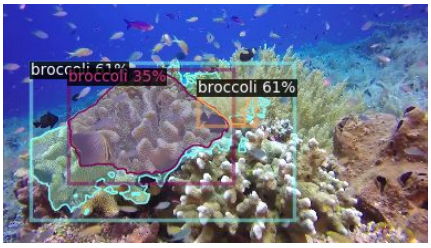
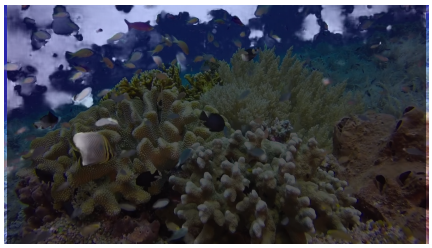
# Learning to Segment Rigid Motions from Two Frames

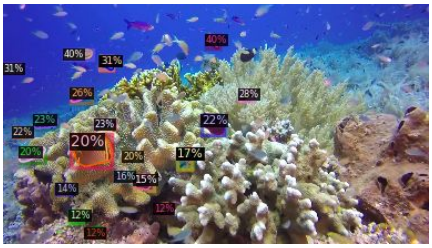
Gengshan Yang[1]*, Deva Ramanan[1,2]
[1]Carnegie Mellon University, [2]Argo AI
{gengshay, deva}@cs.cmu.edu

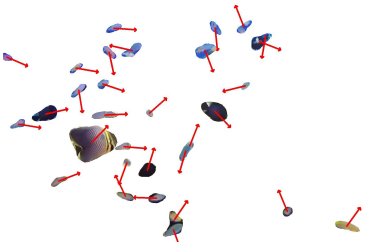

L: Optical flow R: Reference frame



(a) PointRend (trained on MSCOCO)


(b) Geometric (black: rigid background)


(c) Our rigid motion masks


(d) Our rigid motion (projected to 2D)

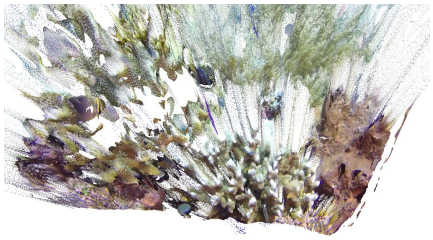

(e) Our triangulated depth (3D point clouds)

Figure 1: (a) Many data-driven segmentation methods heavily rely on appearance cues, and fail for novel test scenes. For instance, PointRend [25] trained on MSCOCO fails to detect coral reef fishes even with a low confidence threshold of 0.1. (b) On the other hand, geometric motion segmentation [5, 53] generalizes to novel appearance, but fails due to noisy flow inputs and degenerate motion configurations. (c)-(e) We propose a neural architecture powered by geometric reasoning that can decompose a scene into a rigid background and multiple moving rigid bodies, parameterized by 3D rigid transformations. It demonstrates generalization ability to novel scenes and robustness to noisy inputs as well as motion degeneracies. The predicted rigidity masks are shown useful for depth and scene flow estimation.

## Abstract

*Appearance-based object detectors achieve remarkable performance on common scenes, benefiting from a high-capacity model and massive annotated data, but tend to fail for scenarios lack of training data. Geometric motion segmentation algorithms, however, generalize to novel scenes, but have yet to achieve comparable performance to appearance-based ones, due to noisy motion estimations and degenerate motion configurations. To combine the best of both worlds, we propose a modular network, whose architecture is motivated by a geometric analysis of what independent object motions can be recovered from an egomotion field. It takes two consecutive frames as input and predicts segmentation masks for the background and rigidly moving objects. Our method achieves state-of-the-art performance for rigidity estimation on KITTI and Sintel. The inferred two-frame rigid motion masks also lead to a significant improvement for depth and scene flow estimation.*

## 1. Introduction

Autonomous agents such as self-driving cars need to be able to navigate safely in dynamic environments. Static environments are far easier to process because one can make use of geometric constraints (SFM/SLAM) to infer scene structure [15]. Dynamic environments require the fundamental ability to both segment moving obstacles and estimate their depth and speed [2]. Popular solutions include object detection or semantic segmentation [26]. While one can build accurate detectors for many categories of objects that

*Code will be available at github.com/gengshan-y/rigidmask.

are able to move, "being able to move" is not equivalent to "moving". For example, there is a profound difference between a parked car and an ever-so-slightly moving car (that is pulling out of parked location), in terms of the appropriate response needed from a nearby autonomous agent. Secondly, class-specific detectors rely heavily on appearance cues and categories present in a training set. Consider a trash can that falls on the street; current *closed-world* detectors will likely not be able to model all types of moving debris. This poses severe implications for safety in the open-world that a truly autonomous agent must operate [4].

**Problem formulation:** We follow historic work on motion-based perceptual grouping [23, 40, 47, 56] and segment moving objects without relying on appearance cues. Specifically, we focus on segmenting *rigid* bodies from *two frames*. We focus on two-frame because it is the minimal set of inputs to study the problem of motion segmentation, and in practice, perception-for-autonomy needs to respond immediately to dynamic scenes, e.g., an animal that appears from behind an occlusion. We focus on rigid body and its *3D* motion parameterizations because it's directly relevant for an autonomous agent acting in a 3D world. While dynamic scenes often contain nonrigid objects such as people, we expect that deformable objects may be modeled as a rigid body over short time scales, or decomposed into rigidly-moving parts [1, 8]. One example of our method decomposing a flying dragon into multiple rigid parts is shown in Fig. 2.

**Challenges:** Earlier work on rigid motion segmentation often makes use of geometric constraints arising from epipolar geometry and rigid transformations. However, there are several fundamental difficulties that plague geometric motion segmentation. First, epipolar constraints fail when scene motion is degenerate, for example, with close-to-planar structures or small camera translations [56]. Second, points moving along epipolar lines cannot be distinguished from the rigid background [60], which we discuss at length in Sec. 3.1. Third, geometric criteria are often not robust enough to noisy motion correspondences and camera egomotion estimates, which can lead to catastrophic failures in practice.

**Method:** We theoretically analyze ambiguities in 3D rigid motion segmentation, and resolve such ambiguities by exploiting recent techniques for upgrading 2D motion observation to 3D with optical expansion [58] and monocular depth cues [36]. To deal with noisy motion correspondences and degenerate scene motion, we design a convolutional architecture that segments the rigid background and an arbitrary number of rigid bodies from a given motion field. Finally, we parameterize the 3D motion of individual rigid bodies by fitting 3D rigid transformations.

**Contributions:** (1) We provide a geometric analysis of ambiguities in 3D rigid motion segmentation from 2D motion fields, and introduce a solution to address such ambiguities. (2) We propose a geometry-aware neural architecture for 3D rigid motion segmentation from two RGB frames. (3) The proposed method is generalizable to novel scenes as well as objects, resilient to different motion types, and robust to noisy motion observations. It achieves SOTA performance of rigid motion segmentation on KITTI and Sintel, and significantly improves the performance of downstream depth and scene flow estimation tasks.

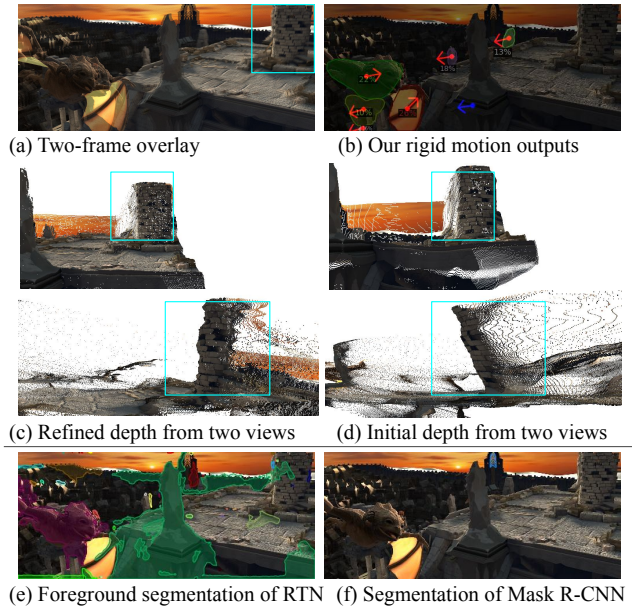



Figure 2: Results on Sintel sequence temple_2, frame 17-18. (a)-(b) Our method segments rigid motions and fits 3D rigid transformations over two frames. The blue and red arrows indicate the estimated motion of the rigid background and parts respectively. (c)-(d) An initial depth is refined by triangulating optical flow within each rigid motion mask. Note that the tower in the cyan rectangle is leaning in the initial MiDaS [36] depth, but "rectified" by our method. (e)-(f) Our method segments rigid objects more reliably than the prior two-frame rigidity estimation method [29] and generalizes to novel appearance compared to appearance-based detectors [20].

## 2. Related Work

**Geometric Motion Segmentation:** The problem of clustering motion correspondences into groups that follow a similar 3D motion model has been extensively studied in the past [44, 45, 47, 49, 56, 60]. However, prior methods either focus on theoretical analysis with toy data, or assume relatively simple scenes where long-term motion trajectories can be obtained by point tracking algorithms. Some recent work [5, 7, 53] tackles more complex scenarios with two-frame optical flow inputs, where geometric constraints, such as motion angle and plane plus parallax (P+P) [39] are considered as cues of "moving versus static". However, such geometric constraints are sensitive to noise in optical flow, and require a low outlier ratio even under a robust estimation framework [16]. Moreover, as we shall see in Sec. 3.1, the

(a) Two frame overlay (b) Motion-depth reconstruction (c) Sampson err. (white indicates moving)

Figure 3: Collinear motion ambiguity. (a) The input scene contains a dynamic object (the car in the lower left) moving parallel to camera translational direction. (b) One can triangulate motion correspondences assuming overall *rigidity* that places the moving car at an elevated height, which illustrates both (1) the commonality of this degenerate case [60] in urban navigation, and (2) one solution of using structural scene priors that do not allow for floating objects above the ground. (c) Due to such ambiguities, the 2D motion of the moving car is *consistent* with the camera egomotion, leaving it indistinguishable under classic motion segmentation metrics such as Sampson error [19].

prior solutions does not deal with several degenerate cases that appear important in real-world applications, including co-planar/co-linear motion [60] and camera motion degeneracy [45]. We address these problems by encoding geometric constraints into a modular neural network that is trained on a large synthetic dataset.

**Learning-Based Video Object Segmentation:** Segmenting salient objects from videos historically stems from the problem of image salient object detection [33, 34], where existing methods often rely either on appearance features or on salient motion from 2D optical flow [24, 28, 42, 43, 59, 62]. Oftentimes, optical flow is interpreted as a color image [24, 62], where geometric information, such as camera egomotion, is ignored. Close to our methodology, Motion Angle Network (MoA-Net) [6], analytically reduces the effect of camera rotation and uses the "rectified" flow angle as input features to a binary segmentation network. Our approach further incorporates 3D flow and depth cues and segments multiple rigid motions.

**Instance Scene Flow:** Scene flow is the problem of resolving dense 3D scene motion from an ego-camera [32, 48], which is inherently challenging due to the lack of visual evidence for correspondence matching in the local patch, for example, occluded or specular surfaces. Prior approaches often utilize scene rigidity priors to resolve such ambiguities, such as piecewise rigidity prior [32, 50] and semantic rigidity prior [3, 30]. Then, scene flow is parameterized as the 3D motion of each rigid segment. However, it is risky to segment the scene purely relying on semantics – object that is able to move is not the same as objects that are moving. Furthermore, such high-level cues do not generalize to an open-world, where algorithms are required to be robust to never-before-seen categories [4]. Instead, we exploit *motion rigidity* for scene flow estimation, which decomposes the scene into multiple rigidly moving segments while preserving the completeness of individual rigid bodies.

## 3. Approach

In this section, we first analyze degeneracies in motion segmentation that arise when dynamic motion is indistinguishable from the camera motion, and what information is required to resolve the ambiguities. We then design a neural architecture for rigid instance motion segmentation that builds on this geometric analysis, producing a pipeline for two-frame rigid motion segmentation.

### 3.1. Two-Frame Geometric Motion Segmentation

**Problem setup:** Given two-frame motion correspondences $(\mathbf{p_0}, \mathbf{p_1}) \in R^2$, observed by a calibrated monocular camera (with known camera intrinsics $\mathbf{K}$), we are interested in detecting points whose motion cannot be described by the camera motion $\mathbf{R_c} \in SO(3)$, $\mathbf{T_c} \in R^3$, such that

$$(\mathbf{R_c P_1} + \mathbf{T_c}) - \mathbf{P_0} \neq \mathbf{0}, \quad \text{(transform of coordinates)} \quad (1)$$

where $\mathbf{P_0} = Z_0 {\mathbf{K_0}}^{-1} \tilde{\mathbf{p}}_\mathbf{0}$ and $\mathbf{P_1} = Z_1 {\mathbf{K_1}}^{-1} \tilde{\mathbf{p}}_\mathbf{1}$ are corresponding 3D points observed in the first and second frame's camera coordinate system, with corresponding depth $Z$ and homogeneous 2D coordinates $\tilde{\mathbf{p}}$. To gain more geometric insights, we re-arrange Eq. (1) into

$$\mathbf{T_{sf}} = {\mathbf{K_0}}^{-1}(Z_1 \mathbf{H_R} \tilde{\mathbf{p}}_\mathbf{1} - Z_0 \tilde{\mathbf{p}}_\mathbf{0}) \neq -\mathbf{T_c}, \quad \text{("rectified" 3D scene flow} \neq \text{negative camera translation)} \quad (2)$$

where $\mathbf{H_R} = \mathbf{K_0 R_c} {\mathbf{K_1}}^{-1}$ is the rotational homography that "rectifies" the second image plane into the same orientation as the first image plane, removing the effect of camera rotation from the 2D motion fields, and $\mathbf{T_{sf}}$ is the rectified 3D scene flow measured with respect to first frame's 3D coordinate system. Eq. (2) states that the rectified 3D scene flow of a moving point $\mathbf{P}$ will not equal the negative camera translation. However, there are three crucial degrees of freedom that are undetermined: depth $Z_0$, $Z_1$ and the magnitude of the ego-camera translation $\mathbf{T_c}$, depending on available sensors.

**Coplanar motion degeneracy:** Solving for $Z_0$ and $Z_1$ equates to estimating 3D scene flow, which itself is challenging [32]. To remove the dependence on depth, classic geometric motion segmentation segments points whose 2D motion is inconsistent with the camera motion, measured either by Sampson distance to the epipolar line [19, 44] or plane plus parallax (P+P) [39] representations that factor out camera rotation, allowing one to evaluate the angular deviation of the 2D motion to the epipole [5, 23]. However, *is 2D motion sufficient to segment points moving in 3D?* The answer is no (Fig. 3). Formally, 3D points that translate within

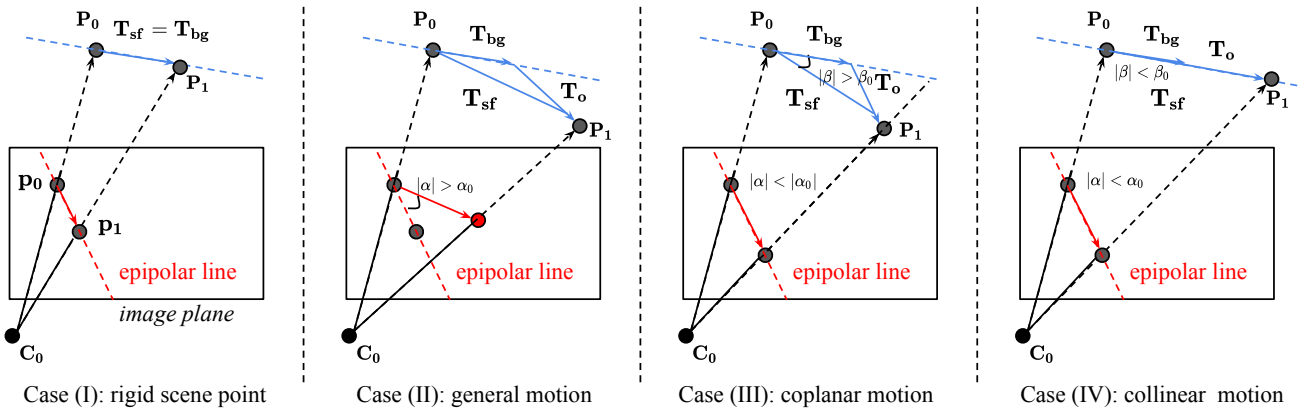



Figure 4: When can a scene point $\mathbf{P}$ be identified as moving? Assuming camera rotation has been removed, we have 3D scene flow (defined as motion relative to the camera) $\mathbf{T_{sf}} = \mathbf{T_{bg}} + \mathbf{T_o}$, where $\mathbf{T_{bg}} = -\mathbf{T_c}$ is the rigid background motion induced by the camera motion, and $\mathbf{T_o}$ is the independent object motion. Case (I): Assuming a rigid scene point $\mathbf{P}$ with zero independent motion $\mathbf{T_o} = \mathbf{0}$, the 2D motion projected by $\mathbf{T_{sf}}$ must lie on the epipolar line. Case (II): In other words, if the 2D motion deviates from the epipolar line, $|\alpha| > \alpha_0$, $\mathbf{P}$ must be moving, analagous to Sampson error [19]. Case (III): *However, the inverse does not hold.* If 2D flow is consistent with the background motion ($|\alpha| < \alpha_0$), $\mathbf{P}$ might still be moving in the epipolar plane. However, if the *angular direction* of 3D flow $\mathbf{T_{sf}}$ – computable from optical expansion [58] – differs from $\mathbf{T_{bg}}$ ($|\beta| > \beta_0$), $\mathbf{P}$ must be moving. Case (IV): If the 3D flow direction is consistent with background motion ($|\beta| < \beta_0$), $\mathbf{P}$ could still be moving in the direction of $\mathbf{T_{bg}}$, making it unrecoverable without knowing the scale (or relative depth).

the epipolar plane defined by the camera translation vector $\mathbf{T}_c$ will project to the epipolar line, making them "appear" as stationary points, as shown in Fig. 4 Case (II).

To address co-planar motion ambiguity, we make use of optical expansion cues that upgrade 2D flow to 3D as suggested by recent work [58]. Optical expansion, measured by the scale change of overlapping image patches, approximates the relative depth $\tau = \frac{Z_1}{Z_0}$ for non-rotating scene elements under scaled orthographic projection [58]. We derive a 3D motion angle criterion that does not require depth, but removes the ambiguity of points moving within the epipolar plane. Normalizing Eq. (2) by depth $Z_0$, we have

$$\tilde{\mathbf{T}}_{\mathbf{sf}} = \mathbf{K_0}^{-1}(\tau \mathbf{H_R} \tilde{\mathbf{p}}_\mathbf{1} - \tilde{\mathbf{p}}_\mathbf{0}) \not\sim -\mathbf{T_c}, \quad \text{(rectified 3D flow direction} \neq \text{neg. camera translation direction)} \tag{3}$$

where $\tilde{\mathbf{T}}_{\mathbf{sf}} = \frac{\mathbf{T_{sf}}}{Z_0}$ is the rectified and normalized 3D flow and $\not\sim$ indicates two vectors are different in direction. Eq. (3) states that a point is moving if the direction of its rectified 3D scene flow is not consistent with the direction of the camera translation, as shown in Fig. 4 Case (III).

**Collinear motion degeneracy:** However, there is still a remaining ambiguity that cannot be resolved. If point $\mathbf{P}$ moves in the opposite direction of the camera translation, both classic criteria and Eq. (3) would fail, as shown in Fig. 4 Case (IV). Such ambiguity remains even given multiple frames [60], but is common in many real-world applications, e.g., two cars passing each other (Fig. 3). To identify moving points in such cases, one could use depth $Z_0$ to recover the metric scale of normalized scene flow $\tilde{\mathbf{T}}_{\mathbf{sf}}$, and compare it with camera translation $\mathbf{T_c}$. However, in a monocular setup, we neither know the scale of $\mathbf{T_c}$ nor trust the overall scale of $Z_0$ [19]. Instead, we derive a depth contrast criterion, inspired by an observation that *dynamic* scene points triangulated from flow assuming overall rigidity will appear "abnormal" in the 3D reconstruction, such as the floating car in Fig. 3 (b). To do so, we contrast the flow-derived depth $Z_0^{flow}$ with a depth prior $Z_0^{prior}$,

$$Z_0^{flow} \neq \gamma Z_0^{prior}, \quad \text{(flow-triangulated depth} \neq \text{depth prior)} \tag{4}$$

where $Z_0^{flow}$ can be computed efficiently using midpoint or DLT triangulation algorithm [19], depth prior $Z_0^{prior}$ can be represented by a data-driven monocular depth network, and the scale factor $\gamma$ that globally aligns $Z_0^{prior}$ to $Z_0^{flow}$ can be determined by robust least squares [41].

**Egomotion/planar degeneracy:** However, when the camera translation is small, $\mathbf{T_c}$ is notoriously difficult to estimate due to small motion parallax. In such cases, rigid-background motion (and objects that deviate from it) is easier to model with a rotational homography model [44, 46].

### 3.2. Learning to Segment Rigid Motions

We now operationalize our motion analysis into a deep network for rigid motion segmentation (Fig. 5). At its heart, *our network learns to transform motion measurements (noisy 3D scene flow) into pixel-level masks of rigid background and instances.* To do so, we construct motion cost maps designed to address the motion degeneracies described earlier. Given such input maps and raw scene flow measurements, we use a two-stream network architecture that separately regresses the rigid background and rigid instance masks.

**Motion estimation:** First, we extract the camera and relative scene motion given two frames. We apply existing meth-

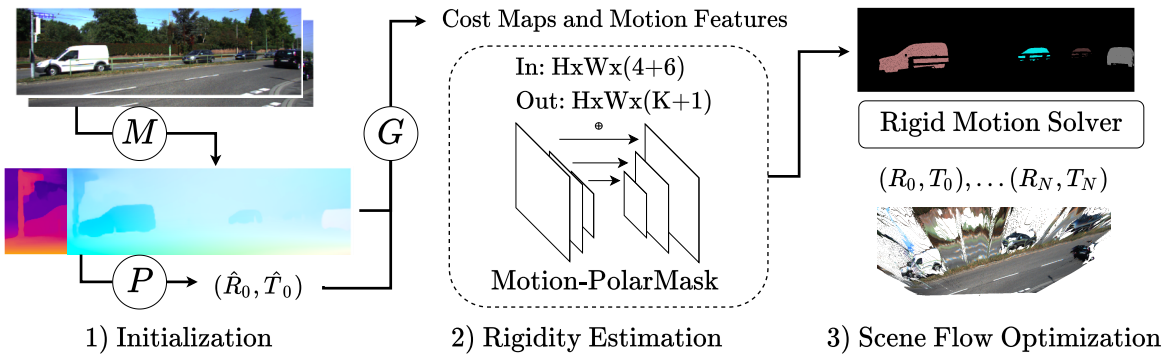



Figure 5: We detect and estimate rigid motions in three steps: First, initial scene flow is computed using off-the-shelf networks (M) and camera motion is estimated by epipolar geometry (P) given two consecutive frames. Then, rigidity cost maps and rectified 3D scene flow are computed and fed into a two-stream network that produces the segmentation masks of a rigid background and an arbitrary number of rigidly moving instances. Finally, we fit rigid transformations within rigid masks using a nonlinear solver and refine the 3D scene flow.

ods to estimate optical flow, optical expansion and monocular depth [36, 58]. We then fit and decompose essential matrices from down-sampled flow maps using the 5-pt algorithm with a differentiable (although not used to back-propagate gradients in experiments) and parallel RANSAC [9].

**Rigidity cost-maps inputs:** We construct rigidity cost maps tailored to camera-object motion configurations analysed in Sec. 3.1, including (1) an epipolar cost (for general motion), (2) a homography cost (for small camera translations), (3) a 3D P+P cost (for coplanar motion), and (4) a depth contrast cost (for colinear motion). Given camera motion and 2D flow, epipolar cost is computed as per-pixel Sampson error [19]. Homography cost is implemented as per-pixel reprojection error according to the rotational homography [14] in Eq. (2). 3D P+P cost and depth contrast cost are computed as Eq. (3) and Eq. (4). For the latter, we use MiDaS [36] as the depth input. The computation of rigidity costs are implemented as a differentiable layer.

**Rectified 3D scene flow inputs:** In addition to the rigidity cost-maps inputs above, we also find it helpful to input rectified 3D scene flow measurements. As discussed in Sec. 3.1, the first frame 3D scene points can be computed by back-projection given camera intrinsics and monocular depth. To compute the rectified 3D scene flow, we upgrade 2D optical flow using optical expansion, as in Eq. (2), where the second coordinate frame is rectified with a rotational homography to reduce the ambiguities caused by camera rotation. Finally, the first frame 3D scene points and rectified scene flow measurements are concatenated as a six-channel feature map. Such rectified 3D motion input is more effective than 2D optical flow, as empirically tested in ablation study (Tab. 4).

**Architecture and losses:** We use a two-stream architecture: (1) a lightweight U-Net architecture to predict binary labels for pixels belonging to the (rigid) background and (2) a CenterNet architecture to predict pixel instance masks. Inspired by the single-shot segmentation head proposed in PolarMask [54], stream (2) outputs a heatmap representing object centers and a regression map of $K = 36$ polar distances at evenly distributed angles. The overall architecture is end-to-end differentiable and can be trained with standard loss functions,

$$L = \alpha_1 L_{\text{binary}} + \alpha_2 L_{\text{center}} + \alpha_3 L_{\text{polar}} \tag{5}$$

where $L_{\text{binary}}$ is binary cross-entropy loss with label balancing, $L_{\text{center}}$ is the focal loss and $L_{\text{polar}}$ is an $L_1$ regression loss evaluated at each ground-truth center location. Weights are empirically chosen as $\alpha_1 = 1^{-3}$, $\alpha_2 = 1^{-3}$ and $\alpha_3 = 1^{-8}$. Intuitively, stream (2) generates coarse instance-level masks that are refined by pixel-accurate background masks from stream (1). Specifically, pixels where rigid background and instance predictions disagree are not used for rigid body fitting below, and marked as wrong prediction in evaluation.

**Rigid motion fitting:** Given segmentations, the rigid background and instances are parameterized with the best fit of rotations and translations using a parallel RANSAC. For stereo scene flow estimation, the rigid transformations are refined by solving PnP with non-linear optimization [19].

## 4. Experiments

Our method is quantitatively compared with state-of-the-art rigidity estimation algorithms on KITTI and Sintel in Sec. 4.1, and then applied to the depth and scene flow estimation tasks in Sec. 4.2. In Sec. 4.3 we conclude with an ablation study.

**Dataset:** We use KITTI-SF (sceneflow) and Sintel for quantitative analysis. KITTI-SF [17, 32] features an urban driving scene with multiple rigidly moving vehicles. Sintel [11] is a synthetic movie dataset that features a highly dynamic environment. It contains viewpoints and objects (such as dragons) that are rare in existing datasets. KITTI provides ground-truth segmentation masks for the rigid background

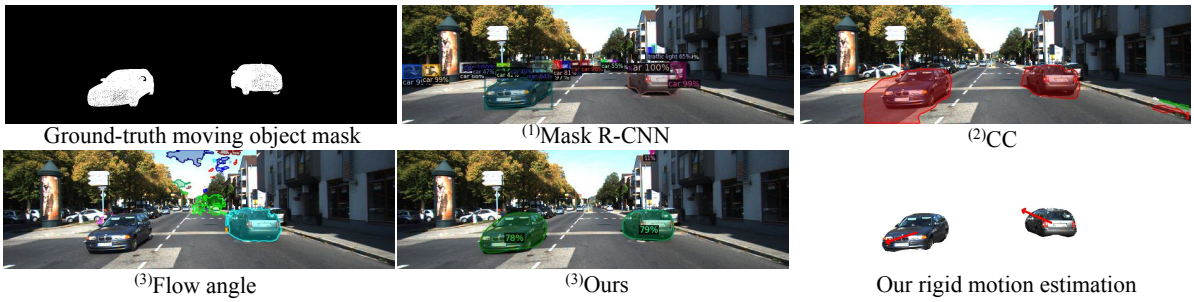



Figure 6: Comparison on KITTI-SF, image 137. The prefix of each method indicates the test-time inputs: (1)Single frame. (2)Multi-frame with appearance features. (3)Multi-frame without appearance. Our best appearance-based segmentation baseline, (1)Mask R-CNN [20] detects all the moving vehicles, but also reports parked cars in the background. (2)CC [37] correctly detects moving cars but also reports the edge of the road as moving objects. (3)Geometric segmentation algorithm [5, 53] fails on the approaching car due to the colinear motion ambiguity, and reports false positives at the background due to the noisy flow estimation. In contrast, (3)our method correctly segments both the moving vehicles and the rigid background. Rigid motions are estimated within each mask and applied to depth and scene flow estimation.

Table 1: Rigidity estimation on KITTI (K) and Sintel (S). (1)Single frame. (2)Multi-frame with appearance features. (3)Multi-frame without appearance. The best result under each metric (IoU in %) is bolded. *:For methods only estimating background masks, we use connected components to obtain object masks. ‡:Methods trained with target dataset mask annotations. For example, MR-Flow-S (K) is trained on KITTI, and MR-Flow-S (S) is trained on Sintel.

| | Method | K: obj ↑ | K: bg ↑ | S: bg ↑ |
|---|---|---|---|---|
| (1) | Mask R-CNN [52] | 88.20 | 96.42 | 81.98 |
| | $U^2$ (Saliency) [35] | 64.80* | 93.34 | 82.01 |
| | MR-Flow-S (K) [53] | 75.59* | 94.70‡ | 76.11 |
| | MR-Flow-S (S) [53] | 11.11* | 84.72 | 92.64‡ |
| (2) | FSEG [24] | 85.08* | 96.27 | 80.22 |
| | MAT-Net [62] | 68.40* | 93.08 | 77.95 |
| | COSNet [28] | 66.67* | 93.03 | 80.86 |
| | CC [37] | 50.87* | 85.50 | ✗ |
| | RTN [29] | 34.29* | 84.44 | 64.86 |
| (3) | FSEG-Motion [24] | 61.29 | 89.41 | 78.25 |
| | CC-Motion [37] | 42.99 | 74.06 | ✗ |
| | Flow angle [5, 53] | 25.83 | 85.52 | 74.23 |
| | Ours | **89.62** | **97.00** | **85.61** |

and moving car instances, where the remaining dynamic objects (such as pedestrians) are manually removed. For Sintel, computing rigid instances masks is an ill-posed problem since most objects are nonrigid. Instead, we obtain ground-truth rigid background segmentation from MR-Flow [53]. Both datasets also provide ground-truth depth and scene flow as well as calibrated cameras.

**Setup I (monocular):** In the monocular case, depth and scene flow are predicted given two consecutive frames. We use MiDaS [36], a state-of-the-art monocular depth estimator to acquire weak, relative scale information from the first frame. The remaining network parameters are trained using synthetic data: optical flow and optical expansion networks are pre-trained on FlyingChairs, SceneFlow, and VIPER [13, 31, 38] following the two-stage protocol [22]; the two-stream rigid motion segmentation network is trained on FlyingThings and Driving (both included in SceneFlow) with flow network fixed. Then, we evaluate segmentation and monocular scene flow performance on KITTI-SF and Sintel.

**Setup II (stereo):** Our method is able to take advantage of reliable depth sensors, such as stereo cameras to produce better segmentation and scene flow estimation. As in the stereo setup, scene flow is estimated from two consecutive stereo pairs. We use GA-Net [61] to acquire metric scale information from the first stereo pair. We mix KITTI-SF with synthetic Scene Flow dataset for training, and evaluate stereo scene flow performance on the KITTI-SF benchmark.

### 4.1. Two-frame Rigid Motion Segmentation

**Metrics:** Following prior works, we compute background IoU [29, 37] for rigid background segmentation and object F-measure [12] for rigid instance segmentation. Only the rigid background segmentation metric is reported on Sintel due to the lack of rigid bodies ground-truth rigid motion segmentation masks.

**Baselines:** We group baselines according to test inputs.

(1)**Single frame methods**. Mask R-CNN with ResNeXt-101+FPN backone is the most accurate model on MSCOCO provided by Detectron2 [20, 27, 52, 55]; $U^2$Net [35] is a state-of-the-art salient object detector; and MR-Flow-S [53] is a semantic rigidity estimation network fine-tuned separately on KITTI and Sintel.

(2)**Multi-frame with appearance features**. FusionSeg [24] is a two-stream architecture that fuses the appearance and optical flow features, and we provide SOTA optical flow on KITTI and Sintel as motion input. COSNet [28] and MAT-Net [62] are SOTA video objection segmentation methods on DAVIS [34]. CC [37] combines "flow-egomotion consensus score" (similar to our epipolar costs) with a foreground probability regressed from five consecutive frames, which is then thresholded to obtain the background mask. RTN [29]

Table 2: Monocular depth and scene flow results on KITTI (K) and Sintel (S). D1: first frame disparity (inverse depth) error. SF: scene flow error (%). The best result under each metric is underlined. The best result without training on the target domain data is bolded.

| Method | K: D1 ↓ | K:SF ↓ | S: D1 ↓ | S:SF ↓ |
|---|---|---|---|---|
| CC [37] | 36.20 | 51.80 | ✗ | ✗ |
| SSM [21] | 31.25 | 47.05 | ✗ | ✗ |
| Mono-SF [10] | <u>16.72</u> | <u>21.60</u> | ✗ | ✗ |
| MiDaS+OE [58] | 43.50 | 51.47 | 60.43 | 63.86 |
| MiDaS+Mask | 19.24 | 24.51 | 56.66 | 60.01 |
| MiDaS+Ours | **18.47** | **23.92** | **55.36** | **59.01** |

Table 3: Stereo scene flow results on KITTI benchmark. D1 and D2: first and second frame disparity error. Fl: optical flow error. SF: scene flow error. Metrics are errors in percentage and top results are bolded. *First frame disparity is not refined by our method.

| Method | D1* ↓ | D2 ↓ | Fl ↓ | SF ↓ |
|---|---|---|---|---|
| PRSM [50] | 4.27 | 6.79 | 6.68 | 8.97 |
| OE [58] | 1.81 | 4.25 | 6.30 | 8.12 |
| DRISF [30] | 2.55 | 4.04 | 4.73 | 6.31 |
| Ours | 1.89 | **3.23** | **3.50** | **4.89** |
| Ours Mask R-CNN | 1.89 | 3.42 | 4.26 | 5.61 |

uses a CNN to predict rigid background masks given two RGBD images. For Sintel, we use the ground-truth depth as input; for KITTI, since the ground-truth depth is sparse, we use MonoDepth2 [18] instead.

(3)**Two-frame without appearance**. We separately evaluate the motion stream of FSEG and the flow-egomotion consensus results of CC. Following prior work [5, 53], we implement a classic motion segmentation pipeline that combines the motion angle and motion residual criteria.

Besides CC, RTN, and the classic pipeline, all baselines are trained or pre-trained on large-scale manually annotated segmentation datasets that contain common objects and scenes, while ours is not.

**Performance analysis:** We show qualitative comparison in Fig. 6 and report results in Tab. 1. On KITTI, our method outperforms the most accurate baseline, Mask R-CNN, in terms of both rigid instance segmentation and background segmentation. Although Mask R-CNN is trained on common scenes (including driving), it cannot tell whether an object is moving or static, similar to other single-frame methods. Therefore, our method compares favorably to Mask R-CNN on rigid motion segmentation task. On Sintel, our method outperforms all the baselines except MR-Flow-S (S), which uses the first half of all Sintel sequences for training. If we compare to MR-Flow-S (K), which is not fine-tuned on Sintel, our method is better. Finally, among the motion-based segmentation methods, our method is the best on both datasets, because of our robustness to degenerate motion configurations as well as noisy flow inputs.

### 4.2. Rigid Depth and Scene Flow Estimation

The rigid body masks are then applied to the two-frame depth and scene flow estimation task, under both monocular and stereo setups. In both cases, we fit a 3D rigid transformation for pixels within each rigid body mask, which is used to refine the depth and scene flow estimation.

**Metrics:** To evaluate depth and scene flow estimation performance on KITTI and Sintel, we report disparity error on both frames (D1, D2), optical flow error (Fl) and scene flow error (SF) following KITTI [32]. In the monocular setup, to remove the overall scale ambiguity, we take an extra step to align the overall scale of the predictions to the ground-truth with their medians [36, 51].

**Setup I (monocular):** For each rigid body (including the background), we fit a rigid body motion model (rotation and up-to-scale translation), which can be used to triangulate motion correspondences and to produce up-to-scale depth. To deal with the scale ambiguity between rigid background and rigid bodies, we use the monocular depth map to estimate scale factors for each triangulated depth map, by aligning both with robust least squares [41]. Once we have the scale factor for each rigid body, we simply report the triangulated depth and the scene flow according to rigid motion models across two-frames.

**Baselines:** We compare against state-of-the-art monocular scene flow baselines. **CC** [37] and **SSM** [21] are representative methods for self-supervised monocular depth and scene flow estimation that does not make use of segmentation priors at inference time. **Mono-SF** [10] trains a monocular depth network with KITTI ground-truth, and solve an optimization problem given semantic instance segmentation provided by Mask R-CNN. The above three methods are trained on KITTI and the results are taken from their papers. **OE** (optical expansion) [58] learns to predict relative depth from dense optical expansion, which together with optical flow, directly yields 3D motion. It is trained on the synthetic SceneFlow dataset, and we use MiDaS to provide the scale. We also implement a baseline that predicts instance segmentation masks by Mask R-CNN, and follows the same rigid body fitting procedure as ours(referred to as **MiDaS+Mask**).

**Performance analysis:** We report results on KITTI-SF and Sintel in Tab. 2. First, it is noted our method reduces the disparity error of MiDaS by more than half on KITTI, and 4.8% on Sintel. Compared to OE, which uses the same monocular depth input as ours, we are better in all metrics. (SF: 23.92% vs 51.47%), which demonstrates the effectiveness of our rigid motion mask. Our method also outperforms the other methods that do not use segmentation priors (CC and SSM). Compared to Mono-SF, which is trained with ground-truth KITTI depth maps, and uses a semantic segmentation prior, our method is slightly worse on KITTI.

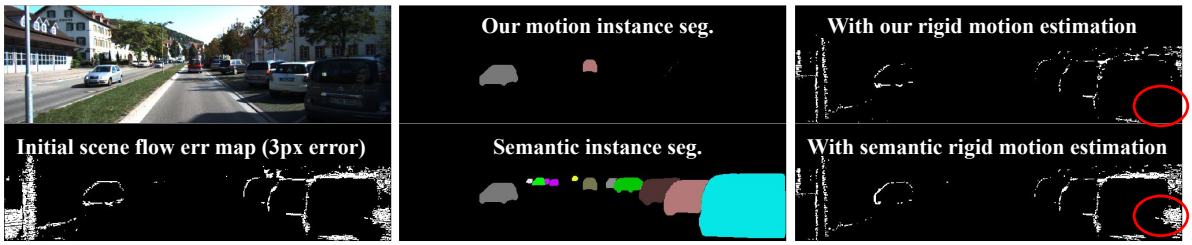



Figure 7: Rigidity vs semantic-based segmentation for instance scene flow. Given instance segmentation masks, scene flow can be optimized by fitting rigid body transforms within each mask. While semantic segmentation fails to improve scene flow estimation on the parked cars (in red circle), our rigid motion mask groups the parked car together with the rigid background and successfully reduces the scene flow error.

Compared to Midas-Mask, our method is strictly better on both KITTI and Sintel, indicating the benefit of usijg our rigid motion masks versus appearance-based masks.

**Setup II (stereo):** We fit rigid transformations over flow correspondences within each rigid mask, and refine them by PnP given first frame depth and flow provided by GA-Net and VCN [57, 61]. The optical flow and scene flow are updated according to the estimated rigid transformations.

**Performance analysis:** We report benchmark performance on KITTI in Tab. 3. Among the baselines, **OE** [58] uses the same network architecture trained on KITTI to upgrade optical flow to 3D scene flow. Same as ours, it uses the GA-Net stereo and the VCN optical flow as inputs. **PRSM** segments an image into superpixels, and fits rigid motions to estimate piece-wise rigid scene flow. Given semantic instance segmentation [20], two-frame depth, and optical flow, **DRISF** [30] casts scene flow estimation as an energy minimization problem and finds the best rigid transformation for each *semantic* instance. Our key difference from PRSM and DRISF is that we use rigid motion segmentation masks instead of superpixel or semantic instance segmentation. Although most objects in KITTI are common cars where appearance-based detectors (such as Mask R-CNN) works very well, our method still achieves state-of-the-art performance on KITTI scene flow benchmark (SF: 4.89 vs 6.31). If we replace the segmentation masks with semantic segmentation, the performance drops noticeably (SF: 4.89% to 5.61%). As illustrated in Fig. 7, our method successfully groups the static objects (e.g. parked cars) with the rigid background, which effectively improves scene flow accuracy by optimizing the whole background as one rigid body.

### 4.3. Diagnostics

We ablate critical components of our approach and retrain networks. Results are shown in Tab. 4. We validate the design choices of using (1)a rigid instance segmentation architecture, (2)explicitly computed rigidity cost-maps inputs, (3)monocular depth input, and (4)optical expansion that upgrades 2D optical flow to 3D. (1)If we replace CenterNet architecture with connected component on foreground pixels, the accuracy in rigid object segmentation drops from 89.53% to 85.38%, which shows the benefit of the architecture. (2)If Removing rigidity cost-maps and directly regressing rigidity masks from 3D motion features leads to a slight drop of accuracy on KITTI, and a large accuracy drop on Sintel (84.63% to 76.81%). This indicates the cost map features are crucial for Sintel, possibly due to complex camera and object motion configurations in Sintel, where explicit domain knowledge is helpful. (3)Further removing monocular depth input leads to an accuracy drop on all metrics, especially for KITTI, which is due to the dominance of collinear motion in autonomous driving scenes. (4)After further removing optical expansion, our method degrades to MoA-Net [6].The performance drops noticeably on rigid instance segmentation, but improves on the background segmentation. This indicates optical expansion is more useful for segmenting foreground objects. (5)Lastly, we remove the binary and rigid instance segmentation networks and our method becomes the classic geometric segmentation algorithm, which is not comparable with our reference method in all metrics.

Table 4: Diagnostics of rigid body motion segmentation on KITTI-SF. Dignostics in the second group are sequential.

| Method | K: obj ↑ | K: bg ↑ | S: bg ↑ |
|---|---|---|---|
| Reference | 89.53 | 97.22 | 84.63 |
| (1)No instance network | 85.38 | 97.22 | 84.63 |
| (2)w/o cost maps | 88.66 | 96.59 | 76.81 |
| (3)w/o monocular depth | 84.46 | 94.84 | 76.14 |
| (4)w/o expansion (MoA [6]) | 81.28 | 95.50 | 77.00 |
| (5)w/o CNN [5, 53] | 25.83 | 85.52 | 74.23 |

## 5. Conclusion

We investigate the problem of two-frame rigid body motion segmentation in an open dynamic environment. We theoretically analyze the degenerate cases in geometric motion segmentation and introduce novel criteria and inputs to resolve such ambiguities. We further propose a modular neural architecture that is robust to noisy observations as well as different motion types, which demonstrates state-of-the-art performance on rigid motion segmentation, depth and scene flow estimation tasks.

## References


[1] Gerald J Agin and Thomas O Binford. Computer description of curved objects. *IEEE Transactions on Computers*, (4):439–449, 1976. 2

[2] Ioan Andrei Bârsan, Peidong Liu, Marc Pollefeys, and Andreas Geiger. Robust dense mapping for large-scale dynamic environments. In *ICRA*, 2018. 1

[3] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *ICCV*, 2017. 3

[4] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015. 2, 3

[5] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, 2016. 1, 2, 3, 6, 7, 8

[6] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moanet: self-supervised motion segmentation. In *ECCVW*, 2018. 3, 8

[7] Pia Bideau, Aruni RoyChowdhury, Rakesh R Menon, and Erik Learned-Miller. The best of both worlds: Combining cnns and geometric constraints for hierarchichal motion segmentation. In *CVPR*, 2018. 2

[8] Irving Biederman. Geon theory as an account of shape recognition in mind and brain. *The Irish Journal of Psychology*, 14(3):314–327, 1993. 2

[9] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 5

[10] Fabian Brickwedde, Steffen Abraham, and Rudolf Mester. Mono-SF: Multi-view geometry meets single-view depth for monocular scene flow estimation of dynamic traffic scenes. In *CVPR*, 2019. 7

[11] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 5

[12] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards segmenting everything that moves. In *ICCVW*, 2019. 6

[13] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 6

[14] Elan Dubrofsky. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, 2009. 5

[15] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 1

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5

[18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 7

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3, 4, 5

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 6, 8

[21] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *CVPR*, 2020. 7

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*. 6

[23] Michal Irani and P Anandan. A unified approach to moving object detection in 2d and 3d scenes. *PAMI*, 1998. 2, 3

[24] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017. 3, 6

[25] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. 2019. 1

[26] Shih-Chieh Lin, Yunqi Zhang, Chang-Hong Hsu, Matt Skach, Md E Haque, Lingjia Tang, and Jason Mars. The architectural implications of autonomous driving: Constraints and acceleration. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 751–766, 2018. 1

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014. 6

[28] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019. 3, 6

[29] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In *ECCV*, 2018. 2, 6

[30] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *CVPR*, 2019. 3, 7, 8

[31] N. Mayer, E. Ilg, P. Häusser, 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. 6

[32] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 3, 5, 7

[33] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 2013. 3

[34] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 3, 6

[35] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 2020. 6

[36] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv:1907.01341*, 2019. 2, 5, 6, 7

[37] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019. 6, 7

[38] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, pages 2213–2222, 2017. 6

[39] Harpreet S Sawhney. 3d geometry from planar parallax. In *CVPR*, 1994. 2, 3

[40] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998. 2

[41] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 4, 7

[42] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 3

[43] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *IJCV*, 2019. 3

[44] Philip HS Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998. 2, 3, 4

[45] Philip HS Torr, Andrew W Fitzgibbon, and Andrew Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *IJCV*, 1999. 2, 3

[46] Philip HS Torr, Andrew Zisserman, and Stephen J Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *CVIU*, 1998. 4

[47] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007. 2

[48] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *ICCV*, 1999. 3

[49] René Vidal and Shankar Sastry. Optimal segmentation of dynamic scenes from two perspective views. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II. IEEE, 2003. 2

[50] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *IJCV*, 2015. 3, 7

[51] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, pages 2022–2030, 2018. 7

[52] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[53] Jonas Wulff, Laura Sevilla-Lara, and Michael J. Black. Optical flow in mostly rigid scenes. In *CVPR*, 2017. 1, 2, 6, 7, 8

[54] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12193–12202, 2020. 5

[55] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 6

[56] Xun Xu, Loong Fah Cheong, and Zhuwen Li. 3d rigid motion segmentation with mixed and unknown number of models. *PAMI*, 2019. 2

[57] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 8

[58] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *CVPR*, 2020. 2, 4, 5, 7, 8

[59] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *CVPR*, 2019. 3

[60] Chang Yuan, Gerard Medioni, Jinman Kang, and Isaac Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *PAMI*, 2007. 2, 3, 4

[61] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided aggregation net for end-to-end stereo matching. In *CVPR*, 2019. 6, 8

[62] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *AAAI*, 2020. 3, 6