To appear in the Proc. of the *Springer ECCV, Lecture Notes in, Computer Science 2014*.

View-Consistent 3D Scene Flow Estimation over Multiple Frames

Christoph Vogel [1], Stefan Roth [2] and Konrad Schindler [1]

[1] Photogrammetry and Remote Sensing, ETH Zurich, Switzerland
[2] Department of Computer Science, TU Darmstadt, Germany

Abstract. We propose a method to recover dense 3D scene flow from stereo video. The method estimates the depth and 3D motion field of a dynamic scene from *multiple consecutive frames* in a sliding temporal window, such that the estimate is *consistent across both viewpoints of all frames* within the window. The observed scene is modeled as a collection of planar patches that are consistent across views, each undergoing a rigid motion that is approximately constant over time. Finding the patches and their motions is cast as minimization of an energy function over the continuous plane and motion parameters and the discrete pixel-to-plane assignment. We show that such a view-consistent multi-frame scheme greatly improves scene flow computation in the presence of occlusions, and increases its robustness against adverse imaging conditions, such as specularities. Our method currently achieves leading performance on the KITTI benchmark, for both flow and stereo.

1 Introduction

The 3D scene flow is a dense description of surface geometry and 3D motion in a dynamic scene. Scene flow estimation analyzes images from two (or more) cameras taken at two (or more) time steps, and delivers depth and 3D motion densely for every pixel. Hence, it can be seen as a generalization of optical flow to 3D, or alternatively as stereo for dynamic scenes. Like these two classical problems, scene flow estimation is ill-posed due to the 3D equivalent of the aperture problem, and requires some form of regularization. Dense 3D shape and motion are useful for a variety of tasks, including motion capture [26], 3D video generation for 3D-TV [12] and driver assistance (*e.g.*, the Daimler *6D-vision* project [16, 19, 33]).

To this date most scene flow methods in the literature, *e.g.* [1, 30, 33], base their reconstruction on two consecutive stereo pairs, and declare one of the four images as a *reference view*, for which the shape and motion vectors are computed. The starting point for this work are two rather straightforward observations: *(i)* the two frames typically originate from a longer stereo video sequence, hence it seems wasteful not to exploit longer time intervals; and *(ii)* there is no conceptual reason for a privileged reference view, since imaging problems (occlusions, lack of contrast, *etc.*) affect all images equally. In the present paper we address these two points. Specifically, we propose to *simultaneously estimate depth and 3D*

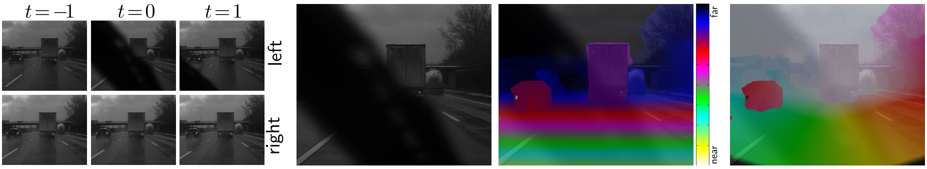

Fig. 1. Consistency over multiple frames makes scene flow estimation robust against severe disturbances like the windscreen wiper. *(left)* Input frames. *(middle)* The left view at time $t = 0$. *(right)* Our scene flow estimate for that viewpoint (shown, from left to right, as disparity and reprojected 2D flow field).

motion over longer time intervals, in such a way that the results are consistent across all views within that interval (see Fig. 1).

It seems evident that, at reasonable frame rates, physically plausible scenes exhibit temporal consistency over more than just two frames. We conjecture that long-term constraints may actually be more helpful for scene flow than for 2D optical flow, where the majority of today's top-performing methods only uses two frames. A scene flow reconstruction resides in 3D space rather than in its 2D projection, hence constraints caused by physical object properties like inertia remain valid in the long term, and can be exploited more directly.

The key motivation for, moreover, estimating the scene flow in all views and demanding consistency (rather than estimating it only in a single reference view) is to overcome viewpoint-dependent adversities like specularities and occlusions, where the image data is not consistent (see Fig. 1). Under difficult imaging conditions (large motions, specular reflections, occlusions, shadows) considering all views equally in our experience greatly improves robustness against outliers, and additionally allows for more reliable and accurate occlusion reasoning.

We propose to integrate both consistency across time and across views into a *single energy function*, such that one can jointly solve for a reconstruction by taking into account all evidence (rather than reconstructing independently for different frames or reference views and merging the results in post-processing). Having said that, we restrict the estimation to short temporal windows of up to 4 frames to limit the computational cost of our integrated solution. Moreover, going to longer and longer time intervals yields diminishing returns, and in many scenarios (*e.g.*, autonomous driving) immediate feedback is required, such that a time-lag of more than one or two frames is not acceptable.

Our approach leverages the scene flow representation of [28], *i.e.* the scene is modeled as a collection of planar and rigidly moving patches. This parameterization is more constrained than others in the literature, *e.g.* [1, 25, 33], but has been shown to be valid for road scenarios and other typical scenes of interest. It is well suited for our view-consistent multi-frame approach, since it drastically reduces the number of unknowns per frame, and inherently provides an (over-)segmentation into patches with simple geometry and motion, which can be expected to remain stable over time. In order to go beyond two time steps we additionally assume that the 3D motion (translation and rotation) of each segment is nearly constant within the examined time interval. Empirically this assumption is valid for the segment sizes and time intervals considered.

This paper makes the following contributions: *(i)* We propose a novel 3D scene flow model that does not rely on an arbitrary reference view, but rather reconstructs 3D shape and motion w.r.t. every image in a time interval, while enforcing consistency of the reconstruction across views; *(ii)* we extend dense scene flow estimation to more than two time steps, with a temporally consistent piecewise-planar segmentation of the scene and a prior that favors constant 3D velocity over time; and *(iii)* we formulate a consistent energy that includes both these aspects, along with a corresponding inference scheme, and can – at least conceptually – handle any number of viewpoints and any number of time steps.

We evaluate our method on the challenging KITTI dataset of real street scenes, using the stereo and flow benchmarks. Compared to two-frame scene flow computation with a fixed reference view [28], the proposed view-consistent estimation over four frames reduces the average endpoint error from 2.5 to 1.4 pixels, and improves the KITTI error metric by 45% for flow, respectively 36% for stereo. In the evaluation on full images, including occlusion areas, our method currently achieves the best results on the benchmark, for both optical flow and stereo. We further show on some particularly hard examples that our model is remarkably robust against missing evidence, outliers, and occlusions.

2 Related work

Scene flow estimation is usually traced back to Vedula *et al.* [26]. With the goal of multi-camera motion capture, optical flow is first estimated independently for each camera and then triangulated to obtain a 3D motion field. Later work, mostly based on only two views, is dominated by variational approaches. Among these, some again decouple the estimation by first estimating stereo correspondence and then finding flow fields consistent with the disparities, *e.g.* [19, 33]. In contrast, [11] still uses a 2D parametrization, but exploits correlations between depth and motion by estimating them jointly. [25] additionally allows for changes in the relative pose of the stereo rig, and alternates between updating the scene flow and the relative pose. To alleviate the bias of regularization in 2D, [1] directly parameterizes the scene flow with depth and 3D motion vectors, and shows that smoothing in 3D improves the reconstructed motion fields. [30] replaces the total variation regularization of the motion field with a prior that penalizes deviations from local rigidity. Taking the idea of rigidity further, [28] proposes to model the scene as a collection of planar regions, each moving rigidly over time. The representation has also been used for tracking with multiple cameras [7]. Here, we adopt the parameterization of [28], which proved to work well on realistic data. As we will show, this allows one to include consistency checks between different views, thus moving away from a single reference view, and to incorporate temporal contraints on a region's motion.

Temporal smoothness assumptions for multi-frame 2D optical flow date back to at least [17], but were limited to small displacements. [2] instead extrapolates motion fields from previous time steps and encourages similarity between the predicted and the estimated flow. This allows for larger displacements, but inference is restricted to the current frame, *i.e.* the past motion field influences the current one, but not vice versa. Later [34] jointly reasons over three consecutive frames, assuming a constant 2D motion field. In contrast, assuming constant 3D scene flow over time here allows us to address more general scenes. [31] relaxes the constant velocity assumption to soft constraints that encourage first and second order smoothness of the motion field. [8] instead uses a soft constraint that requires the 2D motions to lie in a low-rank trajectory space. [21, 22] avoid simple temporal smoothing, and instead jointly estimate the flow and a segmentation into a small number of layers, enforcing constant pixel-to-layer membership. The rationale is that even if the motion changes rapidly, the scene structure should persist over time. In a similar manner, [20] segments a video into several motion layers with long-term temporal consistency. While they estimate a 2D parametric motion for each layer, their primary goal is high-level motion segmentation. Here we make a similar assumption, but for 3D shape and motion: we also group pixels to (planar, rigidly moving) segments and enforce consistency of the segmentation over time. In contrast to motion layers, our model with hundreds of small segments can represent a wider range of scenes.

An important observation here is that exploiting temporal consistency over longer time intervals is easier with an explicit 3D model of shape and motion, because smoothness assumptions are more likely to hold in the 3D scene than in its projections. This fact is exploited in [19], where a Kalman filter at each pixel propagates the geometry and motion estimated by [33] across frames. The prediction is used to detect and remove outliers, but changes neither the present nor the past flow estimate. [12] constructs longer motion trajectories from frame-to-frame stereo and flow. Trajectories that pass several heuristic plausibility checks are included in the final optimization as soft constraints, similar to including feature matches in two-frame optical flow [5]. [18] parameterizes the scene flow in 3D, and also proceeds sequentially, first estimating frame-to-frame scene flow and then smoothing it over time with tensor voting. [6, 13] represent the scene with an explicit deformable 3D mesh, which is fitted to video data. All three approaches target motion capture in controlled settings with many cameras.

Also related to our work are methods that employ (over-)segmentation to make discontinuities explicit, starting with [32] for flow and [23] for stereo matching. While such early work was constrained by the initial segmentation, more recent methods infer or refine the segmentation together with the scene depth [3, 4, 35], the 1D epipolar flow [36], or the 2D optical flow [24]. The representation of [28], which we use here, adapts this idea to scene flow.

Moving away from an arbitrary reference frame, and treating all views equally, has been prominently used in stereo vision in the form of a left-right consistency check. In its simplest form the consistency between the forward and backward disparities is checked in post-processing, *e.g.* [10], but it can also be included directly in the objective [4]. We extend the latter strategy to ensure consistency of the scene flow across all images in a temporal window.

3 Method

Our formulation follows [28] to represent the 3D scene geometry and motion as a collection of piecewise planar regions that move rigidly over time. More specifically, we define the problem of 3D scene flow estimation as determining two assignments, a mapping $\mathcal{S}$ that assigns pixels to spatially localized segments (super-pixels), and a mapping $\mathcal{P}$ that assigns a planar 3D geometry and rigid motion to each segment. These mappings implicitly define the 3D geometry and motion at every pixel. Note that the spatial segmentation $\mathcal{S}$ is free of semantic meaning. Pixels belonging to a moving plane do not necessarily form a connected component. Moreover, an over-segmentation is actually crucial to account for non-planar or articulated objects, as well as to accurately preserve motion and depth discontinuities. There are two key distinctions to the formulation of [28]: First, we not only estimate the scene flow for a reference view, but for all views (in space and time). The main benefit is that we can check consistency of the representation across views, which makes the estimate more robust and allows for improved occlusion handling. This also means that the notion of the segmentation is extended to all views, with the challenge of obtaining a consistent segmentation of the scene over time. Second, we aim to estimate scene flow from more than 2 frames, hence extend the notion of rigid motion through time by assuming constant translational and rotational velocity of the moving planes.

Notation. We formulate our model for the classical two camera stereo-rig configuration, although no actual limitation on the number of cameras exists. We distinguish *left* and *right* camera through a subscript *l,r*. Superscripts $t \in T = \{-1, 0, 1, \ldots\}$ indicate the time step of image acquisition. Despite computing scene flow in all cameras and not having a reference view for representation, we still designate the *left* camera at time step 0 as a canonical view that defines a common coordinate system. This canonical view simplifies the notation and later serves as evaluation basis. W.l.o.g. we assume the camera matrix $\mathbf{K}$ to be identical for both cameras, with projection matrices $(\mathbf{K}|\mathbf{0})$ for the left and $(\mathbf{M}|\mathbf{m})$ for the right camera. For now we assume that the camera rig does not move itself; in Sec. 3.4 we show how to cope with camera ego-motion.

A 3D moving plane $\pi \equiv \pi(\mathbf{R}, \mathbf{t}, \overline{\mathbf{n}})$ is defined by 9 parameters: the rotation matrix $\mathbf{R}$, a translation vector $\mathbf{t}$, and a scaled normal $\overline{\mathbf{n}}$. Note that we assume the motion parameters to describe the rigid motion in one forward time step. Recall that the moving plane is defined in the coordinate system of the canonical view. Assuming that all planes are visible in the canonical view, they cannot pass the origin. We thus define $\overline{\mathbf{n}} \equiv \overline{\mathbf{n}}_l^0$ via the plane equation $\mathbf{x}^\mathsf{T}\overline{\mathbf{n}} = 1$, which holds for all 3D points $\mathbf{x}$ on the plane. Over the course of this section we will need to transform the moving plane also into views (coordinate systems) other than the canonical one. The respective scaled normal can be found by observing that the normal equation must still hold after a rigid transformation. *E.g.*, consider the left camera at time step 1: for all points $\mathbf{x}$ on the transformed normal we have

$$\mathbf{x}^\mathsf{T}\overline{\mathbf{n}}_l^1 = 1 \Leftrightarrow (\mathbf{R}^{-1}\mathbf{x} – \mathbf{R}^{-1}\mathbf{t})^\mathsf{T}\overline{\mathbf{n}}_l^0 = 1 \Leftrightarrow \mathbf{x}^\mathsf{T}\mathbf{R}\overline{\mathbf{n}}_l^0 – \mathbf{t}^\mathsf{T}\mathbf{R}\overline{\mathbf{n}}_l^0 = 1 \Leftrightarrow \overline{\mathbf{n}}_l^1 = \frac{\mathbf{R}\overline{\mathbf{n}}_l^0}{1 + \mathbf{t}^\mathsf{T}\mathbf{R}\overline{\mathbf{n}}_l^0}. \quad (1)$$

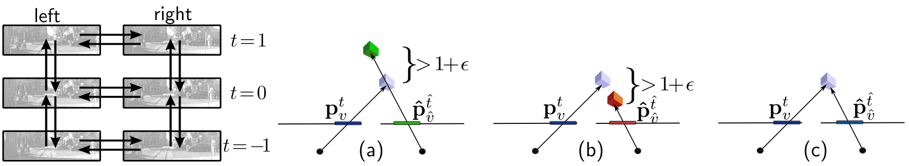
left
right
$t=1$
$t=0$
$t=-1$
$> 1+\epsilon$
$\mathbf{p}_v^t$
$\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$
(a)
$> 1+\epsilon$
$\mathbf{p}_v^t$
$\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$
(b)
$\mathbf{p}_v^t$
$\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$
(c)

Fig. 2. *(left)* Data terms in the three-frame case: Consistency is enforced for spatial and direct temporal neighbors (black arrows). *(right)* Illustration of the per pixel data term: (a) impossible case, (b) occlusion (c) normal case (see text for more details.)

Our scene parameterization, furthermore, allows for a simple transformation of pixel locations to their corresponding position in other views using the homography induced by its assigned moving plane. The homographies from the canonical view I_l^0 to the other views given the moving plane π are written as:

$$ {}^0_l\mathbf{H}^0_r(\pi) = (\mathbf{M} - \mathbf{m}\bar{\mathbf{n}}^\mathsf{T})\mathbf{K}^{-1} \tag{2a}$$

$$ {}^0_l\mathbf{H}^1_l(\pi) = \mathbf{K}(\mathbf{R} - \mathbf{t}\bar{\mathbf{n}}^\mathsf{T})\mathbf{K}^{-1} \tag{2b}$$

$$ {}^0_l\mathbf{H}^1_r(\pi) = \big(\mathbf{M}\mathbf{R} - (\mathbf{M}\mathbf{t} + \mathbf{m})\bar{\mathbf{n}}^\mathsf{T}\big)\mathbf{K}^{-1}. \tag{2c}$$

Homographies between arbitrary view pairs can be obtained by concatenating the transformations above, first transforming back to the canonical view and then into the desired frame, *e.g.* ${}^1_l\mathbf{H}^1_r(\pi) = {}^0_l\mathbf{H}^1_r(\pi) \cdot {}^0_l\mathbf{H}^1_l(\pi)^{-1}$.

Energy. We formally define the problem of 3D scene flow estimation as the minimization of an energy $E(\mathcal{P}, \mathcal{S})$ over two (sets of) mappings: First, the mappings $\mathcal{S} = \{\mathcal{S}_v^t\}$ with $\mathcal{S}_v^t : I_v^t \rightarrow S_v^t$ assign each pixel of camera v at time t to a segment from the set S_v^t, hence define a super-pixel segmentation of each view. This is in contrast to [28], which only infers a segmentation of the reference view. Second, the mappings $\mathcal{P} = \{\mathcal{P}_v^t\}$ with $\mathcal{P}_v^t : S_v^t \rightarrow \Pi$ select a rigidly moving plane for each segment from a candidate set Π of possible moving 3D planes. We define the energy function as

$$E(\mathcal{P}, \mathcal{S}) = E_D(\mathcal{P}, \mathcal{S}) + \lambda E_R(\mathcal{P}, \mathcal{S}) + \mu E_S(\mathcal{S}). \tag{3}$$

The most crucial term is the data term E_D, which unlike [28] not only considers photo-consistency w.r.t. a reference frame, but rather enforces photo-consistency across all neighboring views. Moreover, it considers whether corresponding pixels have a consistent geometric configuration and handles occlusions. The regularization term E_R evaluates the smoothness of motion and geometry at segment boundaries in all images. The final term E_S assesses the quality of the spatial segmentation per view. In the following we first describe our model for only two time steps, and later explain how to extend it to multiple frames in time.

3.1 View-consistent data term

Since we compute 3D scene flow for all views involved, we define a data term for each image. In particular, we check the consistency of the scene flow in each view with its direct neighbors in time, and with the other view(s) at the same time step (Fig. 2, left). With consistency we, on the one hand, mean classical photo-consistency of the images at the corresponding pixel locations; the correspondence is determined from the assigned moving plane $\pi \equiv \pi(\mathbf{R}, \mathbf{t}, \overline{\mathbf{n}})$. On the other hand, because each pixel (in each view) is associated with a moving plane, we can additionally ensure that corresponding pixel locations are geometrically consistent, and detect occlusions. To that end we have to compare depth values induced by the respective moving planes (Fig. 2, right). Note that this form of cross checking is rather different from the occlusion reasoning in [28], and only possible here because we no longer have a reference frame, but instead estimate the scene flow for all views.

Suppose we want to check consistency between a pixel location $\mathbf{p} \equiv \mathbf{p}_v^t$ in view v at time t and its corresponding pixel location $\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$ in view $\hat{v}$ at time $\hat{t}$. Denoting the moving 3D plane of a pixel $\mathbf{p}$ as $\pi_\mathbf{p} = \mathcal{P}_v^t\left(\mathcal{S}_v^t(\mathbf{p})\right)$, the corresponding pixel location in the other view is determined as $\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}} = {}_{v}^{t}\mathbf{H}_{\hat{v}}^{\hat{t}}(\pi_\mathbf{p})\mathbf{p}$. To check geometric consistency, we furthermore determine the depth d of a pixel $\mathbf{p}$ w.r.t. the camera center of an image I_v^t through the inverse scalar product

$$d(\mathbf{p}, \overline{\mathbf{n}}_v^t(\pi)) := \langle \mathbf{K}^{-1}\mathbf{p}, \overline{\mathbf{n}}_v^t(\pi)\rangle^{-1}. \tag{4}$$

This allows us to define our data term for consistency of pixel $\mathbf{p}$ in view v at time-step t and its moving plane $\pi_\mathbf{p}$ with the adjacent view $\hat{v}$ at time-step $\hat{t}$ as

$$\varrho(\mathbf{p}, \hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}) := \begin{cases} \theta_{\text{occ}} & \text{if} \quad d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}, \overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_\mathbf{p}))/d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}, \overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}})) > 1 + \epsilon \\ \theta_{\text{imp}} & \text{if} \quad d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}, \overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}))/d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}, \overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_\mathbf{p})) > 1 + \epsilon \\ \theta_{\text{oob}} & \text{otherwise if} \quad \hat{\mathbf{p}}_{\hat{v}}^{\hat{t}} \notin I_{\hat{v}}^{\hat{t}} \\ \rho(\mathbf{p}, \hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}) + \theta_{\text{mvp}} & \text{otherwise if} \quad \pi_\mathbf{p} \neq \pi_{\hat{\mathbf{p}}_{\hat{\mathbf{v}}}^{\hat{\mathbf{t}}}} \\ \rho(\mathbf{p}, \hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}) & \text{otherwise.} \end{cases} \tag{5}$$

The first two cases consider the relative distance in depth to differentiate between occlusions and implausible geometric configurations, similar to comparing disparity values in the stereo case [4]. In particular, a pixel $\mathbf{p}$ in the first view is being occluded in the second view, if the depth of the moving plane $\pi_\mathbf{p}$ is greater than that of the corresponding plane $\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}$ (both depths determined in the second view). Since we cannot check photo-consistency in case of an occlusion, we assert a fixed penalty θ_{occ}. On the other hand, if the depth of the moving plane $\pi_\mathbf{p}$ is smaller than that of the corresponding plane $\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}$, then an implausible geometric configuration occurs. The 3D point corresponding to the plane stored in pixel $\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$ would be occluded and hence cannot have been observed in the pixel $\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$. We penalize this using the fixed penalty θ_{imp}. The ϵ parameter adds some "softness" to the relative depth comparisons, in order to alleviate aliasing problems induced by the pixel grid resolution and because practical considerations limit us to a finite proposal set of moving planes. Using the third case, we penalize a pixel moving out of the viewing frustum using the fixed penalty θ_{oob}.

The fifth case is the default case when pixels are in geometric correspondence. Specifically, we use the robust census transform ρ_C [37] over a 7×7 neighborhood to measure the photo-consistency, truncated at half the maximum possible data cost $\rho(\cdot,\cdot) = \min\{\rho_C(\cdot,\cdot), 0.5\max(\rho_C)\}$ at a pixel. We impose an additional penalty θ_{mvp} in the fourth case, if pixels are in geometric correspondence, but their moving planes are not the same. This encourages corresponding segments from two views to pick the same moving 3D plane, leading to a view-consistent segmentation.

In practice, we penalize pixels moving out of bounds and classical occlusions identically and set $\theta_{\text{oob}} = \theta_{\text{occ}} = 0.5\max(\rho_C)$. In our experience this ensures a small number of non-submodular edges in the corresponding graph (see Sec. 3.5) and therefore leads to good results. Due to aliasing, we cannot penalize physically implausible configurations with an infinite penalty; we instead set $\theta_{\text{imp}} := 2\theta_{\text{oob}}$, as such a limited penalty prevents deadlocks in the optimization. Note that we rarely encounter implausible configurations in the final estimate. The penalty for not assigning the same plane to pixels in geometric correspondence is empirically set to $\theta_{\text{mvp}} := 5/16\ \theta_{\text{oob}}$, thus allows for deviations from our prior assumption.

Since we compute scene flow for all views involved, we need to sum the per-pixel contribution from Eq. (5) over all pixels of all frames and their considered neighboring views (Fig. 2):

$$E_D(\mathcal{P},\mathcal{S}) := \sum_{t\in T}\sum_{v\in\{l,r\}}\sum_{\mathbf{p}\in I_v^t}\left(\sum_{\hat{v}\neq v}\varrho(\mathbf{p},\hat{\mathbf{p}}_{\hat{v}}^{t}) + \sum_{\substack{\hat{t}\in T\\ |\hat{t}-t|=1}}\varrho(\mathbf{p},\hat{\mathbf{p}}_{v}^{\hat{t}})\right). \tag{6}$$

It is important to note that each view pair is considered twice by the data term, since both of the views have their own scene flow representation.

3.2 Shape and motion regularization

The spatial regularization term promotes piecewise smooth geometry and 3D motion in all views considered. For each of the views, we closely follow [28]. In particular, discontinuities can only occur at segment boundaries, as all pixels within a segment are on the same moving plane. If adjacent pixels are assigned to different moving planes, a penalty is defined by integrating a squared distance function, evaluated at points along the shared edge. Because of the piecewise planarity, the integral can be evaluated in closed form and simplifies to measuring distances only at the endpoints of the shared edge, see [28]. To achieve a certain robustness against object and motion discontinuities, the integrated distance is further embedded into a robust cost function. More formally, assuming that the two adjacent pixels $\mathbf{p}$ and $\mathbf{q}$ lie on different moving planes $\pi_{\mathbf{p}} = \mathcal{P}(\mathcal{S}(\mathbf{p}))$ and $\pi_{\mathbf{q}} = \mathcal{P}(\mathcal{S}(\mathbf{q}))$, then we can define the induced penalty as:

$$e_R(\mathbf{p},\mathbf{q}) := w_{\mathbf{p},\mathbf{q}}\big(\psi\left(||\mathbf{d}_1||^2+||\mathbf{d}_2||^2+\langle\mathbf{d}_1,\mathbf{d}_2\rangle+\gamma^2||\mathbf{d}_n||^2\right) + \tag{7}$$

$$\psi\left(||\mathbf{d}_1^m||^2+||\mathbf{d}_2^m||^2+\langle\mathbf{d}_1^m,\mathbf{d}_2^m\rangle+\gamma^2||\mathbf{d}_n^m||^2\right)\big). \tag{8}$$

Here the vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ describe the distance in geometry, and $\mathbf{d}_1^m$ and $\mathbf{d}_2^m$ the distance in motion at the two endpoints of the shared edge. The vectors $\mathbf{d}_n$

and $\mathbf{d}_n^m$ define the distance of the normals before and after the moving plane induced motion is applied. While it appears natural to measure these distances in 3D space, current scene flow benchmarks are biased toward 2D accuracy, hence a 2D regularization delivers better results. Therefore, we use disparity for geometry regularization, and 2D flow and disparity difference across time to regularize the motion. Robustness is achieved using a truncated penalty function $\psi(y) := \min(\sqrt{y}, \eta)$; we set $\eta := 20$ and $\gamma := 1$ for both geometry and motion.

The weight $w_{\mathbf{p},\mathbf{q}}$ allows to take into account the image structure and the length of the edge between the pixels. Since we found the weighting scheme from [28] based on bilateral filtering to be noisy, we instead follow [34] and employ the anisotropic diffusion tensor $D^{\frac{1}{2}} = \exp(-\alpha|\nabla I|)gg^{\mathsf{T}} + g^{\perp}(g^{\perp})^{\mathsf{T}}$ with $\alpha = 5$, thereby assuming $I \in [0, 1]$. The direction of the image gradient $g = \nabla I / |\nabla I|$ is determined in the middle between $\mathbf{p}$ and $\mathbf{q}$ via bicubic interpolation. We then define the weight as

$$w_{\mathbf{p},\mathbf{q}} := |D^{\frac{1}{2}} \overrightarrow{\mathbf{pq}}|. \tag{9}$$

Because we compute the scene flow simultaneously in all images, we also apply regularization on all views and define the full spatial regularizer as

$$E_R(\mathcal{P}, \mathcal{S}) := \sum_{t \in T} \sum_{v \in \{l,r\}} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{N}(I_v^t)} w_{\mathbf{p},\mathbf{q}} e_R(\mathbf{p}, \mathbf{q}). \tag{10}$$

Here, $\mathcal{N}$ are all neighboring pixels of the respective image (8-neighborhood).

3.3 Spatial segmentation regularization

The segmentation regularizer promotes the spatial coherence of the underlying over-segmentation. We again define the energy for all views considered:

$$E_S(\mathcal{S}) = \left(\sum_{\substack{t \in T, \\ v \in \{l,r\}}} \sum_{\substack{(\mathbf{p},\mathbf{q}) \in \mathcal{N}(I_v^t), \\ \mathcal{S}(\mathbf{p}) \neq \mathcal{S}(\mathbf{q})}} w_{\mathbf{p},\mathbf{q}} \right) + \sum_{\mathbf{p} \in I_l^0} \begin{cases} 0, & \exists\, \mathbf{e} \in \mathcal{E}(s_i) : ||\mathbf{e} - \mathbf{p}||_\infty < N_S \\ \infty, & \text{else.} \end{cases} \tag{11}$$

The first term takes the form of a pairwise Potts model, which encourages segment boundaries to coincide with the image edges. We use the weights from the diffusion tensor (Eq. 9) to take into account the edge contrast. The second term restricts the size of a segment within the canonical view (maximum extent of $2N_S - 1$ with $N_S = 20$) and binds them to their respective seed point $\mathbf{e} \in \mathcal{E}(s_i)$. The seed points are spaced on a regular grid. The key motivation behind this is that it reduces the time needed for optimizing the mapping $\mathcal{S}$, because only a limited set of segments needs to be considered at any pixel. Note that the second term only needs to be applied to the canonical view, since the data term from Eq. (5) encourages the segmentations in the other views to be consistent.

This segmentation regularizer is based on ideas of [27], where a similar energy is used to compute an over-segmentation of one image, and is also employed in [28], but only w.r.t. the reference image.

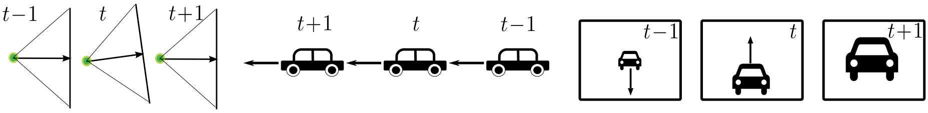

Fig. 3. Variation in camera pitch limits the validity of the constant velocity model. *(left)* A scene observed by a moving camera with varying pitch. *(right)* Camera images with induced 2D flow (black arrow). We compensate camera pitch by removing the ego-motion of the camera.

3.4 Multiple frame extension

Extending our formulation to more than two frames seems straightforward on a first glance: The spatial and segmentation regularizers can be trivially extended to any number of frames, but the data term is more subtle. As discussed, we generally assume motion of constant translational and rotational velocity in case we have more than just two time steps. Note that in many applications this assumption is valid, especially because we restrict ourselves to only a short time interval. Consequently, one could extend the data term by defining appropriate homographies between the views. Assuming constant velocity in rotation and translation, this can be achieved by concatenation of terms from Eq. (2). Care must be taken to use the correct normal, which must be transformed into the appropriate view coordinate system. This can be similarly achieved by repeated application of Eq. (1), again assuming constant velocity.

In certain applications, *e.g.* the automotive application in our experiments, the constant velocity assumption is challenged by a common and high-frequent pitching motion of the stereo rig, which can arise from undulations in the road surface or from not perfectly securing the rig. Because the motion between two time steps is always estimated relative to the respective camera coordinate system, even small changes in relative camera position already lead to significant changes in the relative geometry and motion (Fig. 3). Therefore we extend our formulation to incorporate ego-motion estimates for the different time steps.

In particular, we first estimate the relative ego-motion $\mathbf{E}^t = [\mathbf{Q}^t|\mathbf{s}^t]$ between all consecutive time steps t and $t{+}1$. When computing the homographies between subsequent time steps, we first apply the motion induced by the moving plane representation (disregarding any ego-motion) and then the relative ego-motion: Since the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ of a moving plane come from a proposal (see below), which is computed for the canonical view unaware of any ego-motion, we need to disregard the relative ego-motion of the canonical view $\mathbf{E}^0$ first by applying $(\mathbf{E}^0)^{-1} = [(\mathbf{Q}^0)^{-1}| - (\mathbf{Q}^0)^{-1}\mathbf{s}^0]$. For example, in case of computing a homography between frame t and $t+1$ in the left view, we have

$$ {}_l^t\mathbf{H}_l^{t+1}(\pi) = \mathbf{K}\Big(\mathbf{Q}^t(\mathbf{Q}^0)^{-1}\mathbf{R} - \big(\mathbf{Q}^t(\mathbf{Q}^0)^{-1}(\mathbf{t}-\mathbf{s}^0)+\mathbf{s}^t\big)(\overline{\mathbf{n}}_l^t)^\mathsf{T}\Big)\mathbf{K}^{-1}. \qquad (12)$$

Other homographies can be corrected for ego-motion accordingly.

3.5 Optimization and proposal generation

Our piecewise rigid model energy (Eq. 3) amounts to a CRF with continuous, 9-dimensional variables for motion and geometry, and discrete variables for the pixel-to-plane assignment. Like [28] we perform inference in two steps with fusion moves [14]. First, starting from a fixed segmentation $\mathcal{S}$ we select a moving plane for each segment from a finite set of proposals; then we update the segmentation $\mathcal{S}$, given the geometry and motion $\mathcal{P}$ of the segments. To bootstrap this two-step procedure one needs an initial segmentation $\mathcal{S}$. [28] proposes to start from an intensity-based super-pixel segmentation. We found that the initial segmentation is not critical, and instead start from a regular checkerboard grid (16 pixel edge length) as trivial "segmentation". The center points of the grid cells also serve as seed points $\mathbf{e} \in \mathcal{E}$ (see Eq. 11). Optimization w.r.t. $\mathcal{S}$ will eventually refine the segmentation and adjust it to depth and motion boundaries, consistently in all views. Aside from being more efficient, the grid structure also reduces aliasing from an uneven size of the segments across views.

When first solving for $\mathcal{P}$, we can treat the segments as large pixels and ignore the segmentation term E_S, as it is independent of $\mathcal{P}$. To cope with aliasing induced by the initial (not view-consistent) grid segmentation we relax the consistency constraint and set $\epsilon := 0.1$ and $\theta_{\text{mvp}} := 3/16\ \theta_{\text{oob}}$. This softer setting ensures that proposals are not prematurely discarded because of the inaccurate initial segmentation. Edge weights (Eq. 9) are summed along the segment edges.

For our fusion move framework we need a set of moving plane proposals. These are generated by running 2D stereo [10] and optical flow [29], and refining the output with a two-frame version of our method, leading to a significant reduction of the initial proposal set. The refinement, done only in the canonical view for the same grid segmentation, resembles the segment-to-plane assignment step of [28]. For the multi-frame case we generate proposals for all consecutive frame pairs ($t=-1$ and $t=0$ for 3 time steps, and also $t=1$ for 4 time steps). To avoid unnecessarily inflating the proposal set, proposals from other time steps are only kept if they differ significantly from already extracted ones nearby. Proposals are considered valid only in a 192×144 pixel (12×9 cells) neighborhood centered at the seed point in the canonical frame, to speed up optimization. During a fusion move we project the neighborhood into all other views and only instantiate the graph for segments within the projected box.

Once the segment-to-plane mapping $\mathcal{P}$ has been found, we infer the pixel-to-segment assignment $\mathcal{S}$ in a similar manner. *I.e.*, we discard all unused moving plane proposals and optimize again, this time labeling individual pixels rather than grid cells. The region constraint from Eq. (11) ensures the locality of the fusion move. Because our consistency decisions are made on a per pixel basis we can penalize inconsistencies more strictly now and set $\epsilon := 0.015$.

Our local expansion strategy allows to optimize multiple non-overlapping image regions in parallel. With our current implementation we observe runtimes of 23s (2 time steps) and 46s (3 time steps) to solve for $\mathcal{P}$, respectively 18s and 32s to solve for $\mathcal{S}$. Timings were measured for 0.5 Mpixel images on a dual *Intel Core i7*, working with $\sim$ 1850 segments and proposals from 3 time steps.

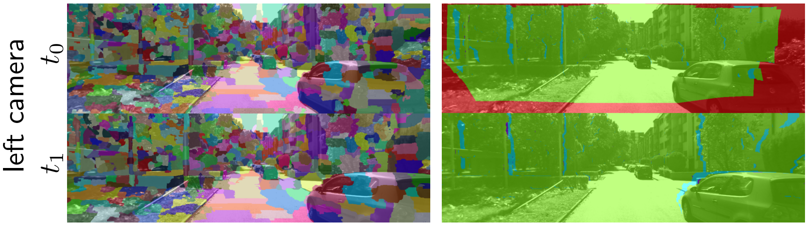

Fig. 4. Example from the KITTI training set (#191). *(left)* Consistent super-pixel segmentation *(right)* Active data term ϱ (Eq. 5). Colors denote normal photo-consistency (*green*), out of bounds (*red*), occluded (*light blue*), and implausible (*dark blue*).

4 Evaluation

For our evaluation we fix the remaining parameters to of our algorithm to $\lambda = 1/50$ and $\mu = \lambda/5$, and scale the census transform to deliver values between 0 and 1.6, thus $\max \rho(\cdot, \cdot) = 0.8$ in Eq. (5). We begin by illustrating the internal representation of our model in Fig. 4. On the left we depict the (consistent) over-segmentation overlayed on two consecutive frames and beside it the assigned states of the data term ϱ from Eq. (5), for the same images.

4.1 Qualitative Evaluation

We first show a hard example from the KITTI benchmark (Fig. 5). Most optical and scene flow methods fail on these images because of severe lens flares in both cameras. However, the presence and the location of the artifacts are not consistent through all views (although they are rather consistent in consecutive frames). Our method is able to exploit the absence of a consistent depth and motion pattern for the flare, and reconstructs the scene flow reasonably well, with only 3.7% of the disparities and 8.1% of the flow vectors (including occluded areas) outside the standard 3-pixel error threshold of KITTI. We note that the improvement is achieved only through view- and multi-frame consistency – imaging artifacts that exhibit a consistent motion pattern across all images still can lead to erroneous reconstructions (which is however rather unlikely because the two views stem from physically different cameras).

In Fig. 6 we present further results on difficult outdoor scenes from [15]. On the left the input images are shown, on the right are the disparities and the flow (reprojected to 2D) estimated with our method from 3 consecutive stereo pairs. Only qualitative results are given, as no ground truth is available. The examples show that even under adverse imaging conditions our model correctly recovers not only the dominant background, but also the motion of smaller, independent objects. The scenes feature challenges such as reflections on the wet road, strong occlusions at a crossroads, and saturated headlights. The most difficult examples even include flares from headlights on the wet windscreen, and heavy snowfall. Also from this dataset is the example shown in Fig. 1, in which the windscreen wiper occludes a large part of the viewing field. This particular scene is extremely hard to reconstruct with only a single reference view.

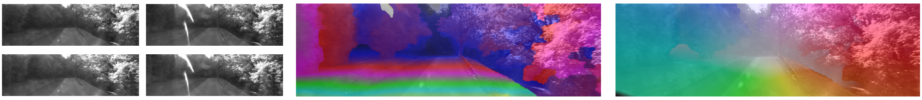

Fig. 5. A hard example (KITTI training set, #74). *(left)* Input frames. *(right)* Reconstructed scene flow, reprojected to disparity and 2D flow field (from left to right).

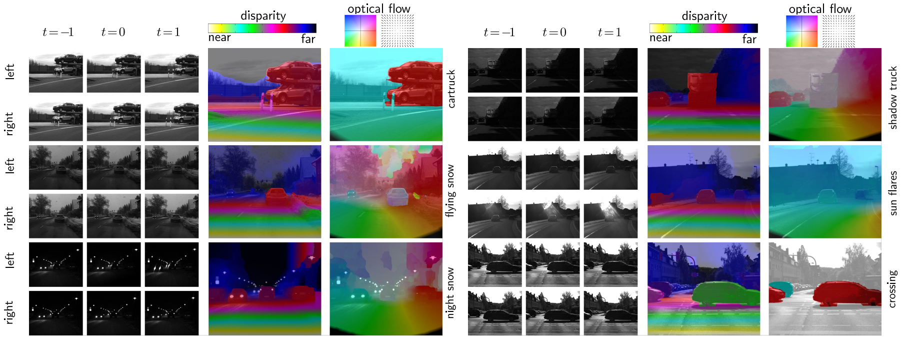

Fig. 6. Results for challenging examples from [15]. *(left)* Input frames. *(right)* Reconstructed scene flow, reprojected to disparity and 2D flow field. *Best viewed on screen.*

4.2 KITTI benchmark

We quantitatively evaluate the performance of our algorithm on the KITTI dataset [9]. The images were acquired with a calibrated stereo rig at a resolution of 1240 × 376 pixels. The cameras are mounted on top of a car together with a laser scanner, which delivers semi-dense ground truth. KITTI has become a standard testbed for modern stereo and optical flow algorithms. It is challenging mainly for two reasons, *(i)* very large displacements in both stereo ($>$150 pixels) and flow ($>$250 pixels); and *(ii)* having real outdoor scenes under realistic lighting, with shadows, saturation, specular reflections, lens flare, *etc.*

The benchmark consists of a “training” set of 194 images with public ground truth and a test set of 195 images, for which the ground truth is withheld. The data is provided in stereo video snippets of 20 frames, so that our multi-frame method can be applied. We analyze different variants of our method on the training set, and run it on the test set to compare to the state of the art.

Table 1 summarizes the results of the evaluation on the training images. As error measures we use the average end point error (AEP) and the KITTI metric, *i.e.* the percentage of pixels that deviate by more than 2/3/4/5 pixels from the ground truth. Both metrics are calculated both for the complete images ($\surd$), and only for the non-occluded pixels ($\times$). The following variants are evaluated: view-consistent estimation for two frames (*VC-2F*), three frames (*VC-3F*) and four frames (*VC-4F*). To separate the impact of propagating proposals over multiple frames from the impact of multi-frame optimization, we also test a variant in which proposals are extracted from 3 frames, but model optimization is done only for two frames (*VC-2F+*). As a baseline we also run our method for only

Table 1. *KITTI* metric (% of flow vectors / disparities above 2/3/4/5 pixels of endpoint error) and average endpoint error [px], for the complete *KITTI* training set.

	Flow										**Stereo**									
	KITTI metric								**AEP**		**KITTI metric**								**AEP**	
Occ pix.	✓				×				✓	×	✓				×				✓	×
	2px	3px	4px	5px	2px	3px	4px	5px			2px	3px	4px	5px	2px	3px	4px	5px		
PRSF [28]	9.9	7.3	6.0	5.2	5.8	4.1	3.3	2.8	2.5	1.2	8.0	5.6	4.4	3.7	7.0	4.8	3.8	3.1	1.2	1.0
VC-2F	8.0	5.5	4.2	3.4	4.4	2.9	2.2	1.7	1.4	0.8	6.8	4.7	3.6	3.0	5.8	3.9	3.0	2.5	1.0	0.8
VC-2F+	7.4	4.9	3.7	3.0	4.2	2.6	1.9	1.5	1.3	0.7	6.4	4.3	3.3	2.8	5.5	3.7	2.8	2.3	0.9	0.8
VC-3F	6.6	4.1	3.0	2.3	4.1	2.6	1.9	1.5	1.1	0.7	5.5	3.7	2.8	2.3	5.0	3.3	2.5	2.1	0.8	0.7
VC-4F	6.5	4.0	3.0	2.4	4.0	2.5	1.9	1.5	1.1	0.7	5.3	3.6	2.7	2.2	4.9	3.3	2.5	2.0	0.8	0.7

two frames and using a single reference view (*PRSF*). The baseline is essentially the same as the basic version of [28], called "PRSPix-2D" in that paper.

Moving from a single reference view to view-consistent estimation (*PRSF vs. VC-2F*) already yields significant improvements. In the standard KITTI metric (3px error threshold) the gains are 25% for flow, and 16% for stereo (respectively 29% and 19% in visible areas). In line with these results also the AEP is reduced by 44% and 17% (respectively, 33% and 20%), showing that view-consistency is especially helpful in the presence of occlusions.

Including proposals from the previous frame (*VC-2F+*) already improves the results further. A much larger improvement however is brought about by moving to three frames (*VC-3F*). Note in particular the strong gains in occluded areas, which significantly reduce the errors on the full images despite the small number of affected pixels. When adding a fourth frame (*VC-4F*) we observe diminishing returns, with only marginal improvements over the three-frame case. Compared to the baseline, our best result reduces the KITTI error on the full images by 45% for flow and by 36% for stereo. The corresponding AEPs drop by 56%, respectively 33%. We submitted the three-view version *VC-3F* to the official KITTI benchmark. In the evaluation on full images including occluded areas ("Out-All") the proposed scene flow method current achieves the best results for both flow and stereo, among >40 submissions. Note that in contrast to the nearest competitor, our method can handle scenes with independently moving objects (see Fig. 6), which are rare in this benchmark, but not in general scenes.

5 Conclusion

In this paper we have addressed the question of how to exploit consistency over time and between viewpoints for dense 3D scene flow estimation. For piecewise planar and rigid scenes, we have shown a way to leverage information from multiple consecutive frames of a stereo video, and thereby significantly improve both shape and 3D motion estimation. The proposed model has proven remarkably robust against outliers, occlusions and missing evidence, and makes it possible to estimate depth and motion of road scenes even under adverse imaging conditions, where most methods fail. In future work we plan to handle deviations from the constant velocity assumption in a more flexible manner.

References

1. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: A view centered variational approach. In: CVPR (2010)
2. Black, M.J., Anandan, P.: Robust dynamic motion estimation over time. In: CVPR (1991)
3. Bleyer, M., Rother, C., Kohli, P.: Surface stereo with soft segmentation. In: CVPR (2010)
4. Bleyer, M., Rother, C., Kohli, P., Scharstein, D., Sinha, S.N.: Object stereo – Joint stereo matching and object segmentation. In: CVPR (2011)
5. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. TPAMI 33(3), 500–513 (2011)
6. Courchay, J., Pons, J.P., Monasse, P., Keriven, R.: Dense and accurate spatio-temporal multi-view stereovision. In: ACCV (2009)
7. Devernay, F., Mateus, D., Guilbert, M.: Multi-camera scene flow by tracking 3-D points and surfels. In: CVPR (2006)
8. Garg, R., Roussos, A., Agapito, L.: A variational approach to video registration with subspace constraints. IJCV pp. 1–29 (2013)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? In: CVPR (2012)
10. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. TPAMI 30(2), 328–341 (2008)
11. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: ICCV (2007)
12. Hung, C.H., Xu, L., Jia, J.: Consistent binocular depth and scene flow with chained temporal profiles. IJCV 102(1-3), 271–292 (Mar 2013)
13. Klaudiny, M., Hilton, A.: Cooperative patch-based 3D surface tracking. In: Proc. of the 8th International Conference on Visual Media Production (2011)
14. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for Markov random field optimization. TPAMI 32(8), 1392–1405 (2010)
15. Meister, S., Jähne, B., Kondermann, D.: Outdoor stereo camera system for the generation of real-world benchmark data sets. Optical Engineering 51(02) (2012)
16. Müller, T., Rannacher, J., Rabe, C., Franke, U.: Feature- and depth-supported modified total variation optical flow for 3D motion field estimation in real scenes. In: CVPR (2011)
17. Murray, D.W., Buxton, B.F.: Scene segmentation from visual motion using global optimization. TPAMI 9(2), 220–228 (Feb 1987)
18. Park, J., Oh, T.H., Jung, J., Tai, Y.W., Kweon, I.S.: A tensor voting approach for multi-view 3D scene flow estimation and refinement. In: ECCV (2012)
19. Rabe, C., Müller, T., Wedel, A., Franke, U.: Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In: ECCV (2010)
20. Schoenemann, T., Cremers, D.: High resolution motion layer decomposition using dual-space graph cuts. In: CVPR (2008)
21. Sun, D., Sudderth, E.B., Black, M.J.: Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In: NIPS (2010)
22. Sun, D., Wulff, J., Sudderth, E., Pfister, H., Black, M.: A fully-connected layered model of foreground and background flow. In: CVPR (2013)
23. Tao, H., Sawhney, H.S.: Global matching criterion and color segmentation based stereo. In: WACV (2000)

24. Unger, M., Werlberger, M., Pock, T., Bischof, H.: Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In: CVPR (2012)
25. Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C., Theobalt, C.: Joint estimation of motion, structure and geometry from stereo sequences. In: ECCV (2010)
26. Vedula, S., Baker, S., Collins, R., Kanade, T., Rander, P.: Three-dimensional scene flow. In: CVPR (1999)
27. Veksler, O., Boykov, Y., Mehrani, P.: Superpixels and supervoxels in an energy optimization framework. In: ECCV (2010)
28. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: ICCV (2013)
29. Vogel, C., Roth, S., Schindler, K.: An evaluation of data costs for optical flow. In: GCPR. pp. 343–353 (2013)
30. Vogel, C., Schindler, K., Roth, S.: 3D scene flow estimation with a rigid motion prior. In: ICCV (2011)
31. Volz, S., Bruhn, A., Valgaerts, L., Zimmer, H.: Modeling temporal coherence for optical flow. In: ICCV (2011)
32. Wang, J.Y.A., Edward, Adelson, H.: Representing moving images with layers. IEEE Transactions on Image Processing 3, 625–638 (1994)
33. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: ECCV (2008)
34. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: BMVC (2009)
35. Yamaguchi, K., Hazan, T., McAllester, D., Urtasun, R.: Continuous Markov random fields for robust stereo estimation. In: ECCV (2012)
36. Yamaguchi, K., McAllester, D., Urtasun, R.: Robust monocular epipolar flow estimation. In: CVPR (2013)
37. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: ECCV (1994)