Robust Optical Flow Estimation For Continuous Blurred Scenes Using RGB-Motion Imaging And Directional Filtering

Anonymous WACV submission

Paper ID 6

Abstract

Optical flow estimation is a difficult task given real-world video footage with camera and object blur. In this paper, we combine a 3D pose&position tracker with an RGB sensor allowing us to capture video footage together with 3D camera motion. We show that the additional camera motion information can be embedded into a hybrid optical flow framework by interleaving an iterative blind deconvolution and warping based minimization scheme. Such a hybrid framework significantly improves the accuracy of optical flow estimation in scenes with strong blur. Our approach yields improved overall performance against three state-of-the-art baseline methods applied to our proposed ground truth sequences as well as in several other real-world cases.

1. Introduction

Scene blur often occurs during fast camera movement in low-light conditions due to the requirement of adopting a longer exposure. Recovering both the blur kernel and the latent image from a single blurred image is known as *Blind Deconvolution* which is an inherently ill-posed problem. Cho and Lee [5] propose a fast deblurring process within a coarse-to-fine framework (Cho&Lee) using a predicted edge map as a prior. To reduce the noise effect in this framework, Zhong *et al.* [19] introduce a pre-filtering process which reduces the noise along a specific direction and preserves the image information in other directions. Their improved framework provides high quality kernel estimation with a low run-time but shows difficulties given combined object and camera motion blur.

To obtain higher performance, a handful of combined hardware and software-based approaches have also been proposed for image deblurring. Tai *et al.* [15] introduce a hybrid imaging system that is able to capture both video at high frame rate and a blurry image. The optical flow fields between the video frames are utilized to guide spatially-

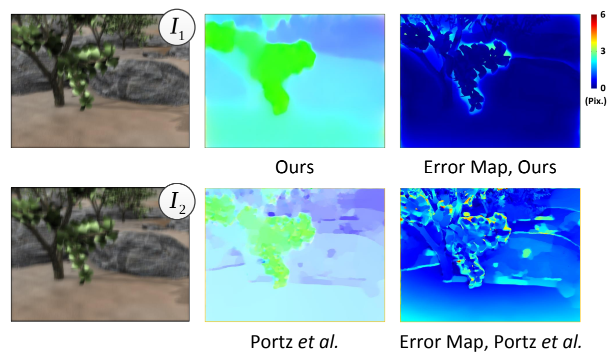


Figure 1. Visual comparison of our method to Portz *et al.* [12] on our ground truth benchmark *Grove2* with synthetic scene blur. **First Column**: the input images; **Second Column**: the optical flow fields calculated by our method and the baseline; **Third Column**: the RMS error maps against the ground truth.

varying blur kernel estimation. Levin *et al.* [9] propose to capture a uniformly blurred image by controlling the camera motion along a parabolic arc. Such uniform blur can then be removed based on the speed or direction of the known arc motion. As a complement to Levin *el al.*'s [9] hardware-based deblurring algorithm, Joshi *et al.* [7] apply inertial sensors to capture the acceleration and angular velocity of a camera over the course of a single exposure. This extra information is introduced as a constraint in their energy optimization scheme for recovering the blur kernel. All the hardware-assisted solutions described provide extra information in addition to the blurry image, which significantly improves overall performance. However, the methods require complex electronic setups and the precise calibration.

Optical flow techniques are widely studied and adopted across computer vision. One of advantages is the dense image correspondences they provide. In the last two decades, the optical flow model has evolved extensively – one landmark work being the variational model of Horn and Schunck [6] where the concept of *Intensity Constancy* is proposed. Under this assumption, pixel intensity does not change spatio-temporally, which is, however, often weakened in real-world images because of natural noise. To ad-

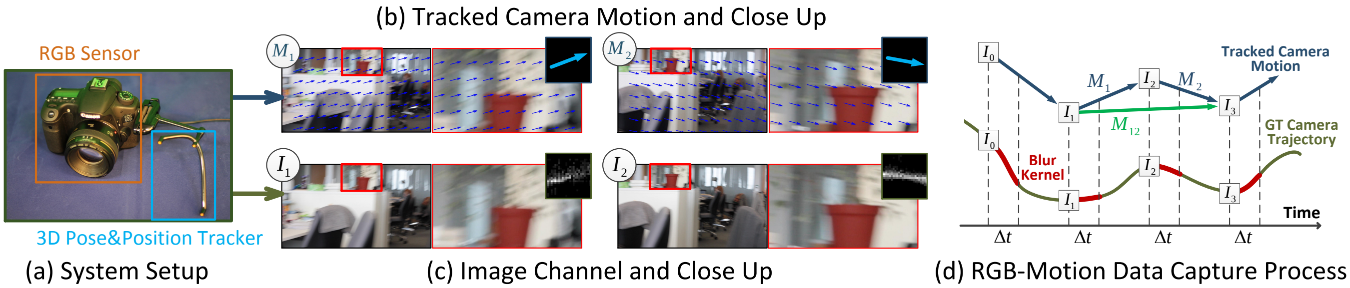


Figure 2. RGB-Motion Imaging System. **(a)**: Our system setup using a combined RGB sensor and 3D Pose&Position Tracker. **(b)**: The tracked 3D camera motion in relative frames. The top-right box is the average motion vector – which has similar direction to the blur kernel. **(c)**: Images captured from our system. The top-right box presents the blur kernel estimated using [5]. **(d)**: The internal process of our system where the Δt presents the exposure time.

dress this, some complementary concepts have been developed to improve performance given large displacements [2], taking advantage of feature-rich surfaces [18] and adapting to non-rigid deformation in scenes [10]. However, flow approaches that can perform well given blury scenes – where the *Intensity Constancy* is usually violated – are less common. Of the approaches that do exist, Schoueri *et al.* [13] perform a linear deblurring filter before optical flow estimation while Portz *et al.* [12] attempt to match un-uniform camera motion between neighboring input images. Whereas the former approach may be limited given nonlinear blur in real-world scenes; the latter requires two extra frames to parameterize the motion-induced blur.

1.1. Contributions

In this paper, our major contribution is to utilize an *RGB-Motion Imaging System* – an RGB sensor combined with a 3D pose&position tracker – in order to propose: **(A)** an iterative enhancement process for scene blur estimation which encompasses the tracked camera motion (Sec. 2) and a *Directional High-pass Filter* (Sec. 3 and Sec. 6.2); **(B)** a *Blur-Robust Optical Flow Energy* formulation (Sec. 5); and **(C)** a hybrid coarse-to-fine framework (Sec. 6) for computing optical flow in blur scenes by interleaving an iterative blind deconvolution process and a warping based minimization scheme. In the evaluation section, we compare our method to three existing state-of-the-art optical flow approaches on our proposed ground truth sequences (Fig. 1, containing blur and baseline blur-free equivalents) and also illustrate the practical benefit of our algorithm given other real-world cases.

2. RGB-Motion Imaging System

Scene blur within video footage is typically due to fast camera motion and/or long exposure times. In particular, such blur can be considered as a function of the camera trajectory supplied to image space during the exposure time Δt. It therefore follows that knowledge of the actual camera motion between image pairs can provide significant information when performing image deblurring [7, 9]. In this paper, we propose a simple and portable setup (Fig. 2(a)), combining an RGB sensor and a 3D pose&position tracker (Trackir, NaturalPoint Inc.) in order to capture continuous scenes along with real-time camera pose&position information. Our tracker provides the rotation (yaw, pitch and roll), translation and zoom information synchronized to the relative corresponding image frame using the middleware of [8]. Assuming objects have similar depth within the same scene (A common assumption in image deblurring which will be discussed in our future work), the tracked 3D camera motion in image coordinates can be formulated as:

$$\mathbf{M}_j = \frac{1}{n} \sum_{\mathbf{x}} K \left([R|T] \, \mathbf{X}_{j+1} - \mathbf{X}_j \right) \qquad (1)$$

where $\mathbf{M}_j$ represents the average of the camera motion vectors from the image j to image $j+1$. $\mathbf{X}$ denotes the 3D position of the camera while $\mathbf{x} = (x, y)^T$ is a pixel location and n represents the number of pixels in an image. K represents the 3D projection matrix while R and T denote the rotation and translation matrices respectively of tracked camera motion in the image domain. Fig 2(b,c) shows sample data (video frames and camera motion) captured from our imaging system. It is observed that blur from the real-world video is spatially-varying but near linear due to the relatively high sampling rate of the camera. The blur direction can therefore be approximately described using the tracked camera motion. Let the tracked camera motion $\mathbf{M}_j = (r_j, \theta_j)^T$ be represented in polar coordinates where r_j and θ_j denote the magnitude and directional component respectively. j is a sharing index between tracked camera motion and frame number. In addition, we also consider the combined camera motion vector of neighboring images as shown in Fig 2(d), e.g. $\mathbf{M}_{12} = \mathbf{M}_1 + \mathbf{M}_2$ where $\mathbf{M}_{12} = (r_{12}, \theta_{12})$ denotes the combined camera motion vector from image 1 to image 3. As one of our main contributions, these real-time motion vectors are proposed to provide additional constraints for blur kernel enhancement (Sec. 6) within our optical flow framework.

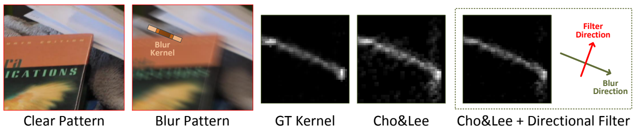


Figure 3. Directional high-pass filter for blur kernel enhancement. Given the blur direction θ, a directional high-pass filter along $\theta + \pi/2$ is applied to preserve blur detail in the estimated blur kernel.

3. Blind Deconvolution

The motion blur process can commonly be formulated:

$$I = k \otimes l + n \tag{2}$$

where I is a blurred image and k represents a blur kernel w.r.t. a specific *Point Spread Function*. l is the latent image of I; $\otimes$ denotes the convolution operation and n represents spatial noise within the scene. In the blind deconvolution operation, both k and l are estimated from I, which is an ill-posed (but extensively studied) problem. A common approach for blind deconvolution is to solve both k and l in an iterative framework using a coarse-to-fine strategy:

$$k = \mathbf{argmin}_k \{\|I - k \otimes l\| + \rho(k)\}, \tag{3}$$
$$l = \mathbf{argmin}_l \{\|I - k \otimes l\| + \rho(l)\}. \tag{4}$$

where ρ represents a regularization that penalizes spatial smoothness with a sparsity prior [5], and is widely used in recent state-of-the-art work [14, 18]. Due to noise sensitivity, low-pass and bilateral filters [16] are typically employed before deconvolution. Eq. 5 denotes the common definition of an optimal kernel from a filtered image.

$$\begin{aligned} k_f &= \mathbf{argmin}_{k_f} \{\|(k \otimes l + n) \otimes f - k_f \otimes l\| + \rho(k_f)\} \\ &\approx \mathbf{argmin}_{k_f} \|l \otimes (k \otimes f - k_f)\| = k \otimes f \end{aligned} \tag{5}$$

where k represents the ground truth blur kernel, f is a filter, and k_f denotes the optimal blur kernel from the filtered image $I \otimes f$. The low-pass filtering process improves deconvolution computation by removing spatially-varying high frequency noise but also results in the removal of useful information which yields additional errors over object boundaries. To preserve this useful information, we introduce a directional high-pass filter that utilizes our tracked 3D camera motion.

4. Directional High-pass Filter

Detail enhancement using directional filters has been proved effective in several areas of computer vision [19]. In this paper, we define a directional high-pass filter as:

$$f_\theta \otimes I(\mathbf{x}) = m \int g(t) I(\mathbf{x} + t\Theta) dt \tag{6}$$

where $\mathbf{x} = (x, y)^T$ represents a pixel position and $g(t) = 1 - exp\{-t^2/2\sigma^2\}$ denotes a 1D Gaussian based high-pass function. $\Theta = (\cos\theta, \sin\theta)^T$ controls the filtering direction along θ. m is a normalization factor defined as $m = \left(\int g(t) dt\right)^{-1}$. The filter f_θ is proposed to preserve overall high frequency details along direction θ without affecting blur detail in orthogonal directions [4]. Given a directionally filtered image $b_\theta = f_\theta \otimes I(\mathbf{x})$, the optimal blur kernel is defined (Eq 5) as $k_\theta = k \otimes f_\theta$. Fig. 3 demonstrates that noise or object motion within a scene usually results in low frequency noise in the estimated blur kernel (Cho&Lee [5]). This low frequency noise can be removed by our directional high-pass filter while preserving major blur details. In this paper, this directional high-pass filter is supplemented into the Cho&Lee [5] framework using a coarse-to-fine strategy in order to recover high quality blur kernels for use in our optical flow estimation (Sec. 6.2).

5. Blur-Robust Optical Flow Energy

Within a blurry scene, a pair of adjacent natural images may contain different blur kernels, further violating *Intensity Constancy*. This results in unpredictable flow error across the different blur regions. To address this issue, Portz *et al.* proposed a modified *Intensity Constancy* term by matching the un-uniform blur between the input images. As one of our main contributions, we extend this assumption to a novel *Blur Gradient Constancy* term in order to provide extra robustness against illumination change and outliers. Our main energy function is given as follows:

$$E(\mathbf{w}) = E_B(\mathbf{w}) + \gamma E_S(\mathbf{w}) \tag{7}$$

A pair of consecutively observed frames from an image sequence is considered in our algorithm. $I_1(\mathbf{x})$ represents the current frame and its successor is denoted by $I_2(\mathbf{x})$ where $I_* = k_* \otimes l_*$ and $\{I_*, l_* : \Omega \subset \mathbb{R}^3 \to \mathbb{R}\}$ represent rectangular images in the RGB channel. Here l_* is latent image and k_* denotes the relative blur kernel. The optical flow displacement between $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ is defined as $\mathbf{w} = (u, v)^T$. To match the un-uniform blur between input images, the blur kernel from each input image is applied to the other. We have new blur images b_1 and b_2 as follows:

$$b_1 = k_2 \otimes I_1 \approx k_2 \otimes k_1 \otimes l_1 \tag{8}$$
$$b_2 = k_1 \otimes I_2 \approx k_1 \otimes k_2 \otimes l_2 \tag{9}$$

Our energy term encompassing *Intensity* and *Gradient Constancy* relates to b_1 and b_2 as follows:

$$\begin{aligned} E_B(\mathbf{w}) = \int_\Omega &\phi(\|b_2(\mathbf{x}+\mathbf{w}) - b_1(\mathbf{x})\|^2 \\ &+ \alpha \|\nabla b_2(\mathbf{x}+\mathbf{w}) - \nabla b_1(\mathbf{x})\|^2) d\mathbf{x} \end{aligned} \tag{10}$$

The term $\nabla = (\partial_{xx}, \partial_{yy})^T$ presents a spatial gradient and $\alpha \in [0, 1]$ denotes a linear weight. The smoothness regularizer penalizes global variation as follows:

$$E_S(\mathbf{w}) = \int_\Omega \phi(\|\nabla u\|^2 + \|\nabla v\|^2)d\mathbf{x} \qquad (11)$$

where we apply the Lorentzian function $\phi(s) = log(1 + s^2/2\epsilon^2)$ with $\epsilon = 0.001$ to both the data term and smoothness term for robustness against flow blur on boundaries [10]. In the following section, our optical flow framework is introduced in detail.

6. Optical Flow Framework

Algorithm 1: Blur-Robust Optical Flow Framework

Input : A image pair I_1, I_2 and camera motion $\theta_1, \theta_2, \theta_{12}$
Output : Optimal optical flow field $\mathbf{w}$

A n-level top-down pyramid is built with the level index i
$i \leftarrow 0$
$l_1^i \leftarrow I_1^i, l_2^i \leftarrow I_2^i$
$k_1^i \leftarrow 0, k_2^i \leftarrow 0, \mathbf{w}^i \leftarrow (0, 0)^T$
for *coarse to fine* **do**
    $i \leftarrow i + 1$
    Resize $k^i_{\{1,2\}}, l^i_{\{1,2\}}, I^i_{\{1,2\}}$ and $\mathbf{w}^i$ with the ith scale
    foreach $* \in \{1, 2\}$ **do**
        $k_*^i \leftarrow$ `IterativeBlindDeconvolve` (l_*^i, I_*^i)
        $k_*^i \leftarrow$ `DirectionalFilter` ($k_*^i, \theta_1, \theta_2, \theta_{12}$)
        $l_*^i \leftarrow$ `NonBlindDeconvolve` (k_*^i, I_*^i)
    endfor
    $b_1^i \leftarrow I_1^i \otimes k_2^i, b_2^i \leftarrow I_2^i \otimes k_1^i$
    $d\mathbf{w}^i \leftarrow$ `EnergyOptimization` ($b_1^i, b_2^i, \mathbf{w}^i$)
    $\mathbf{w}^i \leftarrow \mathbf{w}^i + d\mathbf{w}^i$
endfor

Our overall framework is outlined in Algorithm 1 based on an iterative top-down, coarse-to-fine strategy. Prior to minimizing the *Blur-Robust Optical Flow Energy* (Sec. 6.3), a fast blind deconvolution approach [5] is performed for pre-estimation of the blur kernel (Sec. 6.1), which is followed by kernel refinement using our *Directional High-pass Filter* (Sec. 6.2). All these steps are detailed in the following subsections.

6.1. Iterative Blind Deconvolution

Cho and Lee [5] describe a fast and accurate approach (Cho&Lee) to recover the spatially-varying blur kernel. As shown in Algorithm 1, we perform a similar approach for the pre-estimation of the blur kernel k within our iterative process, which involves two steps of prediction and kernel estimation. Given the latent image l estimated from the consecutively coarser level, the gradient maps $\Delta l = \{\partial_x l, \partial_y l\}$ of l are calculated along the horizontal and vertical directions respectively in order to enhance salient edges and reduce noise in featureless regions of l. Next, the predicted gradient maps Δl as well as the gradient map of the blurry image I are utilized to compute the pre-estimated blur kernel by minimizing the energy function as follows:

$$\begin{gathered} k = \textbf{argmin}_k \sum_{I_*, l_*} \omega_* \|I_* - k \otimes l_*\|^2 + \delta \|k\|^2 \\ (I_*, l_*) \in \{(\partial_x I, \partial_x l), (\partial_y I, \partial_y l), (\partial_{xx} I, \partial_{xx} l), \\ (\partial_{yy} I, \partial_{yy} l), (\partial_{xy} I, (\partial_x \partial_y + \partial_y \partial_x) l/2)\} \end{gathered} \qquad (12)$$

where δ denotes the weight of Tikhonov regularization and $\omega_* \in \{\omega_1, \omega_2\}$ represents a linear weight for the derivatives in different directions. Both I and l are propagated from the nearest coarse level within the pyramid. To minimize this energy Eq. (12), we follow the inner-iterative numerical scheme of [5] which yields a pre-estimated blur kernel k.

6.2. Directional High-pass Filtering

Once the pre-estimated kernel k is obtained, our *Directional High-pass Filters* are applied to enhance the blur information by reducing noise in the orthogonal direction of the tracked camera motion. Although our *RGB-Motion Imaging System* provides an accurate camera motion estimation, we take into account the directional components $\{\theta_1, \theta_2, \theta_{12}\}$ of two consecutive camera motions M_1 and M_2 as well as their combination M_{12} (Fig. 2(d)) for extra robustness. The pre-estimated blur kernel is filtered along its orthogonal direction as follows:

$$k = \sum_{\beta_*, \theta_*} \beta_* k \otimes f_{\theta_* + \pi/2} \qquad (13)$$

where $\beta_* \in \{1/2, 1/3, 1/6\}$ linearly weights the contribution of filtering in different directions. Note that two consecutive images I_1 and I_2 are involved in our framework where the former accepts the weight set $(\beta_*, \theta_*) \in \{(1/2, \theta_1), (1/3, \theta_2), (1/6, \theta_{12})\}$ while the other weight set $(\beta_*, \theta_*) \in \{(1/3, \theta_1), (1/2, \theta_2), (1/6, \theta_{12})\}$ is performed for the latter. This filtering process yields an updated blur kernel k which is used to update the latent image l within a non-blind deconvolution [19].

Having performed blind deconvolution and directional filtering (Sec. 6.1, 6.2), two updated blur kernels k_1^i and k_2^i on the ith level of the pyramid are obtained from input images I_1^i and I_2^i respectively, which is followed by the uniform blur image b_1^i and b_2^i computation using Eq. (9). In the following subsection, *Blur-Robust Optical Flow Energy* optimization on b_1^i and b_1^i is introduced in detail.

6.3. Optical Flow Energy Optimization

As mentioned in Sec. 5, our blur-robust energy is continuous but highly nonlinear. Minimization of such energy function is extensively studied in the optical flow community. In this section, a numerical scheme combining *Euler-Lagrange Equations* and *Nested Fixed Point Iterations* is applied [2] to solve our main energy function Eq. 7. For clarity of presentation, we define the following mathematical abbreviations:

$$\begin{aligned} b_x &= \partial_x b_2(\mathbf{x}+\mathbf{w}) & b_{yy} &= \partial_{yy} b_2(\mathbf{x}+\mathbf{w}) \\ b_y &= \partial_y b_2(\mathbf{x}+\mathbf{w}) & b_z &= b_2(\mathbf{x}+\mathbf{w}) - b_1(\mathbf{x}) \\ b_{xx} &= \partial_{xx} b_2(\mathbf{x}+\mathbf{w}) & b_{xz} &= \partial_x b_2(\mathbf{x}+\mathbf{w}) - \partial_x b_1(\mathbf{x}) \\ b_{xy} &= \partial_{xy} b_2(\mathbf{x}+\mathbf{w}) & b_{yz} &= \partial_y b_2(\mathbf{x}+\mathbf{w}) - \partial_y b_1(\mathbf{x}) \end{aligned}$$

After *Euler-Lagrange Equations* are applied to Eq. (7), we minimize the resulting system in a coarse-to-fine framework within a top-down image pyramid. In the outer fixed point iterations, we initialize the flow field $\mathbf{w} = (0,0)^T$ on the top (coarsest) level of the pyramid and propagate this to the next finer level as $\mathbf{w}^{i+1} \approx \mathbf{w}^i + d\mathbf{w}^i$ where we follow the assumption that the flow field on finer level $i+1$ is estimated by the flow field and the increments from the previous coarser level k. First order *Taylor Expansion* is then applied to the terms of b_z^{i+1}, b_{xz}^{i+1} and b_{yz}^{i+1}, which results in

$$\begin{aligned} b_z^{i+1} &\approx b_z^i + b_x^i du^i + b_y^i dv^i, \\ b_{xz}^{i+1} &\approx b_{xz}^k + b_{xx}^i du^i + b_{xy}^i dv^i, \\ b_{yz}^{i+1} &\approx b_{yz}^k + b_{xy}^i du^i + b_{yy}^i dv^i. \end{aligned}$$

where du^i and dv^i are two unknown increments which will be solved in our inner fixed point iterations. Given the initialization of $du^{i,0} = 0$ and $dv^{i,0} = 0$, we assume that $du^{i,j}$ and $dv^{i,j}$ converge within j iterations. We have the final linear system in $du^{i,j+1}$ and $dv^{i,j+1}$ as follows:

$$\begin{aligned} &(\phi')_B^{i,j} \cdot \{ b_x^i (b_z^i + b_x^i du^{i,j+1} + b_y^i dv^{i,j+1}) \\ &\quad + \alpha b_{xx}^i (b_{xz}^i + b_{xx}^i du^{i,j+1} + b_{xy}^i dv^{i,j+1}) \\ &\quad + \alpha b_{xy}^i (b_{yz}^i + b_{xy}^i du^{i,j+1} + b_{yy}^i dv^{i,j+1}) \} \\ &\qquad - \gamma\, (\phi')_S^{i,j} \cdot \nabla(u^i + du^{i,j+1}) = 0 \end{aligned} \tag{14}$$

$$\begin{aligned} &(\phi')_B^{i,j} \cdot \{ b_y^i (b_z^i + b_x^i du^{i,j+1} + b_y^i dv^{i,j+1}) \\ &\quad + \alpha b_{yy}^i (b_{yz}^i + b_{xy}^i du^{i,j+1} + b_{yy}^i dv^{i,j+1}) \\ &\quad + \alpha b_{xy}^i (b_{xz}^i + b_{xx}^i du^{i,j+1} + b_{xy}^i dv^{i,j+1}) \} \\ &\qquad - \gamma\, (\phi')_S^{i,j} \cdot \nabla(v^i + dv^{i,j+1}) = 0 \end{aligned} \tag{15}$$

where $(\phi')_B^{i,j}$ denotes a robustness factor against flow discontinuty and occlusion on the object boundaries. $(\phi')_S^{i,j}$ represents the diffusivity of the smoothness regularization.

$$\begin{aligned} (\phi')_B^{i,j} &= \phi' \{ (b_z^i + b_x^i du^{i,j} + b_y^{i,j} dv^{i,j})^2 \\ &\quad + \alpha (b_{xz}^i + b_{xx}^i du^{i,j} + b_{xy}^i dv^{i,j})^2 \\ &\quad + \alpha (b_{yz}^i + b_{xy}^i du^{i,j} + b_{yy}^i dv^{i,j})^2 \} \\ (\phi')_S^{i,j} &= \phi' \{ \|\nabla(u^i + du^{i,j})\|^2 + \|\nabla(v^i + dv^{i,j})\|^2 \} \end{aligned}$$

In our implementation, the image pyramid is constructed with a downsampling factor of 0.75. The final linear system in Eq. (14,15) is solved using *Conjugate Gradients* within 45 iterations.

7. Evaluation

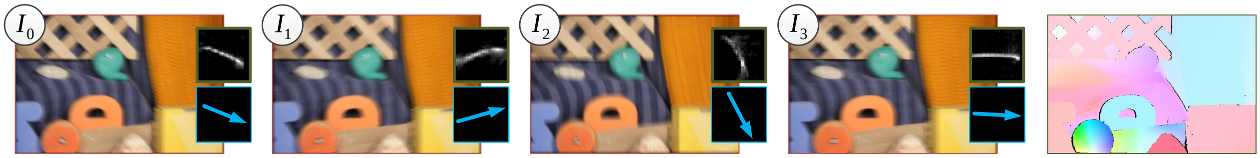
In this section, we evaluate our method on both synthetic and real-world sequences and compare its performance against three existing state-of-the-art optical flow approaches of Xu *et al.*'s MDP [18], Portz *et al.*'s [12] and Brox *et al.*'s [2] (an implementation of [11]). MDP is one of the best performing optical flow methods given blur-free scenes, and is one of the top 3 approaches in the Middlebury benchmark [1]. Portz *et al.*'s method represents the current state-of-the-art in optical flow estimation given spatially-varying object blur scenes while Brox *et al.*'s contains a similar optimization framework and numerical scheme to Portz *et al.*'s, and ranks in the midfield of the Middlebury benchmarks based on overall average. Note that all three baseline methods are evaluated using their default parameters setting.

In the following subsections, we compare our algorithm (*moBlur*) and three different implementations (*nonGC*, *nonDF* and *nonGCDF*) against the baseline methods. *nonGC* represents the implementation **without** the *Gradient Constancy* term while *nonDF* denotes an implementation **without** the directional filtering process. *nonGCDF* is the implementation with neither of these features. The results show that our *Blur-Robust Optical Flow Energy* and *Directional High-pass Filter* significantly improve algorithm performance for blur scenes in both synthetic and real-world cases.

7.1. Middlebury Dataset with Scene Blur

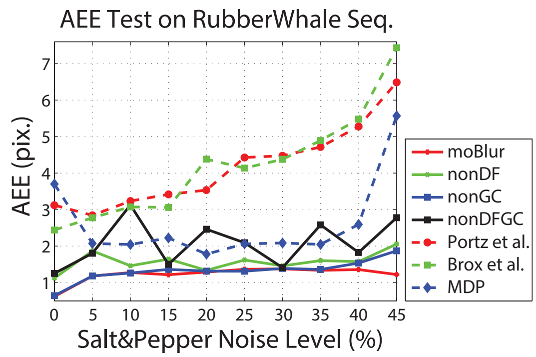
One advance for evaluating optical flow given scenes with object blur is proposed by Portz *et al.* [12] where synthetic *Ground Truth* (GT) scenes are rendered with blurry moving objects against a blur-free static/fixed background. However, their use of synthetic images and controlled object trajectories lead to a lack of global camera blur, natural photographic properties and real camera motion behavior. To overcome these limitations, we render four sequences with scene blur and corresponding GT flow-fields by combining sequences from the Middlebury dataset [1] with blur kernels estimated using our system.

In our experiments we select the sequences *Grove2*, *Hydrangea*, *RubberWhale* and *Urban2* from the Middlebury dataset. For each of them, four adjacent frames are selected as latent images along with the GT flow field $\mathbf{w}_{gt}$ (supplied by Middlebury) for the middle pair. 40×40 blur kernels are then estimated [5] from real-world video streams captured using our *RGB-Motion Imaging System*. As shown in Fig. 4(a), those kernels are applied to generate blurry images denoted by I_0, I_1, I_2 and I_3 while the camera motion direction is set for each frame based on the 3D motion data. Although the $\mathbf{w}_{gt}$ between latent images can be utilized for the evaluation on relative blur images I_* [3, 17],

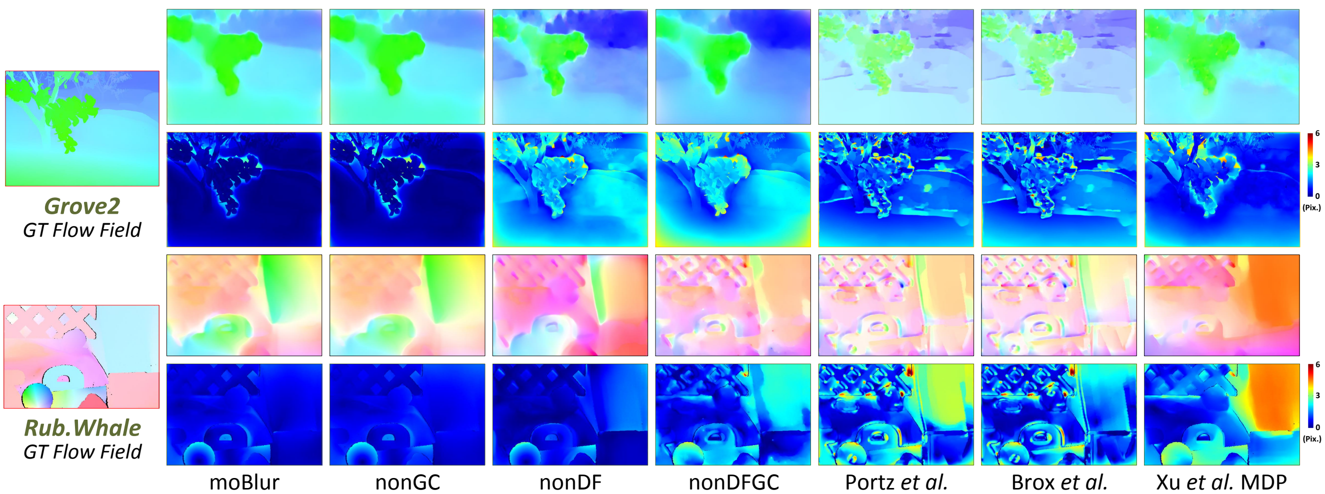


(a) A sample synthetic sequence *RubberWhale* with the blur kernel and tracked camera motion direction.

AEE/AAE	Grove2 AEE	Grove2 AAE	Hydrangea AEE	Hydrangea AAE	Rub.Whale AEE	Rub.Whale AAE	Urban2 AEE	Urban2 AAE
Ours, moBlur	0.47$_1$	2.34$_1$	0.67$_1$	2.19$_1$	0.62$_1$	3.67$_1$	1.36$_1$	2.87$_1$
moBlur-nonGC	0.49$_2$	2.38$_2$	0.95$_2$	2.23$_2$	0.64$_2$	3.71$_2$	1.54$_2$	3.03$_2$
moBlur-nonDF	1.52$_6$	4.96$_6$	1.83$_3$	3.00$_3$	1.12$_3$	6.45$_3$	2.50$_3$	5.19$_5$
nonDFGC	1.62$_7$	5.14$_7$	2.28$_5$	3.21$_4$	1.25$_4$	7.71$_4$	2.98$_5$	5.44$_6$
Portz *et al.*	1.14$_4$	4.11$_4$	2.62$_6$	3.55$_6$	3.12$_6$	8.18$_6$	3.44$_6$	5.10$_4$
Brox *et al.*	1.24$_5$	4.53$_5$	2.26$_4$	3.47$_5$	2.44$_5$	7.98$_5$	2.92$_4$	4.60$_3$
Xu *et al.*, MDP	1.06$_3$	3.46$_3$	3.40$_7$	3.55$_6$	3.70$_6$	8.21$_7$	5.62$_7$	6.82$_7$



(b) **Left**: Quantitative *Average Endpoint Error* (AEE) and *Average Angle Error* (AAE) comparison on our synthetic sequences where the subscripts show the rank in relative terms. **Right**: AEE measure on *RubberWhale* by ramping up the noise distribution.



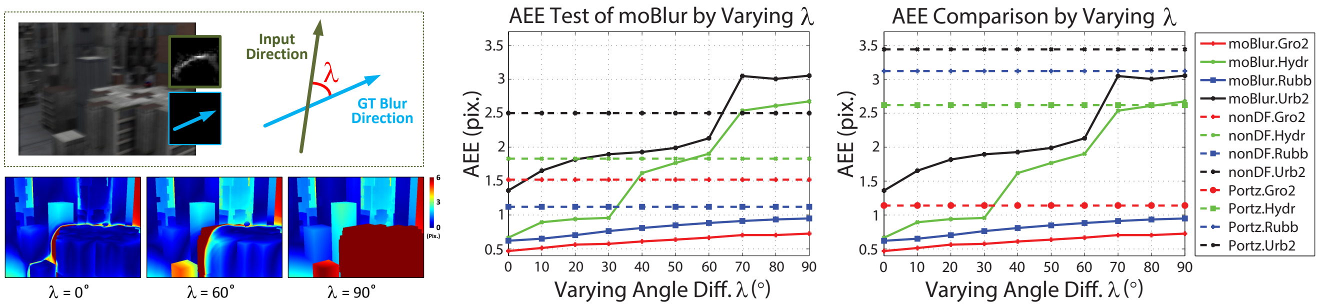
(c) Visual comparison on *Grove2* and *RubberWhale* by varying baseline methods. For each sequence, **First Row**: optical flow fields from different methods. **Second Row**: the error maps against the ground truth.

Figure 4. Quantitative evaluation on four synthetic blur sequences with both camera motion and ground truth.

strong blur can significantly violate the original image intensity, which leads to a multiple correspondences problem: a point in the current image corresponds to multiple points in the consecutive image. To remove such multiple correspondences, we sample reasonable correspondence set $\{\hat{\mathbf{w}} \mid \hat{\mathbf{w}} \subset \mathbf{w}_{gt}, |I_2(\mathbf{x}+\hat{\mathbf{w}}) - I_1(\mathbf{x})| < \epsilon\}$ to use as the GT for the blur images I_* where ϵ denotes a predefined threshold. Once we obtain $\hat{\mathbf{w}}$, both *Average Endpoint Error* (AEE) and *Average Angle Error* (AAE) tests [1] are considered in our evaluation.

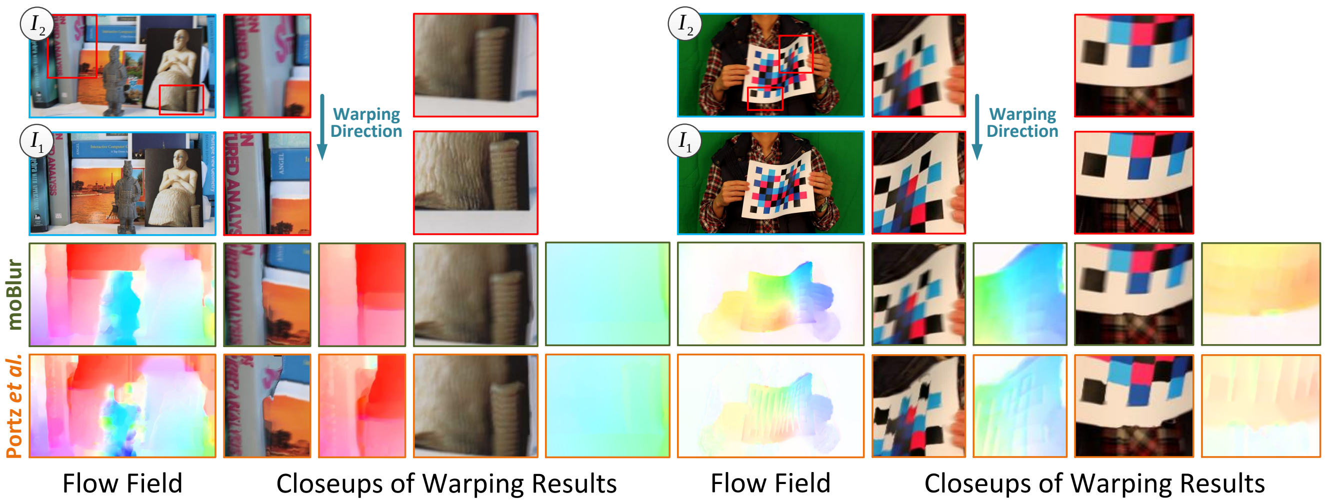
Fig. 4(b) *Left* shows AEE (in pixel) and AAE (in degree) tests on our four synthetic sequences. *moBlur* and *nonGC* lead both AEE and AAE tests in all the trials. Both Brox *et al.* and MDP yield significant error in *Hydrangea*, *RubberWhale* and *Urban2* because those sequences contain large textureless regions with blur, which in turn weakens the inner motion estimation process as shown in Fig. 4(c). Furthermore, Fig 4(b) *Right* shows the AEE metric for *RubberWhale* by varying the distribution of *Salt&Pepper* noise. It is observed that a higher noise level leads to additional errors for all the baseline methods. Both *moBlur* and *nonGC* yield the best performance while Portz *et al.* and Brox *et al.* show a similar rising AEE trend when the noise increases.

To investigate how the tracked 3D camera motion affect-

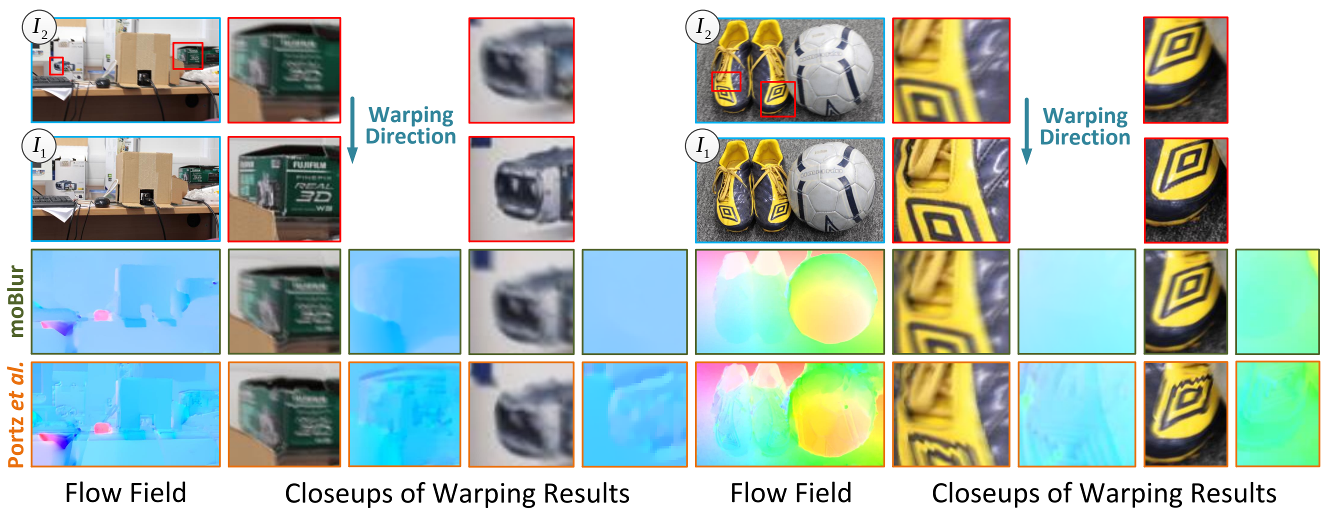


(a) sample sequence and error maps (b) Comparion of on/off filter (c) Comparion to Portz *et al.*

Figure 5. AEE measure of our method (*moBlur*) by varying the input motion directions. **(a)**: the overall measure strategy and error maps of *moBlur* on sequence *Urban2*. **(b)**: the quantitative comparison of *moBlur* against *nonDF* by ramping up the angle difference λ. **(c)**: the measure of *moBlur* against Portz *et al.* [12].



(a) Visual comparison on real-world sequences of *warrior* and *chessboard*.



(b) Visual comparison on real-world sequences of *LabDesk* and *shoes*.

Figure 6. Visual comparison of image warping on real-world sequences of *warrior*, *chessboard*, *LabDesk* and *shoes*, captured by our *RGB-Motion Imaging System*.

s the performance of our algorithm, we compare *moBlur* to *nonDF* and Portz *et al.* by varying the input motion directions. As shown in Fig. 5(a), we rotate the input vector with respect to the GT blur direction by an angle of λ degrees. Fig. 5(b,c) shows the AEE metric by increasing the λ. We observe that the AEE increases during this test. *moBlur* outperforms the *nonDF* (*moBlur* without the directional filter) in both *Grove2* and *RubberWhale* while *nonDF* provides higher performance in *Hydrangea* when λ is larger than $50°$. In addition, *moBlur* outperforms Portz *et al.* in all trials except *Hydrangea* where Portz *et al.* shows a minor advantage (AEE 0.05) when $\lambda = 90°$.

7.2. Real-world Dataset

Fig. 6 shows visual comparison of our method *moBlur* against Portz *et al.* on real-world sequences of *warrior*, *chessboard*, *LabDesk* and *shoes* captured using our *RGB-Motion Imaging System*. Both *warrior* and *chessboard* contain occlusions, large displacements and depth change while the sequences of *LabDesk* and *shoes* embodies the object motion blur and large textureless regions within the same scene. It is observed that our method preserves appearance details on the object surface and reduce boundary distortion after warping using the flow field. In addition, our method shows robustness given cases where multiple types of blur exist in the same scene (Fig.6(b), sequence *shoes*). Please note that more details of our real-world results can be found in the supplementary document.

8. Conclusion

In this paper, we proposed a hybrid optical flow model by interleaving iterative blind deconvolution and a warping based minimization scheme. We also highlighted the benefits of both the RGB-Motion data and directional filters in the image deblurring task. Our evaluation demonstrated the high performance of our method against large scene blur in both noisy and real-world cases. One limitation in our method is that the spatial invariance assumption for the blur is not valid in some real-world scenes, which may reduce accuracy in the case where the object depth significantly changes. Finding a depth-dependent deconvolution is a challenge for future work.

References

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92:1–31, 2011. 5, 6

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, pages 25–36, 2004. 2, 5

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, pages 611–625, 2012. 5

[4] X. Chen, J. Yang, Q. Wu, J. Zhao, and X. He. Directional high-pass filter for blurry image analysis. *Signal Processing: Image Communication*, 27:760–771, 2012. 3

[5] S. Cho and S. Lee. Fast motion deblurring. In *ACM Transactions on Graphics (TOG)*, volume 28, page 145. ACM, 2009. 1, 2, 3, 4, 5

[6] B. Horn and B. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1

[7] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (TOG)*, 29(4):30, 2010. 1, 2

[8] J. Lee, V. Baines, and J. Padget. Decoupling cognitive agents and virtual environments. In *Cognitive Agents for Virtual Environments*, pages 17–36. Springer, 2013. 2

[9] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. In *ACM Transactions on Graphics (TOG)*, volume 27, page 71, 2008. 1, 2

[10] W. Li, D. Cosker, M. Brown, and R. Tang. Optical flow estimation using laplacian mesh energy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2, 4

[11] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 5

[12] T. Portz, L. Zhang, and H. Jiang. Optical flow in the presence of spatially-varying motion blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1752–1759, 2012. 1, 2, 5, 7

[13] Y. Schoueri, M. Scaccia, and I. Rekleitis. Optical flow from motion blurred color images. In *Canadian Conference on Computer and Robot Vision*, 2009. 2

[14] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics (TOG)*, volume 27, page 73. ACM, 2008. 3

[15] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 1

[16] Y.-W. Tai and S. Lin. Motion-aware noise filtering for deblurring of noisy and blurry images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24, 2012. 3

[17] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black. Lessons and insights from creating a synthetic optical flow benchmark. In *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, pages 168–177, 2012. 5

[18] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2, 3, 5

[19] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 3, 4