Non-uniform Motion Deblurring for Bilayer Scenes

C. Paramanand and A. N. Rajagopalan

Department of Electrical Engineering, Indian Institute of Technology Madras

paramanand@gmail.com, raju@ee.iitm.ac.in

Abstract

We address the problem of estimating the latent image of a static bilayer scene (consisting of a foreground and a background at different depths) from motion blurred observations captured with a handheld camera. The camera motion is considered to be composed of in-plane rotations and translations. Since the blur at an image location depends both on camera motion and depth, deblurring becomes a difficult task. We initially propose a method to estimate the transformation spread function (TSF) corresponding to one of the depth layers. The estimated TSF (which reveals the camera motion during exposure) is used to segment the scene into the foreground and background layers and determine the relative depth value. The deblurred image of the scene is finally estimated within a regularization framework by accounting for blur variations due to camera motion as well as depth.

1. Introduction

A common problem faced while capturing photographs is the occurrence of motion blur due to camera shake. The extent of blurring at a point in an image varies according to the camera motion as well as the depth value resulting in space-variant blur. This further compounds the inverse and ill-posed deblurring problem. In this paper, our objective is to estimate the latent image of a static bilayer scene from its space-variantly blurred observations.

Traditionally, motion blur due to camera shake is modeled as the convolution of the latent image with a blur kernel. A lot of approaches for deblurring space-invariant blur exist in the literature [7, 16, 24]. In practical scenarios, blurring due to camera shake is space-variant in nature [14]. Recent techniques model the non-uniformly motion blurred image by considering the transformations undergone by the image plane rather than using a point spread function (PSF) [21, 23]. Tai et al. have proposed a deblurring scheme for the projective blur model based on Richardson Lucy deconvolution in [21]. However, they do not address the problem of determining the blurring function (the TSF). Whyte et al.

[23] propose an image restoration technique for motion blur arising due to non-uniform camera rotations. For the case of blind image restoration, the kernel estimation framework in [7] is employed. When a noisy version of the original image is available, a least-squares energy minimization approach is used for finding the blurring function. Another deblurring scheme has been proposed by Gupta et al. in [8]. The motion blurred image is modeled by considering the camera motion to be comprised of 2D translations and in-plane rotation. They propose an iterative scheme to recover the image and the blurring function. Other recent non-uniform deblurring works include [3, 9, 10, 11, 12, 13, 20].

It must be mentioned that none of the above methods can deal with changes in blur when there are depth variations. For the case of pure in-plane translations alone, [17] and [25] estimate the depth map and the restored image using two blurred observations.

In this paper, we develop a method to estimate the latent image of a bilayer scene using two motion blurred observations. The scene is assumed to consist of two dominant layers of depth, one corresponding to the foreground and the other to the background. There are many practical situations where we encounter this situation [19]. Multiple observations have been used in other restoration techniques to reduce ill-posedness [2, 4, 18, 22]. Such an approach has been followed for the scenario of non-uniform blur as well [3]. Two blurred observations of the same scene can be readily obtained when images are captured in the burst mode of a camera. We relate each of the two blurred observations with the original image of the scene through a TSF (which denotes its corresponding camera motion) and the depth of the scene. According to the blurring model, the observation is regarded as a weighted average of transformed instances of the original image. The weights denote the time spent by the camera at each pose and are referred to as the 'transformation spread function' (TSF) or equivalently, the 'motion density function' (MDF) [8].

We treat the camera motion to consist of 2D translations as well as in-plane rotations since this motion is more typical of camera shakes [8, 14]. Our motion model holds good in many scenarios due to the fact that the image stabiliz-

ers present in camera lenses have built-in gyro sensors that compensate for out-of-plane camera rotations, but not in-plane translations and rotations [1]. Also, camera translations along the optical axis is usually ignored because only large translations cause noticeable blur [8, 12]. When objects in the scene are near to the camera, the parallax effect can cause significant variations in blur [17]. Deblurring cannot be achieved unless the variation of blur due to depth is accounted for [12, 20]. Hence, knowledge of the depth of the scene is necessary for deblurring. We initially estimate the depth values of the two layers and subsequently solve for the latent image.

From the two blurred observations, we initially propose to determine the TSFs using blur kernels that are estimated at randomly chosen points across the image. We follow the multichannel blind deconvolution technique in [18] which can accurately determine blur kernels from two image patches. These blur kernels can be from either of the two depth layers. We derive a constraint to check whether any two given blur kernels of an observation are from the same depth layer. Based on this, we obtain a set of blur kernels corresponding to a depth layer. We relate the PSFs of one of the layers to the TSF through a linear equation and solve for the TSF by minimizing the least-squares error. The TSF is sparse in nature because during image capture, out of all possible transformations, the reference image would have undergone very few transformations. Hence, we also include a sparsity prior in our objective function. The estimated TSF corresponds to one of the depth layers of the scene which we regard as the reference (either foreground or background). From the blurred observations, and their reference TSFs we estimate the depth map which enables us to segment the scene into two layers and estimate the relative depth of the other layer with respect to the depth of the reference layer. The depth map is modeled as a Markov random field (MRF). We define a cost for assigning a depth label at a pixel. The maximum a posteriori (MAP) value of the depth label at each pixel is estimated using loopy belief propagation (BP) [6]. The original image is estimated by minimizing an energy functional with total variation (TV) regularization.

Contributions: This work presents significant contributions over recent works in motion deblurring: i) The performance of the state-of-the-art non-uniform deblurring works [23, 8, 10] is much superior than those that employ convolution model for motion blur. However, these methods ignore the effect of depth variations on blur. In contrast, our method adequately accounts for parallax effects for the case of bilayer scenes. ii) Techniques that do consider the effect of the scene depth [17, 25] restrict the camera motion to only in-plane translations whereas our model allows for in-plane rotations in addition to camera translations. iii) The TSF estimation techniques in [23, 8, 10] are not applicable

when there is parallax. Our approach of using blur kernels for TSF estimation enables us to address parallax effects for a bilayer scene. From a set of PSFs estimated at random points across the image, we develop a method to automatically select the blur kernels from a single depth layer. We arrive at the TSFs of the two observations that explains the local blur kernels of one layer.

2. Motion blur model

Initially, we consider the scene to be of constant depth and relate the blurred observation to the latent image in terms of a TSF. For a bilayer scene, we subsequently modify the model to take the parallax effect into account and relate the blurred image with the depth map and the TSF.

Let f denote the image of the scene captured by a still camera. Consider that all of the scene points are at a depth d_o along the camera's axis. Then the blurred image g can be modeled as a weighted average of warped instances of the original image f [23, 8]. i.e.,

$$g\left(\mathbf{x}\right) = \int_{\lambda \in \mathbf{T}} \omega\left(\lambda\right) f\left(H_\lambda\left(\mathbf{x}\right)\right) d\lambda \qquad (1)$$

where $\omega\ :\ \mathbf{T} \rightarrow \mathbb{R}_+$, is called the *transformation spread function* (TSF) and maps the set of all possible transformations $\mathbf{T}$ to non-negative real numbers. $H_\lambda\left(\mathbf{x}\right)$ denotes the image coordinates when a homography H_λ is applied on the point $\mathbf{x}$. The TSF depicts the camera motion. i.e., for each transformation $\lambda \in \mathbf{T}$, the value of the TSF $\omega\left(\lambda\right)$ denotes the fraction of the total exposure duration for which the camera stayed in the position that caused the transformation H_λ on the image coordinates.

The blurred image g can also modeled with a space-variant PSF h as [17]

$$g\left(\mathbf{x}\right) = f *_v h\left(\mathbf{x}\right) = \int f\left(\mathbf{x} - \mathbf{u}\right) h\left(\mathbf{x} - \mathbf{u}, \mathbf{u}\right) d\mathbf{u} \quad (2)$$

where $h\left(\mathbf{x}, \mathbf{u}\right)$ denotes the blur kernel at the image point $\mathbf{x}$ as a function of the independent variable $\mathbf{u}$. The PSF $h\left(\mathbf{x}, \mathbf{u}\right)$ represents the displacements undergone by a point light source at $\mathbf{x}$ during the exposure and is weighted according to the fraction of the exposure time the light source stays at the displaced position. It can be written in terms of the TSF as

$$h\left(\mathbf{x}, \mathbf{u}\right) = \int_{\lambda \in \mathbf{T}} \omega\left(\lambda\right) \delta\left(\mathbf{u} - \left(H_\lambda^{-1}\left(\mathbf{x}\right) - \mathbf{x}\right)\right) d\lambda \quad (3)$$

where δ indicates the 2D Dirac Delta function. Consider a point light source at $\mathbf{x}$ in the reference image f. Due to the transformation H_λ on the image co-ordinates, the point light source gets displaced by $\bar{\mathbf{x}}_\lambda = H_\lambda^{-1}\left(\mathbf{x}\right) - \mathbf{x}$. Substituting $h\left(\mathbf{x}, \mathbf{u}\right)$ from Eqn. (3) in Eqn. (2) gives us the blurring model in Eqn. (1). While the PSF h can vary at every image point, the TSF ω remains the same for whole image.

3. Bilayer scenes: the parallax effect

When there are depth variations in the scene, the transformation undergone by an image point is not only due to camera motion but also varies as a function of depth. Consequently, blurring cannot be modeled using Eqn. (1) with a single TSF. However, note that the camera motion is the same for all the image points. Hence, if we know the TSF for points at a specific value of depth, then, given the depth map of the scene, it should be possible to decipher TSF corresponding to any other image point. Furthermore, it is only the translation parameters (and not rotation) that are affected by depth changes.

When the camera intrinsics are fixed, the set of allowable transformations $\mathbf{T}$ is 6D due to the six degrees of freedom of camera motion. However, as discussed in section 1, in this work, we model the camera motion as translations along the image plane and in-plane rotations. The homography H_λ then simplifies to the following 2D transformation

$$H_\lambda(\mathbf{x}) = \begin{bmatrix} \cos\theta_\lambda & -\sin\theta_\lambda \\ \sin\theta_\lambda & \cos\theta_\lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_{x_\lambda} \\ t_{y_\lambda} \end{bmatrix}$$

where $[t_{x_\lambda}\ t_{y_\lambda}]^T$ and θ_λ represent the translation and rotation parameters of H_λ, respectively. The set of transformations $\mathbf{T}$ becomes a 3D space defined by axes t_x, t_y and θ.

Let ω_o denote the TSF corresponding to a depth of d_o. The TSF ω_o defines the weight for a transformation $\lambda \in \mathbf{T}$ (parameterized by t_{x_λ}, t_{y_λ} and θ_λ) for points at depth d_o. We express the transformation undergone by a point (i,j) (having depth $d(i,j)$) in terms of the *relative depth* $k(i,j) = \frac{d(i,j)}{d_o}$ and the parameters of the transformation λ. The rotation parameter θ_λ is not affected by depth changes. However, the translation parameters get scaled according to the depth and are given by

$$t_{x_{\lambda_{ij}}} = \frac{t_{x_\lambda}}{k(i,j)}, \ t_{y_{\lambda_{ij}}} = \frac{t_{y_\lambda}}{k(i,j)} \tag{4}$$

Let $H_{\lambda_{d(i,j)}}$ denote the transformation with the parameters $t_{x_{\lambda_{ij}}}$, $t_{y_{\lambda_{ij}}}$ and θ_λ. The PSF at a point $\mathbf{x} = (i,j)$ is

$$h(\mathbf{x},\mathbf{u}) = \int_{\lambda \in \mathbf{T}} \omega_o(\lambda)\,\delta\left(\mathbf{u} - \left(H_{\lambda_{d(i,j)}}^{-1}(\mathbf{x}) - \mathbf{x}\right)\right) d\lambda \tag{5}$$

Ignoring the effects of occlusion, the blurred image of the 3D scene can be related to the original image f through the space variant blurring operation (Eqn. (2)), wherein the PSF h depends on the camera motion (denoted by the TSF ω_o) and the depth map d (according to Eqn. (5)).

For bilayered scenes, the relative depth k takes only two distinct values. Thus, rather than evaluating PSF at every point, the blurred image can be obtained from the original using the TSF ω_o itself with the modifications in Eqn. (4).

Let $h(i,j,;)$ denote the discrete blur kernel at a pixel $\mathbf{p} = (i,j)$. For practical implementation, the set $\mathbf{T}$ is discretized to get N_T different transformations [23, 8]. For $l = 1\ldots N_T$, let $\omega_o(l)$ denote the weight corresponding to the lth transformation of $\mathbf{T}$ and (i_l, j_l) denote the coordinates of the point when a transformation $H_{l_{d(i,j)}}^{-1}$ is applied on $\mathbf{p}$ (translations scaled according to Eqn. (4)). The discrete form of the relationship in Eqn. (5) is given by

$$h(i,j;s,t) = \sum_{l=1}^{N_T} \omega_o(l)\,\delta_d\left(s - (i_l - i), t - (j_l - j)\right) \tag{6}$$

where δ_d denotes the 2D Kronecker delta function. When (i_l, j_l) take non-integer values, we assign values to the pixels neighboring (i_l, j_l) by bilinear interpolation principle. From the discrete PSF, we can write the blurred image in terms of the latent image f as

$$g(i,j) = \sum_{s,t} f(i-s, j-t)\,h(i-s, j-t; s,t) \tag{7}$$

4. Image deblurring

Consider two blurred observations g_1 and g_2 of a bilayer scene which are related to the original image f through the TSFs ω_{1o} and ω_{2o}, respectively, and the depth map d. The objective is to estimate f from g_1 and g_2. We devise a method which uses blur kernels estimated at different points in the image in order to determine the TSFs. Based on the estimated TSFs ω_{1o} and ω_{2o}, and the blurred observations, we determine the relative depth map k. Finally, we solve for the original image f. The details of these steps are explained in the following subsections.

4.1. Reference TSF estimation

Our initial step is to estimate blur kernels at different image points. Following [24], we determine a subset of image points which are suitable for blur kernel estimation (regions with wide edges). Within this subset, $N_\mathbf{s}$ point locations are selected at random, and blurred image patches from both the observations (denoted by $g_1^1\,g_1^2\ldots g_1^{N_\mathbf{s}}$ and $g_2^1\,g_2^2\ldots g_2^{N_\mathbf{s}}$) are cropped around these points. Within each pair of patches (g_1^i and g_2^i), we assume the blur to be space-invariant and use the blind deconvolution technique in [18] (which uses two blurred observations) to get the blur kernels h_1^i and h_2^i. We found in our experiments that the estimates of blur kernels from the method in [18] are quite accurate.

4.1.1 Kernel segmentation

The blur kernels $h_b^1\,h_b^2\ldots h_b^{N_\mathbf{s}}$ can be either from the foreground or the background layers (for $b = 1$ and 2). Out of the $N_\mathbf{s}$ kernels of an observation, given any two PSFs, we

compare the possible transformations of the kernels and determine whether the two kernels belong to the same depth layer. Thence, we segment the N_s kernels into two groups depending on the layer to which they belong. The TSF is determined for the layer that has more number of PSFs.

Consider a point $\mathbf{p_1} = (i, j)$. We refer to a set of transformations Λ_b^1 as the support of the blur kernel h_b^1. The set Λ_b^1 is a subset of $\mathbf{T}$ and is defined as $\Lambda_b^1 = \left\{ \lambda : h_b^1\left(i, j; i_\lambda - i, j_\lambda - j\right) > 0 \right\}$, where (i_λ, j_λ) denotes the co-ordinates of the point when H_λ^{-1} is applied on $\mathbf{p_1}$. In other words, Λ_b^1 contains all possible transformations from $\mathbf{T}$ that shift the point $\mathbf{p_1}$ to a position at which the blur kernel h_b^1 has a positive entry. Since $\mathbf{T}$ contains in-plane rotations as well as translations, many transformations can cause the same displacement. Consequently, Λ_b^1 will be a much larger set than the number of transformation present in the true TSF. However, the cardinality of Λ_b^1 is limited because a motion blur kernel is sparse.

Given two blur kernels, h_b^1 and h_b^2 corresponding to locations $\mathbf{p_1}$ and $\mathbf{p_2}$, respectively, we determine the supports of the blur kernels Λ_b^1 and Λ_b^2. If $\mathbf{p_1}$ and $\mathbf{p_2}$ were at the same depth layer, the blur kernels h_b^1 and h_b^2 would be related to a single TSF, and there would be a common set of transformations between Λ_b^1 and Λ_b^2 that would include the support of the TSF. In contrast, if $\mathbf{p_1}$ and $\mathbf{p_2}$ were at different layers, the transformations in Λ_b^1 and Λ_b^2 would be different. We evaluate the intersection of the two supports $\Lambda_b^{12} = \Lambda_b^1 \cap \Lambda_b^2$. We apply the transformations in Λ_b^{12} on $\mathbf{p_1}$ and $\mathbf{p_2}$ to get two sets of displacements $\hat{h}_b^1$ and $\hat{h}_b^2$, respectively. If h_b^1 has positive entries at locations other than those in $\hat{h}_b^1$, or h_b^2 has positive entries at locations other than those in $\hat{h}_b^2$, then we can conclude that there are no common transformations between Λ_b^1 and Λ_b^2 that can generate both the blur kernels. This would imply that $\mathbf{p_1}$ and $\mathbf{p_2}$ cannot be at the same depth. It must be noted that, only when the effect of parallax is significant, the transformations in the supports of the two blur kernels would be different.

4.1.2 TSF from PSFs

Using our method, out of N_s blur kernels, we select $N_\mathbf{p}$ kernels that are at the same depth. Our objective is to estimate the TSF ω_{bo} that concurs with the $N_\mathbf{p}$ observed blur kernels $h_b^1 \ h_b^2 \ldots h_b^{N_\mathbf{P}}$ and their locations for $b = 1$ and 2. The TSF defined on the discrete transformation space $\mathbf{T}$ can be considered as a vector in $\mathbb{R}^{N_T}$ where N_T is the total number of transformations present in $\mathbf{T}$. In practice, the TSF $\omega_{bo} \in \mathbf{R}^{N_T}$ will be a sparse vector because the camera motion during exposure would result in very few transformations out of all possible elements of $\mathbf{T}$. Hence, while solving for ω_{bo} we impose a sparsity constraint for regularization. In Eqn. (6), we see that at a pixel (i, j), each component of the blur kernel $h(i, j; m, n)$, is a weighted sum of

the components of the TSF ω. Consequently, the blur kernel h_b^i can be expressed as $h_b^i = M_b^i \omega_{bo}$ for $i = 1 \ldots N_\mathbf{p}$, where M_b^i is a matrix whose entries are determined by the location of the blur kernel and the bilinear interpolation coefficients. If the number of elements in the blur kernel is N_h, then the size of the matrix M_b^i will be $N_h \times N_T$. By stacking all the $N_\mathbf{p}$ blur kernels as a vector $\overline{h}_b$, and suitably concatenating the matrices M_b^i for $i = 1 \ldots N_\mathbf{p}$, the blur kernels can be related to the TSF as

$$\overline{h}_b = M_b \omega_{bo} \qquad (8)$$

The matrix M_b is of size $N_\mathbf{p} N_h \times N_T$. To get an estimate of the TSF that is consistent with the observed blur kernels and is sparse as well, we minimize the following cost (separately for $b = 1$ and 2) using the toolbox available at [15].

$$\underset{\omega_{bo}}{\operatorname{argmin}} \left\{ \left\| \overline{h}_b - M_b \omega_{bo} \right\|_2^2 + \lambda_s \left\| \omega_{bo} \right\|_1 \right\} \qquad (9)$$

4.1.3 Alignment of PSFs

The blurred image patch g_b^i which is given by $g_b^i = f^i * h_b^i$ would also be equal to $\mathbf{S}_i^{-1}\left(f^i\right) * \mathbf{S}_i\left(h_b^i\right)$ where $*$ denotes convolution and the term $\mathbf{S}_i\left(\cdot\right)$ denotes a translation shift. Consequently, while solving for the local PSFs (h_b^i), there can be incidental shifts of small magnitude in the estimated blur kernels with respect to the 'true' blur kernels (which are induced at a point as a result of blurring the latent image with the true TSF). Since the translation for each patch may differ, the estimation of ω_{bo} will not be possible from the kernels unless the shifts in the PSFs are compensated. The shift remains the same for both $b = 1$ and 2. Hence, we need to determine the shifts of blur kernels corresponding to only one of the two observations. The TSF ω_{1o} estimated from the aligned blur kernels should have a low value of the error $\left\| \overline{h}_1 - M_1 \omega_{1o} \right\|_2^2$. We consider that one of the blur kernels, say h_1^1 does not undergo any shift and align the other kernels with respect to this. We need to determine two translation parameters for each of the other blur kernels. For all possible combination of the translations, we shift the blur kernels $h_1^2, h_1^3, \ldots h_1^{N_\mathbf{P}}$, and evaluate the solution of the Eqn. (9). Since the magnitude of the shifts are generally small, and the number of blur kernels used is typically low (around 4), finding the optimum shifts (that minimize the error $\left\| \overline{h}_1 - M_1 \omega_{1o} \right\|_2^2$) is computationally quite feasible.

In the case of convolution, a set of blur kernels (which are translated versions of each other) and correspondingly shifted latent images can lead to the same blurred observation. In the scenario of TSF model, there can be multiple solutions for the latent image and the TSF such that the images are rotated and translated versions of one-another. The alignment step provides one possible solution to the shifts of the blur kernels that leads to a TSF.

Even in the kernel segmentation step (section 4.1.1), the shift in the estimated blur kernels needs to be accounted for. Given two blur kernel h_b^1 and h_b^2, we verify the condition discussed in section 4.1.1 for all possible shifted versions of one of the blur kernels.

4.2. Depth estimation and segmentation

We determine the relative depth $k(i,j)$ at every pixel from the blurred observations g_1 and g_2, and the TSFs ω_{1o} and ω_{2o} through a MAP-MRF framework. We segment this depth map into two levels corresponding to the foreground and background. We follow the efficient implementation of the loopy belief propagation (BP) algorithm proposed in [6] to obtain the MAP estimate.

The blur kernel $h_b(i,j,;)$ at a point (i,j) is given by

$$h_b(i,j;m,n) = \sum_{l=1}^{N_T} \omega_{bo}(l)\,\delta_d\left(m - (i_l - i), n - (j_l - j)\right) \tag{10}$$

where (i_l, j_l) denotes the co-ordinates of the point when a transformation H_l^{-1} is applied on (i,j) by accounting for the parallax effect (Eqn. (4)). Then, the blurred image intensity at a point (i,j) can be approximately written as

$$g_b(i,j) = \sum_{m,n} f(i - m, j - n)\,h_b(i,j;m,n) \text{ for } b = 1, 2$$

Consider the blur kernel $h_c(i,j;)$ which is defined as the convolution of $h_1(i,j;)$ and $h_2(i,j;)$. i.e., $h_c(i,j;) = h_1(i,j;) * h_2(i,j;)$. Then, using commutativity

$$\sum_{m,n} f(i - m, j - n)\,h_c(i,j;m,n)$$
$$= \sum_{m,n} g_1(i - m, j - n)\,h_2(i,j;m,n)$$
$$= \sum_{m,n} g_2(i - m, j - n)\,h_1(i,j;m,n) \tag{11}$$

Let $h_{b\mathbf{k}}(i,j;m,n)$ denote the blur kernel generated from the reference TSF ω_{bo} by assuming the relative depth at (i,j) to be $\mathbf{k}$ (according to Eqn. (10)). Based on the relationship in Eqn. (11), we define the data cost of assigning a label $\mathbf{k}$ at a pixel (i,j) as

$$D_{(i,j)}(\mathbf{k}) = \left| \sum_{m,n} g_1(i - m, j - n)\,h_{2\mathbf{k}}(i,j;m,n) \right.$$
$$\left. - \sum_{m,n} g_2(i - m, j - n)\,h_{1\mathbf{k}}(i,j;m,n) \right| \tag{12}$$

Note that although space-variant blurring is not commutative in general, the data cost in Eqn. (12) gets minimized only for the correct depth label at a pixel. For other depth labels, the difference between the two terms on the RHS of Eqn. (12) would be significantly large due to incorrect scaling of the blur kernels $h_{1\mathbf{k}}(i,j;)$ and $h_{2\mathbf{k}}(i,j;)$. A similar cost has been used to get an initial estimate of the depth map from blurred images in [17] for the case of translational blur.

In order to ensure smoothness, the smoothness cost is defined to penalize the difference in the labels of neighboring pixels. To allow for discontinuities, the cost function should take a constant value when the difference becomes large. A commonly used prior is the truncated linear model [6] which is defined as

$$V(\mathbf{k}_p, \mathbf{k}_q) = \min\left(\mu_\mathrm{p}\left|\mathbf{k}_p - \mathbf{k}_q\right|, \lambda_\mathrm{th}\right) \tag{13}$$

where μ_p is a weighting parameter and the threshold λ_th determines when the cost stops increasing.

Based on the data cost and the smoothness cost, we obtain the MAP estimate of the relative depth through the message passing scheme [6]. On the relative depth map, we apply segmentation using the technique in [5] to obtain N_d distinct values. The algorithm in [5] captures perceptually important non-local properties. This helps us in arriving at a reasonably accurate segmentation despite isolated errors in the estimated depth map. We further partition the segmented depth map which contains N_d different depth layers into two segments using k-means clustering. Thus, we arrive at a relative depth map with two distinct depth values.

4.3. Restoration

Once the TSFs ω_{1o} and ω_{2o} and the relative depth k are known, to estimate f, we formulate a regularized energy functional based on the observation error as

$$E(f) = \frac{1}{2}\sum_{b=1}^{2}\|f *_v h_b - g_b\|^2 + \lambda\int|\nabla f| \tag{14}$$

where $*_v$ denotes the space-variant blurring operation of Eqn. (2) and $\int|\nabla f|$ is the total variation norm. We follow the half-quadratic scheme to enforce the total variation prior [18]. For minimization of the cost in Eqn. (14), similar to the approach in [18], we follow preconditioned conjugate gradient method. The gradient of the energy functional is

$$\partial E(f) = \sum_{b=1}^{2}(f *_v h_b - g_b)\circledast_v h_b - \lambda\mathrm{div}\left(\frac{\nabla f}{|\nabla f|}\right) \tag{15}$$

where $\circledast_v$ denotes the adjoint of $*_v$ and is defined [17] as

$$f\circledast_v h(\mathbf{x}) = \int f(\mathbf{x} - \mathbf{u})\,h(\mathbf{x}, -\mathbf{u})\,d\mathbf{u} \tag{16}$$

Substituting the expression for $h(\mathbf{x}, \mathbf{u})$ from Eqn. (5) in Eqn. (16), we get the result of the adjoint operation $\tilde{g}_b$ as

$$\tilde{g}_b(\mathbf{x}) = \int_{\lambda\in\mathbf{T}} \omega_{bo}(\lambda)\,f\left(H_{\lambda_{d(i,j)}}^{-1}(\mathbf{x})\right)d\lambda \tag{17}$$

i.e., the adjoint operation is equivalent to blurring through a TSF except for the fact that the warping to be applied is $H_{\lambda_{d(i,j)}}^{-1}$ instead of $H_{\lambda_{d(i,j)}}$.

Performing the operations $*_v$ and $\circledast_v$ through the space-variant PSFs h_1 and h_2 would be computationally very intensive as compared to blurring through TSFs. Hence, we segment the depth map to two layers so that within each depth layer, the TSF model can be used.

5. Experimental results

We performed synthetic as well as real experiments to test the proposed deblurring algorithm. In the real experiments, the objects in the scene were within a range of 1.5 metres so that the variation of blur with respect to depth due to camera translations is perceivable. The camera used was Canon EOS 60D. The lens used in our camera EF-S 18-135 mm has the option of image stabilization which reduces the effect of out-of-plane camera rotations [1]. We captured two blurred observations of the scene in burst mode with image stabilization enabled. In all our experiments, the exposure time was around one second. Due to camera shake in this duration, there was appreciable blurring in the observations.

In our synthetic experiment, the latent image and the scene depth were assumed to be as shown in Figs. 1 (a) and (e), respectively. We considered two different realistic TSFs to generate the observations. From the depth map and the TSFs, we calculated the two PSFs at every pixel according to Eqns. (4) and (5). The observations generated using the space-variant blur model of Eqn. (7) are shown in Figs. 1 (b) and (c). For deblurring, blur kernels were estimated using image patches that were randomly selected across the image (marked in Figs. 1 (b) and (c)). Out of six blur kernels of an observation, we found that two of the blur kernels were from one of the layers and the remaining four kernels were from the other layer (please refer to the supplementary material). The layer with the four blur kernels was chosen as the reference. The TSFs of the two observations were determined from the corresponding blur kernels after alignment step. To verify our estimate of the TSFs, we generated blur kernels at the image points from where the patches were cropped and found that they were close to the true blur kernels. We show the blur kernels of the patches in the supplementary material.

Based on the estimated reference TSFs, we arrived at the depth map as shown in Fig. 1 (f) by applying the proposed BP-based method. The segmented depth map shown in Fig. 1 (g) is similar to the true depth map of Fig. 1 (e). We then applied our TV regularization-based method to arrive at the latent image of the scene from the estimated reference TSFs and the segmented depth map of Fig. 1 (g). The deblurred image shown in Fig. 1 (d), is quite sharp and is close to the original image of Fig. 1 (a). The third and fourth rows of Fig. 1 show three sets of zoomed-in patches. In each set,

the first patch is from the original, the next two are from the observations, and the fourth patch is from the restored image. In the fourth row, the last patch shows the result of deblurring when depth variations were ignored. Note that the original image has very sharp intensity variations which is hard to recover. Our method (fourth patch) is able to restore the image quite well even in the presence of parallax.

We show the performance of our method on two real examples (Fig. 2). While the first scene consisted of two wallpapers at different depths, the second scene was of a person standing against a background. The blurred observations are shown in the first and second columns of first and third rows. The result of our bilayer scene deblurring technique is shown in the third column. The zoomed-in patches of the observations and the deblurred image are shown below their respective images (rows 2 and 4). The patches indicate that even though the extent of blurring is significantly large, our algorithm estimates the latent image quite well. While the patches in rows 2 and 4 were from the reference depth layer, the fifth row shows patches from the other layer. In each set of patches of the fifth row, the first two patches are from the observations and the third patch is from the proposed deblurring algorithm. The result of deblurring when parallax was ignored is shown in the fourth patch, wherein we see that deblurring is far from perfect. This serves to highlight the importance of accounting for depth changes. In our results, we observe that at the object boundaries, there could be some isolated artifacts due to incorrect depth estimates. The blur kernels and the depth maps of the real experiments are shown in the supplementary material.

All the image files corresponding to our dataset and the outputs are included in the supplementary material. For purpose of comparison, we applied the single image deblurring techniques in [10] and [23] on the images of our dataset. We found that the results are not satisfactory as shown in the sixth row of Fig. 2. The zoomed-in regions are included in the supplementary material. This could be due to fact that in these methods a single image was used and parallax effect was not accounted for. It must be mentioned that this is not a fair comparison because the algorithms in [10, 23] take only one observation as input.

6. Conclusions

We addressed the problem of restoring a bilayered scene when the captured observations are space-variantly motion blurred due to incidental camera shake. Through the notion of TSF, we were able to effectively model blurring due to camera motion. For estimating the TSF corresponding to either the foreground or background, we proposed to use local blur kernels. We developed a method to automatically group the blur kernels corresponding to their depth layers. This enables us to estimate a sparse TSF that is consistent with the observed blur kernels of a depth layer. Based on
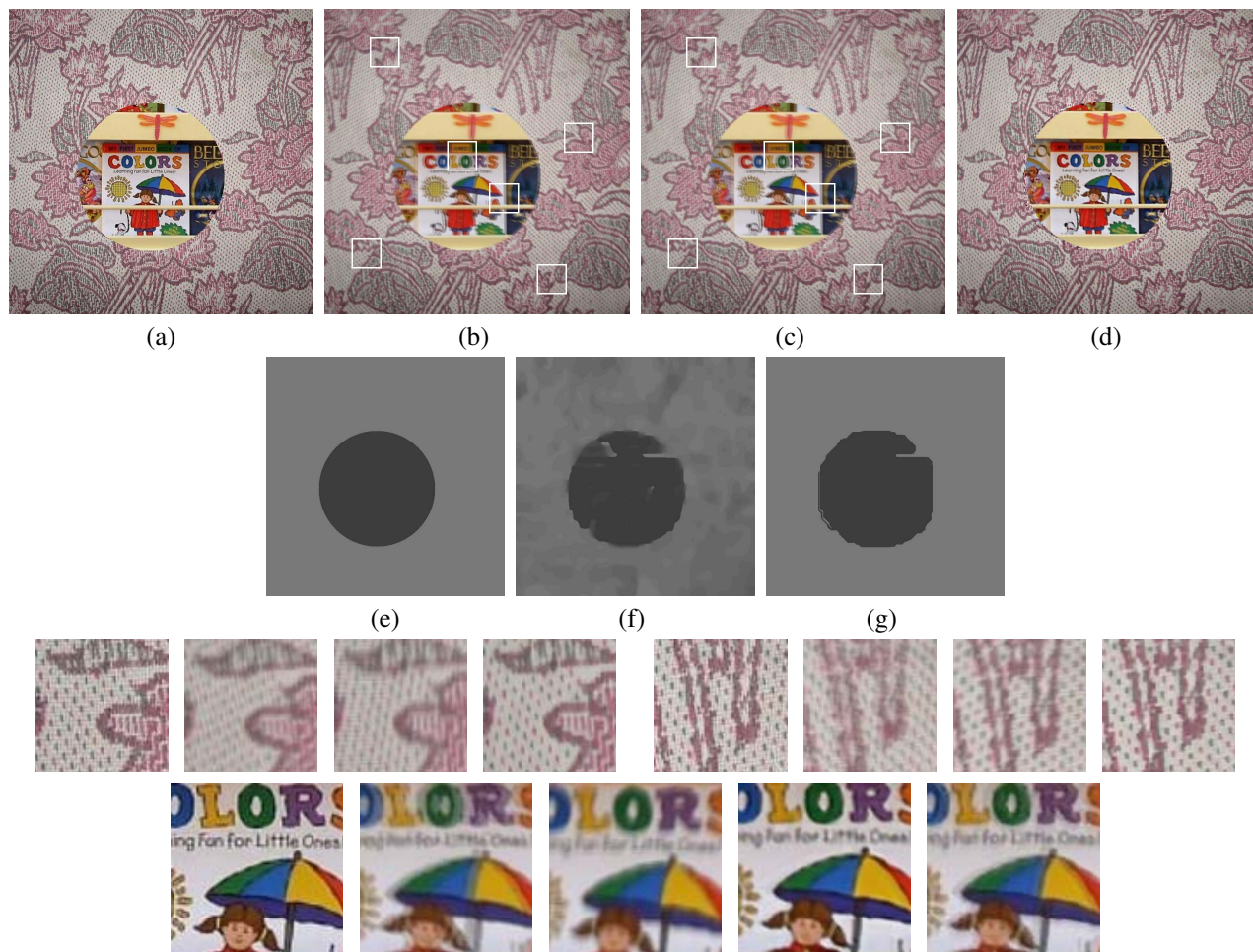
Figure 1. (a) Original image. (b) and (c) Blurred observations. (d) Restored image. (e) True depth map. (f) Estimated depth map. (g) Segmented depth map. Third and fourth rows: zoomed-in regions from the original image, blurred observations, and restored image (in the three sets of patches). Fourth row, last image: result of deblurring when depth changes were ignored.

the reference TSFs and the multilayered depth map, we finally arrived at the deblurred image within a regularization framework. When tested on different synthetic and real experiments, our results reveal that the proposed non-uniform motion deblurring scheme is quite effective in accounting for parallax effects in bilayered scenes. Our method can be extended to scenes with multiple depth layers.

References

[1] http://www.usa.canon.com/cusa/consumer/standard_display/Lens_Advantage_IS.

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

[3] S. Cho, H. Cho, Y. Tai, and S. Lee. Non-uniform motion deblurring for camera shakes using image registration. In *SIGGRAPH Talks*, 2011.

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

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Intl. Jrnl. Comp. Vis.*, 2004.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Intl. Jrnl. Comp. Vis.*, 2006.

[7] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794, 2006.

[8] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proc. ECCV*, 2010.

[9] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *Proc. ICCV*, 2011.

[10] Z. Hu and M. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *Proc. BMVC*, 2012.

[11] H. Ji and K. Wang. A two-stage approach to blind spatially-varying motion deblurring. In *Proc. CVPR*, 2012.

[12] N. Joshi, S. B. Kang, L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *Proc. SIGGRAPH*, 2010.

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

[14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. CVPR*, 2009.

[15] J. Liu, S. Ji, and J. Ye. Slep: Sparse learning with efficient projections. 2009. http://www.public.asu.edu/ jye02/Software/SLEP.

[16] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3).
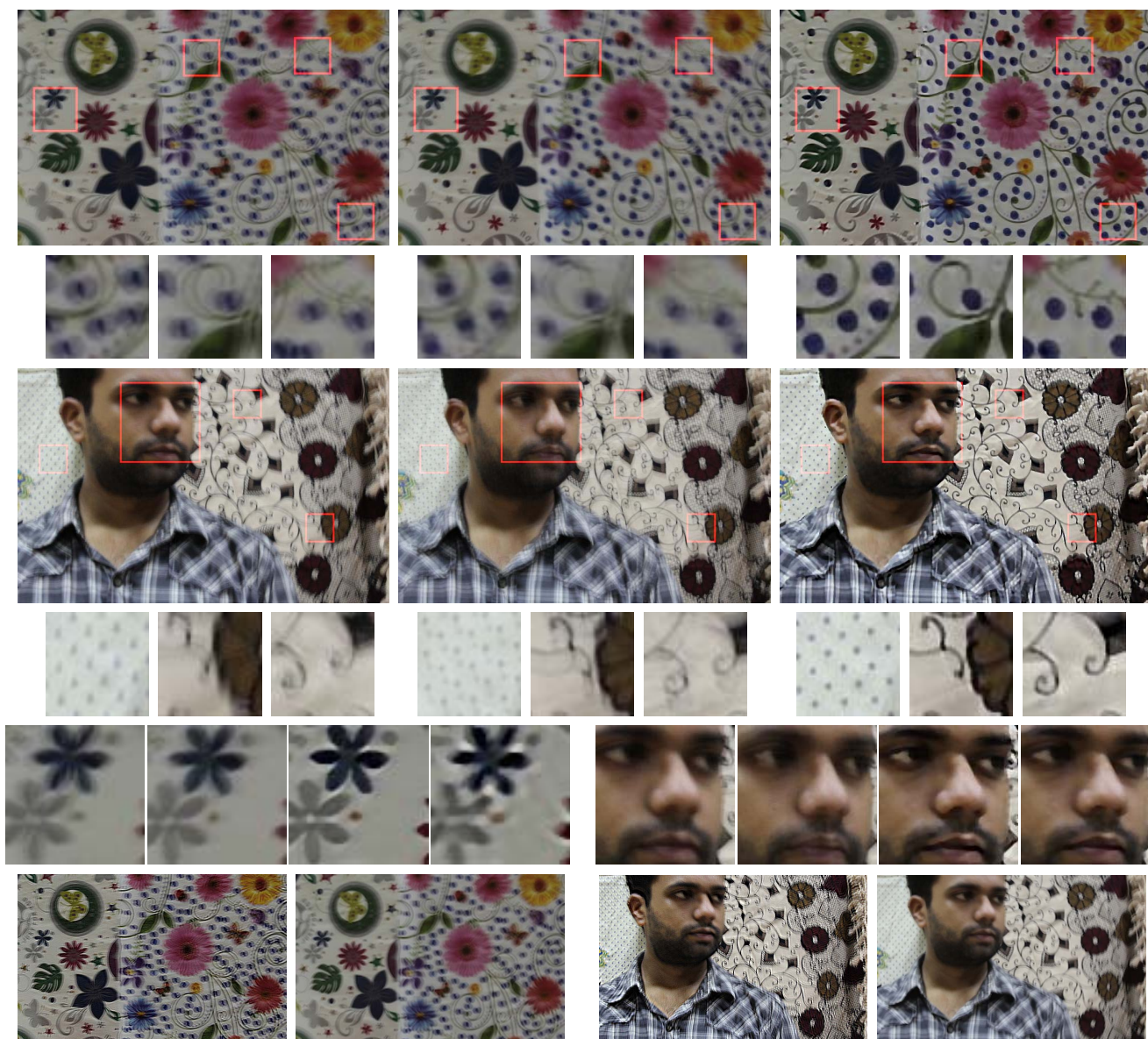
Figure 2. The blurred observations are shown in rows 1 and 3, first two columns. The estimated latent image is shown in the last column (rows 1 and 3). Rows 2 and 4: zoomed-in patches of the reference depth layer. Row 5: zoomed-in regions from the blurred observations, restored image and deblurred image when parallax was ignored (in each of the two sets of patches). Row 6: First and third images- output from [10], second and fourth images- output of [23].

[17] M. Sorel and J. Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Trans. Imag. Proc.*, 17(2):105–116, 2008.

[18] F. Sroubek and J. Flusser. Multichannel blind deconvolution of spatially misaligned images. *IEEE Trans. Imag. Proc.*, 14(7):874–883, 2005.

[19] A. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *Proc. ICCV*, 2007.

[20] Y. Tai, N. Kong, S. Lin, and S. Y. Shin. Coded exposure imaging for projective motion deblurring. In *Proc. CVPR*, 2010.

[21] Y. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under projective motion path. *IEEE Trans. Patt. Anal. Mach. Intell.*, 33(8):1603–1618, 2011.

[22] R. Vio, J. Nagy, and W. Wamsteker. Blind motion deblurring using multiple images. *Jrnl Comput. Phy.*, 228(14):5057–5071, 2009.

[23] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proc. CVPR*, 2010.

[24] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proc. ECCV*, 2010.

[25] L. Xu and J. Jia. Depth-aware motion deblurring. In *Proc. ICCP*, 2012.