Slow Flow: Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data

Joel Janai[1] Fatma Güney[1] Jonas Wulff[2] Michael Black[2] Andreas Geiger[1,3]
[1]Autonomous Vision Group, MPI for Intelligent Systems Tübingen
[2]Perceiving Systems Department, MPI for Intelligent Systems Tübingen
[3]Computer Vision and Geometry Group, ETH Zürich
{joel.janai,fatma.guney,jonas.wulff,michael.black,andreas.geiger}@tue.mpg.de

Abstract

Existing optical flow datasets are limited in size and variability due to the difficulty of capturing dense ground truth. In this paper, we tackle this problem by tracking pixels through densely sampled space-time volumes recorded with a high-speed video camera. Our model exploits the linearity of small motions and reasons about occlusions from multiple frames. Using our technique, we are able to establish accurate reference flow fields outside the laboratory in natural environments. Besides, we show how our predictions can be used to augment the input images with realistic motion blur. We demonstrate the quality of the produced flow fields on synthetic and real-world datasets. Finally, we collect a novel challenging optical flow dataset by applying our technique on data from a high-speed camera and analyze the performance of the state-of-the-art in optical flow under various levels of motion blur.

Figure 1: **Illustration.** This figure shows reference flow fields with large displacements established by our approach. Saturated regions (white) are excluded in our evaluation.

1. Introduction

Much of the recent progress in computer vision has been driven by high-capacity models trained on very large annotated datasets. Examples for such datasets include ImageNet [50] for image classification [26,32], MS COCO [36] for object localization [45] or Cityscapes [14] for semantic segmentation [22]. Unfortunately, annotating large datasets at the pixel-level is very costly [70] and some tasks like optical flow or 3D reconstruction do not even admit the collection of manual annotations. As a consequence, less training data is available for these problems, preventing progress in learning-based methods. Synthetic datasets [12, 19, 25, 48] provide an attractive alternative to real images but require detailed 3D models and sometimes face legal issues [47]. Besides, it remains an open question whether the realism and variety attained by rendered scenes is sufficient to match the performance of models trained on real data.

This paper is concerned with the optical flow task. As there exists no sensor that directly captures optical flow ground truth, the number of labeled images provided by existing real world datasets like Middlebury [3] or KITTI [21, 39] is limited. Thus, current end-to-end learning approaches [16, 38, 44, 61] train on simplistic synthetic imagery like the flying chairs dataset [16] or rendered scenes of limited complexity [38]. This might be one of the reasons why those techniques do not yet reach the performance of classical hand designed models. We believe that having access to a *large* and *realistic* database will be key for progress in learning high-capacity flow models.

Motivated by these observations, we exploit the power of high-speed video cameras for creating accurate optical flow reference data in a variety of natural scenes, see Fig. 1. In particular, we record videos at high spatial (QuadHD:

(a) Input Image (b) High Frame Rate (c) Low Frame Rate

Figure 2: **Motion Blur.** Using high frame rate videos and our technique (described in Section 4.2) we are able to add realistic motion blur (b) to the images (a). In contrast, using low frame rates with a classical optical flow method results in severe staircasing artifacts (c).

2560×1440 Pixels) and temporal (> 200 fps) resolutions and propose a novel approach to dense pixel tracking over a large number of high-resolution input frames with the goal of predicting accurate correspondences at regular spatial and temporal resolutions. High spatial resolution provides fine textural details while high temporal resolution ensures small displacements allowing to integrate strong temporal constraints. Unlike Middlebury [3], our approach does not assume special lighting conditions or hidden texture. Compared to KITTI [21, 39], our method is applicable to non-rigid dynamic scenes, does not require a laser scanner and provides dense estimates. In addition, our approach allows for realistically altering the input images, e.g., by synthesizing motion blur as illustrated in Fig. 2.

To quantify the quality of our reference flow fields, we evaluate our method on a high frame rate version of the MPI Sintel dataset [12] and several 3D reconstructions of static scenes. Next, we process a novel high frame rate video dataset using our technique and analyze the performance of existing optical flow algorithms on this dataset. We demonstrate the usefulness of high frame rate flow estimates by systematically investigating the impact of motion magnitude and motion blur on existing optical flow techniques. We provide our code and dataset on our project web page[1].

2. Related Work

Datasets: After decades of assessing the performance of optical flow algorithms mostly qualitatively [41] or on synthetic data [5], Baker et al. proposed the influential Middlebury optical flow evaluation [3], for which correspondences have been established by recording images of objects with fluorescent texture under UV light illumination. Like us, they use images with high spatial resolution to compute dense sub-pixel accurate flow at lower resolution. They did not, however, use high temporal resolution. While their work addressed some of the limitations of synthetic data, it applies to laboratory settings where illumination conditions and camera motion can be controlled.

More recently, Geiger et al. published the KITTI dataset [21] which includes 400 images of static scenes with semi-dense optical flow ground truth obtained via a laser scanner. In an extension [39], 3D CAD models have been fitted in a semi-automatic fashion to rigidly moving objects. While this approach scales better than [3], significant manual interaction is required for removing outliers from the 3D point cloud and fitting 3D CAD models to dynamic objects. Additionally, the approach is restricted to rigidly moving objects for which 3D models exist.

In contrast to Middlebury [3] and KITTI [21], we strive for a fully scalable solution which handles videos captured under generic conditions using a single flexible hand-held high-speed camera. Our goal is to create reference optical flow data for these videos without any human in the loop.

Butler et al. [12] leveraged the naturalistic open source movie "Sintel" for rendering 1600 images of virtual scenes in combination with accurate ground truth. While our goal is to capture optical flow reference data in real world conditions, we render a high frame rate version of the MPI Sintel dataset to assess the quality of the reference flow fields produced by our method.

Remark: We distinguish between ground truth and reference data. While the former is considered free of errors[2], the latter is estimated from data and thus prone to inaccuracies. We argue that such data is still highly useful if the accuracy of the reference data exceeds the accuracy of state-of-the-art techniques by a considerable margin.

Methods: Traditionally, optical flow has been formulated as a variational optimization problem [15, 28, 43, 49, 57] with the goal of establishing correspondences between two frames of a video sequence. To cope with large displacements, sparse feature correspondences [9,11,62,67] and discrete inference techniques [4,13,34,37,40,55,71] have been proposed. Sand et al. [52] combine optical flow between frames with long range tracking but do so only sparsely and do not use high-temporal resolution video. More recently, deep neural networks have been trained end-to-end for this task [16, 38, 61]. However, these solutions do not yet attain the performance of hand-engineered models [1, 13, 24, 53].

One reason that hinders further progress in this area is the lack of large realistic datasets with reference optical flow. In this paper, we propose a data-driven approach which exploits the massive amount of data recorded with a high-speed camera by establishing dense pixel trajectories over multiple frames. In the following, we discuss the most related works on multi-frame optical flow estimation, ignor-

[1] http://www.cvlibs.net/projects/slow_flow

[2] Note that this is not strictly true as KITTI suffers from calibration errors and MPI Sintel provides motion fields instead of optical flow fields.

ing approaches that consider purely rigid scenes [7, 29].

Early approaches have investigated spatio-temporal filters for optical flow [17, 20, 27]. A very simple formulation of temporal coherence is used in [42, 56, 66, 72] where the magnitude of flow gradients is penalized. As the change of location is not taken into account, these methods only work for very small motions and a small number of frames. [51, 58, 64, 65] incorporate constant velocity priors directly into the variational optical flow estimation process. A constant acceleration model has been used in [6, 30] and layered approaches have been proposed in [59, 60]. Lucas-Kanade based sparse feature tracking has been considered in [35]. Epipolar-plane image analysis [7] provides another approach when imagery is dense in time.

Unfortunately, none of the methods mentioned above is directly applicable to our scenario, which requires dense pixel tracking through large space-time volumes. While most of the proposed motion models only hold for small time intervals or linear motions, several methods do not incorporate temporal or spatial smoothness constraints which is a necessity even in the presence of large amounts of data. Besides, computational and memory requirements prevent scaling to dozens of high-resolution frames.

In this paper, we therefore propose a two-stage approach: We first estimate temporally local flow fields and occlusion maps using a novel discrete-continuous multi-frame variational model, exploiting linearity within small temporal windows[3]. Second, we reason about the whole space-time volume based on these predictions.

3. Slow Flow

Let $\mathcal{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_N\}$ denote a video clip with N image frames $\mathbf{I}_t \in \mathbb{R}^{w \times h \times c}$ of size $w \times h$, captured at high frame rate. Here, c denotes the number of input channels (e.g., color intensities and gradients). In our experiments, we use a combination of brightness intensity [28] and gradients [10] for all color channels as features. This results in $c = 9$ feature channels for each image $\mathbf{I}_t$ in total.

Our goal is to estimate the optical flow $\mathbf{F}_{1\to N}$ from frame 1 to N, exploiting all intermediate frames. As the large number of high-resolution images makes direct optimization of the full space time volume hard, we split the task into two parts. In Section 3.1, we first show how small-displacement flow fields $\{\mathbf{F}_{t\to t+1}\}$ can be estimated reliably from multiple frames while accounting for occlusions. These motion estimates (which we call "Flowlets") form the input to our dense tracking model which estimates the full flow field $\mathbf{F}_{1\to N}$ as described in Section 3.2.

[3]We expect that most objects move approximately with constant velocity over short time intervals due to the physical effects of mass and inertia.

3.1. Multi-Frame Flowlets

Let $\{\mathbf{J}_{-T}, \dots, \mathbf{J}_0, \dots, \mathbf{J}_T\}$ with $\mathbf{J}_t = \mathbf{I}_{s+t}$ denote a short window of images from the video clip (e.g., $T = 2$), centered at reference image $\mathbf{J}_0 = \mathbf{I}_s$. For each pixel $\mathbf{p} \in \Omega = \{1, \dots, w\} \times \{1, \dots, h\}$ in the reference image $\mathbf{J}_0$ we are interested in estimating a flow vector $\mathbf{F}(\mathbf{p}) \in \mathbb{R}^2$ that describes the displacement of $\mathbf{p}$ from frame $t = 0$ to $t = 1$ as well as an occlusion map $\mathbf{O}(\mathbf{p}) \in \{0, 1\}$ where $\mathbf{O}(\mathbf{p}) = 1$ indicates that pixel $\mathbf{p}$ is forward occluded (i.e., occluded at $t > 0$, see Fig. 3). Due to our high input frame rate we expect roughly linear motions over short time windows. We thus enforce constant velocity as a powerful *hard constraint*. In contrast to a constant velocity soft constraint, this keeps the number of parameters in our model tractable and allows for efficient processing of multiple high-resolution input frames.

We now describe our energy formulation. We seek a minimizer to the following energy functional:

$$E(\mathbf{F}, \mathbf{O}) = \int_\Omega \psi^{\mathcal{D}}(\mathbf{F}(\mathbf{p}), \mathbf{O}(\mathbf{p})) + \psi^{\mathcal{S}}(\mathbf{F}(\mathbf{p})) + \psi^{\mathcal{O}}(\mathbf{O}(\mathbf{p}))d\mathbf{p} \tag{1}$$

Here, $\psi^{\mathcal{D}}$ is the data term and $\psi^{\mathcal{S}}, \psi^{\mathcal{O}}$ are regularizers that encourage smooth flow fields and occlusion maps.

The data term $\psi^{\mathcal{D}}$ measures photoconsistency in the forward direction if pixel $\mathbf{p}$ is backward occluded ($\mathbf{O}(\mathbf{p}) = 0$) and photoconsistency in backward direction otherwise[4], see Fig. 3a for an illustration. In contrast to a "temporally symmetric" formulation this allows for better occlusion handling due to the reduction of blurring artefacts at motion discontinuities as illustrated in Fig. 3b.

Thus, we define the data term as

$$\psi^{\mathcal{D}}(\mathbf{F}(\mathbf{p}), \mathbf{O}(\mathbf{p})) = \begin{cases} \psi^{\mathcal{F}}(\mathbf{F}(\mathbf{p})) - \tau & \text{if } \mathbf{O}(\mathbf{p}) = 0 \\ \psi^{\mathcal{B}}(\mathbf{F}(\mathbf{p})) & \text{otherwise} \end{cases} \tag{2}$$

where the bias term τ favors forward predictions in case neither forward nor backward occlusions occur. The forward and backward photoconsistency terms are defined as

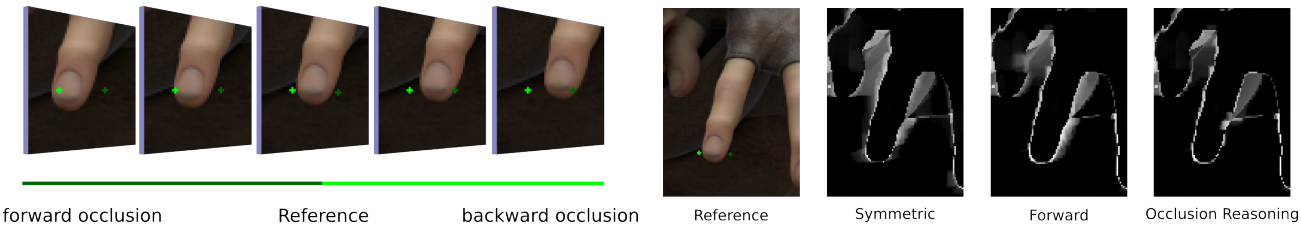
$$\psi^{\mathcal{F}}(\mathbf{F}(\mathbf{p})) = \sum_{t=0}^{T-1} \varphi_1^t(\mathbf{F}(\mathbf{p})) + \sum_{t=1}^{T} \varphi_2^t(\mathbf{F}(\mathbf{p})) \tag{3}$$

$$\psi^{\mathcal{B}}(\mathbf{F}(\mathbf{p})) = \sum_{t=-T}^{-1} \varphi_1^t(\mathbf{F}(\mathbf{p})) + \sum_{t=-T}^{-1} \varphi_2^t(\mathbf{F}(\mathbf{p})) \tag{4}$$

and measure photoconsistency between adjacent frames (φ_1^t) and wrt. the reference frame $\mathbf{J}_0$ (φ_2^t) to avoid drift [65]:

$$\varphi_1^t(\mathbf{F}(\mathbf{p})) = \rho(\mathbf{J}_t(\mathbf{p} + t\mathbf{F}(\mathbf{p})) - \mathbf{J}_{t+1}(\mathbf{p} + (t+1)\mathbf{F}(\mathbf{p})))$$
$$\varphi_2^t(\mathbf{F}(\mathbf{p})) = \rho(\mathbf{J}_t(\mathbf{p} + t\mathbf{F}(\mathbf{p})) - \mathbf{J}_0(\mathbf{p}))$$

[4]For small time windows, it can be assumed that either forward occlusion, backward occlusion or no occlusion occurs.



(a) Forward and backward occlusions (b) Results using different data terms

Figure 3: **Occlusion Reasoning.** (a) Illustration of a forward (dark green) and a backward (light green) occluded pixel. (b) Visualization of the end-point-error (EPE, larger errors in brighter colors) using a symmetric data term ($\psi^{\mathcal{D}} = \psi^{\mathcal{F}} + \psi^{\mathcal{B}}$), forward photoconsistency ($\psi^{\mathcal{D}} = \psi^{\mathcal{F}}$) and our full model ($\psi^{\mathcal{D}}$ as defined in Eq. 2). See text for details.

Here, $\rho(\cdot)$ denotes a robust ℓ_1 cost function which operates on the feature channels of $\mathbf{J}$. In our implementation, we extend the data term normalization proposed in [33, 46, 54] to the multi-frame scenario, which alleviates problems with strong image gradients.

In addition, we impose a spatial smoothness penalty on the flow ($\psi^{\mathcal{S}}$) and occlusion variables ($\psi^{\mathcal{O}}$):

$$\psi^{\mathcal{S}}(\mathbf{F}(\mathbf{p})) = \exp(-\kappa\|\nabla\mathbf{J}_0(\mathbf{p})\|_2) \cdot \rho(\nabla\mathbf{F}(\mathbf{p})) \quad (5)$$

$$\psi^{\mathcal{O}}(\mathbf{O}(\mathbf{p})) = \|\nabla\mathbf{O}(\mathbf{p})\|_2 \quad (6)$$

The weighting factor in Eq. 5 encourages flow discontinuities at image edge. We minimize Eq. 1 by interleaving variational optimization [10] of the continuous flow variables $\mathbf{F}$ with MAP inference [8] of the discrete variables $\mathbf{O}$. This optimization yields highly accurate flow fields for small displacements which form the input to our dense pixel tracking stage described in the following section.

3.2. Dense Tracking

Given the Flowlets $\{\mathbf{F}_{t\to t+1}\}$ from the previous section, our goal is to estimate the final optical flow field $\mathbf{F}_{1\to N}$ from frame 1 to frame N. In the following, we formulate the problem as a dense pixel tracking task.

Let $\mathcal{H} = \{\mathbf{H}_1, \ldots, \mathbf{H}_N\}$ denote the location of each (potentially occluded) pixel of reference image $\mathbf{I}_1$ in each frame of the full sequence. Here, $\mathbf{H}_t \in \mathbb{R}^{w\times h\times 2}$ describes a *location field*. $\mathbf{H}_1$ comprises the location of each pixel in the reference image. The optical flow from frame 1 to frame N is given by $\mathbf{F}_{1\to N} = \mathbf{H}_N - \mathbf{H}_1$.

Let further $\mathcal{V} = \{\mathbf{V}_1, \ldots, \mathbf{V}_N\}$ denote the visibility state of each pixel of reference image $\mathbf{I}_1$ in each frame of the sequence where $\mathbf{V}_t \in \{0,1\}^{w\times h}$ is a *visibility field* (1="visible", 0="occluded"). By definition, $\mathbf{V}_1 = \mathbf{1}^{w\times h}$.

To simplify notation, we abbreviate the trajectory of pixel $\mathbf{p} \in \Omega$ in reference image $\mathbf{I}_1$ from frame 1 to frame N with $\mathbf{h}_\mathbf{p} = \{\mathbf{H}_1(\mathbf{p}), \ldots, \mathbf{H}_N(\mathbf{p})\}$ where $\mathbf{H}_t(\mathbf{p}) \in \mathbb{R}^2$ is the location of reference pixel $\mathbf{p}$ in frame t. Similarly, we identify all visibility variables along a trajectory with $\mathbf{v}_\mathbf{p} = \{\mathbf{V}_1(\mathbf{p}), \ldots, \mathbf{V}_N(\mathbf{p})\}$ where $\mathbf{V}_t(\mathbf{p}) \in \{0,1\}$ indicates the visibility state of pixel $\mathbf{p}$ in frame t.

We are now ready to formulate our objective. Our goal is to jointly estimate dense pixel trajectories $\mathcal{H}_* = \mathcal{H}\backslash\mathbf{H}_1$ and the visibility label of each point in each frame $\mathcal{V}_* = \mathcal{V}\backslash\mathbf{V}_1$. We cast this task as an energy minimization problem

$$\begin{aligned} E(\mathcal{H}_*, \mathcal{V}_*) = {} & \lambda^{\mathcal{D}_A} \sum_{t<s} \underbrace{\psi_{ts}^{\mathcal{D}_A}(\mathbf{H}_t, \mathbf{V}_t, \mathbf{H}_s, \mathbf{V}_s)}_{\text{Appearance Data Term}} \\ & + \lambda^{\mathcal{D}_F} \sum_{s=t+1} \underbrace{\psi_{ts}^{\mathcal{D}_F}(\mathbf{H}_t, \mathbf{V}_t, \mathbf{H}_s, \mathbf{V}_s)}_{\text{Flow Data Term}} \\ & + \lambda^{\mathcal{F}_T} \sum_{\mathbf{p}\in\Omega} \underbrace{\psi_{\mathbf{p}}^{\mathcal{F}_T}(\mathbf{h}_\mathbf{p})}_{\text{Temporal Flow}} + \lambda^{\mathcal{F}_S} \sum_{\mathbf{p}\sim\mathbf{q}} \underbrace{\psi_{\mathbf{pq}}^{\mathcal{F}_S}(\mathbf{h}_\mathbf{p}, \mathbf{h}_\mathbf{q})}_{\text{Spatial Flow}} \\ & + \lambda^{\mathcal{V}_T} \sum_{\mathbf{p}\in\Omega} \underbrace{\psi_{\mathbf{p}}^{\mathcal{V}_T}(\mathbf{v}_\mathbf{p})}_{\text{Temporal Vis.}} + \lambda^{\mathcal{V}_S} \sum_{\mathbf{p}\sim\mathbf{q}} \underbrace{\psi_{\mathbf{pq}}^{\mathcal{V}_S}(\mathbf{v}_\mathbf{p}, \mathbf{v}_\mathbf{q})}_{\text{Spatial Vis.}} \end{aligned} \quad (7)$$

where $\psi_{ts}^{\mathcal{D}_A}$, $\psi_{ts}^{\mathcal{D}_F}$, $\psi_{\mathbf{p}}^{\mathcal{F}_T}$, $\psi_{\mathbf{pq}}^{\mathcal{F}_S}$, $\psi_{\mathbf{p}}^{\mathcal{V}_T}$, $\psi_{\mathbf{pq}}^{\mathcal{V}_S}$ are data, smoothness and occlusion constraints, and $\{\lambda\}$ are linear weighting factors. Here, $\mathbf{p} \sim \mathbf{q}$ denotes all neighboring pixels $\mathbf{p} \in \Omega$ and $\mathbf{q} \in \Omega$ on a 4-connected pixel grid.

The **appearance data term** $\psi_{ts}^{\mathcal{D}_A}$ robustly measures the photoconsistency between frame t and frame s at all visible pixels given the image evidence warped by the respective location fields $\mathbf{H}_t$ and $\mathbf{H}_s$:

$$\begin{aligned} & \psi_{ts}^{\mathcal{D}_A}(\mathbf{H}_t, \mathbf{V}_t, \mathbf{H}_s, \mathbf{V}_s) = \\ & \quad \sum_{\mathbf{p}\in\Omega} \mathbf{V}_t(\mathbf{p})\, \mathbf{V}_s(\mathbf{p})\, \|\mathbf{I}_t(\mathbf{H}_t(\mathbf{p})) - \mathbf{I}_s(\mathbf{H}_s(\mathbf{p}))\|_1 \end{aligned} \quad (8)$$

Here, $\mathbf{V}_t(\mathbf{p}) \in \{0,1\}$ indicates the visibility of pixel $\mathbf{p}$ in frame t. For extracting features at fractional locations $\mathbf{p}'_t$ we use bilinear interpolation.

Similarly, the **flow data term** $\psi_{ts}^{\mathcal{D}_F}$ measure the agreement between the predicted location field and the Flowlets:

$$\begin{aligned} & \psi_{ts}^{\mathcal{D}_F}(\mathbf{H}_t, \mathbf{V}_t, \mathbf{H}_s, \mathbf{V}_s) = \\ & \quad \sum_{\mathbf{p}\in\Omega} \mathbf{V}_t(\mathbf{p})\, \mathbf{V}_s(\mathbf{p})\, \|\mathbf{H}_s(\mathbf{p}) - \mathbf{H}_t(\mathbf{p}) - \mathbf{F}_{t\to s}(\mathbf{H}_t(\mathbf{p}))\|_1 \end{aligned} \quad (9)$$

While the appearance term reduces long-range drift, the flow term helps guide the model to a good basin. We thus obtained best results by a combination of the two terms.

The **temporal flow term** $\psi_{\mathbf{p}}^{\mathcal{F}_T}$ robustly penalizes deviations from the constant velocity assumption

$$\psi_{\mathbf{p}}^{\mathcal{F}_T}(\mathbf{h_p}) = \sum_{t=2}^{N-1} \|\mathbf{h}_{\mathbf{p}}^{t-1} - 2\,\mathbf{h}_{\mathbf{p}}^{t} + \mathbf{h}_{\mathbf{p}}^{t+1}\|_1 \qquad (10)$$

with $\mathbf{h}_{\mathbf{p}}^t$ the location of reference pixel $\mathbf{p}$ in frame t.

The **spatial flow term** $\psi_{\mathbf{pq}}^{\mathcal{F}_S}$ encourages similar trajectories at reference pixels $\mathbf{p}$ and $\mathbf{q}$

$$\psi_{\mathbf{pq}}^{\mathcal{F}_S}(\mathbf{h_p}, \mathbf{h_q}) = \xi(\mathbf{p}, \mathbf{q}) \sum_{t=1}^{N} \|(\mathbf{h}_{\mathbf{p}}^t - \mathbf{h}_{\mathbf{p}}^1) - (\mathbf{h}_{\mathbf{q}}^t - \mathbf{h}_{\mathbf{q}}^1)\|_2 \qquad (11)$$

with a weighting factor which encourages flow discontinuities at image edges $\xi(\mathbf{p}, \mathbf{q}) = \exp(-\kappa \|\nabla \mathbf{I}_1(\frac{\mathbf{p}+\mathbf{q}}{2})\|_2)$.

The **temporal visibility term** $\psi_{\mathbf{p}}^{\mathcal{V}_T}$ penalizes temporal changes of the visibility of a pixel $\mathbf{p}$ via a Potts model (first part) and encodes our belief that the majority of pixels in each frame should be visible (second part):

$$\psi_{\mathbf{p}}^{\mathcal{V}_T}(\mathbf{v_p}) = \sum_{t=1}^{N-1} [\mathbf{v}_{\mathbf{p}}^t \neq \mathbf{v}_{\mathbf{p}}^{t+1}] - \lambda^{\mathcal{V}} \sum_{t=1}^{N} \mathbf{v}_{\mathbf{p}}^t. \qquad (12)$$

Here, $\mathbf{v}_{\mathbf{p}}^t$ denotes if pixel pixel $\mathbf{p}$ in frame t is visible or not.

The **spatial visibility term** $\psi_{\mathbf{pq}}^{\mathcal{V}_S}$ encourages neighboring trajectories to take on similar visibility labels modulated by the contrast-sensitive smoothness weight ξ.

$$\psi_{\mathbf{pq}}^{\mathcal{V}_S}(\mathbf{v_p}, \mathbf{v_q}) = \xi(\mathbf{p}, \mathbf{q}) \sum_{t=1}^{N} [\mathbf{v}_{\mathbf{p}}^t \neq \mathbf{v}_{\mathbf{q}}^t] \qquad (13)$$

3.3. Optimization

Unfortunately, finding a minimizer of Eq. 7 is a very difficult problem that does not admit the application of black-box optimizers: First, the number of variables to be estimated is orders of magnitude larger than for classical problems in computer vision. For instance, a sequence of 100 QuadHD images results in more than 1 billion variables to be estimated. Second, our energy comprises discrete and continuous variables, which makes optimization hard. Finally, the optimization problem is highly non-convex due to the non-linear dependency on the input images. Thus, gradient descent techniques quickly get trapped in local minima when initialized with constant location fields.

In this section, we introduce several simplifications to make approximate inference in our model tractable. As the choice of these simplifications will crucially affect the quality of the retrieved solutions, we provide an in-depth discussion of each of these choices in the following.

Optimization: We optimize our discrete-continuous objective using max-product particle belief propagation, i.e., we iteratively discretize the continuous variables, sample the discrete variables, and perform tree-reweighted message passing [31] on the resulting discrete MRF. More specifically, we create a discrete set of trajectory and visibility hypotheses $\{(\mathbf{h}_{\mathbf{p}}^{(1)}, \mathbf{v}_{\mathbf{p}}^{(1)}), \ldots, (\mathbf{h}_{\mathbf{p}}^{(M)}, \mathbf{v}_{\mathbf{p}}^{(M)})\}$ for each pixel $\mathbf{p}$ (see next paragraph). Given this discrete set, the optimization of Eq. 7 is equivalent to the MAP solution of a simpler Markov random field with Gibbs energy

$$E(\mathbf{X}) = \sum_{\mathbf{p}} \psi_{\mathbf{p}}^{\mathcal{U}}(x_{\mathbf{p}}) + \sum_{\mathbf{p}\sim\mathbf{q}} \psi_{\mathbf{pq}}^{\mathcal{P}}(x_{\mathbf{p}}, x_{\mathbf{q}}) \qquad (14)$$

with $\mathbf{X} = \{x_{\mathbf{p}} | \mathbf{p} \in \Omega\}$ and $x_{\mathbf{p}} \in \{1, \ldots, M\}$. The unary $\psi_{\mathbf{p}}^{\mathcal{U}}$ and pairwise $\psi_{\mathbf{pq}}^{\mathcal{P}}$ potentials can be easily derived from Eq. 7. Technical details are provided in the supplementary.

Hypothesis Generation: A common strategy for max-product particle belief propagation [23, 63] is to start from a random initialization and to generate particles by iteratively resampling from a Gaussian distribution centered at the last MAP solution. This implements a stochastic gradient descent procedure without the need for computing gradients. Unfortunately, our objective is highly non-convex, and random or constant initialization will guide the optimizer to a bad local minimum close to the initialization.

We therefore opt for a data-driven hypothesis generation strategy. We first compute Flowlets between all subsequent frames of the input video sequence. Next, we accumulate them in temporal direction, forwards and backwards. For pixels visible throughout all frames, this already results in motion hypotheses of high quality. As not all pixels are visible during the entire sequence, we detect temporal occlusion boundaries using a forward-backward consistency check and track through partially occluded regions with spatial and temporal extrapolation. We use EpicFlow [46] to spatially extrapolate the consistent parts of each Flowlet which allows to propagate the flow from the visible into occluded regions. For temporal extrapolation, we predict point trajectories linearly from the last visible segment of each partially occluded trajectory. This strategy works well in cases where the camera and objects move smoothly (e.g., on Sintel or recordings using a tripod) while the temporal linearity assumption is often violated for hand-held recordings. However, spatial extrapolation is usually able to establish correct hypotheses in those cases.

After each run of tree-reweighted message passing, we re-sample the particles by sampling hypotheses from spatially neighboring pixels. This allows for propagation of high-quality motions into partial occlusions.

Assuming that the motion of occluders and occludees differs in most cases, we set the visibility of a hypothesis by comparing the local motion prediction with the corresponding Flowlet. If for a particular frame the predicted flow

differs significantly from the Flowlet estimate, the pixel is likely occluded. We leverage non-maximum suppression based on the criterion in Eq. 11 to encourage diversity amongst hypotheses.

Spatial Resolution: While a high (QuadHD) input resolution is important to capture fine details and attain subpixel precision, we decided to produce optical flow reference data at half resolution (1280×1024 Pixels) which is still significantly larger than all existing optical flow benchmarks [3, 12, 21]. While using the original resolution for the data term, we estimate $\mathcal{H}$ and $\mathcal{V}$ directly at the output resolution, yielding a 4 fold reduction in model parameters. Note that we do not lose precision in the optical flow field as we continue evaluating the data term at full resolution. To strengthen the data term, we assume that the flow in a small 3×3 pixel neighborhood of the original resolution is constant, yielding 9 observations for each point $\mathbf{p}$ in Eq. 8.

Temporal Resolution: While we observed that a high temporal resolution is important for initialization, the temporal smoothness constraints we employ operate more effectively at a coarser resolution as they are able to regularize over larger temporal windows. Additionally, we observed that it is not possible to choose one optimal frame rate due to the trade-off between local estimation accuracy and drift over time, which agrees with the findings in [35]. Therefore, we use two different frame rates for the hypotheses generation and choose the highest frame rate based on the robust upper 90% quantile of the optical flow magnitude computed at a smaller input resolutions with classical techniques [46]. This allows us to choose a fixed maximum displacement between frames. In practice, we chose the largest frame rate that yields maximal displacements of $\sim$2 pixels and the smallest frame rate that yields maximal displacements of $\sim$8 pixels which empirically gave the best results. Our dense pixel tracking algorithm operates on key frames based on the smallest frame rate. Flowlet observations of larger frame rates are integrated by accumulating the optical flow between key frames.

4. Evaluation & Analysis

In this section, we leverage our method to create reference flow fields for challenging real-world video sequences. We first validate our approach, by quantifying the error of the reference fields on synthetic and real data with ground truth (Section 4.1). Next, we create reference flow fields for a new high frame rate dataset (see Fig. 1) to systematically analyze state-of-the-art techniques wrt. their robustness to motion magnitude and motion blur (Section 4.2). All of our real-world sequences are captured with a Fastec TS5Q camera[5] which records QuadHD videos with up to 360 fps. Saturated regions which do not carry information are excluded from all our evaluations.

[5] http://www.fastecimaging.com/products/handheld-cameras/ts5

4.1. Validation of Slow Flow

We begin our evaluation by analyzing the quality of the reference flow fields produced using our method. As there exists no publicly available high frame rate dataset with optical flow ground truth, we created two novel datasets for this purpose. First, we re-rendered the synthetic data set MPI Sintel [12] using a frame rate of 1008 fps (a multiple of the default MPI Sintel frame rate) in Blender. While perfect ground truth flow fields can be obtained in this synthetic setting, the rendered images lack realism and textural details. We thus recorded a second data set of static real-world scenes using our Fastec TS5Q camera. In addition, we took a large number ($100 - 200$) of high resolution (24 Megapixel) images with a DSLR camera. Using state-of-the-art structure-from-motion [68] and multi-view stereo [18], we obtained high-quality 3D reconstructions of these scenes which we manually filtered for outliers. High-quality 2D-2D correspondences are obtained by projecting all non-occluded 3D points into the images. We provide more details and illustrations of this dataset in the supplementary document.

MPI Sintel: We selected a subset of 19 sequences from the MPI Sintel training set [12] and re-rendered them based on the "clean" pass of Sintel at 1008 frames per second, using a resolution of 2048×872 pixels. Table 1a shows our results on this dataset evaluated in all regions, only the visible regions, only the occluded regions or regions close to the respective motion boundares ("Edges"). For calibration, we compare our results to Epic Flow [46] at standard frame rate (24fps), a simple accumulation of EpicFlow flow fields at 144 fps (beyond 144 fps we observed accumulation drift on MPI Sintel), our multi-frame Flowlets (using a windows size of 5) accumulated at the same frame rate and at 1008 fps, as well as our full model.

Compared to computing optical flow at regular frame rates ("Epic Flow (24fps)"), the accumulation of flow fields computed at higher frame rates increases performance in non-occluded regions ("Epic Flow (Accu. 144fps)"). In contrast, occluded regions are not handled by the simple flow accumulation approach.

The proposed multi-frame flow integration ("Slow Flow (Accu. 144fps)") improves performance further. This is due to our multi-frame data term which reduces drift during the accumulation. While motion boundaries improve when accumulating multi-frame estimates at higher frame rates ("Slow Flow (Accu. 1008fps)"), the accumulation of flow errors causes drift resulting in an overall increase in error. This confirms the necessity to choose the frame rate adaptively depending on the expected motion magnitude as discussed in Section 3.3.

Methods	All (Edges)	Visible (E.)	Occluded (E.)
Epic Flow (24fps)	5.53 (16.23)	2.45 (10.10)	16.54 (20.68)
Epic Flow (Accu. 144fps)	4.73 (12.76)	1.04 (4.41)	17.09 (18.44)
Slow Flow (Accu. 144fps)	4.03 (12.03)	0.78 (4.43)	15.24 (17.28)
Slow Flow (Accu. 1008fps)	5.38 (11.78)	1.35 **(2.60)**	19.18 (17.93)
Slow Flow (Full Model)	**2.58 (10.06)**	**0.87** (4.65)	**9.45 (14.28)**

(a) EPE on MPI Sintel

	Jaccard Index
All Occluded	16.36%
EpicFlow F/B	62.86%
Our Method	**70.09%**

(b) Occlusion estimates on MPI Sintel

Flow Magnitude	100	200	300
Epic Flow	1.54	9.33	25.11
Slow Flow	**1.47**	**3.47**	**5.13**

(c) EPE on Real-World Scenes

Table 1: This figure shows the accuracy of our dense pixel tracking method and various baselines on MPI Sintel (a) and wrt. different motion magnitudes on real-world scenes (c) with ground truth provided by 3D reconstruction. In addition, we compare the occlusion estimates of two baselines and our method on MPI Sintel (b). See text for details.

Using our full model ("Slow Flow (Full Model)"), we obtain the overall best results, reducing errors wrt. EpicFlow at original frame rate by over 60% in visible regions and over 40% in occluded regions. Especially, in sequences with large and complex motions like "Ambush", "Cave", "Market" and "Temple" we observe a significant improvement. We improve in particular in the occluded regions and at motion boundaries due to the propagation of neighbouring hypotheses and our occlusion reasoning.

In Table 1b we compare the occlusion estimation of our method (last row) to a naïve estimate which sets all pixels in the image to occluded (first row) and two-frame EpicFlow in combination with a simple forward/backward check (second row). Our method outperforms both baselines considering the Jaccard Index and works best at large occluded regions. Several Sintel sequences (e.g. Bamboo) comprise very fine occlusions that are hard to recover. However, we found that failures in these cases have little impact on the performance of the flow estimates.

Note that the MPI Sintel dataset also contains many easy sequences (e.g., "Bamboo", "Mountain") where state-of-the-optical flow algorithms perform well due to the relatively small motion. Thus the overall improvement of our method is less pronounced compared to considering the challenging cases alone.

Real-world Sequences: To assess the performance margin we attain on more challenging data, we recorded a novel real-world data set comprising several static scenes. We used our Fastec TS5Q camera to obtain high frame rate videos and created sparse optical flow ground truth using structure-from-motion from high-resolution DSLR images with manual cleanup as described above.

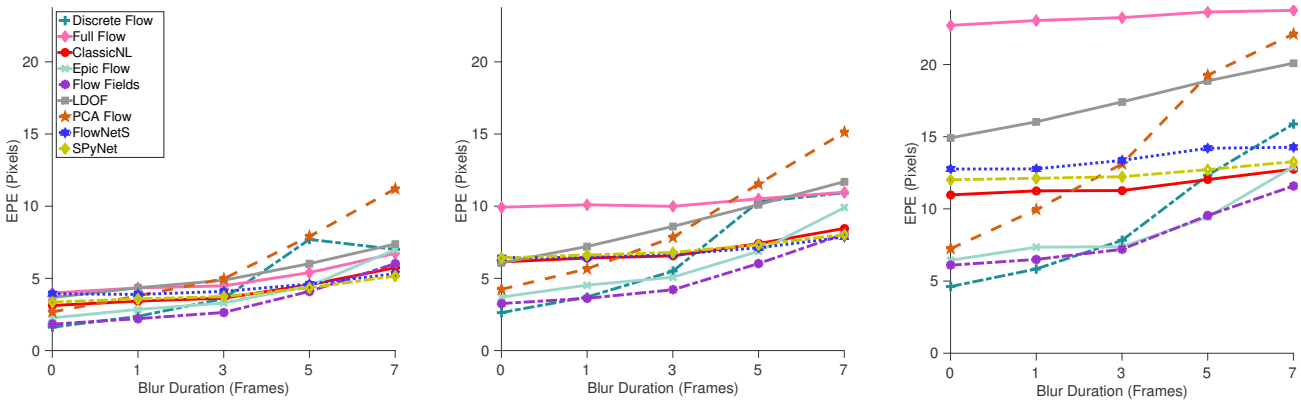
Table 1c shows our results. Again, we compare our approach to a EpicFlow [46] baseline at regular frame rate. While performance is nearly identical for small flow magnitudes of $\sim$ 100 Pixels, we obtain a five-fold decrease in error for larger displacements ($\sim$ 300 Pixels). This difference in performance increases even further if we add motion blur to the input images of the baseline as described in the following section. We conclude that our technique can be used to benchmark optical flow performance in the presence of large displacements where state-of-the-art methods fail.

4.2. Real-World Benchmark

In this section, we benchmark several state-of-the-art techniques on a challenging novel optical flow dataset. For this purpose, we have recorded 160 diverse real-world sequences of dynamic scenes using the Fastec TS5Q high speed camera, see Fig. 1 for an illustration. For each sequence, we have generated reference flow fields using the approach described in this paper. Based on this data, we compare 8 state-of-the-art optical flow techniques. More specifically, we evaluate DiscreteFlow [40], Full Flow [13], ClassicNL [57], EpicFlow [46], Flow Fields [2], LDOF [11], PCA Flow [69], FlowNet [16] and SPyNet [44] using the recommended parameter settings, but adapting the maximal displacement to the input. We are interested in benchmarking the performance of these methods wrt. two important factors: motion magnitude and motion blur, for which a systematic comparison on challenging real-world data is missing in the literature.

To vary the magnitude of the motion, we use different numbers of Flowlets in our optimization such that the 90% quantile of each sequence reaches a value of 100, 200 or 300 pixels. By grouping similar motion magnitudes, we are able to isolate the effect of motion magnitude on each algorithm from other influencing factors.

The second challenge we investigate is motion blur. Using our high frame rate Flowlets, we are able to add realistic motion blur onto the reference and target images. For different flow magnitudes which we wish to evaluate, we blend images over a certain blur length using the Flowlets at the highest frame rate in both forward and backward direc-



(a) 100px Flow Magnitude (b) 200px Flow Magnitude (c) 300px Flow Magnitude

Figure 4: State-of-the-art comparison on the generated reference data wrt. motion magnitude and blur.

tion. In particular, we blur each frame in the reference/target frame's neighborhood, by applying adaptive line shaped blur kernels depending on the estimated flow of the corresponding Flowlet. Tracing the corresponding pixels can be efficiently implemented using Bresenham's line algorithm. Finally, we average all blurred frames in a window around the reference/target frame for different window sizes corresponding to different shutter times. As illustrated in Fig. 2b, this results in realistic motion blur. For comparison, we also show the blur result when applying the adaptive blur kernel on the low frame rate inputs directly (Fig. 2c).

Fig. 4 shows our evaluation results in terms of average end-point-error (EPE) over all sequences. We use three different plots according to the magnitude of the motion ranging from 100 pixels (easy) to 300 pixels (hard). For each plot we vary the length of the blur on the x-axis. The blur length is specified with respect to the number of blurred frames at the highest temporal resolution, where 0 indicates the original unblurred images. Per sequence results are provided in the supplementary material.

As expected, for the simplest case (100 pixels without motion blur), most methods perform well, with DiscreteFlow [40] slightly outperforming the other baselines. Interestingly, increasing the blur length impacts the methods differently. While matching-based methods like PCA Flow [69], EpicFlow [46] and DiscreteFlow [40] suffer significantly, the performance of FlowNet [16], SPyNet [44] and ClassicNL [57] remains largely unaffected. A similar trend is visible for larger flow magnitudes, where the difference in performance becomes more clearly visible. As expected, the performance of all methods decreases with larger magnitudes. We further note that some methods (e.g., Full Flow [13]) which perform well on synthetic datasets such as MPI Sintel [12] produce large error on our dataset. This underlines the importance of optical flow datasets with real-world images as the one proposed in this paper.

5. Conclusion and Future Work

In this paper, we presented a dense tracking approach to generate reference data from high speed images for evaluating optical flow algorithms. The introduction of Flowlets allows to integrate strong temporal assumptions at higher frame rates and the proposed dense tracking method allows for establishing accurate reference data even at large displacements. Using this approach we created a real world dataset with novel challenges for evaluating the state-of-the-art in optical flow. Our experiments showed the validity of our approach by comparing it to a state-of-the-art two frame formulation on a high frame rate version of the MPI Sintel dataset and several real-world sequences. We conclude that the generated reference data is precise enough to be used for the comparison of methods.

In our comparison of state-of-the-art approaches, we observed that all methods except FlowNet, SPyNet and ClassicNL suffer from motion blur. The magnitude of the flow affects in particular learning based and variational approaches which cannot handle large displacements well compared to methods guided by matching or optimizing local feature correspondences.

In future work, we plan to further improve upon our method. In particular, complex occlusions and partial occlusions are the main source of errors remaining. Detecting these occlusions reliably is a difficult task even in the presence of high frame rates. In addition, we plan to derive a probabilistic version of our approach which allows for measuring confidences beyond simple flow consistency or color saturation measures which we have used in this paper. We also plan to extend our dataset in size to make it useful for training high-capacity networks and comparing their performance with networks trained on synthetic data.

Acknowledgements. Fatma Güney and Jonas Wulff were supported by the Max Planck ETH Center for Learning Systems.

References

[1] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 2

[2] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 7

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92:1–31, 2011. 1, 2, 6

[4] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving PatchMatch for large displacement optical flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[5] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. *International Journal of Computer Vision (IJCV)*, 12(1):43–77, 1994. 2

[6] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1991. 3

[7] R. C. Bolles and H. H. Baker. Epipolar-plane image analysis: A technique for analyzing motion sequences. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, 1987. 3

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23:2001, 1999. 4

[9] J. Braux-Zin, R. Dupont, and A. Bartoli. A general dense image matching framework combining direct and feature-based costs. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2013. 2

[10] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2004. 3, 4

[11] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 33:500–513, March 2011. 2, 7

[12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012. 1, 2, 6, 8

[13] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7, 8

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[15] O. Demetz, M. Stoll, S. Volz, J. Weickert, and A. Bruhn. Learning brightness transfer functions for the joint recovery of illumination changes and optical flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014. 2

[16] A. Dosovitskiy, P. Fischer, E. Ilg, P. Haeusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 1, 2, 7, 8

[17] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision (IJCV)*, 5(1):77–104, 1990. 3

[18] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1376, 2010. 6

[19] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[20] T. Gautama and M. M. V. Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *Neural Networks*, 13(5):1127–1136, 2002. 3

[21] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 6

[22] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 1

[23] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[24] F. Güney and A. Geiger. Deep discrete flow. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2016. 2

[25] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding real world indoor scenes with synthetic data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[27] D. J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision (IJCV)*, 1(4):279–302, 1988. 3

[28] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence (AI)*, 17(1-3):185–203, 1981. 2, 3

[29] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 1999. 3

[30] R. Kennedy and C. J. Taylor. Optical flow with geometric occlusion estimation and fusion of multiple frames. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2014. 3

[31] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568–1583, 2006. 5

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 1

[33] S. Lai and B. C. Vemuri. Reliable and efficient computation of optical flow. *International Journal of Computer Vision (IJCV)*, 29(2):87–105, 1998. 4

[34] V. S. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[35] S. Lim, J. G. Apostolopoulos, and A. E. Gamal. Optical flow estimation using temporally oversampled video. *IEEE Trans. on Image Processing (TIP)*, 14(8):1074–1087, 2005. 3, 6

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014. 1

[37] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):978–994, 2011. 2

[38] N. Mayer, E. Ilg, P. Haeusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 1, 2

[39] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[40] M. Menze, C. Heipke, and A. Geiger. Discrete optimization for optical flow. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2015. 2, 7, 8

[41] H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence (AI)*, 33(3):299–324, 1987. 2

[42] J. Ralli, J. Díaz, and E. Ros. Spatial and temporal constraints in variational correspondence methods. *Machine Vision and Applications (MVA)*, 24(2):275–287, 2013. 3

[43] R. Ranftl, K. Bredies, and T. Pock. Non-local total generalized variation for optical flow estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014. 2

[44] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. *arXiv.org*, 1611.00850, 2016. 1, 7, 8

[45] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 1

[46] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4, 5, 6, 7, 8

[47] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 1

[48] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[49] S. Roth and M. J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision (IJCV)*, 74(1):33–50, 2007. 2

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv.org*, 1409.0575, 2014. 1

[51] A. Salgado and J. Sánchez. Temporal constraints in large optical flow. In *Proc. of the International Conf. on Computer Aided Systems Theory (EUROCAST)*, 2007. 3

[52] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision (IJCV)*, 80(1):72, 2008. 2

[53] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[54] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1991. 4

[55] F. Steinbrücker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 1609–1614, 2009. 2

[56] M. Stoll, S. Volz, and A. Bruhn. Joint trilateral filtering for multiframe optical flow. In *Proc. IEEE International Conf. on Image Processing (ICIP)*, 2013. 3

[57] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2):115–137, 2014. 2, 7, 8

[58] D. Sun, E. B. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Advances in Neural Information Processing Systems (NIPS)*, 2010. 3

[59] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3

[60] D. Sun, J. Wulff, E. Sudderth, H. Pfister, and M. Black. A fully-connected layered model of foreground and background flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3

[61] D. Teney and M. Hebert. Learning to extract motion from videos in convolutional neural networks. *arXiv.org*, 1601.07532, 2016. 1, 2

[62] R. Timofte and L. V. Gool. Sparse flow: Sparse matching for small to large displacement optical flow. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. 2

[63] H. Trinh and D. McAllester. Unsupervised learning of stereo vision with monocular cues. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2009. 5

[64] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011. 3

[65] C. M. Wang, K. C. Fan, and C. T. Wang. Estimating optical flow by integrating multi-frame information. *Journal of Information Science and Engineering (JISE)*, 2008. 3

[66] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision (JMIV)*, 14(3):245–255, 2001. 3

[67] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2013. 2

[68] C. Wu. Towards linear-time incremental structure from motion. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2013. 6

[69] J. Wulff and M. J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7, 8

[70] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger. Semantic instance annotation of street scenes by 3d to 2d label transfer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[71] H. Yang, W. Lin, and J. Lu. DAISY filter flow: A generalized discrete approach to dense correspondences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[72] H. Zimmer, A. Bruhn, and J. Weickert. Optic flow in harmony. *International Journal of Computer Vision (IJCV)*, 93(3):368–388, 2011. 3