2016 Fourth International Conference on 3D Vision

Optical Flow for Rigid Multi-Motion Scenes

Tomas Gerlich
Department of Computer Science
University of Illinois at Chicago
tgerli2@uic.edu

Jakob Eriksson*
Department of Computer Science
University of Illinois at Chicago
jakob@uic.edu

Abstract

We observe that in many applications, the motion present in a scene is well characterized by a small number of (rigid) motion hypotheses. Based on this observation, we present rigid multi-motion optical flow (RMM). By restricting flow to one of several motion hypotheses, RMM produces more accurate optical flow than arbitrary motion models.

We evaluate an algorithm based on RMM on a novel synthetic dataset, consisting of 12 photo-realistically rendered scenes containing rigid vehicular motion and a corresponding, exact, ground truth. On this dataset, we demonstrate a substantial advantage of RMM over general-purpose algorithms: going from 36% outliers with the DiscreteFlow algorithm, to 26% with ours, with a mean error reduction from 8.4px to 6.9px. We also perform qualitative evaluation on real-world imagery from traffic cameras.

1. Introduction

Computing optical flow, a dense pixel-wise correspondence field between a pair of successive video frames, is a fundamental problem in computer vision. It provides low-level and rich description of scene motion and many higher-level image understanding applications rely on its accurate estimation. In general, computing optical flow from a pair of images is an under-constrained problem. One widely used additional constraint is that the flow must be smooth: it does not change abruptly from one pixel to the next.

We introduce an additional constraint: the underlying motion in the scene consists of translation and rotation of rigid objects. Specifically, we introduce a multi-motion model, consisting of several distinct fundamental matrices [6, 9]. The multi-motion model is taken as input, and could be determined in several ways. For example, it may be provided by a human user, estimated from long term statistical processing, or estimated from feature matching on a frame-pair basis. Below, we show that restricting the optical flow solution space to this multi-motion model improves performance on conforming scenes.

Road traffic is a common type of scene that is well characterized by rigid multi-motion. We use this scenario as the focus of our evaluation efforts. Applications of optical flow in traffic surveillance video include enforcement, statistical studies, signal control optimization, and more. We hypothesize that leveraging epipolar geometry constraints will result in more accurate optical flow than that offered by general purpose algorithm, while keeping computational complexity low. Moreover, we believe that projective geometry approaches toward estimating optical flow for dynamic scenes have not been sufficiently explored in previous work, a gap which we seek to fill here.

For evaluation, we introduce a new dataset containing rendered traffic monitoring scenes. Starting from 3D models of a street scene, we use Blender, a freely available image rendering engine, to produce photorealistic imagery of moving vehicles. Through modifications to Blender, we produce precise optical flow ground truth, including areas of occlusion. Evaluation is then a simple matter of comparing the output produced to the precise ground truth provided. Our primary contributions are as follows:

- We present a novel optical flow algorithm that performs joint motion (rotation+translation) segmentation and epipolar flow estimation.
- We contribute a novel synthetic imagery dataset with photorealistic features and precise ground truth.
- We quantitatively compare the performance of our algorithm to that of several existing state-of-the-art algorithms, on the synthetic dataset.
- Finally, we qualitatively illustrate the performance of our method on real imagery, and compare it to several state-of-the-art algorithms.

Below, §2 briefly describes the related work, followed by a description of the optical flow problem in §3, our algorithm §4, and the evaluation §5 before we conclude in §6.

*This material is based upon work supported by the U.S. National Science Foundation under grant CNS-1149989, and by the Illinois Department of Transportation under grant ICT R27-169.

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.30

2. Related Work

Until recently, the top performing optical flow algorithms have usually originated from the family of *variational* methods, which aim to formulate a robust differential at each pixel. The inclusion of geometry constraints as part of the variational approach was explored in [17] and [14] which address the joint problem of optical flow and a *single* fundamental matrix estimation. [18] also uses a predetermined fundamental matrix to steer the flow solution to be consistent with the epipolar geometry. While variational optical flow algorithms tend to be very accurate on scenes with *small* pixel movement, they usually rely on a sub-sampling pyramidal scheme to handle large displacements. We design an algorithm suitable for large (in our case 256px) displacements without using image pyramids.

An alternative way of imposing geometry constraints appeared in the work of [12] who observe that general optical flow algorithms frequently contain a smoothness term to encourage flow similarity among neighboring pixels. This often results in unnecessary penalization of any deviation of neighboring flow estimates even though such deviation can often be well explained by the true rigid motion appearing in the 3D scene. As a remedy, the paper proposes to over-parametrize the solution space at each pixel by also adding per-pixel (affine, rigid, or pure translation) motion parameters and, in the variational framework, to impose smoothness constraints on those higher-level parameters. The work in [8] also advocates for the use of over-parametrization but goes a step further and adds a per-pixel homography to model the motion. The high-dimensional energy function is then optimized using a faster variant of a particle belief propagation algorithm.

The seminal work of [2] focused on the problem of *interpreting* an optical flow generated by independently moving rigid objects. The method computes the joint segmentation and rigid motion model estimation for each independent segment. A set of experiments then showed the feasibility and advantages of simultaneously solving for segmentation and rigid multi-motion parameter estimation in order to reason about motion in 3D. On the other hand, our paper focuses on computing the optical flow itself with the assumption of having a small set of pre-estimated motion hypotheses available.

In the context of modern optical flow algorithms, our scenario is closely related to [10] which also focuses on solving for optical flow applied to traffic imagery. Their algorithm is designed for the autonomous driving application and hence it accepts input from a pair of stereo cameras at two time instants for a total of four images. The algorithm then estimates jointly the stereo disparity and optical flow (*i.e*. the scene flow). The traffic scene is assumed to be composed of rigidly-moving piece-wise planar segments where each pixel moves according to the estimated homography of the segment which it is assigned to. The stereo pairs are used to initialize the rotation and translation of the planar segments and therefore a stereo camera is *required* to solve for optical flow. In contrast, our algorithm only assumes as input two images taken at different times.

The work of [19] is closer to ours in the sense that they too process images from a monocular camera. They quantitatively demonstrate the advantages of using epipolar constraints to compute optical flow. However, with their main application focus being on autonomous driving, they estimate epipolar flow with respect to a moving camera as it observes a *static* scene. In our work, we focus on the scenario of a single static camera and moving vehicles.

The recent successful approaches of [13] and [11] follow a two-step process: first they perform sparse matching to produce high-quality matches which they subsequently interpolate at each pixel to produce a dense result. This interpolation step is locally aware of edges and prevents over-smoothing. However, the first step to this pipeline is to produce accurate matches which are free of outliers and our algorithm aims to produce such semi-dense initial matches.

3. Problem Description

To tackle the problem of optical flow estimation for traffic, we leverage the fact that our objects of interest, *i.e*. cars, trucks, or buses, are non-deformable and their motion is governed by geometrical rigid-body motion constraints. This observation allows us to take full advantage of the algorithmic approaches developed in the computer vision sub-area of stereo estimation and structure-from-motion. To better understand the connection between estimating stereo, epipolar and optical flow, we begin with a brief comparison while [6, 9] offer more details.

3.1. Stereo

In *stereo*, we assume a *static* scene is observed by a pair of left and right cameras (C_L, C_R) whose centers of projection are horizontally separated by a non-zero baseline translation $T = (t_x, 0, 0)$. Without loss of generality, we may designate the world origin coordinate system to coincide with C_L and we are interested in deducing the 3D-world coordinates X_i of each pixel x_i observed in image I_L. This setting results in a pair of images (I_L, I_R) where pixels x_i in I_L are horizontally shifted (toward left) as observed at corresponding pixel locations x'_i [1] in I_R. The fact that the corresponding pixel x'_i is to be found on the same image row as x_i is due to the geometry of the camera pair which guarantees the pixel correspondences to lie on the same *epipolar* line. Thus, each pixel in I_L is displaced toward the negative x-direction by a certain positive *disparity* value d_i which

[1] We use the (.)' notation to represent an entity in the second frame with respect to the argument in parentheses that appears in the first frame

results in the correspondence field (x_i, x_i') for each pixel, where $x_i' = x_i - d_i$. Because d_i is inversely proportional to the depth of the corresponding 3D point X_i, estimating d_i along with the knowledge of the baseline translation T and the focal length of the cameras allows us to deduce the 3D-coordinates X_i using triangulation.

3.2. Epipolar Flow

The problem of *epipolar* flow estimation is a generalization of stereo. In this setting, we assume an observer can be represented by a single camera which moves within a static environment and where the pair of images I_L^t and I_L^{t+1} is acquired at times t and $t+1$ (for brevity, we can denote this image pair simply as I_0 and I_1). Unlike in stereo, here the *ego*-motion of the camera is in general not constrained to be represented by horizontal displacement (translation) with 1 degree of freedom (d.o.f) only. Instead, the camera can move freely in space where this rigid-body motion can be concisely described using the translation vector $T \in \mathbb{R}^3$, with 3 d.o.f and a rotation matrix $R \in \mathbb{R}^{3\times3}$, also with 3 d.o.f. For illustration, this type of motion can arise for instance when a person captures images while walking and holding a camera in hand. Nevertheless, despite the added motion complexity compared to the stereo case, the formation of images is also governed entirely by epipolar geometry. As a consequence, each pixel location x_i induces an epipolar line l_i' along which the correspondence x_i' is to be found and where the pair (x_i, x_i') is also separated by disparity d_i. However, unlike in stereo, the pencil of epipolar lines does not generally follow the horizontal lines (image rows) but instead each line intersects a single point in the image plane (the epipole) while each line also changes its direction (smoothly) with respect to its neighbors. Thus, when a camera moves approximately forward toward the direction of its optical axis, the epipole often resides in the neighborhood of the image center and the epipolar lines pass through the epipole.

To ensure that *both* x_i and x_i' lie on l_i', as was the case in *rectified* stereo, it may be necessary to first estimate R in order to remove the effect of the rotational motion observed in the image planes. The images I_0 and I_1 can then be transformed to simulate the simpler case when both image planes are parallel. Consequently, the correspondence field can be parametrized as $x_i' = x_i - d_i e_u$ where e_u is the unit-direction vector *at* x_i toward the epipole. Note that when the observer moves forward the disparity values are positive in the above equation because all pixels appear to be moving away from the epipole.

3.3. Optical Flow

The challenge with estimating *optical* flow is that in this scenario the scene can no longer be assumed to remain static because multiple independently moving *dynamic* objects can appear in the scene, in addition to the possible ego-motion of the camera itself. On the other hand, it is important to notice that the epipolar constraints still hold, and this is the case even in the scenario of static observer (static camera) and a moving object. It can be helpful to imagine the scenario in terms of estimating the epipolar flow as before, with the difference that all pixel movements would now correspond to the inverse of what we would expect in the moving camera with static object case. However, this simple trick enables us to leverage epipolar constraints as before and to reason about optical flow corresponding to rigid-body motion in 3D even with a static camera.

Continuing with the description of the most general scenario of moving camera with multiple moving objects we can see that the different objects' motion(s) can be assumed to be specified entirely using a small set of rigid-body motion models $\{(T_1, R_1), \ldots, (T_K, R_K)\}$ with K corresponding to the number of different models. Note that K can be less than the number of objects in the scene because multiple objects' motions can possibly be specified using a single motion model. This many-to-one relation can arise for example when multiple cars share the same direction of movement or when the ego-motion of the camera is close enough to any of the objects' motions, for instance when the observer moves forward while a car ahead moves in the opposite direction toward the camera. Finally, in contrast to stereo and epipolar flow where the pixel correspondence field was constrained by the epipolar geometry of a single motion model across *all* the pixels in the image plane, here the pixel flow at different regions of the image can be governed by possibly different motion models. Consequently, before estimating the disparity for each pixel, one must assign a motion model to each pixel, and as such segment the image based on the provided motion models.

3.4. Parametrization

To be able to solve for optical flow leveraging the epipolar geometry constraints, one must derive a suitable parametrization for the optical flow solution space. In this paper, we will represent each motion model as it appears in the image plane (in pixel units). This can be accomplished using a *fundamental* matrix $F \in \mathbb{R}^{3\times3}$ which is a matrix encompassing the camera's calibration matrix $K \in \mathbb{R}^{3\times3}$ and the vehicle's motion parameters T and R. Here K and K' contain the focal length and the optical axis offset (in pixels) of the camera for each image pair in the sequence. We can write $F = K'^{-\top}[T]_\times R K^{-1}$ where $K = K'$ in case our camera does not change its intrinsic parameters (*e.g.* by zooming) when capturing I_0 and I_1. Moreover, the operator $[.]_\times$ in the previous equation converts the argument in the brackets into its skew-symmetric matrix representation suitable for expressing a vector cross product.

The parametrization for the correspondence field nat-

urally follows from the fundamental matrix constraint $\tilde{\mathbf{x}}_i'^{\top} F \tilde{\mathbf{x}}_i = 0$. Here, we have used the bold-face notation $\mathbf{x}_i = (x_i, y_i)^{\top}$ to represents the full pixel coordinates, while the operator $\tilde{(.)}$ transforms the argument in parentheses into its homogeneous coordinate representation by simply appending the constant 1 to the vector to form $\tilde{\mathbf{x}}_i = (x_i, y_i, 1)^{\top}$. For each pixel location $\mathbf{x}_i$ in image I_0, the fundamental matrix induces a line in the next frame I_1:

$$l_i' = F\tilde{\mathbf{x}}_i \tag{1}$$

where $l' \in \mathbb{R}^3$ is the parametrization of the line, and the correspondence $\mathbf{x}_i'$ is guaranteed to lie somewhere along that line. On the other hand, it is important to note that because F can contain the rotational component R, the source point $\mathbf{x}_i$ will in general *not* lie on l_i'. As a consequence, a point of origin needs to be established on line l_i' so that the correspondence $\mathbf{x}_i'$ can be parametrized with respect to it. There is an obvious difficulty with the naive approach of designating the point of intersection of l_i' with one of the image borders because, in the worst case, we would need to ensure a large search space up to the length of the image diagonal to ensure that $\mathbf{x}_i'$ can be parametrized. To keep the search space small and hence an inference procedure quick, a heuristic approach should be employed.

The method in [3] uses the set of sparse matches previously used to pre-estimate a given F as the heuristic to find the initial guess for the location of $\mathbf{x}_i'$. First, a similarity transformation is fit to the sparse matches which transforms $\mathbf{x}_i$ to a point in I_1 and that point is designated as the point of origin where $\mathbf{x}_i'$ is parametrized with respect to it. However, this approach explicitly requires the knowledge of the sparse matches for each F which may not always be available or practical as we will observe in a later section. Therefore, in this paper we will develop a different heuristic which is simpler and more general.

First we calculate the closest point from $\mathbf{x}_i$ that lies on l_i'. This is accomplished by a simple perpendicular projection $\mathcal{P}(.,.)$ where the arguments in parentheses specify the source point and the fundamental matrix (FM) and where we utilize Eq. 1. Thus,

$$\mathbf{x}_i^p = \mathcal{P}(\mathbf{x}_i, F) \tag{2}$$

where $\mathbf{x}_i^p$ is the heuristic point of origin. This new point of origin yields the parametrization for the correspondence,

$$\mathbf{x}_i' = \mathbf{x}_i^p + d_i \mathbf{e}_u' \tag{3}$$

where $d_i \in \mathbb{R}$ and $\mathbf{e}_u' \in \mathbb{R}^2$ is the unit-directional vector from x_i^p toward the epipole e'. This vector also corresponds to the unit-direction of the line l_i', however in our implementation we first extract the homogeneous coordinates of the epipole $e' \in \mathbb{R}^3$ (which is constant for a particular F) and then calculate the unit vector from x_i^p toward e'. The epipole corresponds to the right *null*-space of F because $F^{\top} e' = 0$ and hence can be calculated from the singular value decomposition (SVD) of $F^{\top}$.

Both the heuristic for finding $\mathbf{x}_i^p$ and the parametrization for the correspondence field have the advantage that they also naturally cover the earlier cases of stereo and epipolar flow. To see this, first we can observe that both stereo and epipolar flow typically process rectified imagery, where the rectification pre-processing step removes the possible rotational motion component observed in the images and hence in F. As a consequence, Eq. 2 reduces to $\mathbf{x}_i^p = \mathbf{x}_i$ and the parametrization for stereo and epipolar flow follows from Eq. 3, where $\mathbf{e}_u' = (-1, 0)^{\top}$ for the stereo case. Therefore, our heuristic and parametrization can be used in all three cases for stereo, epipolar flow, and optical flow.

3.5. Model

We assume a set of FM hypotheses $\mathcal{F} = \{F_1, \ldots, F_K\}$ is initialized (we will discuss an alternative approach for pre-estimating $\mathcal{F}$ in §4.2). To represent the important special case where an image region is stationary and hence no well-conditioned motion parallax occurs, we include the special *no-motion* FM into $\mathcal{F}$ and define a binary label $f_k \in \{\delta_{mo}, \delta_{nm}\}$ to specify whether a given F_k represents the motion or no-motion hypothesis. This results in the binary vector $f = (f_1, \ldots, f_K)^{\top}$ corresponding to $\mathcal{F}$.

We model the task of joint motion segmentation and optical flow disparity estimation as a discrete labeling problem. Here, the labels consist of the pairs (m_i, d_i), where $m_i \in \mathcal{M} = \{1, \ldots, K\}$ assigns pixel $\mathbf{x}_i$ to follow motion hypothesis F_{m_i} and $d_i \in \mathcal{D} = (\delta_{min}, \ldots, \delta_{max})$ and where integers $\delta_{min} \leq \delta_{max}$ define the ordered set.

Data Cost: this energy term expresses the quality of the $(\mathbf{x}_i, \mathbf{x}_i')$ match parametrized by Eq. 3. We assume that the photometric properties of the match locations among images I_0 and I_1 should be similar to constitute a good match. Here, we measure the difference in *edginess* expressed by computing the $L2$ norm among DAISY descriptors [16]. This descriptor is robust to illumination changes and was originally designed as a dense descriptor similar to SIFT but much faster to compute.

The match quality among two points is measured as

$$\psi(\mathbf{x}_i, \mathbf{x}_i') = -\log\left(1 - \frac{\|\phi_0(\mathbf{x}_i) - \phi_1(\mathbf{x}_i')\|_2}{Z}\right) \tag{4}$$

where the function $\phi_t(.)$ computes the descriptor vector at the location in parentheses in image I_t, and Z is a normalization constant to ensure the argument of the logarithm is in range $[0, 1]$. In practice we can estimate Z as the maximum value of the $L2$ norm of the difference among the descriptors ϕ_0 and ϕ_1 observed across all pixels whose explored matching point hypotheses fall within the image boundary.

We can specify the data cost energy term as

$$E_{data}(\mathbf{x}_i, m_i, d_i \,|\, I_0, I_1, \mathcal{F}, f) = \begin{cases} \psi(\mathbf{x}_i, \mathbf{x}'_i) & \text{if } f_{m_i} = \delta_{mo} \text{ and } \|\mathbf{x}_i - \mathbf{x}'_i)\| \geq \tau_{mo} \text{ and} \\ & \mathbf{x}'_i \text{ is within } I_1 \text{ boundary} \\ \psi(\mathbf{x}_i, \mathbf{x}_i) & \text{if } f_{m_i} = \delta_{nm} \\ \infty & \text{otherwise} \end{cases} \tag{5}$$

where $\mathbf{x}'_i = \mathbf{x}^p_i + d_i \mathbf{e}'_{m_i}$ from Eq. 3, $\mathbf{x}^p_i$ follows from Eq. 2 and $\mathbf{e}'_{m_i}$ is defined by F_{m_i} and is the unit direction at $\mathbf{x}^p_i$ toward the epipole e'_{m_i}. The scalar threshold τ_{mo} specifies the minimum displacement for a match to be considered to follow any other than the stationary background (no-motion) hypothesis.

Smoothness Cost: we encourage the disparities among neighboring pixels to change smoothly if both pixels follow the same motion model. In addition, we also prefer neighboring pixels to remain in the same motion model. Both conditions can be encoded concisely as

$$E_{smo}(d_i, d_j, m_i, m_j) = \begin{cases} \lambda_1 |d_i - d_j| & \text{if } m_i = m_j \text{ and } |d_i - d_j| \leq 2 \\ \lambda_3 & \text{if } m_i = m_j \text{ and } |d_i - d_j| > 2 \\ \lambda_{mch} & \text{if } m_i \neq m_j \end{cases} \tag{6}$$

where $0 \leq \lambda_1 \leq \lambda_3$ and $\lambda_{mch} \leq \lambda_3$ are positive scalars. Here, λ_1 is the small penalty proportional to $0, 1$ or 2 levels of disparity change while λ_3 is the cost of changing disparity by 3 or more levels. Furthermore, λ_{mch} represents the constant penalty for changing motion assignment among neighboring pixels. The piece-wise linear form of the smoothness function makes it robust to outliers and this type of function (and its variations) is often used in the stereo literature to prevent over-smoothing across motion boundaries which is another important property for estimating accurate optical flow. Notably, both the works of [7] and [3] also use a similar smoothness term.

Objective Function: the quality of a particular motion model and disparity assignment at each pixel is measured using the sum of the data and smoothness energy terms. Hence, the full objective function to be minimized is

$$\mathbf{m}^*, \mathbf{d}^* = \arg\min \left\{ \sum_i \Big(E_{data}(\mathbf{x}_i, m_i, d_i) + \sum_{j \in \mathcal{N}_8(i)} E_{smo}(d_i, d_j, m_i, m_j) \Big) \right\}. \tag{7}$$

Method	fg (%)	fg (px)	bg (%)	bg (px)	comb (%)	comb (px)
1280x720						
MMSGM-fGT	26.4	8.87	2.30	1.80	6.96	3.45
MMSGM-fGT+EF	**26.1**	**6.91**	**1.23**	0.46	**5.92**	**1.89**
DISCFLOW+EF [11]	36.3	8.41	1.53	0.44	8.08	2.14
DM+EF [13]	49.4	12.7	2.26	**0.42**	11.2	3.05
FLOWNET [5]	71.3	13.5	10.6	2.31	22.1	4.59
CNL [15]	68.1	29.4	13.0	6.12	22.8	10.7
320x180 (QR)						
MMSGM-fGT	**14.8**	2.91	4.92	1.09	7.11	1.54
MMSGM-fGT+EF	21.3	3.06	2.48	0.33	**5.76**	**0.79**
MMSGM-f64	19.7	3.77	5.97	1.35	8.90	1.93
MMSGM-f64+EF	18.8	**2.35**	3.71	0.41	6.79	0.83
DISCFLOW+EF	22.6	3.26	2.66	0.39	6.30	0.89
DM+EF	29.1	3.40	**1.26**	**0.18**	6.79	0.86
FLOWNET	76.0	7.64	17.3	1.95	27.0	3.05
CNL	46.2	7.57	5.71	0.77	13.5	2.20

Table 1. **Top**: algorithms were tested on full resolution *FlowTruth* testing dataset where both the % outliers above 3px threshold measure and the end-point error (px) for non-occluded pixels w.r.t ground truth is averaged over the 6 photo-realistic scenes. **Bottom:** same as above with input images resized to quarter size.

Here, $\mathbf{m}^*$ and $\mathbf{d}^*$ represent the *optimal* vectors $\mathbf{m} = (m_1, \ldots, m_n)^\top$ and $\mathbf{d} = (d_1, \ldots, d_n)^\top$ of motion and disparity assignment (m_i, d_i) at each pixel i, with $n = width \times height$ being the number of pixels. Furthermore, the operator $\mathcal{N}_8(i)$ represents the 8-pixel neighborhood indices centered around pixel i. We will discuss our method for optimizing the objective function next.

4. Algorithm

The optimization of Eq. 7 can be viewed as an inference problem in a pairwise Markov random field (MRF). This is an NP-hard problem in general and to make the inference procedure tractable we can only seek an approximate solution. However, qualitatively comparable (but still approximate) solutions can be reached using general inference procedures with different time complexity. For instance, the method of [3] minimizes a similar objective function using the multilabel *graph-cut* alpha-expansion procedure. This method produces good results, but tends to suffer from a higher runtime. In their experimental setup, they report a runtime of 10-20 minutes on image resolution of 320x240 pixels with few motion models of 40 disparity labels each and with two image pyramid levels. In contrast, we explore a quicker inference procedure, which accomodates the bigger search space needed for large displacement optical flow *without* pyramidal sub-sampling.

We draw inspiration from the stereo literature and in particular we seek to adapt the recently very popular method of Semi-Global Matching (SGM) [7] for the multi-motion optical flow task. In that method, the full image is traversed over multiple canonical directions starting from the image borders. Often only 8 directions (north, northeast, east,

Algorithm 1 Multi-Motion Semi-Global Matching

Require: $I_0, I_1, \mathcal{F}, f$

1: **for all** pixel i, motion $m \in \mathcal{M}$, disparity $d \in \mathcal{D}$ **do**
2: $\quad C[i,m,d] \leftarrow 0$
3: **end for**
4: **for all** directions $r \in \{n, ne, e, \ldots, nw\}$ **do**
5: $\quad$ **for all** unique line l in direction r **do**
6: $\quad\quad$ **for all** $m \in \mathcal{M}, d \in \mathcal{D}$ **do**
7: $\quad\quad\quad C_{prev}[m,d] \leftarrow 0$
8: $\quad\quad$ **end for**
9: $\quad\quad$ **for all** pixel i in l **do**
10: $\quad\quad\quad$ **for all** $m \in \mathcal{M}, d \in \mathcal{D}$ **do** ▷ Eq. 5, 6
11: $\quad\quad\quad\quad C_{cur}[m,d] \leftarrow E_{data}(\mathbf{x}_i, m, d+\delta_{min}) +$
$\min\{\min_{j \le |2|}\{C_{prev}[m,d+j] + \lambda_1 |j|\},$
$\min_{k \in \mathcal{D}}\{C_{prev}[m,k]\} + \lambda_3,$
$\min_{n \in \mathcal{M} \setminus m, k \in \mathcal{D}}\{C_{prev}[n,k]\} + \lambda_{mch}\}$
12: $\quad\quad\quad$ **end for**
13: $\quad\quad\quad$ **for all** $m \in \mathcal{M}, d \in \mathcal{D}$ **do**
14: $\quad\quad\quad\quad C[i,m,d] \leftarrow C[i,m,d] + C_{cur}[m,d]$
15: $\quad\quad\quad$ **end for**
16: $\quad\quad\quad C_{prev} \leftarrow C_{cur}$
17: $\quad\quad$ **end for**
18: $\quad$ **end for**
19: **end for**
20: **for all** pixel i **do**
21: $\quad m_i^*, d_i^* \leftarrow \arg\min_{m,d}\{C[i,m,d]\}$
22: **end for**
23: **return** $\mathbf{m}^*, \mathbf{d}^*$

etc.) are sufficient for good coverage. The method uses *dynamic programming* for efficient energy minimization and path traversal, and averages the accumulated cost at every pixel from *all* directions equally. As a consequence, the method produces smooth result without streaking artifacts or staircase-like appearance which other stereo methods often suffer from.

We adapt SGM to enable the efficient minimization of our objective function. In our scenario, we seek the joint minimum cost of assigning every pixel to a motion model *and* a disparity while minimizing Eqs. 5 and 6 semi-globally. We call this adaptation of the basic SGM procedure toward the more difficult, larger label-set space problem of joint multi motion segmentation and disparity estimation *Multi-Motion Semi-Global Matching*, or MMSGM. The full procedure is summarized in Algorithm 1.

An efficient implementation can avoid the unnecessary re-computation of the large inner minimum costs appearing in line 11. This can be accomplished for example by keeping track, in a separate lookup array M_{prev} of size $|\mathcal{M}|$, of the overall minimum cost across all disparities in C_{cur} for each motion model m during the process of populating C_{cur}. Alternatively, the same lookup array can be populated at a small (linear) computation cost just before the code reaches line 10. In addition, using M_{prev} we can precompute for a given motion m the minimum cost w.r.t all the *other*-than-m costs in M_{prev} in a computational time which is proportional to $|\mathcal{M}|$. This key additional lookup value M_n enables the efficient computation of the minimum $\min_{n \in \mathcal{M} \setminus m}\{C_{prev}[n,*]\}$ appearing in line 11 when optimizing over a large number of motion models with large sets of disparities.

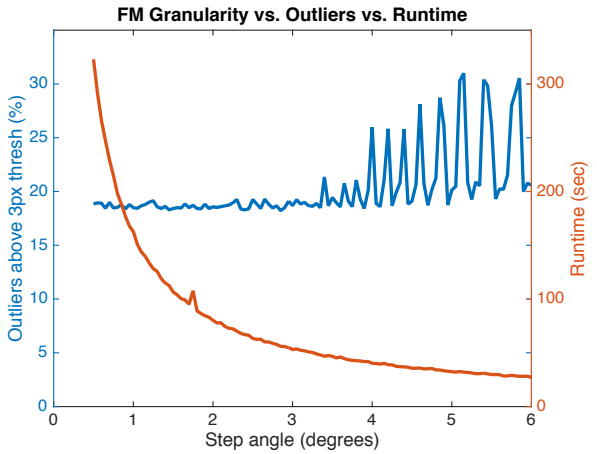

Figure 1. Angular sampling of the the half circle to produce the general fundamental matrix set $\mathcal{F}$. The runtime using 64-threads on 320x180 image with 129 disparities includes all post-processing steps except for EpicFlow.

4.1. Post-processing

After the MMSGM algorithm computes an estimate of the optimal assignment of m_i and d_i for every pixel, we can solve for the optical flow using Eq. 3. As a simple noise reduction technique, we perform the standard consistency check, *i.e*. computing the optical flow also from I_1 toward I_0 and mark a pixel in I_0 occluded if the error among the corresponding flow vectors exceeds a threshold τ_{occ}. In addition, we further investigate all isolated connected components w.r.t each motion assignment and label the segment as occluded if its size falls below a minimum threshold τ_{cc}.

In order to fill the gaps generated by occluded pixels, we use the *EpicFlow* interpolation algorithm described in [13] to produce a smooth and dense result. We denote when this last interpolation step is performed in all our experiments.

4.2. General Motion Models

As a proof-of-concept for the general case, where no pre-estimated set of fundamental matrices $\mathcal{F}$ is supplied as input, we observe that our algorithm can leverage a canonical set of *directional* fundamental matrices. This set is pre-populated using simple angular sampling of the half-circle from 0 to 180 degrees, in sufficient granularity, and ex-

pressing each corresponding F with the directional epipole (at infinity) defined by the angle. Figure 1 illustrates the algorithm performance as a function of the angular sampling interval. A granularity of 3 degrees, which roughly corresponds to 64 directions, achieves a reasonable performance while keeping the computational time within acceptable bounds. Below, we use *f64* to refer to this proof-of-concept approach.

5. Evaluation

We measure the performance and runtime of MMSGM, providing quantitative and qualitative results on synthetic, as well as qualitative results on real-world, imagery. The quantitative evaluation shows MMSGM substantially outperforming the state of the art when accurate motion models are provided, and outperforming or remaining competitive when using uniformly sampled motion models.

5.1. Image Dataset and Ground Truth

We evaluate MMSGM on a novel synthetic dataset, consisting of 12 photo-realistically rendered scenes containing rigid vehicular motion, which we split into 6 training and 6 testing subsets. The scenes were created using 3D modeling and rendering software Blender [1], starting from 3D models of street environments. In contrast with real imagery, this synthetic imagery is accompanied by an exact ground truth, showing the actual movement of the object represented by each pixel of the image. This was produced through a modification to Blender similar to [4].

We also perform qualitative evaluation on real-world traffic camera imagery. Here, no exact ground truth is available for quantitative evaluation. Instead we provide illustrations of the produced flow, for qualitative evaluation.

5.2. Quantitative Evaluation on Synthetic Imagery with Exact Ground Truth

We show our quantitative performance measurements in Table 1. Here, we report the performance of our algorithm with accurate motion models provided (fGT), with uniformly sampled motion models (f64), and both with and without EpicFlow post-processing (EF). Overall, we find that MMSGM significantly outperforms the state of the art, as represented by DISCFLOW+EF (DiscreteFlow [11]). On the full resolution imagery, the number of outliers beyond 3px error was 26% for MMSGM vs. 36% for DISCFLOW+EF, with a mean endpoint error of 6.9px vs. 8.4px. Similar gains occur on quarter resolution imagery.

On the quarter resolution imagery, we also show results from the f64 variant, which uses uniformly sampled motion models instead of accurate provided models as part of the input. Here, the advantage over DISCFLOW+EF is smaller, yet still significant, at 19% outliers vs. 23%, and a mean error of 2.4px vs 3.3px. However, the computational complexity of the f64 variant made it impractical to use on the high-resolution imagery. In Table 2 we report MMSGM runtime including the consistency check and post-processing described in §4.1, without EF interpolation.

resolution	FM #	disparities	1-thread	8-thread	64-thread
1280x720	2+1	513	593s	194s	125s
320x180	2+1	129	10s	3.2s	2.4s
320x180	64+1	129	313s	96s	58s

Table 2. Runtime measurements. The motion hypothesis count (FM #) column includes the no-motion hypothesis.

5.3. Qualitative Evaluation

Figures 2 and 3 illustrate the performance of our proposed algorithms vs. other state-of-the-art algorithms on the synthetic and also real imagery. These illustrative examples showcase the favorable properties of the discrete optimization approach. Overall, the fGT variant of our algorithm produces crisper edges with less smearing of the flow field than the competition. The f64 variant remains competitive, but struggles more perhaps due to the substantial perspective effect, which is not well represented by the uniformly sampled motion models. While no precise ground truth is available for the real imagery in Figure 3, we observe qualitatively similar performance vs the synthetic imagery.

5.4. Parameter Settings

In all our experiments we used DAISY descriptors with 5px radius, and each of the radial, angular, and histogram quantizations set to 4. On our training dataset we empirically found the constants $\lambda_1 = 0.6$, $\lambda_3 = 8$, and $\lambda_{mch} = 8$ to work well. We used $\delta_{min} = -256, \delta_{max} = +256$ disparity levels on the full resolution imagery experiments with $\tau_{mo} = 1, \tau_{occ} = 16$, and $\tau_{cc} = 1000$, and 8 directions for the MMSGM cost accumulation.

For experiments with the smaller images, we found that the parameters for $\lambda_{mch}, \delta_{min}, \delta_{max}, \tau_{occ}$ and τ_{cc} should be (down-)scaled to achieve best performance. We scale the selected constants by 0.25 for the synthetic quarter size (QR) imagery. Furthermore, because the full resolution images from real traffic cameras are smaller, we scale the constants by 0.5 while the constants used with the half resolution traffic images (HR) were scaled by 0.25.

6. Conclusion

We have presented and evaluated a novel algorithm which performs joint rigid multi-motion segmentation and disparity estimation to produce accurate optical flow observed in traffic scenes. Our evaluation both on a novel synthetic dataset with vehicular motion, as well as on real traffic camera imagery, suggests that the proposed approach using a small set of motion hypotheses achieves substantial accuracy gains over prior work in general optical flow.

Figure 2. Qualitative evaluation on our novel synthetic dataset. QR denotes quarter-resolution imagery: f64 was not practical to run on the full resolution images due to memory constraints.

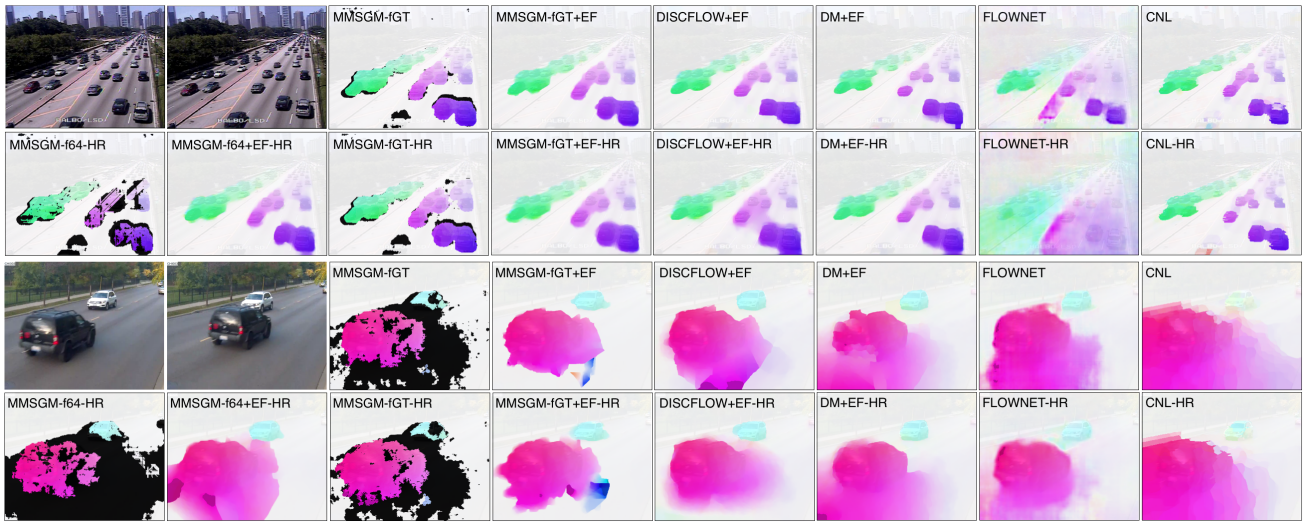

Figure 3. Qualitative evaluation on real-world traffic-camera imagery (top 648x500, bottom 720x576). HR denotes half-resolution.

References

[1] Blender. http://www.blender.org. 7

[2] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1985. 2

[3] P. Bhat, K. C. Zheng, N. Snavely, A. Agarwala, M. Agrawala, M. F. Cohen, and B. Curless. Piecewise image registration in the presence of multiple large motions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 4, 5

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 2012. 7

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Haziba, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 5

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 1, 2

[7] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 5

[8] M. Hornacek, F. Besse, J. Kautz, A. W. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using patchmatch belief propagation. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[9] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003. 1, 2

[10] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[11] M. Menze, C. Heipke, and A. Geiger. Discrete optimization for optical flow. In *German Conference on Pattern Recognition (GCPR)*, 2015. 2, 5, 7

[12] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision (IJCV)*, 2008. 2

[13] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 5, 6

[14] Y. Sheikh, A. Hakeem, and M. Shah. On the direct estimation of the fundamental matrix. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[15] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 5

[16] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010. 4

[17] L. Valgaerts, A. Bruhn, and J. Weickert. *A Variational Model for the Joint Recovery of the Fundamental Matrix and the Optical Flow*. 2008. 2

[18] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality tv-l1 flow with fundamental matrix prior. In *IEEE International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2008. 2

[19] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2