To appear: International Journal of Computer Vision Marr Prize (ICCV 2013) Special Issue. Volume 111, No.3

3D Scene Flow Estimation with a Piecewise Rigid Scene Model

Christoph Vogel · Konrad Schindler · Stefan Roth

Abstract 3D scene flow estimation aims to jointly recover dense geometry and 3D motion from stereoscopic image sequences, thus generalizes classical disparity and 2D optical flow estimation. To realize its conceptual benefits and overcome limitations of many existing methods, we propose to represent the dynamic scene as a *collection of rigidly moving planes*, into which the input images are segmented. Geometry and 3D motion are then jointly recovered alongside an over-segmentation of the scene. This piecewise rigid scene model is significantly more parsimonious than conventional pixel-based representations, yet retains the ability to represent real-world scenes with independent object motion. It, furthermore, enables us to define suitable scene priors, perform occlusion reasoning, and leverage discrete optimization schemes toward stable and accurate results.

Assuming the rigid motion to persist approximately over time additionally enables us to incorporate multiple frames into the inference. To that end, each view holds its own representation, which is encouraged to be consistent across all other viewpoints and frames in a temporal window. We show that such a *view-consistent multi-frame scheme* significantly improves accuracy, especially in the presence of occlusions, and increases robustness against adverse imaging conditions. At the time of writing (August 2014) our method achieves leading performance on the KITTI benchmark, for both flow and stereo.

Keywords 3D Scene Flow · Stereo · Motion Estimation · Piecewise Planarity · Piecewise Rigidity · Segmentation

Christoph Vogel · Konrad Schindler
Photogrammetry & Remote Sensing, ETH Zurich
Zürich, Switzerland
E-mail: {vogel,konrad.schindler}@geod.baug.ethz.ch

Stefan Roth
Department of Computer Science, TU Darmstadt
Darmstadt, Germany
E-mail: sroth@cs.tu-darmstadt.de

1 Introduction

The scene flow of a dynamic scene is defined as a dense representation of the 3D shape and its 3D motion field. Scene flow estimation aims to extract this information from images captured by two (or more) cameras at two (or more) different time instants. Applications that benefit from knowing the scene flow include 3D video generation for 3D-TV (Hung et al, 2013), motion capture (Courchay et al, 2009; Park et al, 2012; Vedula et al, 1999), and driver assistance (*e.g.*, Müller et al, 2011; Rabe et al, 2010; Wedel et al, 2008). The 3D scene flow can be seen as a combination of two classical computer vision problems – it generalizes optical flow to 3D, or alternatively, dense stereo to dynamic scenes.

While progress in dense binocular stereo (Bleyer et al, 2011b; Hirschmüller, 2008; Yamaguchi et al, 2012, *etc.*) and optical flow (Brox et al, 2004; Sun et al, 2010; Unger et al, 2012, among others) has been both steady and significant over the years, the performance of 3D scene flow algorithms (*e.g.*, Basha et al, 2010; Huguet and Devernay, 2007; Wedel et al, 2008) had been lacking in comparison. Only recently, methods emerged (Vogel et al, 2013b, 2014; Yamaguchi et al, 2014) that could leverage the additional information present in stereo video streams and outperform their dedicated two-dimensional counterparts at their respective tasks.

This may seem surprising, because 3D scene flow has a lot of commonalities with stereo and optical flow. This includes some of the principal difficulties, for example matching ambiguities due to insufficient evidence from the local appearance, or the aperture problem (more precisely a 3D version of it). Therefore, 3D scene flow estimation similarly requires prior assumptions about geometry and motion. A recent trend in both stereo and optical flow is to move away from simple pixelwise smoothness priors, as they have been found limiting. More expressive priors have been introduced, for example, by virtue of an over-parameterization (Nir et al, 2008), layered (Sun et al, 2010) or piecewise pla-

nar scene models (Bleyer et al, 2011b). In contrast, there has been relatively little work on using advanced priors in scene flow estimation. One exception is a regularizer that promotes local rigidity (Vogel et al, 2011), a common property of realistic scenes, by penalizing deviations from it.

Piecewise rigid scene model. Our first contribution is to go one step further and represent dynamic scenes as a collection of *planar regions, each undergoing a rigid motion*. Following previous work in stereo (Bleyer et al, 2011b), we argue that most scenes of interest consist of regions with a consistent motion pattern, into which they can be segmented. Consequently, we aim to jointly recover an implicit (over-)segmentation of the scene into planar, rigidly moving regions, as well as the shape and motion parameters of those regions (see Fig. 1). As we will show, such a parsimonious model is well-suited for many scenes of interest: The approximation holds well enough to capture the shape and motion of many real-world scenarios accurately, including scenes with independent object motion, while the stronger regularization affords stability. At the same time, reasoning in terms of rigid planar regions rather than pixels drastically reduces the number of unknowns to be recovered. Thereby, we additionally address the challenge of optimization or inference, one of the other principal difficulties that 3D scene flow shares with stereo and optical flow.

We (implicitly) represent 3D scene flow by assigning each pixel to a *rigidly moving 3D plane*, which has 9 continuous degrees of freedom (3 plane parameters, 6 motion parameters). To bootstrap their estimation, we start not from individual pixels, but from an initial superpixel segmentation of the scene. Based on the superpixels we compute a large, but finite set of candidate (moving) planes, and cast scene flow estimation as a *labeling problem*. The inference thus assigns each pixel to one of the segments (superpixels), and each segment to one of the candidate moving planes. We split the optimization into two steps. First, we find the best moving plane for each segment; reasoning on this coarser level captures long-range interactions and significantly simplifies and stabilizes the inference. Second, we go back to the pixel level and reassign pixels to segments; this step cleans up inaccuracies of the segmentation, whose initial boundaries were generated without taking the previously unknown surface or motion discontinuities into account.

View-consistent multi-frame scene flow. Our second contribution is to exploit this piecewise rigid scene model to overcome two limitations of existing scene flow techniques. We begin by observing that *(i)* there is no conceptual reason for a privileged reference view (*e.g.*, Basha et al, 2010; Rabe et al, 2010; Valgaerts et al, 2010; Vogel et al, 2011; Wedel et al, 2008), as systematic challenges in imaging (specular reflections, occlusions, noise, lack of contrast, *etc.*) affect all

Fig. 1 Example scene from Vaudrey et al (2008): Jointly estimated 3D geometry, 3D motion vectors, and superpixel boundaries, rendered from a different viewpoint.

frames, but not necessarily equally. Thus parameterizing the model w.r.t. a single viewpoint may in fact ignore important evidence present in other views (*c.f.* Fig. 2); *(ii)* data usually comes in the form of a stereo video sequence, and it appears wasteful not to exploit longer time intervals, especially in light of the first observation.

We go on to show that our piecewise planar and rigid scene model can be extended to simultaneously estimate geometry and 3D motion *over longer time intervals,* and to ensure that the estimate is *consistent across all views* within the considered time window. To that end we simultaneously parameterize the scene flow w.r.t. all views. While it may not be surprising that considering longer sequences may help motion estimation, at least in classical 2D optical flow estimation multi-frame extensions have largely not had the desired effect; two-frame methods are still the state of the art (see Baker et al, 2011; Geiger et al, 2012). We argue that long-term constraints may be more helpful in scene flow, since the representation resides in 3D space, rather than in a 2D projection. Constraints caused by physical properties, such as inertia, remain valid in the long term, and can be exploited more directly.

To make the estimate consistent across all views from a longer sequence, we constrain the segmentation to remain stable over time, enforce coherence of the representation between different viewpoints, and integrate a dynamic model that favors *constant velocity of the individual planes*. We empirically found this assumption to be valid as long as segments and temporal windows do not get too large.

Contributions. The main features of our proposed approach are: *(i)* A novel scene flow model that represents the scene with piecewise planar, rigidly moving regions in 3D space, featuring regularization between these regions and explicit occlusion reasoning; *(ii)* a view-consistent model extension that leads to improved results in challenging scenarios, by simultaneously representing 3D shape and motion w.r.t. every

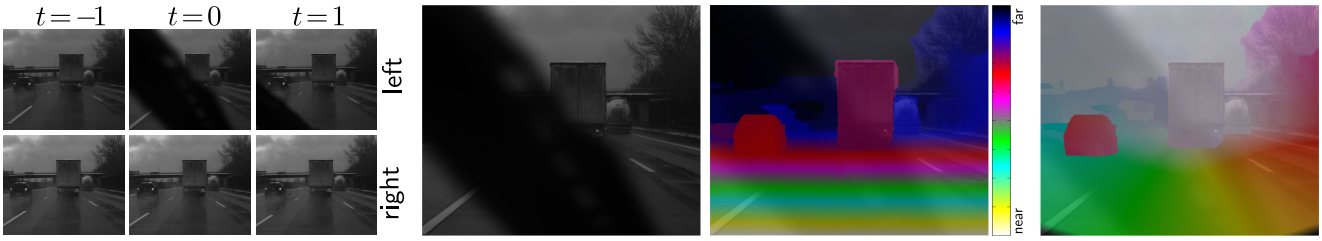


Fig. 2 Consistency over multiple frames makes scene flow estimation robust against severe disturbances like the windscreen wiper. *(left)* Input frames. *(center left)* The left view at time $t = 0$. *(right)* Our scene flow estimate for that viewpoint (shown, from left to right, as disparity and reprojected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

image in a time interval, while demanding consistency of the representations; *(iii)* a multi-frame extension that yields a temporally consistent piecewise-planar segmentation of the scene and favors constant 3D velocity over time; and *(iv)* a clean energy-based formulation capturing all these aspects, as well as a suitable discrete inference scheme. The formulation can – at least conceptually – handle any number of viewpoints and time steps.

We demonstrate the advantages of our model using a range of qualitative and quantitative experiments. On particularly hard qualitative examples, our model turns out to be remarkably resistant to missing evidence, outliers, and occlusions. As a quantitative testbed we evaluate our method on the challenging KITTI dataset of real street scenes, using both stereo and flow benchmarks. In both benchmarks we achieve leading performance, even beating methods that are designed for the specific situation in the benchmark. At the time of writing (August 2014) our full (view-consistent multi-frame) model is the top performing method for both optical flow and stereo, when evaluated on full images including occlusion areas.

The present paper is based on two conference publications (Vogel et al, 2013b, 2014). We here describe the approach in greater detail, including the model itself, the inference scheme, proposal generation, and technical issues of occlusion reasoning. Moreover, we present a deeper analysis and more detailed comparison between the conventional parameterization and the view-consistent model, an experimental investigation of different optimization strategies, and study the influence of parameters on the quantitative results.

2 Related Work

Vedula et al (1999) first defined scene flow as the collective estimation of dense 3D geometry and 3D motion from image data. Their approach operates in two steps. After computing independent 2D optical flow fields for all views of the scene, the final 3D flow field is fit to the 2D flows, thus neglecting the image data in this step. Similarly, Wedel et al (2008) and Rabe et al (2010) proceed sequentially on the data of a calibrated stereo camera system. Starting from a precomputed disparity map, optical flow for a reference frame and disparity difference for the other view are estimated. Possibly the first to calculate geometry and flow jointly in a two-view setup were Huguet and Devernay (2007), addressing the problem in a variational formulation. The problem was generalized by Valgaerts et al (2010) to work with an unknown relative pose between the cameras, solely assuming knowledge of the camera intrinsics. They alternate scene flow calculation with estimating the relative camera pose. Operating entirely with 2D entities, these approaches partially neglect the 3D origin of the data. In particular, the proposed 2D regularizer encourages smooth projections, but not necessarily smooth 3D scene flow.

In contrast, Basha et al (2010) choose a 3D parameterization by depth and a 3D motion vector w.r.t. a reference view and estimate all parameters jointly, extending the popular optical flow method of Brox et al (2004) to scene flow. Arguing that a total variation prior on the 3D motion field is biased for realistic baselines, Vogel et al (2011) propose a regularizer that encourages locally rigid motion. Our model also employs a local rigidity assumption, but here we explicitly identify regions with a consistent motion pattern, into which the image is segmented.

The history of local rigidity priors dates back at least to Adiv (1985), who employed this assumption for sparse motion estimation. The idea was later extended to sparse scene flow by Carceroni and Kutulakos (2002). In a similar manner, Devernay et al (2006) extend the Lucas-Kanade technique (1981) to multi-camera scene flow and track planar, rigidly moving regions in 3D over several frames. While the scene representation of Carceroni and Kutulakos (2002); Devernay et al (2006) is similar to ours, there the regions move independently without interaction imposed by a global objective. Furukawa and Ponce (2008) go one step further and use the locally tracked rigid patch motion as input for a global optimization step, where the connectivity is defined by an explicit surface model, thus limiting admissible scenes to a fixed topology. 3D rigid body motions are further exploited in the context of scene flow estimation from RGB-D data by Hornacek et al (2014). They do not need to assume local surface planarity, but exploit the additional information

from the depth sensor and use a local rigidity prior to overcome large displacements. For computing optical flow, Nir et al (2008) over-parameterize the 2D flow field and explicitly search for rigid motion parameters, while encouraging their smoothness.

Most previous dense 3D scene flow methods have in common that they penalize deviations from spatial smoothness in a robust manner. Explicit modeling of discontinuities by means of segmentation or layer-based formulations has a long history in the context of stereo (Tao and Sawhney, 2000) and optical flow (Black and Jepson, 1996; Wang and Adelson, 1994). Wang and Adelson (1994) start from optical flow and define the regions by k-means clustering, where cluster centers are affine motions. Given an initial segmentation Black and Jepson (1996) similarly obtain a parametric model for each image region, which is then used as an additional constrain a final optical flow computation, thus the model allows for violations of the parametric model. These ideas recently gained renewed attention, however modern methods do not hold the segmentation or motion fixed, but rather infer or refine segmentation together with the scene parameters. Bleyer et al (2010, 2011b) segment the scene into planar superpixels and estimate disparity by parameterizing their geometry. Additionally penalizing deviations from an initial solution, segment-based stereo is also promoted by Yamaguchi et al (2012). More recently, this method was extended to epipolar flow (Yamaguchi et al, 2013) and epipolar scene flow (Yamaguchi et al, 2014), both assuming that the flow fulfills epipolar geometry constraints, *i.e.* is the result of pure camera ego-motion. General 2D optical flow is computed by Unger et al (2012), who parameterize the motion of each segment with 2D affine transformations, and also allow for occlusion handling. Aside from estimating 2D and not 3D motion, the method differs in the sense that no inter-patch regularization is performed, such that motion fields of adjacent segments are estimated completely independently of one another. Ju et al (1996) integrate segmentation and motion estimation in a different manner. Starting from regularly spaced, non-overlapping regions the optical flow in each region is described by multiple affine motions, *motion layers*. Similarly to our proposed spatial regularization, inter-segment constraints are imposed, here between layers across segment boundaries. Because motion layers can interact and thus influence their parametric representation, a segmentation into regions with similar motion is obtained implicitly.

Murray and Buxton (1987) were among the first to perform motion estimation over multiple frames. The admissible 2D optical flow fields are, however, limited to only small displacements. Black and Anandan (1991) instead encourage the similarity between the current and the past flow estimates, extrapolating motion fields from previous frames. While allowing for larger displacements, information is only processed in a feed-forward fashion, in particular the present cannot influence the past. Much later, assuming a constant 2D motion field, Werlberger et al (2009) jointly reason over three consecutive frames. By considering constant 3D scene flow over time, we are able to address more general scenes. This constant velocity constraint is relaxed by Volz et al (2011), who encourage first and second order smoothness of the motion field as soft constraints. The motion is parameterized w.r.t. a single reference frame, thus reasoning about occlusion regions or outliers appears hard to achieve. Irani (2002) operates on much longer time intervals and enforces the estimated 2D motion trajectories to lie in a (rigid) subspace. Similarly, Garg et al (2013) require the 2D motions to lie in a low-rank trajectory space, but instead can use the prior as a soft constraint. Sun et al (2010, 2013) argue that the scene structure is more likely to persist over time than any motion pattern, hence avoid temporal smoothing at all, and instead jointly estimate the flow together with a segmentation into a small number of layers while requiring the pixel-to-layer membership to be constant. With the primary goal of high-level motion segmentation, Schoenemann and Cremers (2008) operate in a similar way: A video is segmented into several motion layers with long-term temporal consistency. Optionally, a 2D parametric motion for each layer is estimated as well. Our view-consistent formulation makes a related assumption, since we group pixels into planar and rigidly moving segments, while enforcing consistency of the segmentation over multiple frames. In contrast to motion layers, this much more fine-grained representation with hundreds of small segments enables us to address a wider range of scenes.

An explicit representation of 3D motion and shape allows scene flow methods to exploit temporal consistency over longer time intervals in a more straightforward manner, since smoothness constraints are better supported in the 3D scene than in its 2D projection. Rabe et al (2010) take advantage of this fact and propagate geometry and 3D motion across frames with the help of a Kalman filter. At each pixel the measurement vector for the filter is composed of scene flow vectors from the current and the previous frame, which are estimated with the method of Wedel et al (2008). Compared to its input, the filtered 3D motion and geometry contains significantly fewer outliers. Hung et al (2013) concatenate frame-to-frame stereo and flow to longer motion trajectories, which are, after passing several plausibility tests, included into the final optimization as soft constraints, similar to including feature matches in two-frame optical flow (Brox and Malik, 2011). The method advocates to propagate information through the whole sequence and, therefore, cannot output the scene flow without significant temporal delay, as is needed for several application scenarios. In their multi-camera setup Park et al (2012) also operate sequentially. Scene flow is first estimated frame-by-

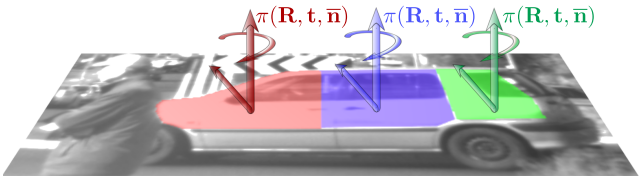


Fig. 3 Schematic sketch of our scene representation: The scene is modeled as a collection of rigidly moving planar segments, here three different segments cover the side of a car.

frame and then smoothed over time by tensor voting. Courchay et al (2009) go further and represent the scene with an explicit deformable 3D mesh template, which is fitted to the video data from multiple cameras over 10–60 frames. The method is theoretically elegant, but computationally expensive. Both approaches target motion capture in controlled settings.

Techniques that avoid an arbitrary reference frame and instead treat all views equally are predominantly used in stereo. The simplest form is the widespread left-right consistency check (*e.g.*, Hirschmüller, 2008) during post-processing. More recently, consistency tests were directly incorporated in the objective (Bleyer et al, 2011b). In our view-consistent formulation, we extend the latter strategy to scene flow, considering consistency across all images within a temporal window.

Introduced by Lempitsky et al (2008) for the case of 2D optical flow, fusion of different proposal sets has become a standard optimization technique. Here we employ such a scheme for the estimation of 3D scene flow.

3 Piecewise Rigid Model for 3D Scene Flow

To estimate 3D scene flow, we describe the dynamic scene as a collection of piecewise planar regions moving rigidly over time (Fig. 3). The motion and geometry of each region is governed by nine degrees of freedom, which we determine by minimizing a single objective function. During optimization, pixels are grouped into superpixels, and a suitable 3D plane and rigid motion is selected for each of these segments. Note that the implicitly obtained spatial segmentation does not aim to decompose the scene into semantic objects. Rather, an over-segmentation is desired to capture geometry and motion discontinuities, and to allow for the accurate recovery of non-planar and articulated objects. We begin our detailed description with the basic parameterization of the scene w.r.t. a single reference view and consider two time steps (Sec. 4). Later, we show how to achieve view-consistent scene flow over multiple frames (Sec. 5).

3.1 Preliminaries and notation

We formalize our model for the classical case of images obtained by a calibrated stereo rig at two subsequent time steps. However, we note that an extension to a larger number of simultaneous views is straightforward. To distinguish between the different views, we use subscripts l,r to identify the *left* and *right* camera[1], and superscripts $t \in T = \{-1,0,1,\ldots\}$ to indicate the acquisition time. We let the left camera at time $t=0$ define a common coordinate system and refer to it as the *canonical* view; this simplifies the notation. This canonical view, on one hand, serves as an evaluation basis, and on the other hand, coincides with the sole *reference* view, in case view consistency is not employed. These choices lead to the projection matrices $(\mathbf{K}|\mathbf{0})$ for the left and $(\mathbf{M}|\mathbf{m})$ for the right camera. For simplicity, we assume *w.l.o.g.* the calibration matrix $\mathbf{K}$ to be identical for both cameras.

In our model a 3D moving plane $\pi \equiv \pi(\mathbf{R},\mathbf{t},\bar{\mathbf{n}})$ is governed by nine parameters, composed of a rotation matrix $\mathbf{R}$, a translation vector $\mathbf{t}$, and a scaled normal $\bar{\mathbf{n}}$, each with three degrees of freedom. Note that we do not explicitly distinguish between camera ego-motion and independent object motion, but describe the full motion in one forward time step. Later, when we extend our model to reason over multiple frames, we show how to cope with high frequent ego-motion of the camera (Sec. 5.3). In case of a single reference view, we assume all planes to be visible in the canonical view. Thus, as the canonical camera center and coordinate origin coincide, no visible plane can pass the origin. We can then define the scaled normal $\bar{\mathbf{n}} \equiv \bar{\mathbf{n}}_l^0$ via the plane equation $\mathbf{x}^\mathsf{T}\bar{\mathbf{n}} = 1$, which holds for all 3D points $\mathbf{x}$ on the plane. Throughout the paper it is convenient to transfer the moving plane also into other views and their respective camera coordinate systems. The plane equation still has to be valid after any rigid transformation, hence the scaled normal transforms in correspondence with 3D points $\mathbf{x}$ on the plane $\bar{\mathbf{n}}_l^0$. For example, for the left camera at time step $t=1$ the normal $\bar{\mathbf{n}}_l^1$ in the respective coordinate system is found as:

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

We can, furthermore, determine the depth d observed at a pixel $\mathbf{p}$ of the image I_v^t, acquired at time t w.r.t. the center of camera v through the inverse scalar product:

$$d(\mathbf{p},\bar{\mathbf{n}}_v^t(\pi)) = \langle \mathbf{K}^{-1}\mathbf{p}, \bar{\mathbf{n}}_v^t(\pi)\rangle^{-1}. \quad (2)$$

This information is later needed to test for occlusions (Sec. 4.7), as well as to check the geometric consistency (Sec. 5.2) of the representation.

[1] "Left" and "right" are only used for intuition and do not necessarily correspond to the geometric configuration of the rig.

Utilizing a planar scene representation allows to map pixel locations conveniently to their corresponding positions from one view to another. In particular, a moving plane π induces homographies from the canonical view I_l^0 to the other views given by:

$$ {}^0_l\mathbf{H}^0_r(\pi) = (\mathbf{M} - \mathbf{m}\bar{\mathbf{n}}^\mathsf{T})\mathbf{K}^{-1} \tag{3a} $$

$$ {}^0_l\mathbf{H}^1_l(\pi) = \mathbf{K}(\mathbf{R} - \mathbf{t}\bar{\mathbf{n}}^\mathsf{T})\mathbf{K}^{-1} \tag{3b} $$

$$ {}^0_l\mathbf{H}^1_r(\pi) = (\mathbf{MR} - (\mathbf{Mt} + \mathbf{m})\bar{\mathbf{n}}^\mathsf{T})\mathbf{K}^{-1}. \tag{3c} $$

Concatenating the transformations above, mappings between arbitrary view pairs can be obtained. This is achieved by first transforming back to the canonical view and then into the desired frame, *e.g.* ${}^1_l\mathbf{H}^1_r(\pi) = {}^0_l\mathbf{H}^1_r(\pi) \cdot {}^0_l\mathbf{H}^1_l(\pi)^{-1}$. For notational convenience we define ${}^0_l\mathbf{H}^0_l(\pi)$ to be the identity, which maps the canonical frame onto itself.

4 Single Reference View

For now our aim is to determine depth and 3D motion for every pixel of the designated reference view I_l^0. To that end, we formally define an energy function $E(\mathscr{P},\mathscr{S})$ over two mappings: a mapping $\mathscr{S} : I_l^0 \to S$ that assigns each pixel of the reference view $\mathbf{p} \in I_l^0$ to a segment $s \in S$; and a mapping $\mathscr{P} : S \to \Pi$ to select a 3D moving plane $\pi \in \Pi$ from a predefined set of proposals Π for each of the segments $s \in S$. To find these mappings, we aim to minimize a single energy consisting of four terms:

$$ \begin{aligned} E(\mathscr{P},\mathscr{S}) = {} & E_D(\mathscr{P},\mathscr{S}) + \lambda E_R(\mathscr{P},\mathscr{S}) \\ & + \mu E_S(\mathscr{S}) + E_V(\mathscr{P},\mathscr{S}). \end{aligned} \tag{4} $$

The data term E_D measures photo-consistency across the four views of our basic model. The regularization term E_R encourages (piecewise) smoothness of geometry and motion at segment boundaries. The boundary term E_S evaluates the quality of the spatial segmentation, encouraging a compact and edge-preserving over-segmentation of the reference image. The visibility term E_V deals with missing correspondences from areas that move out of the viewing frustum (out of bounds). The energy is then minimized in two steps: Starting with a fixed initial over-segmentation $\mathscr{S}$, we establish the link between segments and 3D moving planes, labeling each segment $s \in S$ to belong to one of the moving planes $\pi \in \Pi$. Subsequently, we operate with a fixed mapping $\mathscr{P}$ and re-assign each pixel $\mathbf{p} \in I_l^0$ to one of the segments and, thereby, associated 3D moving planes. Note that the basic form of the energy remains the same when considering view consistency in Sec. 5.

4.1 Data term

In its traditional role, the data term embodies the assumption that corresponding points in different views have similar appearance. Here, we achieve this through four constraints per pixel, two for the stereo pairs at time steps 0 and 1, and two optical flow constraints, one for each camera (see Fig. 4, left). Denoting the 3D moving plane at a pixel $\mathbf{p}$ as $\pi_\mathbf{p} = \mathscr{P}(\mathscr{S}(\mathbf{p}))$ and utilizing the homographies defined in Eq. (3), we can define stereo data terms between the cameras as

$$ D^\mathrm{s}_t = \sum_{\mathbf{p}\in I_l^0} \rho\left({}^0_l\mathbf{H}^t_l(\pi_\mathbf{p})\mathbf{p}, {}^0_l\mathbf{H}^t_r(\pi_\mathbf{p})\mathbf{p}\right), \;\; t \in \{0,1\}, \tag{5} $$

and optical flow data terms across time as

$$ D^\mathrm{f}_i = \sum_{\mathbf{p}\in I_l^0} \rho\left({}^0_l\mathbf{H}^0_i(\pi_\mathbf{p})\mathbf{p}, {}^0_l\mathbf{H}^1_i(\pi_\mathbf{p})\mathbf{p}\right), \;\; i \in \{l,r\}. \tag{6} $$

The corresponding pixel location in a different view is usually a sub-pixel coordinate, hence image intensities are obtained via bilinear interpolation. For increased robustness in general conditions (*e.g.*, outdoors), we utilize the census transform $\rho = \rho_C$ (Zabih and Woodfill, 1994) over a 7×7 neighborhood to assess photo-consistency. We scale the Hamming distances by 1/30. Although we are not limited to this specific choice, all examples and results are generated with the census data cost, unless explicitly stated otherwise. The complete data term is given as the sum of the four terms in Eqs. (5) and (6):

$$ E_D(\mathscr{P},\mathscr{S}) = D^\mathrm{s}_0 + D^\mathrm{s}_1 + D^\mathrm{f}_l + D^\mathrm{f}_r. \tag{7} $$

4.2 Spatial regularization of geometry and motion

In our scene representation, geometry and motion parameters are shared among all pixels within a segment, hence explicit regularization within a segment is not needed. We can thus focus on the segment boundaries. One important benefit over pixelwise regularizers (Basha et al, 2010; Vogel et al, 2011) is that our boundary regularizer does not have to be overly strong to significantly stabilize scene flow estimation. Moreover, it rather naturally deals with discontinuities, a key problem area of previous scene flow techniques (*e.g.*, Vogel et al, 2011). Since boundaries regularly occur within a single object due to the over-segmentation, our regularization term assumes piecewise smooth 3D geometry and motion.

We model shape and motion priors independently (given a segmentation), and define our regularizer $E_R(\mathscr{P},\mathscr{S})$ as the sum of a geometric term $E^G_R(\mathscr{P},\mathscr{S})$ and a term $E^M_R(\mathscr{P},\mathscr{S})$ to measure the regularity of the motion field.

For now assume that two adjacent pixels $\mathbf{p}$ and $\mathbf{q}$ are assigned to the moving planes $\pi_\mathbf{p} = \mathscr{P}(\mathscr{S}(\mathbf{p}))$ and $\pi_\mathbf{q} =$

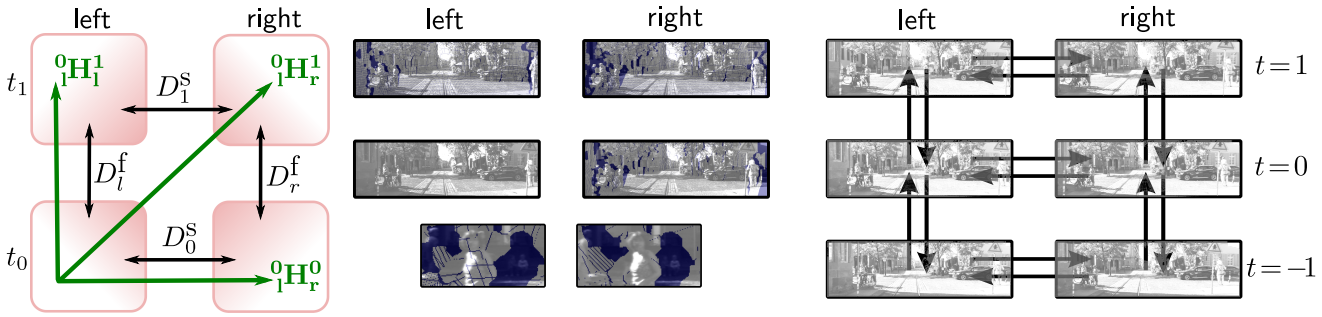


Fig. 4 *(left)* Single reference-view model. Data terms (black arrows) and homographies (green). *(center top)* Pixels without correspondences when using a reference view (blue). Areas that are hard to match may be without correspondence in other views; view-consistency avoids this. *(center bottom)* Enlarged areas containing pixels without correspondence in the right camera. *(right)* Data terms in the three-frame view-consistent model: Consistency is encouraged for spatial and direct temporal neighbors (black arrows). All pixels of all views are considered in the energy.

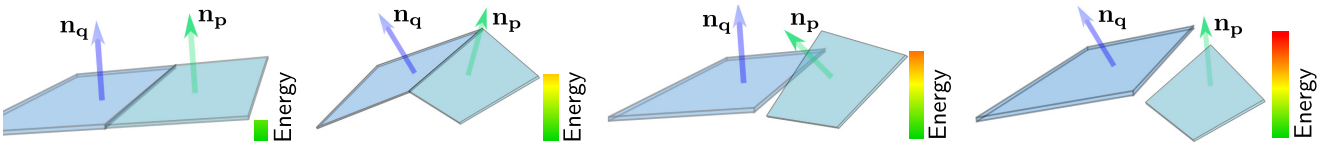


Fig. 5 Schematic sketch of the geometric part of our regularizer: Smoothly connected regions (*left*) are favored over bending (*center left*). The more the situation degenerates, the higher the energy becomes (*center right* and *right*).

$\mathscr{P}(\mathscr{S}(\mathbf{q}))$. We treat pixels as square patches, residing in the image plane in which they share a boundary. To measure the contribution to the regularization term along their common edge, we consider the (2D) endpoints of the edge between the pixels, c^1 and c^2. We begin with the geometry term. By projecting the endpoints onto each of the two 3D planes, we obtain the 3D endpoints $\mathbf{c}^1_\mathbf{p}$, $\mathbf{c}^1_\mathbf{q}$, $\mathbf{c}^2_\mathbf{p}$ and $\mathbf{c}^2_\mathbf{q}$ (see Fig. 6). In case $\mathbf{p}$ and $\mathbf{q}$ lie on different planes, the pixel boundaries will, in general, not coincide in 3D space. We thus compute distance vectors between the 3D endpoints: $\mathbf{d}_1 = \mathbf{c}^1_\mathbf{p} - \mathbf{c}^1_\mathbf{q}$ and $\mathbf{d}_2 = \mathbf{c}^2_\mathbf{p} - \mathbf{c}^2_\mathbf{q}$. Our goal is to penalize the distances along the shared edge. One could compute 3D distances for any point on the boundary in a similar fashion. However, since we are using planes as primitives, the 3D distance along the shared boundary in the image plane is simply a convex combination of the endpoint distances $||\alpha\mathbf{d}_1 + (1-\alpha)\mathbf{d}_2||$.

To consider surface curvature we exploit this observation further and shift the 3D endpoints along their respective plane normals $\mathbf{n_p}$ and $\mathbf{n_q}$ before measuring distances. We denote the difference of the normals as $\mathbf{d}_n = \mathbf{n_p} - \mathbf{n_q}$, and define a distance function (see Fig. 6)

$$f_\gamma(\alpha,\beta) = ||\alpha(\mathbf{d}_1 + \gamma\beta\mathbf{d}_n) + (1-\alpha)(\mathbf{d}_2 + \gamma\beta\mathbf{d}_n)||. \tag{8}$$

The weight γ balances boundary distance *vs.* curvature. The geometry regularizer is then found by integration. Adding a factor $3/2$ for mathematical convenience, we integrate the squared distance function $(f_\gamma)^2$ along the boundary (w.r.t. α) and along the normal direction (w.r.t. β) in closed form:

$$\begin{aligned} E^G_R(\mathscr{P},\mathscr{S}) &= \sum_{(\mathbf{p},\mathbf{q})\in\mathscr{N}} w_{\mathbf{p},\mathbf{q}}\,\psi\left(\frac{3}{2}\int_0^1\int_{-1}^1 f_\gamma(\alpha,\beta)^2\,\mathrm{d}\beta\,\mathrm{d}\alpha\right) \\ &= \sum_{(\mathbf{p},\mathbf{q})\in\mathscr{N}} w_{\mathbf{p},\mathbf{q}}\,\psi\left(||\mathbf{d}_1||^2 + ||\mathbf{d}_2||^2 + \langle\mathbf{d}_1,\mathbf{d}_2\rangle + \gamma^2||\mathbf{d}_n||^2\right). \end{aligned} \tag{9}$$

The summation considers pixels to be adjacent in an (8-) neighborhood $\mathscr{N}$, where the length of the common edge is taken into account through the weight $w_{\mathbf{p},\mathbf{q}}$, which can optionally also incorporate edge information (Eq. 13) of the image data. $\psi(\cdot)$ denotes a (robust) penalty function. The intuition behind this form of regularization is shown in Fig. 5. Setting $\gamma := 1$ our energy favors planar configurations over bending. By integrating squared distances of 3D vectors, the induced penalty increases smoothly as the situation degenerates. This soft transition helps in the realistic case of a limited proposal set of 3D moving planes Π.

The motion regularizer is obtained by first applying the rigid transformation to the moving planes. We then similarly integrate the endpoint distances $\mathbf{d}^M_i = \mathbf{R_p}\mathbf{c}^i_\mathbf{p} + \mathbf{t_p} - \mathbf{c}^i_\mathbf{p} - (\mathbf{R_q}\mathbf{c}^i_\mathbf{q} + \mathbf{t_q} - \mathbf{c}^i_\mathbf{q})$, as well as the differences between the (rotated) normals $\mathbf{d}^M_n = (\mathbf{R_p}\mathbf{n_p} - \mathbf{n_p}) - (\mathbf{R_q}\mathbf{n_q} - \mathbf{n_q})$, leading to

$$\begin{aligned} &E^M_R(\mathscr{P},\mathscr{S}) = \\ &\sum_{(\mathbf{p},\mathbf{q})\in\mathscr{N}} w_{\mathbf{p},\mathbf{q}}\,\psi\left(||\mathbf{d}^M_1||^2 + ||\mathbf{d}^M_2||^2 + \langle\mathbf{d}^M_1,\mathbf{d}^M_2\rangle + \gamma^2||\mathbf{d}^M_n||^2\right). \end{aligned} \tag{10}$$

In both cases, robustness to discontinuities is achieved by employing truncated penalties $\psi(y) = \min(\sqrt{y},\eta)$ (with thresholds η_G, η_M).

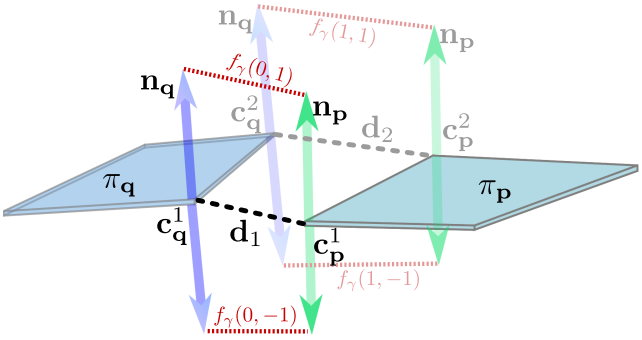


Fig. 6 Illustration of the regularization scheme: The bilinear distance function f_γ considers geometric distance and curvature. Integrating the squared distances along the shared edge as well as along an extrusion of the normals leads to a closed form expression.

The proposed regularization scheme is not limited to 3D. For instance, the endpoint distances can be replaced by 2D entities such as the disparity difference, the difference between optical flow vectors, and the change of disparity over time. This is a popular choice for scene flow (Huguet and Devernay, 2007; Valgaerts et al, 2010) and (optionally) used here. Note, however, that falling back to 2D regularization can only yield a (close) approximation of the true 3D penalties, as projective foreshortening is not considered.

When reasoning at the segment level, we can approximate the regularizers by computing the penalties directly from the endpoints of the segments. By precomputing the length of the boundary (summing the edge weights along the shared border), the evaluation of the regularizer becomes much more efficient. Because superpixels in our framework are near-convex, the overall accuracy of the algorithm is barely affected (Fig. 12, right).

4.3 Spatial regularization of the segmentation

Data term and spatial regularization operate not only on the segment-to-plane mapping $\mathscr{P}$, but also depend on the assignment of pixels to segments $\mathscr{S}$, which in our experience can lead to rather fragmented over-segmentations. To counteract this behavior and to incorporate prior knowledge that segments should be spatially coherent (but not necessarily connected) and preserve image edges, we add an additional regularization term, assessing the quality of the underlying segmentation:

$$E_S(\mathscr{S}) = \sum_{\substack{(\mathbf{p},\mathbf{q})\in\mathscr{N}(I_l^0),\\ \mathscr{S}(\mathbf{p})\neq\mathscr{S}(\mathbf{q})}} u_{\mathbf{p},\mathbf{q}} + \sum_{\mathbf{p}\in I_l^0} \begin{cases} 0, & \exists \mathbf{e}\in\mathscr{E}(s_i): \|\mathbf{e}-\mathbf{p}\|_\infty < N_S \\ \infty, & \text{else.} \end{cases} \tag{11}$$

The first term resembles a contrast sensitive pairwise Potts model, again evaluated over the (8-)neighborhood $\mathscr{N}$ of a pixel. Here, the weight $u_{\mathbf{p},\mathbf{q}}$ allows to take into account the image structure and the length of the edge between the pixels. To define these weights we follow Werlberger et al (2009) and apply the anisotropic diffusion tensor:

$$D^{\frac{1}{2}} = \exp(-\alpha|\nabla I|)gg^\mathsf{T} + g^\perp(g^\perp)^\mathsf{T}. \tag{12}$$

The image gradient direction $g = \nabla I/|\nabla I|$ is determined via bicubic interpolation in the middle between $\mathbf{p}$ and $\mathbf{q}$. Assuming $I \in [0,1]$, we set $\alpha = 5$ and define the weight

$$u_{\mathbf{p},\mathbf{q}} := |D^{\frac{1}{2}}\overrightarrow{\mathbf{pq}}|. \tag{13}$$

The second term links a segment to its seed point $\mathbf{e} \in \mathscr{E}(s_i)$ in order to limit its maximum extent to a size smaller than $(2N_S-1)\times(2N_S-1)$ pixels. This strategy prevents the scene flow from becoming overly simplified, but more importantly also restricts the number of candidate segments for a pixel, thus reducing the time needed for optimizing the energy w.r.t. $\mathscr{S}$. We found that a good strategy to define the seed points is to reuse the center of the original superpixels. Here we set $N_S = 25$, but values between 10 and 30 pixels perform alike (see Sec. 6.1). Note that a similar strategy was proposed by Veksler et al (2010) to compute an over-segmentation of a single image.

4.4 Visibility term

So far we have not considered the problem of visibility, thus areas that fall *out of bounds*, *i.e.* are not visible in some of the images. Especially when dealing with large motions, these regions can cover a significant portion of the image. Configurations with no valid correspondence are not considered by the data term Eq. (7) and contribute 0 cost to the energy. Allowing for arbitrary moving planes in our model could, therefore, easily lead to a solution, where a significant portion of pixels is erroneously assigned a motion that moves them out of bounds. On the other hand, penalizing these kinds of configurations strongly could harm the results. Consider, for instance, a saturated region that actually moves out of bounds. A solution in which this region is mapped to a similarly saturated, but unrelated area in the other images lowers the data cost and would therefore be preferred. Since this regularly happens in challenging scenes, we address the problem as follows: Let us assume that we have access to an "oracle" V, which can predict whether a pixel will stay in the image or move *out of bounds*. Further, let V_l^1, V_r^0 and V_r^1 be the predicted binary visibility masks for all but the reference image (*out-of-bounds*: 0, pixel visible: 1), and let $\Gamma_i^j[\cdot]$ be a binary function that determines whether its argument lies within the boundaries of image I_i^j. We encourage the scene flow estimate to stay near that prediction, by defining

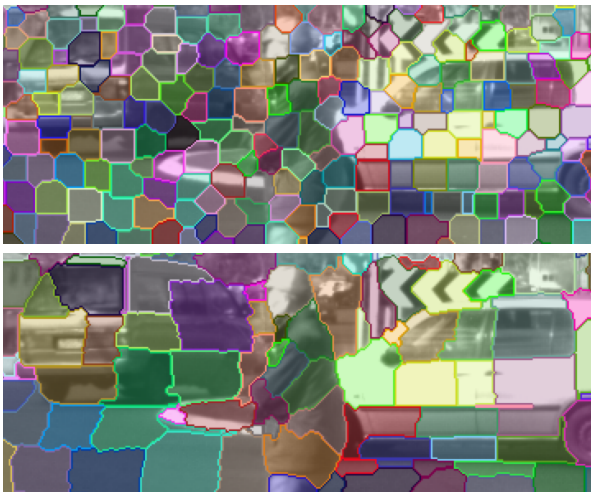


Fig. 7 Demonstration of the per-pixel refinement: *(top)* Initial superpixel segmentation. *(bottom)* Superpixel segmentation after optimization w.r.t. $\mathscr{S}$.

a visibility term that forms part of the energy in Eq. (4):

$$E_{\mathrm{V}}(\mathscr{P},\mathscr{S}) = \theta_{\mathrm{oob}} \sum_{\mathbf{p}\in I_l^0} \left|V_r^0(\mathbf{p}) - \Gamma_r^0\left[{}_l^0\mathbf{H}_r^0(\pi_\mathbf{p})\mathbf{p}\right]\right| + \tag{14}$$
$$\left|V_l^1(\mathbf{p}) - \Gamma_l^1\left[{}_l^0\mathbf{H}_l^1(\pi_\mathbf{p})\mathbf{p}\right]\right| + \left|V_r^1(\mathbf{p}) - \Gamma_r^1\left[{}_l^0\mathbf{H}_r^1(\pi_\mathbf{p})\mathbf{p}\right]\right|,$$

with $\theta_{\mathrm{oob}} := 0.5\max(\rho_C)$ set to half the maximal data cost. In practice, we found that common stereo and variational flow methods can predict pixels moving out-of-bounds sufficiently reliably, and consequently reuse the output of the 2D stereo and optical flow algorithms from the proposal generation step (Sec. 4.6). An alternative visibility predictor could be the ego-motion of the stereo camera system.

4.5 Approximate inference

Inference in our piecewise rigid model entails estimating the continuous 9-dimensional variables describing geometry and motion of each rigidly moving plane, and the discrete assignments of pixels to segments. By restricting the optimization to a finite set of proposal moving planes, the whole problem is transferred into a labeling problem in a discrete CRF. The benefit is two-fold: First, we can leverage robust discrete optimization techniques that cope well with complex energies, particularly here the fusion move framework of Lempitsky et al (2008, 2010). Second, occlusions are discrete events and can thus naturally be integrated in the objective (Sec. 4.7).

To bootstrap the process, we start with a fixed segmentation $\mathscr{S}$ and optimize the energy w.r.t. $\mathscr{P}$, selecting a suitable moving plane for each segment from the proposal set. To obtain the initial superpixel segmentation, we simply minimize the segmentation energy E_S alone, and subsequently split strongly non-convex segments. We alternatively tested a segmentation into regular grid cells. Interestingly, this simplistic initialization works almost as well (see Sec. 6.1). In either case, the seed points $\mathscr{E}$ are selected as the central pixels of the initial segments. When solving for $\mathscr{P}$ we need to consider the data, visibility, and regularization terms only. After we found a solution for $\mathscr{P}$, the mapping is kept fixed and the energy is optimized w.r.t. $\mathscr{S}$, reassigning the pixels to segments and, thereby, implicitly to moving planes (*c.f.* Fig. 7). Because the segment size is restricted to a maximal side length of N_S through Eq. (11), the pseudo-Boolean function (Lempitsky et al, 2008) representing the local energy has at most $(2N_S-1)^2$ variables, which makes the optimization efficient. Distant segments can even be expanded in parallel. We use a similar strategy when optimizing for $\mathscr{P}$: We locally restrict the validity of each moving plane proposal to cover only a certain expansion region in the scene. In practice, we found that a proposal should at least cover 100 of its closest neighboring segments and set the region size accordingly. This allows to test several proposals in parallel. Note that we can iterate the alternating optimization further, but observe no practical benefit.

General pseudo-Boolean energies are usually optimized with QPBO (Rother et al, 2007), which can also handle non-submodular energies, but does not guarantee a complete labeling when supermodular edges are present. One disadvantage compared to standard graph cuts, however, is that the instantiated graph has twice the number of nodes than the (pseudo-Boolean) energy has variables. For our (non-submodular) energy we can alternatively use the local submodular approximation proposed by Gorelick et al (2014). This has the advantage that conventional graph cuts can be used, which is usually faster than QPBO. We particularly use LSA-AUX, which for each α-expansion replaces pairwise supermodular potentials by a local plane approximation that bounds the true energy from above. This idea is very simple to implement and delivers a significantly better approximation than a simple truncation of non-submodular terms. We experimentally compare both approaches in Sec. 6.

4.6 Proposal generation

To perform inference over the 3D geometry and motion of the segments, we require an (initial) set of proposal planes together with their rigid motion. We can create these from either the output of other scene flow algorithms, or from a combination of stereo and optical flow methods. To convert the pixelwise correspondence information to our representation, we separately fit the parameters of a 3D plane and its rigid motion to each superpixel of the initial segmentation. Fitting is complicated by inaccuracies or noise in the stereo and flow estimates, and by superpixels that are not well-aligned with depth and motion discontinuities. We thus opt

for a robust procedure and minimize the transfer error integrated into a robust cost function, particularly the Lorentzian $\phi(x) = \log(1 + \frac{x}{2\sigma^2})$:

$$\sum_{\mathbf{p}\in s} \phi(||P({}^0_l\mathbf{H}^0_r(\bar{\mathbf{n}})\mathbf{p}) - \mathbf{p}'||^2) \quad \rightarrow \min_{\bar{\mathbf{n}}} \tag{15a}$$

$$\sum_{\mathbf{p}\in s} \phi(||P({}^0_l\mathbf{H}^1_l(\mathbf{R},\mathbf{t})\mathbf{p}) - \mathbf{p}'||^2) \quad \rightarrow \min_{\mathbf{R},\mathbf{t}}, \tag{15b}$$

where the dependence of the homographies on the parameters (the normal $\bar{\mathbf{n}}$ and rigid motion $(\mathbf{R},\mathbf{t})$) is made explicit, and P denotes the conventional projection operator. Each pixel $\mathbf{p}$ of segment $s \in S$ is matched to its 2D correspondence $\mathbf{p}'$, determined by the proposal algorithm. We parameterize the rotation in Eq. (15b) by its exponential map to define the derivatives, and use the previously determined scaled normal to derive the homography (*c.f.* Eq. 3). After bootstrapping this non-convex optimization problem with simple iterated least squares, two iterations of the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (LM-BFGS) suffice for our purposes. The quality of the fit is analyzed in Sec. 6.1. Note that since we are treating the estimation of 3D planes and rigid motions independently, the problem of fitting a rigid motion is similar to the computation of the ego-motion of a stereo camera system, such that algorithms for this problem could also be applied (*e.g.*, Badino and Kanade, 2011). Here, however, we only consider the motion of an individual segment and not of the complete stereo rig.

4.6.1 Additional proposals

The strategy of selecting parts of the solution from a set of proposals allows to include additional information in an unbiased way, without the need for altering the energy formulation. We exploit this property by including the estimated ego-motion of the stereo system as an additional proposal. The ego-motion is found by reusing our fitting procedure from above (Eq. 15b) on the segment centers and their correspondences, given by the output of our per-segment solution (obtained after optimizing w.r.t. the mapping $\mathscr{P}$). We then can fuse the current solution with the estimated ego-motion.

Additionally, we use a *local replacement* strategy, motivated by proposal instances for which depth and motion errors are not correlated. We posit that these largely result from the 2D proposal algorithms, which estimate motion and depth independently. We address this with additional proposals: We randomly select proposals and propagate a part of their state to other segments in a 2-neighborhood. This can either be the geometry or the rigid motion, which simply replaces the corresponding state of the neighbors. This procedure is iterated several (≈ 4000) times, leading to a combination of geometry and motion of neighboring segments. The strategy has similarities to the PatchMatch idea (Barnes et al, 2009; Bleyer et al, 2011a), as information is shared and distributed among neighboring segments.

4.7 Occlusion handling

The data term as defined in Eq. (7) assumes that every pixel is visible; no occlusion reasoning takes place. Given our 3D scene representation, we can explicitly reason about occlusions. Compared to stereo, the handling of occlusions for scene flow has the advantage of having two (or more *c.f.* Sec. 5.3) additional views of the scene. Accordingly, pixels that are occluded in a subset of views may still be visible in one of the view pairs.

To leverage this, occlusion handling is applied to all pairs of views for which a data term is formulated. We formalize this only for a single view pair, because the mathematical formulation is equivalent for each summand of the data term. We make use of the well-known principle (dating back at least to Kolmogorov and Zabih, 2001) of applying a constant penalty θ_{occ}, if a pixel is occluded in at least one of the two views of the pair. The penalty is chosen as $\theta_{\text{occ}} := \theta_{\text{oob}}$ (Eq. 14). Although occlusions and out-of-bound areas have different causes, the impact on the correspondence is the same: The pixel correspondence cannot be judged by the appearance, and hence the data costs of Eqs. (5) or (6) are invalid. Note that pixels that are assigned to the same moving plane in our scene representation naturally cannot occlude each other.

To simplify the exposition, we will not present our occlusion model in its most general form, but rather one instantiation within a single fusion/expansion move of the approximate inference procedure from Sec. 4.5. Hence, we are dealing with a binary optimization problem. Assuming a fixed segment-to-plane mapping $\mathscr{P}$, we will first investigate the update of the per-pixel segmentation $\mathscr{S}$. Differences in the update procedure when solving for $\mathscr{P}$ will be discussed later. *W.l.o.g.* let the binary state $x_{\mathbf{p}} = 0$ denote that the pixel $\mathbf{p}$ retains its current segment assignment and, accordingly, $x_{\mathbf{p}} = 1$ indicate a switch to the trial segment α. We begin by expressing the data term from Sec. 4.1 in the form of a pseudo-Boolean function:

$$D(\mathbf{x}) = \sum\nolimits_{\mathbf{p}\in I^0_l} \left(u^0_{\mathbf{p}}(1 - x_{\mathbf{p}}) + u^1_{\mathbf{p}} x_{\mathbf{p}} \right), \tag{16}$$

where the vector $\mathbf{x}$ denotes all binary pixel assignments. The data penalty equals $u^0_{\mathbf{p}}$ if $\mathbf{p}$ remains in its current segment, and $u^1_{\mathbf{p}}$ if $\mathbf{p}$ is assigned to segment α.

Whether a pixel $\mathbf{p}$ is occluded or not depends both on its binary segment assignment $x_{\mathbf{p}}$, and on whether there is any other pixel $\mathbf{q}$ (or possibly multiple pixels) that occludes $\mathbf{p}$. Determining whether $\mathbf{q}$ triggers an occlusion in turn depends on its segment assignment $x_{\mathbf{q}}$. With $\mathscr{O}^i_{\mathbf{p}}$ we identify the set of all pixel-assignment pairs $(\mathbf{q}, j)$, for which pixel $\mathbf{q}$ occludes pixel $\mathbf{p}$ if $x_{\mathbf{p}} = i$ and $x_{\mathbf{q}} = j$. Now we can replace Eq. (16)

with our occlusion-aware data term

$$D_{\mathrm{O}}(\mathbf{x}) = \sum_{\mathbf{p}\in I_l^0} \Big(\theta_{\mathrm{occ}} + \sum_{i=0}^{1} \hat{u}_{\mathbf{p}}^i [x_{\mathbf{p}} = i] \prod_{(\mathbf{q},j)\in\mathscr{O}_{\mathbf{p}}^i} [x_{\mathbf{q}} \neq j] \Big). \quad (17)$$

Here, we denote the difference of the (unoccluded) data penalty and the occlusion cost θ_{occ} by $\hat{u}_{\mathbf{p}}^i = u_{\mathbf{p}}^i - \theta_{\mathrm{occ}}$, and with $[\cdot]$ the Iverson bracket. To facilitate a better understanding of the equation above, let us focus on a single pixel $\mathbf{p}$. The respective summand becomes $\hat{u}_{\mathbf{p}}^0$, if both $x_{\mathbf{p}} = 0$ *and* the product equals to 1. The latter happens if no occlusion occurs, that is either all possibly occluding pixels $\mathbf{q}$ are assigned to a segment $x_{\mathbf{q}}$ in which they do not lead to an occlusion, or the set $\mathscr{O}_{\mathbf{p}}^0$ is empty, meaning that no pixel exists that could possibly occlude $\mathbf{p}$. The data cost overall thus equals θ_{occ} in case of an occlusion, and the standard data penalty $u_{\mathbf{p}}^0$ or $u_{\mathbf{p}}^1$, otherwise.

Recall that we establish the segment-to-plane mapping $\mathscr{P}$ by reasoning over entire segments (see Sec. 4.5). Therefore, we directly extend the occlusion model to the segment level. The potentials of the respective pseudo-Boolean energies in Eqs. (16) and (17) look the same, but with variables representing segments instead of pixels. We consider a segment to be (significantly) occluded if its central pixel is occluded. Because our segments are compact and similarly sized, at least one quarter of a segment has to be occluded by a different region to render the central pixel occluded. To check for occlusions we employ conventional z-buffering, utilizing Eq. (2) to compute the depth at each pixel.

Depending on the number of possibly occluding pixels, the (per-pixel) penalty may be a higher-order pseudo-Boolean function ($|\mathscr{O}_{\mathbf{p}}^i| > 1$). Optimization techniques based on graph cuts, including QPBO, can only be applied to quadratic polynomials, which is why all higher-order terms have to be reduced to pairwise ones. Over the years several reduction techniques have been proposed (*e.g.*, Ali et al, 2008; Ishikawa, 2009; Rother et al, 2009). Each applies a certain transformation that approaches the reduction independently for each higher order summand of the energy. We refrain from presenting these exhaustive details at this point and instead refer to the Appendix A.

5 View-Consistent Model

Equipped with our basic representation and model from Sec. 4, we now generalize it to estimate scene flow for all views and time instants simultaneously. A major benefit compared to using a single reference view is that the entire image evidence of all views has to be explained. This results in a more robust estimate, which is less prone to common imaging artifacts. Occlusion handling can be improved as well. Another benefit is that significantly fewer non-submodular edges occur in the pseudo-Boolean function constructed during the optimization process. We defer details to the experimental evaluation. To enable a view-consistent model, we first need to extend the notion of the segmentation to all views, with the challenge of generating a consistent segmentation of the scene across views and time. An obvious downside of a view-consistent approach is a significantly enlarged set of unknowns, since the assignments from segments to moving planes and pixels to segments have to be computed for each involved view.

After establishing the concept of view-consistency, we aim to estimate scene flow for more than two time steps. We thus extend the idea of rigidity by assuming constant translational and rotational velocity of the 3D moving planes. Note that due to the short time intervals considered, this assumption is valid for many application scenarios. In the following, we start our description for only two time steps, and later explain how to extend our model to multiple frames in time.

5.1 Model overview

As before we strive to determine depth and a 3D motion vector for every pixel, but this time for all the views examined. We thus keep track of a superpixel segmentation in every view, denoted as S_v^t, the set of segments in the image I_v^t in view v at time step t. The energy definition (Eq. 4) is extended to be a function of two *sets* of mappings. The first set of mappings $\mathscr{S} = \{\mathscr{S}_v^t : t, v\}$ with $\mathscr{S}_v^t : I_v^t \to S_v^t$ assigns each pixel of frame I_v^t to a segment of S_v^t. With the second set $\mathscr{P} = \{\mathscr{P}_v^t : t, v\}$, a rigidly moving plane is selected for each segment in each view: $\mathscr{P}_v^t : S_v^t \to \Pi$. Recall that Π denotes a candidate set of possible 3D moving planes. The formal definition of the energy takes the same basic form as Eq. (4):

$$E^{\mathrm{VC}}(\mathscr{P},\mathscr{S}) = E_D^{\mathrm{VC}}(\mathscr{P},\mathscr{S}) + \lambda E_R^{\mathrm{VC}}(\mathscr{P},\mathscr{S}) + \mu E_S^{\mathrm{VC}}(\mathscr{S}). \quad (18)$$

However, in our view-consistent setting the definition of the data term E_D^{VC} is significantly different, as not only photo-consistency w.r.t. a reference view is considered, but also the consistency of the underlying geometric configuration and segmentation of the scene. The regularization term E_R^{VC} and the segmentation term E_S^{VC} are straightforward extensions of their single view counterpart from Sec. 4. In our experience, by explaining the available evidence from all images, this view-consistent formulation does not require an explicit visibility term (Sec. 4.4).

The spatial smoothness assumption is extended to all views, simply summing the contributions of motion (Eq. 9) and geometry (Eq. 10) terms per frame:

$$E_R^{\mathrm{VC}}(\mathscr{P},\mathscr{S}) = \sum_{t\in T} \sum_{v\in\{l,r\}} E_R^G(\mathscr{P}_v^t,\mathscr{S}_v^t) + E_R^M(\mathscr{P}_v^t,\mathscr{S}_v^t). \quad (19)$$

In a similar fashion we extend the regularization of the segmentation (Eq. 11) to all considered views:

$$E_S^{\mathrm{VC}}(\mathscr{S}) = \left(\sum_{\substack{t\in T,\\ v\in\{l,r\}}} \sum_{\substack{(\mathbf{p},\mathbf{q})\in\mathscr{N}(I_v^t),\\ \mathscr{S}(\mathbf{p})\neq\mathscr{S}(\mathbf{q})}} u_{\mathbf{p},\mathbf{q}} \right) + \sum_{\mathbf{p}\in I_l^0} \begin{cases} 0, & \exists \mathbf{e}\in\mathscr{E}(s_i):\|\mathbf{e}-\mathbf{p}\|_\infty<N_S \\ \infty, & \text{else,} \end{cases} \tag{20}$$

where $\mathscr{N}$ is again defined as the 8-neighborhood. Note that the second term is only applied to the canonical view, such that the maximal size of a segment is only restricted in the canonical frame. Also note that we treat the segmentations of the different frames independently. However we encourage the segmentation to be consistent across views (*c.f.* Fig. 11) such that the restriction on the maximal segment size is also propagated to all other images[2].Consistency between the superpixel segmentations is encouraged in the data term, described in the following.

5.2 View-consistent data term

In our view-consistent model we explicitly store a description of the scene in terms of moving planes as observed in each of the views. To exploit the redundancy in this representation, we check the consistency of the scene flow estimate in each view with its direct neighbors in time, as well as with the other views at the same time instant (Fig. 4, right). We here slightly abuse the term consistency: In its classical sense we check for photo-consistency of the images at corresponding pixel locations, determined through their assigned moving planes $\pi \equiv \pi(\mathbf{R},\mathbf{t},\overline{\mathbf{n}})$. However, in our novel scene representation we can also check the geometric configuration for plausibility, test for occlusions, and verify the consistency of the segmentation. This is done by comparing depth values induced by the respective moving plane (Eq. 2), based on the underlying image segmentation (see Fig. 8).

Now let us assume we want to check the 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 $\hat{t}$. We denote the 3D moving plane of the pixel $\mathbf{p}$ by $\pi_{\mathbf{p}} = \mathscr{P}_v^t(\mathscr{S}_v^t(\mathbf{p}))$. The related homography allows to determine the corresponding pixel location in the other view, $\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}} = {}_v^t\mathbf{H}_{\hat{v}}^{\hat{t}}(\pi_{\mathbf{p}})\mathbf{p}$, and the depth function $d(\mathbf{p},\overline{\mathbf{n}}_v^t(\pi))$ from Eq. (2) enables evaluating the geometric configuration at that pixel. The data term for a single pixel $\mathbf{p}$ in view v at time-step t assigned to the moving plane $\pi_{\mathbf{p}}$ with the adjacent view $\hat{v}$ at time-step $\hat{t}$ is then given by

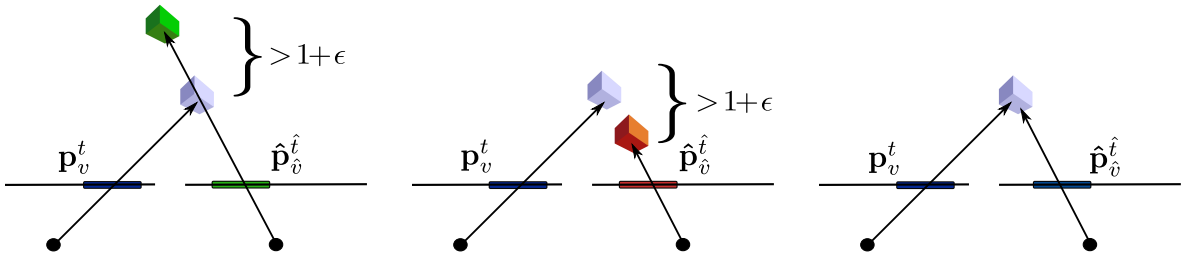
$$\rho(\mathbf{p},\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}) = \begin{cases} \theta_{\mathrm{imp}} & \text{if} \quad \frac{d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}},\overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}))}{d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}},\overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_{\mathbf{p}}))} > 1+\varepsilon \\ \theta_{\mathrm{occ}} & \text{if} \quad \frac{d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}},\overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_{\mathbf{p}}))}{d(\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}},\overline{\mathbf{n}}_{\hat{v}}^{\hat{t}}(\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}))} > 1+\varepsilon \\ \theta_{\mathrm{oob}} & \text{otherwise if} \quad \hat{\mathbf{p}}_{\hat{v}}^{\hat{t}} \notin I_{\hat{v}}^{\hat{t}} \\ \rho_C(\mathbf{p},\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}) + \theta_{\mathrm{mvp}} & \text{otherwise if} \quad \pi_{\mathbf{p}} \neq \pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}} \\ \rho_C(\mathbf{p},\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}) & \text{otherwise.} \end{cases} \tag{21}$$

The first two cases are depicted in Fig. 8a and b. Here the relative difference in depth is used to distinguish between *implausible* and *occlusion* cases. This distinction is similar to comparing disparity values for the stereo case (Bleyer et al, 2011b). The first case (Fig. 8a) describes a geometrically implausible situation, in which the depth of the moving plane $\pi_{\mathbf{p}}$, observed from the 2nd camera in pixel $\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}$, is smaller than the depth assigned to the pixel in that 2nd view. In this situation the 3D point on the plane $\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}$ would be occluded by the moving plane $\pi_{\mathbf{p}}$ and not be visible by the 2nd camera. We apply a fixed penalty θ_{imp} in this case. In the second case (see Fig. 8b), the depth of the moving plane $\pi_{\mathbf{p}}$ is greater than that of the corresponding plane $\pi_{\hat{\mathbf{p}}_{\hat{v}}^{\hat{t}}}$ and, therefore, the pixel $\mathbf{p}$ is occluded in the second view. Again, a fixed penalty θ_{occ} is applied. This concept of occlusion reasoning via cross checking the current solution among views is only possible by simultaneously estimating a solution for all views and rather different from the occlusion detection technique presented in Sec. 4.7 for a single reference view. An additional benefit is that the resulting energy function induces only pairwise edges. In Eq. (17), in contrast, multiple possible labels for the corresponding location in the other view may exist, which in turn leads to higher-order terms in the respective pseudo-Boolean energy. In our experience the view-consistent formulation leads to fewer supermodular edges in the optimization (see Sec. 6.2), resulting in a simpler optimization problem.

Since the set of proposal planes is limited due to practical considerations, we cannot assume that our representation always assigns a fully accurate depth for every pixel. Instead of strictly comparing relative depth values we, therefore, opt for a relaxed test by including the ε parameter, empirically set to $\varepsilon := 0.015$. This additionally alleviates aliasing artifacts introduced by the finite resolution of the pixel grid.

The third case penalizes pixels moving out of the viewing frustum (*out of bounds*) with a fixed penalty θ_{oob}. By employing view consistency, the solution has to respect the information from all views of the scene. Hence the treatment of this event can be a lot simpler than in the case of a single reference frame, where an additional visibility term (Sec. 4.4) was included.

[2] this property is further exploited in the inference procedure



(a) Implausible case **(b)** Occlusion **(c)** Normal case

Fig. 8 Illustration of the per-pixel view-consistent data term (see text for more details).

When pixels are in geometric correspondence we apply the usual census data penalty $\rho = \rho_C$ to measure photo-consistency (*c.f.* Sec. 4.1). In (Vogel et al, 2014) we originally proposed to additionally truncate the data term at half the maximal possible cost at a pixel ($0.5\max(\rho_C)$). An investigation of this particular choice shows that the number of resulting non-submodular terms in the optimization is reduced (Sec. 5.4), however some of the information is lost, which can lead to a decreased accuracy. Consequently, we avoid the truncation here.

If the pixels are in geometric correspondence, but belong to different moving planes, we assert a *moving plane violation* and impose an additional penalty θ_{mvp}. This leads to the desired view-consistent segmentation, as pixels are encouraged to pick the same 3D moving plane in neighboring views.

In practice, it appears prudent to penalize pixels without correspondence equally, thus we set both penalties for occlusions and pixels moving out of bounds to $\theta_{\text{oob}} = \theta_{\text{occ}} = 0.5\max(\rho_C)$. Aliasing again prevents us from penalizing implausible configurations with an infinite penalty; instead we set $\theta_{\text{imp}} := \max(\rho_C)$, which also prevents deadlocks in the optimization. While this can lead to a few implausible assignments in the final estimate, the overall error is reduced. For the same reasons we allow for deviations from our consistency assumption for the segmentation and empirically set $\theta_{\text{mvp}} := 5/16\ \theta_{\text{oob}}$.

All views are treated equally in our model, thus the per-pixel contribution from Eq. (21) is summed over all pixels of all frames. Our data term consists of the summed data costs for all stereo pairs and frames that are direct neighbors in time (Fig. 4, right):

$$E_D^{\text{VC}}(\mathscr{P},\mathscr{S}) = \tag{22}$$
$$\sum_{t\in T}\sum_{v\in\{l,r\}}\sum_{\mathbf{p}\in I_v^t}\left(\sum_{\hat{v}\neq v}\rho(\mathbf{p},\hat{\mathbf{p}}_{\hat{v}}^t)+\sum_{\substack{\hat{t}\in T\\ |\hat{t}-t|=1}}\rho(\mathbf{p},\hat{\mathbf{p}}_v^{\hat{t}})\right).$$

In contrast to the reference-view formulation (*c.f.* Fig. 4, left), each view pair is considered twice by the data term, because every view holds its own scene flow representation.

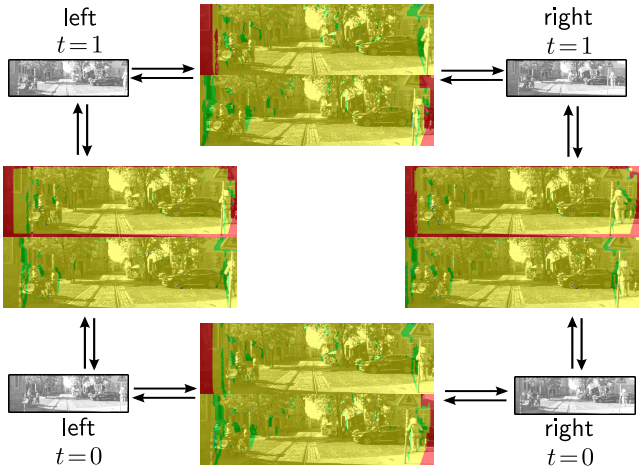


Fig. 9 Example from the KITTI training set (#191): Active data term ρ (Eqs. 21 and 22). Colors denote normal photo-consistency (*yellow*), out of bounds (*red*), occluded (*green*), moving plane violation (*dark blue*) and implausible (*light blue*) cases.

Fig. 9 illustrates the view-consistent data term. The internal states assigned by the data term (cases of Eq. 21) to each view pair are shown for each individual pixel.

5.3 View-consistent multi-frame extension

We now discuss the details of extending our view-consistent model to more than just two frames. As mentioned, geometry, motion and segmentation regularizers can be extended to a larger number of frames in a rather straightforward fashion (Eqs. 19 and 20). The data term however needs special consideration, as we need to define homographies between the additional views and also transform the normals into the specific view coordinate system. Recall that we restrict ourselves to reason only over shorter time intervals and thus can assume the motion of a moving plane to be of constant velocity in both its rotational and translational component. Under this condition suitable homographies can be found by a concatenation of the homographies defined in Eq. (3). Similarly, view-normals for the different time steps are generated by a repeated application of Eq. (1), thus again as-

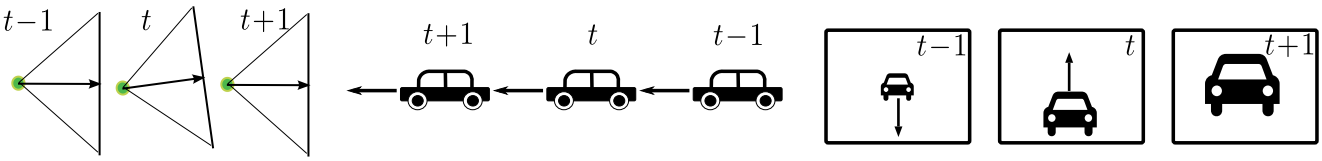


Fig. 10 Variation in camera pitch limits the validity of the constant velocity model: *(left)* A scene observed by a moving camera with varying pitch. *(right)* Camera images with induced 2D flow (black arrows). We compensate camera pitch by removing the ego-motion of the camera.

suming constant velocity. Note here, that the normals in the proposal set Π are always stored in the canonical coordinate system.

Such a model can tolerate small deviations from this constant 3D velocity assumption in the scene, but this is put to a test if the camera system itself is violating this assumption. Especially abrupt rotational changes in the viewing direction affect the whole image of the scene. The automotive application in our experiments is a good example for this. Scene flow estimation is challenged by a common high-frequent pitching motion of the stereo rig, often caused by an unsteady road surface and amplified by the suspension of the vehicle. In our model the motion is encoded relative to the respective camera coordinate system, such that even slight changes in the relative camera position can induce significant changes in the relative geometry and motion (Fig. 10). To address this problem, we introduce the following extension, in which we include ego-motion estimates for the different time steps.

First, we compute the relative ego-motion $\mathbf{E}^t = [\mathbf{Q}^t|\mathbf{s}^t]$ between all consecutive time steps t and $t+1$. The computation of homographies between successive frames then proceeds by first applying the motion induced by the moving plane representation with the ego-motion part removed, and then the relative ego-motion from time step t to $t+1$. Recall that the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ of a moving plane are stored in the coordinate system of the canonical view, thus unaware of any ego-motion. Then we can remove the relative ego-motion of the canonical view $\mathbf{E}^0$ by applying $(\mathbf{E}^0)^{-1} = [(\mathbf{Q}^0)^{-1}|-(\mathbf{Q}^0)^{-1}\mathbf{s}^0]$.

As an example, the homography between the frames t and $t+1$ in the left view becomes

$$
{}^{t}_{l}\mathbf{H}^{t+1}_{l}(\pi) = \qquad (23)
$$
$$
\mathbf{K}\Big(\mathbf{Q}^t(\mathbf{Q}^0)^{-1}\mathbf{R} - \big(\mathbf{Q}^t(\mathbf{Q}^0)^{-1}(\mathbf{t}-\mathbf{s}^0)+\mathbf{s}^t\big)(\bar{\mathbf{n}}^t_l)^\mathsf{T}\Big)\mathbf{K}^{-1}.
$$

Further note the use of the corrected view normal in Eq. (23), for which we can find a similar expression:

$$
\bar{\mathbf{n}}^t_l = \frac{\mathbf{Q}^{t-1}(\mathbf{Q}^0)^{-1}\mathbf{R}\bar{\mathbf{n}}^{t-1}_l}{1+(\mathbf{t}-\mathbf{s}^0)^\mathsf{T}\mathbf{R}\bar{\mathbf{n}}^{t-1}_l+(\mathbf{s}^{t-1})^\mathsf{T}\mathbf{Q}^{t-1}(\mathbf{Q}^0)^{-1}\mathbf{R}\bar{\mathbf{n}}^{t-1}_l}. \qquad (24)
$$

Other homographies and view-normals can be corrected for ego-motion accordingly. The estimation of camera ego-motion of a stereo camera system is a well-studied problem (*e.g.*, Badino and Kanade, 2011). Here we use the method proposed in Sec. 4.6.

5.4 Approximate inference for view-consistency

Our inference procedure closely follows the approach for a single reference view in Sec. 4.5. Again, we perform inference in a discrete CRF and optimize the energy in two steps, first solving for the mappings $\mathscr{P}$, while keeping the segmentation fixed. Then we proceed the other way around, fixing the mappings from segments to moving planes and optimizing w.r.t. to the segmentation mappings $\mathscr{S}$. The alternation can be iterated further, but again without practical benefits. Instead of an initial superpixel segmentation, we prefer to start from a regular checkerboard grid with an edge length of 16 pixels. Seed points $\mathbf{e} \in \mathscr{E}$ (see Eq. 20) are simply the grid centers. This trivial "segmentation" is more efficient and also reduces aliasing artifacts, caused by a possibly uneven size of segments across views. The per-pixel refinement step (Fig. 11) will eventually deliver a consistent over-segmentation across views, adhering to depth and motion boundaries.

Because of the grid structure, segments can be treated as large pixels when solving for $\mathscr{P}$. However, the use of an initially not view-consistent segmentation will lead to aliasing effects. We thus relax the consistency constraints and set $\varepsilon := 0.1$ and $\theta_{\text{mvp}} := 3/16\ \theta_{\text{oob}}$ in the first optimization round, to ensure that proposals are not discarded at an early stage. We generate the proposal set in the same manner as described in Sec. 4.6. We discovered that by first running a single segment-to-plane step of our reference-view version above, and removing unused proposals, the proposal set is filtered without loosing important information, leading to a significantly reduced computation time. When optimizing over more than two frames, proposals are generated for all consecutive frame pairs. *I.e.*, when using 3 frames we generate proposals for time steps $t=-1$ and $t=0$, and additionally for $t=1$ when using 4 frames. The additional proposals are discarded when they are found to be similar to already existing ones nearby. We consider proposals to be valid in a certain expansion region, centered at the seed point in the canonical frame. Empirically, we found that an expansion region size of 13×9 cells (208×144 pixels) yields a good compromise between accuracy and speed. During a fusion

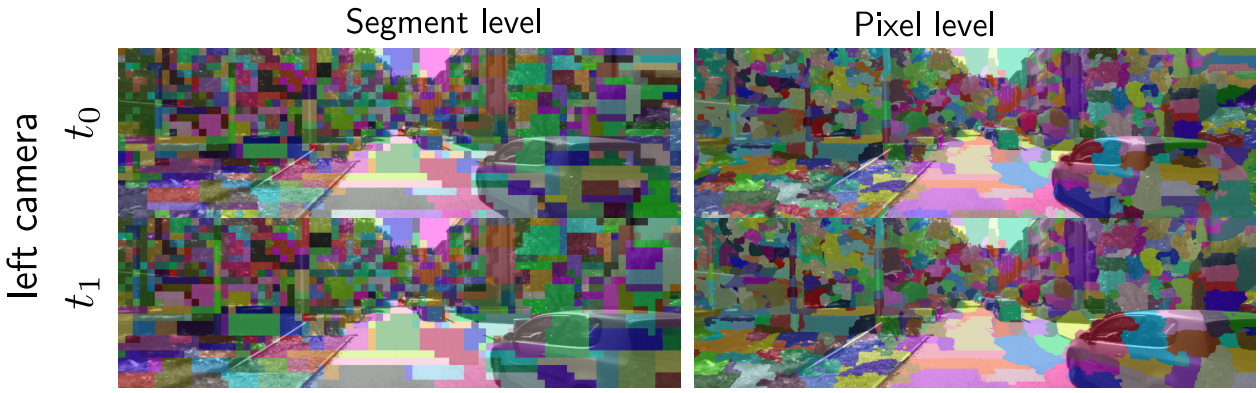


Fig. 11 Example from the KITTI training set (#191): Consistent moving plane assignments at segment level (*left*) and final consistent superpixel segmentation (*right*).

move, we thus only have to instantiate a local graph, which is determined by a projection of the expansion region into all other views.

The inference for the pixel-to-segment mappings $\mathscr{S}$ follows similar principles. Unused moving plane proposals are discarded. The size of the instantiated graph is restricted by the region constraint (Eq. 20), using an expansion region of 39×39 pixels ($N_S = 20$), and determined by projection into the other views. We penalize inconsistencies more strictly here, since the decisions are made on a per-pixel basis, and use the default parameters for ε and θ_{mvp} from Sec. 5.2. Fig. 11 illustrates the computed mappings over the course of the optimization for one of the cameras. Consistent moving plane assignments at segment level are shown on the left, illustrating the distribution of $\mathscr{P}$. The final, consistent superpixel segmentation $\mathscr{S}$ is depicted on the right.

5.4.1 Hierarchical refinement

The grid-based segment structure, furthermore, allows for a very simple refinement procedure, which we found to work well in practice. Instead of directly redistributing pixels to segments by solving for $\mathscr{S}$ after all segments have been assigned a moving plane, we optionally refine the segmentation and solve for $\mathscr{P}$ again. In practice we halve the grid resolution in each image and start the inference from the previous assignment. We prune the initial proposal set by retaining only those moving planes that are in use. In our experience, this hierarchical approach allows to reduce aliasing problems due to the smaller segment size, but still considers a more global context during the optimisation stage. Because we again set the expansion region to 13×9 cells and the set of moving plane proposals is already reduced significantly, this step is very efficient. Note that after the refinement, we also reduce the expansion region (*i.e.* $N_S = 10$) accordingly when re-assigning pixels to segments.

6 Experiments

We begin the experimental evaluation with our basic model based on a single reference view and later examine the view-consistent approach. Quantitative experiments rely on the KITTI dataset (Geiger et al, 2012), which has emerged as a standard benchmark for optical flow and stereo algorithms, with over 50 submissions in both categories. Its images are acquired by a calibrated stereo rig, mounted on top of a car together with a laser scanner, which delivers the semi-dense ground truth. Targeting automotive applications, the scenes are challenging for mainly two reasons. First, the strong forward motion of the car leads to very large displacements in stereo (>150 pixels) and flow (>250 pixels). Consequently, there are also many pixels without direct correspondence in the other image. Second, the images are acquired outdoors under realistic lighting conditions and exhibit oversaturation, shadows and lens flare, but also translucent and specular glass and metal surfaces. The KITTI benchmark is the first large scale dataset that allows evaluating scene flow methods along with their 2D counterparts, stereo and opti-

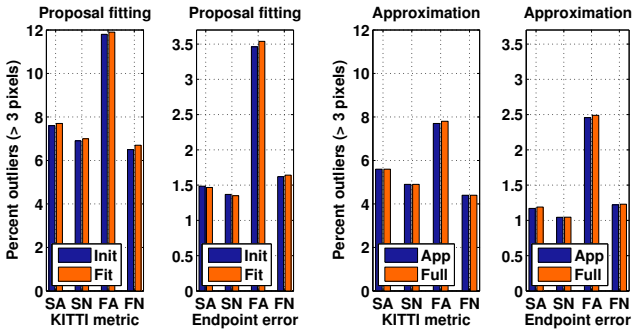


Fig. 12 *(left)* Proposal fitting procedure: Error of the 2D proposal algorithms (blue) and after planar rigid segment fitting (orange). *(right)* Approximation of the regularization term after the per-segment step: Error when evaluating the integral per pixel (orange) and when integrating the distances at the endpoints of the shared edge (blue).

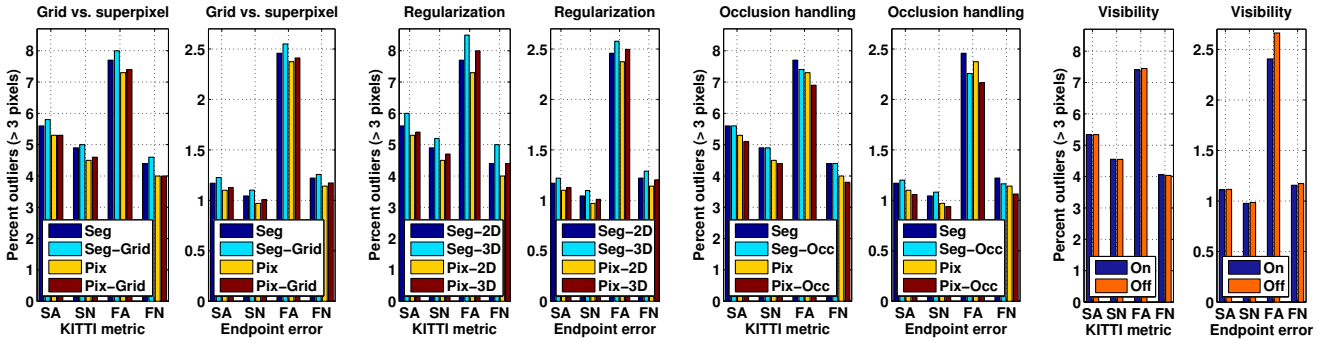


Fig. 13 Evaluation of different model choices on the KITTI training set for the per-segment and the final solution after per-pixel refinement. *(left)* Initial segmentation: Using a grid (light blue and red) compared to a superpixel segmentation (blue and yellow). *(center left)* Regularization: Comparison of 2D (blue and yellow) and 3D regularization (light blue and red). *(center right)* Occlusion handling: Our basic model with (light blue and red) and without (blue and yellow) occlusion handling. *(right)* Visibility term: Predicting all pixels to stay in bounds (orange) compared to our standard predictor (blue) (only per-pixel error).

cal flow. However, it often lacks ground truth for independently moving objects, which leads to a bias toward methods that focus on the dominant background. Nonetheless, we strongly believe that this dataset is better suited for the evaluation of scene flow methods than other existing, synthetic datasets used previously (*e.g.*, Huguet and Devernay, 2007; Vogel et al, 2011).

Our quantitative experiments mainly employ the KITTI training dataset, which is ideal for a detailed performance and parameter study due to its size of 194 images (1240× 376 pixels) with public ground truth. For a comparison to the state of the art, we also submitted our results on the 195 images of the test portion of the KITTI dataset to the official KITTI benchmark (Sec. 6.5); there the ground truth is withheld. Because of inaccuracies in the laser measurements from the moving platform, the standard KITTI metric is to compute the number of outlier pixels that deviate more than a certain threshold from the ground truth. We report results for error thresholds of 2, 3, 4, and 5 pixels for the entire image (*All*), or only for unoccluded areas (*Noc*). We additionally report the endpoint error (*EPE*) for optical flow and stereo. We occasionally use the abbreviations *SN* for stereo without occluded areas, and *SA* when including these regions. Similarly, we shorten the respective identifiers for optical flow as *FN* and *FA*.

6.1 Evaluation of the single reference view model

All experiments use fixed parameters, except where stated. We set the smoothness weight to $\lambda = 1/16$, and the weight of the segmentation term relative to λ as $\mu = 1/10\lambda$. If not mentioned otherwise, we regularize in 2D space and fix $\eta_G = \eta_M = 20$.

We generate the proposal set from the output of 2D optical flow and stereo algorithms. For computing optical flow we employ the algorithm of Vogel et al (2013a), which uses a census data term and a total generalized variation regularizer, a popular and effective combination for the KITTI scenes. To obtain an estimate in a reasonable time, we only apply 3 warps and 10 iterations per warp with an up-scaling factor of 0.9 in the image pyramid. The disparity map is obtained using semi-global matching (Hirschmüller, 2008).

First, we evaluate the proposal fitting procedure from Sec. 4.6. Fig. 12 (left) shows the KITTI metric at the default threshold (3 pixels), as well as the endpoint error of the plain 2D proposal algorithms (*Init*), and after the per-segment fitting took place (*Fit*). We observe only small changes in error, thus can conclude that planar rigid segment fitting does not significantly affect the accuracy. We attribute slight deviations in error to non-planar or non-rigid segments, *e.g.* due to misalignment with depth and motion boundaries.

Next, we investigate the simplification of the smoothness term when reasoning over segments, and how it affects the results. Recall that for computational efficiency we evaluate the spatial regularizer directly on the endpoint distances of the shared edge, instead of accumulating the contribution of all boundary pixels (Sec. 4.2). As we can see in Fig. 12 (right), the approximation *(App)* is quite accurate given our compact superpixels and on par with the exhaustive computation (*Full*), but in our experience $\sim 30\times$ faster. Note that we here report results directly after the segment-level optimization, since both approaches employ the same per-pixel refinement step.

We now demonstrate that our representation and optimization approach are quite robust, in the sense that the results do not strongly depend on the initialization, parameter choice, *etc*. The importance of the initial segmentation is evaluated in Fig. 13 (left). Starting from a trivial "grid" segmentation (edge length 16 pixels) leads to a slight decrease in performance before the per-pixel refinement takes place. This gap is closed after the refinement step. Only a small difference in accuracy remains compared to starting

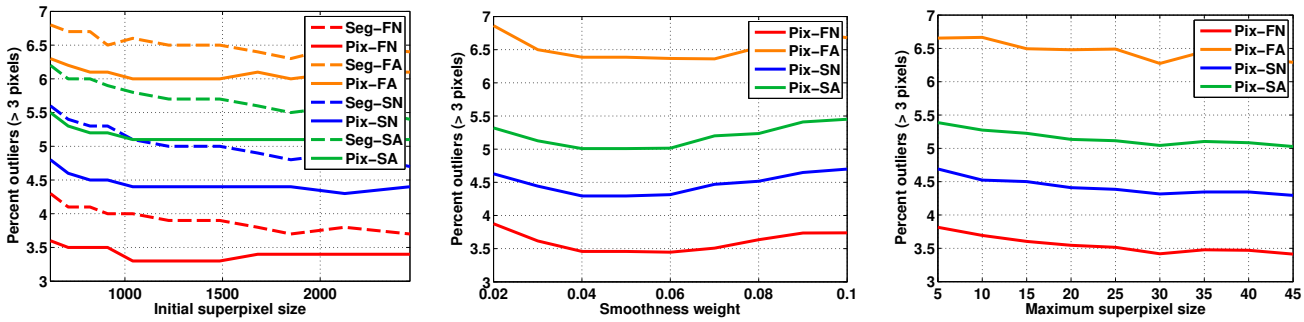


Fig. 14 *(left)* Performance w.r.t. the number of initial superpixels. *(center)* Performance w.r.t. the smoothness weight λ. *(right)* Performance w.r.t. the maximum superpixel size N_S. The legend distinguishes *Pix(el)/Seg(ment)-F(low)/S(tereo)-A(ll)/N(oc)*.

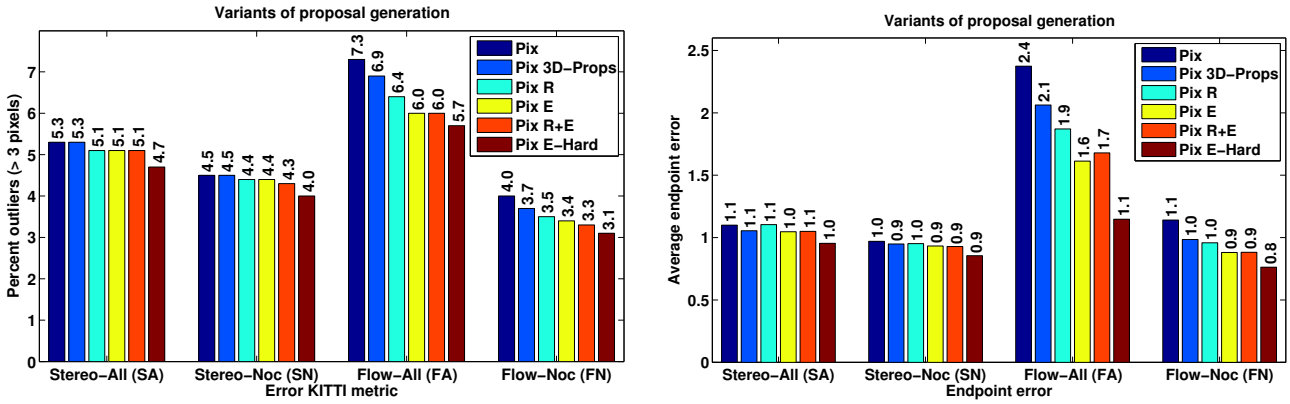


Fig. 15 Comparison of different strategies to generate proposals: We compare using only 2D proposals (*Pix*), adding more proposals from scene flow algorithms (*3D-Props*), local replacement (*R*), ego-motion (*E*), a combination of both (*R+E*), and removing all but ego-motion proposals (*E-Hard (constraint)*). Errors are evaluated using the KITTI metric and w.r.t. the endpoint error after per-pixel refinement.

from a proper superpixel segmentation. Note that this also helps understanding why, as mentioned, iterating the alternating inference further has little practical benefit; energy and error are not significantly reduced further.

The effect of starting with a different number of superpixels is depicted on the left of Fig. 14. After using more than ~ 1000 initial segments, the accuracy of the final result becomes stable, as the per-pixel refinement can compensate for eventual inaccuracies in the coarser initial segmentation. But even starting with fewer segments does not harm the performance dramatically.

Similarly, varying the weight for the regularization term λ (Fig. 14, center) and the maximum superpixel size N_S in the per-pixel refinement (Fig. 14, right) shows that the method is not sensitive to changes in these parameters. In the latter case higher values lead to better results, but also longer computation times.

Next, we investigate the behavior when switching from 2D to 3D regularization. For 3D regularization we set $\eta_G = \eta_M = 5$ and $\lambda = 0.25$, thus increase the robustness in the smoothing process. We can observe from Fig. 13 (center left) that regularization w.r.t. 2D entities is slightly superior in the evaluated measures. This can possibly be explained by the fact that the error measures do not evaluate the 3D quality of the scene flow, but only its reprojection, *i.e.* disparity and 2D optical flow.

Fig. 13 (right) depicts the effect of replacing the visibility prediction (Sec. 4.4) by a trivial predictor, which assumes pixels to always stay in bounds. As we can see, predicting visibility by the initial 2D algorithms has a strong effect on the flow endpoint error in occluded regions. Other measures, however, are nearly unaffected.

The biggest impact on the quality of the estimated scene flow is given by the different proposal algorithms we utilize. In Fig. 15 we extend our standard 2D proposal set by adding proposals from 3D scene flow methods (*3D-Props*), namely L_1-regularized 3D scene flow (Basha et al, 2010) and locally rigid 3D scene flow (Vogel et al, 2011). Furthermore, we evaluate our local replacement strategy (*R*), the ego-motion proposals (*E*, Sec. 4.6.1), and combine both proposal methods (*R+E*). Additionally, we evaluate a variant in which we replace the rigid motion component of our proposals with the estimated camera motion (*E-Hard*), thus simulating a motion stereo algorithm, which enforces a rigid scene with only ego-motion, similar to Yamaguchi et al (2013, 2014). We can observe that adding more proposals improves re-

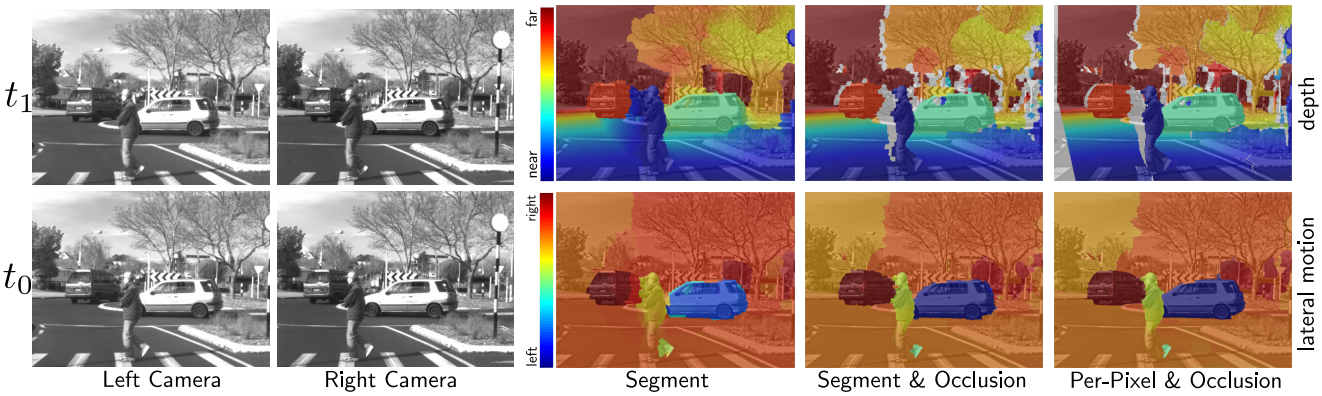


Fig. 16 Example scene from Vaudrey et al (2008), demonstrating occlusion handling and the different processing steps. Results are given for the estimated depth and the lateral 3D motion component. Detected occlusions are highlighted in white.

sults; especially the endpoint error of optical flow is reduced. A larger gain is achieved by local replacement and, furthermore, by adding additional ego-motion proposals. Both approaches are complementary to some extent, as a combination slightly improves the results further. Finally, the best results can be achieved by enforcing ego-motion as a hard constraint, underlining the bias in the KITTI benchmark.

6.1.1 Evaluation of the occlusion model

We begin the evaluation of our occlusion model of Sec. 4.7 with a qualitative example of a street scene[3] from Vaudrey et al (2008). The scene is recorded from a vehicle approaching a roundabout. Several independently moving traffic participants and a rather complex occlusion pattern pose a challenging scenario for our method. Fig. 16 displays the results after the different processing steps of our approach. The estimate appears acceptable without occlusion handling, except for regions that are not visible in the reference image, *e.g.* at the left of the pedestrian. Adding the occlusion handling from Sec. 4.7 allows to detect occluded regions and to extrapolate the lateral motion in a plausible way. The per-pixel refinement (*Per-Pixel & Occlusion*) enhances the object contours and improves the occlusion boundaries even more.

We now quantitatively compare our basic model with and without additional occlusion handling. Fig. 13 (center right) shows a small, but consistent advantage of explicit occlusion handling. The gap is largest for optical flow evaluated over the whole image. Note, however, that with additional proposals the advantage diminishes and the difference between both models becomes smaller. Recall that in order to perform optimization with graph-cut based techniques, like QPBO, the higher-order potentials, which can occur in case of multiple occlusions, have to be reduced to pairwise ones (Sec. 4.7). The resulting optimization problem possesses supermodular edges, such that nodes can remain unlabeled after running QPBO. To approximately minimize this NP-hard problem, Rother et al (2007) proposed the QPBO-I method, which we also apply here. Table 1 summarizes our experience when applying the method on the KITTI training dataset. While the number of supermodular edges and unlabeled nodes appears to be small, employing QPBO-I instead of QPBO has a notable impact on the resulting energies. At the pixel level, the number of nodes that cannot be labeled by QPBO alone appears rather high at 7.7% . Optimization with QPBO-I, however, takes an order of magnitude more time. Another challenge is that this form of occlusion reasoning is sensitive to outliers in the data term, such as specular highlights on the window of the car in Fig. 16. Note that without occlusion handling unlabeled nodes occur only very rarely (<1 per image).

Table 1 Optimization with explicit occlusion handling: Percentages of supermodular edges and resulting unlabeled nodes, resulting energy when using QPBO or QPBO-I. Numbers are averaged over the KITTI training set.

inference stage	supermodular edges	unlabeled nodes	energy w/ QPBO	energy w/ QPBO-I
Seg	4.8%	3.8%	760692	753196
Pix	1.3%	7.7%	678110	664497

6.2 Evaluation of the view-consistent model

As before, we keep all parameters fixed, unless otherwise mentioned. The only parameter that deviates strongly from the reference-view model is the smoothness weight. We set $\lambda = 1/60$, and regularize using the intensity-weighted edge length (Eq. 13), which is now based on multiple images. We

[3] Compared to KITTI images, the less challenging lighting conditions allow us to refrain from our usual census data cost. We use brightness constancy with $\rho(a,b) = \min(|a-b|, \zeta)$, truncated at $\zeta = 10\%$ of the intensity range. We use 3D regularization with rather aggressive truncation parameters ($\eta_G = \eta_M = 1$). Other deviating parameters were set to $\lambda = 0.1, \mu = 0.1, \theta_{\text{occ}} = 0.03$.

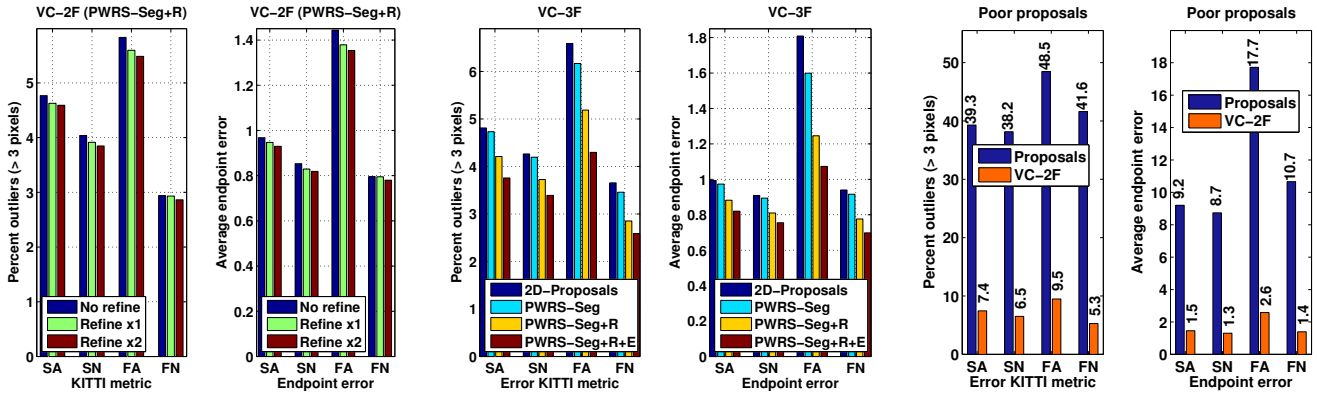


Fig. 17 *(left)* Error after refining the segmentation in the two-frame model, once (green), twice (red), or not at all (blue). *(center)* Effect of the proposals on the view-consistent three-frame model: 2D proposals (blue), single reference-view model (light blue), local replacement (yellow), and additional ego-motion (red). *(right)* Evaluation with a poor proposal set: 2D proposals (blue) and VC-2F (orange), with PWRS-Seg+R pruning. Results are shown w.r.t. the KITTI metric (> 3 pixels) and the endpoint error.

set $N_S = 20$ to speed up the per-pixel refinement, and start from an initial grid segmentation.

We begin with several quantitative analyses to illustrate the different aspects of the proposed approach. First, we investigate whether our model can benefit from the hierarchical refinement of the grid described in Sec. 5.4.1. Fig. 17 (left) compares the performance after a single and two refinement levels to the result without hierarchical refinement. The gain in performance is not large, but consistent throughout the evaluation; we use a single refinement step in the remaining experiments.

As our model is capable of jointly reasoning over multiple frames by assuming constant velocity for the rigidly moving segments, we investigate the performance when considering 2, 3, or 4 consecutive frames in Fig. 18. We further distinguish the addition of proposals from time steps other than the current one ("+"), meaning that we derive proposals from the disparity and 2D flow computed from the other adjacent frame pairs in the time window. Moreover, we include a variant that reasons about only two frames, but is provided with proposals extracted from three frames (*VC-2F+*). For comparison, we also add the single reference-view version *PWRS+R* (with local replacement), which is used to reduce the initial proposal set of the current frame pair. Note again that this reference-view method is only applied at the segment level.

Analyzing Fig. 18 one can observe that moving away from the single reference view (*PWRS+R vs. VC-2F*) already yields a significant improvement, most notably in the optical flow error w.r.t. all pixels. View-consistency benefits by considering the data of all views jointly. Parts that are occluded in the canonical view used for evaluation (and as a reference view *PWRS+R*) can still be visible in two other views. Furthermore, a strong drop in the endpoint error hints at a reduction of gross outliers. Including proposals from the previous time step (*VC-2F+*), and considering the image data of the previous frames (*VC-3F*) improve the results further. But only a combination (*VC-3F+*) of both leads to a larger performance gain in all measures, again affecting occluded regions most strongly. This suggests that a larger set of proposals from multiple frames alone is not sufficient, but that the image evidence from the longer sequence is important. Finally, including a fourth frame into the model yields diminishing returns, with only marginal improvements over the three frame case.

In another experiment we analyze the effect of the proposal set. Recall that we use the reference-view version of our method in order to prune the proposal set in the beginning, with the advantage of a reduced computation time for the view-consistent model. Fig. 17 (center), however, also shows an effect on the accuracy of the algorithm, here evaluated for the three frame case without considering additional proposals from the previous time step. Interestingly, despite the fact that the application of *PWRS-Seg* yields only a subset of the original proposals (*2D-Proposals*), the results improve. An analysis shows that both variants deliver almost the same final energy, such that the cause is not well-captured by our energy formulation. We posit that this may be due to the proposal set not being sufficiently varied in crucial parts of the solution space, which is supported by the fact that the observed accuracy difference diminishes when we use proposals from the previous time step as well (*VC-3F+*). As we would expect given previous results (Fig. 15), we observe a strong gain in accuracy from the local replacement strategy (*PWRS-Seg+R*) and ego-motion proposals (*PWRS-Seg+R+E*); in these cases the additional proposals also noticeably lower the final energy.

Because our method requires proposals for computing scene flow, we investigated how much a poor proposal set affects the performance. To that end we change the param-

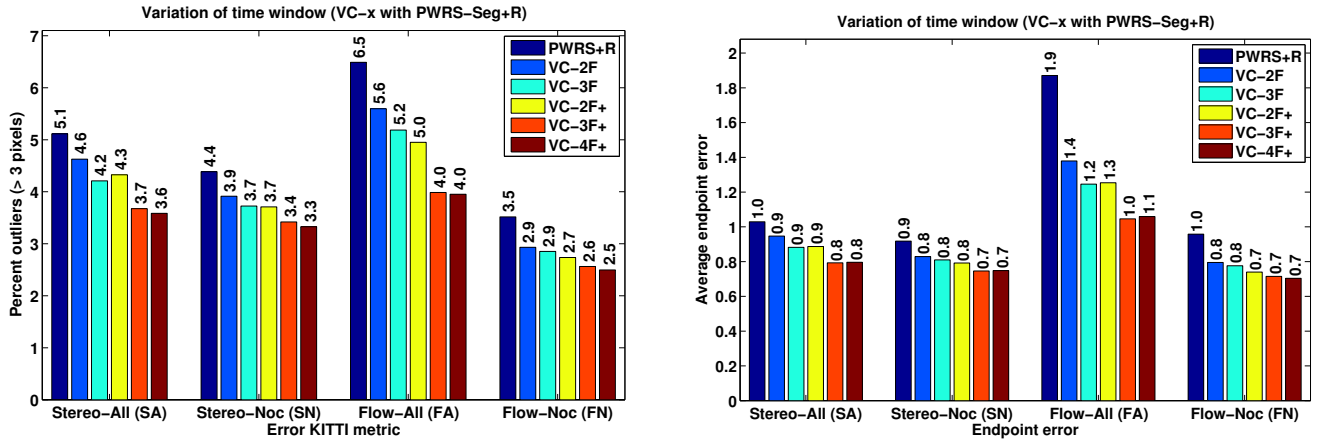


Fig. 18 Evaluation across different number of frames: Single reference-view method on 2 frames (dark blue), view-consistent method on 2 frames (blue), 3 frames (cyan), 2 frames with proposals from the previous frame (yellow), 3 frames with proposals from previous frame (orange), and 4 frames with proposals from all 4 frames (red). Results are shown w.r.t. the KITTI metric (> 3 pixels) and the endpoint error.

Table 2 Optimization in the view-consistent model (3 frames): Average number of nodes and edges in the graph, average percentage of supermodular edges and resulting unlabeled nodes (before applying QPBO-I), and resulting energy when using LSA-AUX or QPBO-I. Numbers are averaged over the KITTI training set.

inference stage	# nodes	# edges	super-modular	unla-beled	energy QPBO-I	energy LSA-AUX
Seg	2064	7485	4.01%	0.4%	3295790	3306453
Pix	3749	20102	0.64%	1.2%	2907666	2908031

eters of the initial 2D stereo and flow algorithms. For instance, in the optical flow case we use only a single warp per image pyramid and change the pyramid scale to 0.5. We then apply our two-frame view-consistent method (*VC-2F*) with *PWRS-Seg+R* to reduce the proposal set. The result is depicted in Fig. 17 (right). The notably high error of the 2D algorithms is reduced by a factor of 6 on average, showing that our scene flow approach can also cope with unfavorable proposal sets. This somewhat surprising result, achieved without considering ego-motion information, can partially be explained by the particularities of the dataset and the algorithms used to compute the proposals. The flow algorithm should deliver reasonable results in areas with only small 2D motion vectors. Given the largely planar nature of the street scenes in the dataset, these parts can then be propagated into other image areas, which have the same 3D motion and geometry, but strongly differing 2D motion. This in turn suggests that 3D scene flow may be well-suited to cope with large motions due to its internal 3D representation.

Recall that the formulation of the data term, although directly leading to only pairwise edge potentials, introduces supermodular edges into the energy. In Table 2 we investigate the situation in a similar manner as for the occlusion handling strategy with a single reference view, again collecting data over the whole KITTI training set. We apply the QPBO-I algorithm to the optimization problem given by our three-frame version *(VC-3F)* and count the number of unlabeled nodes and supermodular edges over the course of the optimization. As we can see, the number of non-submodular edges is not much lower than in the reference-view case, but unlabeled nodes occur significantly less often. This motivates us to also consider a different but more efficient graph construction[4]: The LSA-AUX algorithm (Gorelick et al, 2014) is applied at each expansion step, in order to find a submodular approximation of the problem. Conveniently, the local approximation bounds the true energy from above, such that the overall energy cannot increase, which is not the case if supermodular terms are just truncated. The final solutions show a comparable energy to results obtained with QPBO-I, while being an order of magnitude faster. Instead of LSA-AUX a similar performance can be also obtained by using QPBO without improve moves, where the former method delivers a moderate $(10-15\%)$ speed-up.

6.3 Qualitative examples

We begin with an illustration of several difficult examples from the KITTI benchmark (Fig. 19) recovered by our three-frame method (*VC-3F+*). The most interesting example is shown at the top (#74). In the presence of severe lens flares in both cameras, many optical and scene flow methods fail hopelessly to recover the motion in this scene. While the appearance of these artifacts is rather consistent in consecutive views, their location is not. This allows our approach to recover the scene flow reasonably well. Notably only 8.1% of the flow vectors of all pixels and 5.7% in the visible areas are outside the standard 3-pixel error threshold of KITTI. It

[4] compared to QPBO the graph contains only half of the nodes

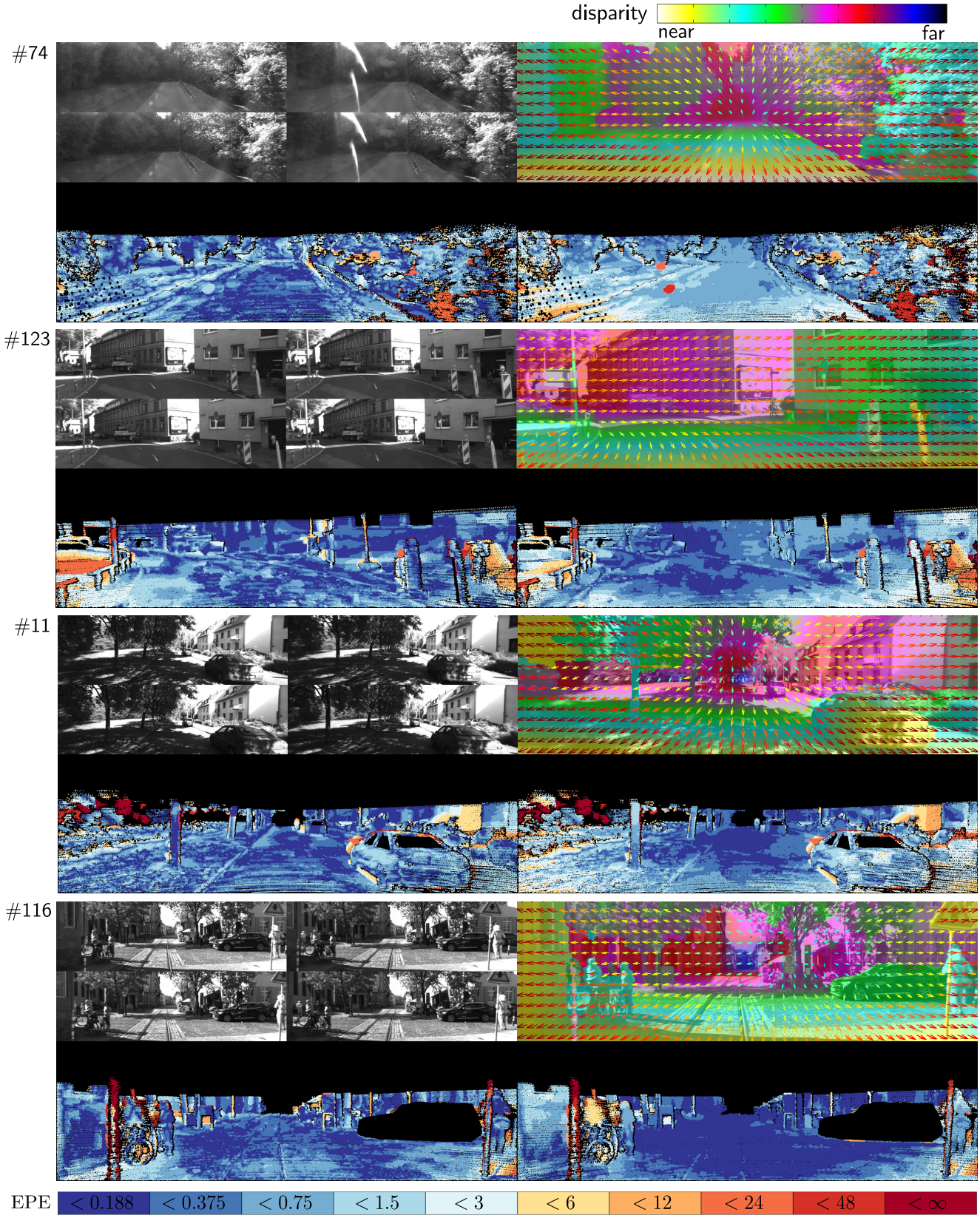


Fig. 19 Examples from the KITTI training set. Input images *(top left)* and recovered scene flow *(top right)*, color coded as disparity (from white – near to blue – far) and motion vectors, reprojected into the image plane. Arrow lengths are depicted with a log-scale. Colors encode the length of the actual 2D displacement (blue – small to red – large). Color coded endpoint error for disparity (*bottom left*) and flow (*bottom right*).

is important to note that the robust handling of these artifacts is achieved only through view- and multi-frame consistency. Also depicted is a scene (#11) with low image contrast in shadow regions. Scene #123 is interesting because of similar problems with lens flare as for #74, here however challenging the reconstruction of the geometry as their location is consistent across frames. Finally #116 has fine structures in the image (*e.g.*, the traffic sign), several areas with occlusions, and a car moving independently, albeit without ground truth.

Fig. 20 illustrates results for different outdoor scenes from Meister et al (2012). We display the input images on the left. Our scene flow estimates (*VC-3F+*) are shown as disparities (center) and reprojected optical flow in the usual color coding. These examples show that our model is capable of handling independent object motion under unfavorable conditions. Even though the motion displacements in the image plane are rather small, the scenes contain many scenarios that are hard for conventional flow and stereo algorithms. The scenes 'car truck' and 'crossing' have saturated highlights and reflections, as well as a rather complex occlusion pattern. The scene 'car truck' also exhibits cast shadows dancing on the truck and the street. More challenging is 'sun flares', where the sun causes lens flares and 'flying snow', which as the name suggests contains heavy snow fall and a wet and reflecting street. The scene from Fig. 2 shows the wiper occluding the view and is, therefore, very difficult to recover for conventional approaches that parameterize the scene in a single camera only. The most complex scene is 'night snow', in which the aperture of the cameras is wide open and the image has only a shallow depth of field. Moreover, the windshield is wet, causing the headlights of approaching cars to flare. We can only give a qualitative evaluation here, as no ground truth for these scenes is available. Apart from the last scene, which has an incorrect depth in the sky region, our estimates appear quite appropriate.

6.3.1 Typical failure cases

Fig. 21 displays some typical failure cases of our method. For example, it is challenged by over-saturated areas, especially if these are located close to the boundary of the images or in occlusion regions. Recall that we replace the data term with a fixed penalty (θ_{occ} or θ_{oob}), if a pixel lacks a correspondence in other images. Now assume that a proposal exists that maps this over-saturated image region to a similarly over-saturated, but incorrect one in the other images. The data penalty in this case is close to zero, which compared to the energy of the true solution in our model (θ_{oob}) is decidedly lower. By demanding view-consistency, this incorrect solution will still incur penalties for the incorrect regions, since the geometry and/or motion is not consistent. However, as the penalties are accumulated per pixel, whether the correct correspondence can be recovered depends on the size of the respective regions in the images.

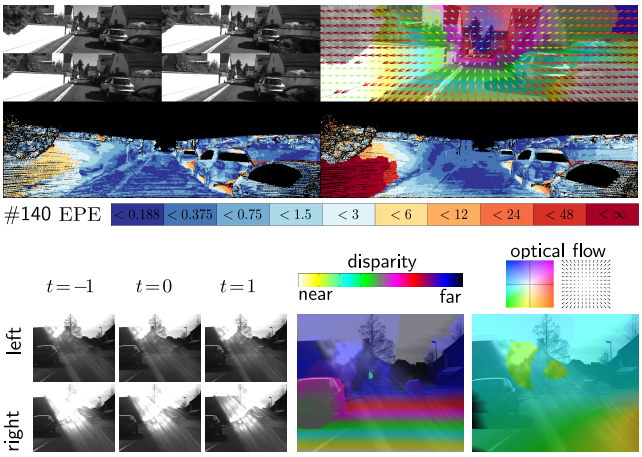


Fig. 21 Typical failure cases from the KITTI training set: *(top)* Over-saturated area moving out of viewing frustum at the wall on the left. *(bottom)* Example from Meister et al (2012) showing a sun flare with a consistent motion pattern.

As already mentioned, a second challenge are imaging artifacts, *e.g.* sun flares (Fig. 21, bottom), that appear consistently in all the views. In the example the sun flare even leads to over-saturation, such that the low data energy may overrule the consistency penalty.

6.4 Quantitative summary and timings

A direct comparison between the view-consistent and single reference-view models is given in Table 3. Note that these differ from the published results in (Vogel et al, 2013b, 2014) due to a change in the KITTI ground truth, slightly different parameter sets, and extensions such as the local replacement strategy. The first row gives results for the 2D algorithms used to derive the proposals (*2D Algorithms*). Otherwise, we use the usual notation: *PWRS* for our basic reference-view model, *PWRS+R* for a version with local replacements, and *PWRS+R+E* to denote the usage of additional ego-motion proposals. For the view-consistent version (*VC*) we use *PWRS+R+E* to prune the proposals and distinguish between the two, three and four-frame versions, with (+) and without proposals from all frames. In general the numbers improve from top to bottom. Already our basic version achieves a significant reduction in all error measures compared to the state-of-the-art 2D proposals. Both strategies to generate additional proposals show their benefit, especially for flow. The view-consistent model leads to a visible reduction of the error in all measures already for the two-frame case. Moving to three frames further improves the results, especially for occluded areas, but considering four frames only yields marginal improvements. Notably,

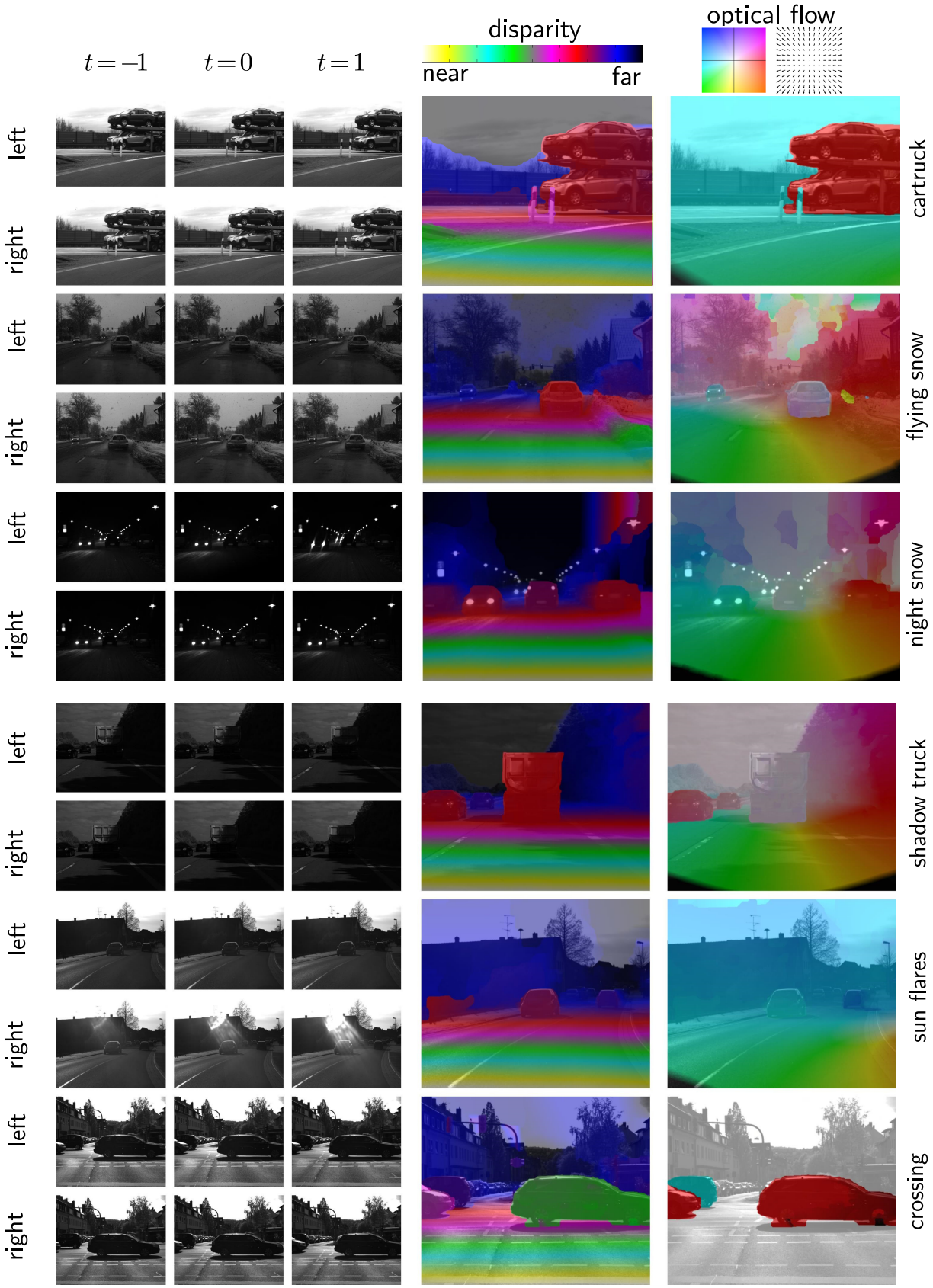


Fig. 20 Challenging examples from Meister et al (2012): Input frames of our method *(left)*. Recovered scene flow, reprojected to disparity *(center)* and 2D flow field *(right)*.

Table 3 Results on the *KITTI training set*: Average KITTI metric (% of flow vectors / disparities above 2, 3, 4, 5 pixels of endpoint error) and average endpoint error [px] with (All) and without (Noc) counting occluded regions.

Method	Flow										Stereo									
	KITTI metric								AEP		KITTI metric								AEP	
	All				Noc				All	Noc	All				Noc				All	Noc
	2px	3px	4px	5px	2px	3px	4px	5px			2px	3px	4px	5px	2px	3px	4px	5px		
2D Algorithms	14.6	11.7	10.1	9.0	8.5	6.5	5.5	4.8	3.5	1.6	10.3	7.5	6.1	5.2	9.4	6.8	5.5	4.7	1.5	1.4
PRSF	9.9	7.3	6.0	5.2	5.7	4.0	3.2	2.7	2.4	1.1	7.7	5.3	4.1	3.4	6.7	4.5	3.5	2.9	1.1	1.0
PRSF+R	9.1	6.4	5.0	4.3	5.2	3.5	2.7	2.3	1.9	1.0	7.4	5.1	4.0	3.3	6.4	4.4	3.4	2.9	1.1	1.0
PRSF+R+E	8.6	6.0	4.7	3.9	5.0	3.3	2.6	2.1	1.7	0.9	7.3	5.1	4.0	3.3	6.3	4.3	3.4	2.8	1.0	0.9
VC-2F	7.8	5.2	4.0	3.2	4.3	2.8	2.1	1.7	1.2	**0.7**	6.6	4.5	3.5	2.8	5.7	3.8	2.9	2.4	0.9	0.8
VC-2F+	7.4	4.9	3.7	3.0	4.2	2.7	2.0	1.7	1.2	**0.7**	6.2	4.2	3.3	2.7	5.4	3.6	2.8	2.3	0.9	0.8
VC-3F	6.9	4.3	3.1	2.5	4.1	2.6	1.9	1.6	**1.1**	**0.7**	5.7	3.8	2.9	2.3	5.1	3.4	2.6	**2.1**	**0.8**	0.8
VC-3F+	6.4	4.0	**2.8**	**2.2**	4.0	**2.5**	1.9	**1.5**	**1.1**	**0.7**	5.4	**3.6**	**2.8**	2.3	5.0	3.4	2.6	**2.1**	**0.8**	**0.7**
VC-4F+	**6.3**	**3.9**	**2.8**	**2.2**	**3.9**	**2.5**	**1.8**	**1.5**	**1.1**	**0.7**	**5.2**	**3.6**	**2.8**	**2.2**	**4.8**	**3.3**	**2.5**	**2.1**	**0.8**	**0.7**

Table 4 Timings on KITTI images (0.5MPixels), measured on a dual *Intel Core i7* computer and two proposals per segment, for two different numbers of initial segments.

# Segments	Proposals		PWRS				VC-SF					
	Init	Fit	Seg	R	E	Pix	Seg-2F	Seg-Ref-2F	Seg-3F	Seg-Ref-3F	Pix-2F	Pix-3F
1850	60s	16s	19s	8s	9s	15s	23s	9s	46s	12s	18s	30s
1150	60s	16s	10s	8s	5s	10s	17s	6s	32s	8s	14s	23s

all but two numbers are at least halved when comparing our best result with the initial 2D solution.

Table 4 illustrates the time spent on the different parts of the algorithm. We distinguish between running the 2D flow and stereo algorithms (*Init*), the proposal fitting procedure (*Fit*), and further time the inference at the segment level (*Seg*) and at the pixel level (*Pix*). We also show the time needed for generating additional proposals (*R* and *E*), and one hierarchical refinement step (*Ref*). We compare numbers when starting with 1850 and 1150 segments. In both cases, our model with a single reference view (*PWRS*) needs less time for the optimization and both additional proposal generation strategies than for computing the initial optical flow and disparity maps. For the view-consistent case, we exploit the reduction in the number of proposals by first running *PWRS+R+E* at the segment level. With a low number of segments, our basic version (*PWRS*) needs only 20s to deliver a result. However, running the 2D proposal algorithms already takes significantly more time. Our most advanced three-frame method needs $\sim$ 3 minutes including proposals.

6.5 Comparison with the state of the art

Table 5 shows a comparison of our piecewise rigid scene model with the state of the art on the KITTI test set. At the time of writing (August 2014), the benchmark has over 50 submissions in both categories. Our scene flow methods rank among the top performers, with the view-consistent model coming out first overall for both stereo and flow, when considering full images with occluded areas. Note that several top-ranked methods assume epipolar motion as a hard constraint (Setting *ms*). In contrast, our method can handle scenes with independently moving objects (*c.f.* Fig. 20), which are uncommon in the benchmark. Considering only methods applicable to general scenes, *i.e.* with independent object motion, the distance to the next best competitor is rather large, which demonstrates that scene flow from our piecewise rigid scene model, has a clear advantage over single camera methods for motion estimation under challenging conditions.

7 Conclusion

In this paper we introduced a scene flow approach that models dynamic scenes as a collection of piecewise planar, local regions, moving rigidly over time. It allows to densely recover geometry, 3D motion, and an over-segmentation of the scene into moving planes, leading to accurate geometry and motion boundaries. Employing unified reasoning over geometry, motion, segmentation and occlusions within the observed scene, our method achieves leading performance in a popular benchmark, surpassing dedicated state-of-the-art stereo and optical flow techniques at their respective task. We extend our basic reference-view technique to leverage information from multiple consecutive frames of a stereo video. Our view-consistent approach exploits consistency

Table 5 Comparison with the state-of-the-art on the *KITTI test set*: Our methods are denoted as PRSF+R (reference view, 2D proposals, local replacement), PRSF+R+E (with ego-motion proposals), and VC-3F+ (view consistent, 3 frames, using PRSF+R to reduce the proposal set). The settings column marks scene flow (sf), multi-frame (mv), and motion stereo (ms) methods.

Stereo evaluation					
Method	Setting	KITTI metric (% > 3 px)		EPE [px]	
		All	Noc	All	Noc
(Yamaguchi et al, 2014)	sf ms	3.64	**2.83**	0.9	**0.8**
VC-3F+	sf mv	**3.31**	3.05	**0.8**	**0.8**
(Yamaguchi et al, 2013)		4.72	3.40	1.0	**0.8**
(Yamaguchi et al, 2013)		5.11	3.92	1.0	0.9
PRSF+R+E	sf	4.87	4.02	1.0	0.9
(Yamaguchi et al, 2012)		5.37	4.04	1.1	0.9
PRSF+R	sf	5.22	4.36	1.1	0.9
(Einecke and Eggert, 2014)		5.94	4.86	1.2	1.0
(Spangenberg et al, 2013)		6.18	4.97	1.6	1.3
(Ranftl et al, 2013)		6.88	5.02	1.6	1.0

Optical flow evaluation					
Method	Setting	KITTI metric (% > 3 px)		EPE [px]	
		All	Noc	All	Noc
VC-3F+	sf mv	**4.84**	**2.72**	**1.3**	**0.8**
(Yamaguchi et al, 2014)	sf ms	5.61	2.82	**1.3**	**0.8**
PRSF+R+E	sf	7.07	3.57	1.6	0.9
(Yamaguchi et al, 2013)	ms	8.28	3.64	2.2	0.9
PRSF+R	sf	7.39	3.76	2.8	1.2
(Yamaguchi et al, 2013)	ms	10.56	3.91	2.7	0.9
(Ranftl et al, 2014)		11.96	5.93	3.8	1.6
(Braux-Zin et al, 2013)		15.15	6.20	4.5	1.5
(Demetz et al, 2014)		11.03	6.52	2.8	1.5
(Vogel et al, 2013a)		14.57	7.11	5.5	1.9

over time and viewpoints, thereby significantly improving 3D scene flow estimation. In particular, our model shows remarkable robustness to missing evidence, outliers, and occlusions, and can recover motion and geometry even under unfavorable imaging conditions, where many methods fail.

In future work we plan to incorporate long-term temporal consistency into our framework, and to relax the constant velocity assumption to a more flexible formulation. Moreover, we aim to explicitly model small deviations from the local planarity and rigidity assumptions. Another promising route may be to include object-level semantic image understanding into the segmentation scheme, with associated class-specific motion and geometry models.

Acknowledgements SR was supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement No. 307942, as well as by the EU FP7 project "Harvest4D" (No. 323567).

References

Adiv G (1985) Determining three-dimensional motion and structure from optical flow generated by several moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 7(4):384–401

Ali AM, Farag AA, Gimel'Farb GL (2008) Optimizing binary MRFs with higher order cliques. In: European Conference on Computer Vision

Badino H, Kanade T (2011) A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion. In: IAPR Conference on Machine Vision Application, pp 185–189

Baker S, Scharstein D, Lewis J, Roth S, Black MJ, Szeliski R (2011) A database and evaluation methodology for optical flow. International Journal of Computer Vision 92(1):1–31, `vision.middlebury.edu/flow`

Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics 28(3):24:1–24:11

Basha T, Moses Y, Kiryati N (2010) Multi-view scene flow estimation: A view centered variational approach. In: IEEE Conference on Computer Vision and Pattern Recognition

Black MJ, Anandan P (1991) Robust dynamic motion estimation over time. In: IEEE Conference on Computer Vision and Pattern Recognition

Black MJ, Jepson AD (1996) Estimating optical flow in segmented images using variable-order parametric models with local deformations. IEEE Trans Pattern Anal Mach Intell 18(10):972–986, DOI 10.1109/34.541407, URL `http://dx.doi.org/10.1109/34.541407`

Bleyer M, Rother C, Kohli P (2010) Surface stereo with soft segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition

Bleyer M, Rhemann C, Rother C (2011a) PatchMatch stereo – Stereo matching with slanted support windows. In: British Machine Vision Conference

Bleyer M, Rother C, Kohli P, Scharstein D, Sinha SN (2011b) Object stereo – Joint stereo matching and object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition

Braux-Zin J, Dupont R, Bartoli A (2013) A general dense image matching framework combining direct and feature-based costs. In: IEEE International Conference on Computer Vision

Brox T, Malik J (2011) Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3):500–513

Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: European Conference on Computer Vision

Carceroni RL, Kutulakos KN (2002) Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance. International Journal of Computer Vision 49:175–214

Courchay J, Pons JP, Monasse P, Keriven R (2009) Dense and accurate spatio-temporal multi-view stereovision. In: Asian Conference on Computer Vision

Demetz O, Stoll M, Volz S, Weickert J, Bruhn A (2014) Learning brightness transfer functions for the joint recovery of illumination changes and optical flow. In: European Conference on Computer Vision, to appear

Devernay F, Mateus D, Guilbert M (2006) Multi-camera scene flow by tracking 3-D points and surfels. In: IEEE Conference on Computer Vision and Pattern Recognition

Einecke N, Eggert J (2014) Block-matching stereo with relaxed fronto-parallel assumption. In: Intelligent Vehicles, pp 700–705

Furukawa Y, Ponce J (2008) Dense 3D motion capture from synchronized video streams. In: IEEE Conference on Computer Vision and Pattern Recognition

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

Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? In: IEEE Conference on Computer Vision and Pattern Recognition, www.cvlibs.net/datasets/kitti/

Gorelick L, Veksler O, Boykov Y, Ben Ayed I, Delong A (2014) Local submodular approximations for binary pairwise energies. In: Computer Vision and Pattern Recognition

Hirschmüller H (2008) Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2):328–341

Hornacek M, Fitzgibbon A, Rother C (2014) SphereFlow: 6 DoF scene flow from RGB-D pairs. In: IEEE Conference on Computer Vision and Pattern Recognition

Huguet F, Devernay F (2007) A variational method for scene flow estimation from stereo sequences. In: IEEE International Conference on Computer Vision

Hung CH, Xu L, Jia J (2013) Consistent binocular depth and scene flow with chained temporal profiles. International Journal of Computer Vision 102(1-3):271–292

Irani M (2002) Multi-frame correspondence estimation using subspace constraints. International Journal of Computer Vision 48(3):173–194

Ishikawa H (2009) Higher-order clique reduction in binary graph cut. In: IEEE Conference on Computer Vision and Pattern Recognition

Ju S, Black M, Jepson A (1996) Skin and bones: multi-layer, locally affine, optical flow and regularization with transparency. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 307–314, DOI 10.1109/CVPR.1996.517090, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=517090

Kolmogorov V, Zabih R (2001) Computing visual correspondence with occlusions using graph cuts. In: IEEE International Conference on Computer Vision, pp 508–515

Lempitsky V, Roth S, Rother C (2008) FusionFlow: Discrete-continuous optimization for optical flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition

Lempitsky V, Rother C, Roth S, Blake A (2010) Fusion moves for Markov random field optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(8):1392–1405

Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, vol 2

Meister S, Jähne B, Kondermann D (2012) Outdoor stereo camera system for the generation of real-world benchmark data sets. Optical Engineering 51(2)

Müller T, Rannacher J, Rabe C, Franke U (2011) Feature- and depth-supported modified total variation optical flow for 3D motion field estimation in real scenes. In: IEEE Conference on Computer Vision and Pattern Recognition

Murray DW, Buxton BF (1987) Scene segmentation from visual motion using global optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(2):220–228

Nir T, Bruckstein A, Kimmel R (2008) Over-parameterized variational optical flow. International Journal of Computer Vision 76(2):205–216

Park J, Oh TH, Jung J, Tai YW, Kweon IS (2012) A tensor voting approach for multi-view 3D scene flow estimation and refinement. In: European Conference on Computer Vision

Rabe C, Müller T, Wedel A, Franke U (2010) Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In: European Conference on Computer Vision

Ranftl R, Pock T, Bischof H (2013) Minimizing TGV-based variational models with non-convex data terms. In: Scale Space and Variational Methods in Computer Vision

Ranftl R, Bredies K, Pock T (2014) Non-local total generalized variation for optical flow estimation. In: European Conference on Computer Vision, to appear

Rother C, Kolmogorov V, Lempitsky V, Szummer M (2007) Optimizing binary MRFs via extended roof duality. In: IEEE Conference on Computer Vision and Pattern Recognition

Rother C, Kohli P, Feng W, Jia J (2009) Minimizing sparse higher order energy functions of discrete variables. In: IEEE Conference on Computer Vision and Pattern Recognition

Schoenemann T, Cremers D (2008) High resolution motion layer decomposition using dual-space graph cuts. In: IEEE Conference on Computer Vision and Pattern Recognition

Spangenberg R, Langner T, Rojas R (2013) Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In: Computer Analysis of Images and Patterns

Sun D, Sudderth EB, Black MJ (2010) Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In: Neural Information Processing Systems

Sun D, Wulff J, Sudderth E, Pfister H, Black M (2013) A fully-connected layered model of foreground and background flow. In: IEEE Conference on Computer Vision and Pattern Recognition

Tao H, Sawhney HS (2000) Global matching criterion and color segmentation based stereo. In: IEEE Workshop on Applications in Computer Vision

Unger M, Werlberger M, Pock T, Bischof H (2012) Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In: IEEE Conference on Computer Vision and Pattern Recognition

Valgaerts L, Bruhn A, Zimmer H, Weickert J, Stoll C, Theobalt C (2010) Joint estimation of motion, structure and geometry from stereo sequences. In: European Conference on Computer Vision

Vaudrey T, Rabe C, Klette R, Milburn J (2008) Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: International Conference on Image and Vision Computing New Zealand

Vedula S, Baker S, Collins R, Kanade T, Rander P (1999) Three-dimensional scene flow. In: IEEE Conference on Computer Vision and Pattern Recognition

Veksler O, Boykov Y, Mehrani P (2010) Superpixels and supervoxels in an energy optimization framework. In: European Conference on Computer Vision

Vogel C, Schindler K, Roth S (2011) 3D scene flow estimation with a rigid motion prior. In: IEEE International Conference on Computer Vision

Vogel C, Roth S, Schindler K (2013a) An evaluation of data costs for optical flow. In: Pattern Recognition (Proc. of GCPR), pp 343–353

Vogel C, Schindler K, Roth S (2013b) Piecewise rigid scene flow. In: IEEE International Conference on Computer Vision

Vogel C, Roth S, Schindler K (2014) View-consistent 3D scene flow estimation over multiple frames. In: European Conference on Computer Vision

Volz S, Bruhn A, Valgaerts L, Zimmer H (2011) Modeling temporal coherence for optical flow. In: IEEE International Conference on Computer Vision

Wang JYA, Adelson EH (1994) Representing moving images with layers. IEEE Transactions on Image Processing 3:625–638

Wedel A, Rabe C, Vaudrey T, Brox T, Franke U, Cremers D (2008) Efficient dense scene flow from sparse or dense stereo data. In: European Conference on Computer Vision

Werlberger M, Trobin W, Pock T, Wedel A, Cremers D, Bischof H (2009) Anisotropic Huber-L1 optical flow. In: British Machine Vision Conference

Yamaguchi K, Hazan T, McAllester D, Urtasun R (2012) Continuous Markov random fields for robust stereo estimation. In: European Conference on Computer Vision

Yamaguchi K, McAllester D, Urtasun R (2013) Robust monocular epipolar flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition

Yamaguchi K, McAllester D, Urtasun R (2014) Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: European Conference on Computer Vision

Zabih R, Woodfill J (1994) Non-parametric local transforms for computing visual correspondence. In: European Conference on Computer Vision

A Higher-order Reductions for Occlusion Handling with a Reference View

We here describe how to convert the occlusion-sensitive data term from Eq. (17) into a quadratic pseudo-Boolean function. Note that the only interesting case is $|\mathscr{O}^0_{\mathbf{p}}| \geq 2$, that is there are two or more possibly occluding pixels. Otherwise, the problem is already in quadratic form ($|\mathscr{O}^0_{\mathbf{p}}| = 1$), or there is no occluding pixel and only the (unary) data term is required ($|\mathscr{O}^0_{\mathbf{p}}| = 0$).

Recall that Eq. (17) is defined as part of a single α-expansion step, *i.e.* a pixel can only be assigned two possible labels (α or its previous label). For simplicity we restrict the analysis to the case $i = 0$. We thus consider the term

$$\hat{u}^0_{\mathbf{p}}[x_{\mathbf{p}} = 0] \prod_{(\mathbf{q},j)\in\mathscr{O}^0_{\mathbf{p}}} [x_{\mathbf{q}} \neq j]. \tag{25}$$

The reduction for $i = 1$ is analogous.

First, let us consider the special case in which there is a pixel $\mathbf{q}$ that occludes pixel $\mathbf{p}$ in both possible assignments of $x_{\mathbf{q}}$, that is $(\mathbf{q},0) \in \mathscr{O}^0_{\mathbf{p}}$ and $(\mathbf{q},1) \in \mathscr{O}^0_{\mathbf{p}}$. In that case the pixel $\mathbf{p}$ is always occluded and Eq. (25) vanishes. For the remaining cases, we distinguish between $\hat{u}^0_{\mathbf{p}} < 0$ and $\hat{u}^0_{\mathbf{p}} > 0$.

Case $\hat{u}^0_{\mathbf{p}} < 0$: We can substitute the whole term with the help of at most one non-submodular term with weight $\hat{u}^0_{\mathbf{p}}$. No non-submodular term is introduced if all Boolean variables in the term are inverted, *i.e.* $j \equiv 1$. In that case Eq. (25) becomes

$$\hat{u}^0_{\mathbf{p}}(1 - x_{\mathbf{p}}) \prod_{(\mathbf{q},1)\in\mathscr{O}^0_{\mathbf{p}}} (1 - x_{\mathbf{q}}). \tag{26}$$

Introducing an additional variable z, the polynomial in Eq. (26) can be replaced by

$$\min_z \hat{u}^0_{\mathbf{p}} \left(1 - z - (1-z)x_{\mathbf{p}} - \sum_{(\mathbf{q},1)\in\mathscr{O}^0_{\mathbf{p}}} (1-z)x_{\mathbf{q}} \right) \tag{27}$$

in quadratic form.

If $x_{\mathbf{p}} = 0$ and the other variables encode a constellation where $\mathbf{p}$ is not occluded, then the expression becomes equal to $\hat{u}^0_{\mathbf{p}}$ (by setting $z = 0$). Otherwise, the minimum is attained at 0 (with $z = 1$).

In the case of there being a $(\mathbf{q},0) \in \mathscr{O}^0_{\mathbf{p}}$, we follow the scheme introduced by Rother et al (2009). With the introduction of two auxiliary variables z_0, z_1, we replace the product in Eq. (25) by

$$\begin{aligned} \min_{z_0,z_1} & -\hat{u}^0_{\mathbf{p}}(z_0 z_1 - z_1 + (1 - z_0)x_{\mathbf{p}}) \\ & - \hat{u}^0_{\mathbf{p}} \sum_{(\mathbf{q},j)\in\mathscr{O}^0_{\mathbf{p}}} \Big(z_1(1 - x_{\mathbf{q}}) + (1 - z_0)x_{\mathbf{q}} \Big). \end{aligned} \tag{28}$$

Here, the term $-\hat{u}^0_{\mathbf{p}} z_0 z_1$ is not submodular. Like in the previous case, if the variables do not encode an occlusion, and if $x_{\mathbf{p}} = 0$, the minimum is $\hat{u}^0_{\mathbf{p}}$ (setting $z_0 = 0$ and $z_1 = 1$). Otherwise the minimum is 0 (setting $z_0 = 1$ and $z_1 = 0$).

Case $\hat{u}^0_{\mathbf{p}} > 0$: We approach this problem using a series of substitutions. Following Ali et al (2008) we replace a product of two variables in Eq. (25), $x_{\mathbf{q}_1} x_{\mathbf{q}_2}$, with a new variable z, and add

$$\min_z \hat{u}^0_{\mathbf{p}}(x_{\mathbf{q}_1} x_{\mathbf{q}_2} - 2x_{\mathbf{q}_1} z - 2x_{\mathbf{q}_2} z + 3z), \tag{29}$$

such that after the substitution Eq. (25) becomes

$$\begin{aligned} & \hat{u}^0_{\mathbf{p}}(x_{\mathbf{q}_1} x_{\mathbf{q}_2} - 2x_{\mathbf{q}_1} z - 2x_{\mathbf{q}_2} z + 3z) + \\ & \hat{u}^0_{\mathbf{p}}(1 - x_{\mathbf{p}})z \prod_{\substack{(\mathbf{q},j)\in\mathscr{O}^0_{\mathbf{p}} \setminus \\ \{(\mathbf{q}_1,0),(\mathbf{q}_2,0)\}}} [x_{\mathbf{q}} \neq j]. \end{aligned} \tag{30}$$

Two inverted Boolean variables can be replaced in the same manner. Note that we are not restricted to replacing only variables from $\mathscr{O}^0_{\mathbf{p}}$, but can also substitute $1 - x_{\mathbf{p}}$ itself.

The substitution introduces one non-submodular term with weight $\hat{u}^0_{\mathbf{p}}$. To arrive at a quadratic polynomial one needs to replace all but two literals of the product as described, leading to $n - 1$ or $n - 2$ non-submodular terms.