A two-stage approach to blind spatially-varying motion deblurring

Hui Ji and Kang Wang
Department of Mathematics, National University of Singapore, Singapore 117543, Singapore
matjh, wangkang@nus.edu.sg

Abstract

Many blind motion deblur methods model the motion blur as a spatially invariant convolution process. However, motion blur caused by the camera movement in 3D space during shutter time often leads to spatially varying blurring effect over the image. In this paper, we proposed an efficient two-stage approach to remove spatially-varying motion blurring from a single photo. There are three main components in our approach: (i) a minimization method of estimating region-wise blur kernels by using both image information and correlations among neighboring kernels, (ii) an interpolation scheme of constructing pixel-wise blur matrix from region-wise blur kernels, and (iii) a non-blind deblurring method robust to kernel errors. The experiments showed that the proposed method outperformed the existing software based approaches on tested real images.

1. Introduction

The image created by the camera represents the integration of all positions of the scene over the exposure time. If there is a relative motion between the camera and the scene during the exposure time, the resulting image will look blurry, known as *motion blurring* in digital photography. *Blind motion deblurring* is then about recovering a clear image with sharp edges from the input motion-blurred image without knowing all information of the blurring process. Camera shake is one common cause of motion blurring which means the camera moved during the exposure.

Most existing blind motion deblurring methods for correcting blurred images caused by camera shake (*e.g*. [2, 8, 16, 18, 22, 28]) assume that the relative movements between all point of the scene and the camera are the same. Such a motion blurring process is then modeled as the convolution of a clear image with a blur kernel plus noise:

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

where the symbol $*$ denotes the convolution operator, g denotes the clear image for recovery, f denotes the given blurry image, k denotes the motion-blur kernel and n denotes the image noise. The convolution model (1) holds true only when the camera translates along the image plane and the scene is nearly flat and parallel to the image plane.

However, as illustrated in [19, 27], the motion blur caused by camera shake is often due to camera rotations which lead to different movements for different scene points. Also, the assumption that the scene is nearly parallel to the image plane is often satisfied in practice. In other words, the motion blurring caused by camera shake often is a spatially varying process. Mathematically speaking, such a spatially varying blurring process can be expressed in a matrix-vector form:

$$\mathbf{f} = \mathbf{K}\mathbf{g} + \mathbf{n} \tag{2}$$

where $\mathbf{f}$, $\mathbf{g}$, $\mathbf{n}$ denote the column-wise vector forms of f, g and n respectively, and the matrix $\mathbf{K}$ denotes the blurring matrix. For spatially invariant blurring process, each row of $\mathbf{K}$ corresponds to the same low-pass filter up to a spatial shift while each row of $\mathbf{K}$ may correspond to a different low-pass filter for spatially invariant blurring process.

This paper aims at developing an efficient blind motion deblurring method for removing spatially varying blind motion blurring from a single photograph. A two-stage approach is proposed in this paper which first estimates the pixel-wise blur matrix $\mathbf{K}$ of the spatially-varying blurring process; and then deblurs the image using a robust non-blind deblurring technique in the presence of the unavoidable system error in $\mathbf{K}$.

1.1. Related work

Image deblurring is one long-last problem in image processing with an abundant research literature. There are two tightly coupled sub-problems: (a) estimating the blur kernel k or the blurring matrix $\mathbf{K}$); and (b) estimating a clear image $\mathbf{g}$ using the kernel k or the matrix $\mathbf{K}$. The so-called *blind image deblurring* refers to how to solve both sub-problems and *non-blind image deblurring* only refers to how to solve the second sub-problem. Due to the limitation of the space, we only discuss the most related software-based non-blind and blind deblurring techniques.

Non-blind image deblurring is an ill-conditioned problem such that simply reversing the blurring process will amplify the noise. Early works either suppress the impact of noise in the frequency domain using the Wiener filter or take a Bayesian-based iterative procedure to deblur images, *e.g*. the *Richardson-Lucy* (RL) method [21]). In recent years, regularization-based methods have been more popular for better performance, which rely on certain priors of clear images. The Tikhonov regularization is proposed in [26] to enforces the smoothness of the underlying image. Total variation (TV) and its variations (see e.g. [4, 5, 22]) are developed to keep image edges sharper as the Tikhonov regularization tends to smooth out image edges. The sparsity regularization of images under wavelet transform also are used to regularize image deblurring (*e.g*., [3]). Recently, some non-blind image deblurring methods have been proposed to handle the images blurred by a spatially varying blurring process. The RL method is adapted in [25] to model the spatially varying motion blurring. Hirsch *et al*. in [12] introduced a so-called *efficient filter flow* technique of filtering images in a spatially varying fashion.

Blind image deblurring is a more challenging problem due to the loss of information on both the image and the blurring process. In recent years, there have been great progresses on removing spatially invariant motion blurring caused by camera shake from images. Most uniform blind motion deblurring methods relies on certain priors on the image/kernel to regularize the estimation of both the clear image and the blur kernel. In [8, 16, 18], certain heavy tailed probabilistic distributions on the image edges of a clear image are first proposed to derive the blur kernel from a single image. The total variation (TV) and its variations (e.g. [4,5,22,28]) are used in blind motion deblurring to regularize the clear image, the blur kernel, or both of them. The normalized TV of images is used in [17] for more accurately regularizing images with sharp edges. The wavelet tight frame based sparsity prior for both the image and the kernel is used in [2] to remove uniform motion blurring from a single image.

There has been relatively little work on spatially varying blind motion blurring. Some existing approaches either use additional information source such as alpha map in Dai *et al*. [6] or require user interactions for came motion cues (e.g., Shan *et al*. [23]). The approach proposed by Levin *et al*. [18] is used to deblur spatially varying motion-blurred images by segmenting the image into several areas with different motion blurring effects and then individually deblurring each region. A multi-frame patch-based deblurring approach is developed by Hirsch *et al*. in [12] to deblur images with spatially varying motion blurring effects,

Another promising approach to spatially varying blind motion deblurring is not to directly estimate spatially-varying blur kernels, but to recover the camera motion from which the blur kernels is then derived. Whyte *et al*. [27] proposed a 3-dimensional rotational camera motion model to model the spatially varying motion blur. Gupta *et al*. [10] introduced the motion density function to represent a a different set of 3D camera motions (in-plane rotation and translations) and assume there is no significant depth variations in the scene. Then the spatially varying kernel function is derived from the estimated motion density function. Also, such ideas are applied in [11] to their multi-frame patch-based deblurring approach.

1.2. Motivation and outline of our approach

This paper focuses on how to remove spatially varying motion blur from a single photograph without using any additional information source. The available methods under the similar problem settings have both their own advantages and disadvantages. The methods proposed by Whyte *et al*. [27] and Gupta *et al*. [10] effectively reduced the dimension of the space of spatially varying kernels by only considering a subset of all possible 3D camera motions. Then the pixel-wise spatially varying blurring model can be derived from the estimated camera motion. The impressive results are reported in both [27] and [10] for sufficiently long focal lengths. But at short focal lengths, the results of these two methods are not very satisfactory owing to the large errors when approximating the actual camera motion using their camera motion models.

The direct motion de-blurring methods proposed by [18] takes a region-wise spatially invariant model and deblurring each region separately. There are several limitations in the method by Levin [18]. One is the poor handling of boundaries between areas with different blur kernels; and the other is instability of local kernel estimation due to the lack of sufficient image details in some local regions. As a result, the visual quality of the results from [18] are not as good as that from [27] and [10]. However, the direct method is still an attractive approach since it is not restricted to any particular type of camera motions.

Thus, in this paper, we take a new direct deblurring approach to spatially varying blind motion deblurring which does not have the limitations existing in the existing approaches. The outline of our tow-stage approach is as follows. After partitioning the image into multiple regions,

Stage 1: Estimation of pixel-wise motion-blur matrix $\mathbf{K}$.

1. Initializing the estimation of region-wise blur kernels by running some existing uniform blind deblurring method on each region.
2. Identify erroneous local kernels and re-estimating them by utilizing both the associated image region information and the correlations among the estimating kernel and the kernels of the neighboring regions.
3. Synthesizing the pixel-wise blur function $\mathbf{K}$ using an

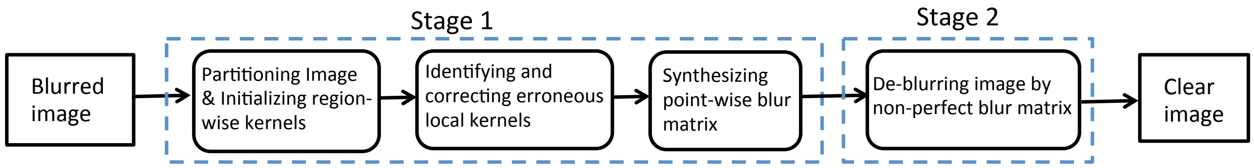


Figure 1. Outline of the proposed spatially varying blind image deblurring approach.

interpolation scheme designed for kernel interpolation.

Stage 2: Deblurring the image globally via a non-blind image deconvolution technique that is robust to errors in the estimated blur matrix $\mathbf{K}$.

It is shown in the experiments that the proposed method is efficient and performs better than the two state-of-art methods developed in [27] and [10] on the tested image data set, including the images from both [10, 27] and ours.

2. Preliminaries on wavelet tight frame

Wavelet tight frame theory is the main tool used in our approach. In this section, we give a very brief review on wavelet tight frame. The readers are referred to [7, 24] for in-depth theoretical analysis and numerical implementation.

A wavelet frame system is a redundant system that generalizes the orthonormal wavelet basis. Wavelet tight frames have greater flexibility than orthonormal bases by sacrificing orthonormality and linear independence while keeping the same perfect reconstruction as orthonormal wavelet bases. The filters associated wavelet frame systems have many attractive properties for representing image, not present in orthonormal wavelet systems: *e.g.*, symmetry (anti-symmetry), smoothness, and shorter support, which make it a powerful tool for image restoration.

A MRA-based wavelet frame system is based on a single scaling function $\phi \in L^2(\mathbb{R})$ and several wavelet functions $\{\psi_1, \dots, \psi_r\} \subset L^2(\mathbb{R})$ that satisfy the following refinable equation:

$$\left\{ \begin{array}{lcl} \phi(t) & = & \sqrt{2}\sum_k h_0(k)\phi(2t-k); \\ \psi_\ell(t) & = & \sqrt{2}\sum_k h_\ell(k)\phi(2t-k),\ \ell = 1,2,\dots,r. \end{array} \right.$$

Let $\phi_k(t) = \phi(t-k)$ and $\psi_{k,j,\ell} = \psi_\ell(2^j t - k)$. Then for any square integrable function $f \in L^2(\mathbb{R})$, we have a multi-scale wavelet frame decomposition of f:

$$\{c_k = \langle f, \phi_k\rangle;\ d_{k,j,\ell} = \langle f\psi_{k,j,\ell}\rangle, j,k \in \mathbb{Z}, j \geq 0, 1 \leq \ell \leq r\}, \tag{3}$$

where $\{c_k\}$ and $\{d_{k,j,\ell}\}$ denote the low-pass and high-pass wavelet frame coefficients respectively. The wavelet tight frame has the so-called *perfect reconstruction* property that allows the exact recovery of f from its wavelet coefficients:

$$f = \sum_{k=-\infty}^{\infty} c_k\phi_k(t) + \sum_{\ell=1}^{r}\sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d_{k,j,\ell}\psi_{k,j,\ell}, \tag{4}$$

The wavelet coefficients can be efficiently calculated by a so-called cascade algorithm (see e.g. [20]). In this paper, we use the linear wavelet frame developed in [7]:

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

In this paper, we use the implementation of the 2D tensor un-decimal multi-level framelet transform same as Cai et al [1]. Interesting readers are referred to [1] for more implementation details. In the remains of this paper, we denote the wavelet frame decomposition (3) by a rectangular matrix $\mathcal{W}$ of size $m \times n$ with $m >> n$. Thus, given any signal $\mathbf{f} \in \mathbb{R}^n$, the discrete version of (4) is expressed as

$$\mathbf{f} = \mathcal{W}^\top(\mathbf{c}) = \mathcal{W}^\top(\mathcal{W}\mathbf{f}),$$

where $\mathbf{c} \in \mathbb{R}^m$ is the frame coefficient vector of $\mathbf{f}$. It is noted that we have $\mathcal{W}^\top\mathcal{W} = \mathcal{I}_n$ but $\mathcal{W}\mathcal{W}^\top \neq \mathcal{I}_n$ unless the tight framelet system degenerates to an orthonormal wavelet system. We need to mention here that there exist fast algorithms for calculating $\mathcal{W}\mathbf{f}$ and $\mathcal{W}^\top\mathbf{c}$, which only involve the convolutions of images by a couple of filters.

3. Main algorithm

In this section, we give a detailed discussion on our proposed spatially varying blind motion deblurring approach. The alternating scheme that iteratively updating the estimation of the blur kernel and the clear image used by most uniform blind motion deblurring method is too slow in our case. Thus we take a more efficient two-stage approach. See Fig. 1 for the outline of the main approach. The details on each step is given as follows.

3.1. Stage I: Estimating point-wise blur matrix K

Given a blurry image, we first partition it into multiple over-lapping image regions, denoted by $\{P_i\}_{i=1}^n$. For each image region, we run a uniform blind motion deblurring method to generate an initial guesss on region-wise blur kernels. There have been great progresses on uniform blind motion deblurring with many powerful methods available, *e.g.* [2, 8, 16, 18, 22, 28]. Owing to its computational efficiency, the two-stage method proposed by Xu *et al.* [28] is used in our implementation for estimating the blur kernel k_i associated with each image region P_i. Its basic idea is to first estimate the support of the blur kernel and then alternatively refine the estimation of the clear image and the blur kernel using the TV based regularization in the iterations.

3.1.1 Finding and correcting erroneous local kernels

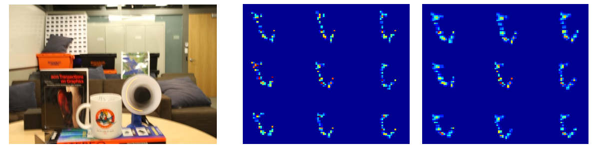
A significant number of local kernels initialized by the uniform blind motion deblurring are erroneous. The reason is either the image regions do not provide enough image edges needed by the uniform blind deblurring method to reliably estimate the kernel, or the actual blurring process in those regions is not well approximated by the convolution model (1). We will deal with the later case in Stage 2. This step is to correct the wrongly estimated kernels associated with image regions without sufficient edge information.

Detecting erroneous initial guesses of region-wise kernels. For each patch P_i, let k_i denote the kernel obtained from the previous step. We use the residual $r_i := \|k_i * u_i - P_i\|_2$ to measure its accuracy and set the accuracy threshold ϵ by $\epsilon := \frac{3}{2} \times \text{median}\{r_1, r_2, \ldots, r_n\}$. Then any local kernel whose residual r_i larger than the accuracy threshold ϵ is considered as wrongly estimated kernel and discarded.

Re-estimating erroneous local kernels. To re-estimate the discarded local blur kernels, we need some additional information outside these regions to help the estimation of blur kernels as these regions by themselves do not have sufficient image content for a reliable kernel estimation. It is observed that the blur kernels of neighboring regions are usually correlated unless the depths of the scene vary significant and fragmented. See Fig. 2 for an illustration. This motivates us to combine the local image information and the correlation among the kernel to estimate and the available kernels in its neighborhood.

The basic idea is as follows. For each image region P with discarded blur kernel, let $\widehat{k}$ denotes the blur kernel of its nearest neighboring image region with a well-estimated kernel. Then we explore the likely correlation by assuming the kernel k can be well approximated by applying an affine transform on $\widehat{k}$. By modeling the kernel k using the kernel $\widehat{k}$ up to an affine transform, we significantly reduce the dimension of unknowns of the kernel. As a result, the image information contained in the region P could be sufficient for estimating such a small number of unknowns.

To estimate the parameters of the affine transformation, we used the normalized ℓ_1 norm of image gradients to facilitate the estimation. It is first shown by Krishnan *et al.* in [17] that an image of sharp edges tends to minimize the normalized TV measure. It is also true for normalized ℓ_1 norm of high-pass wavelet coefficients (see [13]). For an image region u, the normalized ℓ_1 norm of wavelet coefficients is defined as ([13]):



(a) (b) (c)

Figure 2. An illustration of the high correlation of neighboring kernels up to an affine transform. (a) Blurry Image; (b) nine region-wise kernels obtained by the hardware in [15]; (c) eight kernels generated by applying affine transform on the kernel at the center.

$$\mathcal{L}[u] := (\sum_{i\neq 0} \|\mathcal{W}_i u\|_1)/(\sum_{i\neq 0} \|\mathcal{W}_i u\|_2), \tag{5}$$

where $\mathcal{W}_i u$ denote the wavelet tight frame coefficients of u in each high-pass channel.

The above observation leads to a very simple yet effective approach for our purpose. That is, we only need to find the affine transform parameters such that the image deblurred using the corresponding kernel has the minimal normalized ℓ_1 norm of wavelet coefficients. Let $\mathcal{A}_{\vec{\alpha}}$ denotes the affine transform parameterized by $\vec{\alpha}$ that maps the kernel $\widehat{k}$ to k, i.e., $k = \mathcal{A}_{\vec{\alpha}}\widehat{k}$. Then we estimate the parameter vector $\vec{\alpha}$ by minimizing the following function:

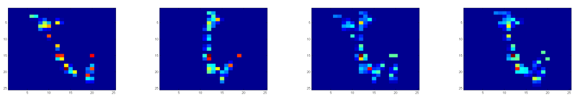
$$\min_{\vec{\alpha}} \mathcal{L}[g(\mathcal{A}_{\vec{\alpha}}\widehat{k}) * P], \tag{6}$$

where $g(\mathcal{A}_{\vec{\alpha}}\widehat{k})$ denote the Wiener filter for P with respect to the blur kernel $\mathcal{A}_{\vec{\alpha}}\widehat{k}$. To solve the minimization (6), we generate a set of random samples of parameters from the Gaussian distribution and then search for the one giving the minimal value of (6) over all the samples. See Fig. 4 (b)–(c) for an illustration of the erroneous kernel correction of one image region. It is noted that although the kernel correction in this step improved the initialized kernels, the corrected one is not perfect as there are still noticeable artifacts in the result. Such remained kernel error will be effectively addressed in Stage 2.

3.1.2 Synthesizing pixel-wise blurring matrix $\mathbf{K}$

Region-wise convolution model is still a very rough approximation to the spatially varying motion blurring. We need to estimate the pixel-wise spatially varying blur matrix $\mathbf{K}$ defined by (2). Considering the fact that millions of kernels are involved when generating $\mathbf{K}$, we take an interpolation scheme to efficiently synthesize the matrix $\mathbf{K}$ from the region-wise blur kernels obtained in the previous step.

The straightforward functional interpolation can not be applied to interpolate kernels. It is shown in Fig. 3 (c) that the regular interpolation scheme like bi-linear interpolation does not yield the result we want. Usually the supports of



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

Figure 3. Illustration of the kernel interpolation. (a)–(b): Two linear motion kernels used for interpolation with the same length and slightly different orientations; (c) the result using bi-linear interpolation; (d) the result using the proposed interpolation.

motion-blur kernels tend to have a dominant orientation. Thus, for each pixel p, before we generate the associated blur kernel, we first align the neighboring kernels to the same mass center point and to the same dominant axis, then use the bilinear interpolation scheme to generate the kernel.

For a given kernel k, let Ω denote the support of the kernel. Then the mass center $[c_x(k), c_y(k)]$ of k is defined by

$$\vec{r}_c(k) = \sum_{p\in\Omega} k(\vec{r})\vec{r}. \tag{7}$$

Let $\mathcal{S} = [\vec{r}]_{\vec{r}\in\Omega}$ denotes the $2\times n$ matrix formed by concatenating the co-ordinate vectors of all points in Ω. Then the dominant axis of the kernel k is defined as the first principal component of the PCA of $\mathcal{S}$. And the dominate direction of k is defined as the angle between its dominant axis and the horizontal axis. The detailed algorithm for synthesizing the pixel-wise blur matrix $\mathcal{A}$ is given in Algorithm 1. It is seen from the simple illustration shown in Fig. 3 (d) that the proposed interpolation scheme yields the desired kernel.

Algorithm 1 Kernel interpolation scheme for synthesizing pixel-wise blur matrix $\mathbf{K}$

Input: the region-wise blur kernels $\{k_1, k_2, \dots, k_n\}$
Output: the pixel-wise blur matrix $\mathbf{K}$ defined in (2)
Main procedure:

1. calculate the mass centers $\vec{r}_c(k_j)$ and the dominant orientation $\theta(k_j)$ of all region-wise kernels $\{k_i\}_{i=1}^n$.
2. For each pixel p, generate its kernel k as follows,
   (a) find all regions including the pixel p and denote their kernels by $\{k'_j\}_{j=1}^m$;
   (b) for each kernel k'_j, its weight ω_j is defined via the distance between p and the region center;
   (c) define the dominant orientation θ of the kernel k of the pixel p by $\theta \longleftarrow \sum_{j=1}^m \omega_j\theta(k'_j)$;
   (d) for each kernel k'_j, translate and rotate k'_j such that it centers at the original point and with the dominant orientation θ, defined by by $\widetilde{k'_j}$.
   (e) set $k \longleftarrow \sum_{j=1}^m \omega_j \widetilde{k'_j}$.
3. Compose the pixel-wise blur matrix $\mathcal{A}$ by collecting the kernels of all pixels.

3.2. Stage 2: Robust non-blind deblurring in the presence of kernel error

In Stage 2, we aim at deblurring the image by solving (2) using the blur matrix $\mathbf{K}$ estimated in Stage 1. The error in the estimated blur matrix $\mathbf{K}$ is unavoidable. However, most existing non-blind deblurring methods do not address such error source. By using existing deblurring methods, there are still noticeable artifacts in the results caused by error in the matrix $\mathcal{A}$. See Fig. 4 (c) for an illustration. Let $\widehat{\mathbf{K}}$ denote the estimated blur matrix containing errors:

$$\widehat{\mathbf{K}} = \mathbf{K} + \delta_{\mathbf{K}}$$

where $\mathbf{K}$ denotes the true blur matrix and $\delta_{\mathbf{K}}$ denotes the error. The EIV (errors-in-variables) model of (2) is then:

$$\mathbf{f} = \mathbf{K}\mathbf{g} + \mathbf{n} = (\widehat{\mathbf{K}} - \delta_{\mathbf{K}})\mathbf{g} + \mathbf{n} \tag{8}$$

Notice that the system error $\delta_{\mathbf{K}}$ cannot be estimated from image information anymore. Thus, there is a need to develop a non-blind deblurring method which is robust to the system error $\delta_{\mathbf{K}}$.

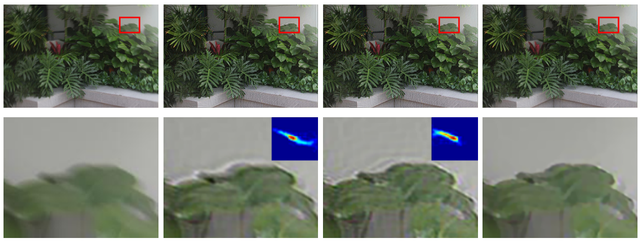
Our basic idea is similar to that of [14] but with a better minimization strategy. Rewriting (8) as

$$\mathbf{f} = (\widehat{\mathbf{K}} - \delta_{\mathbf{K}})\mathbf{g} + \mathbf{n} = \widehat{\mathbf{K}}\mathbf{g} - \delta_{\mathbf{K}}\mathbf{g} + \mathbf{n}. \tag{9}$$

Notice that $\delta_{\mathbf{K}}$ is the difference of two blur matrices which equals to a matrix with each row representing a high-pass filter. Thus the term $\delta_{\mathbf{K}}\mathbf{g}$ (9) is sparse in image domain as shown in Fig. 5. By representing the residual term $\delta_{\mathbf{K}}\mathbf{g}$ as an independent variable, we propose the following so-called *analysis-based* convex regularization for solving (9):

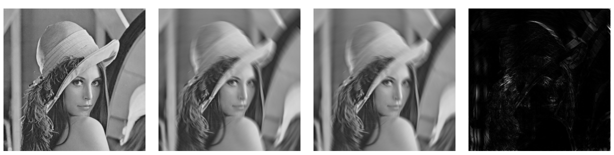
$$\{\mathbf{g}^*, \mathbf{u}^*\} = \operatorname*{argmin}_{\mathbf{g},\mathbf{u}} \frac{1}{2}\|\widehat{\mathbf{K}}\mathbf{g} - \mathbf{u} - \mathbf{f}\|_2^2 + \lambda_1\|\mathcal{W}\mathbf{g}\|_1 + \lambda_2\|\mathbf{u}\|_1, \tag{10}$$

where $\mathcal{W}$ denote the wavelet decomposition operator. The convex minimization model (10) simultaneously estimated both the clear image $\mathbf{g}$ and the residual $\delta_{\mathbf{K}}\mathbf{g}$ denoted by $\mathbf{u}$. The regularization term $\|\mathcal{W}\mathbf{g}\|_1$ in (10) is based on the assumption that the cardinal wavelet coefficients of a clear image is likely to be sparse, which is first used in [3] for non-blind image deconvolution. The regularization term $\|\mathbf{u}\|_1$ enforce the sparsity prior of $\delta_{\mathbf{K}}\mathbf{g}$ in image domain by minimizing its ℓ_1 norm. The model (10) differs from the one used in [14] by adopting an analysis-based model instead of the balanced model used in [14]. Interesting readers are referred to [24] for a more detailed discussions on these models. Also, some additional variables are introduced in [14] to suppress the boundary effect and kernel error propagation which is redundant in our case. Empirical studies showed the proposed model (10) has better performance for the case of spatially-varying blurring. The so-called Split Bregaman method first introduced in [9] can be easily modified to solve the minimization problem (10). The detailed algorithm are given in Alg. 2.



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

Figure 4. An illustration of kernel correction and robust deblurring. The full images are shown in the first row and each of their partitiontions are shown in the second row. (a) Blurry image; (b) the result deblurred by the wavelet method [3] using the blur matrix generated from initialized region-wise kernels; (c) the result deblurred by the wavelet method [3] using the blur matrix generated from the region-wise kernels re-estimated by Algorithm 1; (d) the result deblurred by Algorithm 2 proposed in Section 3.2 using the same blur matrix as (c). The kernel associated with the zoomed region used in (b) and (c) are shown in the top right corner.



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

Figure 5. An illustration of system error $\delta_{\mathbf{K}}\mathbf{g}$. The true blurring is a rotational blurring while the input is a region-wise uniform blurring. (a): the original image; (b) and (c): the blurry images by the true blur matrix $\mathbf{K}$ and the region-wise blurring model $\widehat{\mathbf{K}} = \mathbf{K} + \delta_{\mathbf{K}}$; (d): the system error $\delta_{\mathbf{K}}\mathbf{g}$. The values for (d) are amplified for better illustration.

Algorithm 2 Numerical algorithm for solving (10)

(i) Set initial guesses $\mathbf{g} = \mathbf{f}, \mathbf{u} = 0, \mathbf{d} = 0, \mathbf{b} = 0$ and set appropriate parameter $\lambda_1, \lambda_2, \mu$.

(ii) For $k = 0, 1, \ldots$, perform the following iterations until convergence:

$$\left\{\begin{array}{ll} \mathbf{g}^{k+1} & = (\widehat{\mathbf{K}}^\top\widehat{\mathbf{K}} + \mu\mathcal{I})^{-1}[\widehat{\mathbf{K}}^\top(\mathbf{f} + \mathbf{u}^k) \\ & \quad +\mu\mathcal{W}^\top(\mathbf{d}^k - \mathbf{b}^k)] \\ \mathbf{u}^{k+1} & = \mathcal{T}_{\lambda_2}(\widehat{\mathbf{K}}\mathbf{g}^{k+1} - \mathbf{f}) \\ \mathbf{d}^{k+1} & = \mathcal{T}_{\lambda_1/\mu}(\mathcal{W}\mathbf{g}^{k+1} + \mathbf{b}^k) \\ \mathbf{b}^{k+1} & = \mathbf{b}^k + (\mathcal{W}\mathbf{g}^{k+1} - \mathbf{d}^{k+1}) \end{array}\right. \tag{11}$$

where $\mathcal{T}_\lambda(x)(j) = \operatorname{sgn}(x(j))\max\{0, |x(j)| - \lambda\}$.

4. Numerical experiments and conclusion

Implementation details. For a given image, we uniformly partition it into around 50 image regions with half-size overlaps. When solving the minimization (6), the standard deviation of the Gaussian distribution is set to be 0.1 and 100 parameter samples are generated. The parameters for Alg. 2 are uniformly set as $\lambda_1 = 1e^{-3}, \lambda_2 = 1e^{-2}, \mu = 1$ for all experiments. The average time for processing an color image of size 768×512 is around 30 minutes by using a MATLAB (version 7.11) implementation on a PC with an AMD X4 3.0 Ghz CPU and 8 GB RAM.s One spatially invariant blind image deblurring method [28] (used to generate the initial guess of region-wise kernels in our implementation) and two spatially varying blind image deblurring algorithms, Whyte *et al*. [27] and Gupta *et al*. [10], are compared to our proposed method on various real images. They are chosen as they have the same input requirement as ours and they are completely software-based.

Comparison with Whyte *et al*. [27]. Four tested images are either from [27] or taken by ourselves using a Canon EOS 550D DSLR. See Fig. 6 for a visual comparison. The results using the Whyte *et al*.'s method are obtained as follows. The results for images from [27] are from the paper and the results for our own images are from their implementation provided in their website. From Fig. 6, it is seen that the results from our proposed approach show sharper edges and less artifacts.

Comparison with Gupta *et al*. [10] The implementation of method by Gupta *et al*. [10] is not available online, and its implementation complexity makes it hard for us to produce an implementation with the optimal performance. Thus, we only compared the results from [10] and our results of the images from [10]. It is seen from Fig. 7 for a visual comparison that the results from our proposed method are more visually pleasant with less artifacts.

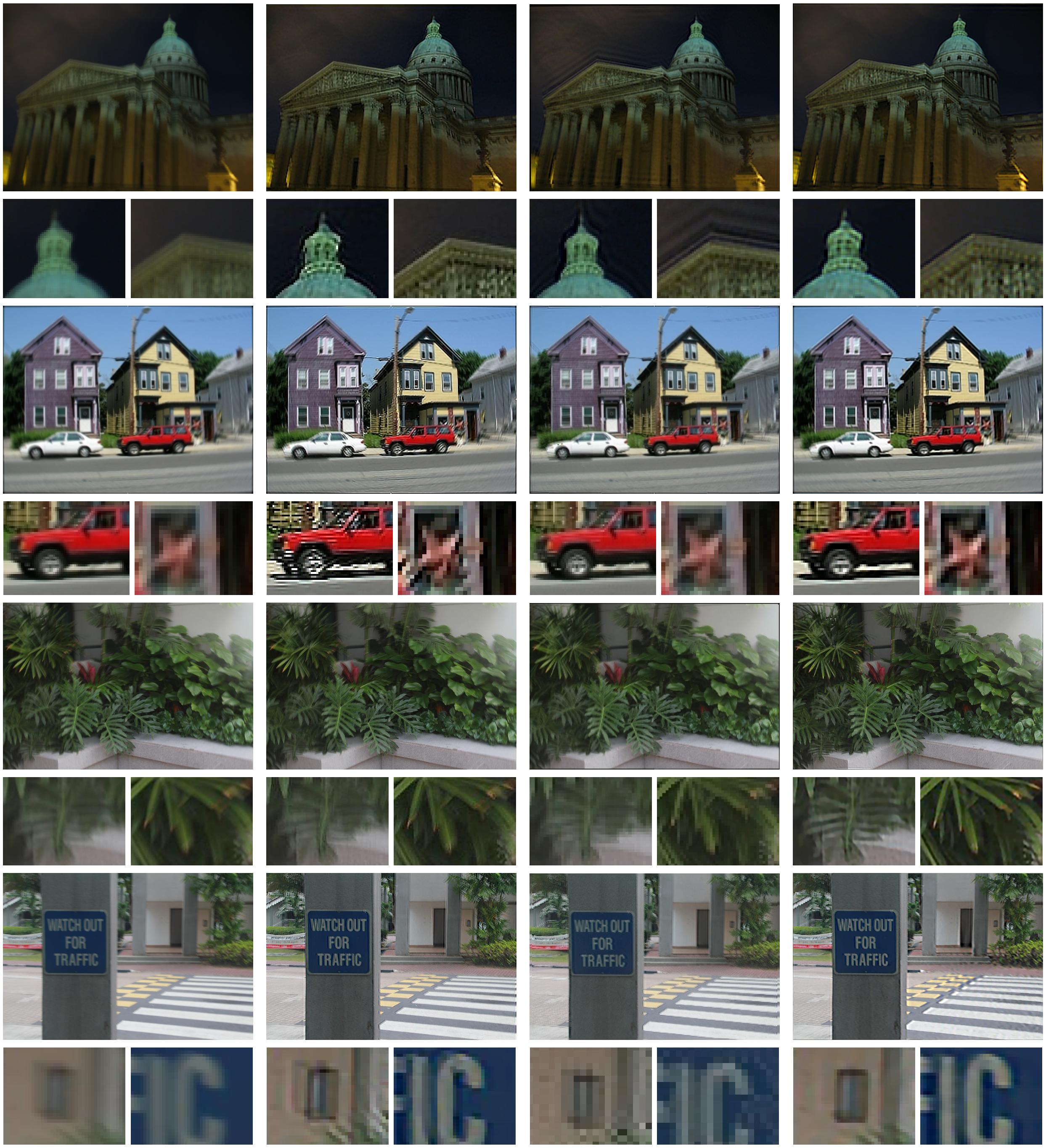
4.1. Conclusions

In this paper, we introduced a two-stage method to remove spatially varying blurring caused by camera shake from a single photograph. The proposed method is applicable to general camera motion and does not require any auxiliary hardware-based assistance. However, the proposed method is not applicable to the case of spatially varying blurring caused by moving objects. Another weakness of the method is that it requires fair initial kernel estimation on a certain percentage of image regions. In future, we would like to study how to improve its robustness to initialization and its applicability to a wider range of spatially varying blurred images.

References

[1] J.-F. Cai, R. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Appl. Comput. Harmon. Anal.*, 24:131–149, 2008. 3

[2] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, 2009. 1, 2, 3



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

Figure 6. Comparison with [27] on tested images. The full images are shown the odd rows and their corresponding zoomed image regions are shown in the blow even row. The images shown in the first and the third row of (a) are from [27] while the other two images are ours. (a) Blurry images; (b)-(d) Deblurred images by Xu *et al*. [28], Whyte *et al*. [27], and the proposed patch based framework, respectively.

[3] J.-F. Cai, S. Osher, and Z. Shen. Split bregman methods and frame based image restoration. *Multiscale Model. Simul.: A SIAM Int. J.*, 8(2):337–369, 2009. 2, 5, 6

[4] T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM J. Sci. Comput.*, 22:503–516, 2000. 2

[5] T. F. Chan and C. K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Process.*, 7(3):370–375, 1998. 2

[6] S. Dai and Y. Wu. Motion from blur. In *CVPR*, 2008. 2

[7] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets:

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

Figure 7. Comparison with [10] on tested images. The full images are shown the odd rows and their corresponding zoomed image regions are shown in the blow even row. The images shown in the first and the third row of (a) are from [10]. (a) Blurry images; (b)-(d): Deblurred images by Xu *et al*. [28], Gupta *et al*. [10], and the proposed patch based framework, respectively.

MRA-based constructions of wavelet frames. *Appl. Comput. Harmon. Anal.*, 14:1–46, 2003. 3

[8] 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. 1, 2, 3

[9] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009. 5

[10] A. Gupta, N. J. N., C. L. Zitnick, and M. Cohen. Single image deblurring using motion density functions. In *ECCV*, 2010. 2, 3, 6, 8

[11] M. Hirsch, C. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, 2011. 2

[12] M. Hirsch., S. Sra, B. Scholkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolutions. In *CVPR*, 2010. 2

[13] H. Ji, J. Li, Z. Shen, and K. Wang. Image deconvolution using a characterization of sharp images in wavelet domain. *Appl. Comput. Harmon. Anal.*, 32(2):295–304, 2012. 4

[14] H. Ji and K. Wang. Robust image deblurring with an inaccurate blur kernel. *IEEE Trans. Image Process.*, 21(4):1624 –1634, april 2012. 5

[15] N. Joshi, S. Kang, C. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (TOG)*, 29(4):1–9, 2010. 4

[16] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, 2008. 1, 2, 3

[17] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, 2011. 2, 4

[18] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2007. 1, 2, 3

[19] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. Technical report, MIT-CSAIL-TR-2009-014, 2009. 1

[20] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999. 3

[21] W. H. Richardson. Bayesian-based iterative method of image restoration. *J. of Optic. Soc. of Ame.*, 62, 1972. 2

[22] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *SIGGRAPH*, 2008. 1, 2, 3

[23] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007. 2

[24] Z. Shen. Wavelet frames and image restorations. In *Proceedings of the International Congress of Mathematicians*, volume 4, pages 2834–2863, 2010. 3, 5

[25] Y.-W. Tai, P. Tan, L. Gao, and M. S. Brown. Richardson-lucy deblurring for scenes under projective motion path. *IEEE PAMI*, to appear. 2

[26] A. N. Tikhonov, V. Y. Arsenin, and F. John. *Solution of ill-posed problems*. VH Washinton D.C., 1977. 2

[27] O. Whyte, J. S. J., A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, 2010. 1, 2, 3, 6, 7

[28] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010. 1, 2, 3, 6, 7, 8