# Single Image Motion Deblurring Using Transparency [*]

Jiaya Jia
Department of Computer Science and Engineering
The Chinese University of Hong Kong
leojia@cse.cuhk.edu.hk

## Abstract

*One of the key problems of restoring a degraded image from motion blur is the estimation of the unknown shift-invariant linear blur filter. Several algorithms have been proposed using image intensity or gradient information. In this paper, we separate the image deblurring into filter estimation and image deconvolution processes, and propose a novel algorithm to estimate the motion blur filter from a perspective of alpha values. The relationship between the object boundary transparency and the image motion blur is investigated. We formulate the filter estimation as solving a Maximum a Posteriori (MAP) problem with the defined likelihood and prior on transparency. Our unified approach can be applied to handle both the camera motion blur and the object motion blur.*

## 1. Introduction

Image blur is caused either by the camera motion [1] or by the object motion [20]. Camera motion is the camera vibration when shutter is pressed. The corresponding motion blur is usually modeled as a linear image degradation process

$$I = L \otimes f + n, \tag{1}$$

where $I$, $L$, and $n$ represent the degraded image, unblurred (or latent) image, and the additive noise respectively. $\otimes$ is the convolution operator and $f$ is an unknown linear shift-invariant point spread function (PSF). Conventional *blind deconvolution* approaches focus on the estimation of $f$ to deconvolve $I$ [13] using image intensities or gradients. One example of the camera motion blur is shown in Fig. 1 (a).

The problem of restoring a still image containing a motion blurred object cannot be completely solved by the blind deconvolution techniques because the background may not undergo the same motion. The PSF has a uniform definition only on the moving object. Raskar *et al*. [20] propose a solution to this problem, in part of their method, assuming that the background is known or has constant colors inside the blurred region. We show one example in Fig. 1 (b) where the motion of an athlete causes the blur while the background is still clear.

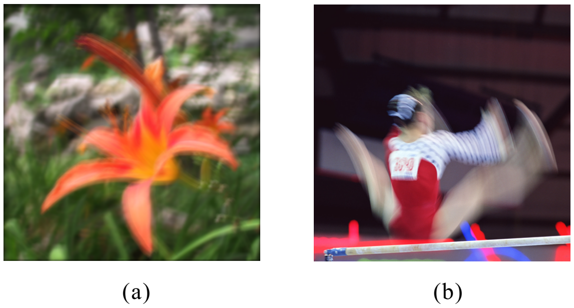
(a) (b)

Figure 1. Two types of motion blur. (a) One example of camera motion blur. The image is blurred by the camera vibration. All pixels in the image are affected. (a) One example of the object motion blur. The image is degraded by object motion.

Although there have been many image restoration techniques proposed, without knowing the blur filter, few of them can be readily applied to solve both of the above two motion deblurring problems. In this paper, as a first attempt, a unified approach is proposed to estimate the motion blur filter from a transparency point of view. Suppose that an object, which is originally opaque and has solid boundary, is motion blurred. Its boundary is blended to the background. *The transparency on this blurred object is primarily caused by its motion during image capture*. Based on this observation, this paper contributes in the following ways:

1. We give a principal analysis on how the image degradation and the transparency is related. We prove that the blur filter can be determined by the transparency on the object boundary.

2. We propose an optimization method to estimate the blur filter by solving a maximum *a posteriori* (MAP)

[*] The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 412206)

1-4244-1180-7/07/$25.00 ©2007 IEEE

problem only using transparency information. Without directly taking all pixel colors and complex image structures into computation, our method is efficient and robust.

3. We introduce the *generalized transparency*, based on which we solve the restoration problem on general camera motion blurred images.

The rest of the paper is organized as follows: We review previous work in Section 2. The analysis of the object motion blur using transparency is given in Section 3. In Section 4, we present our approach to solve the 2-D object motion blur using transparency. We further extend our method to solve the camera motion blur using *generalized transparency* in Section 5. Experimental results are shown in Section 6. Finally, we discuss our approach and conclude our paper in Section 7.

## 2. Previous Work

With unknown linear shift-invariant PSF [5], early approaches usually assume *a priori* knowledge to estimate the blur filter in deblurring the input image. In [2], with the defined parametric model, the values of the unknown parameters are estimated by inspecting the zero patterns in the Fourier transformation. The approaches proposed in [11, 12] model a blurred image as an autoregressive moving average (ARMA) process. The blur filter estimation is transformed to a problem of estimating the parameters of the ARMA process. Many methods solve the image deblurring using generalized cross-validation or maximum likelihood (ML) estimation. Expectation maximization (EM) is commonly adopted to maximize the log-likelihood of the parameter set. Kim *et al.* [9] restore the out-of-focus blurred image by estimating the filter in a parametric form. In [10, 19], recursive inverse filtering methods (RIF) are proposed to iteratively solve the image restoration problem. Extensive surveys on blind deconvolution can be found in [13, 6].

Recently, Fergus *et al.* [4] propose a variational Bayesian approach using an assumption on the statistical property of the image gradient distribution to approximate the unblurred image. An ensemble learning algorithm is employed which is originally proposed in [18] to solve the image separation problem. This method is proven effective in deblurring natural images and estimating complex blur filters.

Image deblurring systems are also developed using hardware or multiple images to obtain more object structure and motion information in restoring degraded images. In [1], a high-resolution still image camera and a low-resolution video camera are connected where the low-resolution video camera is used to estimate the PSF. In [16], using fast image capture in CMOS, the motion deblurring from multiple images is achieved. Rav-Acha *et al.* [21] restore a clear image using multiple images with motion blur along different directions. An iterative reconstruction approach is proposed. Jia *et al.* [8] enhance a short-exposure dark image by transferring the color from a long-exposure blurred image. Raskar *et al.* [20] compute a binary coded exposure sequence to reduce the loss of high spatial frequencies in blurred objects. In [17], out-of-focus blur is considered in video matting using multiple cameras.

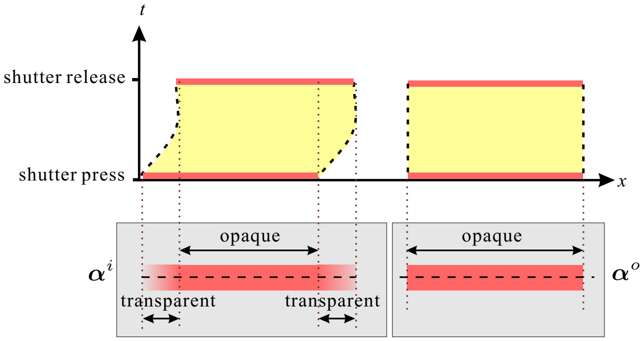

Figure 2. 1-D example showing how the transparency is produced on a motion blurred object. On the left the figure, we show a motion blurred red bar whose boundary on the left and right is blended to the background. The right of the figure shows that if there is no object motion during image capture, a binary transparency map will be produced.

## 3. Analysis of Object Motion Blur

We begin the description of our method from the analysis of the object motion blur.

### 3.1. 1-D object motion blur

For simplicity, we show in Fig. 2 an 1-D illustration, in which a *solid and opaque* object (highlighted as a red bar) only moves horizontally. The vertical coordinate in the figure is time $t$. There are two time stamps on axis $t$ labeled as “shutter press” and “shutter release”.

In the upper-left subfigure, the opaque object moves in duration of the image capture. Its trajectory is shown as the dashed curve. In the captured image shown immediately below, one can observe that the center of the object within the inner two dotted lines is not blended to the background since in any time during the image capture, background is occluded. The object boundary on the left and right partially occludes the background, and is shown as semitransparent. The transparency for each pixel is exactly determined by the proportion of the time that the background is exposed. We denote the alpha values as $\alpha^i$ for the input blurred object on the left of Fig. 2, where $0 \leq \alpha^i \leq 1$. The corresponding unblurred opaque object, without motion, completely occludes the background, producing binary alpha values $\alpha^o$ where $\alpha^o \in \{0, 1\}$, as shown on the right of Fig. 2.

According to the above analysis, we summarize the relationship between the object transparency and the motion blur: The fractional structure of the alpha values on the motion blurred object is caused by a mixture of foreground alpha values ($\alpha^o = 1$) and background alpha values ($\alpha^o = 0$), similar to the color mixture in Eqn. 1 for each pixel. Put differently, *the motion blurred transparency is determined by the PSF and the unblurred object boundary shape*. Accordingly, we introduce a new formulation of motion blur from a transparency point of view, in contrast to the conventional Eqn. 1:

$$\alpha^i = \alpha^o \otimes f, \qquad (2)$$

where $f$ is the blur filter, similar to the one defined in Eqn. 1. We ignore the noise at this moment. In 1-D translational motion, $f$ is usually denoted as a $(2n+1)$-by-1 vector along motion direction

$$f = [f_{-n}, ..., f_0, ..., f_n]^T, \qquad (3)$$

where $f_{-n}$ and $f_n$ are nonzeros. In what follows, we assume that there is a single blurred object in center of the input image. The object is originally opaque and solid. Along motion direction, its width is also larger than the blur filter size $2n + 1$. We prove that using Eqn. 2, $f$ can be uniquely determined by $\alpha^i$.

**Lemma** *Define $\boldsymbol{\alpha}^i$ and $\boldsymbol{\alpha}^o$ as the alpha values in one scanline along the 1-D motion direction (the $x$ direction in Fig. 2) on the blurred and the unblurred objects respectively:*

$$\boldsymbol{\alpha}^i = [\alpha^i_0, \alpha^i_1, ..., \alpha^i_{m-1}]^T,$$
$$\boldsymbol{\alpha}^o = [\alpha^o_0, \alpha^o_1, ..., \alpha^o_{m-1}]^T,$$

*where $\alpha^i_0$ and $\alpha^i_{m-1}$ are the first and the last nonzero alpha values along motion direction. $m$ is the number of the pixels in $\boldsymbol{\alpha}^i$. Then $\boldsymbol{\alpha}^o$ can be expressed as*

$$\boldsymbol{\alpha}^o = [\underbrace{0, ..., 0}_{n}, \underbrace{1, ..., 1}_{m-2n}, \underbrace{0, ..., 0}_{n}]^T,$$

*where $n$ is defined in Eqn. 3.*

*Proof:* First, since the object is opaque and has solid boundary, $\boldsymbol{\alpha}^o$ can be written as $\boldsymbol{\alpha}^o$=$[0, ..., 0, 1, ..., 1, 0, ..., 0]^T$. We discuss the number of continuous 0's at the head of $\boldsymbol{\alpha}^o$ in the following two cases.

Case 1: If there are more than $n$ 0's at the head of $\boldsymbol{\alpha}^o$, we have $\alpha^o_j = 0$ for all $0 \leq j \leq n$. Then there exists $\alpha^i_0 = \sum_{j=0}^{n} \alpha^o_j f_{-j} = 0$ according to the convolution in Eqn. 2, contradicting our assumption that $\alpha^i_0 \neq 0$.

Case 2: If there are less than $n$ 0's at the head of $\boldsymbol{\alpha}^o$, there at least exists $\alpha^o_{n-1} = 1$. We denote the pixels before $\alpha^i_0$ and $\alpha^o_0$ along the motion direction as $\alpha^i_{-1}$ and $\alpha^o_{-1}$ respectively. Then $\alpha^i_{-1} = \sum_{j=0}^{n} \alpha^o_{j-1} f_{-j} \geq \alpha^o_{n-1} f_{-n} > 0$, also contradicting our assumption that $\alpha^i_0$ is the first nonzero alpha value.

Therefore, there are $n$ 0's at the head of $\boldsymbol{\alpha}^o$. Similarly, we can prove that there are also $n$ 0's in the tail of $\boldsymbol{\alpha}^o$.□

**Theorem 1** *In the 1-D object motion blurred image, each element of the blur filter can be explicitly expressed as*

$$f_j = \begin{cases} \alpha^i_{j+n} - \alpha^i_{j+n-1} & -n < j \leq n \\ \alpha^i_0 & j = -n \end{cases} \qquad (4)$$

*given the definition of $\boldsymbol{\alpha}^i$ and $\boldsymbol{\alpha}^o$ in the Lemma.*

*Proof:* Eqn. 2 can be reformulated into a matrix form [7]

$$\boldsymbol{\alpha}^i = F\boldsymbol{\alpha}^o, \qquad (5)$$

where $F$ is a square circulant matrix:

$$F = \begin{pmatrix} f_0 & \cdots & f_{-n+1} & f_{-n} & \cdots & 0 & f_n & \cdots & f_1 \\ f_1 & \cdots & f_{-n+2} & f_{-n+1} & \cdots & 0 & 0 & \cdots & f_2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f_{-2} & \cdots & 0 & 0 & \cdots & f_{n-1} & f_{n-2} & \cdots & f_{-1} \\ f_{-1} & \cdots & f_{-n} & 0 & \cdots & f_n & f_{n-1} & \cdots & f_0 \end{pmatrix}.$$

By solving Eqn. 5, we obtain

$$\alpha^i_j = \sum_{k=-n}^{j-n} f_k, \quad 0 \leq j \leq 2n. \qquad (6)$$

Computing $\alpha^i_j - \alpha^i_{j-1}$ using Eqn. 6 for different $j$, we obtain Eqn. 4. □

According to Theorem 1, the 1-D motion filter computation using transparency is precise and has a closed form solution. In the opaque center of the object, the alpha values do not need to be used in estimating the filter. The proof also shows that even if the 1-D translational motion speed of a rigid-body object varies during the image capture, we are still capable to faithfully reconstruct the blur filter.

## 3.2. 2-D object motion blur

In the object translational motion blur using a 2-D filter, we also assume that the object is originally solid and opaque. In this situation, because the unblurred object shape is unknown, the blur filter cannot be determined without computing the unblurred transparency map. We shall describe in the next section a probability framework to compute both the blur filter and the unblurred transparency map.

In the rest of this section, we show that using transparency, it is possible to automatically compute an upper bound for the width and the height of a 2-D filter, which, in most previous methods, is estimated by the user. For

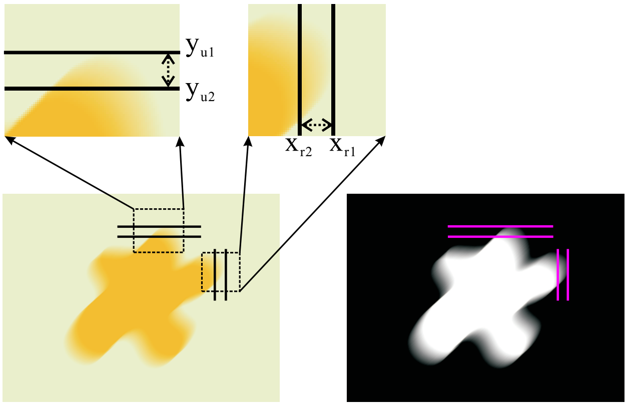

Figure 3. An upper bound for the size of the 2-D motion blur filter. The image shown on the left is the blurred object while the image on the right is the corresponding transparency map. $y_{u1}$ and $y_{u2}$ are the y-coordinates of the topmost pixels with alpha value $\alpha > 0$ and $\alpha = 1$ respectively. $x_{r1}$ and $x_{r2}$ are the x-coordinates of the rightmost pixels with alpha value $\alpha > 0$ and $\alpha = 1$ respectively.

simplicity's sake, we assume that the filter width and height are $2N + 1$ and $2M + 1$ respectively.

**Proposition** *Denote the rightmost pixel with alpha value $\alpha^i \neq 0$ and rightmost pixel with alpha value $\alpha^i = 1$ in the blurred transparency map as $(x_{r1}, y_{r1})$ and $(x_{r2}, y_{r2})$ respectively. Then $x_{r1} - x_{r2} + 1$ is a close upper bound for the width of the 2-D blur filter.*

*Proof:* Using the 2-D filter $f$, Eqn. 2 is written as

$$\alpha^i_{x,y} = \sum_{k=-N}^{N} \sum_{j=-M}^{M} f_{k,j} \alpha^o_{x-k,y-j}. \qquad (7)$$

We also define $(x_o, y_o)$ as the rightmost pixel such that $\alpha^o_{x_o,y_o} = 1$. Then $\alpha^o_{x'_o,y'_o} = 0$ for all $x'_o > x_o$. In the following, we first prove that $x_{r1} = x_o + N$.

If $x_{r1} > x_o + N$, we have

$$\alpha^i_{x_{r1},y_{r1}} = \sum_{k=-N}^{N} \sum_{j=-M}^{M} f_{k,j} \alpha^o_{x_{r1}-k,y_{r1}-j}. \qquad (8)$$

Since $x_{r1} - k \geq x_{r1} - N > x_o$, $\alpha^o_{x_{r1}-k,y_{r1}-j} = 0$ in Eqn. 8 for all possible $k$ and $j$. It follows that $\alpha^i_{x_{r1},y_{r1}} = 0$, contradicting our assumption that $\alpha^i_{x_{r1},y_{r1}} \neq 0$.

It is also easy to show that $x_{r1}$ cannot be smaller than $x_o + N$. If $x_{r1} < x_o + N$, there must exists $x_{r1'} > x_{r1}$ making $\alpha^i_{x_{r1'},y_{r1'}} > 0$, contradicting our assumption that $(x_{r1}, y_{r1})$ is the rightmost pixel with nonzero $\alpha$. So we conclude that $x_{r1} = x_o + N$.

Then, we prove $x_{r2} \leq x_o - N$. Since the width of the filter is $2N + 1$, there must exists element in the first column of the filter, for instance, $f_{-N,c}$, having value larger than 0. If $x_{r2} > x_o - N$, then $x_{r2} + N > x_o$. We get $\alpha^o_{x_{r2}+N,y_{r1}+c} = 0$ according to the definition of $x_o$. It follows that

$$\begin{aligned} \alpha^i_{x_{r2},y_{r2}} &= \sum_{k=-N}^{N} \sum_{j=-M}^{M} f_{k,j} \alpha^o_{x_{r2}-k,y_{r2}-j} \\ &< \sum_{k=-N}^{N} \sum_{j=-M}^{M} f_{k,j} \cdot 1 = 1, \end{aligned}$$

which contradicts our assumption that $\alpha^i_{x_{r2},y_{r2}} = 1$.

Combining $x_{r2} \leq x_o - N$ and $x_{r1} = x_o + N$, we get

$$x_{r1} - x_{r2} \geq -x_o + N + x_o + N = 2N. \qquad (9)$$

□

It can also be proven similarly that a close upper bound for the filter height is the vertical distance $|y_{u1} - y_{u2}| + 1$ between the topmost pixel $(x_{u1}, y_{u1})$ with alpha value $\alpha^i \neq 0$ and the topmost pixel $(x_{u2}, y_{u2})$ with alpha value $\alpha^i = 1$, as illustrated in Fig. 3.

## 4. Solving 2-D Object Motion Blur

We propose a MAP approach to recover the motion blur filter using transparency. Other image deblurring methods, after necessary modification on constraints, may also be applied to solving our problem due to the similar convolution operation. By Bayes' rule,

$$P(f, \alpha^o | \alpha^i) \propto P(\alpha^i | f, \alpha^o) P(\alpha^o) P(f). \qquad (10)$$

We define the likelihood

$$P(\alpha^i | f, \alpha^o) = \prod_{x,y} N(|\alpha^i_{x,y} - \sum_{i,j} \alpha^o_{x-i,y-j} f_{i,j}|; \sigma_1, \mu_1), \qquad (11)$$

where $N(.; \sigma, \mu)$ denotes a Gaussian distribution with mean $\mu$ and variance $\sigma^2$. $|\alpha^i_{x,y} - \sum_{i,j} \alpha^o_{x-i,y-j} f_{i,j}|$ measures the similarity between the input and the convolved alpha values using a blur kernel. The definition of prior $P(\alpha^o)$ is unique in our method. It consists of two parts

$$\begin{aligned} P(\alpha^o) &= \prod_{x,y} \exp(-\lambda \alpha^o_{x,y} |1 - \alpha^o_{x,y}|) \\ & \quad \prod_{\substack{(x,y) \\ (x',y')}} N(|\alpha^o_{x',y'} - \alpha^o_{x,y}|; \sigma_2, \mu_2), \qquad (12) \end{aligned}$$

where $(x', y')$ is the pixel neighboring $(x, y)$. In the first part, $\alpha^o_{x,y}|1 - \alpha^o_{x,y}|$ constraints that the unblurred alpha matte is a tow-tone image [15] where each alpha value should be either 0 or 1. The second part $N(|\alpha^o_{x',y'} - \alpha^o_{x,y}|; \sigma_2, \mu_2)$ is a Gaussian distribution, defined on the alpha difference between neighboring pixels. Since we assume that the object has solid boundary, the transparency

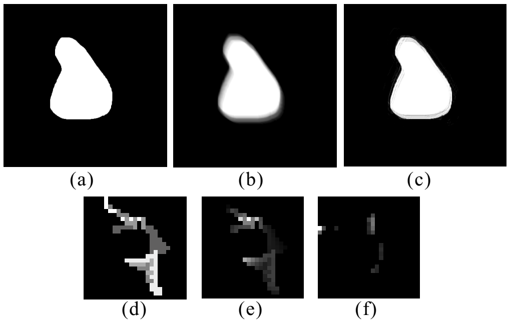

Figure 4. A toy example. (a) The unblurred binary alpha values. (b) The blurred alpha values using the filter shown in (d). (c) Our restored alpha values which is similar to the ground truth (a). (d) Ground truth filter. (e) Our estimated blur filter. (f) The estimated blur filter using the method in [4].

structure is simple. We generally define the prior for the elements of the filter as a uniform distribution. It can also be defined in other ways to encourage zero or other values.

We adopt an iterative optimization method to solve the above MAP problem. We first estimate the unknowns using conjugate gradient optimization and then apply Belief Propagation to further refine the structure of the unblurred alpha values. One toy example is illustrated in Fig. 4. (a) is the ground truth binary alpha matte which is blurred in (b) using the filter shown in (d). (c) shows our deblurred alpha matte, which is quite similar to (a) in structure. Our estimated blur filter is shown in (e) where the main structures are preserved well. (f) is the estimated blur filter using the method proposed in [4]. It is noted that the method in [4] is not designed to solve our problem, and is lack of specific constraints on transparency.

Our method, unlike previous approaches, does not take all pixels in the input image into blur filter estimation. It can be observed that, in an alpha matte, most alpha values are equal to 0 or 1. They do not provide useful information in the optimization process, and can thereby be omitted directly without influencing the result. In our approach, we simply use an alpha layer constructed by first selecting pixels with alpha value $0 < \alpha^i < 1$ and then dilating the selected area by a few pixels. With largely reduced alpha unknowns in the selected layer, the complexity of our optimization process is not very high even using a large size input image.

## 5. Generalized Transparency in Motion Blur

We have discussed how to robustly estimate the blur filter for the object motion using transparency. In this section, we address a more general problem: How to estimate the blur filter in an image blurred by camera shaking using transparency if the entire image is degraded?

To analyze, we first study the images illustrated in Fig. 5. (a) is a natural image without motion blur. It contains buildings with clear boundaries. The entire image in (b) is blurred using a 2-D linear filter. On the right of the figure, we show two sets of small patches extracted from the image (a) and the image (b), and compare them side-by-side in (d) and (e). Each patch contains two adjacent regions from the buildings and the sky respectively. In the unblurred patches, all boundaries are sharp while in the degraded patches in (d), the adjacent regions are blended to each other. For the boundary pixels, fractional alpha values can be computed similar to those computed on the motion blurred object! We show in Fig. 5(c) a blurred night-view image. The buildings are blended to the sky with variant transparency.

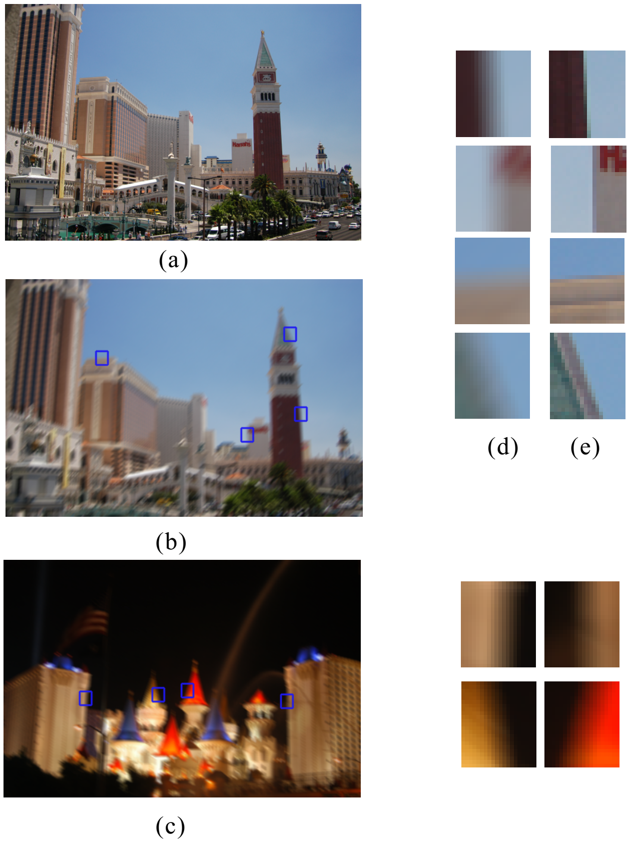

Figure 5. Generalized transparency. (a) is the input image without blur. (b) is the image degraded by a 2-D linear filter. On the right of the figure, (d) and (e) show two sets of amplified patches extracted from the images on the left. A side-by-side comparison is given on the boundary blending. (c) is another degraded night-view image due to the camera shaking. Similarly, the structures are blurred and the buildings are blended to the background sky.

The transparency definition in a camera motion blurred image regarding the region blending is different from that on a motion blurred object. Assuming that the moving object occludes static background behind, in the object motion blur, the transparency map can be uniquely determined. However, in general camera motion blur, if two distinct re-

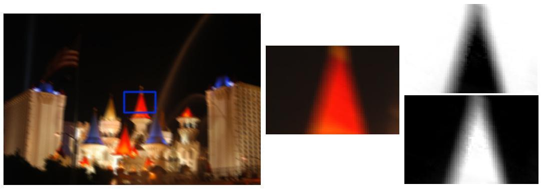

Figure 6. Generalized transparency. (a) Input image. (b) A patch extracted from (a) containing two regions: the roof of the building and the sky. Taking the sky as the foreground, we compute an alpha matte as shown on the top of (c). If the sky is regarded as the background, the corresponding alpha map is computed differently shown on the bottom of (c).

gions are adjacent, the foreground can be interpreted as either of them between the boundary. The new transparency map can be defined differently, so we call it the *generalized transparency*.

## 5.1. Solving the camera motion blur

To obtain a good estimation of the filter using transparency in a camera motion blurred image, it is interesting to show that not the entire image should be taken into computation. As discussed before, we only need to search patches containing distinct and adjacent regions, compute the alpha values on them, and apply the optimization method described in Section 4 to estimate the blur filter. There is no need, in our method, to perform the difficult filter estimation using all pixel colors in a natural image.

We show one example in Fig. 6. The patch (b) is extracted from the blurred image (a) containing two blended regions, i.e., the red roof of the building and the dark sky. Depending on whether we regard the sky region as the background or not, two different transparency maps can be computed, as shown in (c). We prove in the following that both of the representations are valid in our blur kernel estimation.

**Theorem 2** *Different foreground-background configurations to compute alpha values in the selected patches do not influence the 2-D linear shift-invariant filter estimation using the MAP formulation in Eqn. 10 in a camera motion blurred image.*

*Proof:* Suppose that $R_1$ and $R_2$ are the latent colors for two adjacent regions in one motion blurred patch respectively. They are blended on the region boundary. If we take $R_1$ as the foreground color, the alpha blending is expressed as

$$I = \alpha R_1 + (1-\alpha)R_2,$$

where $I$ is the corresponding pixel color in the blurred image. Similarly, defining $R_1$ as the background color, we have

$$I = (1-\alpha')R_1 + \alpha' R_2,$$

where $\alpha' = 1-\alpha$. We prove that in both cases, the posteriors defined in Eqn. 10 are proportional.

By substituting $\alpha^i = \alpha' = 1-\alpha$ into Eqn. 11, we get

$$\begin{aligned} P(\alpha'|f, 1-\alpha^o) &\propto \prod_{x,y} N(|(1-\alpha_{x,y}) - \sum_{i,j}(1-\alpha^o_{x-i,y-j})f_{i,j}|;\sigma_1,\mu_1) \\ &\propto \prod_{x,y} N(|\alpha_{x,y} - \sum_{i,j}\alpha^o_{x-i,y-j}f_{i,j}|;\sigma_1,\mu_1) \\ &\propto P(\alpha|f,\alpha^o). \end{aligned}$$

Also we have the prior

$$P(\alpha^o) = P(1-\alpha^o).$$

Combining the likelihood and prior, we obtain

$$\begin{aligned} P(f, 1-\alpha^o|\alpha') &\propto P(\alpha'|f, 1-\alpha^o)P(1-\alpha^o)P(f) \\ &\propto P(f,\alpha^o|\alpha). \end{aligned} \quad (13)$$

Therefore, we compute the same blur filter $f$ in both cases by introducing different binary alpha maps $\alpha^o$.□

## 5.2. Implementations

In this paper, we focus on analyzing the image blur and estimating the blur filter from a transparency perspective. We do not propose a new method to compute the transparency since there have been several successful natural image matting techniques proposed [3, 22, 14] which estimate foreground, background, and the alpha matte from a single input natural image. In our approach, the transparency is estimated using the closed form solution [14] due to the easy user input and the optimality of the solution given the satisfied color line models for both the background and the foreground.

In the filter estimation for an image containing a moving object, we use the user-drawn strokes to collect the foreground and the background samples. In the filter estimation for camera motion blurred images, we select the patches containing only two adjacent regions with distinct colors around the boundary. The matting algorithm working on these patches produces small alpha errors.

After estimating the blur filter, we either apply the Lucy-Richardson (L-R) method [7] to deconvolve the blurred image, or directly construct an inverse filter to restore the degraded image. We observe in experiments that even if the filter is known, using state-of-art image deconvolution algorithms [5], the artifacts, such as ringing, may still be introduced in the restored image.

We show one example in Fig. 7 where patches are selected containing adjacent regions with nearly constant colors close to the boundary. The computed alpha values are shown in (c). The image deblurring result is shown in (d) by using the L-R method.

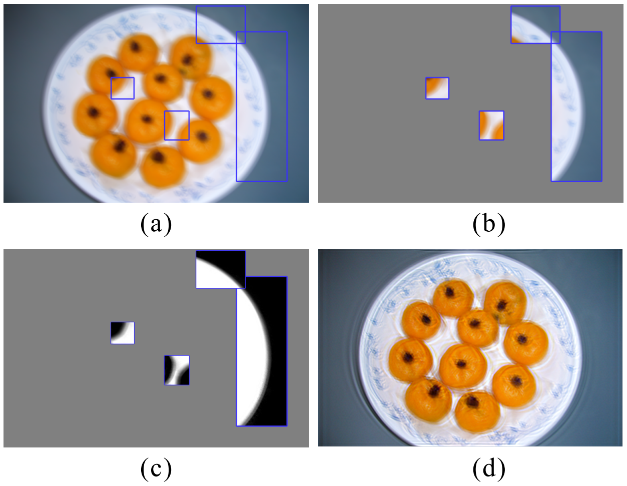

Figure 7. Example of the camera motion deblurring. (a) Input blurred image due to the camera shaking. We select patches containing distinct regions. (b) Some highlighted patches. (c) The computed generalized transparency for pixels in the patches. (d) We estimate the blur filter using the alpha values in patches, and deblur the input image using the Lucy-Richardson method.

## 6. Results

Two examples have already been illustrated in Fig. 4 and 7 respectively. In Fig. 8, we show a night-view deblurring example. The transparency is computed on the building boundary. We restore the input image using the Lucy-Richardson method.

In Fig. 9, an object motion deblurring example is shown. (a) is the input image containing a flying bird blurred due to its motion. (b) shows the strokes to collect the foreground and the background samples for alpha estimation. (c) and (d) are the computed alpha matte and the extracted blurred bird. The blur filter estimated on the transparency map is shown in (e). (f) is the restored alpha map. The boundary is clearer. We use the inverse filter of (e) to convolve image (d), and obtain the deblurred object (g). For comparison, we also show the deblurred image in (h) using the function 'deconvblind(·)' in Matlab for blind deconvolution.

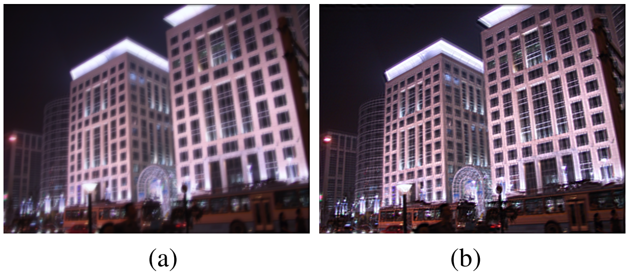

Figure 8. Night view example. (a) Blurred input image. (b) Restored image using our method.

Fig. 10 shows another image deblurring example. (a) is our input image where the blue rectangles highlight some selected patches to compute the alpha values. In each patch, there are adjacent regions with simple boundary shape. (b) shows the result from the method in [21]. (c) is our deblurred image using the estimated blur filter shown on the right.

## 7. Conclusion and Discussion

In this paper, we have proposed a novel approach to restore motion blurred images using transparency. Our approach is general, and is capable of being applied to restore both the camera motion blurred images and the object motion blurred images. We have proven that the blur filter in a 1-D motion can be uniquely determined by the transparency on the object boundary. Thus, solving the 1-D motion is a well-posed problem given precise alpha values. We have also shown that a close upper bound for the filter size in 2-D space can be estimated, and proposed an optimization method to compute the blur filter. The optimization can be neatly applied to solve the problem of camera motion blur by introducing the *generalized transparency*.

In discussion, our approach estimates the filter from the transparency. The accurate alpha estimation is crucial to the success of our method. When applying the state-of-art image deconvolution techniques, even if the filter is known, it is still common that the deconvolved image contains visual artifact. Our future work involves studying the integration of alpha matte and the kernel estimation, and applying our method to videos.

## References

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

[2] M. Chang, A. Tekalp, and A. Erdem. Blur identification using the bispectrum. *IEEE Transactions on Signal Processing*, 39(10):2323 – 2325, 1991.

[3] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. *CVPR, Vol. II, 264-271*, 2001.

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

[5] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, 2002.

[6] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. Society for Industrial and Applied Mathematic, 2006.

[7] A. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1988.

[8] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum. Bayesian correction of image intensity with spatial consideration. In *ECCV*, pages III: 342–354, 2004.

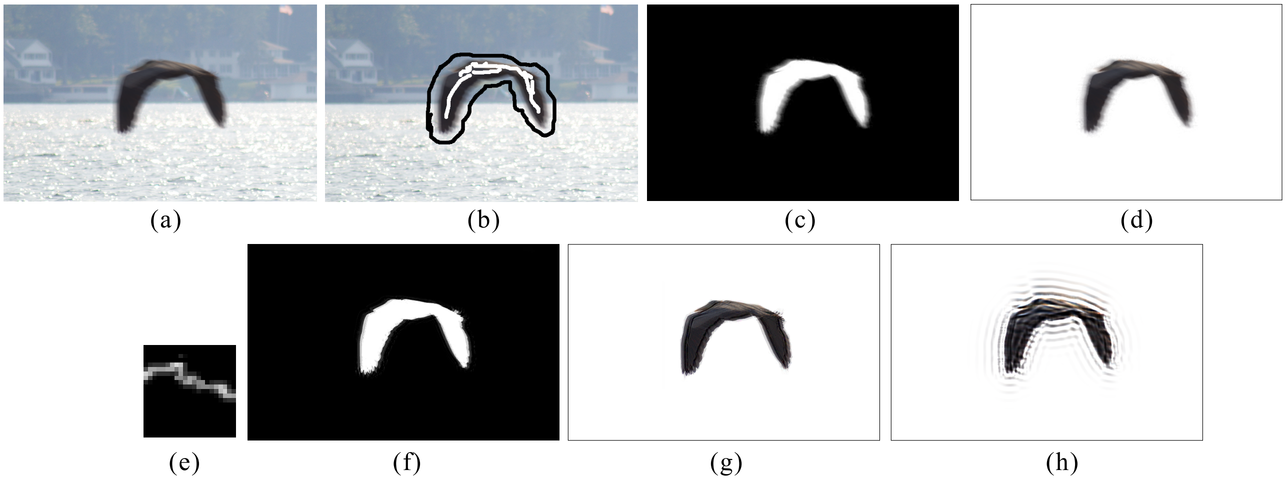

Figure 9. Example of object motion blur. (a) Input image containing a blurred flying bird. (b) The strokes to indicate foreground and background. (c) The computed alpha matte. (d) The extracted flying bird. (e) The estimated blur filter from our method. (f) The restored alpha matte. (g) The deblurred bird from our method. (h) The deblurred bird using 'deconvblind(·)' in Matlab.

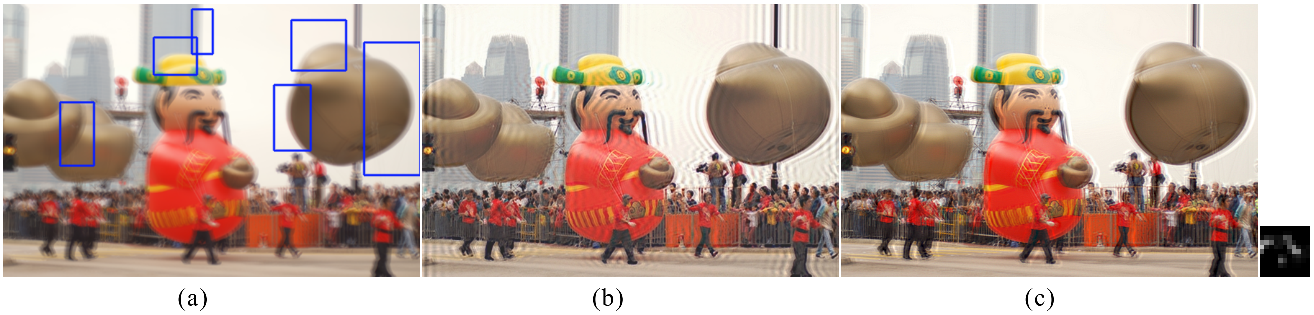

Figure 10. Example of camera motion blur. (a) The input blurred image. Patches are selected within the blue rectangles where the transparency values are computed. (b) The deblurred image from the method in [21]. (c) The deblurred image using our estimated blur filter shown on the right.

[9] S. K. Kim and J. K. Paik. Out-of-focus blur estimation and restoration for digital auto-focusing system. *Electronics Letters*, 34(12):1217 – 1219, 1998.

[10] D. Kundur and D. Hatzinakos. A novel blind deconvolution scheme for image restoration using recursive filtering. *IEEE Tran. on Signal Processing*, 46(2):375 – 390, 1998.

[11] R. Lagendijk, J. Biemond, and D. Boekee. Identification and restoration of noisy blurred images using theexpectation-maximization algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(7):1180 – 1191, 1990.

[12] R. Lagendijk, A. Tekalp, and J. Biemond. Maximum likelihood image and blur identification: a unifying approach. *Opt. Eng.*, 29(5):422 – 435, 1990.

[13] R. Lagendijk, A. Tekalp, and J. Biemond. Maximum likelihood image and blur identification: a unifying approach. *IEEE Sig. Processing Magazine*, 13(3):43 – 64, 1996.

[14] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *CVPR*, 2006.

[15] T.-H. Li and K.-S. Lii. A joint estimation approach for two-tone image deblurring by blind deconvolution. *IEEE Transactions on Image Processing*, 11(8):847– 858, 2002.

[16] X. Liu and A. Gamal. Simultaneous image formation and motion blur restoration via multiple capture. In *Int. Conf. Acoustics, Speech, Signal Processing*, 2001.

[17] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. *ACM Trans. Graph.*, 24(3):567–576, 2005.

[18] J. W. Miskin and D. J. C. MacKay. Ensemble learning for blind image separation and deconvolution. In *Advances in Independent Component Analysis*, July 2000.

[19] M. Ng, R. Plemmons, and S. Qiao. Regularized blind deconvolution using recursive inverse filtering. *Hong Kong Workshop on Scientific Computing*, pages 110 – 132, 1997.

[20] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring via fluttered shutter. *SIGGRAPH*, 25(3):795 – 804, 2006.

[21] A. Rav-Acha and S. Peleg. Two motion blurred images are better than one. *Pattern Recognition Letters*, 26:311–317, 2005.

[22] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *SIGGRAPH*, pages 315–321, 2004.