Optical Flow in the Presence of Spatially-Varying Motion Blur

Travis Portz Li Zhang Hongrui Jiang
University of Wisconsin–Madison

http://pages.cs.wisc.edu/~lizhang/projects/blurflow/

Abstract

This paper extends the classical warping-based optical flow method to achieve accurate flow in the presence of spatially-varying motion blur. Our idea is to parameterize the appearance of each frame as a function of both the pixel motion and the motion-induced blur. We search for the flows that best match two consecutive frames, which amounts to finding the derivative of a blurred frame with respect to both the motion and the blur, where the blur itself is a function of the motion. We propose an efficient technique to calculate the derivatives using pre-filtering. Our technique avoids performing spatially-varying filtering (which can be computationally expensive) during the optimization iterations. In the end, our derivative calculation technique can be easily incorporated with classical flow code to handle video with non-uniform motion blur with little performance penalty. Our method is evaluated on both synthetic and real videos and outperforms conventional flow methods in the presence of motion blur.

1. Introduction

Optical flow has many important applications in computer vision including video denoising, deblurring, stabilization, and structure from motion. In these applications, optical flow needs to be computed on videos with camera motion and/or target motion. These types of videos often contain a significant amount of spatially-varying motion blur, which makes it difficult for traditional optical flow algorithms to produce accurate results since they make assumptions about data conservation between frames.

One motivation of this work is that in the last decade, people have tried to combine frames in a sequence with different exposures such as for HDR video [14], image deblurring [30], and video denoising [21]. Solving these problems requires accurate alignment between frames. It is unrealistic to ignore motion blur in the context of imaging dynamic scenes with variable exposure.

This paper studies the problem of how to achieve accurate optical flow in the presence of spatially- varying motion blur. Our work also takes variable exposure times into account and aims to provide consistent flow between blurred and sharp frames as well as between blurred frames with different blur kernels. This problem is especially challenging due to the difficulty of estimating spatially-varying blur in an efficient manner.

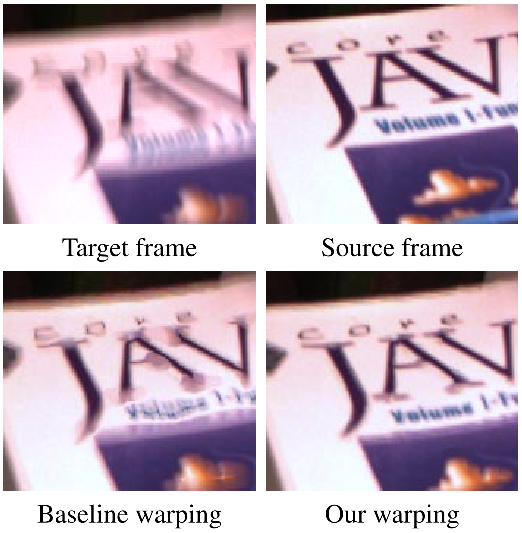

Figure 1. We present a blur-aware optical flow algorithm that provides more accurate and consistent flow than the baseline classical flow method [5, 17] in the presence of motion blur. Using the flow estimated between the top image pair by our method, the warping from the source to the target better preserves the image structure. In contrast, the baseline flow method tries to move pixels in the sharp images to match the blurry appearance of the target.

We approach the problem by extending the classical warping-based optical flow method. Our idea is to parameterize the appearance of each frame as a function of both the pixel motion and the motion-induced blur. Using the metric proposed in [13], we search for the flows that best match two consecutive frames. The search amounts to finding the derivative of a blurred frame with respect to both the motion and the blur; the blur itself is a function of the motion. We propose an efficient technique to calculate the derivatives using pre-filtering. Our technique avoids performing spatially-varying filtering (which can be computationally expensive) during the optimization iterations. In the end, our derivative calculation technique can be easily incorporated with classical flow code to handle video with non-uniform motion blur with little performance penalty.

We also develop error metrics for evaluating optical flow in the presence of motion blur. Standard optical flow error metrics depend on a ground truth flow field for calculation; however, a pixel in one image that is blurred in another has multiple correspondences in the other image, as observed in

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

[22]. This makes the notion of a ground truth flow ambiguous. Our proposed error metrics handle this flow ambiguity.

2. Related work

Optical flow has seen many improvements over the past few decades. A full review is beyond the scope this paper. We refer the readers to the previous work sections in [1, 26] for excellent reviews of recent work in optical flow. For older work, see [2]. Despite the multitude of work on optical flow in general, there is very little in the literature that addresses the problem of spatially-varying motion blur directly. Filter flow [24] is capable of modeling optical flow with blur, and [11] uses it to deblur image sequences. However, its ability to produce accurate flow fields for motion blurred sequences has not been demonstrated.

Sellent *et al.* [25] capture videos with alternating short exposures and long exposures and use the motion blur of the long-exposure frames to more accurately estimate motion between the short-exposure frames. Their algorithm requires the alternating exposure scheme with short delays between frames. We do not place any such constraints on the exposure times of the frames and also address flow between two blurred frames.

Liu and Sun [18] compute flow between high-resolution and low-resolution images for video super resolution. They estimate a spatially-invariant smoothing kernel for the low-resolution images and apply it to the high-resolution image in a warping-based flow context.

He *et al.* [10] perform motion estimation for spatially-varying blur by detecting corners and using hierarchical block matching to obtain flows for the corners. Flows for other pixels are obtained by interpolating between the available flows in a sparse-to-dense approach.

Ben-Ezra and Nayer [3] propose a hybrid camera system where a secondary detector captures a sequence of short-exposure frames whose optical flows are used to estimate a spatially-invariant blur kernel for the single long-exposure image from the primary detector.

Yuan *et al.* [29] perform alignment between motion blurred and sharp image pairs. Their work assumes a spatially-invariant blur kernel and an affine image transform. HaCohen *et al.* [9] compute dense correspondence between motion blurred and sharp image pairs by interleaving kernel estimation and deblurring within the correspondence algorithm. Their work also assumes a spatially-invariant blur kernel and only computes the correspondence within a validity mask.

Our work is related to works on tracking in the presence of motion blur [13, 16, 19, 20]. Of particular importance are [13, 19], which use the commutativity of the blur operation to match blurred images and avoid deblurring. These works track individual blurred patches between images, whereas we compute dense optical flow.

Our work is also related to motion from blur [7, 15, 23], where the motion blur in a single image is used to estimate the motion. These algorithms only provide the direction and speed of motion at the time the image was captured, which is unlikely to provide accurate correspondences between image pairs with arbitrary motion.

3. Baseline method

We use "classical" warping-based optical flow as a baseline method [4, 5, 6, 12]. The problem is formulated as an energy minimization problem with the following energy functional:

$$E(\mathbf{w}) = E_{\text{D}}(\mathbf{w}) + \alpha E_{\text{S}}(\mathbf{w}), \tag{1}$$

where $\mathbf{w} = (u, v)$ is a flow vector, E_{D} is a data conservation term, E_{S} is a smoothness term, and α is a regularization parameter. The data conservation term provides a brightness constancy constraint along the flow:

$$E_{\text{D}}(\mathbf{w}) = \sum_{\mathbf{x}} \psi_{\text{D}} \left((I_2(\mathbf{x} + \mathbf{w}) - I_1(\mathbf{x}))^2 \right), \tag{2}$$

where ψ_{D} is a penalty function and we are computing the flow from image I_1 to image I_2. In practice, we filter the input images to reduce the effects of illumination changes.

The smoothness term has the form

$$E_{\text{S}}(\mathbf{w}) = \sum_{\mathbf{x}} \psi_{\text{S}} \left(|\nabla u|^2 + |\nabla v|^2 \right) \tag{3}$$

where ψ_{S} is again a penalty function. The smoothness term helps to account for the aperture problem (the data term provides no motion information in regions without texture) and also helps to prevent outliers.

As in [5], we use the robust penalty function

$$\psi(x^2) = \sqrt{x^2 + \epsilon^2} \tag{4}$$

for both the data and smoothness terms. The constant ϵ makes the function differentiable, which helps with the minimization. We use $\epsilon = 0.001$.

3.1. Energy minimization

Minimizing the energy functional is achieved using a gradient-based method in a coarse-to-fine fashion. If we have a current flow estimate $\mathbf{w} = (u, v)$, we can write our next estimate as $(u + du, v + dv)$. The problem can then be solved iteratively by linearizing about (u, v) and finding the increment (du, dv) that minimizes the linear energy functional. The linearized data term has the form:

$$E_{\text{D}}(\mathbf{w}) = \sum_{\mathbf{x}} \psi_{\text{D}} \left((I_z + I_x du + I_y dv)^2 \right), \tag{5}$$

where

$$I_z = I_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - I_1(\mathbf{x}), \tag{6}$$

$$I_x = \frac{\partial I_z}{\partial u} = \left. \frac{\partial I_2}{\partial x} \right|_{\mathbf{x}+\mathbf{w}(\mathbf{x})}, \tag{7}$$

$$I_y = \frac{\partial I_z}{\partial v} = \left. \frac{\partial I_2}{\partial y} \right|_{\mathbf{x}+\mathbf{w}(\mathbf{x})}. \tag{8}$$

As shown in [17], the minimization of the linearized energy functional can be formulated as an iterated reweighted least squares problem. Taking the derivative of the energy and setting it equal to zero gives the following:

$$\begin{bmatrix} \boldsymbol{\psi}'_{\text{D}}\mathbf{I}_x^2 + \alpha\mathbf{L} & \boldsymbol{\psi}'_{\text{D}}\mathbf{I}_x\mathbf{I}_y \\ \boldsymbol{\psi}'_{\text{D}}\mathbf{I}_x\mathbf{I}_y & \boldsymbol{\psi}'_{\text{D}}\mathbf{I}_y^2 + \alpha\mathbf{L} \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\psi}'_{\text{D}}\mathbf{I}_x I_z + \alpha\mathbf{L}u \\ \boldsymbol{\psi}'_{\text{D}}\mathbf{I}_y I_z + \alpha\mathbf{L}v \end{bmatrix}, \tag{9}$$

where bolded terms indicate diagonal matrix forms of the images, non-bolded terms indicate vector forms, and $\mathbf{L}$ is the weighted Laplacian filter using ψ'_{S} as weights. The current estimate of (u, v) is used to calculate the weights ψ'_{D} and ψ'_{S}. The matrix on the left-hand side is positive definite, so we can use the conjugate gradient method to solve the system and avoid actually forming the matrix [17].

4. Blur-aware flow algorithm

When one or both of the images used in the flow calculation contain motion blur, the brightness constancy (and/or gradient constancy) of the data term will be violated. In its attempt to minimize the error of the data term, the baseline algorithm will try to warp the source image to match the appearance of the target image. This will produce inaccurate flow in regions where the two images do not have the same blur. Our algorithm modifies the data term to obtain higher quality flow by matching the non-uniform motion blur between the two images.

Let us consider two ground truth images J_1 and J_2 where J_2 is a warped version of J_1 ignoring occlusion and lighting changes, i.e., $J_1(\mathbf{x}) = J_2(\mathbf{x} + \mathbf{w}(\mathbf{x}))$. Then, let $I_1 = J_1 * k_1$ and $I_2 = J_2 * k_2$ be the observed images (minus noise) where k_1 and k_2 are spatially-varying motion blur kernels. Based on the tracking work of [13], we apply the blur of each observed image to the other image to obtain

$$\begin{aligned} B_1(\mathbf{x}) &= (I_1 * k_2)(\mathbf{x}) \approx (J_1 * k_1 * k_2)(\mathbf{x}), \\ B_2(\mathbf{x}) &= (I_2 * k_1)(\mathbf{x}) \approx (J_2 * k_2 * k_1)(\mathbf{x}). \end{aligned} \tag{10}$$

Note that the blur kernels are themselves functions of $\mathbf{x}$, but we omit the function notation to avoid confusion with the evaluation of a single element in the kernel. If the kernels are spatially invariant, then the two convolutions commute. We do not assume they are spatially invariant, so the commutative property does not hold in general. However, we do assume that the motion is locally smooth, so we have $B_1(\mathbf{x}) \approx B_2(\mathbf{x} + \mathbf{w}(\mathbf{x}))$. Now, in place of I_z from the baseline method, we have

$$B_z(\mathbf{x}) = B_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - B_1(\mathbf{x}). \tag{11}$$

Since there is a connection between the optical flow and the motion blur, the kernels used to form B_1 and B_2 may vary with u and v. Therefore, we *do not* have $\partial B_z/\partial u = B_x$ and $\partial B_z/\partial v = B_y$ as in Eq. (7) and Eq. (8). Our main contribution is to adapt the data term to handle this connection between the flow and the blur.

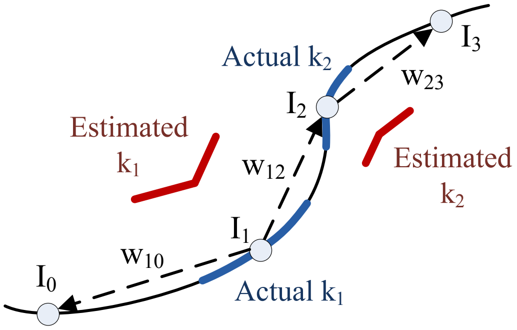

Figure 2. We approximate the true motion blur kernels with piecewise linear kernels obtained from flows to neighboring frames.

Again, we write $\mathbf{w} = (u + du, v + dv)$ and linearize about du and dv. The new data term is

$$E_{\text{D}}(du, dv) = \sum_{\mathbf{x}} \psi_{\text{D}} \left(\left(B_z + \frac{\partial B_z}{\partial u} du + \frac{\partial B_z}{\partial v} dv \right)^2 \right) \tag{12}$$

where

$$\begin{aligned} \left.\frac{\partial B_z}{\partial u}\right|_{\mathbf{x}} &= \frac{\partial}{\partial u}\Big(B_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - B_1(\mathbf{x}) \Big) \\ &= \left[\frac{\partial B_2}{\partial u} + \frac{\partial B_2}{\partial x} \right]_{\mathbf{x}+\mathbf{w}(\mathbf{x})} - \left[\frac{\partial B_1}{\partial u} \right]_{\mathbf{x}}. \end{aligned} \tag{13}$$

The partial derivative $\partial B_2/\partial u$ represents how the blurred image changes with respect to the blur induced by the flow, and the partial derivative $\partial B_2/\partial x$ represents how the value $B_2(\mathbf{x} + \mathbf{w}(\mathbf{x}))$ changes with respect to the flow for a fixed blur. The partial derivative with respect to the vertical flow, $\partial B_z/\partial v$, is similarly defined.

We use the same minimization approach as the baseline method, but with the new blurred image derivatives, B_z, $\partial B_z/\partial u$, and $\partial B_z/\partial v$, in place of the standard image derivatives, I_z, I_x, and I_y. It remains to show how we obtain the blurred images and their derivatives.

4.1. Blur kernel approximation

Motion blur kernels can be very complex for long exposure times as shown in [8]. On the other hand, exposure times in video are limited by the frame rate. Therefore, the motion observed during the exposure of a frame is relatively smooth and a piecewise linear approximation, as shown in Figure 2, is reasonable to represent the motion trajectory over multiple frames. This approximation helps in the case of spatially-varying blur since estimating an arbitrary kernel for each individual pixel without assuming linear motion between frames would be prohibitively expensive.

Let us consider a sequence of four frames I_0, I_1, I_2 and I_3, with each possibly containing motion blur, and suppose we want accurate flow from I_1 to I_2. The blur kernels used in our flow method are approximated using precomputed baseline flows $\mathbf{w}_{10}$ and $\mathbf{w}_{23}$ and the current estimate of the desired flow $\mathbf{w}_{12}$ as shown in Figure 2. Let $k_{\mathbf{w}}$ denote a blur kernel consisting of a line segment from the origin to $\mathbf{w}$. Then the approximated kernels are given by

$$\begin{aligned} k_1(\mathbf{x}) &= \tfrac{1}{2}\left(k_{\mathbf{w}_{10}(\mathbf{x})\tau_1/2} + k_{\mathbf{w}_{12}(\mathbf{x})\tau_1/2}\right), \\ k_2(\mathbf{x}+\mathbf{w}_{12}(\mathbf{x})) &= \tfrac{1}{2}\left(k_{-\mathbf{w}_{12}(\mathbf{x})\tau_2/2} + k_{\mathbf{w}_{23}(\mathbf{x}+\mathbf{w}_{12}(\mathbf{x}))\tau_2/2}\right), \end{aligned} \tag{14}$$

where τ_i is the duty cycle of frame i.

4.2. Computation of blurred image derivatives

We could theoretically compute the blurred images by performing line integrals at each pixel as in [19] and evaluate the derivatives with respect to the flow by differentiating the parametric kernels. For example,

$$\left.\frac{\partial B_1}{\partial u}\right|_{\mathbf{x}} = \left(I_1 * \frac{\partial k_2}{\partial u}\right)(\mathbf{x}). \tag{15}$$

However, differentiating the kernel at each pixel and convolving it with the input image would be very inefficient, especially because this spatially-varying blur needs to be executed in each iteration of the optimization. Instead, we precompute a grid of blurred images where each grid point corresponds to a scaled flow vector $(\tau_i/2)\mathbf{w}$, which acts as a blur kernel. We then evaluate the derivatives with respect to the flow using finite differences. See Section 5 for more details on the grid construction.

With the blurred image grid, we can compute a pixel $B(\mathbf{x}) = (I * k_{\mathbf{w}(\mathbf{x})})(\mathbf{x})$ by bilinearly interpolating the grid at the point $\mathbf{w}(\mathbf{x})$. This amounts to sampling four grid images at $\mathbf{x}$ and weighting the results. In this case, the partial derivatives of the pixel with respect to u and v are trivial to compute using finite differences of the four pixel values. In general, the point at which we evaluate the grid may itself be a function of the flow (see Eq. (14)). We can apply the chain rule in this situation to get the partial derivatives with respect to the current flow estimate $\mathbf{w}_{12}$. See the supplementary material for the necessary equations.

4.3. Handling noise

When computing the flow between a clean long-exposure frame and a noisy short-exposure frame, the blurring of the noisy frame by the blur kernel of the long-exposure frame has a denoising effect. This is a good thing in general, since the long-exposure frame has low noise. However, the true blur kernel is small in regions with very little motion, so the noisy frame should be left mostly as-is in these regions. What we have actually observed is that the optimization introduces some (erroneous) non-zero flow so that the noisy frame is blurred slightly to average out the noise and reduce the energy in the data term.

To account for this undesired effect, we only use the blur-aware flow for pixels whose estimated blur kernels are larger than one pixel. Thus pixels with a baseline flow satisfying $\|\mathbf{w}\| < (\max(\tau_1, \tau_2))^{-1}$, where τ_1 and τ_2 are the duty cycles of the two frames, are assigned the baseline flow instead of the blur-aware flow. Our method is only designed to make improvements to the flow in blurred regions, so the recombination with the baseline in sharp regions is justified.

5. Implementation details

We initialize the flow field $\mathbf{w}_{12}$ at the coarsest level with the baseline flow in order to have reasonable kernel estimates from which to begin. The blur-aware flow can then be seen as a refinement of the baseline. We use fewer pyramid levels for the refinement to make better use of the coarse solution from the baseline (a minimum image width of 100 instead of 30).

The magnitude of the flow is generally large for regions with significant blur, so the error in the baseline flow should not be enough to make the motion approximation unreasonable (considering that even the motion approximated by the ground truth flow is inaccurate when the actual motion is not piecewise linear).

The extent of the blurred image grids are determined by the range of flow values in the baseline flow fields. The spacing of the grid points is chosen such that they are at least one pixel apart and the total number of images is at most M (we use 240). We need a reasonably dense grid to get accurate results at intermediate points. A one pixel difference is not as significant for large kernels as it is for small kernels, so we increase the grid spacing as the points get farther from the origin in order to get closer spacing near the origin. See the supplementary material for more detail.

Our implementation operates on the same feature images as [17], which consist of the luminosity, the gradients (scaled by a weight $\sqrt{\gamma}$ to match the gradient constancy term in [5]), the difference between the green and red channel, and the difference between the green and blue channel (both scaled by some weight). The multiple channels are summed into a single channel before solving Eq. (9).

All of our flows are computed using $\alpha = 0.01$ for the regularization parameter, $\gamma = 1$ for the gradient constancy weight, and a scale factor of 0.75 between pyramid levels. The color channel differences in the feature image are given weight 0.5. We perform 16 iterations of warping at each pyramid level. Apart from using fewer pyramid levels for the blur-aware refinement, we use the same parameter values for the baseline and the refinement.

6. Evaluation

Normally, we would like to evaluate the accuracy of optical flow using endpoint error [1]:

$$EE = \sqrt{(\hat{u} - u)^2 + (\hat{v} - v)^2}, \tag{16}$$

where $(\hat{u}, \hat{v})$ is the estimated flow and (u, v) is the ground truth. In the presence of motion blur, however, the notion of ground truth is unclear as observed in [22]: a point in I_1 has

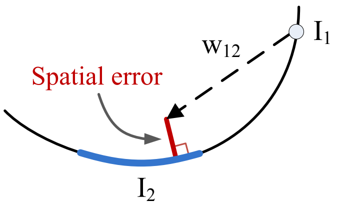

Figure 3. A pixel in sharp image I_1 has multiple corresponding pixels in blurred image I_2 due to motion and a non-zero exposure time. Since there is no unique ground truth flow with which to make an endpoint error calculation, we instead use the point along the true motion that is closest to the estimate.

multiple corresponding points in I_2 if it was moving during the exposure. One could choose the beginning instant of the exposure or the center of the exposure as the time for the ground truth, but the reality is that there is no unique ground truth flow to the blurred regions of I_2. To compare optical flow results in the presence of motion blur, we need to take this ambiguity into account.

If we have a point (x, y) in I_1 which moves along a trajectory $(x + u(t), y + v(t))$ during the exposure of I_2, and we consider any point along the trajectory to be a valid target for the flow, then we define the *spatial error* to be

$$SE = \min_{t_i \le t \le t_f} \sqrt{(\hat{u} - u(t))^2 + (\hat{v} - v(t))^2}, \qquad (17)$$

where t_i is the start time of the exposure and t_f is the end time. This can be seen as the length of the error vector from the estimated point to the actual trajectory, as shown in Figure 3. The average spatial error over the entire image provides one quantitative metric for the optical flow performance. This error metric reduces to the endpoint error when there is no motion or if the exposure time is infinitesimal.

Having a low spatial error is not sufficient for being a good optical flow result: we would also like flows to consistently map to the same time instant. To determine this *temporal consistency*, we keep track of the values of t that minimize the spatial error. Ideally these t values would all be the same, so their variability provides a good measure of the flow quality. We use the median absolute deviation (MAD) as a robust metric for the temporal consistency. This is only calculated over regions of the image with motion since there is no ambiguity in stationary regions.

7. Results

We test our optical algorithm on both synthetic and real sequences with spatially varying motion. More examples are provided in the supplementary material at http://pages.cs.wisc.edu/~lizhang/projects/blurflow/.

The running time of our algorithm is fairly short, with the refinement taking about 90 seconds at a 640x480 resolution; the baseline flow takes about 40 seconds. Most of the extra time is spent creating the blurred image grids prior to the optimization at each pyramid level.

7.1. Synthetic Sequences

We generate synthetic sequences with spatially-varying blur by moving a foreground object along a spline and alpha blending it over a stationary background image. To simulate motion blur from a non-zero exposure time, we move the foreground in small increments from the start time of the exposure to the stop time and average the transformed images together. Gaussian noise is added to the images with standard deviation inversely proportional to the exposure time to simulate the ISO speed adjusting to the changes in the exposure time. The synthetic sequences are assumed to be captured at one frame per second.

The spline used to generate the sequence provides ground truth information for evaluating the results. We calculate the metrics developed in Section 6 by projecting the estimated flows onto the spline. We only compare flows in the foreground since the background flows are mostly identical due to the noise handling mechanism.

In our first synthetic sequence, we vary the exposure time between frames so that some frames are sharp and noisy and some are blurred. Figure 4 shows the results for the flow from a blurred frame to a relatively sharp frame. The flow of the baseline method is inaccurate as it tries to warp the sharp frame to match the blurred frame. This makes the warped images look distorted and slightly blurry. Our result is significantly more accurate with the warped images retaining the sharpness and structure of the source image. The average spatial error for the entire sequence was 1.14 pixels for the baseline and 0.93 pixels for our method. The average temporal variability (MAD) was 0.048 seconds for the baseline and 0.011 for our method.

Our second synthetic sequence uses a constant exposure time so that all frames have significant motion blur. The motion along the spline still produces different blur kernels between neighboring frames, as we can clearly see in Figure 5. The improvements made by our method are more subtle when both images are blurred since the violation of the data term is less severe. However, the flow produced by our method is still noticeably more consistent than the baseline flow. The average spatial error for the entire sequence was 1.50 pixels for the baseline and 1.30 pixels for our method. The average temporal variability was 0.035 seconds for the baseline and 0.013 seconds for our method.

Most of the spatial error remaining in our results is actually caused by a slight uniform bias or shift in the flow rather than by local variability of the flow. This bias is most likely caused by the inaccuracy of the linear motion approximation: when the actual motion is curved, the piecewise linear approximation results in a blur kernel that is weighted more heavily in one direction than the true kernel. Despite the bias, our flow is still more consistent than the baseline and has lower spatial error on average. See the supplementary material for examples where the bias is more evident.

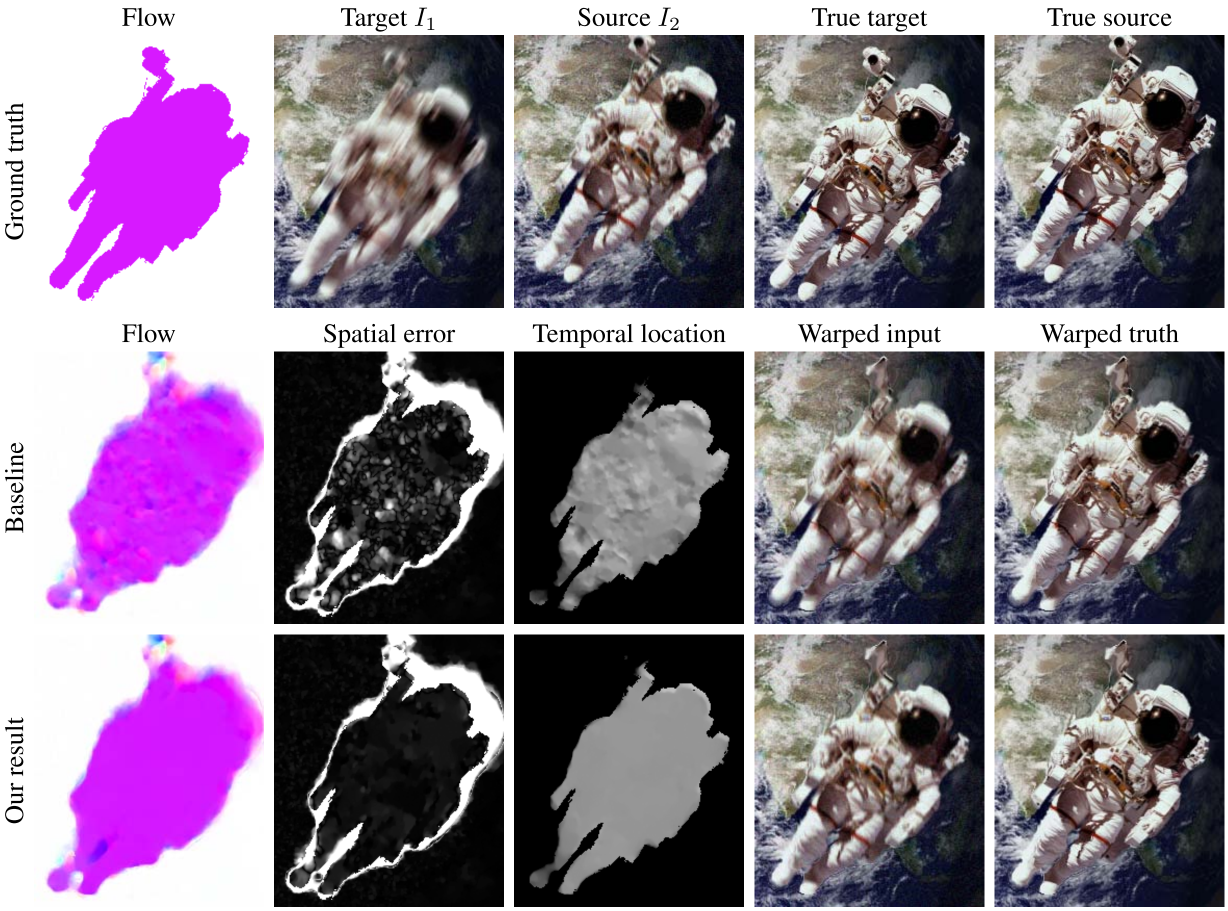

Figure 4. Close-ups of results from the synthetic *astronaut* sequence. **First row:** The ground truth flow using the centers of the exposure times is shown along with the input images. **Second row:** The baseline flow between the input images is inconsistent, and the warped images, which should retain the appearance of the source, are distorted. **Third row:** The results of our method are much more consistent and provide higher quality warped images. Brighter pixels in the spatial error images indicate larger errors with white being an error of 5 or more pixels. The temporal location images show the times that minimize the spatial error. **Best viewed electronically in color.**

7.2. Real Sequences

Our real sequences are captured with a Point Grey Grasshopper at 30 frames per second. Like the first synthetic sequence, we vary the exposure time between frames and adjust the gain to compensate.

The first real sequence consists of a tabletop scene with the camera moving in front of a collection of books. Depth in the scene combined with motion of the camera contribute to spatially-varying motion blur in the frames. The results for a pair of frames are shown in Figure 6. As in the synthetic sequences, the baseline method tries to warp the sharp source frame to match the blurred target frame. The result is that text which was legible in the source frame is distorted and mostly illegible after being warped. Our method does significantly better: the text is properly warped to the space of the target frame while retaining the legibility of the source. Similar improvements are seen in Figure 1, which shows the results for a different pair of frames. See the supplementary material for more results on real sequences.

8. Conclusions

In this paper, we have proposed an accurate optical flow algorithm for videos with spatially-varying motion blur. Unlike many other works in the field of motion estimation, we do not assume uniform blur kernels. Our algorithm provides significantly more accurate and consistent flow than the baseline method from which it is derived. We have also devised error metrics for evaluating flow in the presence of motion blur that do not force the choice of a ground truth time instant. There are several avenues for future work.

The slight bias in some of the flows produced by our algorithm could pose a problem for applications which require flow fields to be concatenated. As the fields are concatenated, the biases may accumulate leading to significant drift. Correcting the bias is an important subject for future work and may involve finding better ways to approximate the blur kernels or adding temporal constraints that directly address the problem of drift.

Since our algorithm is an extension of the classical

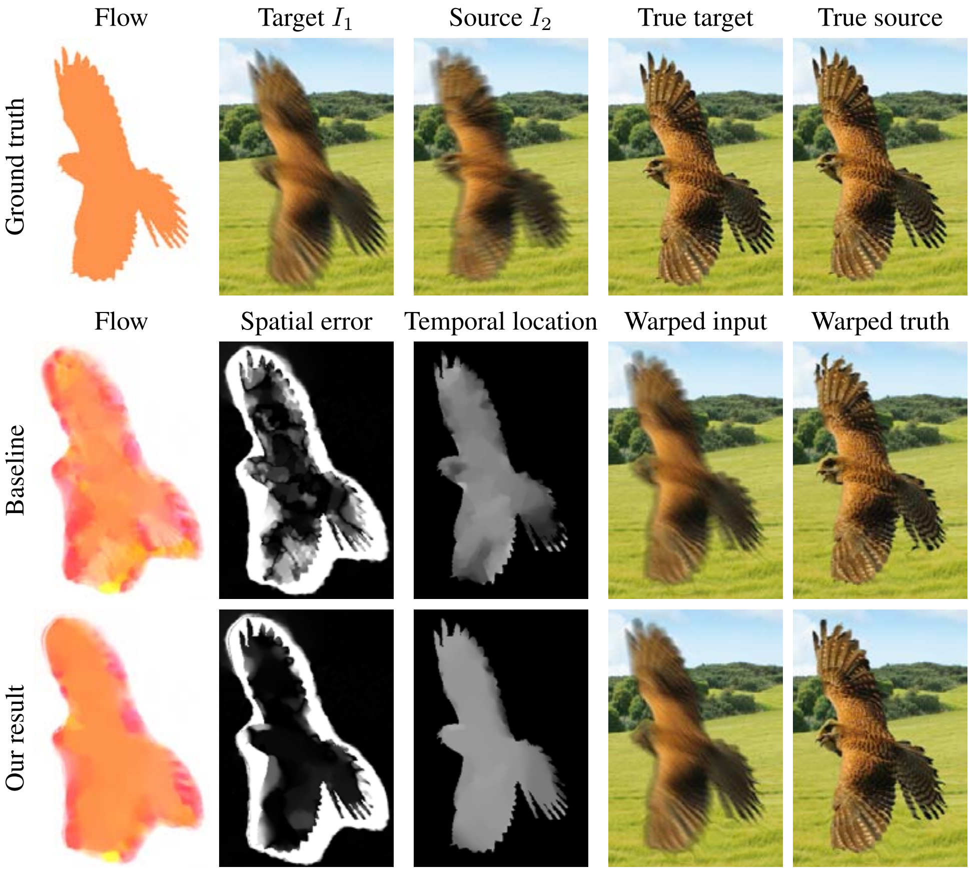

Figure 5. Close-ups of results from the synthetic *falcon* sequence. As the two inputs are both blurred, the improvements made by our method are more subtle. Some distortion is visible in the left wing for the baseline result that is mostly corrected in our result. The tail feathers are also more accurately warped in our result. Overall, our method is more consistent and has lower spatial error.

warping-based flow method, it could benefit from many of the recent innovations made in warping-based optical flow. Of particular importance are modifications to the smoothness term that help to better preserve edges [27, 28], and weighted median filtering of intermediate results to reject outliers in the flow [26]. We leave the exploration of such enhancements for future work.

While the run-time efficiency of our algorithm is fairly good, it currently has a large memory overhead due to the blurred image grids. This makes pipelining and other parallelization techniques difficult since multiple instances of the algorithm may not fit in main memory. Future work can investigate optimizations to the space efficiency of the algorithm. For example, a caching scheme for the blurred images could be devised that maintains run-time efficiency by exploiting the local smoothness of the optical flow field.

9. Acknowledgments

This work is supported in part by NSF EFRI-0937847, NSF IIS-0845916, NSF IIS-0916441, a Sloan Research Fellowship, and a Packard Fellowship for Science and Engineering. Travis Portz is also supported by a University of Wisconsin-Madison, Department of Electrical and Computer Engineering graduate fellowship.

References

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011.

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

[3] M. Ben-Ezra and S. Nayar. Motion-based motion deblurring. *PAMI*, 2004.

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

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[6] A. Bruhn and J. Weickert. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optical Flow Methods. *IJCV*, 2005.

[7] S. Dai and Y. Wu. Motion from blur. In *CVPR*, 2008.

[8] R. Fergus, B. Signh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single image. In *SIGGRAPH*, 2006.

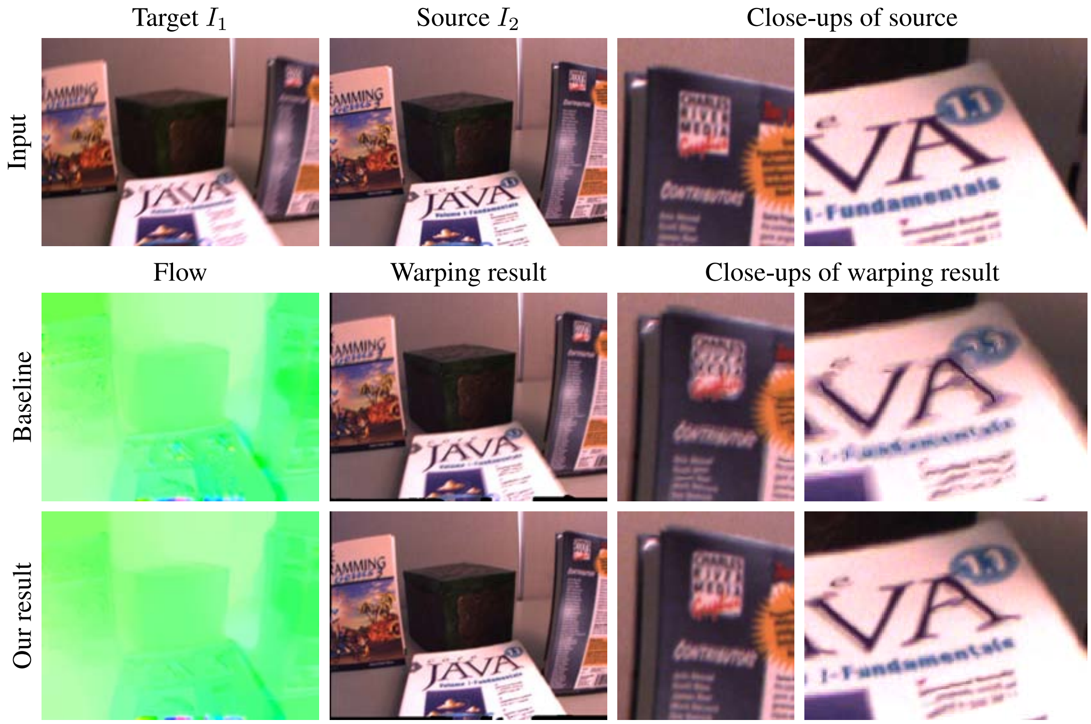

Figure 6. Results from the real *books* sequence. Due to the depth variation in the scene, the blur and flow are spatially variant. The baseline method produces distorted and blurry text when warping the sharp source image back to the blurred target image. Our method does much better at maintaining the appearance of the source image and provides more accurate optical flow as a result.

[9] Y. HaCohen, E. Schechtman, D. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *SIGGRAPH*, 2011.

[10] X. He, T. Luo, S. Yuk, K. Chow, K. Wong, and R. Chung. Motion estimation method for blurred videos and application of deblurring with spatially varying blur kernels. In *ICCIT*, 2010.

[11] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *CVPR*, 2010.

[12] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

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

[14] S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. In *ACM Trans. Graph.*, 2003.

[15] S. Kawamura, K. Kondo, Y. Konishi, and H. Ishigaki. Estimation of motion using motion blur for tracking vision system. In *WAC*, 2002.

[16] H. Lee, J. Kwon, and K. Lee. Simultaneous localization, mapping and deblurring. In *ICCV*, 2011.

[17] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, MIT, 2009.

[18] C. Liu and D. Sun. A bayesian approach to adaptive video super resolution. In *CVPR*, 2011.

[19] C. Mei and I. Reid. Modeling and generating complex motion blur for real-time tracking. In *CVPR*, 2008.

[20] Y. Park, V. Lepetit, and W. Woo. ESM-Blur: Handling & rendering blur in 3D tracking and augmentation. In *ISMAR*, 2009.

[21] T. Portz, L. Zhang, and H. Jiang. High-quality video denoising for motion-based exposure control. In *IWMV*, 2011.

[22] A. Rav-Acha and S. Peleg. Restoration of multiple images with motion blur in different directions. In *WACV*, 2000.

[23] Y. Schoueri, M. Scaccia, and I. Rekleitis. Optical flow from motion blurred color images. In *CRV*, 2009.

[24] S. Seitz and S. Baker. Filter flow. In *ICCV*, 2009.

[25] A. Sellent, M. Eisemann, B. Goldlücke, D. Cremers, and M. Magnor. Motion field estimation from alternate exposure images. *PAMI*, 2011.

[26] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[27] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *ICCV*, 2009.

[28] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. In *CVPR*, 2010.

[29] L. Yuan, J. Sun, L. Quan, and H. Shum. Blurred/non-blurred image alignment using sparseness prior. In *ICCV*, 2007.

[30] L. Yuan, J. Sun, L. Quan, and H. Shum. Image deblurring with blurred/noisy image pairs. In *ACM Trans. Graph.*, 2007.