

## Instantaneous Perception of Moving Objects in 3D

Di Liu<sup>1</sup> Bingbing Zhuang<sup>3</sup> Dimitris N. Metaxas<sup>1</sup> Manmohan Chandraker<sup>2,3</sup>  
<sup>1</sup>Rutgers University <sup>2</sup>University of California, San Diego <sup>3</sup>NEC Labs America

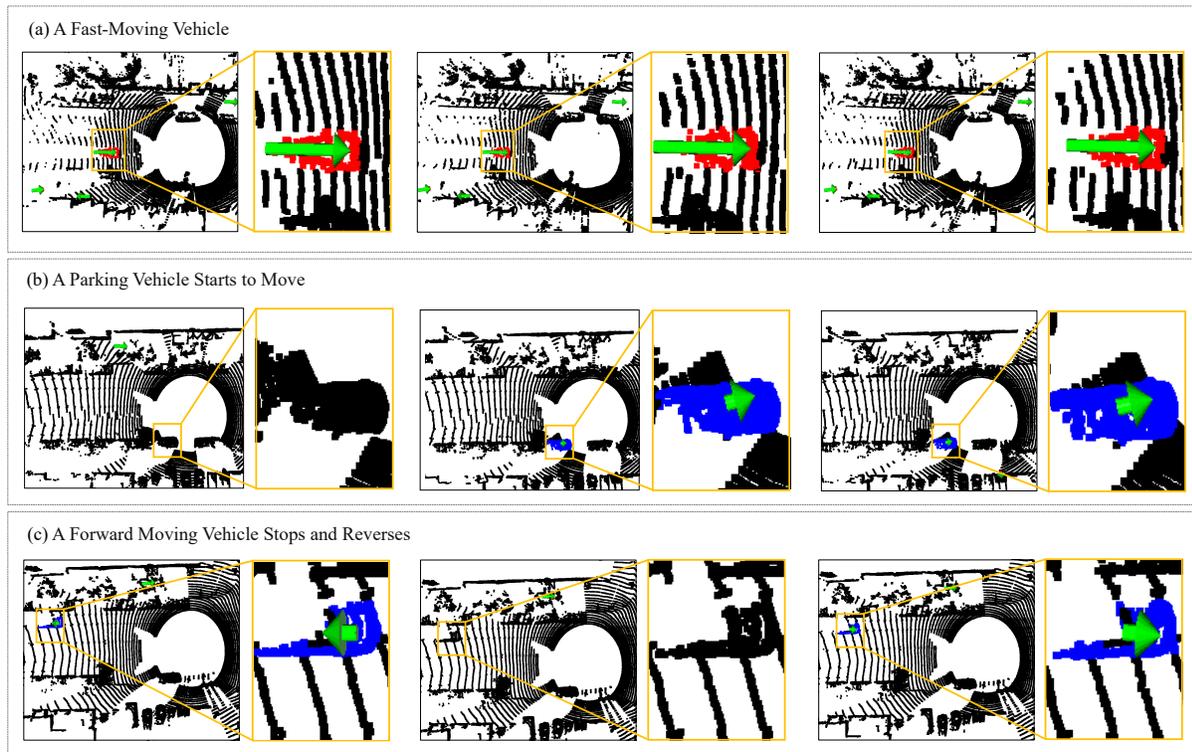


Figure 1. **Illustration of Instantaneous Motion Perception.** We visualize motion of three objects from Waymo dataset [46], each with three consecutive frames. Objects in fast and subtle motions are marked as red and blue, respectively, with arrow length indicating motion magnitude. While standard motion detection handles general large motions such as (a), we focus on instantaneous perception of subtle motions that may indicate changes in driving behavior, for example (b) parking car starts to move, and (c) forward moving car stops and reverses. The visualized subtle motions (b)(c) are output from our framework. We also provide the video visualization in supplementary.

### Abstract

The perception of 3D motion of surrounding traffic participants is crucial for driving safety. While existing works primarily focus on general large motions, we contend that the instantaneous detection and quantification of subtle motions is equally important as they indicate the nuances in driving behavior that may be safety critical, such as behaviors near a stop sign of parking positions. We delve into this under-explored task, examining its unique challenges and developing our solution, accompanied by a carefully designed benchmark. Specifically, due to the lack of correspondences between consecutive frames of sparse Lidar point clouds, static objects might appear to be moving – the

so-called swimming effect. This intertwines with the true object motion, thereby posing ambiguity in accurate estimation, especially for subtle motions. To address this, we propose to leverage local occupancy completion of object point clouds to densify the shape cue, and mitigate the impact of swimming artifacts. The occupancy completion is learned in an end-to-end fashion together with the detection of moving objects and the estimation of their motion, instantaneously as soon as objects start to move. Extensive experiments demonstrate superior performance compared to standard 3D motion estimation approaches, particularly highlighting our method’s specialized treatment of subtle motions.

## 1. Introduction

Human drivers pay special attention to surrounding moving objects to understand and predict their driving behavior, and react accordingly to avoid collisions. Similarly, intelligent autonomous systems must also navigate safely through traffic scenes, where preventing collisions with moving objects is considerably more complex than with static background scenes. This gives rise to several lines of computer vision research centered around motion, ranging from low-level tasks like 3D scene flow [24], to middle-level motion segmentation or detection [49], and high-level perception on 3D object tracking [54].

However, these methods are designed to handle general 3D motion without considering the extent and context of the motion. In this paper, we would like to focus on an important subset of motion – small subtle motions. Such motions are of significance as they often indicate changes in driving intention or behavior; for instance, as illustrated in Fig. 1, parking vehicles start to move and cut into the driving lane, or vehicles in the driving lane start to reverse back for reverse parking. As a more general note, the instant capture of all nuanced changes happening in the scene is essential for situation awareness, especially in corner-scenario cases. This however remains under-explored in computer vision, thus motivates our research in this paper, which aims to detect the presence of subtle motions as well as estimate their motion flow instantaneously.

While prominent motions from fast-moving objects are more feasible to detect and quantify due to a strong signal-to-noise ratio, recovering subtle motions with high accuracy presents its challenges. Specifically, the Lidar sensor captures only a sparse point set of the surrounding scene elements, and the pattern of points varies depending on the relative position between the Lidar and the scene. Consequently, there are typically no point correspondences across frames with a moving Lidar sensor, even for static scene elements. This further implies that static objects may appear to be moving, known as the swimming artifact [8, 21]. It intertwines with and hence obfuscates the true object motion, especially under small motions; as such, we empirically observe that a model trained for general motions does not perform as well with subtle motions.

To address this, our framework proposes to learn shape completion before performing motion detection and estimation. Taking sequential frames of Lidar point clouds within a short period as input, our method voxelizes the point clouds as occupancy grids, and leverages the accumulated Lidar points from nearby frames to generate a denser occupancy grid, which is then applied as supervision for occupancy completion. This effectively densifies and enhances the surface cue to mitigate the impact of the swimming artifact. To prevent the noises brought about by imperfect completion from affecting the downstream motion estimation

task, we refrain from performing full object shape completion but rather do so locally only for the visible surface part, where the major motion signals lie. Our framework takes the point clouds of each object individually as input to our network, which is trained exclusively on the regime of small motion. We name our method S'More, indicating subtle motion regressor. Since there is no standard training dataset and evaluation benchmark specific to subtle motions. We contribute one by extracting small motions from the large-scale Waymo dataset [46], leveraging its existing annotations. We demonstrate the efficacy of the proposed method with the newly proposed benchmark.

In summary, our contributions include:

- Introduce the task of detecting and estimating subtle motions for vehicles, with insights on their practical significance and challenges.
- Design a framework with occupancy completion to mitigate the swimming artifacts on small motion estimation.
- Translate our insights into favorable performance in a new evaluation benchmark tailored for small motions.

## 2. Related Work

**3D Scene Flow.** 3D scene flow aims to estimate the motion field of each observed 3D point. It is an important tool for analyzing scene dynamics and has been extensively studied in computer vision [10, 17, 23, 25, 33, 41, 52, 57]. While the scene flow of background points as the dominant rigid motion may be reliably estimated [8], accurately estimating the motion flow for dynamic foreground objects remains a challenge. This leads to object-aware scene flow works [3, 22, 23, 48] that leverage rigidity prior of objects. However, the nearest neighbor-based approach for motion estimation like scene flow suffers from inherent ambiguity brought by the “swimming” effect [21] of Lidar point clouds, which is more severe with smaller motion magnitude. In this work, we develop insights towards addressing this issue.

**Moving Object Detection.** Moving object detection is an essential capability for autonomous vehicles and other areas, which results in many prior works, such as [7, 13, 15, 20, 29, 38, 39, 45, 49]. However, they detect motions at a coarse level for general large motions. SemanticKITTI [4], a commonly used dataset in this field, labels moving object in a coarse sequence level instead of in an instantaneous manner. In contrast, our approach offers special treatment for detecting small motions instantaneously. Another possible way for moving object detection is through 3D detection [14, 18, 27, 28, 34, 37, 40, 58] and tracking [19, 54]. However, we found empirically that such methods stumble in identifying small motions due to imperfect object localization. Lastly, we also note a concurrent work, M-detector [53], that instantly detects point-level moving events based on occlusion principles.

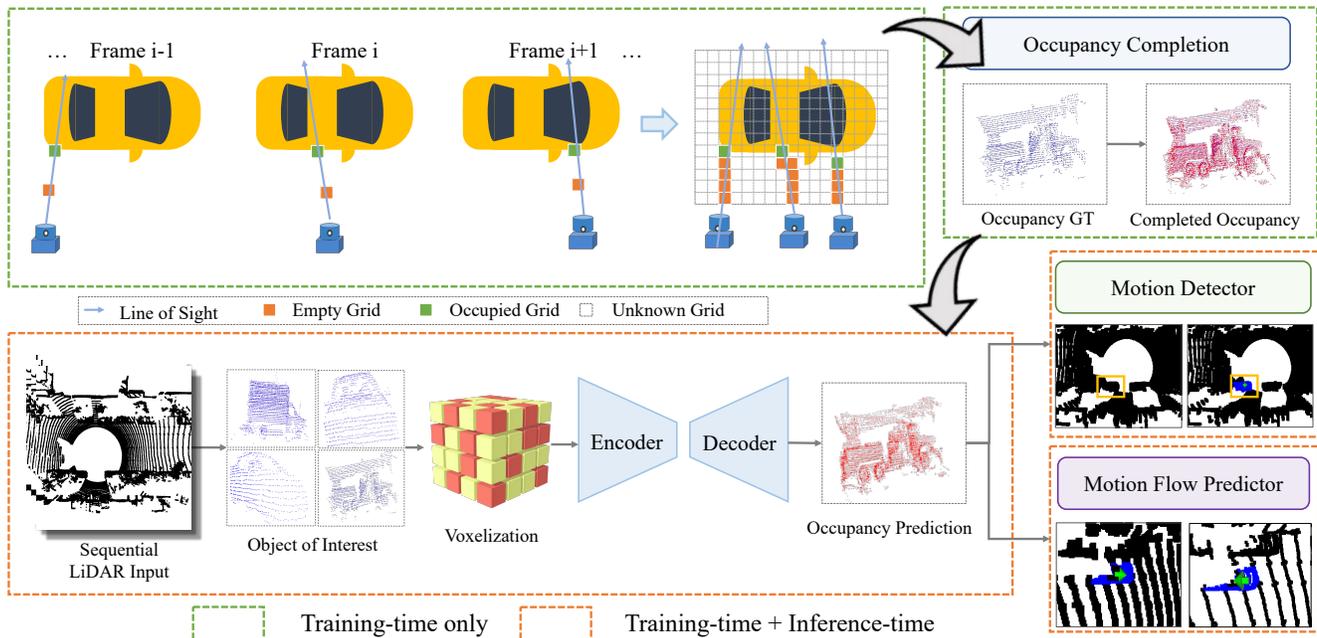


Figure 2. **Overview of S'More.** Given a sequential LiDAR point cloud, we first identify objects of interest by filtering out background and objects with large motion. We then voxelize the point cloud for each object, extract features through an encoder-decoder network, and perform occupancy completion. The output is passed to a motion detector and a motion flow predictor for final detection and estimation.

**Occupancy Prediction.** Occupancy is an effective 3D scene representation that has wide applications in autonomous driving. Argo et al. [1] and Reza et al. [36] apply the occupancy flow field for perception and motion forecasting. [21] performs 4D occupancy forecasting supervised by point cloud forecasting. ALSO [6] utilizes occupancy completion as a tool for self-supervised feature learning for Lidar point clouds. Another line of work [47, 50, 51] learns occupancy prediction from monocular cameras. In this work, we present the first known attempt to use occupancy completion to facilitate the estimation of small motions from LiDAR point clouds.

**Subtle Motions in General.** The field of computer vision has shown a long history of interest in small motions in a broader context. Differential structure-from-motion [35, 60] aims to recover instantaneous camera motion from optical flow. Several works [9, 16, 30–32, 55] utilize accidental camera motion to perform 3D reconstruction. Another line of research [12, 26] targets magnifying invisible subtle motions in videos. Our work focuses on perceiving the subtle motions of surrounding vehicles, a capability critical to the safety of autonomous systems.

### 3. Method

#### 3.1. Problem Definition and Challenges

Our goal is to identify moving objects and estimate their motion using sequential point clouds; we focus in this pa-

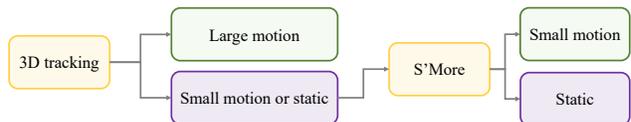


Figure 3. **Integration of S'More with 3D tracking systems.**

per on vehicles while leaving the human category for future work. Unlike existing works, we concentrate on small motion for instantaneous detection as objects begin to move. As a preprocessing step, our framework filters out fast-moving objects, thereby targeting static and slow-moving ones. Fig. 3 illustrates a practical use case of such setting in a 3D tracking system. Further, we make a practical assumption that the ego-vehicle’s motion can be reliably estimated by ICP, possibly aided by GPS/INS, as validated by recent studies [8]. This allows for the exclusion of ego-motion from the observed object motion *w.r.t.* the world coordinate. Our method is object-centric, processing point clouds from five consecutive frames ( $F_t, t=1, \dots, T$ ) to classify objects as static or moving. For moving objects, we estimate the motion flow from  $F_1$  to  $F_T$  for each point in  $F_1$ , setting  $T=5$  as per [20].

**Swimming Effect.** The detection and estimation of small motions present its challenges, primarily arising from the sparse nature of Lidar point clouds. Remarkably, the spatial distribution of captured points closely depends on the

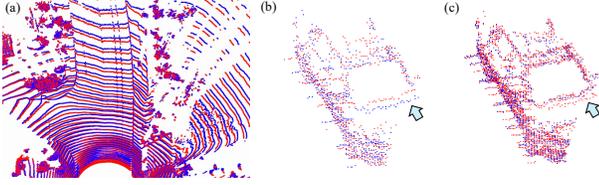


Figure 4. **Illustration of Swimming Effect** on ground (a) and static object (b), and (c) our ground truth occupancy completion for (b). Blue and red points indicate points from two frames.

relative position between the Lidar sensor and surrounding scene elements. Hence, as the Lidar sensor moves along with the ego-vehicle, there are typically no exact point correspondences across frames, and even static scene elements may appear to be moving. This effect manifests itself on both background scene and foreground objects, as illustrated in Fig. 4 (a)(b). In particular, the ground points visually appear to be swimming across frames, hence termed as *swimming effect* [21]. This effect poses challenges, especially to characterizing subtle motions as one would need to distinguish the true object motion from this effect. We note that the sparse nature of Lidar points distinguishes the problem from optical flow [11], where dense correspondences exist and small motions simplify the flow estimation through brightness consistency assumption.

### 3.2. Our Framework

Fig. 2 illustrates the overview of S'More. Given five consecutive Lidar frames, we filter out fast-moving objects and for each remaining ones, our network estimates the small motion or the lack thereof. We start by voxelizing the input point clouds, followed by feature extraction using an encoder-decoder network, and then perform occupancy completion, the output of which is passed to the motion segmentation head and the instantaneous flow estimation head.

#### 3.2.1 Occupancy Completion

**Input Voxelization.** First, following state-of-the-art 3D detection frameworks [34, 54], we voxelize the point cloud  $\mathbf{X}_t \in \mathbf{R}^{N \times 3}$  at each frame as a binary grid of size  $[W_x, W_y, W_z]$ , with the voxels containing Lidar points filled with 1, and 0 elsewhere. This grid may be viewed as an incomplete occupancy grid, in that it indicates part of the visible object surface captured by Lidar at a single timestamp. We stack consecutive frames to form a spatial-temporal grid of size  $[T, W_x, W_y, W_z]$ .

**Local Occupancy Completion.** Recall that the sparse nature of Lidar point clouds poses challenges to accurate small motion estimation, due to the actual subtle motion intertwined with the swimming effect. To mitigate this issue, our framework first learns occupancy completion that effec-

tively densifies the object surface, to offer stronger cues for subsequent networks to reason correspondence, and hence the motion between frames.

Before proceeding, one should be mindful of the potential trade-off brought by this step – the estimated occupancy completion may well be imperfect, introducing extra noise to the system. This may be harmful to the final motion segmentation and estimation if the noises reach a certain level, hence defeating the purpose of occupancy completion. Therefore, while the standard shape completion problem (e.g. [42]) is tasked to recover the entire object shape from a single-frame input, it is an overly complicated and unnecessary task in our case, besides the infeasibility of getting the ground truth in real driving scenes. Instead, since the  $T$  Lidar frames collectively observe only a local part of the object within an instantaneous timeframe, we target local occupancy completion at these observed regions, while refraining from hallucinating areas invisible to all  $T$  Lidar frames. This way, we enhance the critical signal essential for motion characterization while minimizing the extra noises from imperfect completions.

**Supervision for Occupancy.** We densify the local occupancy grid by leveraging nearby frames, as shown in Fig. 2. Specifically, for each frame  $F_t, t \in [1, \dots, T]$  in the input window, we warp all Lidar points from the rest  $T - 1$  frames to  $F_t$ , using the ground truth object motion (recall that ego-motion has been factored out), and then mark the corresponding voxel as occupied, i.e. 1. In addition, we mark the points along the line-of-sight as empty i.e. 0. All other voxels are deemed as unknown. We apply a fast voxel traversal algorithm [2] to implement this step, similarly as in [21], with an example illustrated in Fig. 4(b)(c). Note we only use the Lidar frames inside the input window to generate the target occupancy grid, to simplify the task. By learning occupancy, the network is explicitly enforced to learn the notion of dense shapes in an end-to-end manner, thereby facilitating the task of motion detection and estimation.

#### 3.2.2 Network Architecture and Losses

**Network Architecture.** We apply an encoder-decoder for occupancy grid prediction, which is passed to another encoder-decoder for motion detector and motion flow predictor. The motion detector classifies input objects as static or moving, while the flow estimator regresses a motion vector for each occupied voxel in the grid. We then extract the motion flow for each raw input point as the predicted flow in the voxel that point resides in. Note that we do not enforce rigidity constraints on the flow field, maintaining the method's generality, though we do evaluate the setting with rigidity prior as well. We utilize the encoder-decoder structure as in [56], consisting of simple convolutional layers with skip connection; we follow [21] to treat the height

and temporal dimension as the channel dimension, which allows to use 2D convolutional layers for efficiency; see supplementary for details. Our network processes each object separately, but remains efficient and runs at 27 fps for a scene consisting of 30 objects of interest.

**Overall Losses.** The overall loss function of our model is a weighted combination of five terms:

$$\mathcal{L} = \lambda_{\text{occ}}\mathcal{L}_{\text{occ}} + \lambda_{\text{mot}}\mathcal{L}_{\text{mot}} + \lambda_{\text{epe}}\mathcal{L}_{\text{epe}} + \lambda_{\text{rel}}\mathcal{L}_{\text{rel}} + \lambda_{\text{ang}}\mathcal{L}_{\text{ang}}. \quad (1)$$

Specifically, we apply a binary cross-entropy (BCE) loss  $\mathcal{L}_{\text{occ}}$  for the occupancy grid prediction, a BCE loss  $\mathcal{L}_{\text{mot}}$  on static/moving object classification, a  $L_1$  loss  $\mathcal{L}_{\text{epe}}$  and a scale-aware  $\mathcal{L}_{\text{rel}}$  loss on motion flow prediction for moving objects. Additionally, since the motion direction carries important information about driving intention such as reversing or left/right turning, we add an angular loss  $\mathcal{L}_{\text{ang}}$  for the motion flow. We denote the set of occupied and empty voxels as  $\phi_o$  and  $\phi_e$ , respectively.

**Occupancy Loss** is written as

$$\mathcal{L}_{\text{occ}} = \mathbb{E}_{v \in \{\phi_o, \phi_e\}} \left[ \hat{O}_v \log(O_v) + (1 - \hat{O}_v) \log(1 - O_v) \right], \quad (2)$$

where  $O_v$  and  $\hat{O}_v$  indicate the predicted and ground truth occupancy at voxel  $v$ .

**Flow Prediction Losses.** For each voxel  $v$ , we define the ground truth flow (denoted as  $\hat{\mathbf{f}}_v$ ) as the average of the ground truth flow associated with the points falling into that voxel. The relative flow loss  $\mathcal{L}_{\text{rel}}$  is written as

$$\mathcal{L}_{\text{rel}} = \mathbb{E}_{v \in \phi_o} \frac{\|\hat{\mathbf{f}}_v - \mathbf{f}_v\|_2}{\|\hat{\mathbf{f}}_v\|_2 + \varepsilon}, \quad (3)$$

where  $\hat{\mathbf{f}}_v$  indicates predicted flow at voxel  $v$  and  $\varepsilon$  is a small constant. Note this loss is inverse scaled by the flow magnitude to emphasize learning on small motions. And the angular loss is written as

$$\mathcal{L}_{\text{ang}} = \mathbb{E}_{v \in \phi_o} \text{acos} \left( \frac{\langle \mathbf{f}_v, \hat{\mathbf{f}}_v \rangle}{\|\mathbf{f}_v\|_2 \cdot \|\hat{\mathbf{f}}_v\|_2 + \varepsilon} \right) \quad (4)$$

where  $\langle \cdot, \cdot \rangle$  denotes the dot product between the vectors.

## 4. Experiments

### 4.1. Evaluation of S'More

**Evaluation Benchmark.** In the absence of existing benchmark dedicated to subtle motions, we curate one such dataset from the Waymo open dataset [46], where each sequence provides Lidar frames at 10Hz for about 20s. We collect the point clouds for each object from every five consecutive frames (0.5s) denoted as  $F_i, i=[1, \dots, 5]$ . To generate ground truth motion status, we follow [20] to derive the

Table 1. **Quantitative Comparisons** with ICP, Point-to-Plane ICP, Generalized ICP, CenterPoint and FastNSF.

	EPE ( $\downarrow$ )	Angle Error ( $\downarrow$ )	F1 Score ( $\uparrow$ )
FastNSF [24]	0.1189	0.5592	0.6180
ICP [5]	0.0554	0.4499	0.7456
Point-to-Plane ICP [43]	0.2263	0.4379	0.7856
Generalized ICP [44]	0.1117	0.4170	0.7693
CenterPoint [54]	0.0927	0.5622	0.7270
S'More	<b>0.0437</b>	<b>0.3189</b>	<b>0.8323</b>

spatial transformation from the 3D boxes annotations, and hence compute the scene flow  $\mathbf{f}_i$  from  $F_1$  to  $F_5$  for every point  $\mathbf{x}_i$  in  $F_1$ . To concentrate on small motions, we deem the data sample valid only when the minimal flow magnitude  $f_{\min} = \min_{\mathbf{x}_i \in F_1} \|\mathbf{f}_i\|$  is less than 0.2m. Further, we label the object as static if  $f_{\min} < f_{\text{thre}}$ . We set  $f_{\text{thre}} = 0.05\text{m}$  but also evaluate under other settings shortly. This way, we collect about 140k and 9k samples for training and test sets, respectively. More details are in the supplementary.

**Evaluation Metrics.** We apply the standard F1 score to measure the accuracy of static/moving object classification. We apply end-point error (EPE) and the angular error to measure the object motion flow error.

**Baselines.** In the absence of existing detection methods dedicated to small motions, we mainly compare with: (i) the classical Iterative Closest Point (ICP) [5], which remains competitive [8] for motion flow task; (ii) the point-to-plane ICP [43] and the generalized ICP [44] implemented in Open3D [59]; (iii) the leading scene flow method FastNSF [24]; (iv) the detection and tracking based method CenterPoint [54], where we use ground-truth tracking by associating detected objects with the ground truth, and the motion flows are derived from boxes transformation. For all methods we use their output motion flows to detect moving objects, according to the aforementioned criterion.

**Comparison.** Tab. 1 shows quantitative evaluation results, indicating the significantly superior performance of our model compared to the baselines. We note that the object-tracking method CenterPoint gives decent accuracy but lags behind S'More, likely because their imperfect object localization causes ambiguity in distinguishing small motions from static ones. In Fig. 5, we provide a visualization comparison with ICP and FastNSF. The input comprises two sets of point clouds: the first frame (in red) and the last frame (in green). We visualize the flow accuracy through alignment – we shift the red points with the flow and the resultant points (marked as blue) should ideally align well with the green points if the flows are correct. Our model demonstrates superior alignment accuracy, especially at the subtle level of local registration, attributable to its advanced motion estimation capabilities. More examples are provided in the supplementary material.

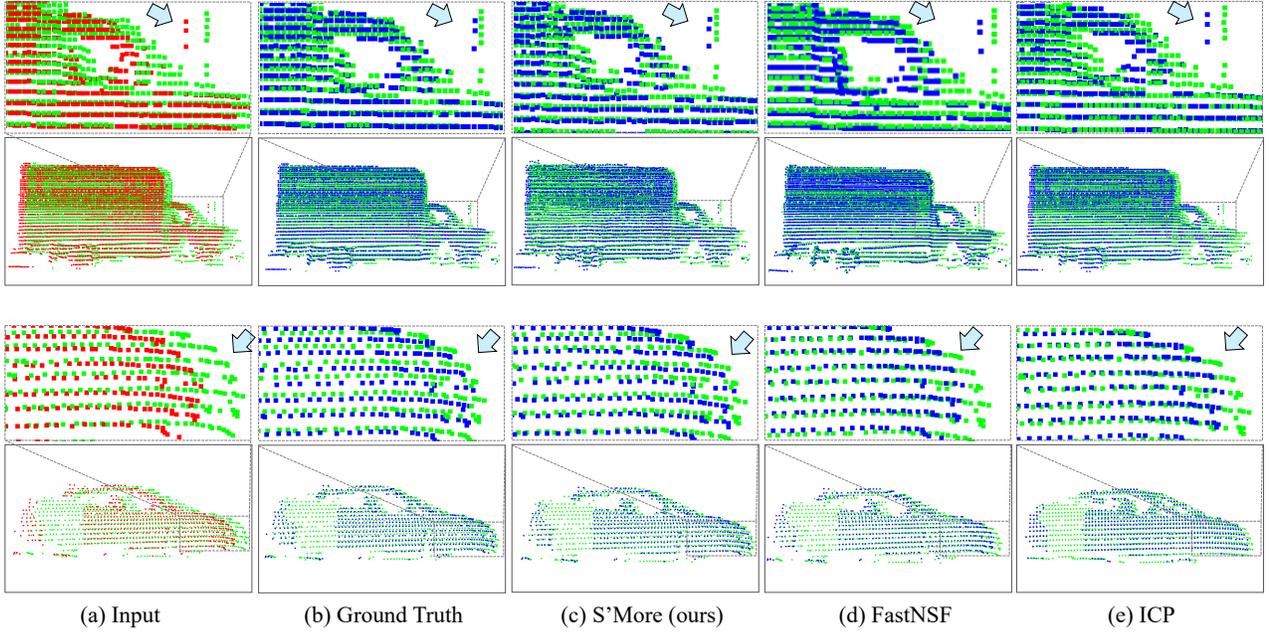


Figure 5. **Qualitative Comparison.** We exhibit point cloud registration results for two point cloud sets: the first frame (in red) and the last frame (in green). The results are shown using (b) ground truth motion, and estimated motions by (c) S'More(ours), (d) FastNSF, and (e) ICP. The blue points indicate resultant positions after adding flow to the red points, which should ideally align with the green points.

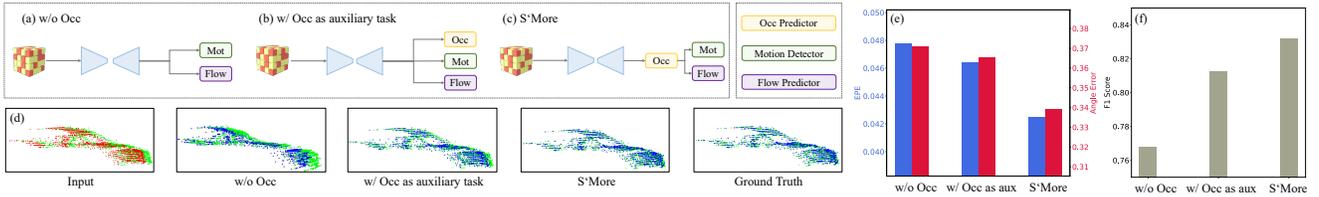


Figure 6. **Ablation study on Occupancy Completion.** (a)(b)(c) illustrate structure of having the occupancy module removed or as an auxiliary task. (d)(e)(f) show the qualitative and quantitative comparisons; the visualization in (d) follows the same protocol in Fig. 5.

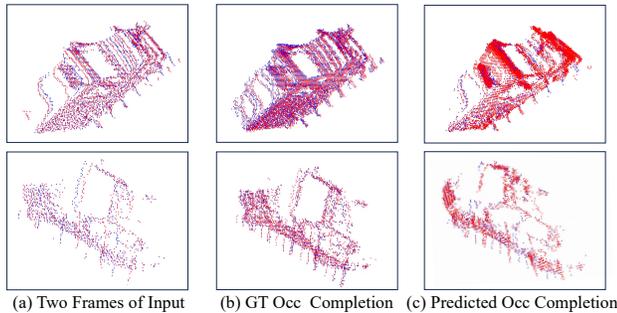


Figure 7. **Example Results of Occupancy Completion.** The blue and red points represent different frames in the input.

## 4.2. Ablation study on Occupancy Completion

To investigate the impact of occupancy completion, we *remove* this module from S'More, with structure shown in Fig. 6(a) in relation to S'More in Fig. 6(c). Further, we also evaluate the setting with the occupancy completion as just

an auxiliary trained in parallel with the motion detector and flow predictor, as shown in Fig. 6(b). We report the accuracy in Fig. 6(e)(f), which indicates the significant impact of occupancy completion towards good performance. We attribute this to its role in effectively densifying object surfaces. In Fig. 6(d) we provide visualization of point cloud registration to evaluate the estimated motion, further supporting the efficacy of occupancy completion. The qualitative results of the occupancy completion itself are demonstrated in Fig. 7.

**False Positive/Negative.** This analysis explores how incorporating occupancy impacts the reduction of false positives and negatives. In Fig. 8(a), we display three sequential frames that highlight how the model, without occupancy, erroneously detects a stationary vehicle as moving, marked by a red box. Conversely, when employing dense occupancy estimation, the model correctly identifies the stationary nature of the vehicle, reducing the false positives. Similarly, in Fig. 8(b), we show that without occupancy, the model fails

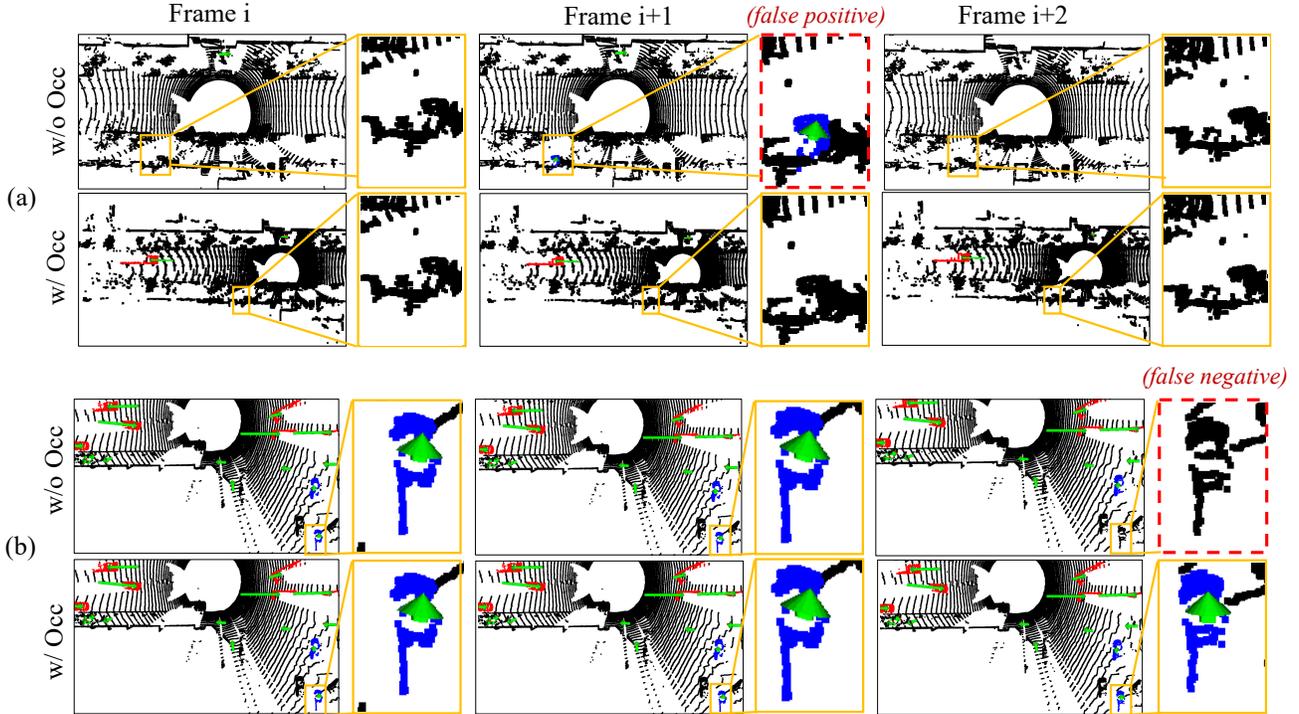


Figure 8. **Visualization of false positive/negative samples in the absence of occupancy completion.** Each row shows three consecutive frames, one per column. The motion of objects is marked with blue for detected subtle motions and red for GT large motions, with the arrow length representing the motion’s magnitude. In (a), a false positive occurs where a stationary vehicle is mistakenly marked as moving. In (b), a moving vehicle is incorrectly identified as static, a false negative. Both of them are rectified with the occupancy completion; note the ground truth is same as the correct detection here hence is not visualized.

Table 2. **Ablation Study of Losses.** This study observes a correlation between decreasing flow thresholds  $f_{\text{thre}}$  and the degradation of model performance. Notably, our model achieves the best average performance over five thresholds in terms of all metrics.

$f_{\text{thre}}$	End-Point-Error (EPE) ( $\downarrow$ )						Angle Error ( $\downarrow$ )						F1 Score ( $\uparrow$ )					
	0.05	0.04	0.03	0.02	0.01	Avg	0.05	0.04	0.03	0.02	0.01	Avg	0.05	0.04	0.03	0.02	0.01	Avg
w/o $\mathcal{L}_{\text{occ}}$	0.0492	0.0443	0.0440	0.0489	0.0439	0.0461	0.3489	0.3455	0.3828	0.4219	0.4405	0.3879	0.8256	0.8202	0.8025	0.7575	0.7603	0.7932
w/o $\mathcal{L}_{\text{angle}}$	0.0458	0.0441	0.0452	0.0444	0.0454	0.0450	0.3469	0.3654	0.4052	0.4295	0.4860	0.4066	0.8344	<b>0.8338</b>	<b>0.8111</b>	0.7704	0.7841	0.8068
w/o $\mathcal{L}_{\text{rel}}$	0.0446	0.0435	0.0439	0.0437	0.0430	0.0437	<b>0.3179</b>	0.3395	<b>0.3780</b>	0.4156	<b>0.4291</b>	0.3760	<b>0.8378</b>	0.8307	0.7937	0.7907	0.7847	0.8075
S'More	<b>0.0437</b>	<b>0.0425</b>	<b>0.0436</b>	<b>0.0428</b>	<b>0.0421</b>	<b>0.0429</b>	0.3189	<b>0.3392</b>	0.3834	<b>0.4053</b>	0.4324	<b>0.3758</b>	0.8323	0.8320	0.8003	<b>0.7997</b>	<b>0.7857</b>	<b>0.8100</b>

to detect a moving vehicle as static. This is rectified using the model with occupancy. This comparison underscores the effectiveness of using occupancy data in enhancing motion detection accuracy in our model. An interesting observation we found is that the occurrence frequency of false positives is much higher than false negatives. We attribute this to the swimming effect, where static objects appear to be moving, especially under subtle motion conditions.

### 4.3. Analysis on Large Motion

**Performance on large motion.** While our focus is on small motions, it would be more complete to study its performance on large motions as well. It is of particular interest to compare against 3D object tracking based method given the practical system illustrated in Fig. 3. To this end, we

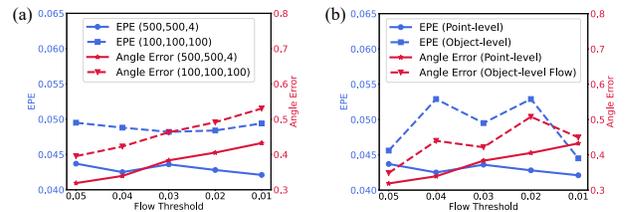


Figure 9. **Performance Analysis** of (a) occupancy with different grid sizes and (b) point/object-level flow estimation. We evaluate under different value of  $f_{\text{thre}}$  for comprehensiveness.

train another model of S'More including large-motion data and evaluate only on the large motion ( $f_{\text{min}} > 0.2\text{m}$ ) regime. As shown in Tab. 3, both S'More and CenterPoint achieve nearly perfect detection accuracy ( $F1 \rightarrow 1.0$ ), as expected

Table 3. Quantitative evaluation on large motions.

	EPE ( $\downarrow$ )	Angle Error ( $\downarrow$ )	F1 Score ( $\uparrow$ )
CenterPoint [54]	<b>0.1155</b>	<b>0.0247</b>	<b>0.9998</b>
S'More	0.2790	0.0486	0.9847

Table 4. Benefits of having small-motion-specific training data.

Training dataset	EPE ( $\downarrow$ )	Angle Error ( $\downarrow$ )	F1 Score ( $\uparrow$ )
Small + large motion	0.0979	0.5550	0.1414
Small motion only	<b>0.0437</b>	<b>0.3189</b>	<b>0.8323</b>

due to the large signal-to-noise ratio. This perfectness supports our focus on the small motion regime for enhancing a practical system. We also note that being a detection-tracking method, CenterPoint yields more precise flow estimation, as its accuracy largely depends on 3D box localization instead of motion.

**Benefit of using small-motion-specific dataset.** Recall that we have filtered out the large-motion data during the dataset curation, *i.e.* they are not in training data. Our experience is that excluding large motions in training data improves the model performance on the small motion regime. We report the accuracy in Tab. 4 to quantify this effect. The benefit may be explained by the unique swimming artifact, which mandates a small-motion-specific dataset.

#### 4.4. Evaluation in terms of Latency

With a focus on instantaneous detection, a time-sensitive task, it is helpful to also evaluate with a time-related metric. Our original task is to detect objects moving more than  $f_{\text{thre}}=0.05\text{m}$  in a 0.5s latency. Here, we increase  $f_{\text{thre}}$  to target at larger motion, which effectively allows proportionally increased latency if assuming constant velocity, hence decreasing the requirement on the latency. We report in Tab. 5 the detection accuracy (F1) across different latencies, indicating the consistently superior performance from S'More.

#### 4.5. Important Design Choices

**Grid Size.** We study the impact of occupancy grid size and find it important in our design. We compare the performance of two grid sizes: a balanced  $100 \times 100 \times 100$  grid and an alternative  $500 \times 500 \times 4$  grid, where the latter significantly reduces the resolution along  $z$ -axis. The results in Fig. 9(a), reveal a notable performance degradation (see the two dashed lines are consistently lower than the solid lines) when the  $z$ -axis resolution is reduced, despite the increased axial resolution to  $500 \times 500$ . This may also result in detection ambiguity which stems from the model's reduced capacity to discern subtle vertical variations, and lead to less reliable vehicle localization and motion detection.

Table 5. Evaluation of detection (F1 score) in terms of latency.

Latency	0.5s	0.7s	0.9s	1.1s	1.3s
ICP [5]	0.7758	0.8129	0.8229	0.8308	0.8346
CenterPoint [54]	0.7270	0.7658	0.7800	0.7899	0.7935
S'More	<b>0.8323</b>	<b>0.9003</b>	<b>0.9224</b>	<b>0.9568</b>	<b>0.9582</b>

**Point-/Object-level Instance Flow Estimation.** We study the behavior of the point-level and object-level training strategies for motion flow estimation. The point-level approach predicts a separate flow for each point, whereas the object-level strategy regresses a single rigid transformation for the entire object to calculate motion flow. The results detailed in Fig. 9(b) in terms of EPE and Angle Error across various flow thresholds, reveal that point-level instance flow estimation (indicated by two solid lines) consistently outperforms the object-level approach (indicated by two dashed lines). Notably, point-level estimation maintains stable performance even at very low thresholds. In contrast, object-level flow prediction exhibits significant fluctuations in performance across different thresholds.

**Loss Components.** We ablate each loss component and report the results under various flow thresholds in Tab. 2. Notably, we observe that as the flow threshold decreases, there is a corresponding degradation in model performance. This trend aligns with our expectation, as lower thresholds are designed to detect subtler motions. Subtler motions often come with more severe swimming effects and thus lead to less accurate predictions. We also observe that each loss component plays a critical role in tuning the model for these subtleties, making them essential for maintaining performance across varying motion dynamics.

## 5. Conclusion

This paper defines the problem of perceiving subtle motion for vehicles, presenting practical significance. To mitigate swimming artifacts causing ambiguity in subtle motion perception, we leverage occupancy completion as an effective strategy to facilitate motion learning. Despite the overall good performance, our method faces challenges under extremely sparse or high-occluded objects. Also, we currently only handle vehicles but not pedestrians or cyclists. We hope our work and its limitations can inspire more research into this important yet under-explored problem.

## Acknowledgement

This research project has been partially funded by research grants to Dimitris N. Metaxas through NSF: 2310966, 2235405, 2212301, 2003874, and FA9550-23-1-0417.

## References

- [1] Ben Agro, Quinlan Sykora, Sergio Casas, and Raquel Urtasun. Implicit occupancy flow fields for perception and prediction in self-driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1379–1388, 2023. 3
- [2] John Amanatides, Andrew Woo, et al. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, pages 3–10. Citeseer, 1987. 4
- [3] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2
- [4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 2
- [5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 5, 8
- [6] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3
- [7] Qi Chang, Zhennan Yan, Mu Zhou, Di Liu, Khalid Sawalha, Meng Ye, Qilong Zhangli, Mikael Kanski, Subhi Al’Aref, Leon Axel, et al. Deeprecon: Joint 2d cardiac segmentation and 3d volume reconstruction via a structure-specific generative method. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 567–577. Springer, 2022. 2
- [8] Nathaniel Chodosh, Deva Ramanan, and Simon Lucey. Re-evaluating lidar scene flow for autonomous driving. *arXiv preprint arXiv:2304.02150*, 2023. 2, 3, 5
- [9] Ilya Chugunov, Yuxuan Zhang, Zhihao Xia, Xuaner Zhang, Jiawen Chen, and Felix Heide. The implicit values of a good hand shake: Handheld multi-frame neural depth refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2852–2862, 2022. 3
- [10] Fangqiang Ding, Andras Palffy, Dariu M Gavrilă, and Chris Xiaoxuan Lu. Hidden gems: 4d radar scene flow learning using cross-modal supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2
- [11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2015. 4
- [12] Brandon Y Feng, Hadi Alzayer, Michael Rubinstein, William T Freeman, and Jia-Bin Huang. 3d motion magnification: Visualizing subtle motions from time-varying radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9837–9846, 2023. 3
- [13] Artem Filatov, Andrey Rykov, and Viacheslav Murashkin. Any motion detector: Learning class-agnostic scene dynamics from a sequence of lidar point clouds. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 9498–9504. IEEE, 2020. 2
- [14] Yunhe Gao, Mu Zhou, Di Liu, Zhennan Yan, Shaoting Zhang, and Dimitris N Metaxas. A data-scalable transformer for medical image segmentation: architecture, model efficiency, and benchmark. *arXiv preprint arXiv:2203.00131*, 2022. 2
- [15] Yunhe Gao, Zhuowei Li, Di Liu, Mu Zhou, Shaoting Zhang, and Dimitris N Meta. Training like a medical resident: universal medical image segmentation via context prior learning. *arXiv preprint arXiv:2306.02416*, 2023. 2
- [16] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5413–5421, 2016. 3
- [17] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. Proxedit: Improving tuning-free real image editing with proximal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4291–4301, 2024. 2
- [18] Xiaoxiao He, Chaowei Tan, Bo Liu, Liping Si, Weiwu Yao, Liang Zhao, Di Liu, Qilong Zhangli, Qi Chang, Kang Li, et al. Dealing with heterogeneous 3d mr knee images: A federated few-shot learning method with dual knowledge distillation. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023. 2
- [19] Kuan-Chih Huang, Ming-Hsuan Yang, and Yi-Hsuan Tsai. Delving into motion-aware matching for monocular 3d object tracking. In *ICCV*, 2023. 2
- [20] Shengyu Huang, Zan Gojic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3d scene analysis by point cloud accumulation. In *European Conference on Computer Vision*, 2022. 2, 3, 5
- [21] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 3, 4
- [22] Kuan-Hui Lee, Matthew Kliemann, Adrien Gaidon, Jie Li, Chao Fang, Sudeep Pillai, and Wolfgang Burgard. Pillarflow: End-to-end birds-eye-view flow estimation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2007–2013. IEEE, 2020. 2
- [23] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34:7838–7851, 2021. 2
- [24] Xueqian Li, Jianqiao Zheng, Francesco Ferroni, Jhony Kaesemodel Pontes, and Simon Lucey. Fast neural scene flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 5
- [25] Zhuowei Li, Long Zhao, Zizhao Zhang, Han Zhang, Di Liu, Ting Liu, and Dimitris N Metaxas. Steering prototypes with

- prompt-tuning for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2523–2533, 2024. 2
- [26] Ce Liu, Antonio Torralba, William T Freeman, Frédo Durand, and Edward H Adelson. Motion magnification. *ACM transactions on graphics (TOG)*, 24(3):519–526, 2005. 3
- [27] Di Liu, Jiang Liu, Yihao Liu, Ran Tao, Jerry L Prince, and Aaron Carass. Label super resolution for 3d magnetic resonance images using deformable u-net. In *Medical Imaging 2021: Image Processing*, pages 606–611. SPIE, 2021. 2
- [28] Di Liu, Zhennan Yan, Qi Chang, Leon Axel, and Dimitris N Metaxas. Refined deep layer aggregation for multi-disease, multi-view & multi-center cardiac mr segmentation. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 315–322. Springer, 2021. 2
- [29] Di Liu, Yunhe Gao, Qilong Zhangli, Ligong Han, Xiaoxiao He, Zhaoyang Xia, Song Wen, Qi Chang, Zhennan Yan, Mu Zhou, et al. Transfusion: multi-view divergent fusion for medical image segmentation with transformers. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 485–495. Springer, 2022. 2
- [30] Di Liu, Xiang Yu, Meng Ye, Qilong Zhangli, Zhuowei Li, Zhixing Zhang, and Dimitris N Metaxas. Deformer: Integrating transformers with deformable models for 3d shape abstraction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14236–14246, 2023. 3
- [31] Di Liu, Long Zhao, Qilong Zhangli, Yunhe Gao, Ting Liu, and Dimitris N Metaxas. Deep deformable models: Learning 3d shape abstractions with part consistency. *arXiv preprint arXiv:2309.01035*, 2023.
- [32] Di Liu, Qilong Zhangli, Yunhe Gao, and Dimitris Metaxas. Leopard: Learning explicit part discovery for 3d articulated shape reconstruction. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [33] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019. 2
- [34] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2, 4
- [35] Yi Ma, Jana Košecká, and Shankar Sastry. Linear differential algorithm for motion recovery: A geometric approach. *International Journal of Computer Vision*, 36:71–89, 2000. 3
- [36] Reza Mahjourian, Jinkyu Kim, Yuning Chai, Mingxing Tan, Ben Sapp, and Dragomir Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):5639–5646, 2022. 3
- [37] Carlos Martín-Isla, Víctor M Campello, Cristian Izquierdo, Kaisar Kushibar, Carla Sendra-Balcells, Polyxeni Gkontra, Alireza Sojoudi, Mitchell J Fulton, Tewodros Weldebirhan Arega, Kumaradevan Punithakumar, et al. Deep learning segmentation of the right ventricle in cardiac mri: The m&ms challenge. *IEEE Journal of Biomedical and Health Informatics*, 2023. 2
- [38] Benedikt Mersch, Xieyuanli Chen, Ignacio Vizzo, Lucas Nunes, Jens Behley, and Cyrill Stachniss. Receding moving object segmentation in 3d lidar data using sparse 4d convolutions. *IEEE Robotics and Automation Letters*, 7(3):7503–7510, 2022. 2
- [39] Benedikt Mersch, Tiziano Guadagnino, Xieyuanli Chen, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Building volumetric beliefs for dynamic environments exploiting map-based moving object segmentation. *IEEE Robotics and Automation Letters*, 2023. 2
- [40] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localization. In *CVPR*, 2023. 2
- [41] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11177–11185, 2020. 2
- [42] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 4
- [43] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, 2001. 5
- [44] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, 2009. 5
- [45] Jiadai Sun, Yuchao Dai, Xianjing Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11456–11463. IEEE, 2022. 2
- [46] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 1, 2, 5
- [47] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023. 3
- [48] Kavisha Vidanapathirana, Shin-Fang Chng, Xueqian Li, and Simon Lucey. Multi-body neural scene flow. 2023. 2
- [49] Jun Wang, Xiaolong Li, Alan Sullivan, Lynn Abbott, and Siheng Chen. Pointmotionnet: Point-wise motion learning for large-scale lidar point clouds sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4419–4428, 2022. 2
- [50] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for

- surrounding semantic occupancy perception. *arXiv preprint arXiv:2303.03991*, 2023. 3
- [51] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21729–21740, 2023. 3
- [52] Song Wen, Hao Wang, Di Liu, Qilong Zhangli, and Dimitris Metaxas. Second-order graph odes for multi-agent trajectory forecasting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5101–5110, 2024. 2
- [53] Huajie Wu, Yihang Li, Wei Xu, Fanze Kong, and Fu Zhang. Moving event detection from lidar point streams. *Nature Communications*, 2024. 2
- [54] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 2, 4, 5, 8
- [55] Fisher Yu and David Gallup. 3d reconstruction from accidental motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3986–3993, 2014. 3
- [56] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019. 4
- [57] Gang Zhang, Xiaoyan Li, and Zhenhua Wang. Sequential multi-view fusion network for fast lidar point motion estimation. In *European Conference on Computer Vision*, 2022. 2
- [58] Qilong Zhangli, Jingru Yi, Di Liu, Xiaoxiao He, Zhaoyang Xia, Qi Chang, Ligong Han, Yunhe Gao, Song Wen, Haiming Tang, et al. Region proposal rectification towards robust instance segmentation of biological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 129–139. Springer, 2022. 2
- [59] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 5
- [60] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, 2017. 3