \gamma\pi)} + \sum_{\ell=1}^{r} \tau_{h_\ell}(\omega)\overline{\tau_{h_\ell}(\omega + \gamma\pi)} = \delta(\gamma)$$

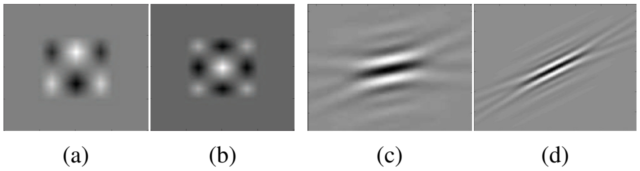


Figure 1. (a) and (b) are two framelets in the same scale. (c) and (d) are one curvelet in different scale.

for $\gamma = 0, 1$, where $\tau_h(\omega)$ is the trigonometric polynomial of the filter h:

$$\tau_h(\omega) = \sum_k h(k) e^{ik\omega}.$$

h_0 is a lowpass filter and $h_\ell, \ell = 1, \cdots, r$ are all high-pass filters. In this paper, we use the piece-wise linear framelet ([24, 13]):

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

2D framelets can be obtained by the tensor product of 1D framelets: $\Phi(x, y) = \phi(x) \otimes \phi(y)$,

$$\begin{aligned} \{\Psi\} \quad = \quad & \{\psi_{\ell_1}(x) \otimes \phi(y),\ \phi(x) \otimes \psi_{\ell_2}(y), \\ & \psi_{\ell_1}(x) \otimes \psi_{\ell_2}(y),\ 1 \leq \ell_1, \ell_2 \leq r\} \end{aligned}$$

The associated filters are also the Kronecker product of 1d filters: $H_0 = h_0 \otimes h_0$ and

$$\{H_\ell\} = \{h_{\ell_1} \otimes h_0, h_0 \otimes h_{\ell_2}, h_{\ell_1} \otimes h_{\ell_2},\ 1 \leq \ell_1, \ell_2 \leq r\}.$$

Then given a function $f \in L_2(\mathbb{R}^2)$, we have

$$f(x, y) = \sum_{\ell, j, k_1, k_2} u_{\ell, j, k1, k_2} \Psi_{\ell, j, k_1, k_2}, \tag{4}$$

where $\{u_{\ell,j,k_1,k_2} = \langle f, \Psi_{\ell,j,k_1,k_2} \rangle\}$ with

$$\Psi_{\ell,j,k_1,k_2} = 2^{j/2} \Psi_\ell(2^j x - k_1, 2^j y - k_2),\ j, k_1, k_2 \in \mathbb{Z}.$$

u_{ℓ,j,k_1,k_2} are called *framelet coefficients* of f.

The curvelet system ([8, 7]) is also a tight frame in $L_2(\mathbb{R}^2)$ where the generating functions are multiple rotated versions of a given function ψ:

$$X := \{\Psi_{\ell,j,k_1,k_2} = \Psi(2^j R_{\theta_\ell}((x, y)^T - x_{\ell,j,k_1,k_2}))\} \tag{5}$$

with

$$x_{\ell,j,k_1,k_2} = R_{\theta_\ell}^{-1}(2^{-j}k_1, 2^{-j/2}k_2)^T, \quad j, k_1, k_2 \in \mathbb{Z},$$

where $\theta_\ell = 2\pi 2^{\lfloor j \rfloor} \ell$ is the equi-spaced sequence of rotation angles such that $0 \leq \theta_\ell < 2\pi$, R_θ is the 2D rotation by θ radians and R_θ^{-1} its inverse. It is noted that curvelet system has different scaling from framelet system: $\mathrm{length} \approx 2^{-j}$ and $\mathrm{width} \approx 2^{-2j}$. Thus the scaling in curvelet system is a parabolic scaling such that the length and width of the curvelets obey the anisotropic scaling relation:

$$\mathrm{width}^2 \approx \mathrm{length}.$$

The combination of the parabolic scaling and the multiple rotations of the wavelet give a very sparse presentation for singularities along curves or hyper-surfaces in the image. Given a function $f \in L^2(\mathbb{R})$, we also have

$$f(x,y) = \sum_{\ell,j,k_1,k_2} v_{\ell,j,k_1,k_2} \Psi_{\ell,j,k_1,k_2}, \tag{6}$$

where $\{v_{\ell,j,k_1,k_2} = \langle f, \Psi_{\ell,j,k_1,k_2} \rangle\}$ with Ψ_{ℓ,j,k_1,k_2} defined in (5). We call v_{ℓ,j,k_1,k_2} the *curvelet coefficients* of f.

2.2. Problem formulation and analysis

2.2.1 Sparse representation under framelet and curvelet system

The perfect reconstruction formulas of both (4) and (6) essentially give us the decomposition and reconstruction of a function in $L_2(\mathbb{R}^2)$ under a tight frame system. In the discrete implementation, we have also a completely discrete form of the decomposition and reconstruction.

If we denote an $n \times n$ image f as a vector $\mathbf{f} \in \mathbb{R}^{n^2}$ by concatenating all columns of the image, The perfect reconstruction (4) can be rewritten as

$$\mathbf{f} = \mathcal{A}^T \mathcal{A} \mathbf{f}, \tag{7}$$

where $\mathcal{A}$ the decomposition operator of the tight framelet system in discrete case and the reconstruction operator is its transpose $\mathcal{A}^T$. Thus we have $\mathcal{A}^T\mathcal{A} = I$. We emphasize that $\mathcal{A}$ usually is a rectangular matrix with row dimension much larger than column dimension and $\mathcal{A}\mathcal{A}^T \neq I$ unless it is the orthonormal basis. In our implementation. We use a multilevel tight framelet decomposition without down-sampling the same as Cai *et al.* [4] does. Discrete Curvelet transform is also linear by thinking of the output as a collection of coefficients obtained from the digital analog to (6) ([7]).

In summary, given an image (kernel) $\mathbf{f} \in \mathbb{R}^{n^2}$, we have

$$\mathbf{f} = \mathcal{A}^T \mathbf{u} = \mathcal{A}^T(\mathcal{A}\mathbf{f}); \quad \text{and} \quad \mathbf{f} = \mathcal{C}^T \mathbf{v} = \mathcal{C}^T(\mathcal{C}\mathbf{f}), \tag{8}$$

where $\mathcal{A}$ and $\mathcal{C}$ are decomposition operator under framelet system and curvelet system respectively, $\mathbf{u} = \mathcal{A}\mathbf{f}$ and $\mathbf{v} = \mathcal{C}\mathbf{f}$ are the corresponding framelet coefficients and curvelet coefficients. In the implementation, there exist fast multiscale algorithms for the framelet (curvelet) decomposition and reconstruction ([20, 7]).

2.2.2 Formulation of our minimization

Given a blurred image $f = g * p + n$, our goal is to recover the latent image g and the blur kernel p. In this paper, we denote the image g (or the kernel p) as a vector $\mathbf{g}$ (or $\mathbf{p}$) in R^{n^2} by concatenating its its columns. Let "$\circ$" denote the usual 2D convolution after column concatenation, then we have

$$\mathbf{f} = \mathbf{g} \circ \mathbf{p} + \mathbf{n}. \tag{9}$$

Let $\mathbf{u} = \mathcal{A}\mathbf{g}$ denote the framelet coefficients of the clear image $\mathbf{g}$, and let $\mathbf{v} = \mathcal{C}\mathbf{p}$ denote the curvelet coefficients of the blur kernel $\mathbf{p}$. The perfect reconstruction property of tight frame system (8) yields

$$\mathbf{f} = \mathbf{g} \circ \mathbf{p} + \mathbf{n} = (\mathcal{A}^T\mathbf{u}) \circ (\mathcal{C}^T\mathbf{v}) + \mathbf{n}. \tag{10}$$

We take a regularization-based to solve the blind motion deblurring problem (10). The most commonly used approach to tackle such a challenging minimization problem is the alternative iteration scheme. Thus, we formulate the basic procedure of our approach as follows: for $k = 0, 1, \cdots$,

1. given the curvelet coefficients of the blur kernel $\mathbf{v}^{(k)}$, compute the framelet coefficients of the latent image $\mathbf{u}^{(k+1)}$ by solving
$$\operatorname*{argmin}_{\mathbf{u}} \|\mathbf{u}\|_1, \text{ s.t. } \|(\mathcal{C}^T\mathbf{v}^{(k)}) \circ (\mathcal{A}^T\mathbf{u}) - \mathbf{f}\|_2 \leq \delta; \tag{11}$$
2. given the framelet coefficients of the latent image $\mathbf{u}^{(k+1)}$, compute the framelet coefficients of the blur kernel $\mathbf{v}^{(k+1)}$ by solving
$$\operatorname*{argmin}_{\mathbf{v}} \|\mathbf{v}\|_1, \text{ s.t. } \|(\mathcal{A}^T\mathbf{u}^{(k+1)}) \circ (\mathcal{C}^T\mathbf{v}) - \mathbf{f}\|_2 \leq \delta. \tag{12}$$

δ in (11) and (12) is the noise parameter of the observed image $\mathbf{f}$.

The role of the objective function and the constraint in (11)–(12) is explained as follows. It is known that ℓ_1 norm is a good measurement on the sparseness of the solution when the linear system is under-determined. Thus, the first term in (11) measures the ℓ_1 norm of framelet coefficients of the latent image $\mathbf{g}$, which penalizes the sparsity of the image $\mathbf{g}$ under framelet system. The constraint in (11) is the fidelity constraint between the observed blur image and the resulted blur image. The same argument is also applicable to (12).

Therefore, the motivation in our formulation is among all solutions which have reasonable good ℓ_2 norm approximation to the given blurred image $\mathbf{f}$, we are seeking for the one whose recovered image $\mathbf{g}$ is the sparsest solution in framelet domain and the estimated blur kernel $\mathbf{g}$ is the sparsest solution in curvelet domain. At a quick glance, (11) and (12) are quite challenging large-scale minimization problems. In next section, we will present an alternative minimization approach to solve these two minimizations efficiently. The key idea is to adopt a modified version of linearized Bregman iteration technique which recently is emerging as a powerful technique in sparse approximation for visual information process.

3. Numerical algorithm and analysis

In practice, the minimizations (11) and (12) may not always yield a physical solution. Thus, we chose to impose the following physical conditions:

$$\left\{\begin{array}{l} \mathbf{p} = \mathcal{C}^T\mathbf{v} \geq 0, \quad and \quad \sum \mathbf{p} = 1; \\ \mathbf{g} = \mathcal{A}^T\mathbf{u} \geq 0. \end{array}\right. \tag{13}$$

which says both the kernel and the image are non-negative and the kernel is normalized. Algorithm 1 outlines our alternative iteration approach. Among all steps of Algorithm 1, there exist only two difficult problems (14) and (15) of the same type. For such a large-scale minimization problem with up to millions of variables, there exists a very efficient algorithm based on so-called *linearized Bregman iteration* technique. Let $[\mathbf{p}]_*$ denote the matrix form of the convolution operator by the kernel $\mathbf{p}$. The algorithm for solving (14) is presented in Algorithm 2. And Algorithm 2 can be applied to solve (15) with little modifications.

Algorithm 1 Outline of the algorithm for blind motion deblurring

```
1. Set $\mathbf{v}^{(0)} = \mathcal{C}\mathbf{f}, \mathbf{u}^{(0)} = \mathcal{A}\delta_0,$, where $\delta_0$ is the Delta function.
2. Iterate on $k$ until convergence.
   a Fixing the curvelet coefficients $\mathbf{v}^{(k)}$, solve (11) w.r.t. $\mathbf{u}$, i.e., set $\mathbf{u}^{(k+1/2)}$ be a solution of
        $\min_{\mathbf{u}} \|\mathbf{u}\|_1 \quad \text{s.t.} \quad \|(\mathcal{C}^T\mathbf{v}^{(k)}) \circ (\mathcal{A}^T\mathbf{u}) - \mathbf{f}\|_2 \leq \delta,$   (14)
     Then impose $\mathbf{u}^{(k+1)} = \mathcal{A}\mathbf{g}^{(k+1)}$, where
        $\mathbf{g}^{(k+1)}(j) = \left\{\begin{array}{ll} \mathcal{A}^T\mathbf{u}^{(k+\frac{1}{2})}(j), & \text{if} \mathcal{A}^t\mathbf{u}^{(k+\frac{1}{2})}(j) \geq 0, \\ 0, & \text{otherwise.} \end{array}\right.$
   b Fixing the framelet coefficients $\mathbf{u}^{(k+1)}$, solve (12) w.r.t. $\mathbf{v}$, i.e., set $\mathbf{v}^{(k+1)}$, be a solution of
        $\min_{\mathbf{u}} \|\mathbf{v}\|_1 \quad \text{s.t.} \quad \|(\mathcal{A}^T\mathbf{u}^{(k+1)}) \circ (\mathcal{C}^T\mathbf{v}) - \mathbf{f}\|_2 \leq \delta.$   (15)
     Then impose $\mathbf{v}^{(k+1)} = \mathcal{C}\frac{\mathbf{p}^{(k+1)}}{\|\mathbf{p}^{(k+1)}\|_1}$, where
        $\mathbf{p}^{(k+1)}(j) = \left\{\begin{array}{ll} \mathcal{C}^T\mathbf{v}^{(k+\frac{1}{2})}(j), & \text{if} \mathcal{C}^T\mathbf{v}^{(k+\frac{1}{2})}(j) \geq 0, \\ 0, & \text{otherwise.} \end{array}\right.$
     followed by the normalization $\mathbf{p}^{(k+1)} := \frac{\mathbf{p}^{(k+1)}}{\|\mathbf{p}^{(k+1)}\|_1}$.
```

Algorithm 2 Algorithm for solving (14)

```
1 Set $\mathbf{w}^{(0)} = \mathbf{x}^{(0)} = \mathbf{0}$.
2 Iterate on $i$ until $\|\mathcal{C}^T\mathbf{v}^{(k)}) \circ (\mathcal{A}^T\mathbf{w}^{(i+1)} - \mathbf{f}\|_2 \leq \delta$,
     $\left\{\begin{array}{l} \mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \mathcal{A}[\mathcal{C}\mathbf{v}^{(k)}]_*^T(\Omega^{(k)}([\mathcal{C}\mathbf{v}^{(k)}]_*(\mathcal{A}^T\mathbf{w}^{(i)} - \mathbf{f}))), \\ \mathbf{w}^{(i+1)} = \nu\Gamma_\mu(\mathbf{x}^{(i+1)}), \end{array}\right.$   (16)
  where $\Gamma_\mu$ is the soft-thresholding operator defined by
     $\Gamma_\mu(\mathbf{x})(j) = sign(\mathbf{x}(j)) \max(|\mathbf{x}(j)| - \mu, 0),$
  and $\Omega^{(k)}$ is a pre-conditioning matrix defined by
     $\Omega^{(k)} = \left([\mathcal{C}\mathbf{v}^{(k)}]_*^T([\mathcal{C}\mathbf{v}^{(k)}]_*) + \lambda\Delta\right)^{-1},$   (17)
  with $\Delta$ being the discrete Laplacian.
3 $\mathbf{u} = \mathbf{w}^{(i+1)}$.
```

Proposition 3.1[6] *The sequence* $\mathbf{w}^{(i)}$ *generated via* (16) *with a proper* ν *converges to the unique solution of*

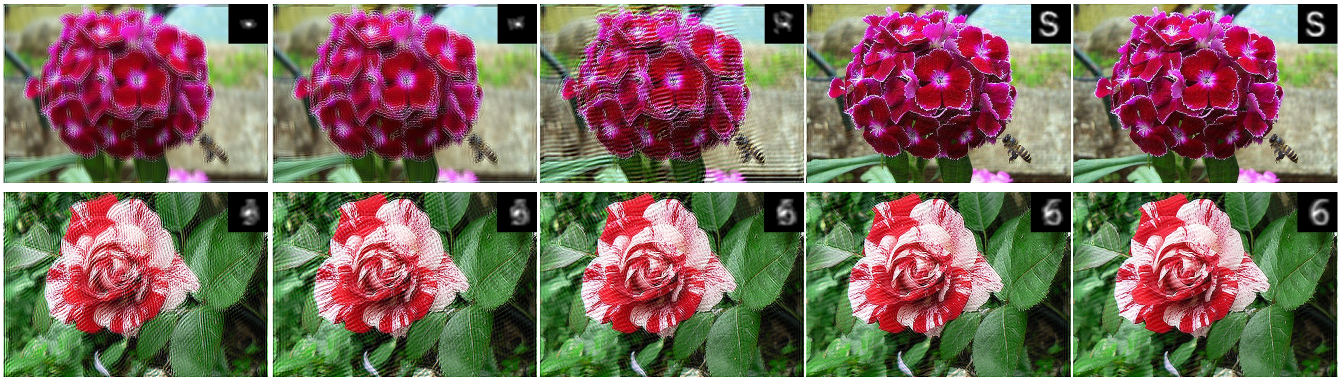
$$\left\{\begin{array}{l} \min_{\mathbf{u}} \|\mathbf{u}\|_1 + \frac{1}{2\nu\mu}\|\mathbf{u}\|_2^2, \\ \text{s.t. } (\mathcal{C}^T\mathbf{v}^{(k)}) \circ (\mathcal{A}^T\mathbf{u}) = \mathbf{f}, \end{array}\right. \tag{18}$$

if there exists at least one solution of $(\mathcal{C}^T\mathbf{v}^{(k)}) \circ (\mathcal{A}^T\mathbf{u}) = \mathbf{f}$.

Proposition 3.1 showed that the sequence $\mathbf{w}^{(i)}$ generated by (16) actually converges to an approximated solution of (14) when $\nu\mu \to \infty$. In other words, the larger is the value of ν or μ in (16), the better the solution from Algorithm 2 approximates the true solution of (14). The method (16) is extremely efficient. In the implementation, the value of μ is set to be small for solving (16) until the last few iterations in Algorithm 1. Usually it takes only a few iterations for (16) to get a fair approximation to the solution of (14) ([6]), and the accuracy of the approximation is adequate during the iterations of Algorithm 1 until the last a few steps.

4. Experiments and conclusions

In our implementation, the initial kernel is set as a delta function and the initial image is the given blurred image. The maximum iteration number of Algorithm 1 is set to 100. λ in (17) is set as 0.001. The parameters in (16) are



(a) Iteration 10

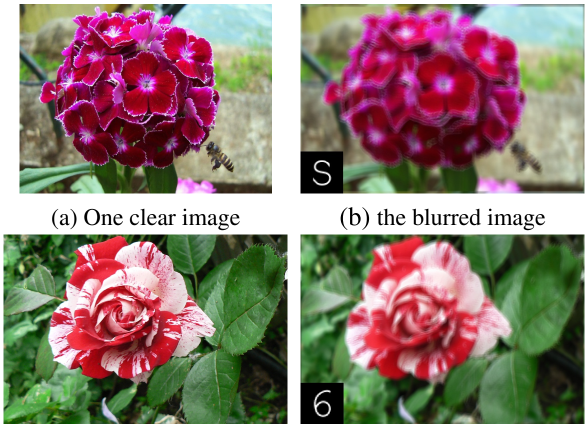
(b) Iteration 20

(c) Iteration 30

Iteration 40

Iteration 50

Figure 3. (a)–(e) are the recovered images during k-th iteration when applying the algorithm in Section 3 on deblurring the image shown in Fig. 2 (a) and (b) respectively, for $k = 10, 20, \cdots, 50$. The estimated blur kernels are shown on the top right of the corresponding images.



(a) One clear image

(b) the blurred image

(c) Another clear image

(d) the blurred image

Figure 2. (a) and (c) are two clear images; (b) and (d) are the blurred image of (a) and (c) with the kernel shown on the bottom left of the images.

chosen as $\mu = 0.2\|\mathbf{x}\|_\infty$ and $\nu = 1$. Each iteration in Algorithm 1 takes roughly 9 seconds on a windows PC with an Intel 2 GHz CPU for blurred color images with the resolution 1280×1024.

4.1. Simulated images

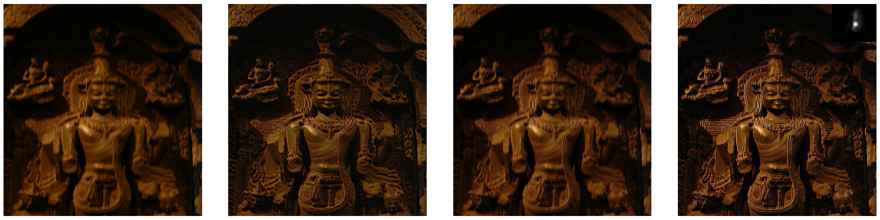
In the first part of the experiment, we synthesized two examples to verify the efficiency and performance of the algorithm proposed in Section 3. The first example used in the experiment is shown in Fig. 2 (a)–(b). (a) is the image of a clear flower with an out-of-focus blurred background. The synthesized blurred image is shown in Fig. 2 (b) with the corresponding blur kernel is shown on the bottom left of the blurred image. The second example is shown in Fig. 2 (c)–(d). (a) is the clear image of a clear flower with a clear background. Its blurred version is shown in Fig. 2 (d) with a more complicated blur kernel.

The intermediate results obtained on each 10 iteration are shown to illustrate the convergence behavior of our algorithm. The intermediately recovered image are shown in Fig. 3 (a)–(f). with the estimated kernel shown on the top right of the corresponding image. Our algorithm is quite efficient, as it only takes fifty iterations to obtain a clear image with accurate estimation on the blur kernel.

4.2. Real images

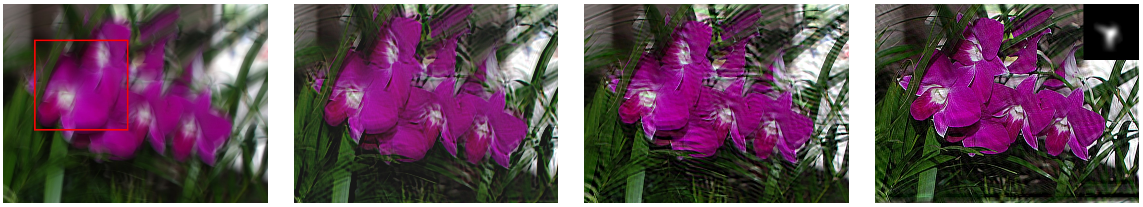
In the second part of the experiment, We tested Algorithm 1 on two real tele-photos taken by a Nikon D80 DSLR camera and one image from [15]. We compared our results against the results of other single-image based methods with available codes. One is Fergus *et al.*'s method ([15]); and the other is Shan *et al.* 's method ([25]). Both methods require the input of the kernel size, which could be very important when the size of motion-blur kernel is the large ($\geq$ 30 pixels). Therefore, we run both methods on the blurred image using three kernel sizes $\{25, 35, 45\}$ as the input. Then the best result are manually selected by visually inspection on all three recovered images.

Fig. 4–6 showed the comparisons among three different methods on three tested real blurred images of different structures. Fig. 7 showed one small region of the results from Fig. 4–6 for easier visual inspection. The blur kernels estimated by Algorithm 1 are shown on the top right of the corresponding recovered images. The blurred image in Fig. 4 is from [15], and our algorithm also successfully recovered the image as Fergus *et al.*'s method did. In Fig. 5, Both Fergus *et al.*'s method and Shan *et al.*'s method only partially deblurred the image and the resulted image still looks blurred. The result from Algorithm 1 is the most visually pleasant. In Fig. 6, the result from Algorithm 1 is more crisp than the results from both Fergus *et al.*'s method and Shan *et al.*'s method. One possible cause why Fergus



(a) Blurred image (b) Fergus *et al*. [15] (c) Shan *et al*. [25] (d) Algorithm 1

Figure 4. (a): the blurred image; (b)–(d): recovered images using the method from [15]; from [25] and from Algorithm 1 respectively.



(a) Blurred image (b) Fergus *et al*. [15] (c) Shan *et al*. [25] (d) Algorithm 1

Figure 5. (a): the blurred image; (b)–(d): recovered images using the method from [15], [25] and Algorithm 1 respectively.

et al.'s method and Shan *et al*.'s method did not perform well on some tested images is that these images do not obey the edge distribution rules used in these two method. Also the local prior edge model used in Shan *et al*.'s method depends on the pre-processing of extracting local smooth structure from blurred image, which could be instable for heavily blurred image with rich textures.

4.3. Conclusions

In this paper, a new algorithm is presented to remove camera shake from a single image. Based on the high sparsity of the image in framelet system and the high sparsity of the motion-blur kernel in curvelet system, our new formulation on motion deblurring leads to a powerful algorithm which can recover a clear image from the image blurred by complex motion. Furthermore, the curvelet-based representation of the blur kernel also provides a good constraint on the curve-like geometrical support of the motion blur kernel, thus our method will not converge to the degenerate case as many other approaches might do. As a result, our method does not require any prior information on the kernel while existing techniques usually needs user interactions to have some accurate information of the blurring as the input. Moreover, a fast numerical scheme is presented to solve the resulted minimization problem with convergence analysis. The experiments on both synthesized and real images show that our proposed algorithm is very efficient and also effective on removing complicated blurring from nature images of complex structures. In future, we would like to extend this sparse approximation framework to remove local motion blurring from the image caused by fast moving objects.

References

[1] 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.

[2] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, pages 573–582, 1996.

[3] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *IEEE Trans. PAMI*, 26(6):689–698, 2004.

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

[5] J. Cai, S. Osher, and Z. Shen. Linearized bregman iterations for frame-based image deblurring. *UCLA CAM Reports (08-52)*, 2008.

[6] J. Cai, S. Osher, and Z. Shen. Linerarized bregman iterations for compressed sensing. *UCLA CAM Reports (08-06)*, 2008.

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

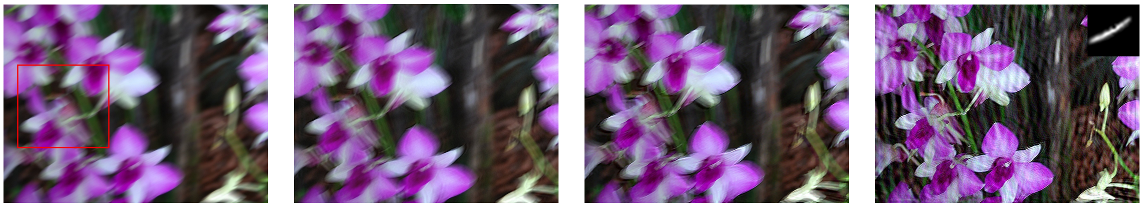
[8] E. Candes and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise-C^2 singularities. *Comm. Pure Appl. Math*, 57:219–266, 2002.

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

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

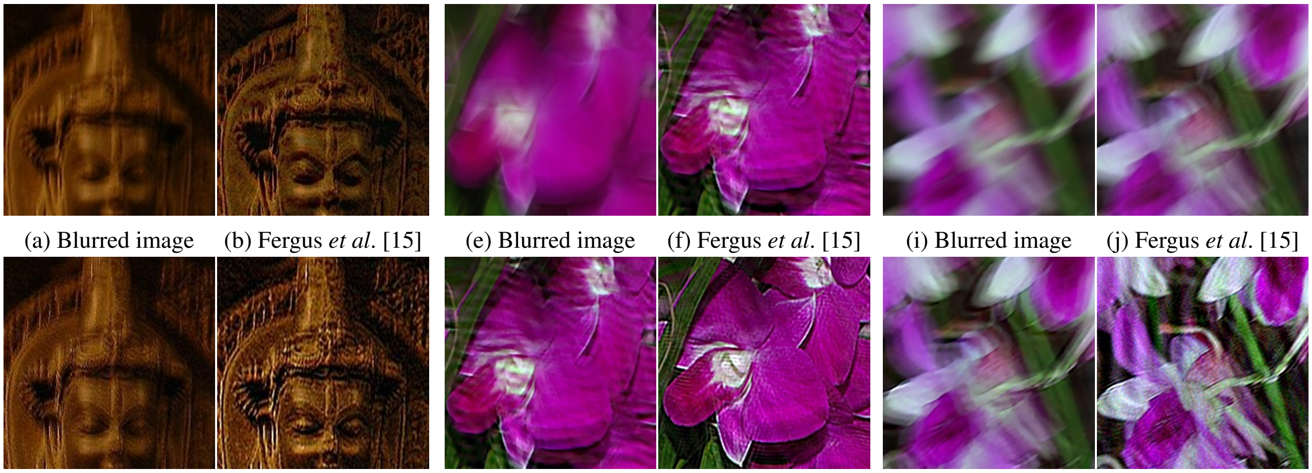
[11] S. H. Cho, Y. Matsushita, and S. Y. Lee. Removing non-uniform motion blur from images,. In *ICCV*, 2007.

[12] I. Daubechies, M. Defrise, and C. D. Mol. An iterativethresholding algorithm for linear inverse problems with asparsity constraint. *Comm. Pure Appl. Math*, 57(11):1413–1457, 2004.



(a) Blurred image (b) Fergus *et al*. [15] (c) Shan *et al*. [25] (d) Algorithm 1

Figure 6. (a): the blurred image; (b)–(d): recovered images using the method from [15], [25] and Algorithm 1 respectively.



(a) Blurred image (b) Fergus *et al*. [15] (e) Blurred image (f) Fergus *et al*. [15] (i) Blurred image (j) Fergus *et al*. [15]

(c) Shan *et al*. [25] (d) Algorithm 1 (g) Shan *et al*. [25] (h) Algorithm 1 (k) Shan *et al*. [25] (l) Algorithm 1

Figure 7. (a)-(d) show the zoomed regions of each image in Fig. 4; (e)-(h) show the zoomed regions of each image in Fig. 5; (i)-(l) show the zoomed regions of each image in Fig. 6. The zoomed regions are marked by red rectangles in the original blurred images.

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

[14] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.

[15] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *SIGGRAPH*, volume 25, pages 783–794, 2006.

[16] J. Jia. Single image motion deblurring using transparency. In *CVPR*, pages 1–8, 2007.

[17] N. Joshi, R. Szeliski, and D. Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, 2008.

[18] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, Dec. 2006.

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

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

[21] S. Osher, Y. Mao, B. Dong, and W. Yin. Fast linearized bregman iteration for compressive sensing and sparse denoising. *Communications in Mathematical Sciences.*, To appear.

[22] G. Pavlovic and A. M. Tekalp. Maximum likelihood parametric blur identification based on a continuous spatial domain model. *IEEE Trans. Image Processing*, 1(4), Oct. 1992.

[23] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring via fluttered shutter. In *SIGGRAPH*, volume 25, pages 795–804, 2006.

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

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

[26] Y. Tai, H. Du, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *CVPR*, 2008.