Alpha Matting of Motion-Blurred Objects in Bracket Sequence Images

Heesoo Myeong[1,⋆], Stephen Lin[2], and Kyoung Mu Lee[1]

[1] Department of ECE, ASRI, Seoul National University, Korea
[2] Microsoft Research, China

Abstract. We present a method that utilizes bracket sequence images to automatically extract the alpha matte of a motion-blurred object. This method makes use of a sharp, short-exposure snapshot in the sequence to help overcome major challenges in this task, including blurred object detection, spatially-variant object motion, and foreground/background color ambiguity. A key component of our matte estimation is the inference of approximate, spatially-varying motion of the blurred object with the help of the sharp snapshot, as this motion information provides important constraints on the aforementioned issues. In addition, we take advantage of other relationships that exist between a pair of consecutive short-exposure and long-exposure frames, such as common background areas and consistencies in foreground appearance. With this technique, we demonstrate successful alpha matting results on a variety of moving objects including non-rigid human motion.

Keywords: Alpha matting, motion blur, exposure bracketing.

1 Introduction

Matting aims to extract a foreground object from an image while accounting for fractional opacity values of foreground pixels. In most previous works on matting, the foreground object is assumed to be static, and fractional opacity arises from partial pixel coverage by the object, such that the pixel color is formed from a combination of foreground and background colors. A common example is a strand of hair, which is often too thin to fully occupy pixels, so that the pixels it appears in are shared with the background behind it. Another source of fractional opacity is object motion, in which a pixel is occupied by the foreground object for only part of the camera exposure period, and occupied by the background for the rest of the snapshot. We deal in this paper with matting of motion-blurred objects, which is a needed step in compositing such objects into other images, and for deblurring of moving objects as well. Important practical applications of matting motion-blurred objects also include video editing and post-production.

While matting techniques designed for static foregrounds can be applied to motion-blurred objects, this can lead to inadequate results as observed in previous works on motion deblurring [9,16]. This problem can be attributed in part

⋆ This work was done while Heesoo Myeong was an intern at Microsoft Research.

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 125–139, 2014.
© Springer International Publishing Switzerland 2014

Fig. 1. (a) and (b) are taken sequentially using the auto-bracket mode of a digital camera. (c) is the binary segmentation of the sharp foreground in (a). (d) is the alpha matte extracted by our method for the motion-blurred object in (b). The matte is computed automatically without any user guidance.

to blurred colors in the foreground region and broad areas of fractional opacity along the motion direction. Recently, a few matting methods have been proposed to specifically deal with motion-blurred objects. To help solve this problem, some of these methods require some form of external input, such as a user-supplied trimap indicating which regions are fully foreground and which are fully background [13], user-drawn scribbles that indicate motion paths of the foreground [22], or known blur kernels [10]. Obtaining this information, however, can itself be challenging or require much user assistance. Alternatively, other methods employ special hardware that obtains motion information to aid in motion-blurred object matting [14,20,21].

In this work, we propose a fully automatic technique for alpha matting of motion-blurred objects that does not require special equipment. We instead utilize bracket sequence imaging, a standard function in digital cameras where photos are taken consecutively at different exposure levels. Exposure bracketing yields images with blurred object motion at longer exposures, and also a short-exposure image in which moving foreground objects typically appear sharp. From computed blur-aware correspondences between a short and a long exposure image, as well as consistencies in their foreground and background, we derive approximate information on both the trimap for the blurred object and spatially variant blur kernels over the foreground region. Constraints formulated from this information are incorporated into a conventional alpha matting framework to more accurately extract the alpha mattes of motion-blurred objects.

Assumptions made in this work are that blurs from hand shake are negligible in the images, that the scale of the moving object changes little between the consecutively captured images, that the background is static, and that moving objects are opaque and contain no saturated pixels. With bracketed images captured under these conditions, our method is shown to generate alpha mattes comparable to those estimated with the help of considerable user interaction. In addition, reasonable performance is obtained in challenging cases such as non-rigid human motion and highly textured backgrounds, as demonstrated in Figure 1.

2 Related Work

A comprehensive survey of general matting techniques was presented in [24]. Here we review matting algorithms most closely related to our method, as well as relevant work on motion deblurring.

Motion-Blurred Object Matting. A few matting techniques explicitly account for the motion blur of a foreground object. In [13], a solution is computed by adding a regularization constraint to existing matting formulations [12,23] that suppresses the matte gradient along the local motion direction. Local motions are estimated from local gradient statistics, based on an assumption that image patches without motion blur have a uniform distribution of image gradients. The method does not detect blurred regions but instead requires a trimap to be provided by the user. By making use of a short-exposure image, our technique avoids the need for a user-supplied trimap, and estimates motion blur information without strong assumptions on local gradient statistics.

In [10], a known blur kernel is incorporated into closed-form matting [12] by modeling the alpha matte as the convolution of a foreground mask and the blur kernel. A trimap also needs to be provided by the user. Our work by contrast infers trimap and blur kernel information from a companion bracket sequence image, and also addresses spatially-variant blur for which blur kernels are difficult to obtain by existing automatic deblurring methods.

A different approach to this problem is to employ a hardware solution for recovering object motion. In [14,20,21], hybrid camera systems are used for this purpose, where one of the cameras captures multiple, unblurred, high-speed images of the scene while at the same time the other camera records a longer-exposure image with object motion blur. Optical flow in the high-speed sequence is used to recover motion trajectories in the longer-exposure image.

A coded exposure camera is utilized in [22] together with scribbles drawn by the user along motion paths of the foreground object. From the motion paths and coded-exposure image, blur kernels are estimated and then used with a sparsity constraint to deblur the matte obtained from closed-form matting [12]. This result is then re-blurred to obtain the alpha matte. Unlike coded exposure imaging, the exposure bracketing used in our work is widely available on digital cameras and does not introduce unsightly discontinuities into the motion blur.

Image Pairs. Previous techniques have utilized image pairs for unsupervised extraction of mattes. In [18], an image with flash and another without flash are taken, and a matte is computed based on the property that only the foreground is significantly brightened by flash, while the background is not if it is sufficiently distant. As this method requires accurate pixel alignment between the two images, the scene is assumed to be static. This approach was later extended to handle some misalignment due to hand shake and scene motion through an analysis of color histograms [19]. This extension deals with foreground segmentation instead of matting, which would be difficult to handle in this color histogram approach because of the changing color blends of moving foreground points with

fractional opacity. Like exposure bracketing, flash/no-flash image pairs can be captured by ordinary cameras, but a flash has a limited effect in daylight settings and for distant foreground objects.

In [25], a method is presented for co-matting, which is the joint matting of two images with the same foreground, different backgrounds, and similar matte properties. This method for static scenes would be difficult to extend to motion-blurred objects since the algorithm requires foreground alignment and assumes consistency between the mattes of the two images.

Similar to our work, a long-exposure and short-exposure image pair is captured as input in [26], which uses this data to estimate a spatially-invariant blur kernel to deblur the long-exposure image. The method deals with camera shake rather than object motion, and focuses mainly on using the short-exposure image to reduce deconvolution artifacts. The blur kernel is estimated using large-scale, sharp image features in the short-exposure image, which may not be sufficiently dense to recover spatially-varying blur.

A blurry photo is deblurred with the help of a sharp reference example in [8], which alternately computes locally aggregated correspondences between the two images, estimates blur kernels, and solves for the sharp image. In contrast to their work on deblurring of camera shake, ours deals with matting of motion-blurred objects, for which we compute globally optimized dense correspondences that cover the boundary of the object. Such an approach often yields a denser set of correspondences needed in our work to constrain the alpha matte.

Motion Deblurring. Given the alpha matte of a blurred object, the method in [4] estimates local motions of the object using a motion blur constraint in which the image gradient at a blurred point is equal to the difference between two non-blurred points along a linear blur direction. With the help of a short-exposure image, our method does not require a greatly simplified motion blur model.

Motion-blurred regions have been segmented from a single image based on their differences in intensity gradient distributions from the background [11] and additionally using a prior model on the local frequency components of sharp images [2]. These methods compute a hard segmentation of motion-blurred objects rather than a blur matte, and they obtain promising results for this challenging problem. In our work, we take advantage of an image that contains the foreground object without blur to more robustly detect and matte motion-blurred foreground objects.

3 Proposed Method

In this section, we present the proposed method for automatic alpha matting of a motion-blurred object in a bracket sequence image pair. Our approach consists of four main steps: (1) estimating a binary segmentation through integrating a foreground/background color model and a background probability model computed from sparse correspondences between the two images; (2) establishing dense correspondence between the moving object in both images while considering potential motion blur; (3) estimating spatially varying blur kernels for the

motion-blurred foreground object in the longer-exposure image with the help of the corresponding sharp object in the shorter-exposure image; (4) extracting an alpha matte of the motion-blurred object through an optimization that accounts for the binary segmentation, dense correspondences, and motion blur information. To improve the estimation of dense correspondences and blur kernels, the second and third steps are iterated until there is convergence in the correspondences. Each of the four steps is presented in the following.

3.1 Binary Segmentation

We start by computing a binary segmentation of the moving object in both of the images. After matching the exposure levels of the two images via the relative response function [5], the appearance of the background should be similar, as the two images are captured in rapid succession. We utilize this as a cue for segmenting the moving object.

We formulate foreground/background segmentation as a binary labeling problem. Given the two images, where the short-exposure image with a sharp foreground is denoted as I^{sharp} and the longer-exposure image as I^{blurry}, we define an energy function measuring the quality of binary segmentation of each image as

$$E(\mathbf{x}) = \sum_{i \in \mathcal{P}} D_i(x_i) + \sum_{(i,j) \in \mathcal{N}} V_{ij}(x_i, x_j), \tag{1}$$

where x_i is the binary label of pixel i, $\mathcal{P}$ denotes the set of all pixels, $\mathcal{N}$ denotes pairs of adjacent pixels, D_i is a data term for i, and V_{ij} is a pairwise term defined between two adjacent pixels i and j.

Data Term: The data term for each pixel consists of two parts:

$$D_i(x_i) = \gamma D_i^b(x_i) + D_i^c(x_i), \tag{2}$$

where $D_i^b(x_i)$ is the background likelihood term, $D_i^c(x_i)$ is the foreground/background color likelihood term, and γ is a weighting parameter set to 5 in our implementation. To compute the background likelihood term, we first obtain sparse correspondences between the two images using the SURF matching algorithm. The matches should primarily be of background points, since the colors and gradients of the foreground object become distorted by motion blur in I^{blurry}. The matched background points should be similar in appearance, and we model this similarity in terms of the intensity difference of pixel i between I^{sharp} and I^{blurry}:

$$\triangle I_i = \frac{h(I_i^{sharp}) - h(I_i^{blurry})}{h(I_i^{sharp}) + h(I_i^{blurry})}, \tag{3}$$

where $h(I_i)$ denotes the exposure normalized pixel value of I_i. Similar to [19], we model the distribution of these background pixel differences as a Gaussian $N(\triangle I_i | \mu, \sigma^2)$ with mean μ and standard deviation σ.

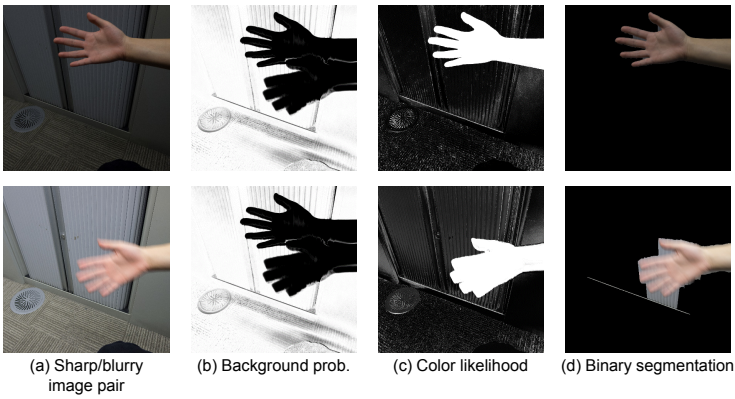

Fig. 2. Binary segmentation results of a short/long exposure image pair. (a) input image pair, (b) background probability, (c) foreground color likelihood, and (d) binary segmentation.

Then, we can define the probability $f_b(i)$ of a pixel i belonging to the background as

$$f_b(i) = exp(-\|\triangle I_i - \mu\|/\sigma). \tag{4}$$

Note that $f_b(i)$ lies within the range of [0,1]. If the difference of a certain pixel is far from μ, then we assign low background probability to that pixel. This probability is computed for every image pixel, using correspondences obtained by registering the two images via a homography computed from the sparse SURF matches. Figure 2 shows an example of the calculated background probability.

Finally, the background likelihood term $D_i^b(x_i)$ is defined as

$$D_i^b(x_i) = \begin{cases} 2\max\{f_b(i), 0.5\} - 1 \text{ if } x_i = 1 \\ 0 \text{ if } x_i = 0 \end{cases}. \tag{5}$$

This energy term assigns a penalty if the pixel i is labeled as foreground ($= 1$) and the background probability $f_b(i)$ is higher than 0.5. From this, the labels are determined using only the color and regularization terms when $f_b(i) < 0.5$.

The foreground/background color likelihood term $D_i^c(x_i)$ is calculated using the negative log likelihood of Gaussian mixture models, which are constructed from all pixels with $f_b(i) > 0.6$ and $f_b(i) < 0.4$, respectively.

Pairwise Term: The pairwise term is defined as the usual contrast dependent energy $V_{ij}(x_i, x_j) = |x_i - x_j| exp(-\beta||I_i - I_j||^2)$, with β is set to the inverse of the standard deviation of I.

Binary graph cut [1] is applied to minimize the objective function in Eq. (1). This is a standard formulation for using motion information to extract a foreground object [3,17].

The results of binary segmentation are shown for an example image pair in Figure 2. Since the boundary of the motion-blurred object is less distinct and has

inconsistent color likelihoods, the binary segmentation result is not as accurate for the blurred foreground as it is for the sharp object. It nevertheless provides a useful cue for matting, and we use it in conjunction with other cues soon to be described.

3.2 Blur-Aware Dense Correspondence Estimation

After binary segmentation of the foreground object in both images, we establish a dense correspondence between the two segmented regions. The blurred/sharp object regions are represented as graphs $\mathcal{G}$ and $\mathcal{G}'$, respectively, whose nodes represent image patches defined by a coarse image grid in each segment. For the patch I_p of node p in $\mathcal{G}$, we denote its associated local motion blur kernel as b_p. Similar to [6], we represent correspondences between $\mathcal{G}$ and $\mathcal{G}'$ as displacements within the graph grid, such that for a node p in $\mathcal{G}$ matched to a node p' in $\mathcal{G}'$, its correspondence is represented as displacement d_p, where $I_{p'} = I_{p+d_p}$.

We define an energy function that measures the quality of displacements as

$$E(d) = \sum_{p}^{N} D_p(d_p) + \sum_{p,q \in \mathcal{E}} P_{pq}(d_p, d_q), \tag{6}$$

where D and P denote unary and binary potentials, N is the number of nodes, and $\mathcal{E}$ is the set of neighboring node pairs. For computational considerations, we set a maximum displacement of K in each direction, which leads to a total of K^2 possible displacements d_p for each node n. We empirically set K to 30 and the patch size to 50.

The unary term D_p is defined as

$$D_p(d_p) = ||A_{(p+d_p)} * b_p - I_p^b||_2, \tag{7}$$

where $A_{(p+d_p)}$ denotes the patch in the sharp object graph $\mathcal{G}'$ with a displacement d_p from the patch of node p in $\mathcal{G}$. Note that this term accounts for the motion blur b_p of p's patch in measuring this match cost. The pairwise term is used to enforce spatial consistency, and is formulated as done in [6]:

$$P_{pq}(d_p, d_q) = P_{pq}^s(d_p, d_q) + P_{pq}^c(d_p, d_q). \tag{8}$$

The first term penalizes the L1 distance between two neighboring displacements:

$$P_{pq}^s(d_p, d_q) = -\lambda ||d_p - d_q||_1, \tag{9}$$

where λ is a positive constant set to 5 in our implementation. The second term penalizes instances where two neighboring displacements indicate a switch in the order of the two patches:

$$P_{pq}^c(d_p, d_q) = \begin{cases} -\mu[dx_q - dx_p]_+ \text{ if } x_q = x_p + 1 \text{ and } y_q = y_p \\ -\mu[dy_q - dy_p]_+ \text{ if } y_q = y_p + 1 \text{ and } x_q = x_p \\ 0 \; otherwise \end{cases}, \tag{10}$$

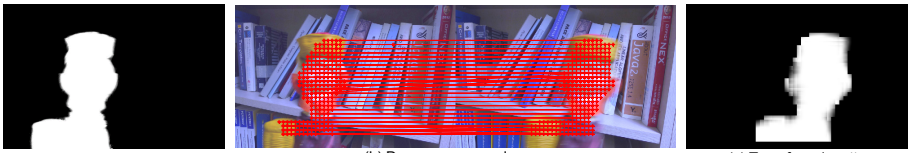

Fig. 3. Dense correspondence of patches and the transferred matte, which is transformed from the binary segmentation of the sharp object. (a) Binary segmentation of the sharp foreground object. (b) Blur-aware dense correspondence. (c) Transferred matte from the sharp object to the blurry object.

where μ is a positive constant set to 10 in our implementation, (x_p, y_p) are the grid coordinates of p's patch, and $[z]_+ = \max\{0, z\}$. Equation (6) is optimized using the curve-expansion method for solving dense graph matching problems.

This optimization can be time-consuming since the unary term in Equation (7) requires K^2 convolutions per patch in the motion-blurred object region. To reduce computation, instead of estimating correspondences for the blurred region patches, we first solve for correspondences from the sharp region patches, which need only one convolution to be computed per patch. Negative values of the computed displacements are then used as initial displacement estimates for the blurred region patches, and the displacement range is reduced from $K \times K$ to a small 5×5 neighborhood around the initial displacement estimate.

The first time that dense correspondence is computed, no blur kernel information is available, so we simply use uniform kernels as the initial kernels. After estimating blur kernels in the next step, this step is repeated with the updated kernels.

An example of dense patch correspondence is shown in Figure 3(b), where the red lines link corresponding patches in the two regions. With this correspondence, the binary segmentation of the sharp object (a) can be transformed into a transferred matte of the blurred object (c), by mapping the sharp patches using their corresponding displacements and applying the associated local motion blur kernels. We utilize this transferred matte as another cue in computing the final matte solution.

3.3 Spatially Varying Blur Kernel Estimation

For each blurred patch I_p^b, we now have its corresponding sharp reference patch $A_{(p+\hat{d}_p)}$. With each patch pair, the local blur kernel $\hat{b}_p$ for I_p^b is solved as follows:

$$\hat{b}_p = \text{argmin}_{b_p \geq 0} ||A_{(p+\hat{d}_p)} * b_p - I_p^b||^2, \tag{11}$$

where $\hat{d}_p$ denotes the optimized displacement vector from the dense correspondence estimation in Section 3.2. After obtaining the blur kernels $\hat{b}_p$ for all the patches, we replace each local blur kernel $\hat{b}_p$ by a weighted average of computed

blur kernels from neighboring patches to impose local smoothness among the blur kernels. The blur kernel estimation of Equation (11) and weighted averaging of neighboring blur kernels are iteratively repeated ten times. In our implementation, we used a kernel size of 31×31. In the iterative process, a weighted average of neighboring kernels is computed using equal weights among the ten nearest neighbor patches. We empirically found the iterative method to converge. The kernels of nearby points on a moving object tend to be similar, and we use this averaging as an easy way to apply this constraint. This regularization could alternatively be imposed in the objective, but would result in a large and complex optimization problem.

The converged blur kernels are used to re-estimate the dense correspondence in the previous step. The correspondence and blur kernel steps are iterated until the correspondences do not change.

3.4 Motion-Blurred Matte Estimation

Conventional image matting algorithms need user guidance to estimate accurate mattes. In our formulation, instead of user guidance we use the binary segmentation, blur-aware dense correspondence, and motion blur kernels computed in the previous steps to automatically guide our matte extraction of motion-blurred objects. Our method is built upon closed-form matting [12], whose energy function is defined as

$$\alpha = \text{argmin}\alpha^T L\alpha + \lambda(\alpha - \tilde{\alpha})^T D_S(\alpha - \tilde{\alpha}) \tag{12}$$

where $\tilde{\alpha}$ is a vector of specified alpha values (e.g., foreground and background pixels in a trimap); D_S is a diagonal matrix whose diagonal elements are set to 1 for pixels with alpha value constraints and set to 0 for unconstrained pixels; λ is a weight typically set to a large value; and L is the matting Laplacian matrix, whose (i, j)-th element is

$$\sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + (I_i - \mu_k)(\sum_k + \frac{\epsilon}{w_k} I_3)^{-1}(I_j - \mu_k))) \tag{13}$$

where $\sum_k$ is a 3×3 covariance matrix, μ_k is a 3×1 mean vector of the colors in a window w_k, and I_3 is the 3×3 identity matrix.

The binary segmentation provides information on a potential trimap for the motion-blurred object, by using morphological operations on the binary segmentation to determine the unknown region of the trimap. Our morphological kernel is set to half the blur kernel size. We specifically use the binary segmentation to obtain a partial trimap that only indicates definite foreground pixels through erosion of the segment. However, as shown in Figure 2, a binary segmentation may fail to accurately capture the boundary of the motion blurred object. The blur-aware dense correspondences and motion blur kernels can also provide a constraint on alpha values in the form of the transferred matte, computed by mapping the patches of the sharp region to the blurred region via the correspondences and then applying the associated blurs to the patches to obtain an alpha

Fig. 4. Effect of each matting component on the alpha matte result. (a) Original image. (b) Binary segmentation. (c) Alpha matting without the transferred matte and motion blur regularization. (d) Alpha matting without motion blur regularization. (e) Proposed method.

map. In addition, the alpha gradients can be constrained as in [13] according to the local motion direction and strength indicated by the blur kernels. Incorporating all of this information into closed-form matting gives us the following energy function:

$$\alpha = \text{argmin}\alpha^T L\alpha + \lambda(\alpha-\tilde{\alpha})^T D_S(\alpha-\tilde{\alpha}) + \mu(\alpha-\hat{\alpha})^T D_T(\alpha-\hat{\alpha}) + \sum_d w_d \nabla_d \alpha^T \nabla_d \alpha \tag{14}$$

where the second, third and fourth terms account for the partial trimap obtained from the binary segmentation, the transferred matte $\hat{\alpha}$ from dense correspondences and blur kernels, and the alpha gradient constraints, respectively. Denoting the partial trimap constraints from the binary segmentation as D_S and $\tilde{\alpha}$ as in Equation (12), we define D_T as a diagonal matrix whose diagonal elements are the intersection of D_S and all pixels with $\hat{\alpha} > 0.99$. $\nabla_d \alpha$ is the α-gradient in direction d, and w_d is the weight of regularization for direction d, where d is sampled in the eight directions of the eight-connected pixel neighbors. The motion strength w_d is obtained by convolving the local blur kernel with a linear motion in direction d.

The impact of each of these components on the alpha matting result is illustrated in Figure 4, which shows the original binary segmentation (b), the result using only the partial trimap computed from the binary segmentation (c), the result using the partial trimap and transferred matte (d), and the result using all the components (e). Improvements are obtained by adding each component into the optimization.

4 Results

We evaluated our method on both synthetic and real bracket sequence pairs. The real bracket sequences were acquired using a Panasonic DMC-LX5 camera, and the synthetic pairs were produced using imagery from this camera as well. We determined the parameters of our algorithm empirically and fixed all of them throughout the experiments.

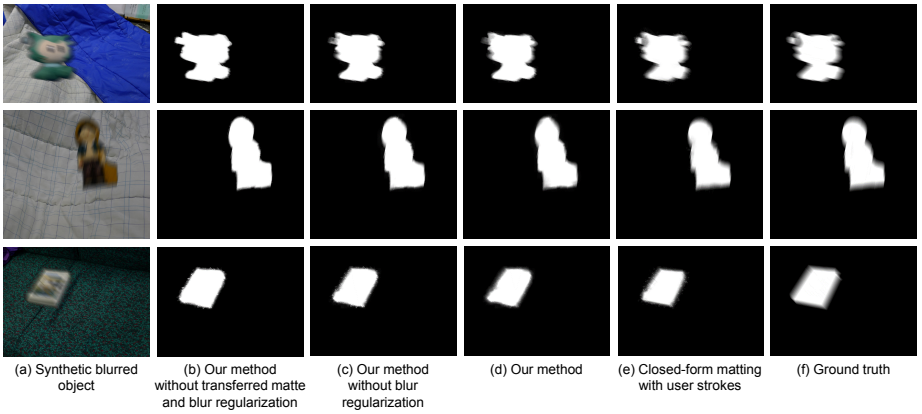

Fig. 5. Alpha matting results on synthetic data. (a) Synthesized image with motion-blurred object. (b) Motion matting without matte transfer and motion blur regularization. (c) Motion matting without motion blur regularization. (d) The proposed motion matting method combining all cues. (e) Closed-form matting with user strokes. (f) Ground truth alpha matte.

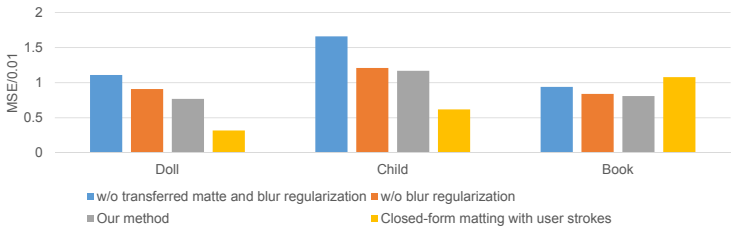

Fig. 6. Quantitative error comparisons. Doll, Child and Book sequences correspond to the first, second and third rows of Figure 5.

4.1 Synthetic Images

We first validate our method on synthetic data, from which ground truth is available for quantitative evaluation. To generate a bracket pair, we first capture two sharp images with the same short-exposure setting: one of a static foreground object in front of a background, and another of just the background. The first image is used as the short-exposure image in the bracket pair. We also apply GrabCut [15] with user strokes to it to extract the foreground object from the background, and apply a known motion blur to this foreground object region. The motion-blurred object is then composited on the second image using the corresponding alpha matte, and a gain is added to the resulting image to synthesize the long-exposure image.

The performance of our proposed approach is exhibited with three such image pairs in Figure 5. To show the contribution of each term in motion-blurred object

matting, we remove the transferred matte term and blur regularization term in computing Figure 5(b), remove only the blur regularization term for Figure 5(c), and show our full results in Figure 5(d). The uncertainty in binary segmentation along the boundary of a motion-blurred object is resolved by matte transfer and blur regularization. The corresponding mean squared error with respect to the ground-truth matte decreases with the addition of each cue as shown in Figure 6. We also compare our automatic image matting to human-guided closed-form matting in Figure 5(e). Though the quantitative error is greater for our results, they are visually similar to the user-guided matte and are computed without any human interaction. In cases like the book example where the background is highly textured, closed-form matting with user strokes may actually perform worse due to color ambiguities along the boundary, while motion information provides a stronger constraint that helps to resolve such uncertainties.

Our unoptimized implementation was coded in MATLAB with embedded C++ functions on a PC with a 3.3GHz Intel quadcore i5 CPU and 16GB RAM. We resize images to 1024×577 for efficiency. For the Doll sequence, the overall run time was about 12 minutes, and peak memory usage was 1 GB.

4.2 Real Images

We also tested our method on a variety of real bracket image pairs. Similar to the results for synthetic images, we compare our method to a version of it without matte transfer and motion blur regularization. The first two rows of Figure 7 contain non-rigid human motion, which is effectively handled by our approach. In the second example, the detailed blur of each finger is better extracted with the help of matte transfer and motion blur regularization. The third and fourth rows are challenging cases of textured foreground objects moving in front of highly textured backgrounds.

5 Discussion

The method of HaCohen *et al.* [8] also computes non-rigid dense correspondences between a blurred and sharp image, using the method of [7] interleaved with kernel estimation and deconvolution. Correspondences computed using only [7] (without iterations of kernel estimation and deconvolution) are shown in Figure 8. The correspondences are less dense than ours, perhaps because their bottom-up aggregation of consistent patches may fail in textured areas. The correspondence density is nevertheless sufficient for the purpose of non-uniform deblurring of camera shake, and would likely be increased by interleaving kernel estimation and deconvolution as done in [8]. By contrast, our alpha matting approach requires denser correspondences for computing the transferred matte and the motion blur regularization constraint along the matte boundary, so the proposed method is designed to compute a much denser correspondence that better serves our purpose.

A limitation of the proposed method is that its performance largely depends on the accuracy of binary segmentation for the sharp foreground object. Errors

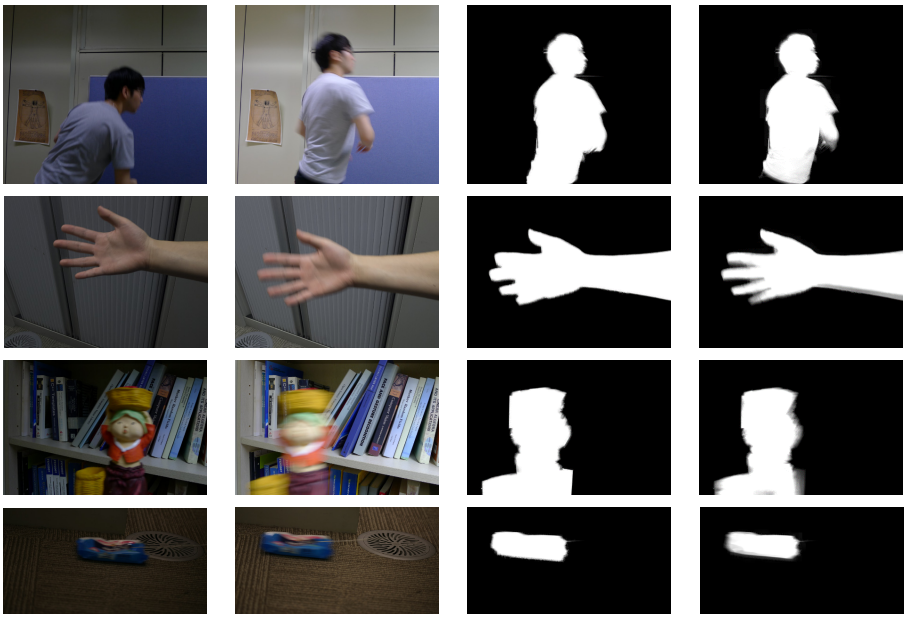

Fig. 7. Alpha matting results of our method on real bracket pairs

in that binary segmentation will degrade the quality of the transferred matte, and may affect the amount of blur kernel information computed. This binary segmentation could potentially be refined in future work by re-estimating it in the matting pipeline using inter-image information. Another limitation is that our current method cannot be applied if the foreground in the short-exposure image is also motion-blurred.

Fig. 8. Differences in correspondence density between our method and NRDC [7]. Our method establishes non-rigid dense correspondence between two segmented region pairs (b). For the motion-blurred object region, (c) shows the matched patches with NRDC, and (d) shows a thresholded version of our transferred matte.

6 Conclusion

In this paper, we presented a method to automatically extract the alpha matte of a motion-blurred object using inter-image information in a bracket sequence pair. To effectively share information between images, we presented a blur-aware dense matching technique, which enables our method to determine a transferred matte and infer dense local blur kernels. Using this information together with a binary segmentation of the motion-blurred object region, our method can generate reasonable matte results without user guidance.

In future work, we plan to extend our technique to matte multiple foreground objects in a dynamic scene. Our current implementation assumes that only one moving object is present in the bracket sequence. Furthermore, scale and orientation change is not considered in our current pipeline. This issue could be addressed through an orientation-aware scale-space matching scheme.

References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 1222–1239 (2001)
2. Chakrabarti, A., Zickler, T., Freeman, W.T.: Analyzing spatially-varying blur. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2010)
3. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2006)
4. Dai, S., Wu, Y.: Motion from blur. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2008)
5. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: ACM SIGGRAPH (1997)
6. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: Proc. Int'l Conf. on Computer Vision, ICCV (2011)
7. HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D.: Nrdc: Non-rigid dense correspondence with applications for image enhancement. In: ACM Trans. Graph (2011)
8. HaCohen, Y., Shechtman, E., Lischinski, D.: Deblurring by example using dense correspondence. In: Proc. Int'l Conf. on Computer Vision, ICCV (2013)
9. Jia, J.: Single image motion deblurring using transparency. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2007)
10. Köhler, R., Hirsch, M., Schölkopf, B., Harmeling, S.: Improving alpha matting and motion blurred foreground estimation. In: Proc. Int'l Conf. on Image Processing, ICIP (2013)
11. Levin, A.: Blind motion deblurring using image statistics. In: NIPS (2006)
12. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. 30(2), 228–242 (2008)
13. Lin, H.T., Tai, Y.W., Brown, M.S.: Motion regularization for matting motion blurred objects. IEEE Trans. Pattern Anal. Mach. Intell. 33(11), 2329–2336 (2011)
14. Nayar, S.K., Ben-Ezra, M.: Motion-based motion deblurring. IEEE Trans. Pattern Anal. Mach. Intell. 26(6), 689–698 (2004)

15. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
16. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: Proc. Int'l Conf. on Computer Vision, ICCV (2007)
17. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: Proc. Int'l Conf. on Computer Vision, ICCV (2009)
18. Sun, J., Lin, Y., Kang, S.B., Shum, H.Y.: Flash matting. ACM Trans. Graph. 25(3), 772–778 (2006)
19. Sun, J., Sun, J., Kang, S.B., Xu, Z.B., Tang, X., Shum, H.Y.: Flash cut: Foreground extraction with flash and no-flash image pairs. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2007)
20. Tai, Y.W., Du, H., Brown, M.S., Lin, S.: Image/video deblurring using a hybrid camera. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2008)
21. Tai, Y.W., Du, H., Brown, M.S., Lin, S.: Correction of spatially varying image and video motion blur using a hybrid camera. IEEE Trans. Pattern Anal. Mach. Intell. 32(6), 1012–1028 (2010)
22. Tai, Y.W., Kong, N., Lin, S., Shin, S.Y.: Coded exposure imaging for projective motion deblurring. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2010)
23. Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2007)
24. Wang, J., Cohen, M.F.: Image and Video Matting. NOW Publishers Inc. (2008)
25. Wang, L., Xia, T., Guo, Y., Liu, L., Wang, J.: Confidence-driven image co-matting. Computers & Graphics 38(2), 131–139 (2013)
26. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. ACM Trans. Graph. 26(3) (2007)