Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?

Aseem Behl[2,*] Omid Hosseini Jafari[1,*] Siva Karthik Mustikovela[1,*]
Hassan Abu Alhaija[1] Carsten Rother[1] Andreas Geiger[2,3]

[1]Computer Vision Lab, TU Dresden
[2]Autonomous Vision Group, MPI for Intelligent Systems Tübingen
[3]Computer Vision and Geometry Group, ETH Zürich

Abstract

Existing methods for 3D scene flow estimation often fail in the presence of large displacement or local ambiguities, e.g., at texture-less or reflective surfaces. However, these challenges are omnipresent in dynamic road scenes, which is the focus of this work. Our main contribution is to overcome these 3D motion estimation problems by exploiting recognition. In particular, we investigate the importance of recognition granularity, from coarse 2D bounding box estimates over 2D instance segmentations to fine-grained 3D object part predictions. We compute these cues using CNNs trained on a newly annotated dataset of stereo images and integrate them into a CRF-based model for robust 3D scene flow estimation - an approach we term Instance Scene Flow. We analyze the importance of each recognition cue in an ablation study and observe that the instance segmentation cue is by far strongest, in our setting. We demonstrate the effectiveness of our method on the challenging KITTI 2015 scene flow benchmark where we achieve state-of-the-art performance at the time of submission.

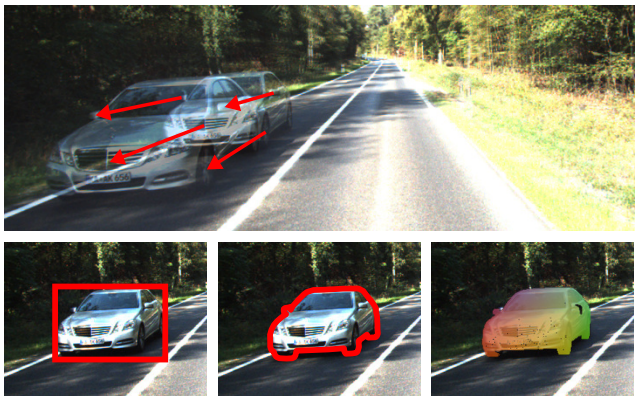

Figure 1: **Motivation.** Top: Two consecutive frames (overlaid) from the KITTI 2015 scene flow dataset. Large displacements and specular surfaces are challenging for current scene flow estimation algorithms. Bottom: Recognition can provide powerful geometric cues to help with this problem. In this work, we investigate: 2D bounding boxes, 2D instance segmentations and 3D object coordinates.

1. Introduction

3D motion estimation is a core problem in computer vision with numerous applications. Consider, for instance, an autonomous car navigating through a city. Besides recognizing traffic participants and identifying their 3D locations, it needs to precisely predict their 3D position in the future.

In this paper, we focus on 3D scene flow estimation for autonomous driving scenarios. More specifically, given two consecutive stereo images we want to predict the 3D motion of every pixel. With the advent of challenging real-world benchmarks, such as the KITTI 2012 [10] and KITTI 2015 [22], great progress has been made in this area. In particular, segmentation-based formulations which gain their strength by reasoning over piece-wise planar patches [41] or segmenting the scene into its rigidly moving components [22] dominate the leaderboards.

However, existing methods rely heavily on local features for computing the data term and for initialization. As a consequence, even state-of-the-art methods fail in the presence of very large displacements or local ambiguities from, e.g., textureless or reflective surfaces. Consider Fig. 1 (top) which shows an image from the KITTI 2015 scene flow dataset [22]. Due to the large displacement between frames, the front wheel in the first frame appears similar to the back

[*] Joint first authors with equal contribution.

wheel in the second frame, resulting in wrong predictions. Furthermore, the small amount of texture and the reflective car surface complicate the matching task.

In this paper, we propose to exploit recognition to facilitate this problem. In particular, we investigate the benefits of semantic grouping and fine-grained geometric recognition for this task as illustrated in Fig. 1. We will formulate the output of the recognition task as a new constraint, besides standard constraints such as local appearance and 3D motion-smoothness within an image, in a CRF-based framework.

For semantic grouping, we consider two scenarios: i) a bounding box around the visible part of each semantic instance, and ii) a pixel-wise segmentation of the visible part of each semantic instance. While the latter output provides a more detailed mask of the object, and hence may be more beneficial for the scene flow task, it is also a harder task to solve, and hence may contain segmentation errors. While these cues do not provide additional geometric evidence, they constrain the space of possible rigid body motions: pixels which are grouped together are likely to move as a single rigid object in the case of vehicles. We integrate both cues into the scene flow task by enforcing consistency between neighboring views (either in time or space). In short, a pixel within an instance region (bounding box or segment) in one frame should be mapped to an instance in the other frame. Furthermore, all pixels within an instance should move as one rigid entity.

To obtain the bounding box or segmentation masks, we utilize an existing state-of-the-art approach - the multi-task network cascade (MNC) from Dai et al. [6]. In contrast to [6], we achieve improved outputs, by providing a dense depth map as additional input to the network. To train the CNN, we annotated 400 stereo images from the KITTI 2015 benchmark of Menze et al. [22] using in-house annotators.

For fine-grained geometric recognition we train a new convolutional neural network (CNN) on 2D instance segmentations of MNC which predicts the continuous 3D object parts, also known as 3D object coordinates [3, 4, 25, 33, 34], for each pixel inside the instance mask. The 3D object coordinates specify the relative 3D location of an object's surface point with respect to the object's local coordinate frame as shown in Fig. 1 (bottom-right) with illustrative colors. Thus, they provide a detailed geometric cue for matching pixels in neighboring frames, even in the presence of large displacements.

The fine-grained geometric recognition and the semantic grouping have, certainly, a different trade-off between modeling-power versus prediction accuracy. If the recognition task was to be solved perfectly well, the continuous object part would be the strongest geometric cue, followed by the segmentation mask and the bounding box. On the other hand, as we will see experimentally, the continuous object parts are most challenging to predict precisely, followed by the segmentation mask and the bounding box. Given this trade-off, the key question addressed in this work is:

Which level of recognition is most beneficial for the scene flow task, when combining different cues, i.e. local appearance, 3D motion-smoothness and recognition-based geometric constraint, in a CRF-based framework?

Our CRF framework is based on [22] and termed Instance Scene Flow. We validate the benefits of recognition cues for scene flow on the challenging KITTI 2015 benchmark [22]. Firstly, we conduct a detailed ablation study for the three levels of recognition granularity, i.e. 2D bounding box, 2D segmentation mask, and continuous 3D object parts. From this study, we conclude that the instance segmentation cue is by far strongest, in our setting. Secondly, we show that our Instance Scene Flow method significantly boosts the scene flow accuracy, in particular in challenging foreground regions. Alongside, we obtain the lowest overall test errors amongst all methods at time of submission. Our code and datasets will be made available upon publication.

In short, our **contributions** are:

- A new 3D scene flow method, leveraging recognition-based geometric cues to achieve state-of-the-art performance on the challenging KITTI 2015 scene flow benchmark, at the time of submission.
- A detailed ablation study of the importance of recognition granularity, from coarse 2D bounding boxes over 2D instance segmentations to fine-grained 3D object part predictions, within our scene flow framework.
- High-quality, instance-level annotations of 400 stereo images from the KITTI 2015 benchmark [22].

2. Related Work

In the following, we review the most related works on *image-based* scene flow estimation, semantic priors and object coordinates. For an overview of RGB-D methods (e.g., using the Kinect), we refer to [9, 13, 14, 29, 43, 43].

Scene Flow: Following the work of Vedula et al. [36, 37] several approaches formulate 3D scene flow estimation as a variational optimization problem [2, 15, 28, 35, 39, 42]. Unfortunately, coarse-to-fine variational optimization suffers when displacements are large. Thus, slanted-plane models [44, 45] have recently been introduced [20, 22, 24, 38, 40] which gain their robustness by decomposing the scene into a collection of rigidly moving planes and exploiting discrete-continuous optimization techniques.

While these methods have demonstrated impressive performance on the challenging KITTI benchmark [10, 22], they fail to establish correspondences in textureless, reflective or fast-moving regions due to violations and ambiguities of the data term and weak prior assumptions. In this pa-

per, we propose to approach this problem using fine-grained instance recognition and 3D geometry information, resulting in significant accuracy gains.

Semantic Priors: Several works have considered semantic information for stereo or optical flow estimation. Güney et al. [12] presented a model for stereo estimation where 3D object hypotheses from a CAD model database are jointly estimated with the disparity map. Hur et al. [16] proposed a model for joint optical flow and semantic segmentation. In particular, they classify the scene into static and dynamic regions and introduce a constraint which measures how well the homography of a superpixel meets the epipolar constraint. Sevilla-Lara et al. [30] used semantic segmentation to identify object instances and combine per-object layered flow predictions [31] with DiscreteFlow [23]. Bai et al. [1] proposed a model which first identifies car instances and then predicts each rigidly moving component in the scene with a separate epipolar flow constraint.

While existing methods leverage recognition to aid either reconstruction or motion estimation, in this paper we consider recognition for the 3D scene flow problem, i.e., the combination of the two. In contrast to flow-only methods [1] that need to search for correspondences along epipolar lines, our method exploits the semantic cues as well as the geometry which allows us to estimate the rigid motion between frames more robustly and leads to significantly improved results compared to all baselines. Furthermore, we are (to the best of our knowledge) the first to investigate the impact of recognition granularity on the task, ranging from coarse 2D boxes to fine grained 3D object coordinates predictions.

3D Object Coordinates: Continuous 3D object parts, also known as 3D object coordinates, have so far mainly been leveraged in the context of 3D pose estimation [3, 4, 25, 33], camera re-localization [34] and model-based tracking [19]. The typical approach is to train a random forest for predicting instance-specific object probabilities and 3D object coordinates with respect to a fixed local coordinate system. These predictions are used to fit a 3D model of the known object, resulting in the 3D object pose.

In this work, we explore the possibility of using object coordinates as a continuous labeling for the surface points of the object.

3. Method

This section describes our approach to 3D scene flow estimation. The overall work flow of our approach is illustrated in Fig. 2. Given the 4 RGB images we extract 3D points (XYZ) for each pixel in camera coordinate system using a stereo method (see Section 4 for details). Based on the RGB and XYZ values, we train a multi network cascade (MNC) [6] to predict 2D bounding boxes and 2D instance segmentations. We train a CNN to predict object coordinates for car instances. Finally, we integrate the bounding box, instance and object coordinates cues into a slanted-plane formulation and analyze the importance of each cue for the scene flow estimation task. The remainder of this section is structured as follows.

As our goal is to analyze the impact of different levels of recognition granularity, we first describe the inputs to our method in Section 3.1 and Section 3.2. In particular, we are interested in improving scene flow estimation of vehicles which are challenging due to their large motion and non-lambertian appearance. In Section 3.3, we finally show how these predictions can be integrated into a CRF model for 3D scene flow estimation.

3.1. 2D Bounding Boxes and Instances

As discussed before, recognizing and segmenting objects is imperative for our approach. For this task, we use the state-of-the-art Multi-task Network Cascades (MNC) proposed by Dai et al. [6] to obtain bounding boxes and segmentation masks of all cars present in the scene. Unlike the standard MNC framework which operates on RGB images, we provide the network with an RGB-XYZ image, where XYZ denotes the 3D scene coordinates, i.e., the 3D location of every pixel in the scene in camera coordinates. The 3D location for each pixel is computed from disparity maps, see Section 4 for details.

We pre-train the network using Pascal VOC and fine-tune it using 3200 coarse annotations of KITTI [10] provided by [5]. We obtained 200 images with 1902 pixel-accurate instance annotations from the KITTI 2015 Stereo benchmark [22] using in-house annotators. We further refined the model with these fine annotations. The final accuracy (IoU-50%) on the validation set of [5] is 83% as compared to 78% when using only RGB images.

3.2. 3D Object Coordinates

3D object coordinates specify the 3D location of the surface of an object with respect to a local coordinate frame, see Fig. 1 for an illustration. They can be viewed as a fine grained unique geometric labeling of the object's surface which is independent of the viewpoint and can be used to establish correspondences between frames, which we expect to be more robust to appearance changes than correspondences based on sparse feature matching.

While random forests have been used for estimating object coordinates when the target instance is known [3, 4, 19, 34], we found that a CNN-based approach leads to significantly better object coordinates predictions as also evidenced by our experiments in Section 4. We use a modified version of the encoder-decoder style CNN proposed in [32] for estimating the object coordinates at each pixel. As above, the input to the CNN is an RGB-XYZ image as

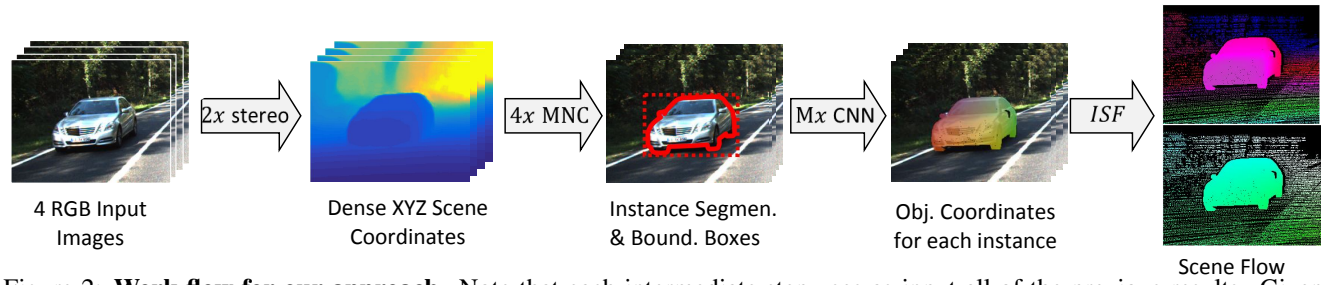

Figure 2: **Work flow for our approach**. Note that each intermediate step uses as input all of the previous results. Given the four RGB input images (t/t+1, left/right) we compute 3D points (XYZ) for each pixel. For each of the four RGB,XYZ image-blocks we obtain instance segmentations, alongside bounding boxes. The M instances are processed individually to obtain object coordinates for each instance, using our object coordinates CNN. Finally, all this information is integrated into our Instance Scene Flow method (ISF) to produce the final output.

well as the instance prediction from the MNC. The output of the CNN is a 3 layer regression which stores the X, Y and Z coordinates for each point of the input. We refer to the architecture of our CNN in supplementary material. The encoder part comprises a set of 5 convolutional layers with a stride of 2 in each layer, followed by a set of 3 fully connected layers. The decoder part has 5 deconvolutional layers followed by a 3D regression layer which predicts the 3D object coordinates.

3.3. Scene Flow Model

We now describe our scene flow model which is based on [22] but adds two new components to it, in particular an instance sensitive appearance term and a term which encourages object coordinates to align across frames. For making the paper self-contained, we specify the full model.

Notation: We first describe the notation of the inputs to the scene flow model which are pre-computed as described in the previous section and fixed during inference. Let $\mathcal{V} = \{0, 1, 2, 3\}$ denote the set of views as illustrated in Fig. 3 and let $\mathbf{I}_v \in \mathbb{R}^{w \times h \times 3}$ denote the input image corresponding to each view. For each view $v \in \mathcal{V}$, we compute the following information: first, our MNC predicts instance label maps $\mathbf{M}_v \in \{0, \dots, |M_v|\}^{w \times h}$ which determine the predicted semantic instance label for each pixel in each view. In our experiments, these maps are either coarse 2D bounding box segmentations or more accurate 2D instance segmentations which are both computed via MNC. Background pixels are assigned the label $\mathbf{M}_v(\mathbf{p}) = 0$ and foreground pixels are assigned positive labels. Note that instance labels do not correspond across frames as the correspondence is unknown a-priori and needs to be inferred jointly with the scene flow. Furthermore, we denote the 3D object coordinates predicted by the network with $\mathbf{C}_v \in [-1, 1]^{w \times h \times 3}$.

We now describe the parameters of our model which are optimized during inference. Let $\mathcal{S}$ denote the set of superpixels in the reference view and $\mathcal{O}$ denote the set of objects in the scene. Each superpixel $i \in \mathcal{S}$ is associated with a region $\mathcal{R}_i$ in the reference image and a variable $\mathbf{s}_i = (\mathbf{n}_i, k_i)^\mathsf{T}$, where $\mathbf{n}_i \in \mathbb{R}^3$ describes a plane in 3D via $\mathbf{n}_i^\mathsf{T}\mathbf{x} = 1$. Further, let $k_i \in \{0, \dots, |\mathcal{O}|\}$ index the object which the superpixel is associated with. Here $|\mathcal{O}|$ denotes an upper bound on the number of objects we expect to see, and $k = 0$ refers to the background object with $k > 0$ to other traffic participants. Each object $j \in \mathcal{O}$ is associated with a variable $\mathbf{o}_j \in SE(3)$ which describes its rigid body motion in 3D. Each superpixel associated with object j inherits its rigid motion parameters $\mathbf{o}_j \in SE(3)$. In combination with the plane parameters $\mathbf{n}_i$, this fully determines the 3D scene flow at each pixel inside the superpixel.

Energy Model: Given the left and right input images of two consecutive frames (Fig. 3), our goal is to infer the 3D geometry of each superpixel $\mathbf{n}_i$ in the reference view, the association to objects k_i and the rigid body motion of each object $\mathbf{o}_j$. We formulate the scene flow estimation task as an energy minimization problem comprising data, smoothness and instance terms:

$$\hat{\mathbf{s}}, \hat{\mathbf{o}} = \underset{\mathbf{s},\mathbf{o}}{\operatorname{argmin}} \underbrace{\varphi(\mathbf{s}, \mathbf{o})}_{\text{data}} + \underbrace{\psi(\mathbf{s})}_{\text{smooth.}} + \underbrace{\chi(\mathbf{s}, \mathbf{o})}_{\text{instance}} \quad (1)$$

We summarize all variables involved in the optimization with $\mathbf{s} = \{\mathbf{s}_i | i \in \mathcal{S}\}$ and $\mathbf{o} = \{\mathbf{o}_i | i \in \mathcal{O}\}$. For clarity of exposition we omit all weight parameters of the model.

Data Term: Our data term encodes the assumption that corresponding points across all images should be similar in appearance:

$$\varphi(\mathbf{s}, \mathbf{o}) = \sum_{i \in \mathcal{S}} \sum_{\mathbf{p} \in \mathcal{R}_i} \sum_{v \in \mathcal{V}} \varphi_v^{\mathrm{D}}(\mathbf{p}, \mathbf{q}) \quad (2)$$

$$\mathbf{q} = \underbrace{\mathbf{K}\left(\mathbf{R}_v(\mathbf{o}_{k_i}) - \mathbf{t}_v(\mathbf{o}_{k_i})\,\mathbf{n}_i^\mathsf{T}\right)\mathbf{K}^{-1}}_{\text{homography (view } 0 \to \text{view } v)}\,\mathbf{p} \quad (3)$$

Here, $\mathcal{V} = \{1, 2, 3\}$ denotes the set of target views and $\mathbf{q}$ is the location of pixel $\mathbf{p}$ in reference view 0 mapped into the

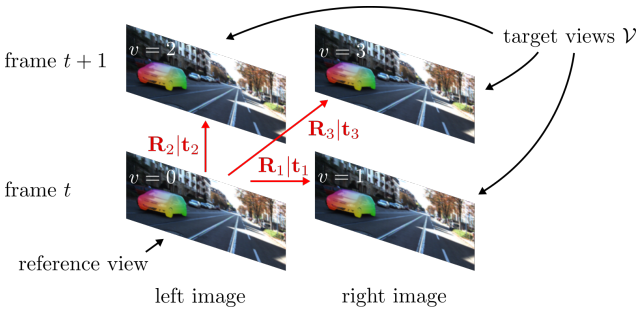

Figure 3: **Geometric Relationship** between reference and target views. Pixels in the reference view are mapped to a pixels in a target view via their depth and rigid body motion.

target view $v \in \mathcal{V}$ according to the calibration matrix $\mathbf{K}$, the rigid body motion $\mathbf{R}_v|\mathbf{t}_v$ of the corresponding object $\mathbf{o}_{k_i}$ and the plane parameters of the associated superpixel $\mathbf{n}_i$. See Fig. 3 for an illustration.

The data cost $\varphi_v^{\mathrm{D}}(\mathbf{p}, \mathbf{q})$ compares the appearance at pixel $\mathbf{p}$ in reference image 0 with the appearance at pixel $\mathbf{q}$ in the target view $v \in \mathcal{V}$. In our experiments, we use Census descriptors [46] which are robust to simple photometric variations [22, 41, 45]. To guide the optimization process and overcome local minima we additionally add a robust ℓ_1 loss to φ_v^{D}. This loss measures the difference with respect to sparse DiscreteFlow correspondences [23] for the flow terms ($v = 2, 3$) and depth estimates from SPS-stereo [45] for the stereo term ($v = 1$).

Smoothness Term: The smoothness term encourages coherence of adjacent superpixels in terms of depth, orientation and motion. It decomposes as

$$\psi(\mathbf{s}) = \gamma_{ij}^{\mathrm{S}} \sum_{i \sim j} \psi_{ij}^{\mathrm{G}}(\mathbf{n}_i, \mathbf{n}_j) + \psi_{ij}^{\mathrm{M}}(\mathbf{s}_i, \mathbf{s}_j) \tag{4}$$

with the following geometry (G) and motion (M) terms:

$$\begin{aligned} \psi_{ij}^{\mathrm{G}}(\mathbf{n}_i, \mathbf{n}_j) &= \sum_{\mathbf{p} \in \mathcal{B}_{ij}} \rho\left(d(\mathbf{n}_i, \mathbf{p}) - d(\mathbf{n}_j, \mathbf{p})\right) \\ &\quad + \rho\left(1 - |\mathbf{n}_i^{\mathsf{T}} \mathbf{n}_j| / (\|\mathbf{n}_i\| \|\mathbf{n}_j\|)\right), \\ \psi_{ij}^{\mathrm{M}}(\mathbf{s}_i, \mathbf{s}_j) &= \gamma_{ij}^{\mathrm{M}}(\mathbf{n}_i, \mathbf{n}_j)\, [k_i \neq k_j]. \end{aligned}$$

Here, $d(\mathbf{n}, \mathbf{p})$ denotes the disparity of plane $\mathbf{n}$ at pixel $\mathbf{p}$ in the reference image, $\mathcal{B}_{ij}$ is the set of shared boundary pixels between superpixel i and superpixel j, and ρ is the robust ℓ_1 penalty. The instance-sensitive weight γ_{ij}^{S} is defined as

$$\gamma_{ij}^{\mathrm{S}} = 1 - \beta_{\mathrm{S}} \cdot [(i, j) \in \mathcal{M}] \tag{5}$$

where $\mathcal{M}$ denotes the set of adjacent superpixel pairs where exactly one of the superpixels lies on an object instance. $\beta \in [0, 1]$ is a hyper-parameter which weighs down the costs of discontinuities for adjacent superpixels in $\mathcal{M}$. Note that this weighting is only possible in the presence of instance predictions.

The geometry-sensitive motion weight is defined as

$$\begin{aligned} \gamma_{ij}^{\mathrm{M}}(\mathbf{n}_i, \mathbf{n}_j) = \exp\left(-\frac{\lambda}{|\mathcal{B}_{ij}|} \sum_{\mathbf{p} \in \mathcal{B}_{ij}} \left(d(\mathbf{n}_i, \mathbf{p}) - d(\mathbf{n}_j, \mathbf{p})\right)^2\right) \\ \times |\mathbf{n}_i^{\mathsf{T}} \mathbf{n}_j| / (\|\mathbf{n}_i\| \|\mathbf{n}_j\|) \end{aligned}$$

encouraging motion boundaries to align with 3D folds and discontinuities rather than within smooth surfaces.

Instance Term: The instance term $\chi(\mathbf{s}, \mathbf{o})$ measures the compatibility of appearance and part-labeling induced by the 3D object coordinates when warping the detected instances into the next frame. It takes the following form

$$\chi(\mathbf{s}, \mathbf{o}) = \sum_{i \in \mathcal{S}} \sum_{\mathbf{p} \in \mathcal{R}_i} \sum_{v \in \mathcal{V}} \chi_v^{\mathrm{I}}(\mathbf{p}, \mathbf{q}) \tag{6}$$

with

$$\begin{aligned} \chi_v^{\mathrm{I}}(\mathbf{p}, \mathbf{q}) = \quad & [\mathbf{M}_0(\mathbf{p}) = 0 \vee \mathbf{M}_v(\mathbf{q}) = 0] \cdot \lambda \; + \\ & [\mathbf{M}_0(\mathbf{p}) > 0 \wedge \mathbf{M}_v(\mathbf{q}) > 0] \cdot \left(\chi_v^{\mathrm{A}}(\mathbf{p}, \mathbf{q}) + \chi_v^{\mathrm{L}}(\mathbf{p}, \mathbf{q})\right) \end{aligned} \tag{7}$$

Here, $\mathbf{q}$ is calculated as in Eq. 3 and the appearance (A) potential and the part labeling (L) potential are defined as

$$\chi_v^{\mathrm{A}}(\mathbf{p}, \mathbf{q}) = \|\mathbf{I}_0(\mathbf{p}) - \mathbf{I}_v(\mathbf{q})\|_1 \tag{8}$$

$$\chi_v^{\mathrm{L}}(\mathbf{p}, \mathbf{q}) = \|\mathbf{C}_0(\mathbf{p}) - \mathbf{C}_v(\mathbf{q})\|_1 \tag{9}$$

and measures the difference in appearance $\mathbf{I}$ and 3D object coordinates $\mathbf{C}$ between image location $\mathbf{p}$ in the reference view and $\mathbf{q}$ in the target view, respectively. While the data term in Eq. 2 also evaluates appearance, we found that the Census descriptors work well mostly for textured background regions. In contrast, it returns noisy and unreliable results in the presence of textureless, specular surfaces such as on cars. However, as evidenced by our experiments, including an additional ℓ_1 constraint on appearance for instances leads to significantly better estimates in those cases. This observation is in accordance with recent works on direct visual odometry and SLAM [7, 8, 27] which use similar measures to reliably estimate the camera pose in weakly textured environments. In contrast to them, here we exploit this constraint to estimate the relative pose of each individual weakly textured object in the scene.

The intuition behind this term is as follows: when warping the recognized instances from the reference frame into the target frame according to the estimated geometry and motion, the appearance as well as the part labeling induced by the object coordinates should agree. The term $[\mathbf{M}_0(\mathbf{p}) > 0 \wedge \mathbf{M}_v(\mathbf{q}) > 0]$ ensures that these constraints are only evaluated if both the reference and the target pixel belong to a detected instance (i.e., when $\mathbf{M} > 0$). However,

Algorithm 1 Optimization

Input: $\mathbf{I}_v$, $\mathbf{M}_v$, $\mathbf{C}_v$ for $v \in \mathcal{V}$
Initialize $\mathbf{s}$ and $\mathbf{o}$ as described in "Initialization"
for all iterations $n = 1, \ldots, N$ **do**
  for all $i \in \mathcal{S}$ **do**
    Draw samples for $\mathbf{s}_i$ (Gaussian)
  for all $j \in \mathcal{O}$ **do**
    Draw samples for $\mathbf{o}_j$ (MCMC)
  Run TRW-S [18] on discretized problem
Output: $\hat{\mathbf{s}}$, $\hat{\mathbf{o}}$

as both χ_v^{A} and χ_v^{L} are positive terms the model prefers to associate instances with background regions which incurs no additional cost compared to associating instances with instances. We therefore incorporate an additional term which yields an appropriate bias λ and favors the association of instances.

Optimization: For optimizing Eq. 1, we leverage max-product particle belief propagation with TRW-S [18] as proposed in [22]. At each iteration this optimization algorithm discretizes the continous variables by drawing samples around the current maximum-a-posteriori (MAP) solution and runs TRW-S until convergence before resampling. However, we found that this approach gets easily trapped in local minima, in particular due to the highly non-convex energy function associated with the appearance of the foreground objects.

We thus modify their sampling strategy as shown in Algorithm 1, which leads to better results. While the original algorithm [22] discretizes the continuous variables (in particular the rigid body motions $\mathbf{o}$) by sampling from a Gaussian centered at the current MAP estimate, we create samples by running a Markov chain based on the appearance term in Eq. 6 for each object individually. More specifically, our sampling energy warps all pixels inside an instance based on the predicted depth and the rigid body motion of the sample, and measures the photoconsistency between reference and target views using the ℓ_1 norm. This "informed" way of sampling ensures that TRW-S has access to high-quality samples compared to noisy estimates drawn around the current MAP solution. For sampling the geometry variables (i.e., normals), we follow [22].

The hyper parameters in our model are estimated using block coordinate descent on a train/validation split.

Initialization: As we are presented with a complex NP hard optimization problem in Eq. 1, initialization of parameters is an important step. We describe the details of our initialization procedure in the following.

We initialize the geometry parameters using dense disparity maps and the object hypotheses/motion variables using sparse flow predictions and the predicted bounding boxes/instances. More specifically, we robustly fit all superpixel parameters to the disparity estimates and aggregate all sparse flow estimates for each instance to robustly fit a rigid body motion to it via RANSAC. We refer to Section 4 for details on the particular choice of input algorithms.

For the instance-based algorithms, we aggregate the sparse flow estimates directly based on the instance masks and robustly fit a rigid body motion to it via RANSAC. Given the high quality of the instance predictions, this leads to very robust initializations. For baselines which do not leverage recognition we follow [22] and initialize objects based on clustering sparse 3D scene flow estimates which disagree with the background motion. Based on this clustering, we initialize the associations and poses using robust fitting using RANSAC. We refer the reader to [22] for further details.

While we have also experimented with color histograms and pose prediction to support the assignment of instances across frames, we found that aggregating sparse flow vectors [23] and dense geometry [21] allows for correctly associating almost all objects. We thus don't use such an additional term in the model, which, however, could be easily integrated to solve more challenging association problems as present in the KITTI 2015 scene flow benchmark.

Runtime: Our MATLAB implementation with C++ wrappers requires on average 40 seconds for each of the 10 iterations of the optimization described in Algorithm 1. This leads to a runtime of 7 minutes for processing one scene (4 images) on a single i7 core running at 3.0 Ghz. In addition, the inputs to our methods namely, dense disparity maps, sparse flow predictions and predicted bounding boxes/instances require on average 3 minutes for processing one scene, leading to a total runtime of 10 minutes.

4. Experimental Evaluation

4.1. Effect of recognition granularity

In this section, we study the impact of different levels of recognition granularity for estimating the 3D scene flow of dynamic (i.e., foreground) objects. In addition to the recognition cues, we use sparse optical flow from sparse DiscreteFlow correspondences [23] and dense disparity maps from SPS-stereo [45] for both rectified frames. We obtain the superpixel boundaries using StereoSLIC [44]. Table 1 provides a quantitative comparison of the performance of *OSF* [22] (no recognition input), *ISF-BBox* (2D bounding boxes as recognition input), *ISF-SegMask* (2D instance segmentations as recognition input) and *ISF-SegMask-ObjCoord* (2D instance segmentations in conjunction with 3D object coordinates as recognition input) on our validation set (which is a subset of the KITTI 2015 [22] scene flow training set). We report the error with respect to four different outputs: disparities in the first and second frame (D1,D2), optical flow

	D1			D2			Fl			SF		
	bg	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*
OSF	4.00	8.86	4.74	5.16	17.11	6.99	6.38	20.56	8.55	7.38	24.12	9.94
ISF-BBox	3.94	8.81	4.69	5.10	10.77	5.97	6.46	12.90	7.44	7.42	17.11	8.90
ISF-SegMask	4.06	7.97	4.66	5.26	9.20	5.86	6.72	10.78	7.34	7.74	14.60	8.79
ISF-SegMask-ObjCoord	4.08	7.98	4.68	5.27	9.20	5.87	6.72	10.84	7.35	7.75	14.66	8.80
ISF-SegMask-CNNDisp	3.55	3.94	3.61	4.86	4.72	4.84	6.36	7.31	6.50	7.23	8.72	7.46

Table 1: **Quantitative results from ablation study on KITTI 2015 Validation Set.** We report the disparity (D1,D2), flow (Fl) and scene flow (SF) error averaged over our validation set for *OSF* [22] (no recognition input), *ISF-BBox* (bounding box input), *ISF-SegMask* (segmentation input) and *ISF-SegMask-ObjCoord* (segmentation + object coordinates input). Additionally, we also report results of *ISF-SegMask-CNNDisp* (ISF-SegMask with higher quality CNN based disparity input).

(FL) and scene flow (SF).

Our results indicate that recognition (both 2D bounding box and 2D instance segmentation) provides large improvements for optical flow estimation (and in turn scene flow estimation) on foreground parts of the scene. Note that we do not tackle the recognition of static background objects in this paper which are estimated relatively well without such priors. Furthermore, we note that instance segmentations as input improve performance in particular at the boundary of objects. We attribute this effect to the more fine grained nature of the 2D segmentation input compared to using rough 2D bounding boxes input. We remark that for evaluation, the KITTI 2015 benchmark considers only a subset of the cars in the scene as foreground, specifically dynamic cars within a threshold distance to the camera. In contrast, as defined in section 3.3, our scene flow estimation model considers all car instances detected by our CNN as foreground. Figure 4 provides a qualitative comparison illustrating the differences between the scene flow errors of methods employing different levels of recognition granularity. We encourage the reader to have a look at the first section of our supplementary material for additional examples.

Moreover, we observe that 3D object coordinates do not increase performance beyond 2D instance segmentations (and hence yield the same estimates, i.e., the weight of the object coordinate terms are zero after optimization). We attribute this to the quality of the state-of-the-art object coordinate predictions. Specifically, the accuracy of 3D object coordinate predictions from state-of-the-art CNN-based methods is below what is required to further improve 3D scene flow estimation due to the high level of accuracy requested by current benchmarks (i.e., 3 pixels error in KITTI). We refer the reader to the second section of our supplementary material for a detailed analysis of why 3D object coordinate predictions do not provide any further gains.

Experiments with CNN based disparity as input: Furthermore, we evaluated our method with the optimal level of recognition granularity selected based on the ablation study (*ISF-SegMask*) on higher quality disparity inputs computed from DispNetC [21] and MC-CNN-acrt [47]. In particular, combining DispNetC results for instance pixels and MC-CNN-acrt results for the rest performed best on our validation set. We report the results of our top performing method *ISF-SegMask-CNNDisp* on the validation set in Table 1.

4.2. Results on the KITTI Benchmark

In this section, we present results of our top performing method *ISF-SegMask-CNNDisp* evaluated on the KITTI 2015 scene flow evaluation server. Table 2 compares the performance of our method with other leading methods on the benchmark: PRSM [41], OSF [22] and OSF-TC [26]. We obtain state-of-the-art performance at the time of submission, even including anonymous submissions. Notably, our method also outperforms methods which use more than two temporal frames (OSF-TC [26] and PRSM [41]). Figure 5 shows scene flow errors for examples where other leading methods fail to effectively estimate scene flow in foreground regions. Scene flow estimation is challenging for this region as the texture-less car undergoes large displacement and is occluded in the second frame. Our method employing recognition cues performs comparatively better than other methods. Without recognition cues, only very few matches could be established in those regions and most of them would be wrong. We encourage the reader to have a look at our supplementary material for additional qualitative comparisons of our method to other state-of-the-art methods.

4.3. 3D object coordinates prediction

We model our encoder-decoder network for 3D object coordinate prediction in Caffe [17]. The network is trained by minimizing a robust and smooth Huber loss [11] using the Adam solver with a momentum of 0.9 and learning rate of 1e-5. Furthermore, in order to compare the performance of our CNN model with existing state-of-the-art approaches for predicting the object coordinates, we train a random forest [3] with 3 trees and maximum depth of 64. We use the RGB image and the depth map as input feature and sample 3 million random points during training. We find the quality of 3D object coordinate predictions significantly better using our CNN architecture with an average Euclidean er-

	D1			D2			Fl			SF		
	bg	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*	*bg*	*fg*	*bg+fg*
PRSM* [40]	3.02	10.52	**4.27**	5.13	15.11	6.79	**5.33**	13.40	6.68	6.61	20.79	8.97
OSF+TC* [26]	4.11	9.64	5.03	5.18	15.12	6.84	5.76	13.31	7.02	7.08	20.03	9.23
OSF [22]	4.54	12.03	5.79	5.45	19.41	7.77	5.62	18.92	7.83	7.01	26.34	10.23
ISF-SegMask-CNNDisp	**4.12**	**6.17**	4.46	**4.88**	**11.34**	**5.95**	5.40	**10.29**	**6.22**	**6.58**	**15.63**	**8.08**

Table 2: **Quantitative Results on the KITTI 2015 Scene Flow Evaluation Server.** This table shows the disparity (D1/D2), flow (Fl) and scene flow (SF) errors averaged over all 200 test images for our method in comparison to other leading methods (OSF [22], OSF-TC [26] and PRSM [40]) on the KITTI 15 benchmark. Methods with * use more than two temporal frames.

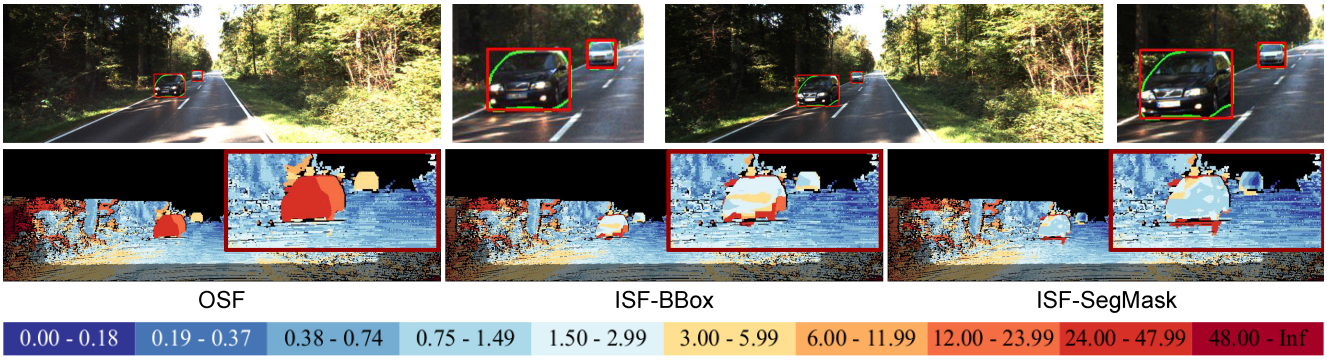

Figure 4: **Qualitative Results from Ablation Study on KITTI 2015 Validation Set.** The top row shows predicted masks and bounding boxes which form the input to our method overlaid onto the left camera image at time t and $t+1$. Each figure in the bottom row (from left to right) shows scene flow error of *OSF* [22] (no recognition input), *ISF-BBox* (bounding box input) and *ISF-SegMask* (segmentation input) using the color scheme depicted in the legend. The red box shows zoom-ins.

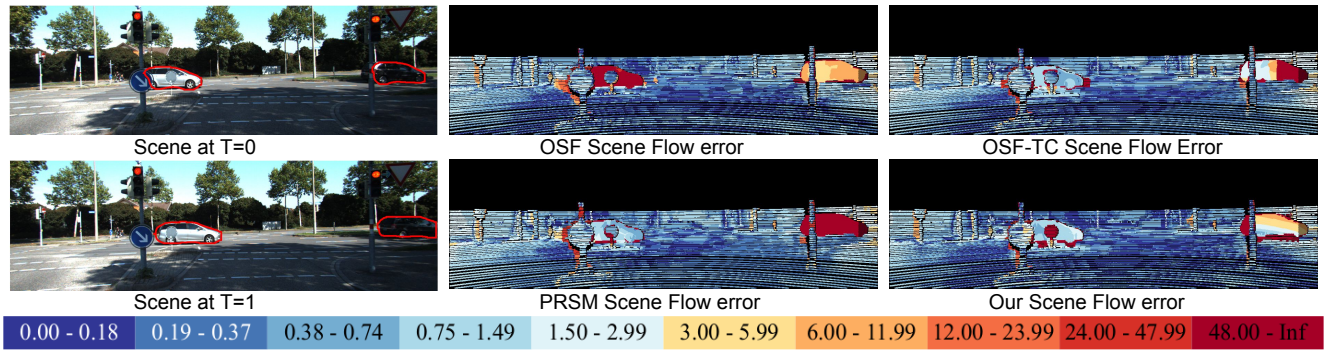

Figure 5: **Qualitative Comparison on KITTI-15 Test Set.** The first column shows the input images, followed by scene flow error maps of OSF [22], OSF-TC [26], PRSM [40] and our method using the color scheme depicted in the legend.

ror of 0.6 meters in comparison to an error of 2.89 meters using the random forest method. We attribute the lower error of the CNN based predictions to the network's ability to generalize well to cars with high intra class variation. We refer the reader to the last section of the supplementary material for a detailed description of the CNN architecture we employed and a comparison of the quality of our 3D object coordinate predictions to other state-of-the-art methods.

5. Conclusion

In this work, we studied the impact of different levels of recognition granularity on the problem of estimating scene flow for dynamic foreground objects. Our results indicate that recognition cues such as 2D bounding box and 2D instance segmentation provide large improvements on foreground parts of the scene in the presence of challenges such as large displacement or local ambiguities. Furthermore, we showed that our method achieves state-of-the-art performance on the challenging KITTI scene flow benchmark.

Acknowledgments This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (grant agreement No 647769).

References

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

[2] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. *International Journal of Computer Vision (IJCV)*, 101(1):6–21, 2013.

[3] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[4] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] L.-C. Chen, S. Fidler, A. L. Yuille, and R. Urtasun. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *arXiv.org*, 1607.02565, 2016.

[8] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: large-scale direct monocular SLAM. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[9] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6D-Vision: fusion of stereo and motion for robust environment perception. In *Proc. of the DAGM Symposium on Pattern Recognition (DAGM)*, 2005.

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

[11] R. B. Girshick. Fast R-CNN. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015.

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

[13] E. Herbst, X. Ren, and D. Fox. RGB-D flow: Dense 3D motion estimation using color and depth. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2013.

[14] M. Hornacek, A. Fitzgibbon, and C. Rother. SphereFlow: 6 DoF scene flow from RGB-D pairs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2007.

[16] J. Hur and S. Roth. Joint optical flow and temporally consistent semantic segmentation. In *Proc. of the European Conf. on Computer Vision (ECCV) Workshops*, 2016.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. of the International Conf. on Multimedia (ICM)*, 2014.

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

[19] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-dof model based tracking via object coordinate regression. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2014.

[20] Z. Lv, C. Beall, P. Alcantarilla, F. Li, Z. Kira, and F. Dellaert. A continuous optimization approach for efficient and accurate scene flow. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

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

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

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

[24] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *Proc. of the ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[25] F. Michel, A. Krull, E. Brachmann, M. Y. Yang, S. Gumhold, and C. Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2015.

[26] M. Neoral and J. Šochman. Object scene flow with temporal consistency. In *Proc. of the Computer Vision Winter Workshop (CVWW)*, 2017.

[27] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: dense tracking and mapping in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011.

[28] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision (IJCV)*, 72(2):179–193, 2007.

[29] J. Quiroga, T. Brox, F. Devernay, and J. L. Crowley. Dense semi-rigid scene flow estimation from RGB-D images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

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

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

[32] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network.

In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

[33] J. Taylor, J. Shotton, T. Sharp, and A. W. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[34] J. P. C. Valentin, M. Nießner, J. Shotton, A. W. Fitzgibbon, S. Izadi, and P. H. S. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[35] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010.

[36] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[37] S. Vedula, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 27(3):475–480, 2005.

[38] C. Vogel, S. Roth, and K. Schindler. View-consistent 3D scene flow estimation over multiple frames. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[39] C. Vogel, K. Schindler, and S. Roth. 3D scene flow estimation with a rigid motion prior. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011.

[40] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2013.

[41] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision (IJCV)*, 115(1):1–28, 2015.

[42] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision (IJCV)*, 95(1):29–51, 2011.

[43] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2008.

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

[45] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2014.

[46] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 1994.

[47] J. Žbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)*, 17(65):1–32, 2016.