Forward Motion Deblurring

Shicheng Zheng Li Xu Jiaya Jia
The Chinese University of Hong Kong
http://www.cse.cuhk.edu.hk/leojia/projects/forwarddeblur/

Abstract

We handle a special type of motion blur considering that cameras move primarily forward or backward. Solving this type of blur is of unique practical importance since nearly all car, traffic and bike-mounted cameras follow out-of-plane translational motion. We start with the study of geometric models and analyze the difficulty of existing methods to deal with them. We also propose a solution accounting for depth variation. Homographies associated with different 3D planes are considered and solved for in an optimization framework. Our method is verified on several natural image examples that cannot be satisfyingly dealt with by previous methods.

Figure 1. Out-of-plane translational motion blur examples.

1. Introduction

Motion blur is one ubiquitous problem in photo taking. Previous deblurring approaches model the degradation in different ways. For example, it is common to assume uniform blur with only in-plane translation or take into account camera rotation. While prior models are effective on images produced under their respectively defined conditions, there are still a bunch of blurred images that find no solution in restoration using existing techniques. Motion blur caused by out-of-plane translation falls into this set.

There are millions of, or even more, images that can be found easily on internet degenerated by forward or backward motion during image capture. It is because out-of-plane translation represents one dominating type of camera motion in many commonly seen scenarios. For instance, car-mounted cameras, which are getting very popular in recent years, produce a load of images and videos. Most surveillance cameras placed on highway record moving vehicles, which could also produce this type of motion blur. In addition, out-of-plane translational motion is common from wearable sport cameras and other smart devices, such as Google glasses. A few examples are shown in Fig. 1.

Obviously, if these images can be deblurred correctly, not only image visual quality is much improved, but also many practical applications can be immediately benefitted, which include fast-moving car license recognition and face/object detection in egocentric vision. We present the fact that all existing methods, unfortunately, cannot solve this problem in their respective models. Our study also shows out-of-plane is not a simple extension of prior models, but instead a new one with unique physical properties.

In this paper, we focus on dealing with images blurred mainly by *vehicle movement* or alike, which makes it possible to safely ignore severe camera rotation for algorithmic tractability. Our objective is further narrowed down to deblurring 3D-planes with the understanding of general geometric models in image formation and its present limitation. The reason to only consider 3D-planes is twofold. On the one hand, the majority of useful information alongside highway, such as characters and illustrations, is generally shown on planer boards. Human face and car license can also be regarded as planar regions from afar. On the other hand, 3D planes introduce notable computation flexibility.

Note our 3D-plane configuration has counted in depth variation. Each plane is modeled by its normal, which is initialized and automatically updates in the proposed method. We further explore parametric blur solution space and therefore handle photos taken from rapidly moving vehicles.

2. Related Work

Image deblurring finds many previous methods. State-of-the-arts are roughly categorized to spatially-invariant and spatially-variant configurations, based on different assumptions of the underlying blur model.

2.1. Uniform Blind Deconvolution

In uniform deblurring, Fergus et al. [2] proposed a blind deconvolution scheme applicable to natural images. It employs a variational Bayesian approach to estimate the blur kernel through marginal probability maximization. Efficient marginalization was later introduced in [15]. Another line of work is by extending the MAP framework to estimate latent images and blur kernels iteratively. Shan et al. [18] used adaptive regularization weights to avoid trivial solutions. Cho and Lee [1] predicted edges to guide kernel optimization, which shortens running time in kernel estimation. Levin et al. [14] analyzed the MAP framework with regard to its limitation and extensibility. Xu and Jia [24] found small structures in images could be detrimental to kernel estimation and proposed a method to remedy this problem. Krishnan et al. [13] used normalized sparsity in their MAP framework to estimate kernels. Xu et al. [26] sought an unnatural representation for blur kernel estimation and proposed a fast solver for restoration.

2.2. Non-Uniform Blind Deblurring

The fact that blur caused by camera shake in images are usually non-uniform motivates a series of work with method generalization to model spatially variant blur. Shan et al. [19] first addressed blur caused by in-plane rotation. Tai et al. [21] used a projective model to handle spatially variant blur. Whyte et al. [23, 22] used 3D rotation to model camera shake. In [4], the motion density function was introduced. In-plane translation and orthogonal rotation are used to model camera shake in another way. Hirsch et al. [6] assumed that blur is locally invariant and proposed a fast non-uniform framework based on efficient filter flow [7]. Joshi et al. [8] developed a hardware solution to record camera shake and restore blurred images. Xu and Jia [25] used stereo images and incorporated depth into the deblurring framework. In addition, they proposed a hierarchical framework for non-uniform deblurring.

2.3. Other Work

Optical aberration can be regarded as a special case of non-uniform blur caused by imperfection of lens. Joshi et al. [9] estimated PSFs with calibration sheets and recovered sharp images. In [10, 16], optical blur was estimated for the lens. Schuler et al. [17] designed a set of bases to describe optical blur and proposed a blind deconvolution method to address optical aberration.

3. Background

Image capture is the process that each element with coordinate $\mathbf{x}$ in the camera sensor receives light from the scene $\mathbf{X}$. Under homogeneous coordinates, $\mathbf{x}$ is denoted as $(x, y, 1)^T$ and $\mathbf{X} = (x, y, z, 1)^T$, where T is the transpose operator. Image blur typically stems from two sources during exposure, i.e., camera shake and object motion. Both types make the sensor element $\mathbf{x}$ receive light from a series of scene points during exposure.

In modeling the geometric formation process of camera-caused motion blur, nearly all prior methods assume constant scene depth z under the condition that the scene is distant or front-parallel. Xu and Jia [25] took depth into consideration; but they need stereo images to tackle the depth problem, which may not be applicable to single image deblurring. Under the constant z assumption, blur image capture can be regarded as a sharp image $\mathbf{l}$, which is formed in a very short time interval, moves in a longer duration. The output image is the weighted sum of all these transformed sharp images expressed as [23]

$$\mathbf{b} = \sum_i w_i \mathbf{P}_i \mathbf{l} + \varepsilon, \tag{1}$$

where $\mathbf{b}$ denotes the blurred image and i indexes the transformed sharp image. All images are in the vector form in this paper. $\mathbf{P}_i$ is a $N \times N$ coefficient matrix that transforms the latent sharp image $\mathbf{l}$ to $\mathbf{l}_i$ in status i, – that is, $\mathbf{P}_i\mathbf{l} = \mathbf{l}_i$. ε is additive noise. w_i is a weight corresponding to the duration that the latent image stays in status i.

Actually, $\mathbf{l}_i$ and $\mathbf{l}$, which are images captured with two different poses, are related by a projection matrix according to two-view geometry [5]. Then any pixel $\mathbf{x}$ in $\mathbf{l}$ relates to the corresponding $\mathbf{x}'$ in another view as

$$\mathbf{x}' = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{x} + \mathbf{K}\mathbf{t}/z = \mathbf{H}\mathbf{x}, \tag{2}$$

where $\mathbf{R}$ and $\mathbf{t}$ denote the pose of camera with a total of six degrees of freedom. $\mathbf{R}$ represents rotation around the x, y, and z axes with angles θ_x, θ_y, and θ_z respectively; $\mathbf{t}$ is translation in three directions. $\mathbf{K}$ is the camera intrinsic matrix, which has focal length f measured in pixels and image center (c_x, c_y) as parameters. During one shot, $\mathbf{K}$ is fixed due to generally constant focal length.

Eq. (2) indicates that a view can be modeled as a homography warping of the latent sharp image $\mathbf{l}$. By assuming small rotation angles as that in [23], the transformation matrix $\mathbf{H}$ is written as

$$\mathbf{H} = \mathbf{K} \begin{pmatrix} 1 & -\theta_z & \theta_y + t_x/z \\ \theta_z & 1 & -\theta_x + t_y/z \\ -\theta_y & \theta_x & 1 + t_z/z \end{pmatrix} \mathbf{K}^{-1}. \tag{3}$$

The problem of deblurring is actually to compute weight w_i for each pose $\mathbf{H}_i$, physically corresponding to duration of

each pose. There is a mapping between homography $\mathbf{H}_i$ and warping matrix $\mathbf{P}_i$, where $\mathbf{P}_i$ is the $N \times N$ matrix and each row is formed by the coefficient of bilinear interpolation. The blur kernel corresponds to $\mathbf{w} = (w_1, w_2, ..., w_n)^T$ in Eq. (1), weights for each camera pose. n is the total number of poses. Only the ratio of $t_x/z, t_y/z, t_z/z$ is required to parameterize homography $\mathbf{H}$. Under constant depth z, the family of homography in the above blur process forms a 6D parameter space.

H and Uniform Deblurring In the uniform blur model, θ_x, θ_y, θ_z and t_z are all set to zeros. The remaining parameters are t_x/z and t_y/z in estimation. The solution space is thus in 2D, corresponding to shift horizontally and vertically in image plane.

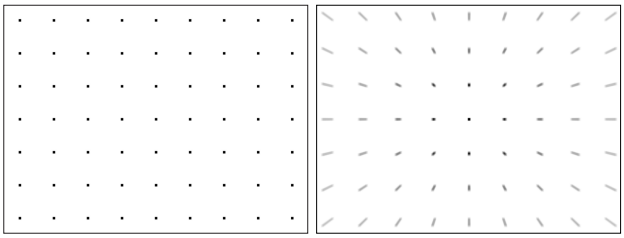
H and Non-uniform Deblurring The full six-parameter model is excessively complex even for non-uniform deblurring. Previous methods, such as those of [23, 4], tackled this problem in 3D subspaces with only rotation or in-plane rotation plus translation. Effectiveness of these methods in approximating blur caused by typical hand-held camera shake was verified [11]. Specifically, a full homography-based model was adopted in [20] for non-blind deconvolution. It is however unknown how to blindly estimate the blur kernel and latent image simultaneously, given the large solution space.

Importance of Out-of-Plane Translation In nearly all prior deblurring methods, out-of-plane translation is ignored, assuming no movement orthogonal to the image plane. As noted in Section 1, this type of translation, however, is dominant in ubiquitous forward motion situations. In what follows, we discuss if motion caused by out-of-plane translation can be modeled by previous methods.

We create an image δ, which contains a few regularly placed dots, as shown in Fig. 2(a). This image is blurred with forward motion only, i.e. the homography in Eq. (3) by setting all elements to 0s except for t_z. The blurred image is shown in Fig. 2(b), visually indicating that the blur is spatially variant in a radial shape. At image center, there is almost no blur; along the boundary, moving the camera 1cm causes a pixel shifting 30 pixels in the 2000×2000 image when setting scene depth $z = 1$ meter.

Note this is a special example used only for illustration. Our method actually handles a more general problem where blur occurs by translation with components existing along all axes and the motion is not necessarily perpendicular to the image plane.

Inherent Difficulty Existing non-uniform methods, such as those of [23, 4], cannot tackle this problem. Another fundamental issue is on the depth value assumption. Almost all practical single-image motion deblurring methods with



(a) Dotted pattern (b) Forward blur

Figure 2. Simple out-of-plane translational blur illustration.

implementation available online assume distant or front-parallel scene, which is actually not appropriate for general forward motion. It is because requiring objects completely undergo translational motion perpendicular to the camera sensor plane is overly restrictive. For example, traffic surveillance cameras are normally placed higher than vehicles or beside highway. Their viewing planes are slanted and pixels are not moving strictly forward.

In our method, we allow for varying depth and modeling them in a parametric form, related to 3D plane normals. This strategy balances system practicality and problem tractability, making the method a reasonable one for forward motion deblurring.

4. Our Model

In our framework, points on various 3D planes are modeled. Their projection on the blurred image is constrained, availing following optimization.

For a 3D plane denoted as $\pi = (\mathbf{n}, d)$, where $\mathbf{n}$ is the normal vector and d is the offset to the camera center, any point $\mathbf{X}$ on the plane satisfies $\mathbf{X}^T\pi = 0$. By convention, the latent sharp image $\mathbf{l}$ is captured by a camera with projection center at the origin. The projection matrix between the world and image plane coordinates is $\mathbf{K}[\mathbf{I}|\mathbf{0}]$. Two images produced by varying camera positions are linked by a homography as

$$\mathbf{H} = \mathbf{K}(\mathbf{R} + \frac{\mathbf{t}\mathbf{n}^T}{d})\mathbf{K}^{-1}, \tag{4}$$

where $\mathbf{R}$ refers to rotational motion and $\mathbf{t}$ denotes translation. This representation is different from Eq. (3) for its consideration of varying depth. Generally, points lying on a 3D plane can be with different depth values.

Because we aim to deal with images primarily produced by car or traffic surveillance cameras, the rotation matrix is set to an identity $\mathbf{I}$. In addition, plane points make $\mathbf{t}/d$ in Eq. (4) a three-variable vector, which is still represented as $[t_x\ t_y\ t_z]^T$. The homography representation similar to Eq. (4) is thus

$$\mathbf{H} = \mathbf{K} \begin{pmatrix} 1 + t_x n_1 & t_x n_2 & t_x n_3 \\ t_y n_1 & 1 + t_y n_2 & t_y n_3 \\ t_z n_1 & t_z n_2 & 1 + t_z n_3 \end{pmatrix} \mathbf{K}^{-1} \tag{5}$$

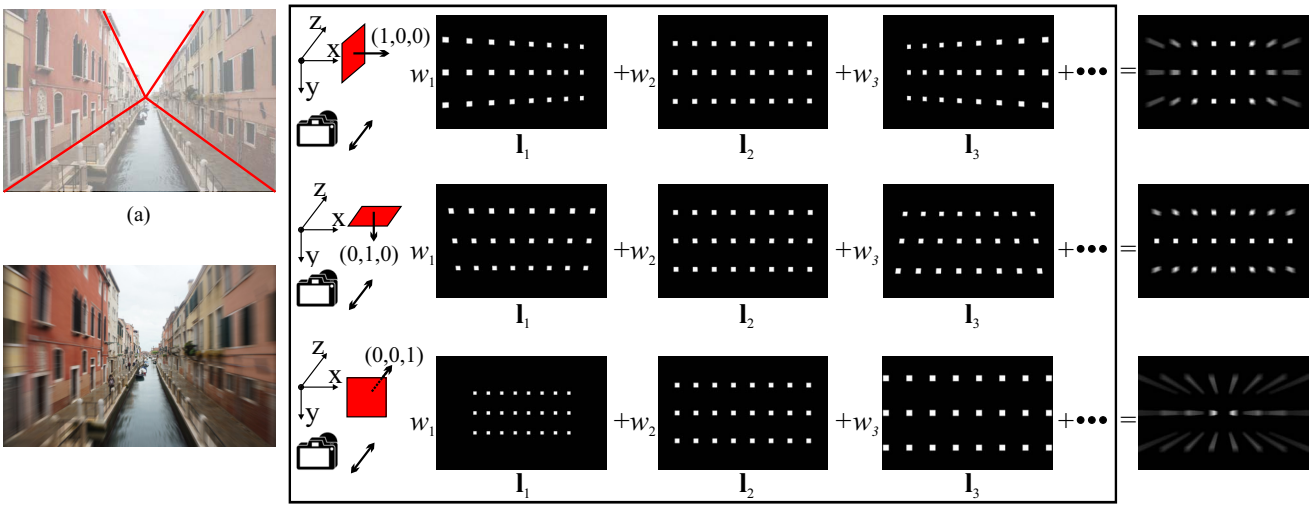


Figure 3. Demonstration of homography bases considering three special normals. (a) Natural images generally contain planar surfaces. (b) Corresponding forward motion blurred image to (a). (c) Three special cases. Each blurred surface is a weighed sum of a few transformed planes $\mathbf{l}_i$ defined in Eq. (1). (d) Resulting blurred planes.

where $\mathbf{n} = (n_1, n_2, n_3)^T$ in 3D space. Given t_x, t_y, t_z and $\mathbf{n}$, we can uniquely determine a homography, which also corresponds to a $N \times N$ warping matrix $\mathbf{P}$ described in Eq. (1) – one $\mathbf{P}$ maps to one homography $\mathbf{H}$. In this regard, we transform originally very difficult whole-image deblurring to a plane-wise tractable problem, counting in non-frontal 3D planes.

Homography Space Eq. (5) indicates that one homography, or the corresponding camera pose, is determined uniquely by vectors $\mathbf{t} = (t_x, t_y, t_z)^T$ and $\mathbf{n} = (n_1, n_2, n_3)^T$ with a total of 6 variables (or 5 of them if $\mathbf{n}$ is normalized). We thus propose constructing 3D homography space $\mathbf{t} = (t_x, t_y, t_z)^T$ and sample each t_x, t_y, and t_z discretely to predefine a few camera poses. The normal $\mathbf{n}$, contrarily, is set as another parameter updated in passes. Put differently, our method uses a series of discrete $\mathbf{H}_i^{\mathbf{n}}$ to present the original continuous homography space, each homography or status is determined by a corresponding $\mathbf{n}$ and by a pose $\mathbf{t}_i$ indexed by i.

Fig. 3 shows an example for demonstrating the specialty of forward motion blur. We use the dotted pattern to visualize point trajectories and homography basis, which make this kind of blur formation easy to comprehend. (a) is to show that forward motion blurred images, such as that in (b), can generally find a few planes.

We then consider three special cases with plane normals $\mathbf{n}$ being respectively $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, as illustrated in the three rows in Fig. 3(c). Previous methods, even for non-uniform deblurring, mostly consider the case $\mathbf{n} = (0, 0, 1)$, whereas our method handles all of them as well as planes with all non-zero elements in normal $\mathbf{n}$.

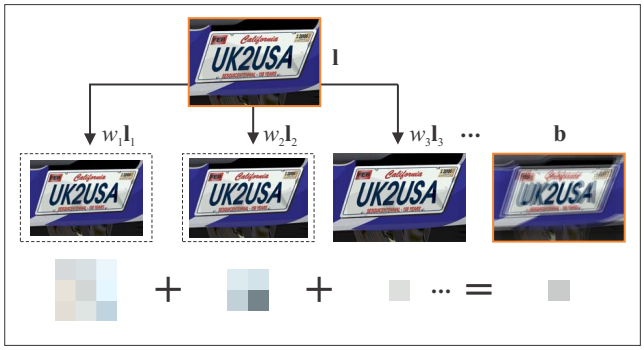


Figure 4. Scale variance in forward motion blur formation. Each $\mathbf{l}_i = \mathbf{P}_i^{\mathbf{n}}\mathbf{G}_i\mathbf{l}$ is with possible scale change as illustrated.

Fig. 3(d) illustrates representative blur structures. For planes with normals $(1, 0, 0)$ and $(0, 1, 0)$ (top two rows in (c)), a column and a row of pixels do not blur at all. It is because these points are infinitely distant or the plane passes the camera center.

Color Mixing Issue The blur formation represented as $\mathbf{b} = \sum_i w_i \mathbf{l}_i + \varepsilon$ in Eq. (1) given $\mathbf{l}_i = \mathbf{P}_i^{\mathbf{n}}\mathbf{l}$ does not consider color mixing arising in forward motion blur, which is caused by scale variation in each transformed image $\mathbf{l}_i$. Previous work does not have this problem because without out-of-plane translation, the transformed images $\mathbf{l}_i$ captured at different camera poses are with similar resolutions.

Note in out-of-plane motion, a similarity transform for each $\mathbf{l}_i$ is resulted in, which changes image scales. In this regard, one pixel in the blurred image is *not* a summation (or integration) of a few isolated unblurred pixels, but rather a combination of several *patches*, as illustrated in Fig. 4.

We generally take the reference image $\mathbf{l}$ as the one with the highest resolution among all. It is with $t_z = 0$ and all other $\mathbf{l}_i$ are with $t_z < 0$, corresponding to down-scaled versions of $\mathbf{l}$. To practically model the resulting blurred image $\mathbf{b}$ and avoid aliasing, we regard $\mathbf{b}$ as the sum of latent images $\mathbf{l}_i$ blurred by Gaussian filter, whose standard deviation is determined by t_z corresponding to each $\mathbf{l}_i$. Typically, we set the standard deviation of Gaussian in $[0.1, 0.5]$ and linearly interpolate pixels according to t_z. This process is similar to image sampling, but with different convolution kernels. The final blur model is finely expressed as

$$\mathbf{b} = \sum_i w_i \mathbf{P}_i^{\mathbf{n}} \mathbf{G}_i \mathbf{l} + \varepsilon, \tag{6}$$

where $\mathbf{G}_i$ is a BTTB (block-Toeplitz with Toeplitz-block) matrix representing the Gaussian blur kernel in a matrix form. Our final $\mathbf{l}_i$ is expressed as $\mathbf{P}_i^{\mathbf{n}} \mathbf{G}_i \mathbf{l}$. We describe in the next section our deblurring algorithm based on this model.

5. Forward Motion Deblurring

The model in Eq. (6) depends on three sets of variables, namely $\mathbf{w}$, $\mathbf{l}$, and normal $\mathbf{n}$. Solving for $\mathbf{w}$ and $\mathbf{l}$ with a fixed $\mathbf{n}$ corresponds to a non-uniform deblurring problem. We resort to alternating minimization to estimate them iteratively.

5.1. Kernel and Image Restoration

The first sub-problem is to fix $\mathbf{n}$ and estimate $\mathbf{w}$ and $\mathbf{l}$, which is referred to as blind deconvolution. $\mathbf{w}$ is known as blur kernel, since it records the duration of each camera pose, conceptually similar to 2D uniform blur PSFs.

One nice property of Eq. (6) is the bilinear form it takes, since $\mathbf{G}_i$ is a linear translation-invariant operator. We thus write

$$\sum_i w_i \mathbf{P}_i^{\mathbf{n}} \mathbf{G}_i \mathbf{l} = \mathbf{B}^{\mathbf{n}} \mathbf{l} = \mathbf{A}^{\mathbf{n}} \mathbf{w}, \tag{7}$$

where $\mathbf{B}^n = \sum_i w_i \mathbf{P}_i^{\mathbf{n}} \mathbf{G}_i$ and $\mathrm{col_i}(\mathbf{A}^n) = \mathbf{P}_i^{\mathbf{n}} \mathbf{G}_i \mathbf{l}$. $\mathrm{col_i}(\cdot)$ returns the i-th column of matrix $\mathbf{A}^n$. $\mathbf{w}$ is the vector $(w_1, w_2, ...)^T$. We define the quadratic data cost term $\|\mathbf{b} - \sum_i w_i \mathbf{P}_i^{\mathbf{n}} \mathbf{G}_i \mathbf{l}\|^2$ following tradition.

Kernel Update We update $\mathbf{w}$ by solving energy function

$$\begin{aligned} E(\mathbf{w}) \quad &= \|\mathbf{A}^{\mathbf{n}} \mathbf{w} - \mathbf{b}\|^2 + \gamma \|\mathbf{w}\|^2, \\ &s.t. \textstyle\sum_i w_i = 1 \end{aligned} \tag{8}$$

where γ controls the smoothing strength. The constraint $\sum_i w_i = 1$ is for energy conservation. The objective function is quadratic with respect to $\mathbf{w}$. However, directly solving Eq. (8) involves inversion of $\mathbf{A}^{\mathbf{n}}$, which is computationally expensive and unstable. We use the local uniform assumption [6] for acceleration considering smoothly changing blur kernels under depth variation on 3D planes.

Conjugate gradient (CG) is employed to update $\mathbf{w}$ in this step. Normalization to make $\sum_i w_i = 1$ is applied after the result of $\mathbf{w}$ is obtained.

Image Update In uniform deblurring, extra steps with shock filter [1] are generally employed to help kernel estimation. Recent development [13, 26] shows that sparsity-pursuit regularization can replace these ad-hoc steps and generate similar or better representations in a unified energy minimization framework. Speed of convergence and result quality can both be enhanced [26]. Our method follows this line and similarly adopts the high-sparsity form as

$$E(\mathbf{l}) = \|\mathbf{B}^{\mathbf{n}} \mathbf{l} - \mathbf{b}\|^2 + \lambda \phi(\nabla \mathbf{l}), \tag{9}$$

where $\phi(\nabla \mathbf{l})$ is the high-sparsity regularization term on image gradients, approximating L_0-norm (defined in [26]). λ is a weight. The scale-invariant property of L_0-sparsity is vital to guide blur kernel estimation. We do not perform shock filtering and instead use the efficient solver in [26] for optimization.

The image and kernel estimates are updated iteratively in a multi-scale scheme [23]. It converges quickly. The finally restored image, based on the kernel result, is produced by optimization incorporating a natural image hyper-Laplacian [12] prior. Other image priors, such as that in [27], can also be adopted.

5.2. Normal Refinement

The above method depends on a specified normal $\mathbf{n}$. We will discuss in Section 6 the way to initialize it. In what follows, we parameterize $\mathbf{n}$ for its refinement.

A plane normal is parameterized in the spherical coordinate system as

$$\mathbf{n} = (n_1, n_2, n_3)^T = (\cos\alpha \sin\beta, \sin\alpha \sin\beta, \cos\beta)^T, \tag{10}$$

where α and β are polar and azimuthal angles respectively. With the constraint that each $\mathbf{n}$ is normalized, the number of parameters is actually 2. The energy function with respect to $\mathbf{n}$ is

$$E(\mathbf{n}) = \|\sum_i w_i \mathbf{P}_i^{\mathbf{n}} G_i \mathbf{l} - \mathbf{b}\|^2. \tag{11}$$

The derivatives of $E(\mathbf{n})$ are non-linear. Gradient descent [3], such as Matlab *fminunc* function, fails to produce reasonable results in practice.

Our strategy to update the normal estimate is sampling and testing. Given the input normal specified by angles α_0 and β_0, we update them within $\alpha_0 \pm 15°$ and $\beta_0 \pm 15°$, with each interval $5°$. We have therefore a total of 49 candidates, which are fed into Eq. (11) for evaluation. The one with the smallest energy is kept as the new normal in the current pass. This process guarantees energy decreasing.

After the normal is updated, we solve for $\mathbf{w}$ and $\mathbf{l}$ again. Only 3 passes are enough to obtain a reasonable result in our experiments.

Figure 5. Illustration for finding vanishing lines. Two sets of parallel lines on a blurred plane are drawn to find two vanishing points. It is easy to locate parallel lines based on scene structure by humans.

6. Implementation and Discussion

We give more details about algorithm implementation, including normal initialization in images. We set λ and γ in Eqs. (9) and (8) to $6E-3$ and $5E-3$ respectively for most examples.

Where Are the Planes? Traffic sign boards, building facade, to name a few, are typical scenes captured by cameras on moving cars. Vehicles are contrarily targets of surveillance cameras. They all consist of planes. We in general set the input scene initially as frontal parallel and let the normal evolve automatically during optimization, as described in previous sections. If there is a quite slanted surface to deblur, we use the following method for manual initialization.

Finding Initial Plane Normals Our method makes use of multi-view geometry [5] if frontal-parallel initialization is not suitable. It is based on the fact that planes' orientation relative to camera coordinates can be determined from vanishing lines. A plane with vanishing line $\mathbf{v}$ has it normal determined as $\mathbf{n} = \mathbf{K}^T\mathbf{v}$.

In our system, two sets of lines parallel to the plane are drawn by the user, snapping to edges in the image, as shown in Fig. 5. We note there are several methods that can automatically or semi-automatically find parallel lines. But they are not reliable enough on blurred images. To develop a robust automatic plane detection method for forward motion blur will be our future work.

Sampling Details We regularly sample t_x, t_y, and t_z to get our homography basis $\mathbf{H}_i^{\mathbf{n}}$, accounting for out-of-plane and in-plane translation. Sampling ranges can be adjusted according to the degree of blurriness. It is also set according to intrinsic parameter $\mathbf{K}$ so that one sample in each direction roughly causes one pixel displacement in the corresponding plane for numerical tractability. Another general principle is to sample z more densely than x and y.

7. Experimental Results

We first evaluate the influence of normal estimation in our model and show how our approach evolves the estimate starting from a coarse initialization. To this end, we use the same blurred image illustrated in Fig. 5, whose normal is much deviated from the initial $(0, 0, -1)^T$. In fact, the human labeled normal is $(0.48, -0.07, -0.87)^T$.

Our system solves for the latent image and blur kernel, and updates this normal iteratively. The outputs from the initial and final (3^{rd}) passes are shown in Fig. 6(c) and (d). Their respective blur kernel and normal estimates are visualized in (g)-(h). These images demonstrate that the initial normal yields a central-symmetric kernel, which is not correct. The deblurred image thus contains visual artifacts and leftover blur (see the close-ups). The final estimate in (h) is with normal $(0.32, -0.12, -0.94)^T$, close to the user labeling result. It manifests the ability of our method to optimize the normal from a coarse initialization.

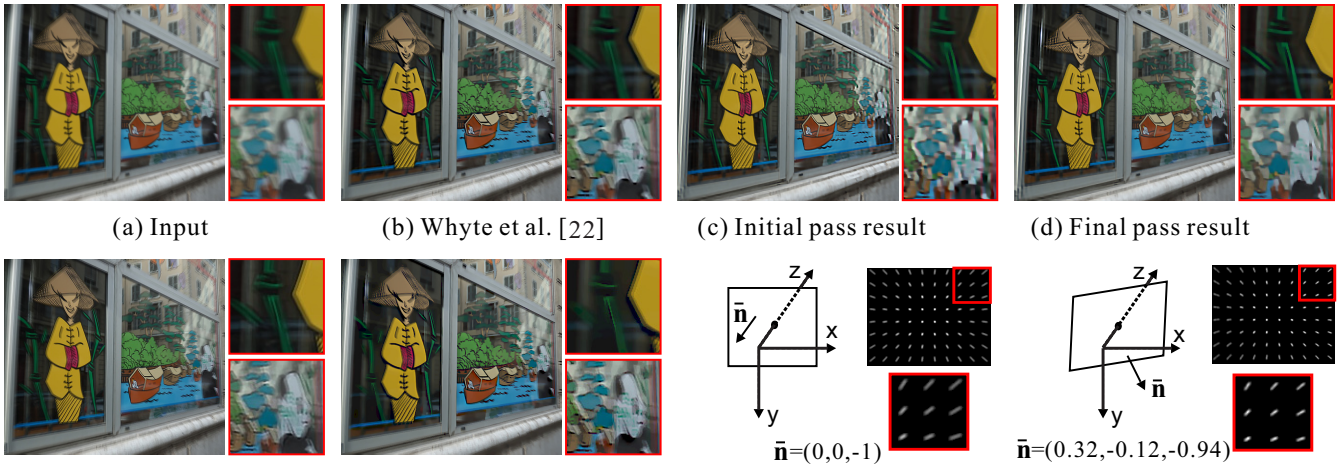
We further compare our result with those produced by other uniform and non-uniform approaches with publicly available implementation. The results of [1, 24, 23] are shown in (b), (e) and (f). Not surprisingly, these methods do not model out-of-plane translation and by nature cannot deblur this image correctly.

A traffic image is shown in Fig. 7 to further demonstrate that existing methods cannot nicely address this type of blur, as shown in (b)-(e). Our method produces a more compelling result from the single image input, shown in (f), with the corresponding 3D plane and blur kernel visualized in (g) and (h). Fig. 8 contains another example.

We show the last challenging image in Fig. 9, with the input also downloaded from internet. It contains a blurred highway traffic sign. The image is taken in a poor lighting condition with high-speed car movement, producing significant blur. The left-most characters in (a) cannot be read. This image is not possible to be restored by prior methods. Our result is shown in (f), with the associated plane and kernel visualized in (g) and (h). It is not perfect due to the existence of heavy noise and JPEG artifacts. But the characters are recognizable after restoration. More examples and data can be downloaded from our project website (see the title page).

8. Conclusion

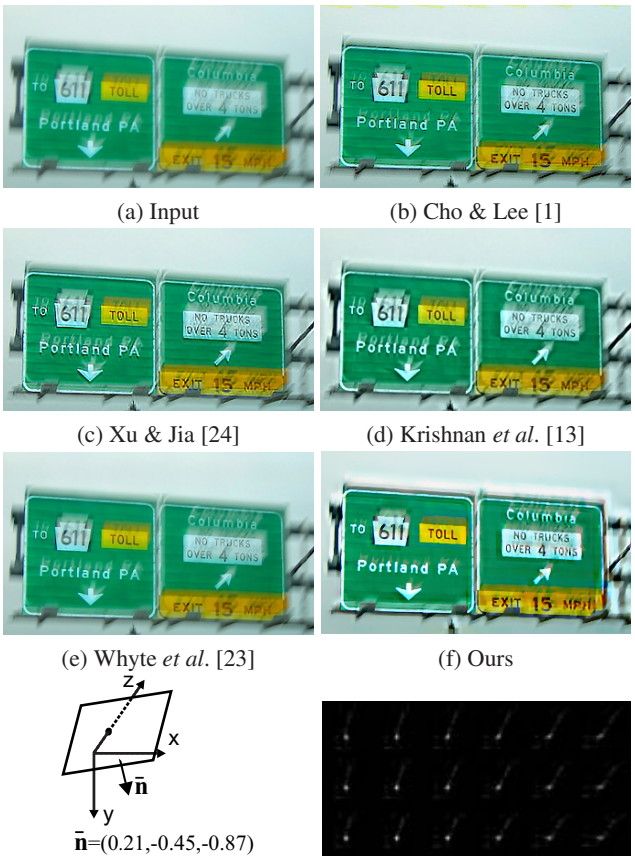
In this paper, we focused on addressing a special and important type of motion deblurring problem, namely for-



(a) Input (b) Whyte et al. [22] (c) Initial pass result (d) Final pass result

(e) Xu & Jia [23] (f) Cho & Lee [1] (g) Initial normal and kernel (h) Final normal and kernel

Figure 6. A slanted surface blurred with forward camera motion. Given the input in (a), we show results of other methods in (b), (e), and (f). The initial and final deblurring results are shown in (c) and (d). The corresponding kernel and normal results are shown in (g) and (h).



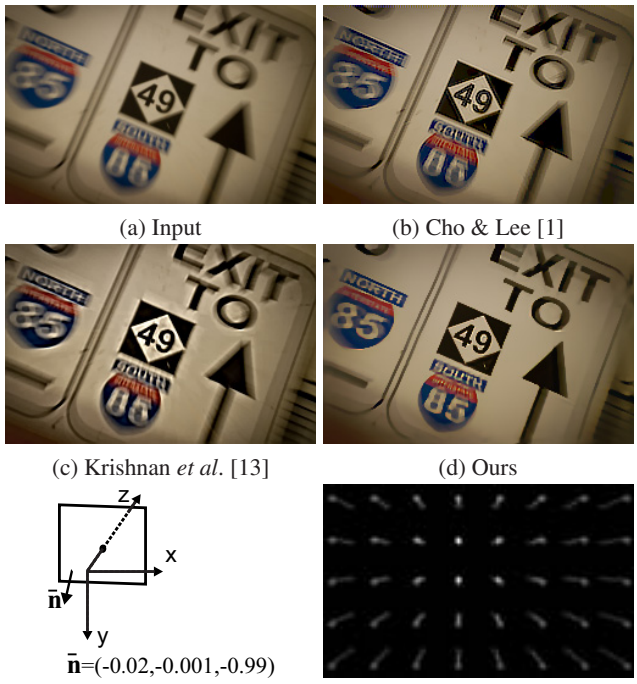
(a) Input (b) Cho & Lee [1]

(c) Xu & Jia [24] (d) Krishnan *et al*. [13]

(e) Whyte *et al*. [23] (f) Ours

(g) Normal (h) Kernel

Figure 7. Traffic sign deblurring.



(a) Input (b) Cho & Lee [1]

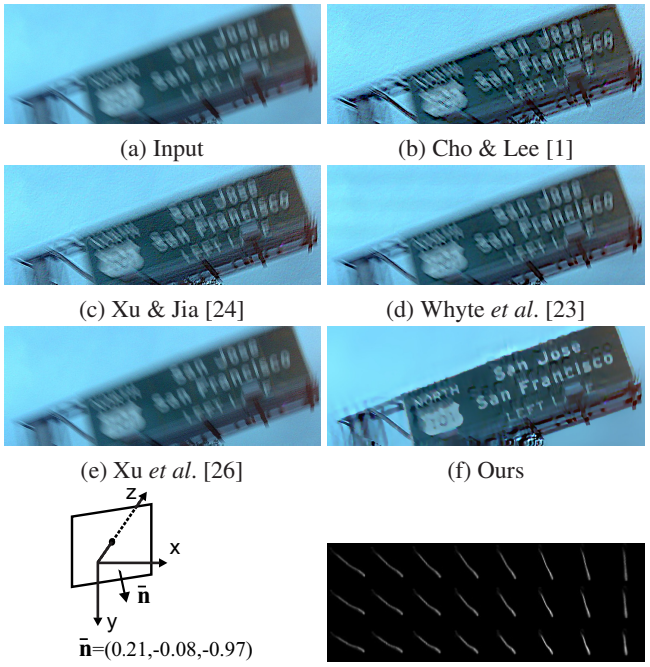
(c) Krishnan *et al*. [13] (d) Ours

(e) Normal (f) Kernel

Figure 8. Road sign deblurring.

ward/backward blur removal, which generally arises for traffic surveillance or vehicle cameras. Its specialty lies on modeling depth variation and pixel blending with high diversity. We presented a method based on 3D plane models, which only needs rough plane normal initialization. Our method has been applied to several challenging examples.

The limitations of the current system include occasional requirement to manually initialize planes and incapability to handle arbitrary-moving objects. For textured areas, saturated pixels in images taken at night, or small patches that lack structural information, our method also does not work very well. Our future work will be to develop methods to



(a) Input (b) Cho & Lee [1] (c) Xu & Jia [24] (d) Whyte *et al*. [23] (e) Xu *et al*. [26] (f) Ours (g) Normal (h) Kernel

Figure 9. Significant forward motion blur. Existing approaches cannot properly handle it. Our result contains characters and numbers, good for recognition.

relax these conditions. It is also possibly doable to fit many images into other geometric representations for deblurring.

Acknowledgements

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 413110).

References

[1] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5), 2009.

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

[3] R. Fletcher and M. J. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2).

[4] A. Gupta, N. Joshi, C. L. Zitnick, M. F. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, pages 171–184, 2010.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[6] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, pages 463–470, 2011.

[7] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *CVPR*, pages 607–614, 2010.

[8] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29(4), 2010.

[9] N. Joshi, R. Szeliski, and D. J. Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, pages 1–8, 2008.

[10] E. Kee, S. Paris, S. Chen, and J. Wang. Modeling and removing spatially-varying optical blur. In *ICCP*, pages 1–8. IEEE, 2011.

[11] R. Koehler, M. Hirsch, S. Harmeling, B. Mohler, and B. Schölkopf. Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database. In *ECCV*, pages 27–40, 2012.

[12] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, pages 1–9, 2009.

[13] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 233–240, 2011.

[14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009.

[15] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664, 2011.

[16] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf. Non-stationary correction of optical aberrations. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 659–666. IEEE, 2011.

[17] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Blind correction of optical aberrations. In *Computer Vision–ECCV 2012*, pages 187–200. Springer, 2012.

[18] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3), 2008.

[19] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, pages 1–8, 2007.

[20] Y. Tai, P. Tan, and M. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1603–1618, 2011.

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

[22] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. In *ICCV Workshops*, pages 745–752, 2011.

[23] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, pages 491–498, 2010.

[24] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170, 2010.

[25] L. Xu and J. Jia. Depth-aware motion deblurring. In *ICCP*, pages 1–8, 2012.

[26] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013.

[27] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486. IEEE, 2011.