Directional Priors for Multi-Frame Optical Flow

Daniel Maurer
maurer@vis.uni-stuttgart.de
Michael Stoll
stoll@vis.uni-stuttgart.de
Andrés Bruhn
bruhn@vis.uni-stuttgart.de

Institute for Visualization and
Interactive Systems
University of Stuttgart, Germany

Abstract

Pipeline approaches that interpolate and refine an initial set of point correspondences have recently shown a good performance in the field of optical flow estimation. However, so far, these methods are typically restricted to two frames which makes exploiting temporal information difficult. In this paper, we show how such pipeline approaches can be extended to the temporal domain and how directional constraints can be incorporated to further improve the estimation. To this end, we not only suggest to exploit temporal information in the prefiltering step, we also propose a trajectorial refinement method that lifts successful concepts of recent variational two-frame methods to the multi-frame domain. Experiments demonstrate the usefulness of our pipeline approach. They do not only show good results in general, they also demonstrate the clear benefits of using multiple frames and of imposing directional constraints on the prefiltering step and the refinement.

1 Introduction

Driven by numerous applications such as video editing [31], object tracking [36], action recognition [38] or driver assistance [28], there has been an enormous progress in the field of optical flow estimation within the last years. One class of methods that has recently enjoyed large popularity are pipeline approaches; see e.g. [21, 24, 30]. Combining the large displacement capabilities of feature matching techniques with the regularization properties of variational methods such approaches achieve a remarkable accuracy in the meantime [21, 24].

Following the EpicFlow framework of Revaud *et al.* [30] such pipeline approaches proceed in four steps. In a first step, they calculate initial matches by a nearest neighbor search of feature descriptors. While this allows to handle large displacements, the resulting matches typically contain outliers due to occlusions and ambiguous data. These outliers are then removed in a second step using some kind of consistency check. Since the remaining matches after filtering are non-dense, the missing locations are then inpainted in a third step. Finally, the resulting flow field is refined using a continuous variational method. This not only yields sub-pixel accurate results but also introduces some form of spatial regularization.

As such pipeline approaches are very modular, there has been lots of progress regarding all four steps since the introduction of EpicFlow [30]. Instead of relying on DeepMatching [39], recent approaches are based on Coarse-to-fine PatchMatch [20] or dense results from discrete energy-based methods [27]. Moreover, the original foward-backward filtering has been complemented by additional concepts such as eliminating small regions [27] or using combinatorial regularization [12]. Furthermore, also the edge-preserving inpainting method has been replaced by a robust RANSAC-based variant relying on affine SLIC superpixels [1, 21]. And finally, methods are used for the refinement that can handle affine illumination changes and adapt the order of the regularization to the underlying type of motion [24].

However, despite of all this progress, existing pipeline approaches only use two input frames so far. This is surprising, since at least some works from the literature have been able to report improvements by using multiple frames [23, 33, 35, 37, 41]. Most promising are explicit motion models in terms of hard [40, 41] or soft constraints [2, 14, 37]. Evidently, such models could also be valuable for pipeline approaches, since they may help to identify erroneous matches in the initial filtering or to improve the quality in the final refinement. However, as observed in [37, 40] employing such models is not straightforward. In particular, results may deteriorate if the chosen model does not fit the actual motion. Hence, less restrictive contraints may be worthwhile – especially in the context of real-world data.

Contributions. We address the aforementioned problem by proposing a *three-frame pipeline approach* based on *directional constraints*. Thereby, we contribute in three ways: (i) We suggest to filter the initial set of point correspondences based on checking directional consistency. By preferring short time trajectories with small directional changes, we are able to discard erroneous matches while not constraining the velocity. (ii) Moreover, we propose a novel variational refinement method that lifts successful concepts of recent two-frame methods to the multi-frame domain. By combining a three-frame illumination compensated data term with a joint anisotropic order-adaptive regularizer, we obtain an accurate refinement method that is robust under illumination changes. (iii) Finally, inspired by our filtering step, we extend our three-frame model by a directional consistency term. Compared to approaches that employ an explicit motion model, this constraint is much less restrictive and hence more useful in typical real-world scenarios. Experiments on all major benchmarks give evidence that using multiple frames in pipeline approaches is indeed beneficial.

Related Work. Despite no multi-frame pipeline approach has been proposed in the literature so far, there are various works on how to integrate temporal information into the estimation. These approaches can be divided into four classes with decreasingly restrictive assumptions:

Hard Motion Constraints. Those constraints parametrize the estimation in such a way that an underlying motion model is enforced by construction. While Werlberger *et al.* [40], Janai *et al.* [22] and Kennedy *et al.* [23] use a constant motion model, Garg *et al.* [13] learn their model from training data. Moreover, Wulff *et al.* [41] consider a rigid motion model that relies on semantic scene information. Although hard constraints can be beneficial if they fit the actual motion, they typically generalize poorly if not integrated adaptively.

Soft Motion Constraints. Instead of strictly enforcing a motion model, one may only penalize deviations from it. Such soft constraints comprise coupling the estimation to a given model as done by Garg *et al.* [14] or modeling trajetorial smoothness as proposed by Salgado *et al.* [32] and Volz *et al.* [37]. Evidently, soft constraints are less restrictive than their hard analogues. Yet, complex model selection strategies may be required to avoid a loss of quality [37].

Directional Constraints. Instead of relying on an explicit motion model that imposes constraints on both the velocity and the direction, directional constraints do not restrict the actual velocity. For instance, Chaudhury *et al.* in [9] propose a method that makes use of curvature-based trajectorial regularization. While their approach is similar to ours in the sense that it prefers straight trajectories, the regularizer is rather complex. Moreover, the remaining model is directly based on Horn and Schunck [19] and hence not competitive at all.

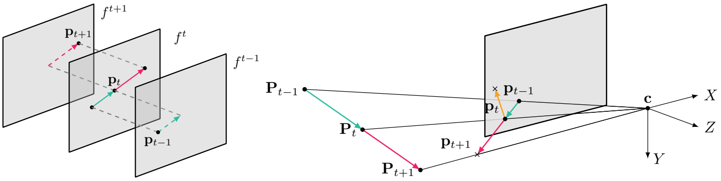

Figure 1: **Left:** Forward and backward correspondences w.r.t. a reference frame f^t. The locations $\mathbf{p}^{t-1}$, $\mathbf{p}^t$ and $\mathbf{p}^{t+1}$ denote the projections of a single 3D point at time step $t-1$, t and $t+1$. **Right:** The 3D point $\mathbf{P}$ is visible during the time step $t-1$ and t, but moves outside the scene within time step $t+1$. The cross (orange vector) depicts a possible false match.

Structural Constraints. Finally, structural constraints only restrict the shape of objects over time. For instance, Volz *et al.* [37] propose a joint spatial regularization that couples the discontinuities of all flow fields along the trajectory. A similar contraint is also imposed by Sun *et al.* [35] in the context of layered motion. While structural constraints are beneficial in most scenarios, they typically give the smallest benefits, since they carry the least information.

2 Trajectorial Outlier Filtering

Our approach follows the EpicFlow pipeline [30] as described in the introduction. It consists of four subsequent steps: matching, filtering, inpainting and refinement. Since matching approaches applied in the first step of the pipeline often do not contain explicit regularization, computing matches can be quite error prone, especially in low textured or occluded regions. Therefore, it is important to remove erroneous matches by performing some kind of outlier filtering. A common strategy in this context is the bidirectional consistency check. It computes matches between both frames in forward and backward direction and verifies that both directions are consistent. In our work, we seek to leverage additional temporal information to improve the estimation. Hence, we introduce an additional trajectorial[1] outlier filtering step. This step is based on the observation, that fast moving objects typically do not exhibit an abrupt change in direction. While this assumption holds for a broad variety of scenarios, a few rare cases may lead to violations (e.g. circular moving objects).

Directional Consistency. To check if the directional consistency is fulfilled, we additionally need the backward correspondences w.r.t. the previous frame, see Fig. 1 (left, green arrow). The corresponding backward and forward correspondences then form a short term trajectory. We neglect those trajectories for which the angle between both motion vectors exceeds a certain threshold. A scenario, where this turns out to be very beneficial is the case where objects move outside the scene; see Fig. 1 (right). In this case, it is impossible to find the correct forward pointing outwards the image (green vector) and hence one will typically obtain some false match pointing inwards the image (e.g. the orange vector).

Magnitude Check. It is important to note that it only makes sense to evaluate the directional consistency criterion in case of fast moving objects, i.e. large motion vectors. In our case, we define large motion vectors as vectors with a magnitude greater than three pixels. Choosing the threshold not too small is important because many feature matching approaches only operate with pixel precision and are hence likely introduce large directional errors in case of slowly moving objects (while the actual Euclidean error is still small).

[1] Please note that in case of optical flow the term trajectory refers to trajectories of points on the image plane.

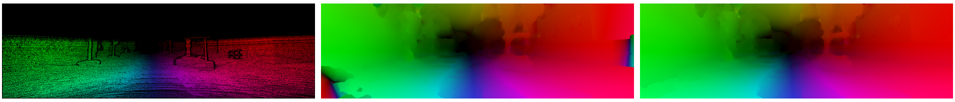

Figure 2: KITTI 2012 sequence [15] showing the benefit of the trajectorial outlier filtering. **From left to right:** Ground truth, inpainted matches without and with filtering.

3 Multi-Frame Variational Refinement

After discussing the trajectorial outlier filtering, we now turn towards the refinement of the inpainted matches. Here, likewise, we aim at exploiting temporal information. Moreover, we lift successful concepts of recent two-frame approaches to the multi-frame domain.

Let us consider three consecutive frames f^{t-1}, f^t and f^{t+1} of an image sequence $f^k(\mathbf{x}) : \Omega \to \mathbb{R}^2$ where $k \in [0,...,T]$ is the time and $\mathbf{x} = (x,y)^\top$ denotes the location within an rectangular image domain $\Omega \subset \mathbb{R}^2$. Moreover, let $\mathbf{w} = (\mathbf{w}^t, \mathbf{w}^{t-1})^\top$ describe the two optical fields $\mathbf{w}^t = (u^t, v^t)^\top : \Omega \to \mathbb{R}^2$ and $\mathbf{w}^{t-1} = \left(u^{t-1}, v^{t-1}\right)^\top : \Omega \to \mathbb{R}^2$ between frames f^t and f^{t+1} (forward) and frames f^t and f^{t-1} (backward), respectively. Furthermore, let $\mathbf{c} = (\mathbf{c}^t, \mathbf{c}^{t-1})^\top$ denote two sets of coefficient fields $\mathbf{c}^t = (c_1^t, \ldots, c_n^t)^\top : \Omega \to \mathbb{R}^n$ and $\mathbf{c}^{t-1} = \left(c_1^{t-1}, \ldots, c_n^{t-1}\right)^\top : \Omega \to \mathbb{R}^n$ that describe the local illumination changes related to the corresponding flows. Finally, let $o : \Omega \to [0,1]$ be a spatially varying weighting function that locally selects between first- and second-order regularization. Then our novel variational refinement model reads

$$E_{\text{MF}}(\mathbf{w}, \mathbf{c}, o) = \int_\Omega D(\mathbf{w}, \mathbf{c}) + \beta \cdot R_{\text{illum}}(\mathbf{c}) + \alpha \cdot R_{\text{oar}}(\mathbf{w}, o) + \gamma \cdot S_{\text{oar}}(o) + \lambda \cdot R_{\text{traj}}(\mathbf{w})\, d\mathbf{x}. \quad (1)$$

It is composed of an illumination compensated three-frame data term D, a spatial illumination regularizer R_{illum}, an order-adaptive spatial flow regularizer R_{oar}, a selection term S_{oar}, and a directional flow regularizer R_{traj}.

Data Term. Rather than relying purely on invariant image features w.r.t. illumination, we pursue the idea of change estimation. This allows our approach to keep up with more powerful descriptors that are typically used during the preceding matching step. Hence, we extend the concept of explicitly modeling illumination changes on a per-pixel level via locally parametrized brightness transfer functions $\Phi(f^t, \mathbf{c}^t)$ from [11] to a three-frame setting. To this end, we resort to a general trajectorial parametrization, i.e. we parametrize both flow fields $\mathbf{w}^t$, $\mathbf{w}^{t-1}$ as well as the coefficient fields $\mathbf{c}^t$, $\mathbf{c}^{t-1}$ w.r.t. the reference frame f^t. As a result, all unknowns share the same coordinate system. This not only simplifies the modeling of the trajectorial regularization but also leads to a temporal alignment of flow discontinuities [37]. Combining illumination compensation with brightness and gradient constancy assumption [4] balanced by a global weight μ, we obtain the following three-frame data term:

$$\begin{aligned} D(\mathbf{w}, \mathbf{c}) = \quad & \Psi_{\text{c}}\left(\left(f^{t+1}(\mathbf{x}+\mathbf{w}^t) - \Phi(f^t, \mathbf{c}^t)\right)^2\right) + \mu \cdot \Psi_{\text{c}}\left(\left|\nabla\left(f^{t+1}(\mathbf{x}+\mathbf{w}^t) - \nabla\Phi(f^t, \mathbf{c}^t)\right)\right|^2\right) \\ & + \Psi_{\text{c}}\left(\left(f^{t-1}(\mathbf{x}-\mathbf{w}^{t-1}) - \Phi(f^t, \mathbf{c}^{t-1})\right)^2\right) + \mu \cdot \Psi_{\text{c}}\left(\left|\nabla\left(f^{t-1}(\mathbf{x}-\mathbf{w}^{t-1}) - \nabla\Phi(f^t, \mathbf{c}^{t-1})\right)\right|^2\right). \end{aligned}$$

Here, each constraint is robustified separately [5] using the Charbonnier function $\Psi_{\text{c}}(s^2) = 2\varepsilon^2\sqrt{1+s^2/\varepsilon^2}$ [8] with $\varepsilon = 0.01$. Regarding the illumination compensation applied to the reference frame f^t via the the brightness transfer function Φ, we intenionally use a different set of coefficients, i.e. $\mathbf{c}^t$ and $\mathbf{c}^{t-1}$, for the forward and the backward flow. This allows good results even in case the illumination varies differently between all three frames. Also, in contrast to [11] where the employed basis functions $\bar{\phi}$ and ϕ_i of the brightness transfer function are learned, however, we resort to a more general solution. More precisely, we use the normalized affine function

$$\Phi(f,\mathbf{c}) = \bar{\phi}(f) + \sum_{k=1}^{2} c_k \cdot \phi_k(f) \quad \text{with} \quad \bar{\phi}(f) = f, \;\; \phi_1(f) = \frac{f}{n_1}, \;\; \phi_2(f) = \frac{1}{n_2}, \tag{2}$$

where n_1 and n_2 are normalization factors such that $\|\phi_k(f^t)\|_2 = 1$. Being able to represent multiplicative and additive illumination changes, this choice of the brightness transfer function typically constitutes a good trade-off between complexity and quality [26].

Spatial Illumination Regularizer. Due to the explicit modeling of illumination changes on a per-pixel level, regularization must be applied to the coefficient fields $\mathbf{c}^t$ and $\mathbf{c}^{t-1}$. To this end, we consider an anisotropic first-order regularizer that adapts to the underlying image structure by incorporating local directional information [11]:

$$R_{\text{illum}}(\mathbf{c}) = \sum_{i=0}^{1} \sum_{m=1}^{2} \Psi_m \left(\sum_{k=1}^{2} \left(\mathbf{r}_m^\top \nabla c_k^{t-i} \right)^2 \right). \tag{3}$$

Here, the directions $\mathbf{r}_1$, $\mathbf{r}_2$ are obtained as eigenvectors of the regularization tensor – a generalization of the structure tensor based on the constancy assumptions of the data term [43]. Moreover, to allow edge-enhancement in the dominant direction and edge-preservation in the orthogonal direction, we follow [11] and choose $\Psi_1(s^2) = \varepsilon^2 \log\left(1 + s^2/\varepsilon^2\right)$ to be the Perona-Malik penalizer [29] and $\Psi_2(s^2) = 2\varepsilon^2\sqrt{1 + s^2/\varepsilon^2}$ to be the Charbonnier penalizer [8], each with $\varepsilon = 0.01$. Please note that we penalize both sets $\mathbf{c}^{t-1}$ and $\mathbf{c}^t$ separately, since illumination changes between the three frames are not necessarily correlated.

Spatial Flow Regularizer. In case of the spatial flow regularizer we build upon an order selection strategy that has been recently proposed for two-frame optical flow methods [25]. Instead of relying on a smoothness term with fixed regularization order, it locally switches between a first-order smoothness term that allows to model piecewise constant flow fields (e.g. fronto-parallel-motion) and a second-order smoothness term that models piecewise affine flow fields (e.g. the ego-motion of the camera). In practice, this is realized by jointly estimating the optical flow and a continous spatially varying selection function $o(\mathbf{x}) \in [0,1]$ that locally determines the order of the regularization. The corresponding regularizer reads

$$R_{\text{oar}}(\mathbf{w}, o) = \inf_{\mathbf{a},\mathbf{b}} \left\{ \bar{o} \cdot S_1(\mathbf{w}) + (1 - \bar{o}) \cdot \left(S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + T \right) + \delta \cdot S_3(\mathbf{a}, \mathbf{b}) \right\}, \tag{4}$$

which is a combination of a first-order regularizer S_1 and an indirect second-order regularizer, that is composed of a coupling term S_2 and a smoothness term S_3 weighted by a parameter δ. The parameter $T = 10^{-5}$ denotes an activation cost for second-order smoothness that avoids overfitting to small fluctuations. Moreover, the selection process is rendered more robust by averaging the order weighting function o within a small rectangular neighborhood $\mathcal{N}(\mathbf{x})$ via

$$\bar{o}(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} o(\mathbf{y}) \, d\mathbf{y}, \tag{5}$$

where $|\mathcal{N}(\mathbf{x})|$ is the size of the neighborhood. While the basic order selection strategy can be readily applied in the three-frame case, we have to replace the corresponding spatial regularizers in [24] by suitable three-frame counterparts.

As first-order regularizer we use the same anisotropic model as for the illumination coefficients. However, in contrast to Eq. (3) where we apply a separate penalization of the coefficient fields, we now employ a joint penalization of all flows. This can be seen as a structural constraint, since the parametrization w.r.t. the reference frame leads to an alignment of flow discontinuities along trajectories [43]. The corresponding regularizer reads

$$S_1(\mathbf{w}) = \sum_{m=1}^{2} \Psi_m \left(\sum_{i=0}^{1} \left(\mathbf{r}_m^\top \nabla u^{t-i} \right)^2 + \left(\mathbf{r}_m^\top \nabla v^{t-i} \right)^2 \right) , \tag{6}$$

where $\Psi_1(s^2)$ and $\Psi_2(s^2)$ are again chosen as the Perona-Malik and the Charbonnier penalizer, respectively, with $\varepsilon = 0.01$. Also the directions $\mathbf{r}_1$ and $\mathbf{r}_2$ are the same as before.

As second-order regularizer we employ a three-frame variant of the anisotropic coupling model from [17]. Again, we use a joint penalization for the flow fields which imposes a structural constraint. The resulting coupling term that couples the flow gradients to the auxiliary functions $\mathbf{a} = (\mathbf{a}^t, \mathbf{a}^{t-1})^\top$ and $\mathbf{b} = (\mathbf{b}^t, \mathbf{b}^{t-1})^\top$ reads

$$S_2(\mathbf{w},\mathbf{a},\mathbf{b}) = \sum_{m=1}^{2} \Psi_m \Big(\sum_{i=0}^{1} \left(\mathbf{r}_m^\top \left(\nabla u^{t-i} - \mathbf{a}^{t-i} \right) \right)^2 + \left(\mathbf{r}_m^\top \left(\nabla v^{t-i} - \mathbf{b}^{t-i} \right) \right)^2 \Big) , \tag{7}$$

while the corresponding smoothness term on the auxiliary variables is given by

$$S_3(\mathbf{a},\mathbf{b}) = \sum_{m=1}^{2} \Psi_m \big(\sum_{l=1}^{2} \sum_{i=0}^{1} \left(\mathbf{r}_l^\top \mathcal{J} \mathbf{a}^{t-i} \mathbf{r}_m \right)^2 + \left(\mathbf{r}_l^\top \mathcal{J} \mathbf{b}^{t-i} \mathbf{r}_m \right)^2 \big) . \tag{8}$$

The latter one enforces smoothness on the auxiliary functions via penalizing their Jacobians $\mathcal{J}\mathbf{a}^*$ and $\mathcal{J}\mathbf{b}^*$. In this context, a weight δ determines the amount of smoothness; cf. Eq. (4). Again, the directions $\mathbf{r}_1$, $\mathbf{r}_2$ and the penalizer functions Ψ_1, Ψ_2 are defined as before.

Selection Term. The selection term S_{oar}, associated with our novel three-frame order-adaptive regularizer reads

$$S_{\text{oar}}(o) = (1-o) \cdot \ln(1-o) + o \cdot \ln(o) . \tag{9}$$

As shown in [25] such a choice leads to an order-adaptive selection behavior based on the energy difference of S_1 and S_2 within the neighborhood region $\mathcal{N}$. Please note that we use the same selection function o for all flow fields. This implies the temporal assumptions that the order of the spatial regularization does not change along the trajectory.

Directional Regularizer. Beyond the structural constraints that our model already imposes, there is the possibility to also impose soft motion constraints like *e.g.* the ones in [37] that approximate first- and second-order derivatives in trajectorial direction. Apart from the fact that the underlying motion models are typically not very realistic in practice, they also unnecessarily constrain the velocity of moving objects. Hence, in our approach, we will refrain from using such soft motion constraints and we resort to less restrictive directional constraints. To this end, we propose a novel regularizer that is in accordance with our assumptions from the filtering step. As before, we assume that fast moving objects do not exhibit abrupt changes

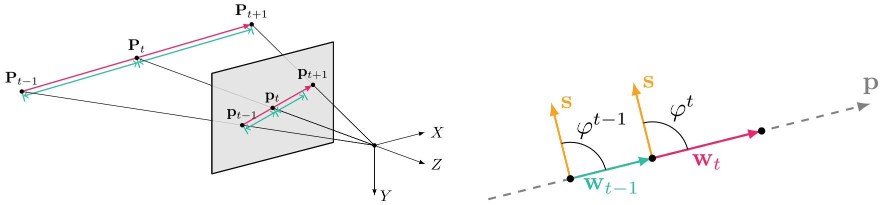

Figure 3: **Left:** Illustration showing the projection of a 3D point moving along a straight line with constant velocity. The projection also lies on a straight line, but the projected spacing varies. **Right:** Illustration showing the forward and backward flow as well as the vector $\mathbf{s}$ that is orthogonal to the direction prior $\mathbf{p}$.

in direction, i.e. they approximately move on a straight line in 3D space; see Fig. 3. Since this does not impose any assumptions on the actual velocity, the corresponding constraint is much more likely to be fulfilled in real-world scenarios than traditional soft constraints.

In general, given a spatially varying prior direction $\mathbf{p}$ and its associated orthogonal direction $\mathbf{s} = \mathbf{p}^\perp$ as shown in Fig. 3 (right), the corresponding directional regularizer reads

$$R_{\text{traj}}(\mathbf{w}) = \Psi_c\left(\left(\cos(\varphi^{t-1})\right)^2 + \left(\cos(\varphi^t)\right)^2\right) = \Psi_c\left(\left(\mathbf{s}^\top \frac{\mathbf{w}^{t-1}}{|\mathbf{w}^{t-1}|}\right)^2 + \left(\mathbf{s}^\top \frac{\mathbf{w}^{t}}{|\mathbf{w}^{t}|}\right)^2\right), \quad (10)$$

where the robust Charbonnier penalizer Ψ_c with $\varepsilon = 0.01$ renders the constraint more robust against violations of the underlying assumption. In our particular case, we choose the prior direction $\mathbf{p}$ by averaging the direction of the forward and backward flow, i.e. $\mathbf{w}^t$ and $\mathbf{w}^{t-1}$.

4 Minimization

After introducing our novel variational refinement model let us provide a brief summary of the corresponding minimization strategy. Like similar refinement approaches [24, 30] we resort to the well-known warping strategy [4]. This strategy is based on solving a discretized version of the associated Euler-Lagrange equations by means of two nested fixed point iterations. Traditionally, the outer fixed point iteration that introduces and incremental computation of the unknowns is embedded into a coarse-to-fine scheme with downsampling factor η, in order to avoid local minima during an estimation from scratch [4, 11, 25, 39]. Due to the good initialization with the inpainted initial matches, however, in case of pipeline approaches, such a hierarchical scheme can either be omitted [30] or used with a reduced pyramid size [24]. We opt for the second strategy, since using a reduced coarse-to-fine scheme allows to benefit from good initial matches while still offering some sort of error correction capability. We further apply constraint normalization to each data constraint [43]. The inner fixed point iteration, in contrast, allows to compute the resulting non-linear system of equations at each coarse-to-fine level as a series of linear systems. To this end, we keep the non-linear expressions related to the outer derivatives of the sub-quadratic penalizer functions of the terms D, R_{illum}, R_{oar}, R_{traj} as well as the prior direction $\mathbf{p}$ of R_{traj} fixed and solve the resulting linear system for the increments of the forward and backward flow $\mathbf{w}$, the illumination coefficients $\mathbf{c}$ and the auxiliary variables $\mathbf{a}, \mathbf{b}$ by means of a cascadic multicolor variant of the successive over-relaxation (SOR) method. In case of the weighting function o the update can be computed explicitly as shown in [25].

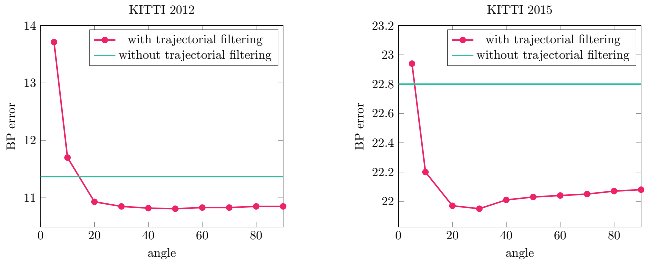

Figure 4: Impact of trajectorial filtering. Plot showing the bad pixel (BP) error of the inpainted flow field (without variational refinement) using different angle thresholds for the trajectorial outlier filtering on the KITTI training datasets [15, 28].

5 Evaluation

For all our experiments we computed two sets of matches, using Coarse-to-fine PatchMatch (CPM) [20] and DiscreteFlow (DF) [27], respectively, and performed the respective inpainting by means of edge-preserving interpolation [30]. To this end, we used the available code with the default parameters provided by the respective authors. In case of our refinement model we set $\mu = 5$, $\gamma = 10^{-5}$, $\eta = 0.9$ fixed and learned the remaining weighting parameters α, β, δ and λ (Tab. 1) from the training data by minimizing the bad pixel error (BP). The learning is performed using the framework of [34] based on the the downhill simplex method. To evaluate the performance of our method, we used the KITTI 2012 [15] benchmark and the KITTI 2015 [28] benchmark. The runtime of the entire method is 52s (CPM) and 104s (DF) for an image sequence of size 1242×375 (KITTI 2015) using two cores on an Intel® Core™ i7-7820X CPU running at 3.60GHz. Hereof the variational refinement takes 44s.

Table 1: Learned parameter settings.

	CPM Matches [20]				DF Matches [27]			
KITTI 2012	$\alpha = 12.44$	$\beta = 0.07$	$\delta = 10.44$	$\lambda = 0.18$	$\alpha = 13.44$	$\beta = 0.06$	$\delta = 10.70$	$\lambda = 0.20$
KITTI 2015	$\alpha = 15.56$	$\beta = 0.04$	$\delta = 4.75$	$\lambda = 2.25$	$\alpha = 15.85$	$\beta = 0.08$	$\delta = 6.32$	$\lambda = 2.83$

Trajectorial Outlier Filtering. In our first experiment we investigate the impact of our novel trajectorial outlier filtering. Therefore, we evaluated error measures on the inpainted CPM matches and compared the outcome with and without trajectorial filtering. As one can see in Fig. 4, the BP error significantly decreases for both KITTI benchmarks. This holds as long as the threshold on the angle is not chosen too small. This, however, is not surprising, since small thresholds discard too many matches by allowing only completely straight trajectories.

Multi-frame Variational Refinement. In our second experiment we analyze the benefit of our novel three-frame refinement model. To this end, we evaluated different configurations of our model on the training datasets of both KITTI benchmarks. To investigate the impact of the different multi-frame concepts, we thereby considered (i) a two-frame setting using EpicFlow refinement [30], (ii) a two-frame setting using OIR refinement [24], (iii) a three-frame setting with trajectorial filtering, (iv) a three-frame setting with trajectorial filtering and three-frame refinement, and (v) our full model; see Tab. 2. Please note that the two-frame setting with OIR refinement can be seen as a two-frame variant of our approach. By comparing the results of the two-frame methods one already notices the benefit of a more

Table 2: Results for the KITTI 2012 [15] and KITTI 2015 [28] training sequences in terms of the average enpoint error (AEE) and bad pixel (BP) error with a threshold of 3px.

Method					**CPM Matches [20]**				**DF Matches [27]**				runtime*
					KITTI 2012		**KITTI 2015**		**KITTI 2012**		**KITTI 2015**		**KITTI 2015**
	#	$\lhd$	R	$\rightsquigarrow$	AEE	BP	AEE	BP	AEE	BP	AEE	BP	seconds
EpicFlow refinement [30]	2	–	–	–	3.00	14.58	7.78	22.86	2.97	14.89	6.68	23.14	~0.4
OIR refinement [24]	2	–	–	–	2.78	9.68	7.36	19.21	2.34	9.29	5.89	18.10	~30
our method	3	✓	–	–	2.24	9.20	6.36	18.30	2.09	9.07	5.76	17.42	~30
our method	3	✓	✓	–	2.19	9.02	6.33	18.25	**2.07**	9.02	**5.72**	17.37	~41
our method	3	✓	✓	✓	2.19	**8.98**	6.31	18.14	**2.07**	8.99	**5.72**	**17.23**	~44

#: number of frames, $\lhd$: trajectorial filtering, R: three-frame refinement, $\rightsquigarrow$: directional regularizer

*: runtime of the variational refinement step measured on KITTI 2015 (excludes preceding steps)

Table 3: Results of the KITTI [15, 28] test sets. Top 10 non-anonymous methods and methods related to our approach, excluding stereo and scene flow approaches.

KITTI 2012	**Out-Noc**	**Out-All**	**Avg-Noc**	**Avg-All**	**KITTI 2015**	**Fl-bg**	**Fl-fg**	**Fl-all**
SDF[1]	3.80 %	7.69 %	1.0 px	2.3 px	PWC-Net	9.66 %	9.31 %	9.60 %
PWC-Net	4.22 %	8.27 %	0.9 px	1.7 px	MirrorFlow	8.93 %	17.07 %	10.29 %
UnFlow	4.28 %	8.42 %	0.9 px	1.7 px	SDF[1]	8.61 %	23.01 %	11.01 %
MirrorFlow	4.38 %	8.20 %	1.2 px	2.6 px	UnFlow	10.15 %	15.93 %	11.11 %
ImpPB+SPCI	4.65 %	13.47 %	1.1 px	2.9 px	CNNF+PMBP	10.08 %	18.56 %	11.49 %
our method (CPM)	**4.69 %**	**9.63 %**	**1.0 px**	**2.4 px**	MR-Flow[1]	10.13 %	22.51 %	12.19 %
CNNF+PMBP	4.70 %	14.87%	1.1 px	3.3 px	DCFlow	13.10 %	23.70 %	14.86 %
FlowFieldCNN	4.89 %	13.01 %	1.2 px	3.0 px	SOF[1]	14.63 %	22.83 %	15.99 %
IntrpNt-df	4.94 %	14.13 %	1.0 px	2.4 px	**our method (DF)**	**14.93** %	**23.37** %	**16.33** %
RicFlow	4.96 %	13.04 %	1.3 px	3.2 px	DF+OIR	15.11 %	23.45 %	16.50 %
our method (DF)	**4.97 %**	**10.02 %**	**1.1 px**	**2.6 px**	**our method (CPM)**	**15.57** %	**23.84** %	**16.95** %
DF+OIR	5.17 %	10.43 %	1.1 px	2.9 px	RicFlow	18.73 %	19.09 %	18.79 %
CPM-Flow	5.79 %	13.70 %	1.3 px	3.2 px	DiscreteFlow	21.53 %	21.76 %	21.57 %
DiscreteFlow	6.23 %	16.63 %	1.3 px	3.6 px	CPM-Flow	22.32 %	22.81 %	22.40 %
EpicFlow	7.88 %	17.08 %	1.5 px	3.8 px	EpicFlow	25.81 %	28.69 %	26.29 %

[1] exploits semantic information

sophisticated refinement approach. However, in terms of quality both methods are already clearly outperformed by our three-frame method that relies on trajectorial filtering. Going a step further to our approach with additional three-frame refinement, one can observe that even without trajectorial regularization further improvements can be achieved due to the structural constraint given by the joint penalization of the flow discontinuities. Finally, it becomes evident that the best results are obtained with our complete model that – apart from the structural constraint – also imposes a directional constraint. An exemplary result of our full model for the KITTI 2015 benchmark is given in Fig. 5. Please note that, in contrast to our approach, applying a soft or hard motion constraint based on a widely used constant motion model did not yield any improvements, since it is actually not valid for most of the scenes.

While the quality improves using our three-frame method, the computational complexity becomes larger – in particular the one of the refinement step. Here, the runtime increases from 30 seconds up to 44 seconds as shown in Tab. 2. Please note, that the large runtime of the refinement step compared to the EpicFlow refinement is due to the illumination-aware data term that allows to deal with affine illumination changes, the order-adaptive regularizer that can deal with fronto-parallel and affine motion as well as the (reduced) coarse-to-fine scheme that allows for an error correction. In contrast, the EpicFlow refinement uses a much simpler model, works only on a single resolution level and only performs a few iterations, since the underlying first-order regularizer is not able to handle affine motion.

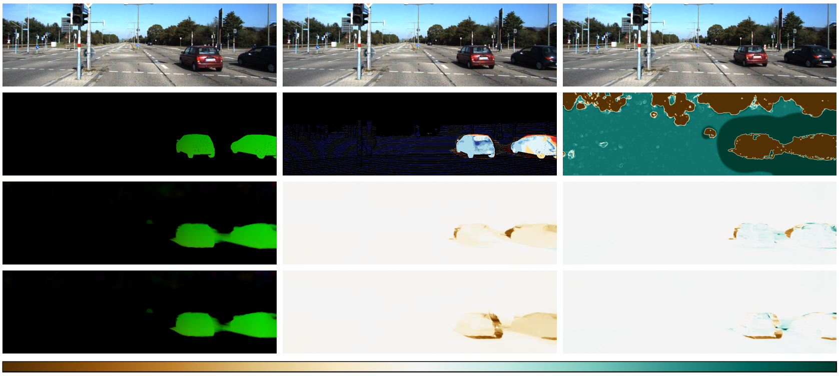

Figure 5: Exemplary result for a sequence of the KITTI 2015 [28] benchmark. **First row:** Input frames f^{t-1}, f^t and f^{t+1}. **Second row:** Ground truth forward flow, bad pixel visualization and computed order selection map o (first-order: turquoise, second-order: brown). **Third row:** Computed forward flow $\mathbf{w}^t$, coefficients c_1^t and c_2^t (shifted and rescaled, zero maps to white). **Fourth row:** Computed backward flow $\mathbf{w}^{t-1}$, coefficients c_1^{t-1} and c_2^{t-1}.

Comparison to the Literature. In our final experiment we compare our method to the literature. To this end, we used the evaluation data sets of both KITTI benchmarks. The corresponding results can be found in Tab. 3. Apart from the top 10 non-anonymous approaches – excluding methods that make use of additional stereo constraints – we listed results of recent two-frame pipeline approaches such as EpicFlow [30], CPM-Flow [20], RicFlow [21], and DF+OIR [24]. Evidently, with Rank 6/11 and Rank 9/11 on the KITTI 2012 and the KITTI 2015 benchmark, respectively, our three-frame pipeline approach achieves highly competitive results. In particular, it outperforms all other related two-frame pipeline approaches.

6 Conclusion and Outlook

In this paper, we addressed the problem of extending pipeline approaches for computing the optical flow to the multi-frame domain. Thereby we focused on two of the four stages: the prefiltering of the initial matches and the variational refinement of the final flow. For both tasks, we proposed directional constraints that are less restrictive than explicit motion models and hence turned out to be quite useful in practice. While applying such constraints for prefiltering enabled us to remove erroneous matches right at the beginning of the pipeline, embedding them into a specifically tailored three-frame refinement method allowed us to improve the accuracy at the very end. Experiments demonstrated the validity of our approach. They not only showed competitive results of our three-frame approach on the KITTI benchmarks, they also demonstrated an increase in accuracy by using directional constraints.

While our method may serve as a starting point to develop further multiframe pipeline approaches, it also shows that less strict constraint may be also an option for other tasks in computer vision. In the future, we plan to extend our approach to more frames and higher order directional constraints. This would allow us to model curved motion trajectories – still without constraining the velocity. Furthermore, we plan to improve the runtime by using more advanced optimization schemes [3, 6, 7, 10, 18] and parallel hardware [16, 42].

Acknowledgments. We thank the German Research Foundation (DFG) for financial support within project B04 of SFB/Transregio 161.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[2] M. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–302, 1991.

[3] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

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

[5] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy and real-time performance. In *Proc. IEEE International Conference on Computer Vision*, pages 749–755, 2005.

[6] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, 70(3):257–277, 2006.

[7] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1): 120–145, 2011.

[8] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.

[9] K. Chaudhury and R. Mehrotra. A trajectory-based computational model for optical flow estimation. *IEEE Transactions on Robotics and Automation*, 11(5):733–741, 1995.

[10] J. Chen, Z. Cai, J. Lai, and X. Xie. Fast optical flow estimation based on the split bregman method. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(3):664–678, 2018.

[11] 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. European Conference on Computer Vision*, pages 455–471, 2014.

[12] B. Drayer and T. Brox. Combinatorial regularization of descriptor matching for optical flow estimation. In *Proc. British Machine Vision Conference*, pages 1–13, 2015.

[13] R. Garg, L. Pizarro, D. Rueckert, and L. Agapito. Dense multi-frame optic flow for non-rigid objects using subspace constraints. In *Proc. Asian Conference on Computer Vision*, pages 460–473, 2010.

[14] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 104(3):286–314, 2013.

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[16] P. Gwosdek, H. Zimmer, S. Grewenig, A. Bruhn, and J. Weickert. A highly efficient GPU implementation for variational optic flow based on the euler-lagrange framework. In *Proc. ECCV Workshop on Computer Vision with GPUs*, pages 372–382, 2012.

[17] D. Hafner, C. Schroers, and J. Weickert. Introducing maximal anisotropy into second order coupling models. In *Proc. German Conference on Pattern Recognition*, pages 79–90, 2015.

[18] D. Hafner, P. Ochs, J. Weickert, M. Reißel, and S. Grewenig. FSI schemes: Fast semi-iterative solvers for PDEs and optimisation methods. In *Proc. German Conference on Pattern Recognition*, 2016.

[19] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[20] Y. Hu, R. Song, and Y. Li. Efficient Coarse-to-fine PatchMatch for large displacement optical flow. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5704–5712, 2016.

[21] Y. Hu, Y. Li, and R. Song. Robust interpolation of correspondences for large displacement optical flow. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 481–489, 2017.

[22] J. Janai, F. Güney, J. Wulff, M. Black, and A. Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3597–3607, 2017.

[23] R. Kennedy and C. Taylor. Optical flow with geometric occlusion estimation and fusion of multiple frames. In *Proc. International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 364–477, 2015.

[24] D. Maurer, M. Stoll, and A. Bruhn. Order-adaptive and illumination-aware variational optical flow refinement. In *Proc. British Machine Vision Conference*, pages 1–13, 2017.

[25] D. Maurer, M. Stoll, and A. Bruhn. Order-adaptive regularisation for variational optical flow: Global, local and in between. In *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*, pages 550–562, 2017.

[26] D. Maurer, M. Stoll, S. Volz, P. Gairing, and A. Bruhn. A comparison of isotropic and anisotropic second order regularisers for optical flow. In *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*, pages 537–549, 2017.

[27] B. Menze and A. Geiger. Discrete optimization for optical flow. In *Proc. German Conference on Pattern Recognition*, pages 16–28, 2015.

[28] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

[29] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:629–639, 1990.

[30] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[31] R. Sadek, G. Facciolo, P. Arias, and V. Caselles. A variational model for gradient-based video editing. *International Journal of Computer Vision*, 103(1):127–162, 2013.

[32] Agustín Salgado and Javier Sánchez. Temporal constraints in large optical flow estimation. *Computer Aided Systems Theory – EUROCAST 2007*, pages 709–716, 2007.

[33] M. Stoll, S. Volz, and A. Bruhn. Joint trilateral filtering for multiframe optical flow. In *Proc. International Conference on Image Processing*, pages 3845–3849, 2013.

[34] M. Stoll, S. Volz, D. Maurer, and A. Bruhn. A time-efficient optimisation framework for parameters of optical flow methods. In *Proc. Scandinavian Conference on Image Analysis*, pages 41–53, 2017.

[35] D. Sun, E. Sudderth, and M. Black. Layered segmentation and optical flow estimation over time. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1768–1775, 2012.

[36] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *Proc. European Conference on Computer Vision*, pages 438–451, 2010.

[37] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. In *Proc. International Conference on Computer Vision*, pages 1116–1123, 2011.

[38] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. International Conference on Computer Vision*, pages 1–8, 2013.

[39] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: large displacement optical flow with deep matching. In *Proc. IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[40] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anistropic Huber-L^1 optical flow. In *Proc. British Machine Vision Conference*, pages 1–11, 2009.

[41] J. Wulff, L. Sevilla-Lara, and M. J. Black. Optical flow in mostly rigid scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6911–6920, 2017.

[42] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L^1 optical flow. In *Proc. German Conference on Pattern Recognition*, pages 214–223, 2007.

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