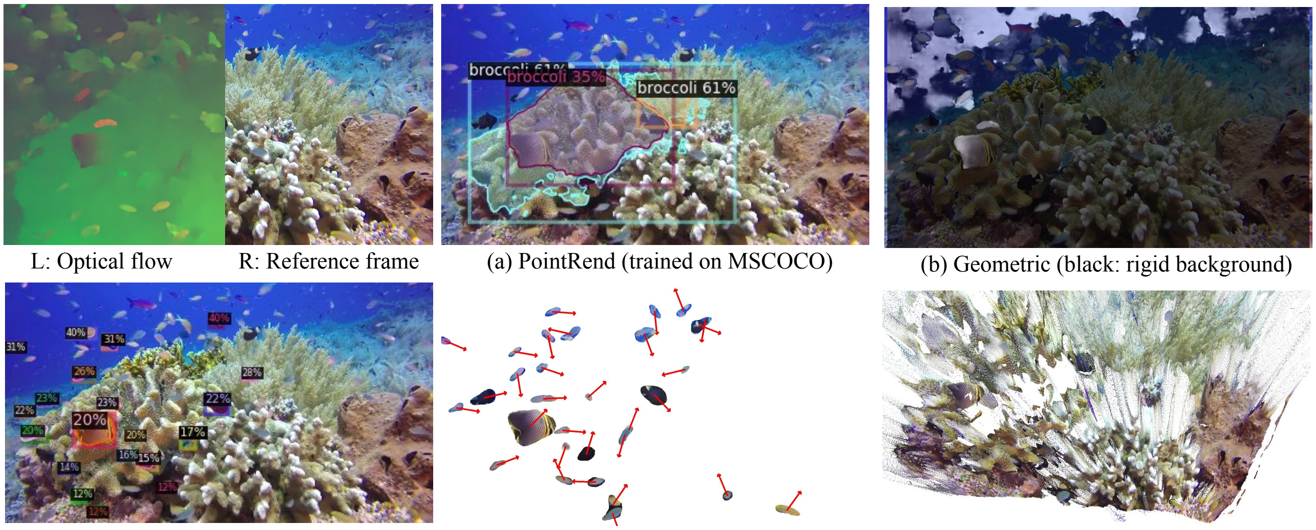
Learning to Segment Rigid Motions from Two Frames

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 two-frame reconstruction

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, 58] 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 decomposes 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 inferred rigid motions significantly improve depth and scene flow accuracy.

Abstract

Appearance-based detectors achieve remarkable performance on common scenes, benefiting from high-capacity models and massive annotated data, but tend to fail for scenarios that lack 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 multiple rigidly moving objects, which are then parameterized by 3D rigid transformations. Our method achieves state-of-the-art performance for rigid motion segmentation on KITTI and Sintel. The inferred rigid motions 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 [27]. While one can build accurate detectors for many categories of objects that are able to move, "being able to move" is not equivalent to

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

"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, 43, 50, 56, 61] 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].

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 camera motion is close to zero [61]. Second, points moving along epipolar lines cannot be distinguished from the rigid background [65], 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 [63] and monocular depth cues [38]. 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.

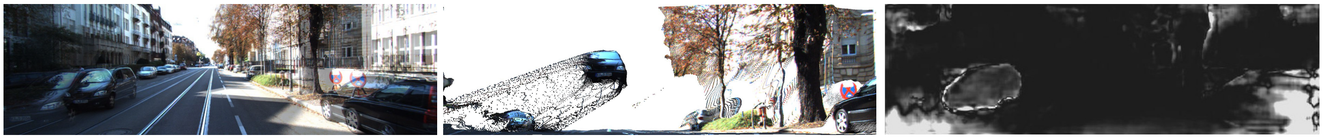
Contributions: (1) We provide a geometric analysis for ambiguities in 3D rigid motion segmentation from 2D motion fields, and introduce solutions to deal with such ambiguities. (2) We propose a geometry-aware architecture for 3D rigid motion segmentation from two RGB frames, which is generalizable to novel appearance, resilient to different motion types and robust to noisy motion observations. (3) Our method achieves state-of-the-art (SOTA) performance of rigid motion segmentation on KITTI/Sintel. The inferred rigidity masks significantly improve the performance of downstream depth and scene flow estimation tasks.

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 [47, 48, 50, 52, 53, 61, 65]. However, prior methods either focus on theoretical analysis with noisy-free data, or assume relatively simple scenes where long-term motion trajectories can be obtained by point tracking algorithms. Some recent work [5, 7, 58] tackles more complex scenarios with two-frame optical flow inputs, where geometric constraints, such as motion angle and plane plus parallax (P+P) [42] are considered as cues of "moving versus static". However, such geometric constraints are sensitive to noise in optical flow even under a robust estimation framework [16]. Moreover, as we shall see in Sec. 3.1, the prior two-frame solutions do not deal with several degenerate cases, including co-planar/co-linear motion [65] and camera motion degeneracy [48]. We address these problems by encoding geometric constraints into a modular neural network.

Learning-Based Video Object Segmentation: Segmenting salient objects from videos historically stems from the problem of image salient object detection [35, 36], where existing methods often rely either on appearance features or on salient motions from 2D optical flow [24, 30, 45, 46, 64, 67]. Oftentimes, optical flow is interpreted as a color image [24, 67], 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 3D rigid motions.

Instance Scene Flow: Scene flow is the problem of resolving dense 3D scene motion from an ego-camera [34, 51], which is challenging due to the lack of visual evidence to find correspondence matches, for example, when occlusion occurs. Prior approaches often utilize scene rigidity priors to resolve such ambiguities, such as piecewise rigidity prior [34, 54] and semantic rigidity prior [3, 32]. However, it is risky to segment the scene purely relying on semantics – an object that is able to move is not the same as an object that is 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.



(a) Two frame overlay (b) Naive flow-triangulated reconstruction (c) Sampson err. (white indicates moving)

Figure 2: 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 [65] 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].

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 3D Motion Segmentation

Problem setup: Given two-frame 2D motion correspondences written in homogenous coordinates $(\tilde{\mathbf{p}}_\mathbf{0}, \tilde{\mathbf{p}}_\mathbf{1})$ with depths (Z_0, Z_1) observed by camera intrinsics $(\mathbf{K_0}, \mathbf{K_1})$, the corresponding 3D points in the camera coordinate system of each frame is given by $\mathbf{P_0} = Z_0 \mathbf{K_0}^{-1} \tilde{\mathbf{p}}_\mathbf{0}$ and $\mathbf{P_1} = Z_1 \mathbf{K_1}^{-1} \tilde{\mathbf{p}}_\mathbf{1}$. We wish to detect points whose 3D motion cannot be described by camera motion $\mathbf{R_c} \in \mathbf{SO}(3)$, $\mathbf{T_c} \in \mathbb{R}^3$:

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

To gain more geometric insights, we re-arrange Eq. (1) into

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

where $\mathbf{T_{sf}} = \mathbf{R_c P_1} - \mathbf{P_0}$ is the “rectified” 3D scene flow, with the motion induced by camera rotation $\mathbf{R_c}$ removed through “rectifying” the second frame coordinate system to have the same orientation as the first frame; and $\mathbf{H_R} = \mathbf{K_0 R_c K_1}^{-1}$ 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. Eq. (2) states that the rectified 3D scene flow of a moving point $\mathbf{P}$ will not equal the negative camera translation. However, assuming camera intrinsics and motion are known, there are still two crucial degrees of freedom that are undetermined: depth Z_0 and Z_1.

Coplanar motion degeneracy: Solving for Z_0 and Z_1 equates to estimating the depth and 3D scene flow, which itself is challenging [34]. To remove such dependencies, 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, 47] or plane plus parallax (P+P) [42] 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. 2). Formally, 3D points that translate within 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. 3 Case (II).

To detect such co-planar motion, we make use of optical expansion cues that upgrade 2D flow to 3D as suggested by recent work [63]. 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 [63]. 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

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

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 their directions. 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. 3 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. 3 Case (IV). Such ambiguity remains even given multiple frames [65], but is common in many real-world applications, e.g., two cars passing each other (Fig. 2). To identify moving points in such cases, one could use depth Z_0 to recover the metric scale of normalized rectified 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. 2 (b). To do so, we contrast the flow-derived depth Z_0^{flow} with a depth prior Z_0^{prior},

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

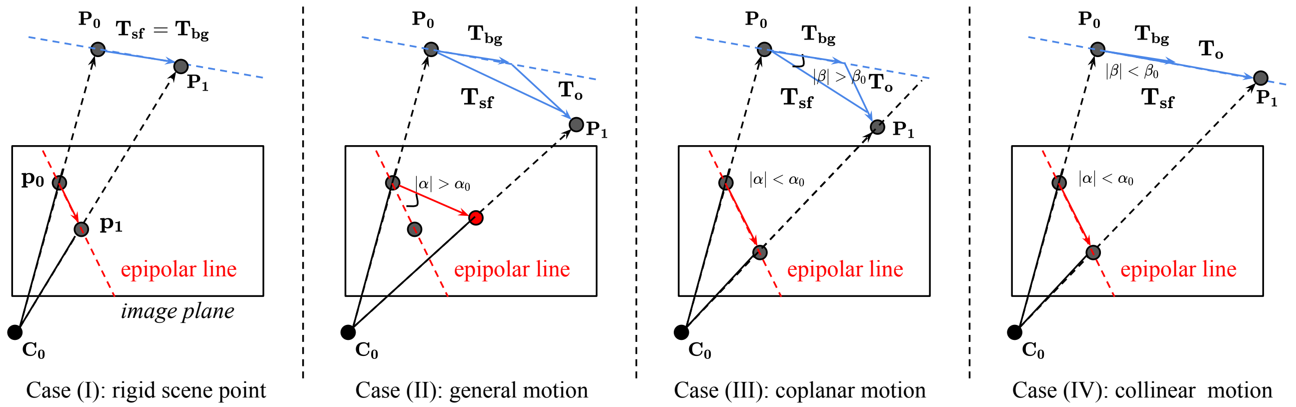


Figure 3: When can a moving scene point $\mathbf{P}$ be identified from a moving camera? Rectified 3D scene flow of $\mathbf{P}$ (that assumes camera rotation has been removed) can be written as the sum of rigid background motion (induced by the camera) and independent object motion $\mathbf{T_{sf}} = \mathbf{T_{bg}} + \mathbf{T_o}$, where $\mathbf{T_{bg}} = -\mathbf{T_c}$. 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 [63] – 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).

where Z_0^{flow} can be computed efficiently using midpoint triangulation [19], depth prior Z_0^{prior} can be represented by a data-driven monocular depth network, and the scale factor γ that globally aligns Z_0^{prior} to Z_0^{flow} can be determined by the ratio of their medians or robust least squares [44].

Egomotion degeneracy: Furthermore, 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 [47, 49].

3.2. Learning to Segment Rigid Motions

We now operationalize our motion analysis into a deep network for rigid motion segmentation (Fig. 4). 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 methods to estimate optical flow, optical expansion and monocular depth [38, 63]. To estimate camera motion, we fit and decompose essential matrices from flow maps using the five-point algorithm with a differentiable and parallel RANSAC [9].

Rigidity cost-maps inputs: We construct rigidity cost maps tailored to camera-object motion configurations analyzed in Sec. 3.1, including (1) an epipolar cost for general configurations, computed as per-pixel Sampson error [19]; (2) a homography cost to deal with small camera translations, implemented as per-pixel symmetric transfer error [14] with regard to the rotational homography, $\mathbf{H_R} = \mathbf{K_0 R_c K_1}^{-1}$; (3) a 3D P+P cost to detect coplanar motions, computed as

$$c_{\text{3D}} = ||\tilde{\mathbf{T}}_{\mathbf{sf}}|| \cdot |\sin\beta|, \tag{5}$$

where $\beta = |\angle(\tilde{\mathbf{T}}_{\mathbf{sf}}, -\mathbf{T_c})|$ is the measured angle between normalized scene flow $\tilde{\mathbf{T}}_{\mathbf{sf}}$ (computed by Eq. 3) and negative camera translation $-\mathbf{T_c}$; and (4) a depth contrast cost to deal with colinear motion ambituity, computed as

$$c_{\text{depth}} = |\log(\frac{Z_0^{\text{flow}}}{\gamma Z_0^{\text{prior}}})|. \tag{6}$$

Please refer to the supplement for visuals and more details.

Rectified scene flow inputs: Besides rigidity cost-maps, we find it helpful to also input raw scene flow measurements, represented as an 8-channel feature map, containing the first frame 3D scene points $\mathbf{P_0}$, rectified motion fields $\mathbf{T_{sf}}$, and uncertainty estimations of flow and optical expansion (σ_1, σ_2). To compute $\mathbf{P_0}$, we back-project the first frame pixel coordinates given monocular depth inputs; to compute $\mathbf{T_{sf}}$, we upgrade optical flow using optical expansion τ,

$$\tilde{\mathbf{T}}_{\mathbf{sf}} = \mathbf{K_0}^{-1}(\tau \mathbf{H_R}\tilde{\mathbf{p}}_{\mathbf{1}} - \tilde{\mathbf{p}}_{\mathbf{0}}), \tag{7}$$

where the second coordinate frame is rectified by rotational homography $\mathbf{H_R} = \mathbf{K_0 R_c K_1}^{-1}$ to remove the effect of camera rotation. Finally, the uncertainty of optical flow and optical expansion are computed as out-of-range confidence score and Gaussian variance respectively [22, 62]. Such rectified scene flow inputs are more effective than 2D optical flow, as empirically tested in ablation study (Tab. 4).

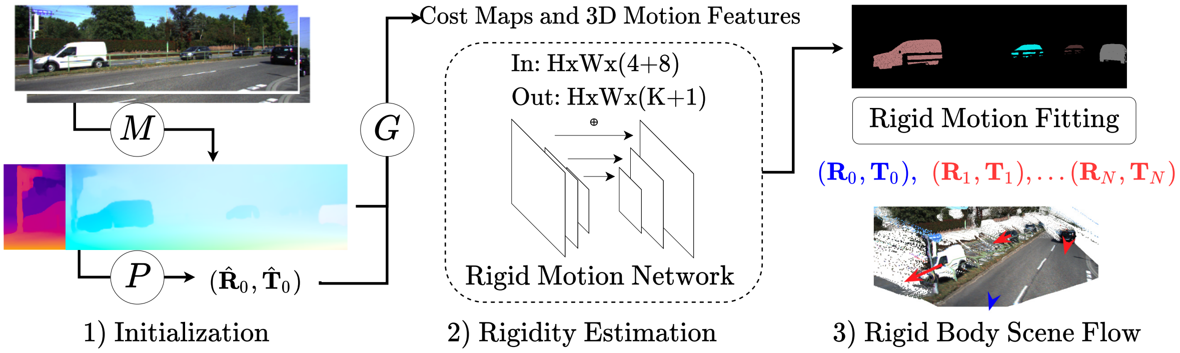

Cost Maps and 3D Motion Features
In: HxWx(4+8)
Out: HxWx(K+1)
Rigid Motion Network
Rigid Motion Fitting
M
G
P
$(\hat{\mathbf{R}}_0, \hat{\mathbf{T}}_0)$
$(\mathbf{R}_0, \mathbf{T}_0)$, $(\mathbf{R}_1, \mathbf{T}_1), \dots (\mathbf{R}_N, \mathbf{T}_N)$
1) Initialization
2) Rigidity Estimation
3) Rigid Body Scene Flow

Figure 4: We detect and estimate rigid motions in three steps: First, depth and optical flow are computed using off-the-shelf networks (M) and camera motion is estimated by epipolar geometry (P) given two frames. Then, rigidity cost maps and rectified scene flow are computed (G) 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 for the background and each rigid instance to update their depth and 3D scene flow.

Architecture: We use a two-stream architecture: (1) a lightweight U-Net [41] architecture to predict binary labels for pixels belonging to the (rigid) background and (2) a CenterNet [68] architecture to predict pixel instance masks. Inspired by the single-shot segmentation head proposed in PolarMask [59], stream (2) outputs a heatmap representing object centers and a regression map of $K = 36$ polar distances at evenly distributed angles. 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 incorrect during evaluation.

Losses: The overall architecture is end-to-end differentiable and can be trained with standard loss functions,

$$L = \alpha_1 L_{\text{binary}} + \alpha_2 L_{\text{center}} + \alpha_3 L_{\text{polar}} \tag{8}$$

where L_{binary} is binary cross-entropy loss with label balancing, L_{center} is the focal loss [28, 68] and L_{polar} is the polar loss defined in PolarMask [59]. Given ground-truth contours of M objects, we convert them to polar coordinates quantized as K rays uniformly emitted from their mass-centers. Then the polar loss is computed as

$$L_{polar} = \frac{1}{KM} \sum_{i=1}^{M} \sum_{k=1}^{K} |d_{i,k} - d^*_{i,k}|, \tag{9}$$

where $d^*_{i,k}$ is the ground-truth distance of the k-th ray to the mass-center. Weights are balanced as $\alpha_1 = 1^{-4}$, $\alpha_2 = 1^{-3}$ and $\alpha_3 = 1^{-7}$ through grid search.

Rigid body scene flow: Given segmentations of rigid bodies, our goal is to parameterize 3D scene flow as 3D geometry and transformations of rigid bodies by fitting flow and depth observations. We provide details in Alg. 1. To find the best fit of rotations and up-to-scale translations, we estimate and decompose essential matrices over flow correspondences with RANSAC [19]. To obain a more reliable 3D reconstruction than back-projected monocular depth, we triangulate flow using rigid motion estimations for each rigid body, and determine their scales by aligning each triangulated depth map to monocular depth inputs with RANSAC [44]. Given the above parameterization, the second frame coordinates are computed as

$$\mathbf{P_1} = \textstyle\sum_{i=0}^{N} \mathbf{S}_i(\mathbf{R_i P_0} + \mathbf{T_i}), \tag{10}$$

where $\mathbf{S}_i$ is a one-hot rigid motion segmentation vector.

Algorithm 1 Rigid body scene flow (monocular)

Input: Rigid body segmentation maps $\{\mathbf{S}_0, \dots, \mathbf{S}_N\}$, flow correspondence $(\mathbf{p}, \mathbf{p}')$, first frame depth map Z_{prior}, intrinsics $(\mathbf{K}, \mathbf{K}')$.

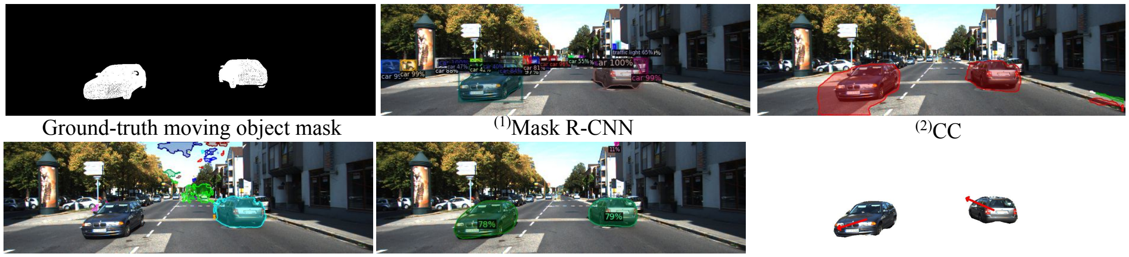
Output: Rigid transformation $\{(\mathbf{R}_0, \mathbf{T}_0) \dots, (\mathbf{R}_N, \mathbf{T}_N)\}$, first frame scene points $\{\mathbf{P}_0, \dots, \mathbf{P}_N\}$.

```
Normalize coordinates $\tilde{\mathbf{p}} \leftarrow \mathbf{K}^{-1}[\mathbf{p}, \mathbf{1}]^T$,   $\tilde{\mathbf{p}}' \leftarrow \mathbf{K}'^{-1}[\mathbf{p}', \mathbf{1}]^T$
For $i = 0 \cdots N$                          ▷ $i = 0$ indicates the rigid background
  $(\tilde{\mathbf{p}}_i, \tilde{\mathbf{p}_i}') \leftarrow \{(\tilde{\mathbf{p}}, \tilde{\mathbf{p}}'), \mathbf{S}_i(\mathbf{p}) = 1\}$        ▷ points on the current body
  Fit essential matrix $\mathbf{E}_i$ over $(\tilde{\mathbf{p}}_i, \tilde{\mathbf{p}_i}')$ with 5-pt+RANSAC.
  Decompose $\mathbf{E}_i$ and select the best $(\mathbf{R}_i, \mathbf{T}_i)$ by cheirality check [19].
  Triangulate 3D points $\mathbf{P}_i$ from $(\tilde{\mathbf{p}}_i, \tilde{\mathbf{p}_i}')$ and $(\mathbf{R}_i, \mathbf{T}_i)$.
  Align $\mathbf{P}_i^{(3)}$ to $Z_{prior}$ by scale $s_i$ with RANSAC.   ▷ scale ambiguity
  $\mathbf{T}_i \leftarrow s_i \mathbf{T}_i$,   $\mathbf{P}_i \leftarrow s_i \mathbf{P}_i$
  if $c_{\text{hom}} < 4$          ▷ when parallax motion is small: supp.mat. Eq. (2)
  then $\mathbf{T}_i \leftarrow \mathbf{0}$,   $\mathbf{P}_i \leftarrow Z_{prior}\tilde{\mathbf{p}}_i$            ▷ rely on depth prior
```

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- 4.3. In Sec. 4.4 we conclude with an ablation study.

Dataset: We use KITTI-SF (sceneflow) and Sintel for quantitative analysis. KITTI-SF [17, 34] 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


Ground-truth moving object mask

$^{(1)}$Mask R-CNN

$^{(2)}$CC

$^{(3)}$Flow angle

$^{(3)}$Ours

Our rigid motion estimation

Figure 5: 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 [39] correctly detects moving cars but also reports the edge of the road as moving objects. $^{(3)}$Geometric segmentation algorithm [5, 58] 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.

as dragons) that are rare in existing datasets. KITTI provides ground-truth segmentation masks for the rigid background 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 [58]. Both datasets also provide ground-truth depth and scene flow as well as camera intrinsics.

Implementation: We use MiDaS [38], a state-of-the-art monocular depth estimator to acquire imprecise, up-to-scale depth of the first frame as inputs. The remaining networks are trained without target domain data: optical flow and optical expansion networks are trained using FlyingChairs, SceneFlow, VIPER, and HD1K [13, 26, 33, 40]; the rigid motion segmentation network is trained on SceneFlow [33].

4.1. Two-frame Rigid Motion Segmentation

Metrics: Following prior works, we compute background IoU [31, 39] 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, 29, 57, 60]; U^{2}Net [37] is a state-of-the-art salient object detector; and MR-Flow-S [58] 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 [30] and MAT-Net [67] are SOTA video objection segmentation methods on DAVIS [36]. CC [39] 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 [31] 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.

Table 1: Rigidity estimation on KITTI (K) and Sintel (S) without fine-tuning. $^{(1)}$Single frame. $^{(2)}$Multi-frame with appearance features. $^{(3)}$Multi-frame without appearance. The best result under each metric (IoU in %) is in bold. *:For methods only estimating background masks, we use connected components to obtain object masks. ‡:Methods trained on target dataset. 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 [57]	88.20	96.42	81.98
	U^2 (Saliency) [37]	64.80*	93.34	82.01
	MR-Flow-S (K) [58]	75.59*	94.70‡	76.11
	MR-Flow-S (S) [58]	11.11*	84.72	92.64‡
(2)	FSEG [24]	85.08*	96.27	80.22
	MAT-Net [67]	68.40*	93.08	77.95
	COSNet [30]	66.67*	93.03	80.86
	CC [39]	50.87*	85.50	✗
	RTN [31]	34.29*	84.44	64.86
(3)	FSEG-Motion [24]	61.29	89.41	78.25
	CC-Motion [39]	42.99	74.06	✗
	Flow angle [5, 58]	25.83	85.52	74.23
	Ours	**90.71**	**97.05**	**86.72**

$^{(3)}$**Two-frame without appearance**. We separately evaluate the motion stream of FSEG and the flow-egomotion consensus results of CC. Following prior work [5, 58], 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

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 is in bold, and underlined if not trained on the target domain data. On Sintel, we evaluate on 330 frame pairs with average flow magnitude greater than 5 pixel.

Method	K: D1 ↓	K:SF ↓	S: D1 ↓	S:SF ↓
CC [39]	36.20	51.80	✗	✗
SSM [21]	31.25	47.05	✗	✗
Mono-SF [10]	**16.72**	**21.60**	✗	✗
MiDaS+OE [63]	37.27	44.87	49.89	55.43
MiDaS+Mask	17.33	22.47	39.60	47.40
MiDaS+Ours	16.98	22.19	**38.29**	**46.05**

and scenes, while ours is not.

Performance analysis: We show qualitative comparison in Fig. 5 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. Monocular Scene Flow

We then apply the estimated rigid motion masks to two-frame depth and scene flow estimation on KITTI and Sintel. Following Alg. 1, we estimate 3D scene flow by fitting rigid transformations to initial depth and optical flow estimations.

Metrics: Following the convention of KITTI [34], we arrange Sintel as pairs of adjacent frames, and report the average 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). 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 [38, 55].

Baselines: We compare against state-of-the-art monocular scene flow baselines. **CC** [39] 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) [63] 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 (**MiDaS+Mask**) that predicts instance segmentation masks by Mask R-CNN, and follows the same rigid body fitting procedure as ours.

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 in bold. *First frame disparity is not refined by our method.

Method	D1* ↓	D2 ↓	Fl ↓	SF ↓
PRSM [54]	4.27	6.79	6.68	8.97
OpticalExp [63]	1.81	4.25	6.30	8.12
DRISF [32]	2.55	4.04	4.73	6.31
Ours Mask R-CNN	1.89	3.42	4.26	5.61
Ours Rigid Mask	1.89	**3.23**	**3.50**	**4.89**

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 50% on KITTI, and 20% on Sintel. Compared to OE, which uses the same monocular depth input as ours, we are better in all metrics. (SF: 22.19% vs 44.87%), 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. Compared to Midas-Mask, our method is strictly better on both KITTI and Sintel, indicating the benefit of using our rigid motion masks versus appearance-based masks.

4.3. Stereo Scene Flow

Our method is also able to take advantage of reliable depth sensors, such as stereo cameras, to produce better two-frame rigid motion segmentation and scene flow estimations. To take advantage of stereo inputs, we make two algorithmic changes. First, we triangulate stereo disparities as the depth input to the segmentation network. Second, we refine each rigid body transformation by solving a perspective-n-point problem (via LM optimization):

$$\min_{(\mathbf{R}_i,\mathbf{T}_i)} \sum_j ||\mathbf{p}_{i,j}{}' - \pi_{\mathbf{K}}(\mathbf{R}_i\mathbf{P}_{i,j} + \mathbf{T}_i)||^2, \tag{11}$$

where $\pi(\cdot)$ is a projection function, $\mathbf{P}_{i,j}$ is the j-th point from the i-th rigid body, and $\mathbf{p}'_{i,j}$ is the second frame flow correspondence. We use the results of Alg. 1 as initial values and update rigid body transformations for 20 iterations.

Implementation: We use off-the-shelf networks that are fine-tuned on KITTI-SF to estimate stereo disparity and optical flow [63, 66]. We also fine-tune our rigid motion segmen-

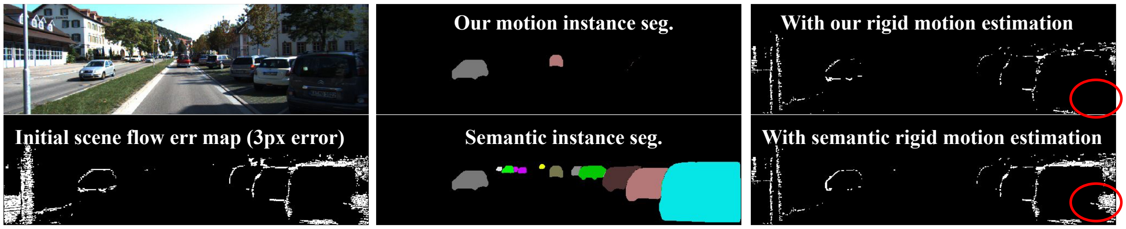

Our motion instance seg.
With our rigid motion estimation
Initial scene flow err map (3px error)
Semantic instance seg.
With semantic rigid motion estimation

Figure 6: 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.

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)w/o cost maps	88.66	96.59	76.81
(2)w/o uncertainty	85.09	95.72	77.25
(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 learning [5, 58]	25.83	85.52	74.23

tation network by mixing KITTI-SF and SceneFlow datasets. The performance is reported on the KITTI benchmark.

Baselines: Our method segments rigid motions based on two-frame rigidity and fits rigid body transformations over depth-flow correspondences, which is used to update the second frame depth and flow estimations. Among the baselines, **OE** [63] uses the same architecture (as in the monocular setup) fine-tuned on KITTI to upgrade optical flow to 3D scene flow. Same as ours, GA-Net stereo and VCN optical flow are used as inputs. **PRSM** [54] segments an image into superpixels, and fits rigid motions to estimate piece-wise rigid scene flow. Given semantic instance segmentation [20], depth, and optical flow, **DRISF** [32] casts scene flow estimation as an energy minimization problem and finds the best rigid transformation for each *semantic* instance. The key difference between our method and DRISF is that we use rigid motion segmentation masks.

Performance analysis: As reported in Tab. 3, our method demonstrates state-of-the-art performance on KITTI scene flow benchmark (SF: 4.89 vs 6.31). If we replace the segmentation masks with semantic instance segmentation, i.e., Mask R-CNN, the performance drops noticeably (SF: 4.89% to 5.61%). As illustrated in Fig. 6, 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, while semantic instance segmentation methods fail to do so.

4.4. 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)explicitly computed rigidity cost-maps inputs, (2)uncertainty estimation inputs, (3)monocular depth inputs, (4)optical expansion that upgrades 2D optical flow to 3D, and (5) our rigid motion segmentation network. (1)Removing rigidity cost-maps leads to a slight drop of accuracy on KITTI, and a significant 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 which cases explicit geometric priors are helpful. (2)Removing uncertainty inputs leads to a noticeable drop of performance on KITTI (88.66% to 85.09%). We posit uncertainty estimation contains rich information about motion distribution, and is therefore useful for segmentation. (3)Further removing monocular depth inputs leads to an accuracy drop on all metrics, especially on KITTI, which shows the importance of using depth cues to deal with collinear motions in autonomous driving scenes. (4)After further removing optical expansion, our method degrades to MoA-Net [6]. The performance drops noticeably on KITTI rigid instance segmentation metric (84.46% to 81.28%), which indicates optical expansion is useful for segmenting foreground objects. (5)Lastly, if we directly apply the rigidity cost maps with manually-tuned thresholds to decide the background region without the neural architecture and learning, the method becomes worse in all metrics due to the loss of robustness to noisy inputs and degenerate motion.

5. Conclusion

We investigate the problem of two-frame rigid body motion segmentation in an open environment. We 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.

Acknowledgements: This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research. We thank Rui Zhu for proving the code of single-image camera intrinsics estimation. We thank Jason Zhang, Tarasha Khurana, Jessica Lee and many others for their useful feedback.

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. 2

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

[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, 8

[6] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moanet: self-supervised motion segmentation. In *ECCVW*, 2018. 2, 8

[7] Pia Bideau, Aruni RoyChowdhury, Rakesh R Menon, and Erik Learned-Miller. The best of both worlds: Combining cnns and geometric constraints for hierarchical 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. 4

[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. 4

[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. 6

[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. 6, 8

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

[22] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *ECCV*, 2018. 4

[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. 2, 6

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

[26] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *CVPRW*, 2016. 6

[27] 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

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5

[29] 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

[30] 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. 2, 6

[31] 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. 6

[32] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *CVPR*, 2019. 2, 7, 8

[33] 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

[34] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2, 3, 5, 7

[35] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 2013. 2

[36] 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. 2, 6

[37] 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

[38] 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, 4, 6, 7

[39] 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

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

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

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

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

[44] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 4, 5

[45] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 2

[46] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *IJCV*, 2019. 2

[47] 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

[48] 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

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

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

[51] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *ICCV*, 1999. 2

[52] René Vidal, Yi Ma, Stefano Soatto, and Shankar Sastry. Two-view multibody structure from motion. *IJCV*, 2006. 2

[53] René Vidal and Shankar Sastry. Optimal segmentation of dynamic scenes from two perspective views. In *CVPR*, 2003. 2

[54] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *IJCV*, 2015. 2, 7, 8

[55] 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

[56] Joseph Weber and Jitendra Malik. Rigid body segmentation and shape description from dense optical flow under weak perspective. *TPAMI*, 19(2):139–143, 1997. 2

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

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

[59] 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

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

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

[62] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 4

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

[64] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *CVPR*, 2019. 2

[65] 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

[66] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided aggregation net for end-to-end stereo matching. In *CVPR*, 2019. 7

[67] 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. 2, 6

[68] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 5