UNIVERSITY OF CALIFORNIA

SANTA CRUZ

INVESTIGATION INTO OPTICAL FLOW PROBLEM IN THE PRESENCE OF SPATIALLY-VARYING MOTION BLUR

A thesis submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

MOHAMMAD HOSSEIN DARAEI HAJI TOOEI

June 2014

The Thesis of M. HOSSEIN DARAEI is approved:

Professor Peyman Milanfar, Chair

Professor Roberto Manduchi

Professor Alex Pang

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Investigation Into Optical Flow Problem in the Presence of Spatially-Varying Motion Blur

by

Mohammad Hossein Daraei

The problem of optical flow computation has various applications in Computer Vision, and serves as a key problem that has been well studied over the past decades. While most of the techniques for inferring optical flow are based on the brightness constancy assumption, various conditions including the presence of motion blur evidently violate this fundamental presumption. In low illumination scenarios and other conditions under which the shutter must be kept open for a relatively long interval, motion blur artifacts are inevitable. If the source image and the target image appear to be dissimilar due to different blur kernels, traditional methods will fail to achieve accurate results. After exploring advantages and shortcomings of various optical flow methods, e.g. CLG [9], Black-Anandan [6], and BlurFlow [19], we address the problem of optical flow in the presence of motion blur. In particular, we present a new approach that considers constructing a new pair of blurred frames, followed by regular optical flow computation. The proposed method, MB-CLG, eliminates the effect of non-uniform blur levels over the sequence. A proof is also provided to show the estimated flows are roughly equal to the ground truth flows that match the latent frames. The key observation is that if we applied the blur functions of the source image to the target image and vice versa, the brightness constancy assumption would be valid for the new frames. The proposed method employs a coarse-to-fine approach, in conjunction with a smoothness matrix to account for moving objects and occluded regions. Rather than warping frames or precomputing a large grid of derivatives as in Portz et al [19], MB-CLG directly warps the flows in the optimization process. This leads to lower computational cost, and requires less data storage. Based on the results for various synthetic sequences, MB-CLG outperforms existing optical flow algorithms in the sense of AAE, AEP and MSE.

Dedicated to my parents,

and my sister, Sara.

Acknowledgments

First of all, I would like to thank my supervisor, Professor Peyman Milanfar, for his help and support, and allowing me to explore this problem in much more depth, and in particular to focus on the problem of optical flow in the presence of motion blur.

I would also like to thank the members of my Master's Thesis Reading Committee, Professor Roberto Manduchi and Professor Alex Pang, not only for their help for reading this thesis, but also for being supportive and putting up with all of my last-minuteness. Graduate advisor Emily Greg also deserves much appreciation for helping me with guidelines on how to properly submit this work.

I must also mention my fellow members in MDSP lab: Hossein Talebi, Sujoy Biswas, Amin Kheradmand and Rob Sumner. Throughout the entire time of working on this project, there have been many theoretical and practical challenges for which they provided me with helpful ideas. For his comments on the optimization step, I would also like to thank my friend, Mohammad Sadegh Ebrahimi, who is currently a PhD student at Stanford University.

Above all, I believe my family deserve much gratitude, for their encouragement and support. None of this work would have been done without the love of my mother, Azita, my father, Ali Asghar, and my sister, Sara. They never stopped supporting me over the course of my graduate career.

Chapter 1

Introduction

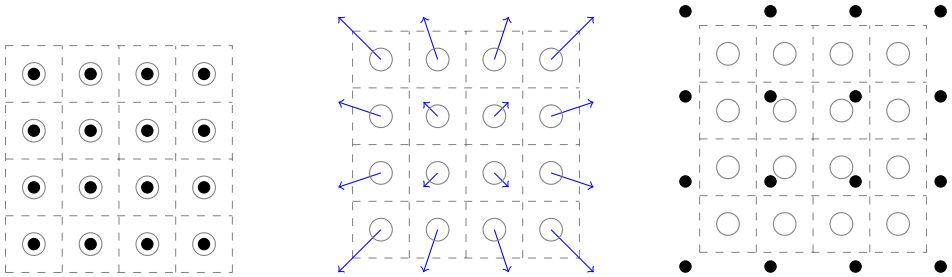
After an introduction on optical flow problem in the presence of motion blur, this chapter gives an overview of previous work and an outline of thesis.

1.1 Background

Optical flow of a sequence captures the dynamics of the underlying scene. Let us consider a moving camera recording video in a scene with various moving objects. Given two consequent frames f_i and f_{i+1} in a video sequence, we denote $\mathbf{w_{i,i+1}}$ as the displacement field (also referred to as the flow field) that warps f_i onto f_{i+1}. The vector field $\mathbf{w_{i,i+1}}(\mathbf{d}) : \mathbb{N}^2 \rightarrow \mathbb{R}^2$ assigns to each pixel at position $\mathbf{d} = (x, y)$ a vector $(u(x, y), v(x, y))$ that describes the motion at that particular pixel. The problem of optical flow can be addressed as finding the flow field such

$$\begin{aligned} f_{i+1}(\mathbf{d}) &\approx f_i(\mathbf{d} + \mathbf{w_{i,i+1}}(\mathbf{d})), \qquad \forall \mathbf{d} \\ or, \qquad f_{i+1}(x, y) &\approx f_i(x + u(x, y), y + v(x, y)), \qquad \forall (x, y) \end{aligned} \tag{1.1}$$

where $u(x, y)$ and $v(x, y)$ denote the horizontal and vertical components of $\mathbf{w_{i,i+1}}$. Fig. 1.1 explains the right hand side of Eq. 1.1, and how it varies with $\mathbf{w_{i,i+1}}$. This expression, also regarded as the *brightness constancy assumption*, can be quantitatively evaluated using a criterion like mean squared error. Solving Eq. 1.1 requires extra constraints, as a single equation per pixel is not sufficient to solve for two



(a) $f_i(\mathbf{d})$ (b) $\mathbf{w_{i,i+1}}(\mathbf{d})$ (c) $f_i(\mathbf{d} + \mathbf{w_{i,i+1}}(\mathbf{d})$

Figure 1.1: (a) shows a sample image $f_i(x, y)$ of size 4×4 represented as a grid of intensity values, (b) depicts an arbitrary flow field $\mathbf{w_{i,i+1}}(x, y)$ which corresponds to *scale motion*, and (c) demonstrates the warped version of f_i according to $\mathbf{w_{i,i+1}}$. Warping must be followed by interpolation to fill in white circles. For the optimal flow field, $f_i(\mathbf{d}+\mathbf{w_{i,i+1}}(\mathbf{d})$ will be identical to the next frame f_{i+1} after interpolation.

unknowns, $u(x, y)$ and $v(x, y)$. This problem is referred to as the *aperture problem*, and the extra constraint is usually the smoothness of flow field gradients, i.e. the smoothness constraint.

Solving the optical flow of a dynamic scene has a wide range of applications including but not limited to video denoising, deblurring, stereo, and structure-from-motion. In these applications, optical flow has to be estimated ahead, and generally on moving cameras with relative motion with respect to the scene or moving objects. Differential methods belong to the most widely used techniques for optical flow computation in image sequences. They can be classified into local methods such as Lucas-Kanade method [17], and global methods such as Horn-Schunck method [14] and its extensions [9, 8, 22]. However, traditional optical flow algorithms would fail to generate accurate flow fields if the motion was large enough to introduce motion blur artifacts. These methods are mostly based on the brightness constancy assumption in Eq. 1.1. This assumption is likely to be violated in the presence of motion blur or other unwanted artifacts that independently affect the frames. If the sequence is corrupted by motion blur, the target image is presumably blurred with a different blur kernel from the source image. In addition, standard visual tracking algorithms may fail to track regions of interest because motion blur cannot be modeled neither by lone geometric deformations, nor by lone photometric deformations. In this work, we relax the assumptions of Jin et al [15] about constant blur kernels, and consider

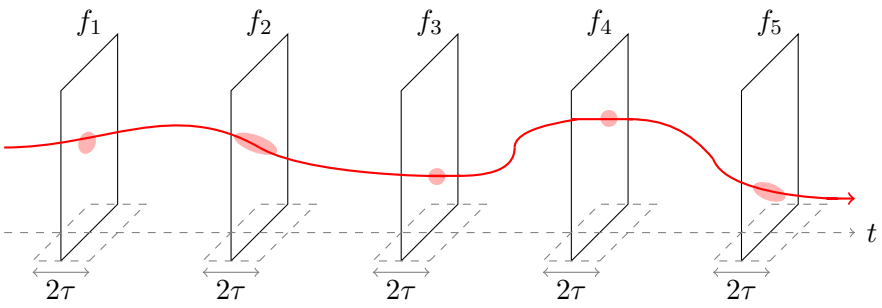


Figure 1.2: A moving object (red path) captured by a stationary camera with non-zero shutter time τ at different instances. If the object moves while the shutter is open, motion blur artifacts will arise (f_1,f_2, and f_5).

the general case of spatially-varying motion blur.

Motion blur is a common image and video distortion and a result of relative motion between the camera and the objects in the scene. Despite being regarded as a major deterioration, motion blur has practical interest in the measurement of the apparent motion in an image [4, 13, 11]. As the image acquisition model in Fig. 1.2 represents, practical cameras have to keep the shutter open for a finite time interval, denoted as 2τ. If either the camera or any object (red path) in the scene moves while shutter is open, a single point in the scene will project as a particular curve onto the sensor, hence introducing motion blur degradations. The strength of motion blur increases as either the shutter interval (exposure) or the relative motion between the camera and the scene increases. As evident in Fig. 1.2, motion blur affects the frames differently. For instance, an accurate estimate for $\mathbf{w_{3,4}}$, which projects f_3 onto f_4, could be simply achieved by employing regular optical flow method. Because the object was almost stationary in the acquisition interval of both f_3 and f_4, it appears similarly on each frame. However, finding $\mathbf{w_{2,3}}$ is challenging, for the moving object projects differently in f_2 and f_3 due to dissimilar blur kernels. As a concrete example, Fig. 1.3 provides two synthetic sequences f_i and g_i. The former is referred to as the latent/unblurred sequence ($\tau = 0$), and the latter is corrupted with motion blur ($\tau = 8$). Evidently, matching frames g_3 and g_4 is not as trivial as matching f_3 and f_4.

Traditionally, motion blur has been treated as an undesirable artifact that should be removed prior to any processing. Motion blur is in most realistic cases

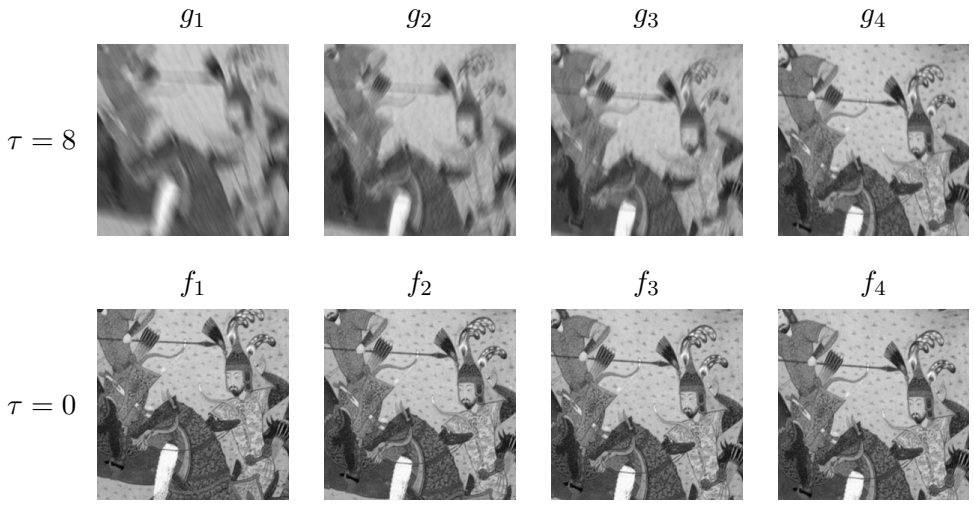


Figure 1.3: Two sets of artificially generated sequences, (top) motion blurred sequence g_i with $\tau = 8$, and (bottom) latent unblurred sequence f_i with $\tau = 0$.

a spatially-varying degradation. In order to remove this artifact, one should either have a knowledge of the underlying blur kernel or identify and estimate it. In the spatially-varying case, the identification of the blur is a difficult task, and blind deconvolution even in the case of shift invariant kernels is an ill-posed problem, and requires huge amounts of computation. Traditional methods usually couple optical flow with deblurring in order to correctly account for the effects of motion blur [24]. Such coupling is usually appropriate, but computationally wasteful when visual tracking is the lone objective. In order to avoid deblurring, Jin et al [15] propose to match regions by blurring them. To tackle this problem, one must take the advantage of the commutativity property of convolution. Although this property would not hold if the kernels were spatially-varying, it is shown that this property is still locally valid if the changes in kernels are also locally small. This property enables us to avoid applying computationally expensive methods for performing blind deconvolution.

This work particularly studies the problem of optical flow computation in the presence of spatially-varying motion blur. The proposed approach (MB-CLG) copes with this problem by filtering the source and the target images in order for the brightness constancy assumption to hold. For the sake of simplicity, we avoid image deblurring, warping intermediate frames, or precomputing a large grid of derivatives

with respect to warps. Instead, we present the idea of *warping a flow* at intermediate levels of pyramid. Given a video sequence with frames f_i, the proposed method constructs a Gaussian pyramid for each image. Starting from the coarsest level, MB-CLG employs a baseline method to align each image with previous and next frame. After estimating flows at a single level over the whole sequence, these flows are warped onto the coordinates of nearby frames and are exploited to blur them accordingly. We also address how to each flow field a unique motion blur function is associated. This additional motion blur is to ideally balance the effect of different spatially-varying blur kernels on different frames, and to compensate for the violation of brightness constancy assumption. Then, we exploit these artificially blurred frames when processing finer levels of the pyramid, and readdress the initial problem as solving optical flow for the new pair of frames. The baseline method we employ is a modified implementation of the Combined Local-Global (CLG) method proposed by Bruhn and Weickert [9]. They define a new energy functional incorporating both the data fidelity and the smoothness terms with nonquadratic penalizers, and minimize it using Euler-Lagrange equations and Conjugate Gradient (CG) method.

In the results section we use common metrics for the evaluation of estimated flow fields with respect to the ground truth, e.g. Average Angular Error (AAE), Average Endpoint Error (AEP) and Mean Squared Error (MSE) between the target and the warped source frames. Despite Portz et al [19] argue there could be no unique ground truth flow associated with a sequence corrupted by motion blur, we investigate and show the existence of such flow. The ground truth flow is defined as the flow that would match the frames if the exposure time was zero. First, we synthesize unblurred sequences f_i for various scenarios, and generate a ground truth flow associated with each one. Then we artificially add motion blur, and give as input the blurred sequence g_i to different techniques to estimate flow fields. Then we compare the results of each method with the ground truth flows that match the unblurred sequence. We will also visualize flows in conjunction with their divergence, as we will point out the divergence of a flow field demonstrates potential occluded regions and matching errors introduced by motion blur. We compare our results

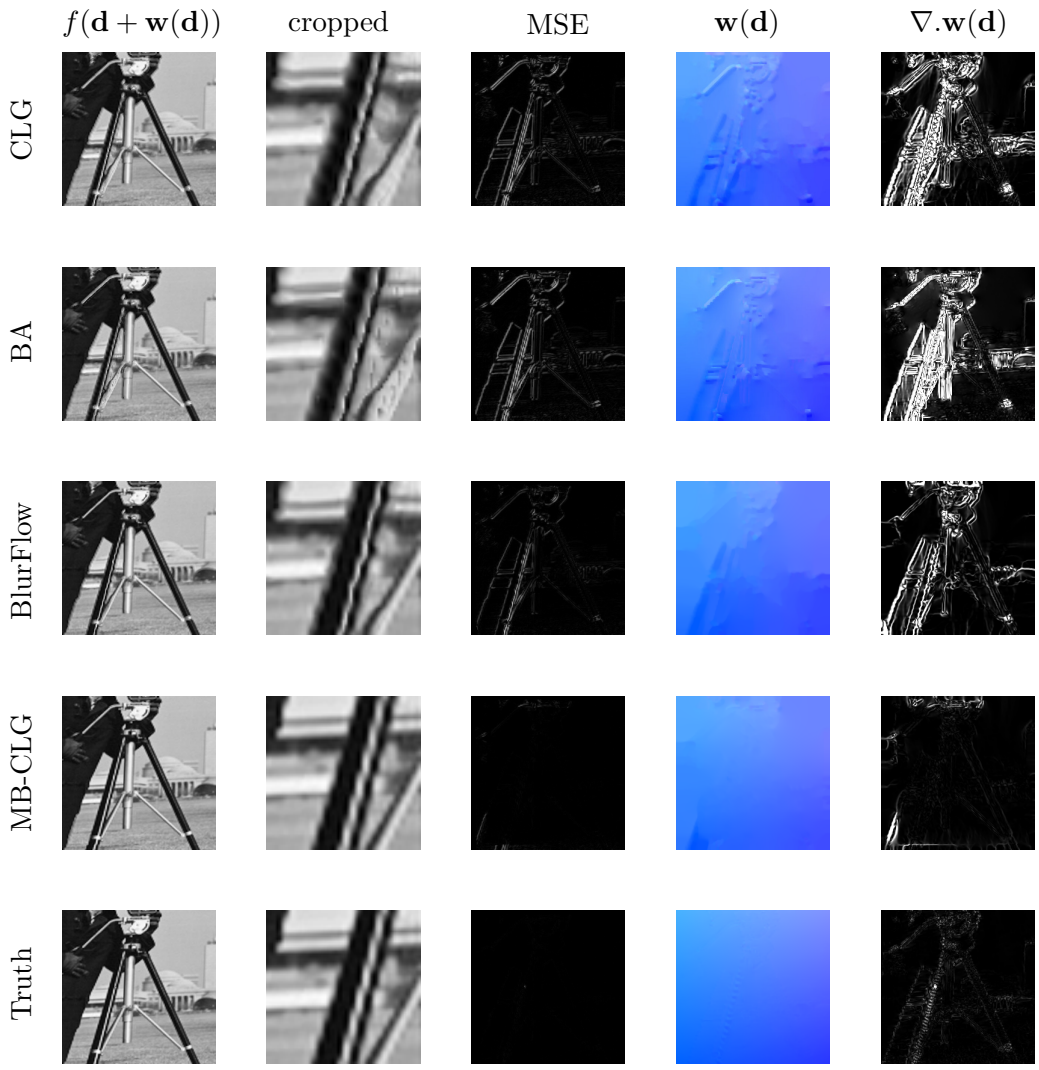


Figure 1.4: Results for CLG, , BA, BlurFlow, MB-CLG, and Ground Truth flows. Column (1) represents warped images using estimated flows, and a zoomed patch is depicted in column (2), while column (3) shows MSE between warped frame and the next frame, column (4) visualizes estimated flows, and column (5) is the corresponding flow field divergence.

in each scenario with those of other methods. Fig. 1.4 depicts a sample image warped based on (a) CLG [9], (e) MB-CLG and (i) ground truth flow fields. Third column corresponds to absolute error between warped frame and next frame, fourth column includes flow visualization, and the last column represents the divergence of the estimated flow fields. As an example of methods that do not consider motion blur, CLG introduces wrinkles in the estimated flow, as it tries to match sharp and blurred regions forced by brightness constancy assumption. Second column depicts magnified patches cropped from the warped images for each method. As it is evident, traditional methods, e.g. CLG and BA, result in deformed regions and wrinkle artifacts, while blur-aware methods, e.g. BlurFlow and the proposed method, MB-CLG, achieve better results.

1.2 Related Work

Many successful techniques have been proposed for optical flow computation over past decades. In their well-known paper, Lucas and Kanade [17] proposed an efficient local differential method in order to compute optical flow, parametrized by either a single translation or a homography. Their method is robust against noise, but does not generate global and dense flows. Horn and Schunck [14] proposed another differential method that achieves global and dense flow fields. They introduce a smoothness constraint in order to cope with the *aperture problem*, and minimize the Euler-Lagrange equations associated to their objective functional. However, their method was not as robust as that of Lucas and Kanade against noise. There have been many extensions to Horn and Schunck [14] in order to improve its robustness against noise. For an overview of these extensions, we refer the reader to the paper *Secrets of Optical Flow Estimation and Their Principles* [22]. Bruhn and Weickert [9] address how to combine Lucas-Kanade and Horn-Schunck methods, in order to improve robustness of global algorithms. They modify the regular objective functional of Horn-Schunck by adding non-quadratic penalizers and Gaussian smoothing. Sun et al [22] derive a novel objective function incorporating median filtering of intermediate flow fields for finer results. Brox et al [8] avoid conventional linearizations

in the brightness constancy assumption, and present a numerical scheme based on two nested fixed point iteration that results in smaller angular errors. Baker et al, [2] introduce a new set of benchmarks and evaluation methods, as well as visualization tools for flow fields. Black and Anandan [6] release the classic single motion assumption, and address the violation of brightness constancy and smoothness assumptions due to multiple motions. Bhat et al, [5] consider the problem of piecewise image registration in the presence of multiple large motions. They proceed by extracting and matching features, fitting fundamental matrices and solving a multi-label optimization. But, they do not address the effect of motion blur. Sand and Teller [20] propose a new motion estimation method for videos that exploit a set of particles. They also efficiently detect and handle occluded regions. Aside from *differential methods*, PatchMatch algorithm [3] is an example of *feature-based methods*, and finds k nearest neighbor patches in the target image over a range of different scales, rotations and translations.

Other authors including Chen and Nandhakumar, [11], Dai and Wu [13] and Schoueri and Scaccia [21], address the problem of estimating motion by looking at a single motion blurred frame. Despite similarities, their purposes and objective functions are intuitively different from the problem we address. Ben-Ezra and Nayar [4] also introduce a new hybrid camera that can measure its own motion during image acquisition. Then they use these motion information to eliminate motion blur degradations.

In general, methods that try to cope with the problem of optical flow computation in the presence of motion blur exploit the relation between motion and motion blur, but they mostly are coupled with image deconvolution during optimization, e.g. [24, 7, 28, 12]. Tull and Katsaggelos [24] formulate an objective function that restores motion blurred images as it estimates the underlying flow fields, and point spread functions. They only consider the case of constant velocity and translational motion, which cannot be extended to spatially-varying motion blur. Yuan et al, [28] consider the problem of aligning a pair of blurred and non-blurred images, under the assumption that the motion blur kernel is *spatially invariant* and *sparse*. Using a

regularizer enforcing sparseness, they keep refining their estimates for both the kernel and motion parameters at the same time. Jin et al, [15] argue that coupling motion estimation with image deblurring is computationally wasteful when visual tracking is the lone objective. They propose a novel method that avoids deconvolution, and matches regions by blurring them - taking the advantage of commutative property of convolution. Mei and Reid [18] extend the work of Jin et al, [15] from translational blur to any constant-velocity blur, and try to improve the overall computation complexity in the minimization process. These methods are aimed to track a blurred patch, while we are to compute dense optical flow for a video sequence. Portz et al [19], however, consider the problem of computing a dense optical flow using the commutative property of convolution. Estimated flows using a baseline method are computed, and then refined at the next stage. Although they avoid performing spatially-varying filtering during optimization, they numerically precompute a large grid of derivatives for the whole sequence and all possible locations with respect to flow vectors. Creating and computing such large grid not only takes extra time, but it also requires a big amount of storage.

1.3 Summary of Thesis

This thesis addresses the problem of estimating optical flow for a set of video frames in the presence of motion blur. The main contribution of this work is as follows. Instead of estimating numerical estimations of motion derivatives as in the work of Portz et al [19], we mathematically prove, justify by examples, and argue that in a coarse-to-fine pyramid one could take the flow estimates of one level and utilize them to cancel out the effect of motion blur. In order to cancel this effect, we intend to apply the spatially-varying motion kernel of f_i to f_{i+1} and vice versa. We take the estimated flows of a particular frame f_i, warp them onto the coordinates of f_{i+1} and use them to artificially add more blur. Once we add extra motion blur to each pair of frames, they will look similar despite in occluded regions. No matter how much sharp or blur they initially were, we argue that after this step brightness constancy assumption will hold. We argue that unlike most scenarios,

traditional methods would fail to accurately estimate optical flow when frames are motion blurred. The second contribution of this work is the idea of *warping a flow*, rather than warping frames. We'll also propose a simple way to account for occluded regions and handle moving objects.

1.3.1 Organization

Chapter 2 introduces the problem of optical flow, and explores traditional methods that tackle this problem, including Lucas-Kanade and Horn and Schunck. Then an overview of Combined Local-Global (CLG) [9] method and penalization of data term is provided, in addition to a summary of Euler-Lagrange equations and optimization of objective functions.

Chapter 3 describes the blur model in cameras, the problem statement for optical flow in the presence of motion blur, and the violation of brightness constancy assumption in the case of motion blur. Then BlurFlow [19] method is reviewed as a method with a similar approach for handling motion blur to our proposed method.

Chapter 4 describes the building blocks of the proposed method, and discusses how applying blur kernels of target and template images to each other would cancel the effect of different blur sizes. Therefore, brightness constancy assumption does hold again. Handling and accounting for occluded regions that may or may not come from the presence of moving objects will also be discussed in this chapter. Implementation notes for MB-CLG are also included in the last section.

Chapter 5 lists quantitative measures for evaluating estimated flow fields, i.e. AAE, AEP and MSE, as well as methods for visualizing flow fields for qualitative assessment. Having defined these metrics, Chapter 6 contains various experiments and quantitative results based on both synthetic and real datasets, as well as an investigation of noise effect, pre-processing and parameter tuning. For each sample video sequence, MB-CLG results are evaluated and compared with those of CLG [9], BA [6], BlurFlow (BF) [19]. Datasets that include moving objects are also generated and considered in the last section.

A proof for Theorem 4.8, details of *warp-the-flow* technique, a description of

flow interpolation method, the pseudo-code for the proposed method, and instructions on installing and using the developed software are also given as appendices.

Chapter 2

The Problem of Optical Flow

In addition to briefly reviewing the basics of optical flow, introducing well-known Lucas-Kanade and Horn-Schunck methods, this chapter includes an overview of optimization techniques utilized, and the concept of penalizers that modify regular objective functions.

2.1 Problem Statement

Let's assume a video sequence $f_i(\mathbf{d})$ of size $M \times N$ and length T, where i represents the frame index, and $\mathbf{d} = (x, y)$ denotes the coordinates vector. Then, the problem of computing optical flow that matches frame i to frame j can be addressed as the minimization of this objective function,

$$\widehat{\mathbf{w_{i,j}}} = \underset{\mathbf{w_{i,j}}}{\operatorname{argmin}} \sum_{\mathbf{d}} \| f_i(\mathbf{d} + \mathbf{w_{i,j}}(\mathbf{d})) - f_j(\mathbf{d}) \|^2 \tag{2.1}$$

where $\widehat{\mathbf{w_{i,j}}}$ is the optimal displacement field that minimizes the mean squared error criterion. With a change of notation, the objective function can be expressed as

$$E_D(\mathbf{u}, \mathbf{v}) = \sum_{x,y} \| f_j(x, y) - f_i(x + u(x, y), y + v(x, y)) \|^2 \tag{2.2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are respectively the horizontal and vertical components of $\mathbf{w_{i,j}}$, and the summation is over the $M \times N$ domain. This objective function - which is exploited

by various differential methods - comes from the *brightness constancy assumption*, denoted as:

$$f_i(x, y) \approx f_{i+1}(x + u(x, y), y + v(x, y)) \tag{2.3}$$

which states that a single point in a scene would move along a trajectory and would still have the same gray level in the next frame. This important assumption is of course violated in the case of change in the illumination, occlusion, optics blur, motion blur, additive noise and other deteriorations. Novel methods [9] cope with the noise by pre-smoothing the images by a Gaussian kernel, but other deteriorations that result in violations of brightness constancy assumption are not as easy to deal with. Assuming the components of the flow field are relatively small in Eq. 2.3, one can take a first-order Taylor approximation and get,

$$\begin{aligned} f_{i+1}(x + u, y + v) &\approx f_i(x, y) + \frac{\partial f_i}{\partial x}u + \frac{\partial f_i}{\partial y}v + \frac{\partial f_i}{\partial t} \\ &\rightarrow \frac{\partial f_i}{\partial x}u + \frac{\partial f_i}{\partial y}v + \frac{\partial f_i}{\partial t} = f_{i,x}u + f_{i,y}v + f_{i,t} = 0 \end{aligned} \tag{2.4}$$

where $f_{i,x}$ and $f_{i,y}$ are derivatives of f_i along horizontal and vertical directions, and can be estimated practically using simple shift-invariant filters. $f_{i,t}$ is also the derivative along t and is usually served as the difference between $f_{i+1}(x, y)$ and $f_i(x, y)$. Of course, this approximation is valid if and only if u and v are small, while this might not be the case. Therefore, by establishing a coarse-to-fine approach using a Gaussian pyramid, we ensure the changes in displacement field from one level to the next are small enough so this approximation is valid. Eq. 2.4 is called the *optics flow constraint*, and would turn Eq. 2.2 into,

$$E_D(\mathbf{u}, \mathbf{v}) = \sum_{x,y} \|f_{i,x}u + f_{i,y}v + f_{i,t}\|^2 \tag{2.5}$$

which is the data fidelity term in the ultimate objective function. Obviously, this objective function when minimized would only give the flow component perpendicular to the edges. In fact, this single equation is not sufficient to solve both u and v, and it could only compute a linear combination of u and v, a vector parallel to

$\nabla f_i = (f_{i,x}, f_{i,y})$. This so called *aperture problem* should be coped with using some extra constraints on the smoothness of displacement field. In addition, minimizing E_D would not result in any u and v in regions with no or low texture.

2.2 Traditional optical flow algorithms

In this section, brief introductions for various traditional methods are given. In particular, we study the objective functions proposed by Lucas-Kanade [17], Horn-Schunck [14], Combined local-global [9], and Black-Anandan [6]. This section is followed by an overview of optimization procedute for optical flow functionals. Respectively denoted by **CLG** and **BA**, Combined local-global and Black-Anandan methods are employed in Chapter 6 for comparing results and evaluations.

2.2.1 Lucas-Kanade method

Lucas-Kanade method [17] is one of the most widely used methods for optical flow estimation. In order to cope with the aperture problem, Lucas and Kanade in their differential method that lies under *local techniques*, assume that the flow is essentially small, constant and the same as (u_i, v_i) in a local neighborhood $\mathbf{\Omega}_\rho$ of size ρ about the pixel $p_i = (x_i, y_i)$. Constructing a Gaussian pyramid and taking a coarse-to-fine approach helps the assumption of small displacement vectors to be valid. For each p_i, they write Eq. 2.4 for all the pixels in that neighborhood, to form an over-determined system of $N = \rho^2$ equations and two unknowns, i.e., u_i and v_i. Finally, the Lucas-Kanade method obtains a compromise solution according to the least squares criterion,

$$E_{i,LK}(u_i, v_i) = \sum_{\mathbf{q}\in\mathbf{\Omega}_\rho} [f_x(\mathbf{q})u_i + f_y(\mathbf{q})v_i + f_t(\mathbf{q})]^2 \tag{2.6}$$

Let us start by writing Eq. 2.4 for all q_j where q_j is a pixel in the neighborhood of p_i. In matrix form,

$$\mathbf{Av} = \mathbf{b} \tag{2.7}$$

where,

$$A = \begin{bmatrix} f_x(\mathbf{q_1}) & f_y(\mathbf{q_1}) \\ f_x(\mathbf{q_2}) & f_y(\mathbf{q_2}) \\ \vdots & \\ f_x(\mathbf{q_N}) & f_y(\mathbf{q_N}) \end{bmatrix}, \quad v = \begin{bmatrix} u_i \\ v_i \end{bmatrix}, \quad b = \begin{bmatrix} -f_t(\mathbf{q_1}) \\ -f_t(\mathbf{q_2}) \\ \vdots \\ -f_t(\mathbf{q_N}) \end{bmatrix} \tag{2.8}$$

where f_x, f_y and f_t are the partial derivatives of the image $f(\mathbf{d})$ with respect to x direction, y direction and time t. A solution for the over-determined system of Eq. 2.7 is given as,

$$\begin{aligned} &\mathbf{A}^T\mathbf{A}\mathbf{v} = \mathbf{A}^T\mathbf{b} \\ &\rightarrow \mathbf{v} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \end{aligned} \tag{2.9}$$

which is the minimizer of the objective function in Eq. 2.6. If $\mathbf{\Omega}_\rho$ belongs to a smooth region of the image, $\mathbf{A}^T\mathbf{A}$ will consequently tend to be singular and non-invertible, and Eq. 2.9 fails to estimate (u_i, v_i). This problem turns out to be the reason Lukas-Kanade method is listed under *local techniques* for optical flow. In practice, it is usually better to give more importance to the pixels that are closer to p_i. This modification also makes Lucas-Kanade method more robust against noise [1]. Therefore, they refine the solution of Eq. 2.9 to get,

$$\mathbf{v} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b} \tag{2.10}$$

where $\mathbf{W}$ is an $N \times N$ diagonal matrix containing weights $\mathbf{W}_{ii}$ to be associated to the equation of $\mathbf{q}_i$. The weights are computed by a Gaussian function of the distance between $\mathbf{p}_i$ and $\mathbf{q}_j$. This is equivalent to filtering the objective function with a Gaussian kernel,

$$E_{LK} = K_\rho * (f_{i,x}u + f_{i,y}v + f_{i,t})^2 \tag{2.11}$$

where K_ρ is a Gaussian kernel and ρ serves as an integration scale over which the main contribution of the least square fit is computed. When E_{LK} is minimized, a local flow field is yield which is robust against noise due to the behavior of smooth-

ing Gaussian kernel. However, Lucas-Kanade method fails to estimate the flow in smooth regions, and cannot compute a global and dense flow field for a pair of images.

2.2.2 Horn-Schunck method

As another type of differential techniques, Horn-Schunck [14] method of estimating optical flow is a *global* method that computes dense flow field over the whole image. In order to cope with both the aperture problem and texture-less regions, they make the assumption that displacement field is smooth. Thus, Horn-Schunck method tries to minimize distortions in flow and prefers solutions which show more smoothness. As an important feature, the flow information missing in inner parts of homogeneous objects is filled in from the motion boundaries. By introducing a smoothness regularization term of the form,

$$E_S(\mathbf{u}, \mathbf{v}) = \sum_{x,y} \|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{v}\|^2 \tag{2.12}$$

the algorithm will be robust against aperture problem and would fill in similar u and v to those of the neighbor pixels in regions without texture. $\nabla \mathbf{u}$ and $\nabla \mathbf{v}$ are respectively gradient vectors of horizontal and vertical components of the flow field. Minimization of Eq. 2.12 results in smooth discontinuities in the displacement field. Therefore, the ultimate objective function can be derived as,

$$E_{HS}(\mathbf{u}, \mathbf{v}) = E_D(\mathbf{u}, \mathbf{v}) + \alpha E_S(\mathbf{u}, \mathbf{v}) \tag{2.13}$$

where α serves as a *regularization parameter* that should be tuned in order to maintain a compromise between field smoothness and fidelity to data. Larger values of α lead to a smoother flow. The global energy functional is formulated as,

$$E_{HS} = \int_\Omega \|f_{i,x}u + f_{i,y}v + f_{i,t}\|^2 + \alpha(\|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{v}^2\|)dxdy \tag{2.14}$$

where Ω is the image domain. This functional can be minimized by solving the associated Euler-Lagrange equations - addressed in upcoming sections. Despite it yields a high density of flow vectors, Horn-Schunck method (and global methods in general) is more sensitive to noise than local methods. In order to minimize Eq. 2.14 one should consider the associated Euler-Lagrange equations. The optimization process in described in detail in Section 2.3.

2.2.3 Combined local-global method

Many authors have tried to improve and extend the objective function of Horn-Schunck method. In [22], the authors explore some of these modifications, their advantages and shortcomings. Wedel et al [25], suggest a non-linear pre-filtering of the images to reduce the influence of illumination changes, and median filtering of intermediate flow results once after each optimization step. Other extensions include texture decomposition [25], nonquadratic penalizers [6, 9], temporal averaging of image derivatives, graduated non-convexity to minimize non-convex energies [6], and perhaps most importantly coarse-to-fine estimation to deal with large motions [9]. Combined local-global (CLG) method, introduced by Bruhn and Weickert [9] is considered as a successful extension of Horn-Schunck method. They aim to combine different features from local and global methods, to devise a hybrid approach that benefits both from robustness against noise, and from a densely computed global flow field. In addition, CLG establishes a coarse-to-fine approach, with nonquadratic penalizers added to both data and smoothness terms.

Initially, Bruhn and Weickert [9] reformulate the problem with a new notation. They define,

$$\begin{aligned} \mathbf{w} &= (u, v, 1)^T, \\ |\nabla \mathbf{w}|^2 &= |\nabla u|^2 + |\nabla v|^2, \\ \nabla_3 f &= (f_x, f_y, f_t)^T, \\ J_\rho(\nabla_3) &= K_\rho * (\nabla_3 f \nabla_3 f^T) \end{aligned} \tag{2.15}$$

where K_ρ is a Gaussian kernel specified by ρ. Then, they use this notation to express

the objective function of Lucas-Kanade in Eq. 2.11 as,

$$E_{LK}(\mathbf{w}) = \mathbf{w}^T J_\rho(\nabla_3 f)\mathbf{w}, \tag{2.16}$$

while the Horn-Schunck can be reformulated as minimizing,

$$E_{HS}(\mathbf{w}) = \int_\Omega (\mathbf{w}^T J_0(\nabla_3 f)\mathbf{w} + \alpha|\nabla \mathbf{w}|^2)dxdy \tag{2.17}$$

Comparing Eq. 2.16 and Eq. 2.17 suggests a natural way to extend the Horn-Schunck functional to the desired CLG functional. They simply replace the matrix $J_0(\nabla_3 f)$ by the structure tensor $J_\rho(\nabla_3 f)$ with some integration scale $\rho > 0$. Although in their final implementation this filtering is limited to the pixels in a 4-neighborhood, their results outperform those of Lucas-Kanade and Horn-Schunck methods. Therefore, they propose to minimize the functional

$$E_{CLG}(\mathbf{w}) = \int_\Omega (\mathbf{w}^T J_\rho(\nabla_3 f)\mathbf{w} + \alpha|\nabla \mathbf{w}|^2)dxdy \tag{2.18}$$

They are actually taking the idea of smoothness term from Horn-Schunck method, and Gaussian filtering of the data term from Lucas-Kanade method. In our previous notation, Eq. 2.18 can be expressed as

$$E_{CLG} = \int_\Omega K_\rho * (f_{i,x}u + f_{i,y}v + f_{i,t})^2 + \alpha(\|\nabla u\|^2 + \|\nabla v\|^2)dxdy \tag{2.19}$$

which in a similar way as Horn-Schunck functional must be restated as Euler-Lagrange equations, and numerically minimized by conjugate gradient.

In order to render the method more robust against outliers in both the data and the smoothness term, CLG replaces the traditional quadratic optimization by nonquadratic optimization [26]. In this approach, outliers are less severely penalized than in quadratic approach, and the results are more accurate at locations with flow discontinuities [9]. Thus, they add nonquadratic penalizers for the data and the smoothness terms. In order to maintain well-posedness during minimization, the authors first limit themselves to the set of convex penalizers, then pick the one

proposed by Charbonnier et al [10], and change the functional as,

$$E_{CLG}(u,v) = \sum_{x,y} \psi(K_\rho * (f_{i,x}u + f_{i,y}v + f_{i,t})^2) + \alpha \sum_{x,y} \psi(\|\nabla u\|^2 + \|\nabla v\|)^2 \quad (2.20)$$

where,

$$Charbonnier: \quad \psi(s^2) = 2\beta^2\sqrt{1 + \frac{s^2}{\beta^2}} \quad (2.21)$$

is the Charbonnier nonquadratic penalizer, and β denotes a scale parameter. Under a wise choice of β and when coupled with a smoothness term, the objective function in Eq. 2.20 will be convergent when minimized. For both data conservation and smoothness terms, we use $\beta = 0.001$ as also suggested in many differential methods [19, 9, 22]. Black and Anandan [6], however, employ a non-convex penalty function, referred to as Lorentzian or Perona-Malik penalizer,

$$Lorentzian: \quad \psi(s^2) = \epsilon^2 log(1 + \frac{s^2}{\epsilon^2}) \quad (2.22)$$

In [22], Sun et al express the superiority of Charbonnier over Lorentzian [6] (non-convex), and investigate a generalized version of Charbonnier penalty function of the form

$$Generalized\ Charbonnier: \quad \psi(s^2) = (s^2 + \beta^2)^a \quad (2.23)$$

that is analogous to the Charbonnier penalizer when $a = 0.5$, and non-convex when $a < 0.5$. They optimize the parameters again, and find a slightly non-convex penalty with $a = 0.45$ to perform consistently better than the Charbonnier penalty. However, in our proposed method we exploit the regular Charbonnier penalizer. One reason is that non-convex functions are more difficult to optimize, causing the optimization process to find a poor local minimum. Fig. 2.1 depicts different penalty functions for different parameters. Note, that when $n = 0.5$, Generalized Charbonnier simplifies to Charbonnier which is a convex function. It is evident that Lorentzian penalizer is non-convex regardless of ϵ, and Generalized Charbonnier is also non-convex for $n = 0.45$ and $n = 0.25$.

A coarse-to-fine approach, as another feature of CLG, will help our assump-

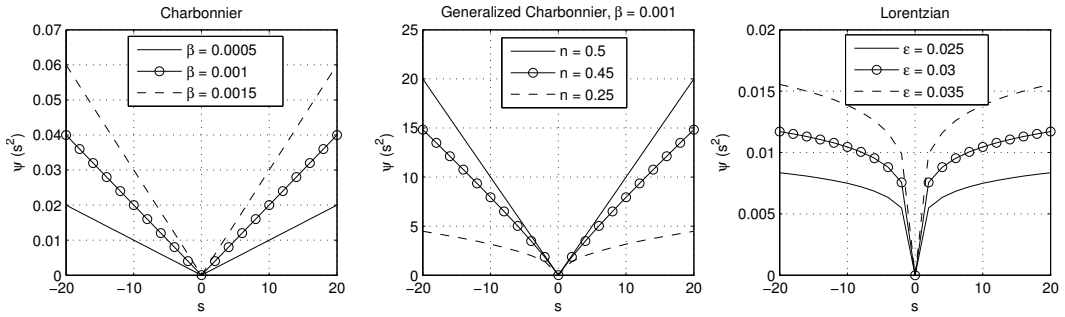


Figure 2.1: Different nonquadratic penalty functions for the spatial terms: Charbonnier [10] ($\beta = 0.0005$, $\beta = 0.001$, and $\beta = 0.0015$), Generalized Charbonnier [22] ($n = 0.35$, $n = 0.4$, and $n = 0.45$) with β fixed as 0.001, and Lorentzian [6], referred to as Perona-Malik penalizer ($\epsilon = 0.025$, $\epsilon = 0.03$, and $\epsilon = 0.035$)

tions to hold, and improve the optimization process. As mentioned previously, the first order Taylor approximation would be valid if u and v are small.

Bruhn and Weickert [9] propose constructing a Gaussian pyramid $\{f_i\}_l$, where pyramid level $l \in \{0, 1, \ldots, L-1\}$ for each image f_i prior to optical flow computation. Initially, $E(u, v)$ is minimized over the coarsest pyramid level $\{f_i\}_0$, resulting in an optimal flow field denoted as $\mathbf{w_0} = (u_0, v_0)$. Afterwards, $\mathbf{w_0}$ is upscaled and used to warp $\{f_{i+1}\}_1$ for the next level $l = 1$. While $\{f_i\}_1$ is not modified. Then, values for partial derivatives are updated, and $E(u, v)$ is minimized for the new level, resulting in another flow field $\mathbf{w_1}$. This step is repeated until the functional is optimized for the finest level $l = L-1$, and $\mathbf{w_{L-1}}$ is determined. Eventually, one can compute the global flow field as,

$$\mathbf{w} = \mathbf{w_0} + \mathbf{w_0} + \cdots + \mathbf{w_{L-1}} \tag{2.24}$$

which is the summation of all flow increments. As an implementation note, one should avoid warping a single image multiple times, as blur artifacts might show up. Instead, we compute $\mathbf{w_1}' = \sum_{i=0}^{l}$, and warp $\{f_{i+1}\}_l$ accordingly.

Combined local-global (CLG) method is served as our baseline method, and a modified version of an implementation by C. Liu [16] in C++ is at the heart of our proposed method. Despite superior results and various advantages, CLG does not consider the case of motion blurred video sequence. In that case, one major assump-

tion of CLG, the brightness constancy assumption, is no longer valid. Our main contribution is that we perform additional motion blur using estimated flow fields prior to moving to the next pyramid level. This additional step, discussed in detail in Chapter 4, guarantees that despite spatially-varying blur we still have brightness constancy assumption to hold, as well as its first-order Taylor approximation.

2.2.4 Black and Anandan method

Black and Anandan [6] argue that single motion assumption is violated in common situations, involving transparency, depth discontinuities, independetly moving objects, shadows, and specular reflections. Hence, this assumption must be relaxed. They provide insight on the reasons behind violation of brightness constancy and smoothness assumption caused by multiple motions. In particular, they modify the objective function of Horn-Schunck [14] by adding a Lorentzian penalty function to get,

$$\begin{aligned} E(u,v) = &\sum_{s\in S}[\lambda_D \psi_D((f_x u_s + f_y v_s + f_t)^2) \\ &+\lambda_S[\sum_{n\in\Omega_s} \psi_S((u_s - u_n)^2) + \sum_{n\in\Omega_s} \psi_S((v_s - v_n)^2)]] \end{aligned} \tag{2.25}$$

where λ_D and λ_S are data term and smoothness parameters, ψ_D and ψ_S denote the penalty functions for the data and the smoothness terms, respectively, and Ω_s is a 3×3 neighborhood about p_s. For historical reasons, they set $\psi_D = \psi_S$ to be the Lorentzian error norm. Then, they continue by employing Successive Over-relaxation (SOR) optimization technique. SOR is a variant of the Gauss-Seidel method for solving a linear system of equations. Accordingly, they form update equations for each iteration

$$\begin{aligned} u_s^{(n+1)} &= u_s^{(n)} - \omega \frac{1}{T(u_s)} \frac{\partial E}{\partial u_s} \\ v_s^{(n+1)} &= v_s^{(n)} - \omega \frac{1}{T(v_s)} \frac{\partial E}{\partial v_s} \end{aligned} \tag{2.26}$$

where ω denotes the over-relaxation parameter, and $T(u_s)$ and $T(v_s)$ are upper bounds on the second partial derivatives derived as,

$$T(u_s) = \frac{\lambda_D f_x^2}{\sigma_D^2} + \frac{4\lambda_S}{\sigma_S^2}, \qquad T(v_s) = \frac{\lambda_D f_y^2}{\sigma_D^2} + \frac{4\lambda_S}{\sigma_S^2} \tag{2.27}$$

with σ_D and σ_S denoting Lorentzian parameters for the data and the smoothness terms, respectively. Partial derivatives of E are also carried out as,

$$\begin{aligned} \frac{\partial E}{\partial u_s} =& \sum_{s\in S}[\lambda_D f_x \psi'((fxu_s + f_y v_s + f_t)^2) \\ &+ \lambda_S \sum_{n\in\Omega_s} \psi'((u_s - u_n)^2)] \\ \frac{\partial E}{\partial v_s} =& \sum_{s\in S}[\lambda_D f_y \psi'((fxu_s + f_y v_s + f_t)^2) \\ &+ \lambda_S \sum_{n\in\Omega_s} \psi'((v_s - v_n)^2)] \end{aligned} \tag{2.28}$$

which are numerically updated at each SOR iteration. The optimization stops once variations in u_s and v_s are smaller than a predefined value ϵ_0. Although Black-Anandan method accounts for multiple motion scenario and different reasons that might violate the brightness constancy assumption, they do not consider the case of motion blur deteriorations.

2.3 Optimizing the Objective Function

In order to minimize the aforementioned objective function, we start with the case of well-known Horn and Schunck [14] global energy functional without penalizers,

$$E_{HS} = \int_\Omega \|f_{i,x}u + f_{i,y}v + f_{i,t}\|^2 + \alpha(\|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{v}^2\|)dxdy \tag{2.29}$$

In order to minimize E_{HS} over the range of $u(x, y)$ and $v(x, y)$, one should first form and then solve the corresponding Euler-Lagrange equations. Then, we use some Conjugate Gradient (CG) iterations in order to optimize the convex functional.

Theorem 2.3.1 *Let $F(\alpha, \beta, \gamma)$ be a function with continuous first and second partial derivatives with respect to (α, β, γ), and let $I : C'[a, b] \rightarrow R$ be a function of form:*

$$I(x) = \int_a^b F(x(t), x'(t), t)dt, \qquad x \in C'[a, b] \tag{2.30}$$

if I has a local extremum at x_0, then x_0 satisfies the Euler-Lagrange equation:

$$\frac{\partial F}{\partial \alpha}(x_0(t), x_0'(t), t) - \frac{d}{dt}(\frac{\partial F}{\partial \beta}(x_0(t), x_0'(t), t)) = 0, \qquad t \in [a, b] \tag{2.31}$$

together with transversality conditions:

$$\frac{\partial F}{\partial \beta}(x_0(t), x_0'(t), t)\Big|_{t=a} = 0, \qquad \frac{\partial F}{\partial \beta}(x_0(t), x_0'(t), t)\Big|_{t=b} = 0 \tag{2.32}$$

The objective function in Eq. 2.29 is consistent with the assumptions of Theorem 2.3.1, and we'll have,

$$F([u\ v]^T, [\nabla_{\mathbf{d}} u\ \nabla_{\mathbf{d}} v]^T, \mathbf{d}) = [f_{i,x}u + f_{i,y}v + f_{i,t}]^2 + \alpha(\|\nabla_{\mathbf{d}} u\|^2 + \|\nabla_{\mathbf{w}} v^2\|) \tag{2.33}$$

where $\nabla_{\mathbf{d}}$ denotes gradient operator with respect to $\mathbf{d} = (x, y)$. By writing out the expression for $\frac{\partial F}{\partial \alpha}$ and $\frac{\partial F}{\partial \beta}$ and taking another gradient with respect to $\mathbf{d}$, the corresponding pair of Euler-Lagrange equations are expressed as,

$$0 = \nabla^2 u - \frac{1}{\alpha}(f_{i,x}^2 u + f_{i,x} f_{i,y} v + f_{i,x} f_{i,t}) \tag{2.34}$$

$$0 = \nabla^2 v - \frac{1}{\alpha}(f_{i,x} f_{i,y} u + f_{i,y}^2 v + f_{i,y} f_{i,t}) \tag{2.35}$$

where ∇^2 represents the Laplacian operator, $\nabla^2 = \partial_{xx} + \partial_{yy}$. As a major advantage over local methods, this formulation is embed with smoothness regularization and as mentioned previously, it would handle regions with small $\|\nabla f_i\|$. However, since global methods are deprived of a smoothing kernel K_ρ, they are not as robust as local methods, e.g. Lucas-Kanade [17] against noise. As Bruhn and Weickert [9] argue, we can introduce such smoothing kernel in the objective function to get Euler-Lagrange

equations,

$$0 = \nabla^2 u - \frac{1}{\alpha}(K_\rho * (f_{i,x}^2)u + K_\rho * (f_{i,x}f_{i,y})v + K_\rho * (f_{i,x}f_{i,t})) \tag{2.36}$$

$$0 = \nabla^2 v - \frac{1}{\alpha}(K_\rho * (f_{i,x}f_{i,y})u + K_\rho * (f_{i,y}^2)v + K_\rho * (f_{i,y}f_{i,t})) \tag{2.37}$$

which according to Bruhn and Weickert [9] would benefit from robustness against noise, as an inherent property of local methods. In practice, K_ρ is a 3×3 Gaussian kernel embedded into minimization process. By introducing the penalizers ψ into the cost function, the corresponding Euler-Lagrange equation would be of the form,

$$\begin{aligned}
0 &= div\{\psi_2'(\|\nabla u\|^2 + \|\nabla v\|^2)\nabla u\} \\
&\quad - \frac{1}{\alpha}\psi_1'(Q)(K_\rho * (f_{i,x}^2)u + K_\rho * (f_{i,x}f_{i,y})v + K_\rho * (f_{i,x}f_{i,t})) \\
0 &= div\{\psi_2'(\|\nabla u\|^2 + \|\nabla v\|^2)\nabla v\} \\
&\quad - \frac{1}{\alpha}\psi_1'(Q)(K_\rho * (f_{i,x}f_{i,y})u + K_\rho * (f_{i,y}^2)v + K_\rho * (f_{i,y}f_{i,t}))
\end{aligned} \tag{2.38}$$

$$Q = \begin{bmatrix} u & v & 1 \end{bmatrix} \begin{bmatrix} f_{i,x}^2 & f_{i,x}f_{i,y} & f_{i,x}f_{i,t} \\ f_{i,x}f_{i,y} & f_{i,y}^2 & f_{i,y}f_{i,t} \\ f_{i,x}f_{i,t} & f_{i,t}f_{i,y} & f_{i,t}^2 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{2.39}$$

which is utilized by Bruhn and Weickert [9] as their combined local and global (CLG) optical flow Euler-Lagrange equations. We use these functionals at each pyramid level, and minimize them using a predefined number of conjugate gradient (CG) iterations. Once minimization of this objective function is done for a specific pyramid level, we estimate spatially-varying motion blur kernels using estimated displacement fields, project the flows onto neighbor frame coordinates by warping them and applying the motion blur kernels. Then, we proceed to the next pyramid level.

Chapter 3

Optical Flow in the Presence of Motion Blur

After giving an overview of the conventional motion blur model in both continuous and discrete time cases, this chapter strives to justify the violation of *brightness constancy assumption* in the presence of motion blur, and readdress the problem. As a similar approach to our work, the method of Portz et al [19] for tackling this problem is also discussed in the last section.

3.1 Motion Blur Model

Fig. 3.1 summarizes the acquisition model in the continuous time that results in motion blur deterioration. Given a frame f_i, the exposure time τ, an ideal flow field $\mathbf{w_{i,i+1}} = (u_{i,i+1}, v_{i,i+1})$ that projects f_i onto f_{i+1}, and $\mathbf{w_{i,i-1}} = (u_{i,i-1}, v_{i,i-1})$

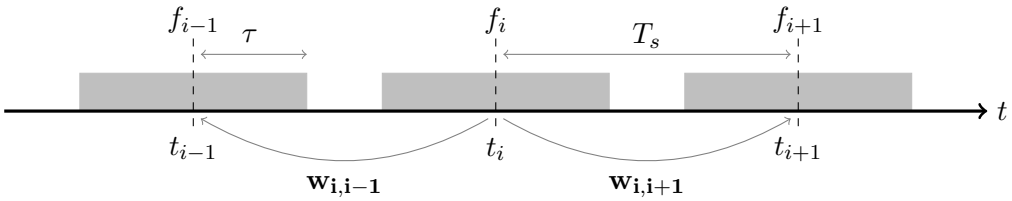


Figure 3.1: Acquisition model in continuous time domain. Motion blur is a result of non-zero exposure time τ, and can be expressed in terms of τ, T_s and a set of forward and backward warps that match neighbor frames

that projects f_i onto f_{i-1}, one can uniquely generate the estimated blurred frame g_i due to motion, and determine the underlying blur kernel. The expression for motion blur includes two terms: accounting for movements of pixels in f_i towards (1) previous frame f_{i-1} in $(t_i-\tau, t_i)$, and (2) next frame f_{i+1} in $(t_i, t_i+\tau)$. By assuming a linear path for pixel movements, we linearly estimate flow fields for the time interval between two actual frames. For $t_{i-1} < t < t_i$, interpolation of $f_i(\mathbf{d} + \frac{t-t_{i-1}}{T_s}\mathbf{w_{i,i-1}})$ is served as the estimate of intermediate frames. When $t = t_{i-1}$, it is the same as $f_i(\mathbf{d})$, and when $t = t_i$ it is $f_i(\mathbf{d} + \mathbf{w_{i,i+1}})$. Based on the definition of warps, $f_i(\mathbf{d} + \mathbf{w_{i,i+1}})$ is identical to $f_{i+1}(\mathbf{d})$, which is the next frame. In the same way, $f_i(\mathbf{d} + \frac{t-t_i}{T_s}\mathbf{w_{i,i+1}})$ describes the estimates of intermediate frames for $t_i < t < t_{i+1}$. Fig. 3.2 represents a general case where an object moves along the trajectory specified by the red curve. The blue dashed lines are linearized path segments for the moving object, and thick blue lines represent the blurred integrated image.

In discrete time domain, we use interpolation to generate fictitious frames, and develop a simple motion blur model. We start by addressing the model in continuous time domain,

$$\begin{aligned} g_i(x,y) &= \int_{t=0}^{\tau} f_i(\mathbf{d} + \frac{t}{T_s}\mathbf{w_{i,i-1}})dt + \int_{t=0}^{\tau} f_i(\mathbf{d} + \frac{t}{T_s}\mathbf{w_{i,i+1}})dt \\ &= \int_{t=0}^{\tau} f_i(x + \frac{t}{T_s}u_{i,i-1}, y + \frac{t}{T_s}v_{i,i-1})dt + \int_{t=0}^{\tau} f_i(x + \frac{t}{T_s}u_{i,i+1}, y + \frac{t}{\tau}v_{i,i+1})dt \end{aligned} \tag{3.1}$$

where T_s denotes the time interval between acquisition of frames, and the exposure time τ represents half the time the shutter is kept open. Obviously, $u_{i,i-1}$ and $v_{i,i-1}$ are also functions of $\mathbf{d} = (x, y)^T$. Eq. 3.1 approximates the movement trajectory of each point by two line sections, along which the integration is performed. This assumption is not accurate if rather than straight lines, displacement components are large and points move on trajectories with high curvature. However, it is simple and satisfactory enough in most practical cases.

For implementing this blur model, we will have to turn the integration into summation and rather than continuous time domain consider a discrete space. As a consequence, T_s and τ should necessarily be integers. Fig. 3.3 demonstrates the new

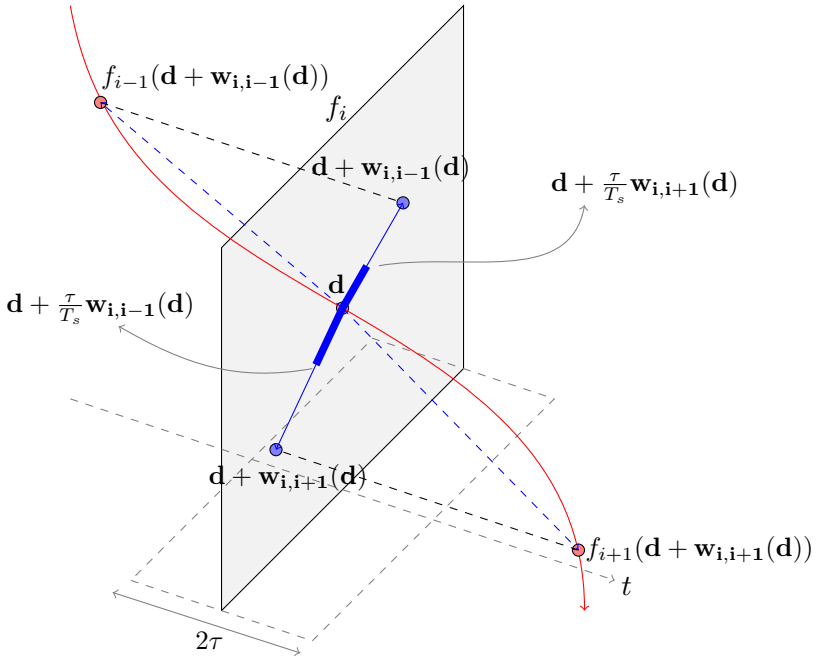


Figure 3.2: As an arbitrary object in the scene moves on a trajectory (red curve), we take a linear approximation (blue curve). Two thick blue lines are estimated blur kernels responsible for forward and backward warps.

model. If T_s denotes the number of discrete time instants, we can suppose a set of $T_s - 1$ fictitious frames between each actual frame. Line sections that approximate each point trajectory are again utilized to estimate flow fields for fictitious frames. Then, the aggregation of $2\tau + 1$ frames in the neighborhood of actual frames will serve as the motion blurred frame in the discrete time domain,

$$\begin{aligned} g_i(x,y) &= \sum_{t=0}^{\tau} f_i(\mathbf{d} + \frac{t}{T_s}\mathbf{w_{i,i-1}}) + \sum_{t=0}^{\tau} f_i(\mathbf{d} + \frac{t}{T_s}\mathbf{w_{i,i+1}}) \\ &= \sum_{t=0}^{\tau} f_i(x + \frac{t}{T_s}u_{i,i-1}, y + \frac{t}{T_s}v_{i,i-1}) + \sum_{t=0}^{T_s} f_i(x + \frac{t}{T_s}u_{i,i+1}, y + \frac{t}{T_s}v_{i,i+1})dt \end{aligned} \tag{3.2}$$

This can also be expressed with a different notation,

$$g_i(x,y) = B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d})) + B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d})) \tag{3.3}$$

where the function $B_{\mathbf{w}}(f(\mathbf{d})$ adds motion blur to frame $f(\mathbf{d})$ according to the integrals in Eq. 3.1 and based on flow field $\mathbf{w} = (u, v)$. For a simple notation, Eq. 3.3 can

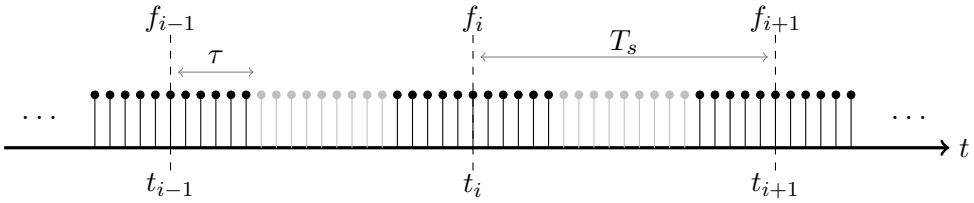


Figure 3.3: Acquisition model in discrete time domain. Simply, a discrete version of the model demonstrated in Fig. 3.1. In this case, $\tau = 5$ and $T_s = 20$ are integers.

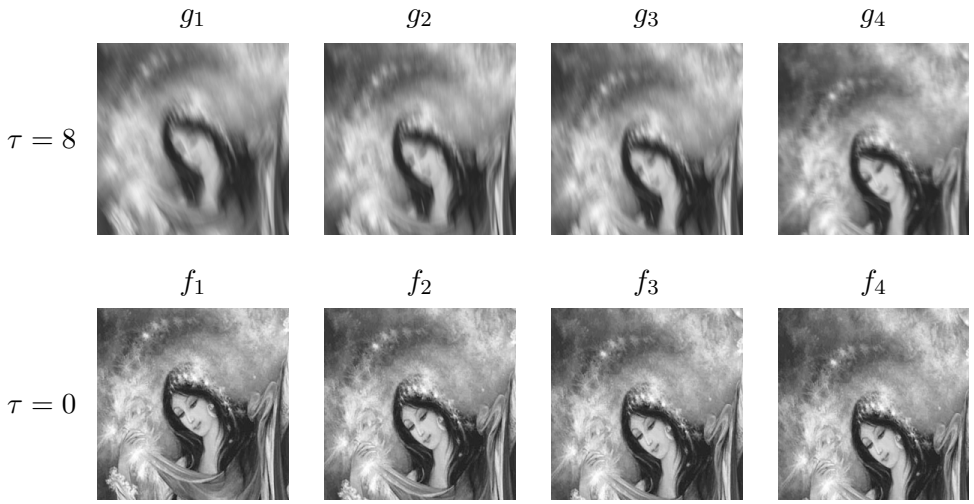


Figure 3.4: Generating motion blurred sequence g_i (top) from unblurred frames f_i (bottom) by using $B_{\mathbf{w}}(f_i)$ blurring functions, with $\tau = 8$ and $T_s = 20$.

also be expressed as the convolution of the clean image $f_i(\mathbf{d})$ with a spatially-varying kernel $B_{\mathbf{w_i}}$ that is uniquely determined in terms of $\mathbf{w_{i,i+1}}$ and $\mathbf{w_{i,i-1}}$. Therefore,

$$g_i(x, y) = (f_i * B_{\mathbf{w_i}})(x, y) \tag{3.4}$$

Let's assume for the rest of this discussion that τ and T are constant over the whole sequence. If we had variable exposure times, the same arguments would hold unless one should substitute for different values of τ in the expression of $B_{\mathbf{w}}(f(\mathbf{d})$. Fig. 3.4 portrays a synthetic sequence of unblurred frames and their corresponding blurred frames computed as $B_{\mathbf{w_i}} * f_i$.

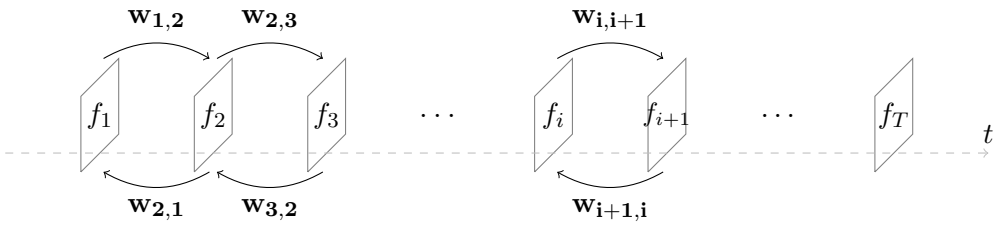


Figure 3.5: A general representation of the underlying unblurred seuqnece f_i, and the corresponding forward and backward flow fields.

3.2 Optical Flow in the Presence of Motion Blur

In this section, we address the problem of computing optical flow in the presence of motion blur. When frames are deteriorated with different spatially-varying blur kernels, *brightness constancy assumption* of the data term will be violated. For instance, one frame may still look sharper than the neighbor frame we are trying to match, even if we ignore occlusion and lighting conditions. For a formal problem statement, let g_i denote a video sequence of T frames with motion blur and shutter time τ, where $i \in \{1, 2, \ldots, T\}$. Let us also assume an ideal video sequence f_i without motion blur, as if $\tau = 0$. Then, given the blurred sequence g_i, the problem is to infer forward and backward flow fields that match the adjacent frames in the unblurred sequence f_i, i.e. $\mathbf{w_{i,i+1}}$, and $\mathbf{w_{i,i-1}}$. In our previous notation, the general problem can be represented as,

$$\begin{aligned}\{\widehat{\mathbf{w_{i,i+1}}}, \widehat{\mathbf{w_{i+1,i}}}\}_{i=1}^{T-1} = \underset{\{\mathbf{w}\}}{\operatorname{argmin}} & \sum_{i=1}^{T-1} \sum_{\mathbf{d}} \|\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d})) - f_{i+1}(\mathbf{d})\|^2 \\ & + \sum_{i=1}^{T-1} \sum_{\mathbf{d}} \|f_i(\mathbf{d}) - \Psi_{\mathbf{w_{i+1,i}}}(f_{i+1}(\mathbf{d}))\|^2\end{aligned} \tag{3.5}$$

where $\Psi_{\mathbf{w}}$ denotes warping operator followed by interpolation. Eq. 3.5 includes terms for both forward and backward flows, depending on the underlying f_i which is not given. Fig. 3.5 gives an outline of the latent sequence f_i and the associated flow fields. We aim to minimize Eq. 3.5 by employing an approach that does not require the direct knowledge of f_i. But for evaluation purposes, we utilize synthetic sequences that come with the latent sequence f_i and ground truth flows. Despite images are of different sharpness levels, traditional methods will try to warp

the source image to match the appearance of the target image. This will result in inaccurate estimates of the flow, deformation of flow field, and introduction of wrinkle artifacts which are better discriminated by looking at the divergence of estimated flows. In order to account for different blur kernels applied to each frame, Jin et al [15] and Portz et al [19] argue that the apparent motion blur itself depends on forward and backward flow fields that align a particular frame to neighbor frames. This relationship is approximated using Eq. 3.1, by assuming linear motion of all the points in the scene from one frame to the next. In addition to mean squared error for matching sequence f_i, we also employ metrics that evaluate the flows by directly comparing them with the ground truth flows, e.g. AAE and AEP. Chapter 5 gives an overview of these quality metrics.

3.2.1 BlurFlow Method

As a novel method to tackle the problem of optical flow in the presence of spatially-varying motion blur, we study BlurFlow by Portz et al [19] in this section. Initially, they run a baseline optical flow method on the video sequence to get an estimate for flow fields. Then, the estimates are refined by optimizing a new data term for the objective function. Although the commutative property for convolution does not hold when kernels are spatially-varying, they assume that the motion is locally smooth, so convolution is approximately commutative.

Let f_i and f_{i+1} denote two ground truth images where f_{i+1} is a warped version of f_i ignoring lighting, occlusion and interpolation errors, i.e., $f_i(\mathbf{d}) = f_{i+1}(\mathbf{d} + \mathbf{w_{i+1,i}})$. Then, let $g_i(\mathbf{d}) = f_i(\mathbf{d}) * B_i$ and $g_{i+1}\mathbf{d} = f_{i+1}(\mathbf{d}) * B_{i+1}$ be the observed images where B_i and B_{i+1} represent spatially-varying motion blur kernels described by Eq. 3.1, and $*$ denotes convolution with a spatially-varying kernel, and rather than a mathematical representation serves only as a notation. Based on the tracking work of Jin et al [15], they apply the blur functions of each observed image

to the other image,

$$\begin{aligned} k_i(\mathbf{d}) &= (g_i * B_{i+1})(\mathbf{d}) \approx (f_i * B_i * B_{i+1})(\mathbf{d}) \\ k_{i+1}(\mathbf{d}) &= (g_{i+1} * B_i)(\mathbf{d}) \approx (f_{i+1} * B_{i+1} * B_i)(\mathbf{d}) \end{aligned} \tag{3.6}$$

Then they argue that the blur kernels are themselves functions of $\mathbf{d}$, but they 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. As mentioned earlier, if the motion is locally smooth, the two convolutions will commute even if kernels are spatially-varying. In order to form their proposed data term, they define

$$k_z(\mathbf{d} = k_{i+1}(\mathbf{d} + \mathbf{w_{i+1,i}}) - k_i(\mathbf{d}) \tag{3.7}$$

Since there is a connection between the optical flow and the motion blur, the kernels used to generate k_i and k_{i+1} may vary with u and v. Therefore, we cannot simply get $\partial k_z/\partial u = k_x$ and $\partial k_z/\partial v = k_y$. The main contribution of Portz et al [19] then is to adapt the data term to handle this connection between the flow and the blur.

By writing $\mathbf{w_{i+1,i}} = (u + du, v + dv)$ and linearizing about du and dv, the new data term they consider is,

$$E_D(du, dv) = \sum_{\mathbf{d}} \Psi_D((k_z + \frac{\partial k_z}{\partial u} du + \frac{\partial k_z}{\partial v} dv)^2) \tag{3.8}$$

where

$$\begin{aligned} \left.\frac{\partial k_z}{\partial u}\right|_{\mathbf{d}} &= \frac{\partial}{\partial u}(k_{i+1}(\mathbf{d} + \mathbf{w_{i+1,i}}) - k_i(\mathbf{d})) \\ &= [\frac{\partial k_{i+1}}{\partial u} + \frac{\partial k_{i+1}}{\partial x}]\Big|_{\mathbf{d}+\mathbf{w_{i+1,i}}} - [\frac{\partial k_i}{\partial u}]\Big|_{\mathbf{d}} \end{aligned} \tag{3.9}$$

The partial derivative $\partial k_{i+1}/\partial u$ represents how the blurred image changes with respect to the blur induced by the flow, and the partial derivative $\partial k_{i+1}\partial x$ represents how the value $k_{i+1}(\mathbf{d} + \mathbf{w_{i+1,i}})$ changs with respect to the flow for a fixed blur. The partial derivative with respect to the vertical flow, $\partial k_z/\partial v$ is similarly defined.

The minimzation approach they use is the same as the baseline method, but with new blurred image derivatives, k_z, $\partial k_z/\partial u$ and $\partial k_z/\partial v$ in place of the standard image derivatives, $f_{i,t}$, $f_{i,x}$, and $f_{i,y}$. However, in order to actually minimize the objective function, we need numerical values for the new blurred image derivatives. They could theoretically compute the blurred images by performing line integrals at each pixel as in [15] and evaluate the derivatives with respect to the flow by differentiating the parametric kernels. Because this spatially-varying blur functions need to be repeatedly executed in each iteration of the optimization, they argue it would be very inefficient. Instead, they manage to precompute a large four-dimensional grid of $k_z(u, v, x, y)$ prior to optimization.

In order to find numerical values for k_z and its partial derivatives, Portz et al [19] in their supplementary attachment show it is sufficient to precompute a four-dimensional grid for $(g_i * B_{\mathbf{w}})(\mathbf{d}, \mathbf{w})$, i.e., a two dimensional space described by u and v axes, for which each point $\mathbf{w_0} = (u_0, v_0)$ represents a motion blurred version of $g_i(\mathbf{d})$ based on $B_{\mathbf{w_0}}$. We skip the details of constructing such grid, but mention they consider a predefined table of u and v values at which numerical values are derived. With the blurred image grid, one can compute a pixel $(g_i * B_{\mathbf{w}})(\mathbf{d}, \mathbf{w})$ at any desired (u, v) by bilinearly interpolating the grid at the point $\mathbf{w}$ over the (u, v) plane. After computing the grid, the minimization is performed with the new blurred images.

BlurFlow [19] proposes a new approach for solving optical flow in the presence of motion blur. However, not only precomputing such large grid for all of the frames in the sequence is computationally expensive, but also requires a large memory to store them. In addition, in order to perform bilinear interpolation over the grids when optimizing the new objective function, extra computations are required. For longer video sequences, the size of stored data in the memory is so large that the algorithm may crash in some experiments.

Chapter 4

The Proposed Method

This chapter describes the proposed method for solving the problem of optical flow in the presence of motion blur. In brief, Motion Blur Aware Combined A discussion on the proposed approach for coping with motion blur, an introduction on handling occlusions, a new method to account for moving objects and implementation notes for the proposed method are provided in this chapter.

4.1 Discussion

Given the shutter time τ and a set of motion blurred video frames g_i where $i \in \{1, 2, ..., T\}$, the problem of optical flow in the presence of motion blur is to find a set of forward and backward warps, $\mathbf{w}_{i,i+1}$ and $\mathbf{w}_{i+1,i}$, that would match corresponding latent sharp frames f_i (as if $\tau = 0$) with their previous and next neighbor frames.

Let's start by writing g_i and g_{i+1} in terms of their latent frames,

$$\begin{aligned} g_i(\mathbf{d}) &= B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d})) + B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d})) \\ g_{i+1}(\mathbf{d}) &= B_{\mathbf{w_{i,i+1}}}(f_{i+1}(\mathbf{d})) + B_{\mathbf{w_{i+1,i+2}}}(f_{i+1}(\mathbf{d})) \end{aligned} \tag{4.1}$$

However, we know $f_{i+1}(\mathbf{d})$ could be represented as,

$$f_{i+1}(\mathbf{d}) = \Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d})) \tag{4.2}$$

where $\Psi_{\mathbf{w}}(f_{\mathbf{d}})$ simply warps the image $f_{\mathbf{d}}$ according to the flow field $\mathbf{d} = (u, v)$ and accompanied by Bilinear or Bicubic interpolation. This representation is only valid for the latent frames f_i and would not align blurred frames g_i. Let's continue by applying $\Psi_{\mathbf{w_{i,i+1}}}$ to both sides of $g_i(\mathbf{d})$,

$$\begin{aligned}\Psi_{\mathbf{w_{i,i+1}}}\{g_i(\mathbf{d})\} &= \Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d})) + B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\} \\ &= \Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\} + \Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\end{aligned} \tag{4.3}$$

It is evident that warping a motion blurred frame is equivalent to warping both the latent frame and the underlying blur kernel, followed by performing motion blur. In other words,

$$\Psi_{\mathbf{w_1}}\{B_{\mathbf{w_2}}(f(\mathbf{d})\} = B_{\Psi_{\mathbf{w_1}}(\mathbf{w_2})}\{\Psi_{\mathbf{w_1}}(f(\mathbf{d})\} \tag{4.4}$$

where $\Psi_{\mathbf{w_1}}(f(\mathbf{d})$ is the warped version of $f(\mathbf{d})$ using $\mathbf{w_1}$, and $\Psi_{\mathbf{w_1}}(\mathbf{w_2})$ denotes a flow field resulted from warping a two-channel image $\mathbf{w_2} = (u_2, v_2)$ according to another flow field, $\mathbf{w_1}$. Warping a flow field is not as straightforward as warping a regular image, since $\mathbf{w_1}$ might have different values at the start point and the endpoint of vector field w_2. First, a regular $M \times N$ grid is warped using $\mathbf{w_1}$. Then, the endpoint coordinates of all the vectors in $\mathbf{w_2}$ are warped onto a new grid using $\mathbf{w_1}$. Finally, we can compute the resulting flow field by exploiting a non-uniform interpolation technique. If we use Eq. 4.4 to simplify RHS of Eq. 4.3, we will get,

$$\begin{aligned}\Psi_{\mathbf{w_{i,i+1}}}\{g_i(\mathbf{d})\} = B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d})\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d})\}\end{aligned} \tag{4.5}$$

which could be further simplified using Eq. 4.2 to yield,

$$\Psi_{\mathbf{w_{i,i+1}}}\{g_i(\mathbf{d})\} = B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{f_{i+1}(\mathbf{d})\} + B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{f_{i+1}(\mathbf{d})\} \tag{4.6}$$

which states if we used ground truth flows, warped and projected a motion blurred frame g_i onto the next frame it would be possible to represent it in terms of the latent frame f_{i+1} corrupted by some motion blur. Portz et al, [19] use this equation, define

blurred versions of $g_{i+1}(\mathbf{d})$ and $\Psi_{\mathbf{w_{i,i+1}}}\{g_i(\mathbf{d})\}$ and show that they should be equal. Then they form an objective function that must be minimized in a MSE sense, take the derivative with respect to flow fields and put it precomputed values. However, we avoid using $\Psi_{\mathbf{w_{i,i+1}}}\{g_i(\mathbf{d})\}$ or formulating a least squared error problem. Rather we convert the original optical flow problem into a new optical flow problem, that could be solved using conventional methods.

We define k_{i+1} to be the motion blurred version of g_{i+1} using warped blurring functions in RHS of Eq. 4.6, and k_2 to be the motion blurred version of $g_i(\mathbf{d})$ using warped blurring functions in the definition of $g_{i+1}(\mathbf{d})$ in Eq. 4.1,

$$\begin{aligned} k_i(\mathbf{d}) &= B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{g_i(\mathbf{d})\} + B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{g_i(\mathbf{d})\} \\ k_{i+1}(\mathbf{d}) &= B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{g_{i+1}(\mathbf{d})\} + B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{g_{i+1}(\mathbf{d})\} \end{aligned} \tag{4.7}$$

where $\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})$ and $\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})$ are flow fields that would match f_{i+1} with the previous and next frames, but projected onto the coordinates of f_i by warping them according to $\mathbf{w_{i+1,i}}$. In order to continue, we need to establish a relation between optical flow problem of $k_i(\mathbf{d}) - k_{i+1}(\mathbf{d})$ and that of $f_i(\mathbf{d}) - f_{i+1}(\mathbf{d})$.

Theorem 4.1.1 *If k_i and k_{i+1} are constructed based on Eq. 4.7, the problem of aligning them is the same as aligning latent frames f_i and f_{i+1}. If we substitute for g_i and g_{i+1} from Eq. 4.1 in the expression of Eq. 4.7 and expand, it becomes apparent that regardless of blur strength, $\Psi_{\mathbf{w_{i,i+1}}}(k_i)$ and k_{i+1} will be equal, assuming functions $B_{\mathbf{w}}(f(\mathbf{d})$ have commutative property.*

$$\|\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) - k_{i+1}(\mathbf{d})\|^2 \approx 0 \tag{4.8}$$

which suggests the flow field that aligns k_i with k_{i+1} as defined in Eq. 4.7, happens to be the same flow field that matches f_i with f_{i+1}. A proof for this theorem is given in Appendix A.

Generally, commutative property is violated for spatially-varying convolution. However, if the fluctuations in the underlying kernel are locally small, this property will be locally valid. Adopting a coarse-to-fine approach not only ensures

our first order Taylor approximation is valid, but it also helps the commutative property of $B_{\mathbf{w}}(f(\mathbf{d})$.

Finally, the problem can be reformulated as,

$$\begin{aligned}\{\widehat{\mathbf{w_{i,i+1}}}, \widehat{\mathbf{w_{i+1,i}}}\}_{i=1}^{T} = \underset{\{\mathbf{w}\}}{\operatorname{argmin}} &\sum_{i=1}^{T}\sum_{\mathbf{d}} \|\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) - k_{i+1}(\mathbf{d})\|^2 \\ &+ \sum_{i=1}^{T}\sum_{\mathbf{d}} \|k_i(\mathbf{d}) - \Psi_{\mathbf{w_{i+1,i}}}(k_{i+1}(\mathbf{d}))\|^2\end{aligned} \tag{4.9}$$

where the first term in the right hand side represents data fidelity term for forward flows, and the second term considers backward flows. Since k_i's also depend on both forward and backward flows in a non-convex manner, direct minimization of Eq. 4.9 is ill-posed and requires numerical computation of derivatives at certain $\mathbf{d}$'s throughout the sequence. However, at each level of the pyramid we can compute k_i's using the estimated and upscaled $\mathbf{w}$'s from the previous level, starting with a zero field as initialization. By assuming constant k_i's at each level, the objective function could be split and addressed as a set of regular optical flow computations,

$$\begin{aligned}\widehat{\mathbf{w_{i,i+1}}} &= \underset{\mathbf{w_{i,i+1}}}{\operatorname{argmin}} \sum_{\mathbf{d}} \|\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) - k_{i+1}(\mathbf{d})\|^2, \qquad \forall i \in \{1,2,...,T\} \\ \widehat{\mathbf{w_{i+1,i}}} &= \underset{\mathbf{w_{i+1,i}}}{\operatorname{argmin}} \sum_{\mathbf{d}} \|k_i(\mathbf{d}) - \Psi_{\mathbf{w_{i+1,i}}}(k_{i+1}(\mathbf{d}))\|^2, \qquad \forall i \in \{1,2,...,T\}\end{aligned} \tag{4.10}$$

which is the same as solving $2T$ optical flow problems for both forward and backward flows over the whole sequence. If we had negligible exposure time, $\tau \approx 0$, blurring operations in Eq. 4.7 would come down to identity functions, and k_i's would reduce to g_i's. Therefore, the proposed method would be the same as regular optical flow computation if there were no motion blur. Needless to say that objective functions in Eq. 4.9 should be optimized over the whole sequence at each pyramid level, embedded with proper smoothness constraints as regularization and penalizing functions $\psi(s^2)$. Then we use these updated warps in order to generate k_i's for the next level, and keep updating our estimates.

4.2 Occlusion and Moving Objects

One challenging aspect of optical flow estimation is how to accurately account for occlusions. In addition to other reasons that violate the brightness constancy assumption, many methods including that of Brox et al [8] use penalty functions Ψ in order to handle occlusions. However, using conventional robust functions to account for occlusions is not ideal [20, 27]. They are basically allowing the algorithm to fail if producing large errors, instead of designing a mechanism that would capture and group together the occluded pixels. Sand and Teller [20] use a combination of *flow divergence* and *pixel projection difference* to identify occluded regions. As pointed out earlier, the divergence of an optical flow reveals information about different motion boundaries and if a region is likely to be occluded. For an estimated optical flow $\mathbf{w} = (u, v)$,

$$div(\mathbf{w}(x, y)) = \frac{\partial}{\partial x}u(x, y) + \frac{\partial}{\partial y}v(x, y) \tag{4.11}$$

Sand and Teller [20] exploit a Bilateral filter [23] with predefined coefficients which incorporate not only Eulidean distance and radiometric difference, but also divergence and flow field differences. They intend to find occluded pixels and give them less weight when running a weighted moving average window,

$$\mathbf{w\prime}(x, y) = \frac{\sum_{x_1,y_1} \mathbf{w}(x_1, y_1, t)w(x, y, x_1, y_1, t)}{\sum_{x_1,y_1} w(x, y, x_1, y_1, t)} \tag{4.12}$$

Initially, pixels that are more likely to be in the occluded region are determined by thresholding the magnitude of flow field gradient. Then for a specific patch centered at (x, y) that includes a 21×21 neighborhood of (x_1, y_1), the weights are calculated

as,

$$
\begin{aligned}
w(x, y, x_1, y_1, t) = & N(\sqrt{(x - x_1)^2 + (y - y_1)^2}; \sigma_x) \\
& \times N(I(x, y, t) - I(x_1, y_1, t); \sigma_i) \\
& \times N(\sqrt{(u - u_1)^2 + (v - v_1)^2}; \sigma_m) \\
& \times r(x_1, y_1, t)
\end{aligned}
\tag{4.13}
$$

where the first two terms correspond to geometric and radiometric distance, third term measures differences in the estimated flow field and the last term includes weights based on a Gaussian function of flow divergence and pixel projection differences. They also argue that negative values in Eq. 4.11 indicate regions that are to be occluded in the adjacent frame, so they filter positive values in the flow divergence. However, their proposed method fails to properly spot occluded regions, since at each level of the pyramid we are matching filtered versions of the images. One other reason is that applying a bilateral filter directly to the flows would make it ambiguous for the algorithm to distinguish between *occluded* regions and regions that seem to be occluded but actually are *motion blurred boundaries*. Therefore, handling motion blur and occlusion simultaneously is not as straightforward as applying conventional occlusion detection methods.

The proposed method to handle occlusions and moving objects is in brief to reduce the effect of data term in regions with large divergence, and put more weight on smoothness term. Therefore, we replace the smoothing parameter α with a $M \times N$ matrix $\mathbf{A}$ which varies throughout the frame, and is point-wise multiplicated in the ultimate objective function,

$$
E(\mathbf{w}(x, y)) = E_D(\mathbf{w}(x, y)) + \mathbf{A}(x, y) E_S(\mathbf{w}(x, y)) \tag{4.14}
$$

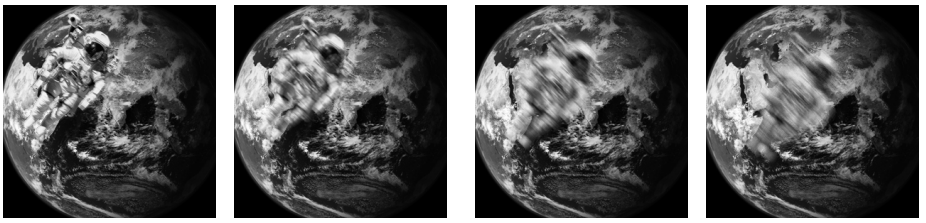
where E_D and E_S are respectively data fidelity and smoothness terms as discussed earlier. The smoothing matrix $\mathbf{A}$ is determined at each level of the pyramid, for

each pair of frames and prior to updating $\mathbf{w}$,

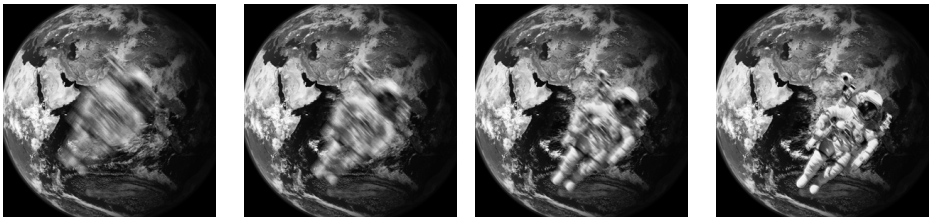
$$\mathbf{A}(x,y) = \alpha \times (K - (K-1)exp(\frac{-(div\mathbf{w})^2(x,y)}{2\sigma_d^2})) \tag{4.15}$$

where $\mathbf{w}$ denotes the estimated flow field at the previous pyramid level, $\alpha = 0.012$ is the regular smoothing parameter, and we set $\sigma_d = 0.4$ and $K = 10$ based on experimental observation of occluded scenarios. According to Eq. 4.15, $\mathbf{A}$ produces values close to default α in regions with low divergence, and results in larger values in regions with large divergence. Therefore, the algorithm relaxes the data term in regions that are more likely to be occluded, and looks at the flow field in the neighborhood to fill in occluded regions. This slight change enables the proposed method to outperform CLG [9], BA [6] and BlurFlow [19] in an AAE and AEP sense. In the Results section, we will test video sequences that include a moving and motion blurred object on a still background. We argue that, unlike other scenarios, mean squared error (MSE) is not a suitable evaluation metric, since in the presence of occlusion not even the ground truth flow perfectly matches two adjacent frames. Obviously, one frame contains regions of the background that are completely absent in the next frame. Thus, we stick with AAE and AEP as reliable evaluation metrics for this scenario.

Fig. 4.1 depicts first 8 frames of an artificially generated sequence which includes the image of a moving astronaut which is motion blurred, and some regions of each frame is obviously occluded in the next and previous frames. When generating the sequence, the ground truth forward and backward flows are also available. After estimating the flow fields using various methods, one can look at Fig. 4.3 which represents flow divergence maps for the estimated flows for frame # 6 using each method. The proposed method (MB-CLG) generates flows with the most similar divergence map to the ground truth. Since occluded regions result in large divergence in the flow, MB-CLG is capable of handling these regions by increasing the divergence and leaving other regions unchanged (the black region inside the astronaut body.) Fig. 4.2 represents the smoothness matrix $\mathbf{A}(x,y)$ for different frames of the sequence. Evidently, $\mathbf{A}$ returns large values at motion boundaries where the pixels



(a) frame #1 (b) frame #2 (c) frame #3 (d) frame #4



(e) frame #5 (f) frame #6 (g) frame #7 (h) frame #8

Figure 4.1: Sample frames of an artificially generated motion blurred sequence of a still background and a moving object - an astronaut.

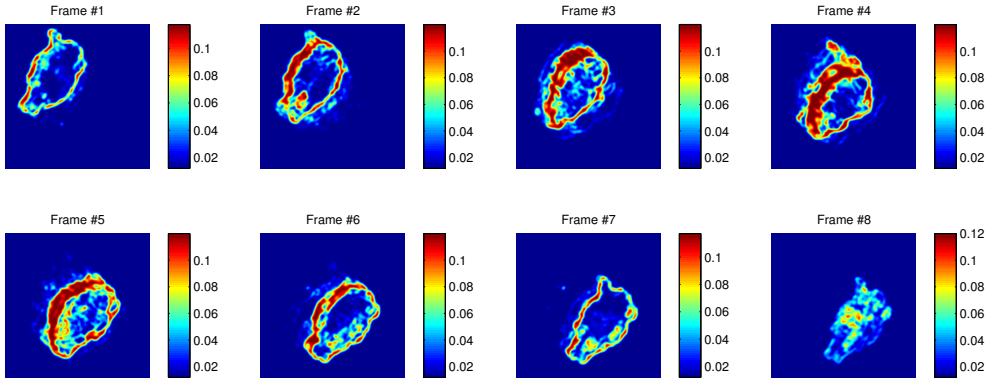


Figure 4.2: Smoothing matrix $\mathbf{A}(x, y)$ for the frames depicted in Fig. 4.1.

are most likely to be occluded. Directly visualizing flows also justifies the superiority of MB-CLG in terms of generating flows most similar to the ground truth. Fig. 4.4 demonstrates visualized flows generated by CLG [9], BA [6], BlurFlow [19] and the proposed method MB-CLG. Quantitative evaluations for this sequence and others are included in the Results section.

4.3 Implementation

In this section we discuss the implementation notes and techniques utilized to minimize objective functions in Eq. 4.10. As a brief overview, the proposed method alternates between two stages at each level in a coarse-to-fine pyramid: (1)

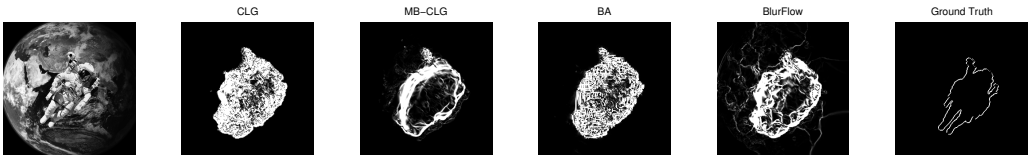


Figure 4.3: Divergence of estimated flow fields for the astronaut sequence, frame # 6

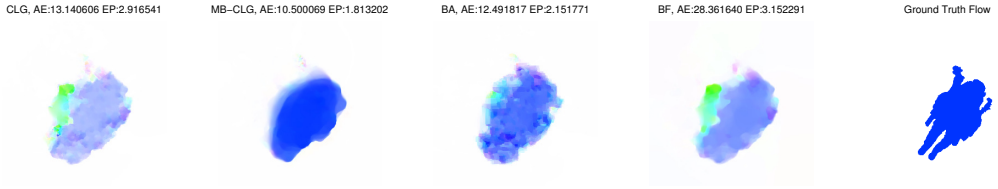


Figure 4.4: Visualized estimated flow fields for the astronaut sequence, frame # 6

estimating k_i's for all frames, and (2) utilizing them to refine flow fields. We construct a Gaussian pyramid of scale 0.75 for each motion blurred frame g_i, denoted as $\{g_i\}_l(\mathbf{d})$ and represented in Fig. 4.5. l denotes the pyramid level, $l = 1$ corresponds to the coarsest level, while $l = L$ corresponds to the finest level which is the same as g_i.

Accordingly, we define $\{k_i\}_l(\mathbf{d})$ to represent k_i's that we generate at each level. In a similar way, we define $\{\mathbf{w_{i,i+1}}\}_l$ and $\{\mathbf{w_{i+1,i}}\}_l$ to respectively denote forward and backward flows for lth pyramid level. We start by initializing flows for the first level,

$$\{\mathbf{w_{i,i+1}}\}_1 := \mathbf{0}, \qquad \{\mathbf{w_{i+1,i}}\}_1 := \mathbf{0}, \qquad \forall i \in \{1, 2, ..., T-1\} \tag{4.16}$$

At each level l and for each frame i, we check if upscaled flow fields from the previous level $l-1$ would actually result in a visible motion blur. If at a certain level, $\mathbf{w}(\mathbf{d})$ multiplied by exposure time $\frac{\tau}{2}$ is on average smaller than 1 pixel, the resulting motion blur would have no effect. In order to avoid unnecessary computations, we avoid applying motion blur at coarser levels if,

$$\frac{\tau}{M_l \times N_l} \sum_{\mathbf{d}} \uparrow \{\mathbf{w_{i,i+1}}\}_{l-1} + \uparrow \{\mathbf{w_{i,i-1}}\}_{l-1} \leq \gamma \tag{4.17}$$

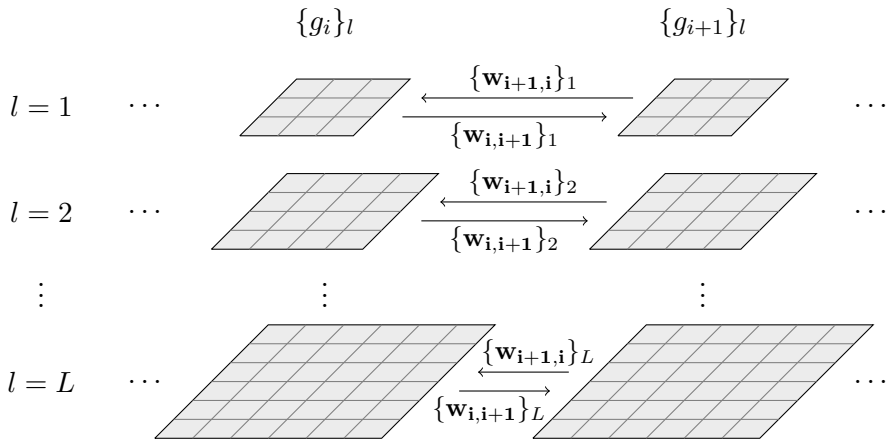


Figure 4.5: Gaussian pyramids $\{g_i\}_l$ and $\{g_{i+1}\}_l$. Arrows represent forward and backward flows for each pyramid level. Once the flows for a particular level l are carried out over the whole sequence, we proceed to the next level $l+1$ to refine them in a coarse-to-fine approach.

and rather than computing $\{k_i\}_l$, we use $\{g_i\}_l$ in the objective function. In above notation, M_l and N_l are respectively width and height in lth pyramid level, γ is a tunable constant, and $\uparrow$ denotes upscaling by a factor of 0.75 followed by Bicubic interpolation, applied on both channels of flow fields. A similar inequality could be formed to determine either to use $\{g_{i+1}\}_l$, or to consider generating $\{k_{i+1}\}_l$ by performing motion blur. Therefore, if the inequality didn't hold,

$$\begin{aligned}\{k_i\}_l(\mathbf{d}) &= B_{\Psi_{\uparrow\{\mathbf{w_{i+1,i}}\}_{l-1}}(\uparrow\{\mathbf{w_{i+1,i}}\}_{l-1})}\{g_i(\mathbf{d})\}_l + B_{\Psi_{\uparrow\{\mathbf{w_{i+1,i}}\}_{l-1}}(\uparrow\{\mathbf{w_{i+1,i+2}}\}_{l-1})}\{g_i(\mathbf{d})\}_l \\ \{k_{i+1}\}_l(\mathbf{d}) &= B_{\Psi_{\uparrow\{\mathbf{w_{i,i+1}}\}_{l-1}}(\uparrow\{\mathbf{w_{i,i-1}}\}_{l-1})}\{g_{i+1}(\mathbf{d})\}_l + B_{\Psi_{\uparrow\{\mathbf{w_{i,i+1}}\}_{l-1}}(\uparrow\{\mathbf{w_{i,i+1}}\}_{l-1})}\{g_{i+1}(\mathbf{d})\}_l\end{aligned} \tag{4.18}$$

which suggests that in order to generate k's for lth level, we use g's from the same level, but we upscale necessary warps from previous level in order to apply blurring. Having generated $\{k_i\}_l$ for all i's, we use them to update all forward and backward

flow fields for level l by minimizing objective functions below,

$$E_f(\mathbf{w_{i,f}}) = \sum_{\mathbf{d}} \psi(\|\{k_i\}_{l,x}.u_f + \{k_i\}_{l,y}.v_f + \{k_i\}_{l,t}\|^2) + \alpha \sum_{\mathbf{d}} \psi(\|\nabla u_f\|^2 + \|\nabla v_f\|^2)$$
$$E_b(\mathbf{w_{i,b}}) = \sum_{\mathbf{d}} \psi(\|\{k_{i+1}\}_{l,x}.u_b + \{k_{i+1}\}_{l,y}.v_b - \{k_i\}_{l,t}\|^2) + \alpha \sum_{\mathbf{d}} \psi(\|\nabla u_b\|^2 + \|\nabla v_b\|^2) \tag{4.19}$$

which should be minimized for all i. In above's notation, $\mathbf{w_{i,f}} = (u_f, v_f)$ is the *forward* flow that projects $\{k_i\}_l$ onto $\{k_{i+1}\}_l$, while $\mathbf{w_{i,b}} = (u_b, v_b)$ is the corresponding *backward* flow that does the opposite. These objective functions are optimized by considering their Euler-Lagrange equations and using conjugate gradients method. We set $\alpha = 0.12$, and use a Charbonnier penalizing function $\psi(s^2)$ with $\beta = 0.001$. Once $\mathbf{w_{i,f}}$ and $\mathbf{w_{i,b}}$ are found, we update flow fields for level l,

$$\{\mathbf{w_{i,i+1}}\}_l = \mathbf{w_{i,f}}, \qquad \{\mathbf{w_{i+1,i}}\}_l = \mathbf{w_{i,b}}, \qquad i \in \{1, 2, \ldots, T-1\} \tag{4.20}$$

Then we can keep alternating between these two steps as we move towards finer levels of pyramid: (1) estimating $\{k_i\}_l$ using $\{\mathbf{w_{i,i+1}}\}_{l-1}$ and $\{\mathbf{w_{i+1,i}}\}_{l-1}$, and (2) exploiting them to refine the flows and get $\{\mathbf{w_{i,i+1}}\}_l$ and $\{\mathbf{w_{i+1,i}}\}_l$ for the next finer level. Obviously, this alternation is stopped as we reach the finest level, $l = L$. Fig. 4.6 gives a block-diagram of the proposed method for refining flows. We use a modified version of Liu's C++ implementation [16] of CLG [9]. The modified version, referred to as `x2-CLG`, employs smoothness matrix $\mathbf{A}$ to account for occlusions, and would work only over a specific pyramid level.

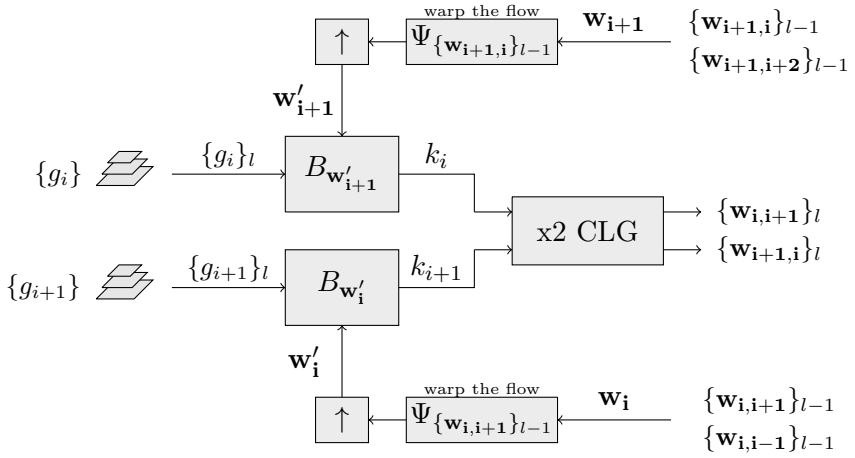


Figure 4.6: Block diagram for refining flow estimates of level $l-1$, and achieving new estimates for level l. x2 CLG denotes forward/backward CLG algorithm.

Variable	Description	Default Value
T	The number of fictitious frames between two actual frames, when generating artificial motion blur	20
τ	Half the number of fictitious frames that lie in the open shutter interval and contribute to the motion blur	8
α	Smoothness parameter	0.012
$ratio$	Gaussian pyramid ratio, i.e. the ratio of the width of consequent levels in the pyramid	0.75
$bPixels$	The number of boundary pixels thrown away when evaluating estimated flow fields based on AAE, AEP and MSE.	20
γ	The threshold used to decide if prior to flow estimation, applying the blur kernels of the other image is essential.	$\frac{T}{4\tau}$
$minWidth$	Minimum width of a level, when resizing an image and generating the corresponding Gaussian pyramid.	20
N_o	Number of outer iterations	16
N_{cg}	Number of conjugate gradient iterations	50
β	Charbonnier penalizer's parameter, applied on both data and smoothness terms.	0.001
σ_d	Smoothness parameter for applying a Gaussian filter on the divergence map	0.5
K	Linear coefficient used in generating the smoothness matrix, $\mathbf{A}$	10

Table 4.1: List of parameters and their default value in MB-CLG.

Chapter 5

Evaluation

This chapter introduces evaluation metrics that we will be using in the next chapter, when comparing the results of MB-CLG with other optical flow methods. It also includes a describtion of a common method for visualizing flows.

5.1 Definition of metrics

In order to evaluate the estimated flow fields we use three different metrics: (1) average angular error (AAE), (2) average endpoint error (AEP) and (3) mean squared error (MSE) between $\Psi_{\hat{\mathbf{w}}}(f_i)$ and f_{i+1} where $\hat{\mathbf{w}}$ is the estimated forward flow for frame i and f_i's are the unblurred frames. In this section we define these metrics, as well as divergence for flow fields. First, we need to point out that calculation of AAE and AEP requires the knowledge of the ground truth flows denoted by $\mathbf{w}$, and in order to find MSE we need to have the clean sequence f_i. We only consider g_i when testing each method, while f_i is employed for evaluation. Therefore, quantitative evaluation of blur-aware optical flow algorithms require synthetic datasets that come with the underlying ground truth flows and the unblurred sequence.

Given an estimated flow $\mathbf{w}(\mathbf{d}) = (u(\mathbf{d}), v(\mathbf{d}))$ and the ideal ground truth flow $\hat{\mathbf{w}}(\mathbf{d}) = (\hat{u}(\mathbf{d}), \hat{v}(\mathbf{d}))$ that matches the unblurred frames, angular error (AE) at pixel $\mathbf{d}$ is defined as the angle between vectors $(u(\mathbf{d}), v(\mathbf{d}))$ and $(\hat{u}(\mathbf{d}), \hat{v}(\mathbf{d}))$. Average

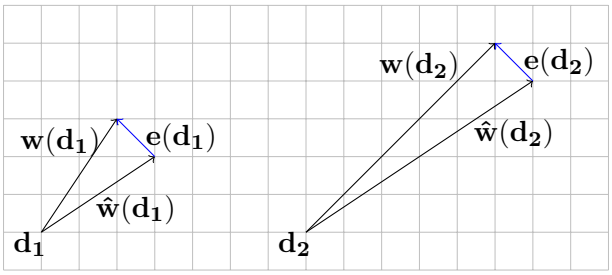


Figure 5.1: A simple example justifying independency of AAE and AEP. While $\mathbf{e}(\mathbf{d}) = \mathbf{w}(\mathbf{d}) - \hat{\mathbf{w}}(\mathbf{d})$ and AEP are equal for $\mathbf{d_1}$ and $\mathbf{d_2}$, AAE is smaller for $\mathbf{d_2}$.

angular error (AAE) is then defined as,

$$AAE = \frac{1}{M \times N} \sum_{\mathbf{d}} cos^{-1}\{\frac{\hat{\mathbf{w}}(\mathbf{d}).\mathbf{w}(\mathbf{d})^T}{\|\mathbf{w}(\mathbf{d})\|.\|\hat{\mathbf{w}}(\mathbf{d})\|}\} \tag{5.1}$$

Endpoint error (EP) at pixel $\mathbf{d}$ is defined as norm of the difference between the same vectors, and average endpoint error (AEP) is then represented as,

$$AEP = \frac{1}{M \times N} \sum_{\mathbf{d}} \|\mathbf{w}(\mathbf{d}) - \hat{\mathbf{w}}(\mathbf{d})\| \tag{5.2}$$

As Fig. 5.1 represents using a an example, AAE and AEP are in general independent metrics, i.e. if AAE is increased, AEP is necessarily not. AAE measures the directional error in the field estimates, and AEP considers error vector $\mathbf{e}(\mathbf{d}) = \mathbf{w}(\mathbf{d}) - \hat{\mathbf{w}}(\mathbf{d})$. In most applications, e.g. deblurring, structure-from-motion, denoising, etc., accurate ending points of flow vectors matter, and AEP gives a better evaluation of error.

In a same way, MSE for a forward flow that projects f_i onto f_{i+1} is,

$$MSE = \frac{1}{M \times N} \sum_{\mathbf{d}} \|\Psi_{\hat{\mathbf{w}}}(f_i)(\mathbf{d}) - f_{i+1}(\mathbf{d})\|^2 \tag{5.3}$$

Occlusions are usually detected by looking at $\nabla.\mathbf{w}$, flow field divergence [20]. The larger the divergence at a point, the larger occluded that region happens to be. However, applying traditional optical flow algorithms to sequences with motion blur would result in distorted flows, since the algorithm strives to match sharp regions from one frame to the corresponding region in another frame that might be blurred.

Therefore, one way to evaluate an optical flow method is to look how mistakenly it distorts the flow field, increases the divergence and introduces wrong occluded regions.

Depending on the motion speed, boundary pixels of a source image are likely to lie outside of the warped image grid. Therefore, there is no accurate way to handle motion blur near boundaries in the proposed method. In addition, in most applications, e.g., tracking and deblurring, pixels close to the margin are of less interest. Therefore, rather than evaluating estimated flows over the whole image plane, we only consider pixels that are not closer than `bpixels = 20` to the boundary.

In a similar manner to boundary pixels, boundary frames are also likely to introduce biased error. For instance, the knowledge of backward flow $\mathbf{w_{1,0}}$ is required for accurately accounting for blur kernels in frame g_1. However, we are unaware of frame g_0, and determining such flow field is not feasible. We need to deal with the same problem regarding the last frame g_T. In order to cope with these issues, we express the unknown warps as

$$\mathbf{w_{1,0}} = -\mathbf{w_{1,2}}, \qquad \mathbf{w_{T,T+1}} = -\mathbf{w_{T,T-1}} \tag{5.4}$$

based on the assumption that all of the points in the scene move along the same linear path in the previous frame. Needless to clarify that these modifications are applied to the results of all methods.

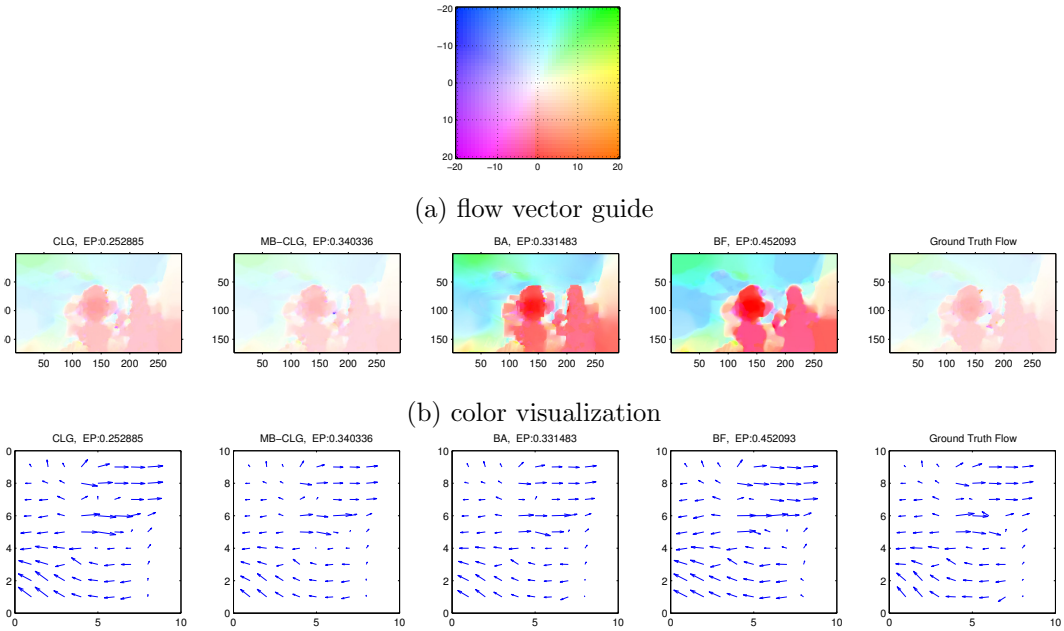
5.2 Visualization

In order to visualize estimated flows, we use quiver (`quiver()` in MATLAB), in addition to color representations. The latter is a color coding for each displacement vector, and the former is a set of arrows representing vectors on a grid of pixels. Color coding allows more accurate comparisons, as with a change of flow vector direction the hue changes. We use MATLAB implementation of [22] for `flowToColor()` function. Fig. 5.2 represents two frames of the mequon sequence. Fig. 5.3 shows quiver and color representations of estimated flows for the frames in

Fig. 5.2, as well as a color coding guide to interpret flows.

(a) frame # 6 (b) frame # 7

Figure 5.2: Two sample frames in the *mequon* sequence.



(a) flow vector guide

(b) color visualization

(c) quiver visualizations

Figure 5.3: (a) color coding guide, (b) color visualizations for the output flows of various methods, and (c) corresponding quiver representations.

Chapter 6

Results

A broad set of experiments are performed on various video sequences using various optical flow algorithms, e.g. CLG [9], Black-Anandan [6], BlurFlow [19], and the proposed method MB-CLG. In this chapter, the results are carried out in terms of metrics defined in Chapter 5, and are depicted and compared.Results for different levels of noise, different types of motion, different motion speed, and different shutter time τ are represented.

6.1 Homography results

In order to generate synthetic results that could be quantitatively evaluated, we take grayscale images of size 512×512 and generate a set of $T = 20$ homographies along a predefined and tunable trajectory. Then, we warp the sequence according to homographies, crop images of size 256×256 and save them as the unblurred sequence f_i. In addition, we have the ground truth flow fields for both forward and backward cases, that could be used for evaluation purposes. It also is possible to apply motion blur on the sequence according to the underlying flow fields, and save the resulting motion blurred sequence g_i. In order to generate motion blur according to Eq. 3.1, we change the integral into a summation, use $T_s = 20$ and exposure time $\tau = 0.4$. We perform all simulations using a 2.4 GHz Intel Core i5 processor and under Mac OS 10.7.5 environment. The proposed method (MB-CLG) is implemented partly in C++ and MATLAB. In each experiment we compare our results with those of

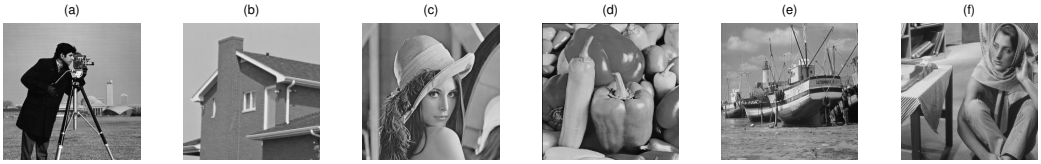


Figure 6.1: Sample 512×512 grayscale images used to generate a motion blurred sequence of 20 frames each of size 256×256. (a) cameraman, (b) house, (c) lena, (d) pepper, (e) boat and (f) barbara.

CLG [9], Black and Anandan (BA) [6] and BlurFlow [19]. We use C. Liu's C++ implementation [16] for CLG, Black's MATLAB implementation for BA and Portz et al's C++ implementation for BlurFlow method which are available online on their project website.

6.2 Preliminary datasets

Fig. 6.2 shows mean squared error (MSE) for sample datasets portrayed on Fig. 6.1. The sequences are generated based on the same trajectory, and MSE is computed for various methods and for each individual frame. Estimated flow fields are used to warp each frame backward and forward onto the coordinates of neighbor frames, then mean squared error is calculated. In a similar way, Fig. 6.3 and Fig. 6.4 demonstrate AAE and AEP measures respectively. The dashed lines represent the average over the sequence, and their numerical values are listed in Table 6.1. As if it could be anticipated, traditional methods like CLG [9] and BA [6] that are deprived of a mechanism to handle motion blur would obviously fail in most cases to produce accurate flows. BlurFlow [19], as a blur-aware method, produces flows with remarkably less MSE, AAE and AEP. The proposed method (MB-CLG), however, computes flows with even less error in the sense of all metrics. Fig. 6.13 depicts the divergence of the estimated flow fields using each method for three samples. Comparing the divergence results reveals that traditional methods produce so-called *wrinkles* and distort the flow as they try to align sharp and blurred regions. However, blur-aware algorithms and particularly MB-CLG perform better estimations. A key observation here is that MB-CLG cannot improve the flows

in regions near the boundary, since it uses the information from nearby frames and pixels near boundary are likely to lie outside of image coordinates when warped back and forth. Table 6.2 represents average time in seconds required to compute forward and backward flows for a single frame. Since the implementation for BA [6] is done in MATLAB it takes much more time compared to CLG - which is implemented in C++ [16].

6.3 Noise Results

In this section, we add white Gaussian noise of different variances, and compare the results of various methods. We use the same method as the previous section to generate a sequence of synthetic frames by warping a single image. Prior to saving the sequence, we add the noise,

$$g_i = \Psi_{\mathbf{w_{0,i}}}(f) * B_{\mathbf{w_i}} + n_i, \qquad \forall i \in \{1, 2, ..., T\} \tag{6.1}$$

where g_i denotes the generated frame, f represents the still image, n_i is a vector of IID Gaussian noise with standard deviation σ, and $B_{\mathbf{w_i}}$ is a spatially-varying motion blur kernel based on flow fields that map ith frame to next and previous frames. In addition, $\Psi_{\mathbf{w_{0,i}}}(f)$ is a warped version of f according to $\mathbf{w_{0,i}}$, and $\mathbf{w_{0,i}}$ denotes an arbitrary flow field that generates ith frame based on the still image f. We set $\sigma \in \{0, 2.5, 5, 7.5, 10, 12.5\}$, repeat each experiment 5 times, and look at MSE, AEP and AAE measures.

Fig. 6.12 represents the robustness against noise for each method in an MSE sense. Traditional methods, i.e., CLG [9] and BA [6], that are deprived of a technique to handle motion blur, seem to be less robust against noise. While MSE increases linearly for MB-CLG and BlurFlow [19], traditional methods seem to be more sensitive to noise. This might come from the built-in blurring in MB-CLG and BlurFlow, as they try to cancel the effect of noise by further blurring frames, and this will result in denoising of the sequence. Fig. 6.5 also represents AEP and AAE results for estimated flow fields, evaluated with respect to ground truth flows.

6.4 Trajectory results

In order to generate a synthetic sequence from a still image, we first need to determine a set of homographies. In this section, we try various homographies by tuning the trajectory parameters. Then, we evaluate the estimated flows by each technique to see which one outperforms others in slow/fast movements, pure translational/rotational movements, etc. Homography $\mathbf{H}_i$ is determined as,

$$\mathbf{H}_i = s_i \begin{bmatrix} cos\theta_i & -sin\theta_i & A_i cos\alpha_i \\ sin\theta_i & cos\theta_i & A_i sin\alpha_i \\ 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

where s_i is a scale that accounts for zooming, A_i denotes the magnitude of translation vector, α_i is the angle of translation vector, and θ_i serves as the rotation angle. For each i, we construct $\mathbf{H}_i$ according to,

$$\begin{aligned} A_i &= A_0 sin\frac{2\pi i}{T_m} \\ \theta_i &= \theta_0 sin\frac{2\pi i}{T_m} \\ s_i &= 1 + s_0 sin\frac{2\pi i}{T_m} \\ \alpha_i &= \alpha_{i-1} + \alpha_0 |sin\frac{2\pi i}{T_m}| \end{aligned} \tag{6.3}$$

where T_m denotes the period of the sinusoidal motion and is set to 10 frames. The default values are: $A_0 = 50$, $\theta_0 = 2\pi/72$, $\alpha_0 = 2\pi/72$, and $s_0 = 0.05$. In this section, we try changing these variables to evaluate different methods using new sequences. Fig. 6.6 represents both MSE and AEP results, as we change A_0. According to Eq. 6.2, video sequences generated with smaller A_0 will have smaller displacement field and less motion blur.

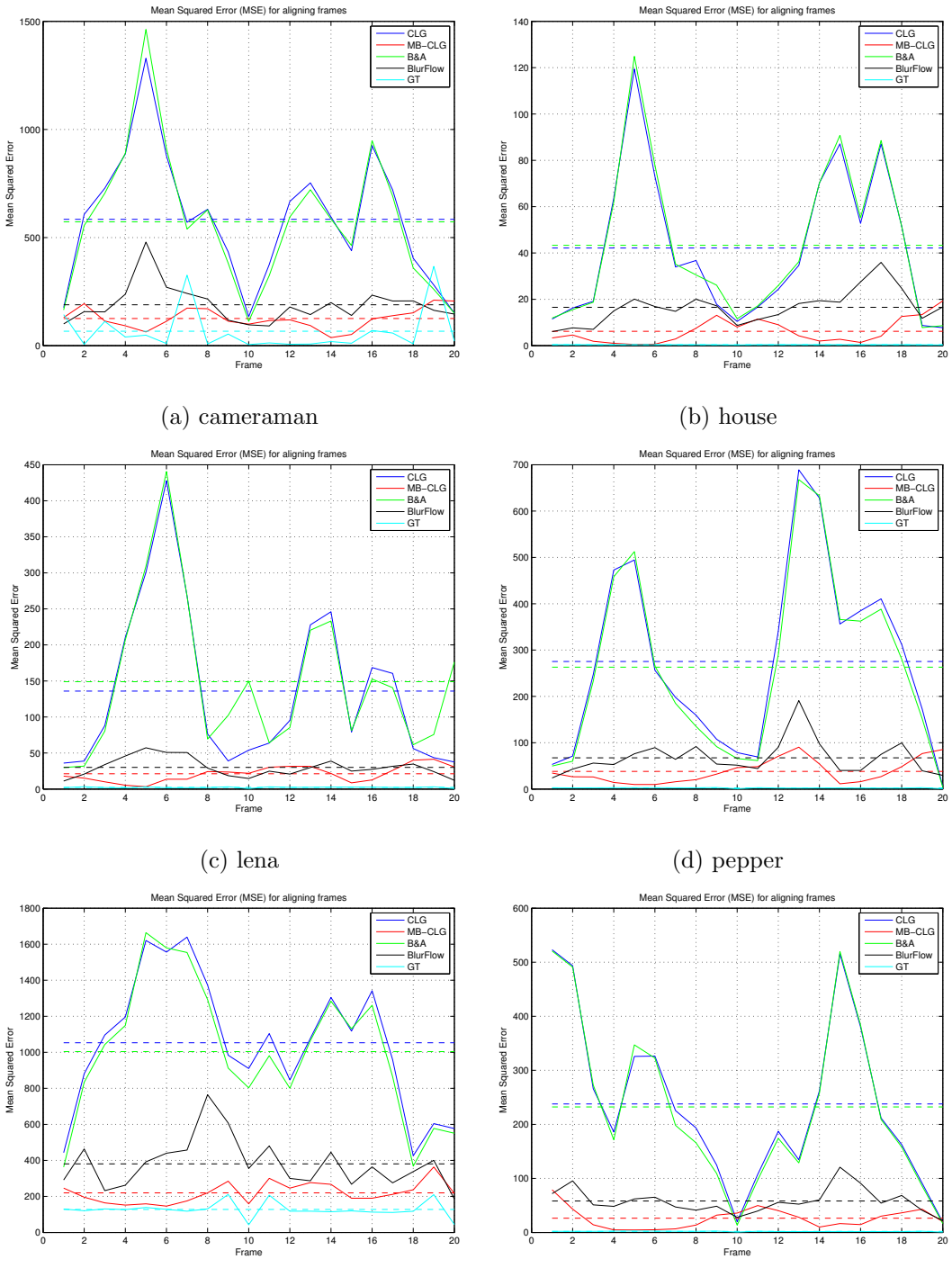
Next, we repeat the same experiment on a synthetically generated sequence with pure rotational motion. Fig. 6.7 depicts MSE and AEP results for a pure rotational motion. A_0 is set to zero, and θ_0 is varied in chosen to be $\{2.5, 3, 3.5, 4, 4.5, 5\}\times$ $\pi/72$. As Fig. 6.7(a) demonstrates, MB-CLG (red curve) and BlurFlow (black curve)

end up outperforming traditional methods as we increase θ_0. As portrayed in Fig. 6.8 which represents estimated flows for $\theta_0 = 5\pi/72$, displacement field is necessarily spatially-varying when the underlying scene dynamics involve large rotations. Thus, we conclude blur-aware methods' superiority is more evident in the case of large and spatially-varying motions. Although BlurFlow [19] outperforms traditional methods, MB-CLG gives better results.

Finally, we varied the scale parameter $s_0 \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$, and gave as input each generated sequence to each optical flow technique. The results are represented in Fig. 6.9. As it is evident, increasing s_0 causes a large gap between the results of blur-aware and traditional techniques. In other words, MB-CLG and BlurFlow are more robust against motion blur deterioration. Similar to rotational and translational motions, MB-CLG achieves the best results. Fig. 6.10 visualizes the results of each method for $s_0 = 0.25$, in addition to ground truth flows. The artifacts in estimated flows are perfectly clear by a subjective evaluation.

6.5 Shutter time results

In a different experiment, we varied the shutter time $\tau \in \{1, 2, \ldots, 10\}$. Increasing τ results in more sever motion blur, as the shutter is kept open for longer. Fig. 6.11 represents the results for (a) MSE, and (b) AEP. Although Black-Anandan method turns out to achieve better results by increasing τ, other methods produce larger errors. Denoising contribution of motion blur explains why Black-Anandan, previously proved to be highly sensitive on noise, actually performs better when the sequence is blurred. In terms of both MSE and AEP, MB-CLG still achieves best results for different shutter times.



(a) cameraman

(b) house

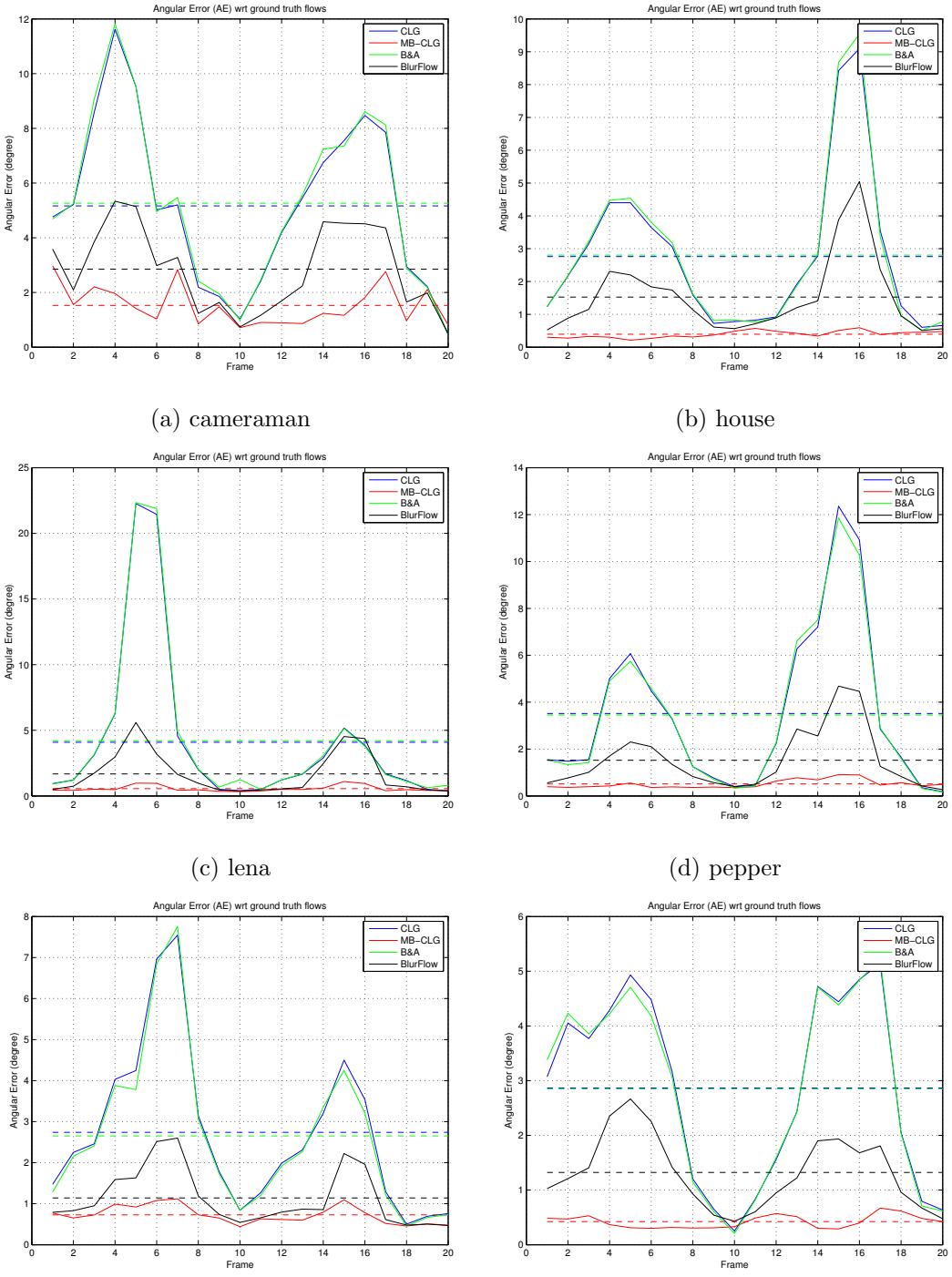
(c) lena

(d) pepper

(e) boat

(f) barbara

Figure 6.2: Mean squared error results for matching frames using both forward and backward estimated flow fields. In each dataset we generate 20 synthetically motion blurred frames, and try to estimate the flows using CLG [9], BA [6], BlurFlow [19] and the proposed method MB-CLG. MSE values are also calculated for ground truth flows (GT).



(a) cameraman

(b) house

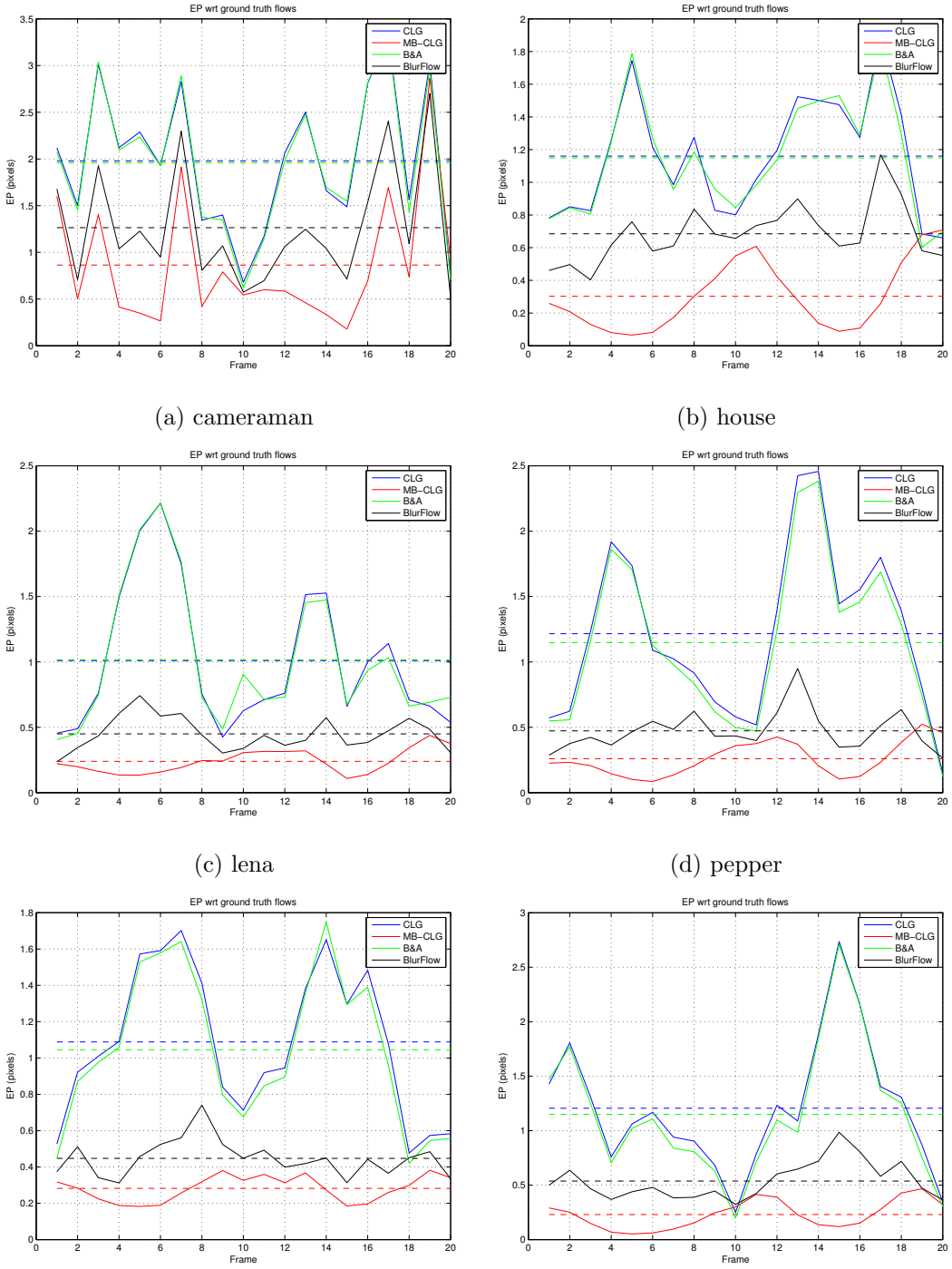
(c) lena

(d) pepper

(e) boat

(f) barbara

Figure 6.3: Average angular error (AAE) for a set of sample images, computed with respect to the underlying ground truth flows. AAE is calculated for each frame separately, and for each flow field computation technique, e.g. CLG [9], BA [6], BlurFlow [19] and the proposed method MB-CLG.



(a) cameraman

(b) house

(c) lena

(d) pepper

(e) boat

(f) barbara

Figure 6.4: Average endpoint error (AEP) for a set of sample images, computed with respect to the underlying ground truth flows. AEP is calculated for each frame separately, and for each flow field computation technique, e.g. CLG [9], BA [6], BlurFlow [19] and the proposed method MB-CLG.

dataset	**AAE**				**AEP**				**MSE**			
method	CLG	MB-CLG	BA	BF	CLG	MB-CLG	BA	BF	CLG	MB-CLG	BA	BF
camera	5.16	1.52	5.26	2.85	1.97	0.86	1.95	1.26	584	125	572	189
house	2.76	0.39	2.80	1.52	1.16	0.30	1.14	0.68	42	6	43	16
lena	4.08	0.55	4.21	1.67	1.00	0.23	1.01	0.45	135	21	148	30
pepper	3.51	0.51	3.45	1.52	1.21	0.25	1.14	0.47	275	38	263	67
boat	2.74	0.73	2.65	1.14	1.01	0.28	1.04	0.44	1052	219	1038	380
barbara	2.86	0.42	2.84	1.32	1.20	0.23	1.15	0.54	237	26	232	58

Table 6.1: Average AAE, AEP and MSE over the whole sequence, computed for different sample datasets and various algorithms, CLG [9], BA [6], BlurFlow (BF) [19] and the proposed method MB-CLG.

method	CLG	MB-CLG	BA	BF
time (s)	2.72	17.92	29.69	21.71

Table 6.2: Average computation time required to calculate forward and backward flow fields per frame.

6.6 Multiple Motion Results

In this part, two sample unblurred datasets available on `http://vision.middlebury.edu/flow/` and depicted on Fig. 6.14 are used, as they incorporate multiple motions. We run a baseline optical flow method, CLG [9] to get ground truth flows. Then, we add artificial motion blur according to 3.1 and estimated ground truth flows. Fig. 6.15 shows the corresponding visualized flows for frame 2 for (a) schefflera and (b) urban dataset. We use the conventional method described in [2] for visualization. Flow fields estimated by BlurFlow are smoother than ground truth flows, and BA results look sharper and contain square artifacts. For this set of samples, Fig. 6.16 shows MSE values for aligning frames using forward and backward flows. It would have been better not to *estimate* ground truth flows, since the baseline method is not optimal due to complicated motion in these datasets. However, MB-CLG still outperforms other traditional or blur-aware techniques in a mean squared error sense.

6.7 Moving Objects

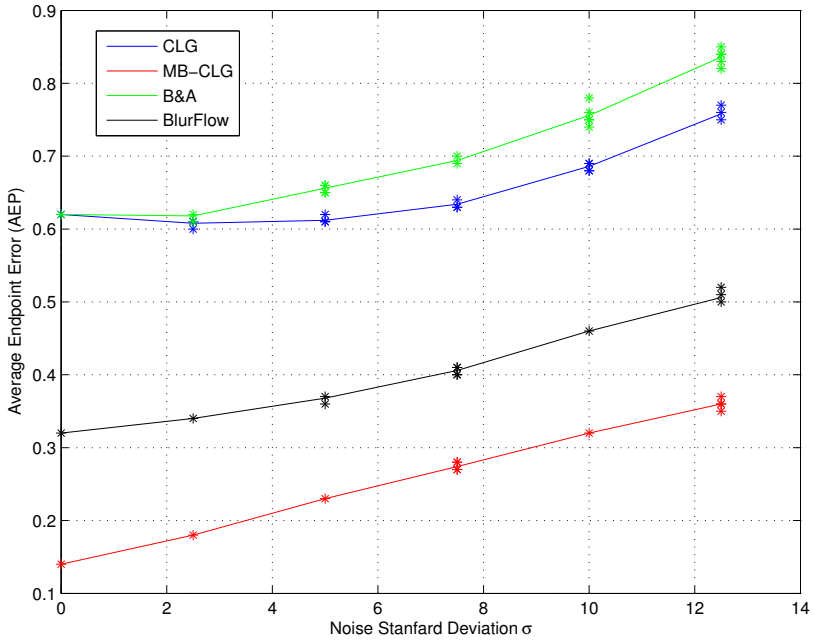
This section includes the results for synthetic sequences containing a moving object, i.e. astronaut, and bird. We choose a background, and move the moving object on a predefined path. Motion blur artifacts are then artificially added based

on the ground truth flow fields.

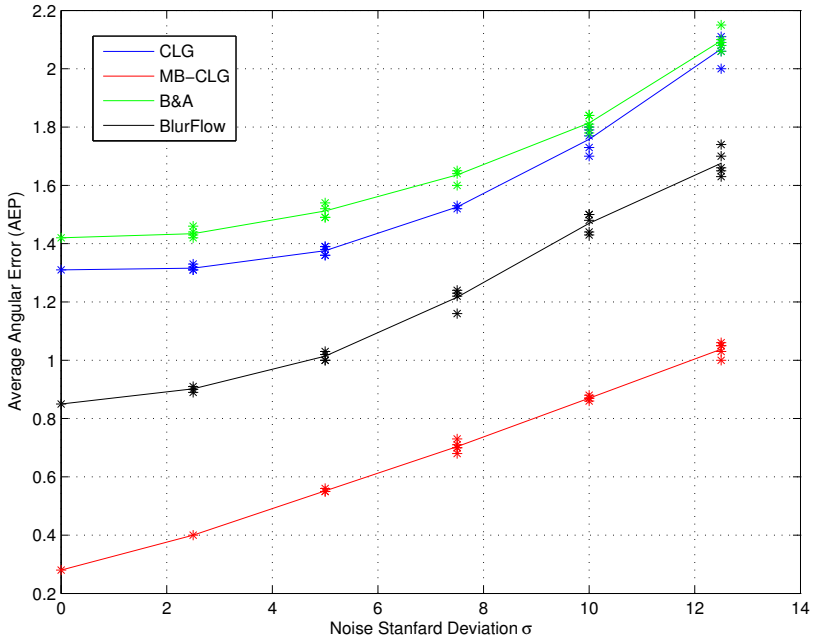
Fig. 4.1 depicts some of the frames in the astronaut sequence. Fig. 6.17 represents (a) AAE, and (b) AEP results for different methods. In addition, Fig. 6.18 demonstrates estimated flows for frames $i \in \{1, 2, \ldots, 7\}$. In both objective and subjective evaluations, MB-CLG achieves the best results. Specifically, the results for frame $i = 4$ (represented in the 4th row in Fig. 6.18) prove the superiority of MB-CLG. The astronaut has the highest motion speed in frame $i = 4$, hence motion blur artifacts are more intense. Fig. 6.19 demonstrates some of the frames in the bird sequence. In this sequence, the moving object is relatively smaller in size. As a consequence, estimating the flow is more difficult. In a similar way as the astronaut sequence, Fig. 6.20 represents quantitative results, and Fig. 6.21 visualizes estimated flows for the bird sequence. While MB-CLG achieves the best flows, the results of all methods for $frame\#4$ are far from the ground truth. One major reason is the high speed associated with a small moving object in the scene. Table 6.3 summarizes the average values for AAE and AEP metrics for both datasets.

dataset	**AAE**				**AEP**			
method	CLG	MB-CLG	BA	BF	CLG	MB-CLG	BA	BF
astronaut	8.35	7.55	8.65	24.3	1.53	1.01	1.38	1.71
bird	6.02	5.35	9.58	24.54	1.00	0.61	0.94	1.38

Table 6.3: Average AAE and AEP over the whole sequence, computed for the astronaut and the bird datasets and various algorithms, CLG [9], BA [6], BlurFlow (BF) [19] and the proposed method MB-CLG.

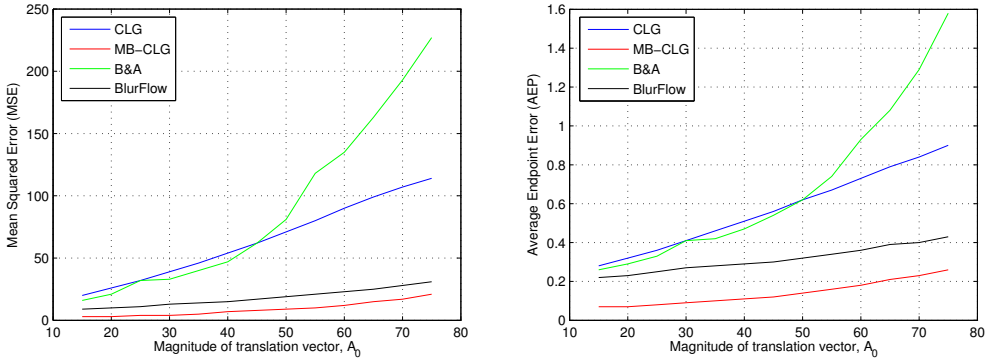


(a) AEP



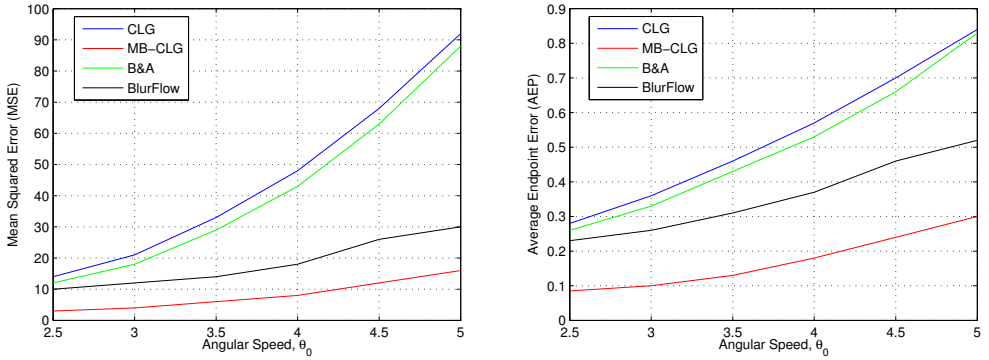
(b) AAE

Figure 6.5: Results for (a) AEP and (b) AAE measures, when synthetic input video sequence is corrupted with additive Gaussian noise of standard deviation σ.



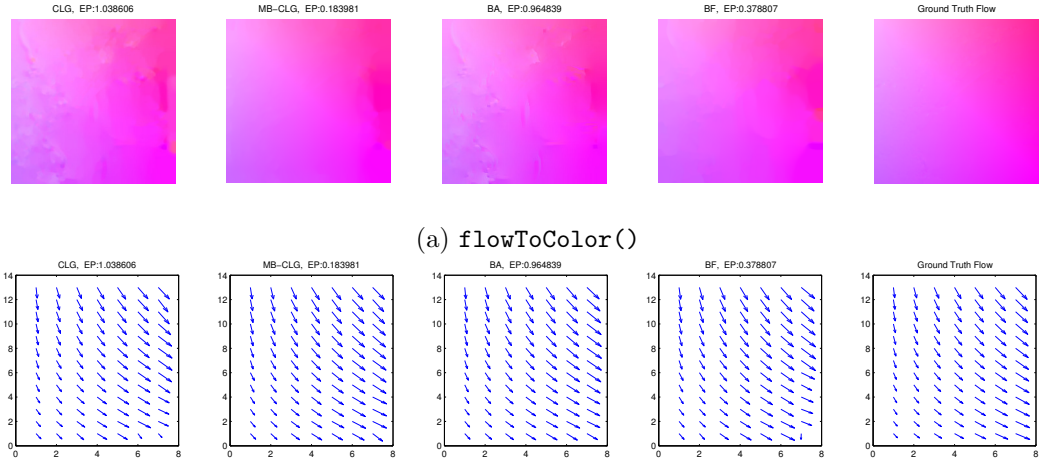
(a) MSE vs. A_0 (b) AEP vs. A_0

Figure 6.6: Results for (a) MSE and (b) AEP measures, as the motion speed (controlled by A_0) is varying. MB-CLG outperforms other methods regardless of speed.



(a) MSE vs. θ_0 (b) AEP vs. θ_0

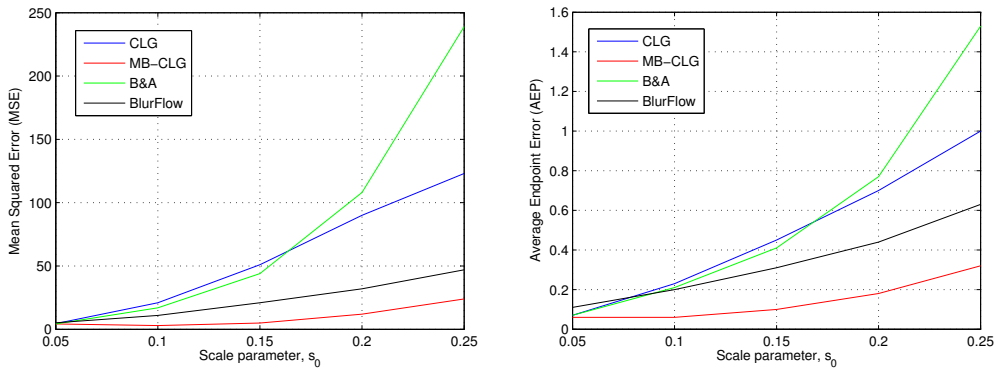
Figure 6.7: Pure Rotational Motion. Results for (a) MSE and (b) AEP measures, as the angular speed (controlled by θ_0) is varying. MB-CLG outperforms other methods regardless of speed.



(a) `flowToColor()`

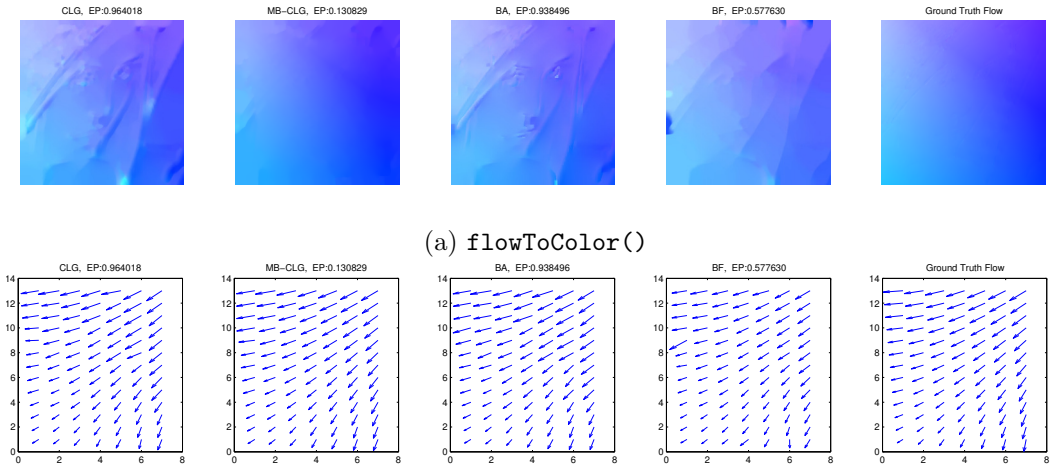
(b) `quiver()`

Figure 6.8: Estimated flow fields for a pure rotational motion, visualized by (a) `flowToColor()` function, and (b) `quiver()` in MATLAB



(a) MSE vs. s_0 (b) AEP vs. s_0

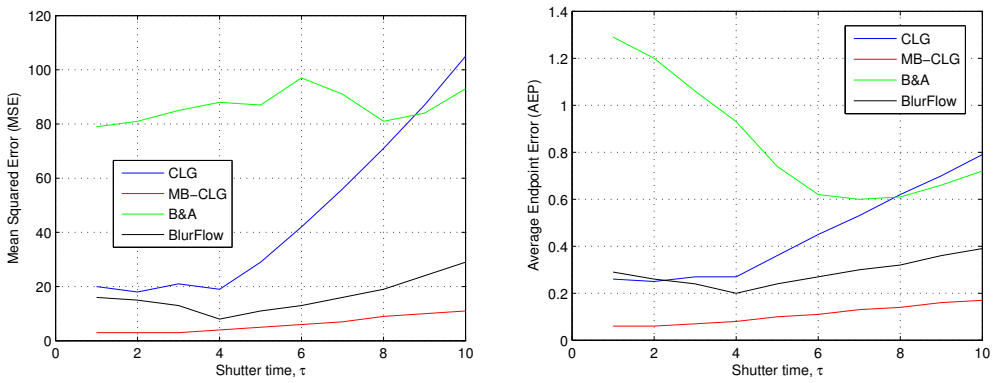
Figure 6.9: Pure Scale Motion. Results for (a) MSE and (b) AEP measures, as the scale parameter (denoted as s_0) is varying. Blur-aware methods, specifically MB-CLG, are evidently more resilient against motion blur artifacts.



(a) `flowToColor()`

(b) `quiver()`

Figure 6.10: Estimated flow fields for a pure scale motion, visualized by (a) `flowToColor()` function, and (b) `quiver()` in MATLAB



(a) MSE vs. τ (b) AEP vs. τ

Figure 6.11: Results for (a) MSE and (b) AEP measures, as the shutter time τ is varying.

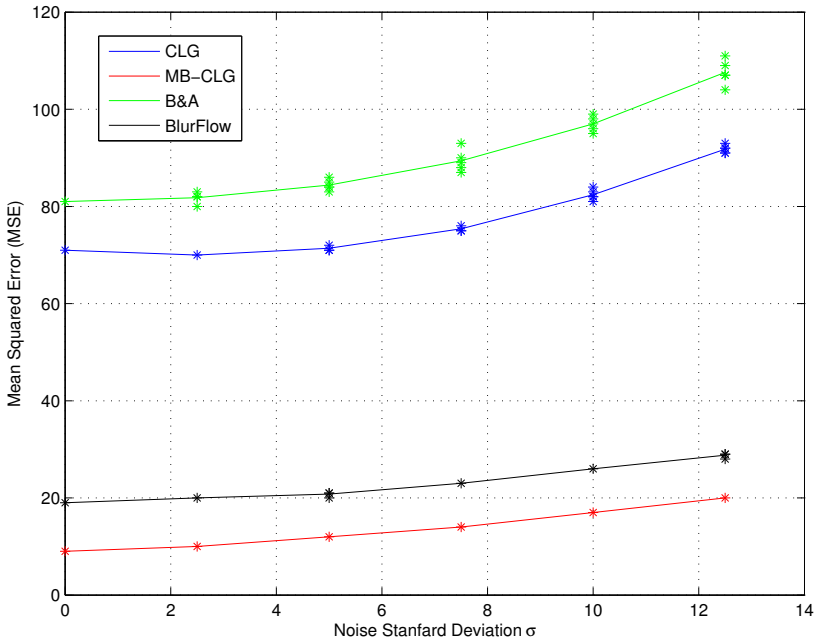


Figure 6.12: Mean Squared Error (MSE) between matched frames, when input video sequence is artificially corrupted with additive Gaussian noise of standard deviation σ. The results are shown for various methods.

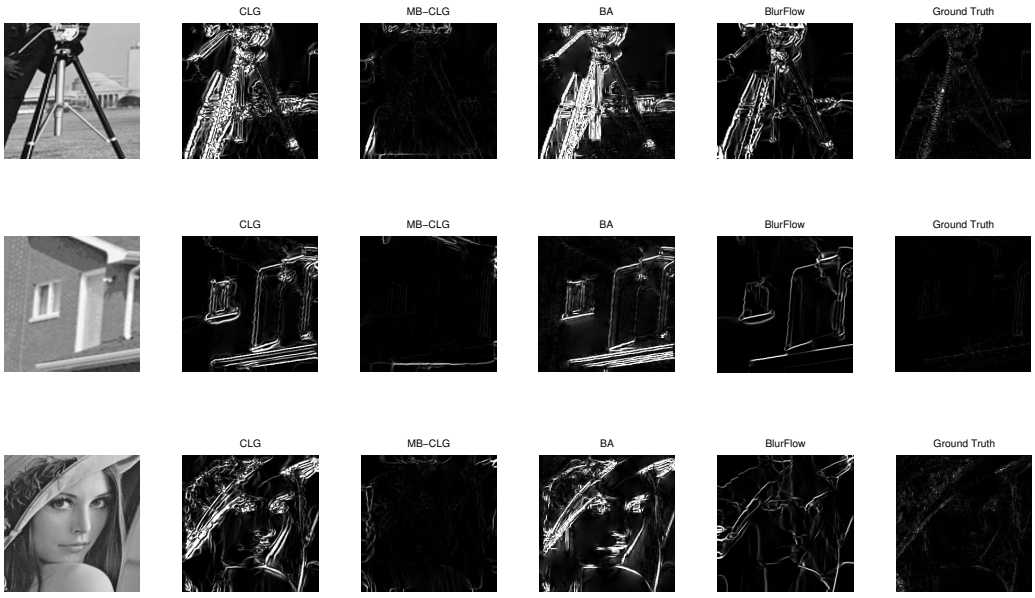
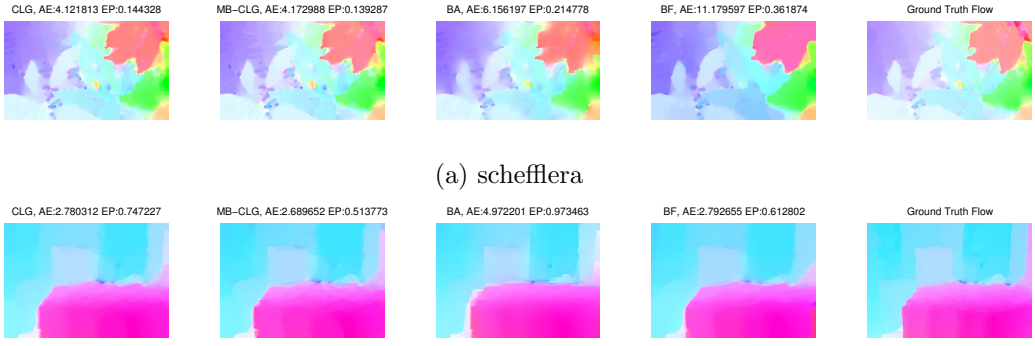


Figure 6.13: Divergence of flow fields for three video sequences, estimated using CLG [9], BA [6], BlurFlow (BF) [19] and the proposed method MB-CLG.

(a) schefflera (b) urban

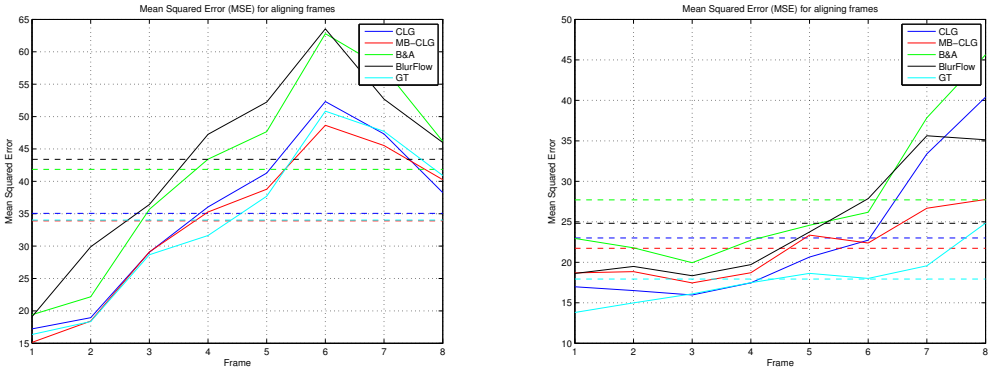
Figure 6.14: Sample datasets that contain multiple motions.



(a) schefflera

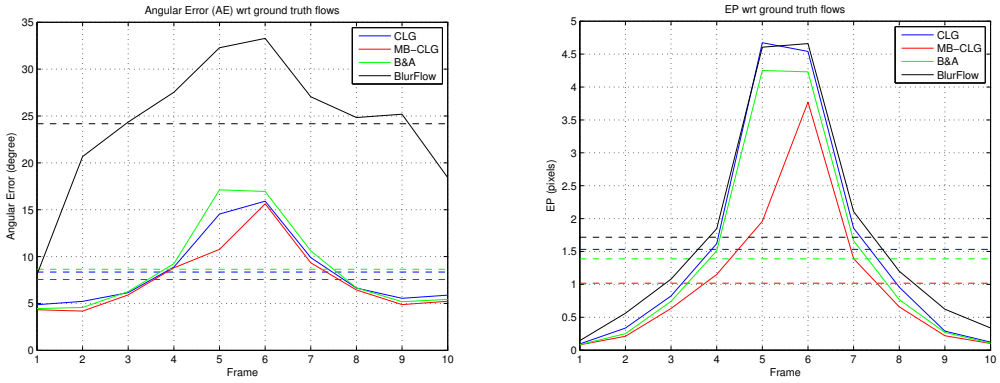
(b) urban

Figure 6.15: Visualized flow fields for datasets containing mutliple motions.



(a) schefflera (b) urban

Figure 6.16: Mean squared error (MSE) results for datasets with multiple motions.



(a) AAE (b) AEP

Figure 6.17: (a) AAE, and (b) AEP results for estimated flows for the astronaut sequence. Average values are indicated by horizontal lines.

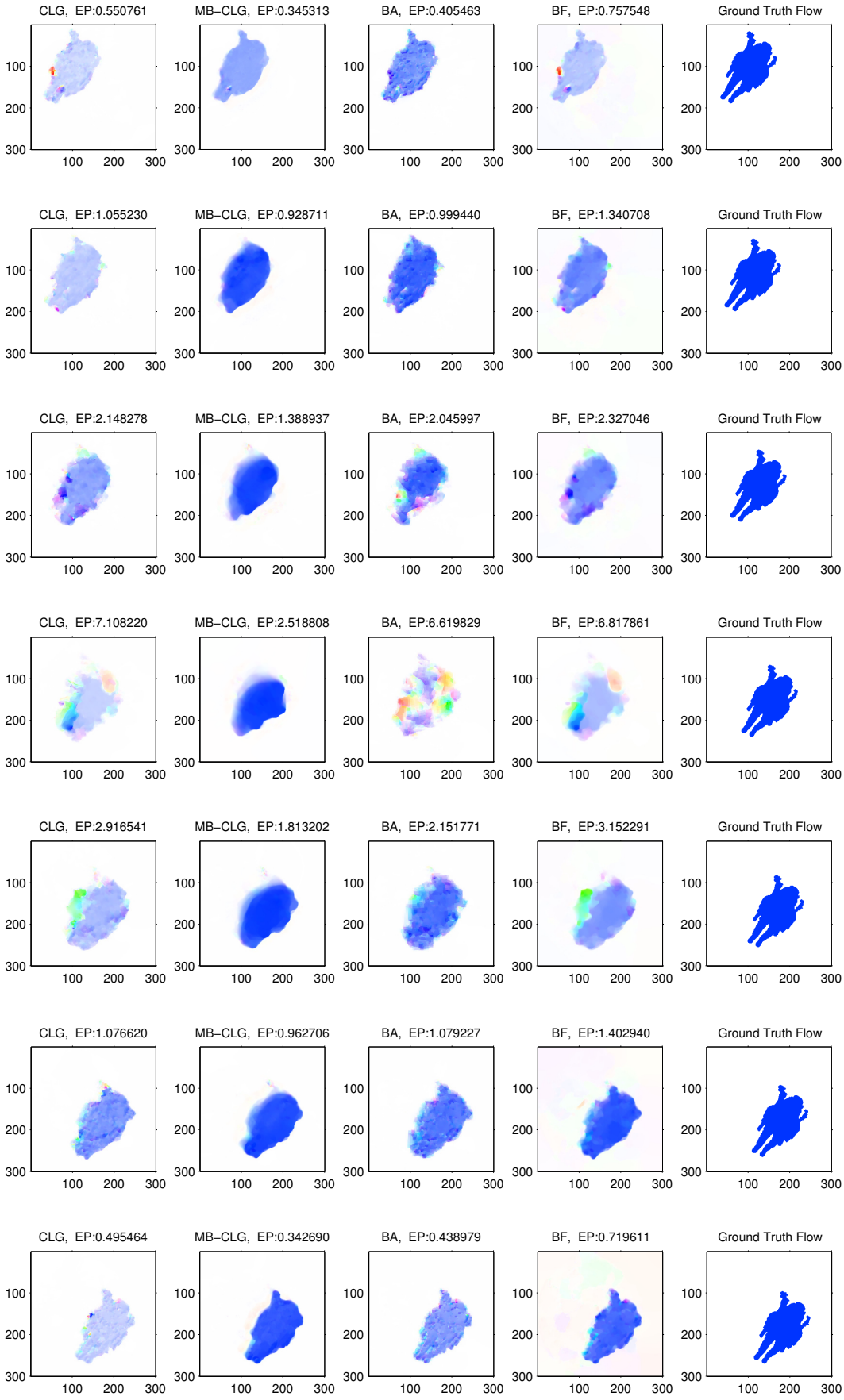
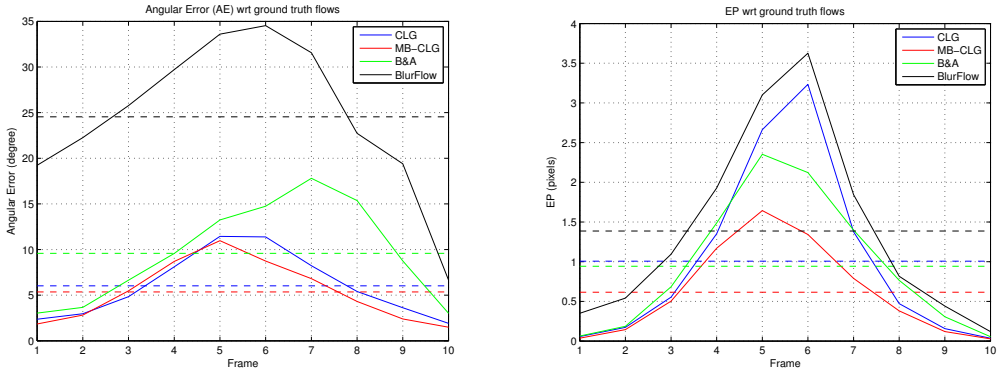


Figure 6.18: Estimated flows of various methods (columns) for seven frames of the astronaut sequence (rows).

(a) frame #1 (b) frame #2 (c) frame #3 (d) frame #4

(e) frame #5 (f) frame #6 (g) frame #7 (h) frame #8

Figure 6.19: Sample frames of an artificially generated motion blurred sequence of a moving bird in the sky.



(a) AAE (b) AEP

Figure 6.20: (a) AAE, and (b) AEP results for estimated flows for the bird sequence. Average values are indicated by horizontal lines.

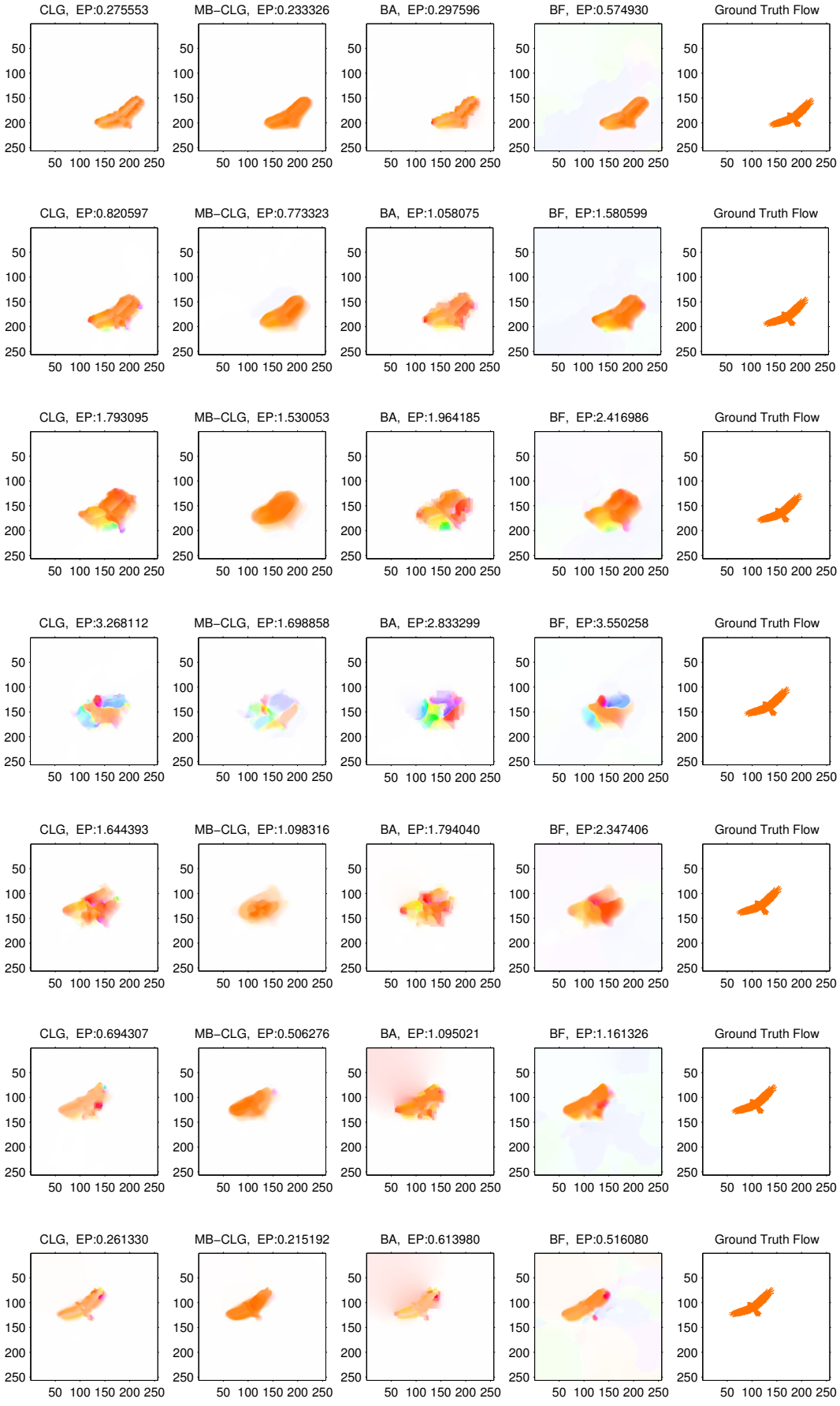


Figure 6.21: Estimated flows of various methods (columns) for seven frames of the bird sequence (rows).

Chapter 7

Conclusion

In this thesis we addressed the problem of optical flow in the presence of motion blur, and have made two major contributions. The first is applying blur functions of the source image to the target image, and vice versa. This step turns out to be balancing the effect of different blur levels in different video frames. For instance, If the source image happens to be sharper than the target image, they will have identical levels of blur after this step. We also argue that displacement fields that match modified source and target images are approximately those that would match them if there was no motion blur deterioration. The second is the adaptation of *warp-the-flow* technique, rather than directly warping images. In particular, flows are warped by flows and projected onto the coordinates of the target image. Based upon warped flows, blur functions are carried out and applied on the target image. The same procedure is repeated for the source image.

Among traditional methods that do not account for motion blur in the sequence, we review and discuss their superiorities and shortcomings in Section 2.2. First we give an overview of Lucas-Kanade method, as a well-known local technique. It became evident that while local techniques are more robust against noise, they cannot produce a dense and global displacement field. As a global technique, we study the objective functional proposed by Horn-Schunck [14], different extensions for it, and their corresponding Euler-Lagrange equations. Combined local-global (CLG) method [9] is subsequently reviewed. In brief, CLG serves as a hybrid tech-

nique that employs a global objective functional, while benefits from robustness against noise which is an inherent property of local methods. We also review the advantages of various Horn-Schunck proposed extensions, e.g., adding nonquadratic penalty functions to both the data and the smoothness terms, employing a coarse-to-fine approach, etc. Black and Anandan method [6] is also briefly studied, as they relax the assumption of single motion and adapt a new objective function. However, none of them consider blur deteriorations caused by motion.

Chapter 3 addresses the motion blur model employed, the effect of motion blur in estimating flow fields, followed by an overview of a novel blur-aware optical flow technique, BlurFlow, proposed by Portz et al [19]. It turned out that despite BlurFlow results in more accurate flows compared to traditional methods, it requires a large amount of computations and data storage. Portz et al, first apply a baseline method, then compute a large four dimensional grid for each video frame. Once this big grid of data is precomputed, they minimize their own criterion over the whole sequence by interpolating the grid at certain positions in each iteration. Generating and interpolating such large grid, as experimented as well, is extremely time and storage consuming.

A detailed discussion for the proposed method, MB-CLG, was given in Chapter 4. We utilize the result of Theorem 4.8 to conclude one can extract the latent displacement field by pre-filtering both the target and the source images. The proposed method also accounts for independently moving objects and occlusions. We relaxed the assumption that the smoothing parameter α should be constant over the whole image. Instead, we defined the smoothing matrix $\mathbf{A}$ according to the gradient map of the estimated flow field. Based on the results of the astronaut sequence, we observed this slight modification made the proposed method more robust against moving objects and occlusion.

In Chapter 6, various experiments on different sample sequences were studied. Furthermore, we have directly compared the results of MB-CLG with those of three other methods for solving the optical flow: Combined local-global (CLG) [9], Black-Anandan (BA) [6], and BlurFlow (BF) [19]. In Chapter 5, we defined Average

Angular Error (AAE) and Average Endpoint Error (AEP) as conventional metrics to evaluate estimated flows with respect to the ground truth flow. In addition, we employ Mean Squared Error (MSE) to measure how accurate an estimate of the flow matches the frames in the sequence. Then we used these metrics for each scenario. First we generated various sequences out of a single image according to a set of homographies. Then, we changed homography variables and considered pure translational, pure rotational and pure scale motions. In each case, motion-blur would get more severe as the controlling parameters, i.e. θ_0, A_0 and s_0, were increased. This led to a large gap between the results of blur-aware and traditional methods in terms of MSE and AEP, and MB-CLG achieved the best flow fields. In a separate experiment, we varied the shutter time τ, hence generated sequences had different levels of motion blur. We repeated these experiments for different still images, and the proposed method, MB-CLG, achieved superior results in terms of MSE, AEP, and AAE for all of the experiments.

In addition to homography results, we also tested each method on datasets containing multiple motions, and sequences including a moving object, e.g. astronaut, bird. We argue that the ground truth flows for these scenarios can also be generated. Therefore, we can evaluate flows in terms of AEP and AAE.

Appendix A

Proof for Theorem 4.8

We start by expanding the expression for $k_i(\mathbf{d})$ by substituting for $g_i(\mathbf{d})$ from Eq. 3.3. Therefore, we get,

$$\begin{aligned} k_i(\mathbf{d}) &= B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{g_i(\mathbf{d})\} + B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{g_i(\mathbf{d})\} \\ &= B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\} \\ &+ B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\} \\ &+ B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\} \\ &+ B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\} \end{aligned} \tag{A.1}$$

In the same way we can expand $k_{i+1}(\mathbf{d})$,

$$\begin{aligned} k_{i+1}(\mathbf{d}) &= B_{\Psi_{\mathbf{w_{i,i-1}}}(\mathbf{w_{i,i-1}})}\{g_{i+1}(\mathbf{d})\} + B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{g_{i+1}(\mathbf{d})\} \\ &= B_{\Psi_{\mathbf{w_{i,i-1}}}(\mathbf{w_{i,i-1}})}\{B_{\mathbf{w_{i+1,i}}}(f_{i+1}(\mathbf{d}))\} \\ &+ B_{\Psi_{\mathbf{w_{i,i-1}}}(\mathbf{w_{i,i-1}})}\{B_{\mathbf{w_{i+1,i+2}}}(f_{i+1}(\mathbf{d}))\} \\ &+ B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{B_{\mathbf{w_{i+1,i}}}(f_{i+1}(\mathbf{d}))\} \\ &+ B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{B_{\mathbf{w_{i+1,i+2}}}(f_{i+1}(\mathbf{d}))\} \end{aligned} \tag{A.2}$$

We continue by writing and expanding the expression for $\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d}))$ which is the warped version of $k_i(\mathbf{d})$ based on $\mathbf{w_{i,i+1}}$,

$$\begin{aligned}\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) = \Psi_{\mathbf{w_{i,i+1}}}\{B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\}\} \\ + \Psi_{\mathbf{w_{i,i+1}}}\{B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\} \\ + \Psi_{\mathbf{w_{i,i+1}}}\{B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\}\} \\ + \Psi_{\mathbf{w_{i,i+1}}}\{B_{\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\}\end{aligned} \tag{A.3}$$

According to Eq. 4.4 we get,

$$\begin{aligned}\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) = B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{\Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\}\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{\Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{\Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i-1}}}(f_i(\mathbf{d}))\}\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{\Psi_{\mathbf{w_{i,i+1}}}\{B_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\}\end{aligned} \tag{A.4}$$

Also, we can expand further to get,

$$\begin{aligned}\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) = B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{\Psi_{\mathbf{w_{i,i+1}}}(f_i(\mathbf{d}))\}\} \\ = B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{f_{i+1}(\mathbf{d})\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{f_{i+1}(\mathbf{d})\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{f_{i+1}(\mathbf{d})\} \\ + B_{\Psi_{\mathbf{w_{i,i+1}}}\Psi_{\mathbf{w_{i+1,i}}}(\mathbf{w_{i+1,i+2}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{f_{i+1}(\mathbf{d})\}\end{aligned} \tag{A.5}$$

However, one should notice that $\Psi_{\mathbf{w_{i,i+1}}}$ and $\Psi_{\mathbf{w_{i+1,i}}}$ as consequent operators have the interpretation of projecting an input image from the coordinates of f_i onto f_{i+1}, and then projecting the result from the coordinates of f_{i+1} onto f_i. Therefore, they

would cancel each other out,

$$\begin{aligned}\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) = B_{(\mathbf{w_{i+1,i}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{f_{i+1}(\mathbf{d})\} \\ + B_{(\mathbf{w_{i+1,i}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{f_{i+1}(\mathbf{d})\} \\ + B_{(\mathbf{w_{i+1,i+2}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i-1}})}\{f_{i+1}(\mathbf{d})\} \\ + B_{(\mathbf{w_{i+1,i+2}})}\{B_{\Psi_{\mathbf{w_{i,i+1}}}(\mathbf{w_{i,i+1}})}\{f_{i+1}(\mathbf{d})\}\end{aligned} \tag{A.6}$$

If we compare RHS of A.6 with that of A.2, it becomes apparent that the order of blurring functions is changed. These blurring functions are in essence spatially-varying convolution operators that in general do not have commutative property. However, they are locally commutative if the kernel changes locally small. Therefore, we conclude that under certain conditions,

$$\|\Psi_{\mathbf{w_{i,i+1}}}(k_i(\mathbf{d})) - k_{i+1}(\mathbf{d})\|^2 \approx 0 \tag{A.7}$$

A.1 Pseudocode for MB-CLG

Data: A sequence of T motion blurred frames: $\{f_i\}$

Result: Forward and backward flow fields: $\{\mathbf{w}_{i,i+1}\}$, $\{\mathbf{w}_{i+1,i}\}$

Generate Gaussian pyramid of L levels: $\{g_i\}_l$;

$\{\mathbf{w}_{i,i+1}\}_1 \leftarrow 0$, $\{\mathbf{w}_{i+1,i}\}_1 \leftarrow 0$; `// initialization`

for $l \leftarrow 1$ **to** L **do**

  for $i \leftarrow 1$ **to** T **do**

    if $mean\ \{\mathbf{w}_{i,i+1}\}_{l-1} + \{\mathbf{w}_{i,i-1}\}_{l-1} > \gamma$ **then** `// blurring required`

      $\Psi_{\uparrow\{\mathbf{w}_{i+1,i}\}_{l-1}}(\uparrow \{\mathbf{w}_{i+1,i}\}_{l-1})$; `// warp the flow`

      $\Psi_{\uparrow\{\mathbf{w}_{i+1,i}\}_{l-1}}(\uparrow \{\mathbf{w}_{i+1,i+2}\}_{l-1})$;

      $\Psi_{\uparrow\{\mathbf{w}_{i,i+1}\}_{l-1}}(\uparrow \{\mathbf{w}_{i,i-1}\}_{l-1})$;

      $\Psi_{\uparrow\{\mathbf{w}_{i,i+1}\}_{l-1}}(\uparrow \{\mathbf{w}_{i,i+1}\}_{l-1})$;

      ;

      $\{k_i\}_l \leftarrow addblur(\{g_i\}_l)$; `// according to`

      $\{k_{i+1}\}_l \leftarrow addblur(\{g_{i+1}\}_l)$;

    end

    else `// if blurring is not required`

      $\{k_i\}_l \leftarrow \{g_i\}_l$;

      $\{k_{i+1}\}_l \leftarrow \{g_{i+1}\}_l$;

    end

    $bA = \alpha \times (K - (K-1)exp(\frac{-\nabla\uparrow\{\mathbf{w}_{i+1,i}\}_{l-1}}{2\sigma_d^2}))$; `// backward A`

    $fA = \alpha \times (K - (K-1)exp(\frac{-\nabla\uparrow\{\mathbf{w}_{i,i+1}\}_{l-1}}{2\sigma_d^2}))$; `// forward A`

    ;

    $\{\mathbf{w}_{i,i+1}\}_l \leftarrow estimateflow(\{g_i\}_l, \{g_{i+1}\}_l, \{\mathbf{w}_{i,i+1}\}_{l-1}, fA)$;

    $\{\mathbf{w}_{i+1,i}\}_l \leftarrow estimateflow(\{g_{i+1}\}_l, \{g_i\}_l, \{\mathbf{w}_{i+1,i}\}_{l-1}, bA)$;

  end

end

Algorithm 1: MB-CLG implementation algorithm

A.2 Frame interpolation details

Unless otherwise specified, we use Bicubic Interpolation for constructing a warped image $g(\mathbf{d})$, given the original image $f(\mathbf{d})$ of size $M \times N$, and a flow field $\mathbf{w}(\mathbf{d}) = (u(\mathbf{d}), v(\mathbf{d})$. Vector $\mathbf{d} = (x, y)$ is the position vector. Such warping followed by proper interpolation is then denoted as,

$$g(\mathbf{d}) = \Psi_{\mathbf{w}}(f(\mathbf{d})) \tag{A.8}$$

Initially, let's consider a grid of size $M \times N$ for the warped image g. By definition, each pixel $\mathbf{p_i} = (x_i, y_i)$ in image f is projected onto the grid of g according to,

$$\begin{aligned} & \mathbf{p_i} \rightarrow \mathbf{p_i} + \mathbf{w}(\mathbf{p_i}) \\ or, \quad & \mathbf{p_i} : (x_i, y_i) \rightarrow \mathbf{q_i} : (x_i + u(x_i, y_i), y_i + v(x_i, y_i)) \end{aligned} \tag{A.9}$$

Since transformed pixels do not necessarily fall in integer coordinates, we will need to perform interpolation on the grid. Thus, we find 16 nearest $\mathbf{q_i}$s to each integer position, calculate weights a_{ij} according to Bicubic Interpolation to get,

$$g(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \tag{A.10}$$

It is worthwhile to mention in most scenarios, the warped image g will include a region of NaN values, especially near the boundary that lie outside of the new grid. However, other stages of the algorithm cannot deal with such region. Therefore, they need to be detected and replaced with zero. If we happen to have the target image as well, it would be possible to replace the same region from the target image, rather than zeros.

A.3 Warping the flow

In chapter 4, a step in the proposed method, MB-CLG, was briefly described and referred to as *warping the flow*. In other words, given $\mathbf{w_i} = (u_i, v_i)$ that results in motion blur, and $\mathbf{w_t} = (u_t, v_t)$ that warps source image onto target image, the task is to warp $\mathbf{w_i}$ according to $\mathbf{w_t}$, denoted as

$$\mathbf{w_f} = \Psi_{\mathbf{w_t}}(\mathbf{w_i}) \tag{A.11}$$

In general, a flow field can be thought of a set of linked scene points on image f_i where the first set of points are the regular $M \times N$ grid of pixels, i.e. $\mathbf{W_i} = \{(\mathbf{p_i}, \mathbf{q_i}) | \mathbf{p_i} \in \mathbb{N}^2, \mathbf{q_i} \in \mathbb{R}^2\}$. When we warp a flow onto another image f_j, the set of linked points should ideally be identical, i.e. the same scene points are linked by $\mathbf{w_f}$ in image f_j. The set of linked points by $\mathbf{w_i}$, therefore, must be separately warped by $\mathbf{w_t}$ and be linked again. Let $\mathbf{p}(x, y)$ denote the coordinates of the first point in the paired linked points, and let $\mathbf{q}(x, y)$ the coordinates of the second point. Therefore,

$$\mathbf{w_t}(x, y) = \mathbf{q}(x, y) - \mathbf{p}(x, y) \tag{A.12}$$

is another expression for the flow field aligning source and target images. Now we proceed by defining

$$\begin{aligned} \mathbf{p}'(x, y) &= \mathbf{p}(x, y) + \mathbf{w_t}(\mathbf{p}(x, y)) \\ \mathbf{q}'(x, y) &= \mathbf{q}(x, y) + \mathbf{w_t}(\mathbf{q}(x, y)) \end{aligned} \tag{A.13}$$

as projected coordinates, respectively for the first and the second points in the linked set. However, $\mathbf{q}(x, y)$ are not necessarily integer coordinates. Therefore, one should perform interpolation to get values for $\mathbf{w_t}(\mathbf{q}(x, y))$. Although a nearest neighbor sounds to be sufficient, we use Bilinear interpolation. Now we can define a new set of linked points: $\mathbf{W_f} = \{(\mathbf{p'_i}, \mathbf{q'_i}) | \mathbf{p'_i}, \mathbf{q'_i} \in \mathbb{R}^2\}$. This new set describes the same linked scene points, but transformed onto the coordinates of f_j. Hence, it is the same warp $\mathbf{w_i}$, warped by $\mathbf{w_t}$. Furthermore, $\mathbf{p'_i}$ are not necessarily integers in the

expression for $\mathbf{W_f}$. A practical flow field was defined to be: $\mathbf{w} : \mathbb{N}^2 \rightarrow \mathbb{R}^2$. Therefore, we need to perform another interpolation on the non-uniform grid of $\mathbf{p'_i}$ positions. We employ a Bicubic interpolation on $\mathbf{W_f}$ to fill in the grid $\mathbf{p_i}$, with interpolated values $\mathbf{q''_i}$. In particular, we use `scatteredInterpolant()` function in MATLAB implementation of MB-CLG. After non-uniform interpolation, the set of new linked points can be expressed as $\mathbf{W'_f} = \{(\mathbf{p_i}, \mathbf{q''_i}\}$. With our regular notation, this flow field is represented as,

$$\mathbf{w'_f} = \mathbf{q''}(x, y) - \mathbf{p_i}(x, y) \tag{A.14}$$

where $\mathbf{p_i}(x, y) = (x, y)$ is the regular grid of pixels.

Warping a flow is particularly important in applying blur functions of the source image to the target image. Based on the blur model described in Chapter 3, a blur functions can be uniquely expressed in terms of the underlying flow field. These blur functions, when transformed onto the coordinates of another image, must still produce exact the same blurring artifacts on the target image. Therefore, we warp the underlying flow fields, and generate new blur functions based on them.

Bibliography

[1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vision*, 56(3):221–255, February 2004.

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision*, 92(1):1–31, March 2011.

[3] Connelly Barnes, Eli Shechtman, DanB. Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 29–43. Springer Berlin Heidelberg, 2010.

[4] Moshe Ben-Ezra and Shree K. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, 2004.

[5] Pravin Bhat, Ke Colin Zheng, Noah Snavely, Aseem Agarwala, Maneesh Agrawala, Michael F. Cohen, and Brian Curless. Piecewise image registration in the presence of multiple large motions. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern, CVPR 2006*, pages 2491–2497. IEEE Computer Society, 2006.

[6] Michael J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63(1):75–104, January 1996.

[7] Dong bok Lee, Shin-Cheol Jeong, Yun-Gu Lee, and Byung Cheol Song. Video deblurring algorithm using accurate blur kernel estimation and residual deconvolution based on a blurred-unblurred frame pair. *Image Processing, IEEE Transactions on*, 22(3):926–940, March 2013.

[8] Thomas Brox, Andrs Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. pages 25–36. Springer, 2004.

[9] Andrs Bruhn, Joachim Weickert, and Christoph Schnrr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[10] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 168–172 vol.2, Nov 1994.

[11] Wei-Ge Chen, N. Nandhakumar, and Worthy N. Martin. Image motion estimation from motion smear-a new computational model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):412–425, Apr 1996.

[12] Sunghyun Cho, Y. Matsushita, and Seungyong Lee. Removing non-uniform motion blur from images. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[13] Shengyang Dai and Ying Wu. Motion from blur. In *In Proc. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[14] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981.

[15] Hailin Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 18–25 vol. 2, June 2005.

[16] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. Doctoral thesis, Massachusetts Institute of Technology, May 2009.

[17] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *a*, pages 674–679, 1981.

[18] C. Mei and I. Reid. Modeling and generating complex motion blur for real-time tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[19] Travis Portz, Li Zhang, and Hongrui Jiang. Optical flow in the presence of spatially-varying motion blur. In *CVPR*, pages 1752–1759. IEEE, 2012.

[20] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.

[21] Y. Schoueri, Milena Scaccia, and I. Rekleitis. Optical flow from motion blurred color images. In *Computer and Robot Vision, 2009. CRV '09. Canadian Conference on*, pages 1–7, May 2009.

[22] Deqing Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439, June 2010.

[23] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.

[24] D.L. Tull and A.K. Katsaggelos. Regularized blur-assisted displacement field estimation. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 85–88 vol.3, Sep 1996.

[25] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l1 optical flow. In Daniel Cremers,

Bodo Rosenhahn, AlanL. Yuille, and FrankR. Schmidt, editors, *Statistical and Geometrical Approaches to Visual Motion Analysis*, volume 5604 of *Lecture Notes in Computer Science*, pages 23–45. Springer Berlin Heidelberg, 2009.

[26] Joachim Weickert and Christoph Schnrr. A theoretical framework for convex regularizers in pde-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, 2001.

[27] Jiangjian Xiao, Hui Cheng, Harpreet Sawhney, Cen Rao, and Michael Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *Computer Vision ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 211–224. Springer Berlin Heidelberg, 2006.

[28] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Blurred/non-blurred image alignment using sparseness prior. In *ICCV'07*, pages 1–8, 2007.