Motion from Blur

Shengyang Dai and Ying Wu
EECS Department, Northwestern University, Evanston, IL 60208, USA
{sda690,yingwu}@ece.northwestern.edu

Abstract

Motion blur retains some information about motion, based on which motion may be recovered from blurred images. This is a difficult problem, as the situations of motion blur can be quite complicated, such as they may be space-variant, nonlinear, and local. This paper addresses a very challenging problem: can we recover motion blindly from a single motion-blurred image? A major contribution of this paper is a new finding of an elegant motion blur constraint. Exhibiting a very similar mathematical form as the optical flow constraint, this linear constraint applies locally to pixels in the image. Therefore, a number of challenging problems can be addressed, including estimating global affine motion blur, estimating global rotational motion blur, estimating and segmenting multiple motion blur, and estimating nonparametric motion blur field. Extensive experiments on blur estimation and image deblurring on both synthesized and real data demonstrate the accuracy and general applicability of the proposed approach.

1. Introduction

Within an exposure period, the movements of the camera or the objects produce motion blurred images, as the illuminance changes are integrated over time and the sharpness is smeared. On the other hand, a motion blurred image retains information about motion that parameterizes the blur. Thus it gives us clues to recover motion from this single image.

A motion blur is characterized by its point spread function (PSF) whose parameters are closely related to the motion. The simplest motion blur is the space-invariant linear motion blur, which has been studied extensively. In practice, however, as motion can be quite complex, motion blurs can be much more complicated than this simple case. For example, the blur can be space-variant, nonlinear, local and multiple. Some examples are shown in Fig. 1. Recently, people began to study some of these cases, such as multiple linear invariant blur and rotational blur. However, even for these special cases, current solutions usually rely on multiple input images, user interactions, or extra assumptions.

In order to infer or recover the motion blur (*i.e.*, the hidden unknowns) from the blurred image (*i.e.*, the observable), a critical issue is to establish an observation model. A simple and elegant model will surely lead to effective solutions. A straightforward observation model is the blur generation model where the blurred image is the convolution of the PSF and a sharp image (*i.e.*, the unblurred image). This is not a good observation model, as the sharp image is also not known but free parameters. So, the estimation of the motion blur is always coupled with image deblurring to estimate the sharp image which is a difficult and demanding task. This is the main reason why it is hard to make progress based on this model. It is clear that this convolution model is not a good one, as it has too many degrees of freedom. So, is there a better observation model that has the minimum number of free parameters?

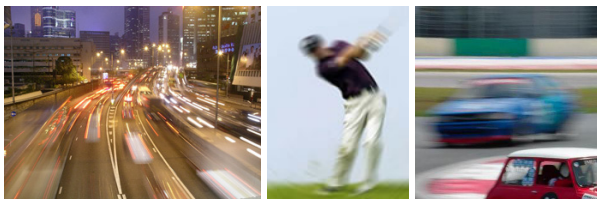


Figure 1. Examples of space-variant motion blur.

A major contribution of this paper is the new finding of a much more elegant and powerful observation model than this convolutional model. We call it the *α-motion blur constraint model*. Instead of working on the blurred image itself, we look at its α-channel. Based on a very mild assumption, we find a linear constraint between the image derivatives of the α-channel, the motion blur parameters, and a simple binary free parameter (*i.e.*, $+1$ or -1). In addition, this constraint applies locally to pixels. The beauty of this model is that we can estimate blur without deblurring. This leads to many other benefits.

Another interesting finding is that this α-motion blur constraint shares a very similar mathematical form as the optical flow constraint. Thus many successful flow-based motion estimation techniques can be adapted to blur estimation. Based on this, many difficult blur estimation problems can be easily unified, including (1) global affine motion blur estimation, (2) global rotational motion blur estimation, (3) multiple blur estimation and segmentation, (4) nonparamet-

ric motion blur field estimation.

This unification enables motion recovery from a single blurred image. In addition, no user interaction is needed, and this is fully automatic. No prior knowledge such as blur direction is assumed, and this achieves blind estimation.

The related work is summarized in Sec. 2. The motion blur constraint is derived in Sec. 3, its analogy with optical flow is presented. The new estimation algorithm for the space-invariant linear case is shown in Sec. 4. Sec. 5 addresses various space-variant blur estimation problems. Experiments are shown in Sec. 6. Sec. 7 concludes the paper.

2. Related work

Motion blur estimation method has been greatly advanced recently. Besides extensive study on estimation of the space-invariant linear motion blur, current works are mainly focused on two directions, *i.e.*, space-variant motion blur estimation and nonparametric blur kernel estimation.

The space-variant motion blur estimation task is motivated by the wide existence of fast object motion within a very short exposure period. The challenge is mainly on the space-variant property of the blur kernel. While the good news is that linear motion blur model can be reasonable assumed in this case, as a first order approximation of the object motion path within an exposure period. The rotational motion deblurring problem is discussed in [25] by using the Fourier transform, which is space-variant but with a global parametric form. For the multiple piecewise invariant blur estimation and segmentation problem, several works [3, 10] utilize multiple input images or an image sequence, which enables image matching. Accurate segmentation results are obtained in [10]. They greatly extend previous works on deblurring with two globally motion blurred images [9, 16, 23]. The idea is that the PSF function could be identified by exploring the commutation property of two motion blurring process. Elongated Gaussian motion blur kernel is assumed in [10, 16] to obtain analytical solution. In [3], a variational approach is proposed, integrated with the background extraction technique. The problem of multiple motion blur estimation from a single input image is approached by [17] based on natural image statistics of image gradient. In stead of estimating the blur in a global way (such as in methods on frequency domain or matching multiple inputs), a local blur estimation method is proposed based on image windows. A single global motion direction is assumed to be known in this work.

Recent advance on kernel estimation is mainly focused on nonparametric blur kernels, which are usually produced by handshaking during the image capturing process with relatively long exposure period under poor lighting conditions. Thus it is reasonable to assume that those kernels are space-invariant over the entire image. In [12], a variational approach is proposed based on natural image statistics. In [15], transparency information is used with user-specified processing regions. A fully automatic algorithm is proposed in [26] by combining information from a blurred/noisy image pair. The Bayesian framework is also used in [1] to estimate the distribution of hyperparameters for kernel estimation. A hardware solution is proposed in [6].

It is easy to see the physical relationship between motion estimation and blur estimation [9]. In [5], space-variant motion model is estimated from the tracking result. Besides estimating the blur kernel, people also try to make use of it by manipulating the aperture [13, 18, 22].

3. Motion blur constraint

In this section, we first derive a motion blur constraint. Based on alpha channel image modeling and sharp edge assumption, an α-motion blur constraint is obtained, and its relationship with the optical flow constraint is revealed.

3.1. General motion blur constraint

We denote the original 2D unblurred continuous signal, the motion blur kernel (point spread function, or PSF), and the blurred signal by I, h, and $I_{\mathbf{b}}$ respectively. In case of space-invariant h, the generation function of the blurred signal is $I_{\mathbf{b}} = I * h$, where $*$ is the convolution operator. Instead of using its direction θ and length l, we parameterize h by its projection length on x and y axes as a vector $\mathbf{b} = (u, v)^{\mathrm{T}}$, where $u = l\cos\theta$, $v = l\sin\theta$. We have

Theorem 1 *A 2D continuous signal I is motion blurred by kernel h to get $I_{\mathbf{b}}$. The following* motion blur constraint

$$\nabla I_{\mathbf{b}}\,|_{\mathbf{p}} \cdot \mathbf{b} = I(\mathbf{p} + \frac{\mathbf{b}}{2}) - I(\mathbf{p} - \frac{\mathbf{b}}{2}) \qquad (1)$$

holds for any position $\mathbf{p}$, where $\nabla I_{\mathbf{b}}\,|_{\mathbf{p}} = (\frac{\partial I_{\mathbf{b}}}{\partial x}, \frac{\partial I_{\mathbf{b}}}{\partial y})^{\mathrm{T}}|_{\mathbf{p}}$.

Please see Appendix for proof. The motion blur constraint is only related to the local gradient and two unblurred pixels, instead of an integral over a large number of pixels as in the generation process of motion blur. It is a local constraint of the blur parameter, and this property is important since it enables blur parameter estimation in a local fashion.

3.2. α-motion blur constraint

Eqn.1 is much simpler than the blur generation equation. However, given only a blurred image, $I(\mathbf{p} - \frac{\mathbf{b}}{2})$, $I(\mathbf{p} + \frac{\mathbf{b}}{2})$, and $\mathbf{b}$ are all unknowns. The idea of alpha channel image modeling can further reduce the degree of freedom.

Alpha channel modeling has been successfully applied on image deblurring [15, 25] and super resolution [11]. With this technique, the image processing task can be much simplified since the edge contrast on alpha channel is normalized to a 0 to 1 transition, instead of arbitrary values in the color space. Recently, the spectral matting [20] method is proposed to extract alpha components automatically.

Image matting techniques are designed to decompose the input image I as a linear combination of foreground image F and background image B through an alpha channel by

$$I = \alpha F + (1-\alpha)B. \quad (2)$$

We call α the alpha channel model of image I (see Fig. 2(a)(b) for an example). If we assume both F and B are locally smooth[1], then

$$I * h = \alpha F * h + (1-\alpha)B * h = \alpha_{\mathbf{b}} F + (1-\alpha_{\mathbf{b}})B, \quad (3)$$

where $\alpha_{\mathbf{b}} = \alpha * h$ is α blurred by kernel h. From Eqn. 3, we know that $\alpha_{\mathbf{b}}$ is the alpha channel model of $I_{\mathbf{b}}$ (see Fig. 2(c)(d) for an example). So the LHS of Eqn. 1 is

$$\nabla(\alpha_{\mathbf{b}} F + (1-\alpha_{\mathbf{b}})B) \cdot \mathbf{b} = (F-B)\nabla\alpha_{\mathbf{b}} \cdot \mathbf{b}. \quad (4)$$

From Eqn. 2, RHS of Eqn. 1 is

$$\big(\alpha(\mathbf{p} + \frac{\mathbf{b}}{2}) - \alpha(\mathbf{p} - \frac{\mathbf{b}}{2})\big)(F-B). \quad (5)$$

Combine Eqn. 4, 5, divide both sides by $F-B$, we have

$$\nabla\alpha_{\mathbf{b}} \cdot \mathbf{b} = \alpha(\mathbf{p} + \frac{\mathbf{b}}{2}) - \alpha(\mathbf{p} - \frac{\mathbf{b}}{2}). \quad (6)$$

The above equation is equivalent to replacing I in Eqn. 1 by its alpha channel model α. If we further assume that the alpha values for most pixels in a non-blurred image are either 0 or 1, then from Eqn. 6, we have $\nabla\alpha_{\mathbf{b}} \cdot \mathbf{b} \in \{0, \pm 1\}$. Experiments show that 0 is taken mostly when $\nabla\alpha_{\mathbf{b}} = 0$. It rarely happens that $\nabla\alpha_{\mathbf{b}}$ is non-zero and perpendicular to $\mathbf{b}$ [2], and those points are treated as outliers in the following sections. Based on this, we assume that for positions where $\|\nabla\alpha_{\mathbf{b}}\| \neq 0$, the following equation holds:

$$\nabla\alpha_{\mathbf{b}} \cdot \mathbf{b} = \pm 1. \quad (7)$$

We call it *α-motion blur constraint*. The elimination of zero case can simplify the parameter estimation process by avoiding trivial solutions. A pre-filtering procedure for outlier rejection is presented in Sec. 4.

Due to the high image quality of current cameras, under good lighting conditions, most image boundaries are clear and sharp for static or slow moving objects. Thus it is reasonable to assume binary α values for most pixels in unblurred images, and α channel transition is mostly caused by motion blur, which is also assumed in [15, 25]. Even in presence of some transition pixels with non-binary α value, Eqn. 7 still holds for most pixels. Other image degradations such as out-of-focus blur are not considered in this work.

Fig. 2(e) illustrates the idea of Eqn. 7 by showing the distribution of pixels in $(\frac{\partial\alpha_{\mathbf{b}}}{\partial x}, \frac{\partial\alpha_{\mathbf{b}}}{\partial y})$ coordinate. Most of the pixels with non-zero gradient lies on two parallel lines, correspondingly to the two linear functions in Eqn. 7. Hough domain for pixels with non-zero gradient is shown in Fig. 2(f), where each gradient vector $(\frac{\partial\alpha_{\mathbf{b}}}{\partial x}, \frac{\partial\alpha_{\mathbf{b}}}{\partial y})^{\mathrm{T}}$ votes for two lines according to Eqn. 7. The two salient points correspond to the two equivalent blur parameters $\mathbf{b}$ and $-\mathbf{b}$.

Please notice that Eqn. 7 is not limited to space-invariant linear motion blur. For space-variant or nonlinear case, if the blur model is locally invariant approximately, and can be locally approximated by linear motion blur, the proposed α-motion blur constraint can still be applied. In this sense, Eqn. 7 can be considered as a local constraint of the motion blur parameter. More detailed discussion of various space-variant or nonlinear cases will be presented in Sec. 5.

[1] It is also assumed in many existing matting algorithms and other works based on alpha channel modeling.

[2] Examples of such points can be found in the Fig. 2(d), where two blurred circles intersect. But even in this case, $\nabla\alpha_{\mathbf{b}}$ is very close to zero.

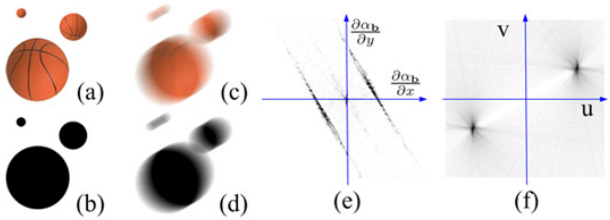


Figure 2. An example with $\mathbf{b} = (30\cos\frac{\pi}{6}, 30\sin\frac{\pi}{6})^{\mathrm{T}}$. (a) I, (b) α, (c) $I_{\mathbf{b}}$, (d) $\alpha_{\mathbf{b}}$, (e) $\nabla\alpha_{\mathbf{b}}$ distribution, (f) Hough domain.

3.3. Analogy between motion and blur

The proposed motion blur constraint is closely related to the well-known optical flow constraint. In this section, we will analyze their similarities and also the difference.

For optical flow, the *brightness constancy assumption* gives $I(x+u, y+v, t+\Delta t) = I(x, y, t)$. Using first order Taylor expansion, the following *optical flow constraint* should hold for every pixel,

$$\nabla I \cdot \mathbf{m} = -I_t, \quad (8)$$

where $\mathbf{m} = (u, v)^{\mathrm{T}}$ is the motion vector, $\nabla I = (I_x, I_y)^{\mathrm{T}}$ is the spatial image gradient, and I_t is the temporal gradient.

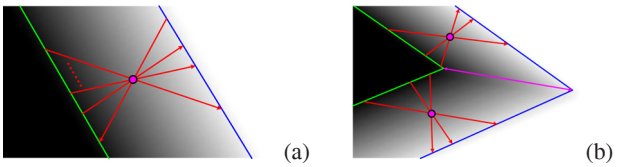
Since motion blur is produced by object motion relative to the camera imaging plane, so it is natural to consider the blur vector $\mathbf{b}$ also as a motion vector, which represents the motion within one camera exposure period.

Besides having similar physical meaning, both the optical flow constraint and the proposed motion blur constraint are also similar mathematically in the following ways:

1. By considering the RHS of Eqn. 1 as the temporal image gradient (the difference between unblurred images taken before and after the motion), the motion blur constraint will have exactly the same form as Eqn. 8.

2. They are both linear constraints of the motion parameters. Particularly, it is interesting to notice that they have very similar aperture problems. The one for motion blur is illustrated in Fig. 3(a). Given only one blurred straight edge, the blur vector could correspond to any arrow shown. This problem could be solved if blurred edges of different directions are given as in Fig. 3(b). Fig. 3(a) also provides an intuitive explanation why the constraint is linear.

3. They are both local constraints, which hold for pixels. In another word, a single pixel alone can provide some cues for the local motion property. Thus more complex (*e.g.*, space-variant, or even nonparametric) motion model could potentially be extracted. This property is especially significant for blur estimation. It can greatly advance the blur estimation technique, since almost all previous methods rely



(a) (b)

Figure 3. The aperture problem for motion blur estimation.

on some global image properties (such as methods on frequency domain or using matching between multiple input images), or statistics over local image windows [17].

These similarities enable the possibility of extending the vast amount of achievement of flow estimation onto the motion blur estimation problem. However, the main difficulty is that the RHS of Eqn. 1 is unknown given single blurred image. Even with alpha channel image modeling and the sharp edge assumption, there is still an unknown parameter $z \in \{\pm 1\}$ in Eqn. 7. It is impossible to remove this unknown, since it is in essential due to the lack of time information for the blur estimation problem with only one input image. In fact, both $\mathbf{b}$ and $-\mathbf{b}$ can produce the same blur result. This property poses extra difficulty for the blur estimation problem comparing with the optical flow problem.

4. Space-invariant motion from blur

The space-invariant linear motion blur can be estimated by minimizing the following objective function

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} \sum_{\mathbf{p}\in\Omega} \min_{z_\mathbf{p}=\pm 1} \left(\nabla\alpha_\mathbf{b}|_\mathbf{p} \cdot \mathbf{b} - z_\mathbf{p}\right)^2, \quad (9)$$

where $\mathbf{p}$ indices pixels, $z_\mathbf{p} = \pm 1$ indicates which constant is actually applied in Eqn. 7 for $\mathbf{p}$. Ω is the set of involved pixels chosen by a pre-filtering technique introduced later in this section. Assume

$$z^*_\mathbf{p} = \arg\min_{z_\mathbf{p}=\pm 1} \left(\nabla\alpha_\mathbf{b}|_\mathbf{p} \cdot \mathbf{b} - z_\mathbf{p}\right)^2, \quad (10)$$

it is obvious that once all $z^*_\mathbf{p}$s are known, Eqn. 9 becomes a standard Least Square Fitting problem, and the solution is

$$\mathbf{b}^* = \mathbf{A}^\dagger \mathbf{Z}, \quad (11)$$

where $\mathbf{A}_{m\times 2} = (\nabla\alpha_\mathbf{b}|_{\mathbf{p}_1}, \nabla\alpha_\mathbf{b}|_{\mathbf{p}_2}, \cdots)^\mathrm{T}$, $\mathbf{A}^\dagger$ is the pseudo inverse of $\mathbf{A}$, $\mathbf{Z}_{m\times 1} = (z^*_{\mathbf{p}_1}, z^*_{\mathbf{p}_2}, \cdots)^\mathrm{T}$, $m = |\Omega|$.

In practice, we use EM-like algorithm to optimize $z^*_\mathbf{p}$ and $\mathbf{b}^*$ iteratively. The initial estimation is obtained by RANSAC to tolerant noisy data. To be more specific, in each round, we select two pixels and randomly assign corresponding z values to solve for the initial blur parameter. The algorithm for space-invariant linear blur parameter estimation is summarized in Fig. 4. The inliers are pixels satisfying $\min_{z=\pm 1} \left|\nabla\alpha_\mathbf{b}|_\mathbf{p} \cdot \mathbf{b} - z\right| < 0.1$ in our experiment.

The second step in Fig. 4 is trying to remove noisy data, since it is well-known that less noisy data can increase the robustness, and also fasten the RANSAC process. In our case, noise comes from two sources: outliers for Eqn. 7 as discussed in Sec. 3.2 and imperfectness of matting component extraction. A local consistency check is applied first,

Input Space-invariantly motion blurred image $I_\mathbf{b}$.
Output Motion blur parameter estimation $\mathbf{b}^* = (u, v)^\mathrm{T}$

1. Get α by using spectral matting [20]. Compute $\nabla\alpha$.
2. Get the set Ω of locally consistent pixels. $n^* = 0$.
3. For $t = 1$ to T:
   (a) Randomly select two pixels in Ω, assign corresponding z values randomly ($\in \{\pm 1\}$).
   (b) Solve motion blur parameter $\mathbf{b}$ by Eqn. 11.
   (c) Get the number of inlier pixels n.
   (d) If $n > n^*$, $n^* = n$, $\mathbf{b}^* = \mathbf{b}$
4. Update z^* and $\mathbf{b}^*$ iteratively by Eqn. 10 and Eqn. 11 for inlier pixels in Ω.

Figure 4. Space-invariant linear motion blur estimation algorithm.

to only keep those pixels which can share the same linear blur model and z value with its neighbors. More formally, we measure the local consistency for pixel $\mathbf{p}$ as

$$C(\mathbf{p}) = \min_{\mathbf{b}} \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} (\nabla\alpha_\mathbf{b}|_\mathbf{q} \cdot \mathbf{b} - 1)^2, \quad (12)$$

where $\mathcal{N}_\mathbf{p}$ is the 3×3 neighborhood window of $\mathbf{p}$. $\mathbf{p} \in \Omega$ if and only if $C(\mathbf{p})$ below a threshold and $\nabla\alpha_\mathbf{b}|_\mathbf{p} \neq 0$. This procedure is applied for all of the following experiments.

Based on the analogy between motion estimation and blur estimation, other techniques for motion estimation could also be adapted to the blur model estimation problem, such as multi-resolution strategy, robust penalty function, more sophisticated gradient computation method, *etc*.

5. Space-variant motion from blur

The local property of the proposed motion blur constraint enables estimating more complex blur model. In this session, after analyzing the generation process of space-variant motion blur, we demonstrate how to apply the α-motion blur constraint on to affine blur estimation, rotational blur estimation, multiple blur model estimation and segmentation, and nonparametric blur estimation.

5.1. Generation of space-variant motion blur

Assume during the exposure period, the integration path of pixel $\mathbf{p} = (x, y)^\mathrm{T}$ is $\mathbf{r}_\mathbf{p}(t) = (x_\mathbf{p}(t), y_\mathbf{p}(t))^\mathrm{T}$, where $t \in [-\frac{t_0}{2}, \frac{t_0}{2}]$ is the time variable, t_0 is the total length of the exposure period. Then the blurred result at $\mathbf{p}$ is

$$I_\mathbf{b}(\mathbf{p}) = \frac{1}{t_0} \int_{t\in[-\frac{t_0}{2}, \frac{t_0}{2}]} I(\mathbf{r}_\mathbf{p}(t))dt. \quad (13)$$

For static regions, $\mathbf{r}_\mathbf{p}(t) = \mathbf{p}$.

For space-invariant linear motion blur, $\mathbf{r}_\mathbf{p}(t) = \mathbf{p} + \frac{\mathbf{b}}{t_0}t$, where $\mathbf{b}$ is the blur parameter. $\mathbf{b} = \mathbf{r}'_\mathbf{p}(t) \times t_0$ is the speed times the moving time, thus can be considered as the first order approximation of the motion path (it is exact here).

For more complex function $\mathbf{r}$, which could be nonlinear or have space-variant derivative, by using the first order approximation and local smoothness assumption, the proposed α-motion blur constraint can still be applied.

5.2. Affine motion from blur

Two dimensional affine motion can be used to better approximate the blur field of a 3D motion, which may not be parallel to the imaging plane, than the simple space-invariant model. Assume the affine blur model as follows

$$\mathbf{r}_{\mathbf{p}}(t) = \mathbf{p} + \mathbf{v}_{\mathbf{p}} t, \tag{14}$$

$$\mathbf{b}_{\mathbf{p}} = \mathbf{v}_{\mathbf{p}} t_0 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}. \tag{15}$$

Put Eqn. 15 in Eqn. 7, we have

$$d_x\big(a_{11}x+a_{12}y+a_{13}\big)+d_y\big(a_{21}x+a_{22}y+a_{23}\big) = z_{\mathbf{p}}, \tag{16}$$

where $d_x = \frac{\partial \alpha_{\mathbf{b}}}{\partial x}|_{\mathbf{p}}$ and $d_y = \frac{\partial \alpha_{\mathbf{b}}}{\partial y}|_{\mathbf{p}}$. Thus

$$(xd_x, yd_x, d_x, xd_y, yd_y, d_y)^{\mathrm{T}} \cdot \mathbf{b}_a = z_{\mathbf{p}}, \tag{17}$$

which is linear for $\mathbf{b}_a = (a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23})^{\mathrm{T}}$, the motion blur parameter. Thus the model estimation problem becomes a standard Least Square Fitting problem given $z_{\mathbf{p}}$. It can be solved in a similar way as the algorithm in Fig. 4.

5.3. Rotational motion from blur

Rotational motion blur is also commonly observable in real life. It is space-variant and nonlinear (the integration path is a circular arc). Assume the object rotates with a center $\mathbf{p}_0 = (x_0, y_0)^{\mathrm{T}}$ and a constant angular speed ρ, then

$$\mathbf{r}_{\mathbf{p}}(t) = \mathbf{p}_0 + R(\rho t)(\mathbf{p} - \mathbf{p}_0), \tag{18}$$

where $R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is the rotation matrix. So

$$\begin{cases} x_{\mathbf{p}}(t) = x_0 + \cos(\rho t)(x - x_0) - \sin(\rho t)(y - y_0) \\ y_{\mathbf{p}}(t) = y_0 + \sin(\rho t)(x - x_0) + \cos(\rho t)(y - y_0) \end{cases} \tag{19}$$

If ρt_0 is very small, $\cos(\rho t) \simeq 1$, and $\sin(\rho t) \simeq \rho t$, then

$$\begin{cases} x'_{\mathbf{p}}(t) = -\rho(y - y_0) \\ y'_{\mathbf{p}}(t) = \rho(x - x_0) \end{cases} \tag{20}$$

So the local motion blur parameter is

$$\mathbf{b}_{\mathbf{p}} = \mathbf{r}'_{\mathbf{p}}(t) t_0 = \big(-\rho(y - y_0)t_0, \rho(x - x_0)t_0\big)^{\mathrm{T}}. \tag{21}$$

Let $\mathbf{b}_r = (\beta, a, b)^{\mathrm{T}} = (\rho t_0, \rho x_0 t_0, \rho y_0 t_0)^{\mathrm{T}}$, thus from Eqn. 7

$$(b - \beta y)d_x + (\beta x - a)d_y = z_{\mathbf{p}}, \tag{22}$$

or

$$(xd_y - yd_x, -d_y, d_x)^{\mathrm{T}} \cdot \mathbf{b}_r = z_{\mathbf{p}}, \tag{23}$$

which again becomes a Least Square Fitting problem given $z_{\mathbf{p}}$, and it can be solved in a similar way as the algorithm in Fig. 4. In fact, after linearizing $\mathbf{r}_{\mathbf{p}}(t)$, the rotational motion blur becomes a special case of affine motion blur.

5.4. Motion segmentation from blur

There might be multiple motion blurred objects with a still background in a single image. To handle this case, a multiple blur model estimation algorithm is proposed, followed by an MRF segmentation method.

To estimate multiple motion blur models, RANSAC is first applied to extract those models one-by-one as initialization, until no more model could be fitted by a meaningful proportion of pixels. The number of models is also automatically determined. After initialization, an EM-like algorithm can further refine those models by updating the model assignment and model parameters iteratively.

Once having the model parameters, MRF is used to enforce region continuity of the motion segments by minimizing the following energy function

$$E(\{l_{\mathbf{p}}\}_{\mathbf{p}\in I}) = \sum_{\mathbf{p}\in I} \phi_{\mathbf{p}}(l_{\mathbf{p}}) + \beta_1 \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{N}} \psi_{\mathbf{p},\mathbf{q}}(l_{\mathbf{p}}, l_{\mathbf{q}}), \tag{24}$$

where $l_{\mathbf{p}}$ is the pixel label, indicating which blur model is applied for pixel $\mathbf{p}$ or it is non-blurred. Assume $\mathbf{b}_l$ is the l-th extracted model, $\phi_{\mathbf{p}}(l) = \min_{z=\pm 1} |\nabla\alpha_{\mathbf{p}} \cdot \mathbf{b}_l - z|$ is the likelihood term which favors better fitting of Eqn. 7. The penalty for setting a pixel as non-blurred is fixed as β_2. The prior term $\psi_{\mathbf{p},\mathbf{q}}(l_{\mathbf{p}}, l_{\mathbf{q}}) = (1 - \delta(l_{\mathbf{p}}, l_{\mathbf{q}}))e^{\beta_3 \|I_{\mathbf{p}} - I_{\mathbf{q}}\|}$ prefers label continuity, $\delta(l_1, l_2) = 1$ if $l_1 = l_2$, and 0 otherwise. The pairwise penalty is large for position around edges to favor segmentation boundary at homogeneous regions. Since each image edge usually undergoes the same motion model and homogeneous regions tend to be ambiguous for different motion models. $\mathcal{N}$ is the set of neighboring pixel pairs, β_1, β_2, and β_3 are constants, and set to 1.8, 0.6, and 5.0 in our experiments respectively. The above energy function can be optimized efficiently by Graph Cuts [8].

5.5. Nonparametric motion from blur

We propose a nonparametric motion blur estimation method to handle more complex motion patterns.

Starting from the classical Lucas-Kanade [21] and Horn-Schunck [14] methods, nonparametric optical flow estimation has been extensively investigated [2, 4]. Due to the analogy between optical flow and motion blur, those methods could potentially be applied. We extend the HS method directly, resulting the following objective function,

$$\sum_{\mathbf{p}\in\Omega} \min_{z_{\mathbf{p}}=\pm 1} (\nabla\alpha_{\mathbf{b}} \cdot \mathbf{b}_{\mathbf{p}} - z_{\mathbf{p}})^2 + \lambda \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{N}} \|\mathbf{b}_{\mathbf{p}} - \mathbf{b}_{\mathbf{q}}\|_2^2, \tag{25}$$

where the variables are motion vector field $\mathbf{b}_{\mathbf{p}}$ for each pixel $\mathbf{p}$ in the image. The first term is the data term based on Eqn. 7, and the second term is the prior term preferring smooth motion field. $\mathcal{N}$ is the set of all neighboring pixel pairs. The difference with HS method is that we have $z_{\mathbf{p}}$ due to the lack of time information, which makes the objective function non-convex, thus very difficult to be optimized. To address this issue, blur model with parametric form is estimated first, and the initial model is used to get the z value. Better initialization strategy such as using a number of piece-wise parametric models as in [7] could also be applied. Once having z, the motion field is optimized by standard iterative algorithm based on Jacobi method.

(a) (b)

Figure 5. (a) A color image and its alpha channel representation α, (b) top: original unblurred image (70×70) and one blurred image with $\mathbf{b} = (20, 20)^{\mathrm{T}}$, bottom: error ($\|\mathbf{b}^* - \mathbf{b}\|$) map of parameter estimation for all $\mathbf{b} \in \{(u, v)^{\mathrm{T}} \,|\, u, v \in \{0, 1, ..., 30\}\}$.

5.6. Removing space-variant motion blur

We modify the Richardson-Lucy method [24] for removing space-variant but locally smooth motion blur. Assume the blurred image $I_{\mathbf{b}}$ is generate by $I_{\mathbf{b}}(\mathbf{p}) = \sum_{\mathbf{q}} I(\mathbf{q}) P_{\mathbf{pq}}$, where I is the clear image we want to recover, $\mathbf{p}$ and $\mathbf{q}$ are pixels, $P_{\mathbf{pq}}$ is the blur kernel weight from $\mathbf{q}$ to $\mathbf{p}$. Then the RL iteration is $I^{t+1}(\mathbf{q}) = I^t(\mathbf{q}) \sum_{\mathbf{p}} \frac{I_{\mathbf{b}}(\mathbf{p})}{I^t_{\mathbf{b}}(\mathbf{p})} P_{\mathbf{pq}}$, where $I^t_{\mathbf{b}}(\mathbf{p}) = \sum_{\mathbf{q}} I^t(\mathbf{q}) P_{\mathbf{pq}}$. However, for the space-variant case, $\sum_{\mathbf{p}} P_{\mathbf{pq}}$ may not equal to 1. Thus even if $I^t = I$ is a perfect estimation, $I^{t+1} \neq I$ will be wrong. This will produce artifacts. To avoid this problem, we first compute $Q_{\mathbf{pq}} = P_{\mathbf{pq}} / \sum_{\mathbf{p}} P_{\mathbf{pq}}$ as a normalized P *w.r.t.* each $\mathbf{p}$, then iterate with Eqn. 26. Applying this method on images with multiple parametric form motions [17, 10] requires precise segmentation, and we leave it as future work.

$$I^{t+1}(\mathbf{q}) = I^t(\mathbf{q}) \sum_{\mathbf{p}} \frac{I_{\mathbf{b}}(\mathbf{p})}{I^t_{\mathbf{b}}(\mathbf{p})} Q_{\mathbf{pq}}. \tag{26}$$

6. Experiments

We first test our algorithm with synthesized data to get quantitative evaluation, then use real data to demonstrate the general applicability. Please refer the electronic version for better visualization of all results in this section.

The spectral matting algorithm [20] is applied independently on overlapping sliding windows of the input image for computational reason, which is very robust in our experiment. In fact, the effectiveness of a closely related approach [19] has been demonstrated by several recent works [11, 15, 25]. The number of alpha components is limited to 2 for each window. The alpha value for each window is set to be one of these two components. The computation of $\nabla\alpha$ and the pre-filter step are also performed within each window. One example is shown in Fig. 5(a). The alpha value of one homogeneous region in two neighboring blocks could be flipped. Some edges are missing due to the limit of component number. All motion blur kernels are generated by Matlab. To synthesize space-variant motion blur given the pixel path function, we sample the exposure period uniformly, and average those warped images.

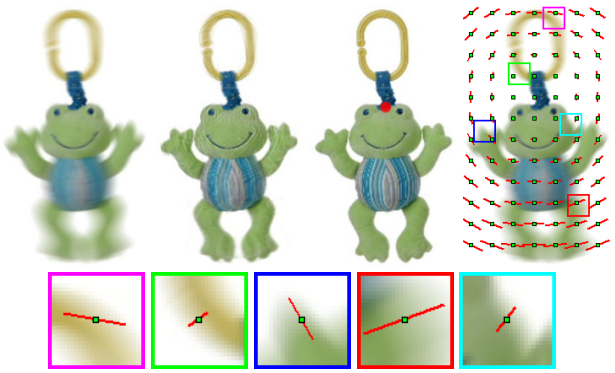


Figure 7. Rotational motion from blur with synthesized data. Top: blurred image, our deblur result, ground truth, visualization of the rotational blur field estimation result, bottom: close-up views.

6.1. Synthesized data

Fig. 5(b) illustrates the accuracy of the proposed algorithm for space-invariant motion with a proof-of-concept experiment. The proposed algorithm is tested for all $\mathbf{b} \in \{(u, v)^{\mathrm{T}} | \, u, v \in \{0, 1, ..., 30\}\}$, which is a fairly large range for blur parameters. The estimation error is visualized. Accurate motion parameters are obtained under the non-ideal conditions with discretized image grid, discretized blur kernel, and not purely binary edges in the non-blurred image. The error for small blur vector ($\|\mathbf{b}\| \leq 2$) is due to very small number of pixels for edge transition.

Fig. 6 shows an affine blur estimation result. The input image is synthesized with an affine blur model $\mathbf{b}_a = (-0.6, 1.8, 1000, 2.4, -3.6, -1000)^{\mathrm{T}} \times 10^{-2}$, and the estimation using the proposed algorithm is $\mathbf{b}^*_a = (-0.54, 2.23, 940, 2.55, -3.54, -1054)^{\mathrm{T}} \times 10^{-2}$. The maximum estimation error over the entire image is 0.96, and the average is 0.43, which is very small comparing with the average length of motion vector 15.8. The deblur result is clear and sharp. Three close-up views are shown together with their local blur vectors. The estimated blur vectors match with the blur extent precisely. Fig. 7 shows a rotational blur estimation result. The input image is synthesized with parameters $x_0 = 90$, $y_0 = 120$, $\beta = \pi/18 \simeq 0.175$. The estimation result is $x^*_0 = 89.6$, $y^*_o = 113$, $\beta^* = 0.169$, which are very close, thus successful deblurring is enabled.

Fig. 8 shows a result for multiple motion blur estimation and segmentation. To synthesis the input image, we first use alpha matting [19] to extract foreground of Lena f together with its alpha channel m. Assume the two blur kernels are h_1 and h_2, then the input image I is generated by $I = (f * h_1) \cdot (m * h_1) + (b * h_2) * (1 - m * h_1)$, where b is the non-blurred background image. The blur parameters are $\mathbf{b}_f = (15.0, 0.0)$ for h_1 and $\mathbf{b}_b = (6.0, 10.4)$ for h_2. The initial estimations by RANSAC are $\mathbf{b}^*_f = (14.7, -0.4)$ and $\mathbf{b}^*_b = (6.5, 10.0)$. They are refined to $\mathbf{b}^*_f = (15.0, -0.2)$ and $\mathbf{b}^*_b = (6.5, 10.4)$ finally. Fig. 8 (d) shows the voting map in the blur parameter space. Two pairs of positions get

Figure 6. Affine motion from blur with synthesized data. Left to right: affine blurred image, deblur result, ground truth, visualized affine motion field estimation result with close-up views (please notice that the estimated local blur vector matches the blur extend precisely).

Figure 8. Motion segmentation from blur. (a)(b) The penalty terms ϕ for two models in range $[0, 2]$, (c) segmentation result, (d) voting map in the Hough domain and visualized blur parameters.

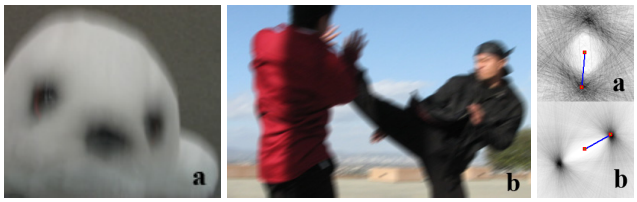


Figure 9. Space-invariant linear motion blur and Hough domain.

higher votes, corresponding to the two motion blur vectors visualized in this figure. The penalty terms shown in (a)(b) can provide valid cues for model fitting, which enables the final segmentation result in (c). Some pixels fit both models well (such as those in the leftup corner), which is due to the aperture problem discussed in Sec. 4. The correct model can be selected by enforcing region continuity with MRF. There are some outliers in the texture region (such as Lena's hair) for Eqn. 7, which is discussed in Sec. 3.2.

6.2. Real data

Fig. 9 shows some results for space-invariant motion blur estimation, and the estimated motion blur vectors overlaid on the voting maps in the Hough domains, where the peaks can be clearly observable. Fig. 10 and Fig. 11 show the blur estimation results with affine, rotational, and nonparametric blur models. From the close-up views, we can see that the local blur vectors match with the blur extents precisely. Image details are greatly enhanced (such as book title, fingers) due to the correctly extracted space-variant blur parameters with our estimation and the modified RL debluring algorithm. In Fig. 10, we compare our result with a space-invariant blur model combined with deconvlucy function in Matlab (#iterations is 30 for both). Our result is much

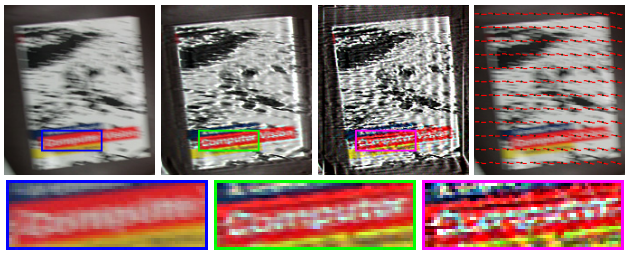


Figure 10. Affine motion from blur with real data. 1st row: blurred image, our deblur result, deblur with space-invariant motion blur model obtained by averaging the estimated affine blur field, and estimated blur field, 2nd row: close-up views of the three images in the 1st row, the book title becomes recognizable in our result.

Figure 12. More results for space-variant motion from blur. The motion blur models for the top row: nonparametric, space-invariant plus non-blurred, bottom row: affine, space-invariant plus non-blurred, and rotational plus non-blurred.

better due to the space-variant property of the underlining blur model. There exist some ringing effects in the deblur results, which is commonly observable [12, 26] in the literature for iterative restoration methods. More results including the motion segmentation results are shown in Fig. 12.

7. Conclusion

In this paper, the α-motion blur constraint is obtained, which is a local linear constraint for the blur parameter. The analogy between the optical flow constraint and the motion blur constraint enables solution for a number of space-variant motion blur estimation problems, including estimation of affine blur, rotational blur, nonparametric blur, and

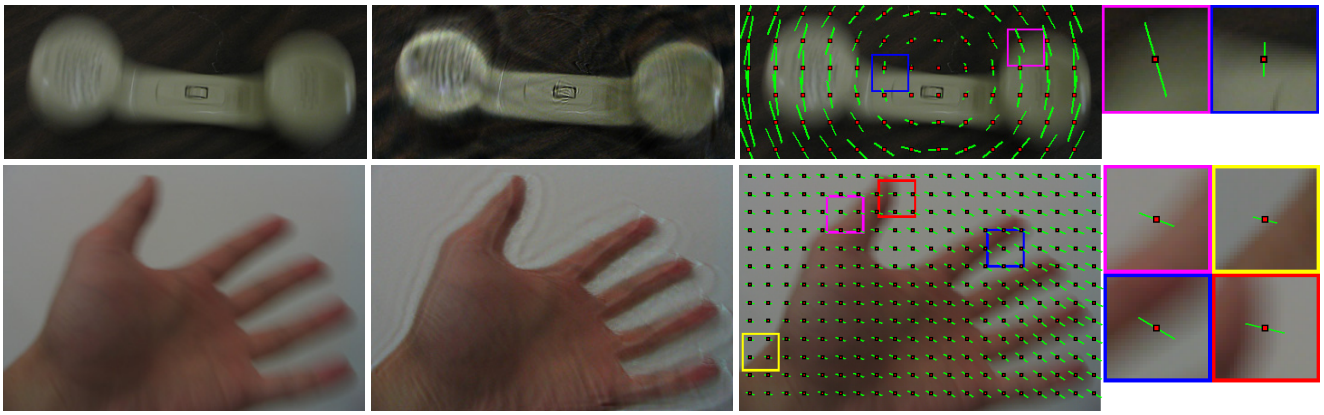


Figure 11. Rotational (top) and nonparametric (bottom) motion from blur with real data. From left to right: blurred image, our deblur result, visualization of estimated motion blur vector field and close-up views.

multiple blur estimation and segmentation. Future work includes more principled approach to deal with outliers of Eqn. 7, especially for texture regions, and more sophisticated estimation of z in Eqn. 10 for nonparametric case.

Acknowledgments

This work was supported in part by National Science Foundation Grants IIS-0347877 and IIS-0308222.

References

[1] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Total variation image restoration and parameter estimation using variational distribution approximation. In *ICIP*, 2007. 2

[2] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007. 5

[3] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *ICCV*, 2007. 2

[4] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *IJCV*, 1994. 5

[5] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and superresolution from an image sequence. In *ECCV*, 1996. 2

[6] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *IEEE Trans. on PAMI*, 2004. 2

[7] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 1996. 5

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on PAMI*, 23(11):1222–1239, 2001. 5

[9] W.-G. Chen, N. Nandhakumar, and W. N. Martin. Image motion estimation from motion smear - a new computational model. *IEEE Trans. on PAMI*, 1996. 2

[10] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007. 2, 6

[11] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. In *CVPR*, 2007. 2, 6

[12] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2006. 2, 7

[13] S. W. Hasinoff and K. N. Kutulakos. A layer-based restoration framework for variable-aperture photography. In *ICCV*, 2007. 2

[14] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981. 5

[15] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007. 2, 3, 6

[16] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *CVPR*, 2005. 2

[17] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2006. 2, 4, 6

[18] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2007. 2

[19] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. *IEEE Trans. on PAMI*, 2007. 6

[20] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *CVPR*, 2007. 2, 4, 6

[21] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 5

[22] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2006. 2

[23] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 2005. 2

[24] W. H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 1972. 6

[25] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007. 2, 3, 6

[26] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2007. 2, 7

A. Proof of Theorem. 1

Without loss of generality, assume $v = 0$. Since otherwise, we can rotation both h and I to make the blur direction align with the x-axis, which will not change the values of both sides of Eqn. 1. In this case, the blur kernel is

$$h(x,y) = \begin{cases} \frac{1}{u}\delta(y) & -\frac{u}{2} \le x \le \frac{u}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

where δ is the Dirac delta function. We have $\frac{\partial h}{\partial x} = \frac{1}{u}[\delta(x + \frac{u}{2}, y) - \delta(x - \frac{u}{2}, y)]$. Since $\mathbf{b} = (u, 0)^{\mathrm{T}}$, so $\forall\, \mathbf{p} = (x, y)^{\mathrm{T}}$,

$$\begin{aligned} \nabla I_{\mathbf{b}}|_{\mathbf{p}} \cdot \mathbf{b} &= \frac{\partial I_{\mathbf{b}}}{\partial x} \cdot u + \frac{\partial I_{\mathbf{b}}}{\partial y} \cdot 0 = \big(I * \frac{\partial h}{\partial x}\big) \cdot u \\ &= I * \frac{1}{u}[\delta(x + \frac{u}{2}, y) - \delta(x - \frac{u}{2}, y)] \cdot u \\ &= I(\mathbf{p} + \frac{\mathbf{b}}{2}) - I(\mathbf{p} - \frac{\mathbf{b}}{2}). \end{aligned} \tag{28}$$