**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

# SSF-PAN: Semantic Scene Flow-Based Perception for Autonomous Navigation in Traffic Scenarios

Yinqi Chen[1], Meiying Zhang[1,*], Qi Hao[1,*], Guang Zhou[2]

***Abstract*—Vehicle detection and localization in complex traffic scenarios pose significant challenges due to the interference of moving objects. Traditional methods often rely on outlier exclusions or semantic segmentations, which suffer from low computational efficiency and accuracy. The proposed SSF-PAN can achieve the functionalities of LiDAR point cloud based object detection/localization and SLAM (Simultaneous Localization and Mapping) with high computational efficiency and accuracy, enabling map-free navigation frameworks. The novelty of this work is threefold: 1) developing a neural network which can achieve segmentation among static and dynamic objects within the scene flows with different motion features, that is, semantic scene flow (SSF); 2) developing an iterative framework which can further optimize the quality of input scene flows and output segmentation results; 3) developing a scene flow-based navigation platform which can test the performance of the SSF perception system in the simulation environment. The proposed SSF-PAN method is validated using the SUScape-CARLA [3] and the KITTI [1] datasets, as well as on the CARLA simulator. Experimental results demonstrate that the proposed approach outperforms traditional methods in terms of scene flow computation accuracy, moving object detection accuracy, computational efficiency, and autonomous navigation effectiveness.**

***Index Terms*—Semantic Scene Flow, SLAM, Moving Object Detection, Autonomous Navigation.**

## I. Introduction

As intelligent transportation systems and autonomous driving continue to evolve, there is a growing demand for perception systems that can understand and operate in complex traffic environments. Traditional navigation frameworks rely on Simultaneous Localization and Mapping (SLAM) [2], which often suffer from high computational cost. They also require separate modules for detecting moving objects to avoid collisions. Recently, scene flow methods have been explored for map-free perception, as they estimate point-wise motion between consecutive point cloud frames. This information can support ego-vehicle localization [3] and instance segmentation [4]. Although scene flow-based perception has shown promise, several challenges remain in real-world traffic scenarios:

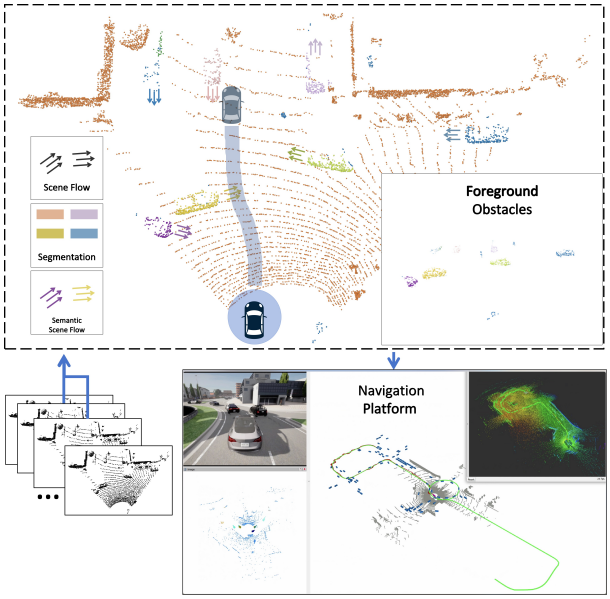



Fig. 1: An illustration of SSF estimation for autonomous navigation with dynamic and static object classification as well as moving object instance segmentation.

- **High-Quality Semantic Segmentation**: Many methods exploit semantic information to identify static environments and dynamic objects, but their effectiveness is limited by the quality of object feature extraction and classification [5], [6]. Usually, these methods do not fully utilize motion information, especially for complex dynamic traffic scenes where occlusions frequent happen.
- **Accurate Scene Flow Estimation**: Scene flow methods are often divided into (1) sparse point cloud approaches [7]–[9] which can run fast and are suitable for real-time use, but their accuracy drops with frame-to-frame inconsistency, occlusion, and low point density, and (2) dense point cloud approaches [10]–[13] which require heavy computation for segmentation and still do not use semantic information.
- **Comprehensive Navigation Testing Platform**: Existing systems often do not fully address the specific requirements for scene flow and navigation performance in dynamic environments. SSF-PAN provides a scene flow-based navigation platform testable in the CARLA simulator, ensuring the perception system's accuracy and dependability in intricate traffic scenarios. Scene flow based navigation systems are advantageous in no need of the high-definition maps for ego-vehicle localization. A comprehensive navigation testing platform requires the

Manuscript received: March 30, 2025; Revised August 10, 2025; Accepted September 26, 2025.

This paper was recommended for publication by Editor Ashis Banerjee upon evaluation of the Associate Editor and Reviewers' comments. This work is jointly supported by the National Natural Science Foundation of China (62261160654), the Shenzhen Fundamental Research Program (JCYJ20220818103006012), and the Shenzhen Major Project for Science and Technology (KJZD20231023092600001).

[1]Yinqi Chen, Meiying Zhang, Qi Hao are with the Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China.

[2]Guang Zhou is with Operation Department, Shenzhen Deeproute.ai Co.,Ltd, Shenzhen, China.

[*]Corresponding author: Meiying Zhang (email: zhangmy@sustech.edu.cn) and Qi Hao (email: hao.q@sustech.edu.cn)

Digital Object Identifier (DOI): see top of this page.

[3]https://suscape.net/datasets/sceneflow

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

development of a set of scene flow based perception modules for ego-vehicle localization and moving object detection, as well as prediction, decision-making, and motion planning modules [14].

The proposed SSF-PAN system addresses these issues by utilizing the semantic information of scene flows, enhancing performance in pose estimation, odometry, and perception robustness in complex traffic scenarios. By recursively optimizing the quality of semantic segmentation and the accuracy of scene flow estimation, SSF-PAN can achieve higher quality of self-localization and moving object detection, as illustrated in Fig. 1, leading to more reliable navigation systems. The main contributions of this work include:

- **Advanced Scene Flow-Based Point Cloud Segmentation**: We develop a novel neural network which utilizes scene flow information to effectively segment point clouds, distinguishing between static environment and dynamic objects with various motion features. This enhances the understanding and differentiation of object types and their movements in complex traffic scenarios.
- **Refined Iterative Optimization Framework**: We develop an iterative framework that continuously refines both the input scene flow data and the segmentation outputs. This iterative process improves the accuracy and robustness of the system and generates higher quality data throughout the perception pipeline.
- **Robust Map-Free Navigation Platform**: We develop a comprehensive navigation testing platform based on scene flow data, capable of evaluating the performance of the SSF perception system within various simulated environments. This platform supports robust autonomous navigation without the need of pre-built high definition maps and adapts well to many traffic conditions.

## II. RELATED WORK

### A. Scene Flow based Motion Segmentation

Estimating scene flow helps improve semantic segmentation. Recent learning-based methods use scene flow for point-wise segmentation to handle data sparsity and representation challenges. However, many current methods [15], [16] rely on ground truth or focus only on simple foreground-background segmentation. For example, Thomas et al. [17] used multiple recordings to classify static, slow-moving, and fast-moving objects but had difficulty segmenting dynamic objects accurately. The OGC method [4] offers self-supervised segmentation but struggles to detect background points well. Our SSF-PAN framework overcomes these issues by using self-supervised learning with scene flow for motion segmentation, accurately separating background and dynamic objects. This leads to better motion information extraction in traffic scenes.

### B. Scene Flow Optimization

Many methods improve scene flow estimation by using extra information. Some [18], [19] use rigidity constraints on RGB-D frames and piecewise rigid planar models. Dewan et al. [20] applied local geometric consistency on factor graphs with 3D descriptors, GraphFlow [21] depends on sparse keypoints. Data-driven methods [22] combine depth and flow but require instance segmentation. PointFlowNet [23] integrates scene flow with PointNet++ [24]. Some self-supervised approaches [25], [26] require pre-training or extra processing. Recent dense point cloud methods, such as ZeroFlow [11], DeFlow [12], and SeFlow [13], optimize estimation by either distilling from optimization-based teachers for scalable inference, refining voxel features back to point-level details and handling static/dynamic imbalance, or incorporating dynamic/static classification with cluster consistency constraints. However, these methods increase segmentation and computation overhead and generally do not use instance segmentation. Our SSF-PAN uses sparse point clouds with self-supervised refinement, avoiding extra labels or dense inputs, and achieves high accuracy and efficiency in complex traffic scenes.

### C. Map-Free Navigation in Dynamic Environments

Recent works combine scene flow estimation with semantic segmentation to improve navigation and decision-making [11], [27]. However, these methods often require large datasets and high computational cost. To avoid reliance on HD maps, several map-free approaches have been proposed. Image-based methods [28], [29] often fail under varying lighting, dynamic objects, or changes in infrastructure. LiDAR-based methods [30] are more robust but usually depend on local structures. End-to-end learning approaches [31], [32] do not need maps, but they require large amounts of data and computation [33]. So far, no map-free navigation system has used scene flow for real-time applications. Our navigation testing platform enables real-time adaptation to dynamic traffic, improving flexibility and performance. The closed-loop system in SSF-PAN supports stable and scalable navigation, setting a new standard for map-free systems.

## III. SYSTEM SETUP AND PROBLEM STATEMENT

### A. System Setup

This system consists of two key components: a neural network for semantic scene flow estimation and a navigation platform based on the estimated scene flow, as shown in Fig. 2. The input to the system is a pair of sequential point cloud frames, $P_t$ and $P_{t+1}$. The first frame $P_t$, along with a coarse semantic mask, is passed into the *active scene flow* (ASF) network [8] to predict the scene flow. This predicted flow is then combined with $P_t$ and fed into the segmentation network to produce semantic scene flow. This process is repeated iteratively using a mutual refinement framework, where the updated semantic mask helps refine both the flow and segmentation until results converge. This iterative approach helps distinguish between static scenes and dynamic objects more accurately. Once the scene flow and motion segmentation stabilize, dynamic objects are treated as obstacles in the navigation system. The estimated scene flow helps determine each obstacle's velocity. Using this information and a planned path, a planner (e.g., RDA planner [34]) performs real-time obstacle avoidance. Meanwhile, static points and their scene flow can be used to compute odometry or support SLAM.

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

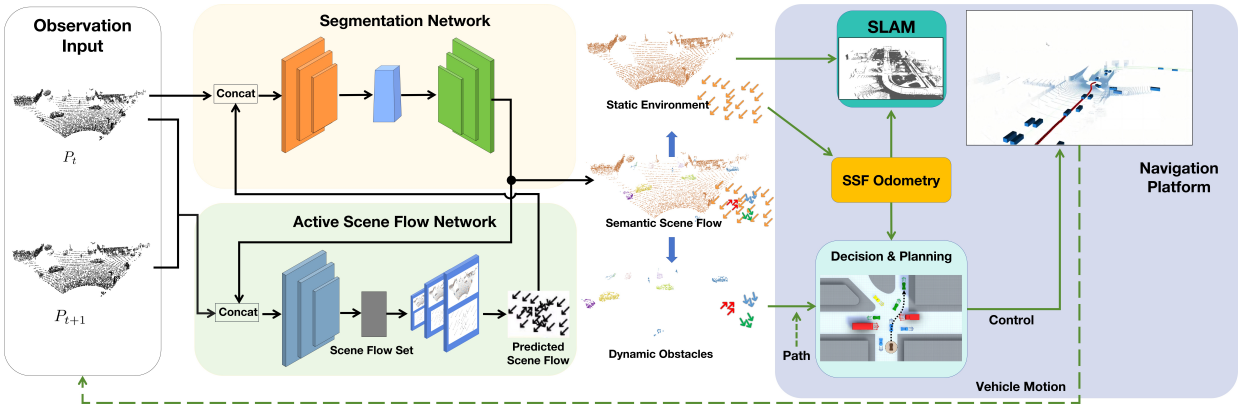



Fig. 2: An illustration of the SSF-PAN system diagram, which consists of two main components: scene flow estimation and motion segmentation. The system iteratively refines scene flow and segmentation results, aiding in dynamic obstacle recognition for navigation and SLAM.

### B. Problem Statement

Therefore, we focus on solving the following problems:

1) How to design effective constraints and representations for accurate segmentation of point clouds into static and dynamic objects, leveraging scene flow for enhanced motion segmentation?
2) How to iteratively refine scene flows and segmentation outputs to mutually improve their accuracy, enhancing overall scene flow estimation and motion segmentation?
3) How to integrate SSF into navigation modules for improved pose estimation and obstacle avoidance, enhancing autonomous navigation in traffic scenarios?

## IV. Proposed Methods

### A. Scene Flow-based Network for Motion Segmentation

As shown in Fig. 3, this method starts with two consecutive point cloud frames and an initial motion mask. The semantic scene flow network is mainly based on modified ASF [8] and OGC [4] networks. ASF processes the point clouds by using spatio-temporal feature consistency to estimate the initial scene flow, $\overrightarrow{S_t}$. This flow is then used as an extra input for the segmentation network. Given the point cloud $P_t$ and scene flow $\overrightarrow{S_t}$, the network performs motion segmentation to generate a new mask $M_t$, separating dynamic objects from the static environment. Our motion segmentation minimizes a total loss function $\mathcal{L}^*$, which combines motion loss $\mathcal{L}_{mot}$, semantic scene flow consistency loss $\mathcal{L}_{SC}$, and per-cluster rigidity loss based on Chamfer distance (CD), $\mathcal{L}_{CD}$.

$$\mathcal{L}^* = \min_{\mathcal{L}} \mathcal{L}_{mot} + \mathcal{L}_{SC} + \mathcal{L}_{CD} \tag{1}$$

Here, we introduce the following self-supervised components to satisfy the geometry consistency between the $t^{th}$ frame $P_t$ and its predicted next frame $\widetilde{P}_{t+1}$, where $\widetilde{P}_{t+1} = P_t + \overrightarrow{S_t}$.

**Rigid Motion Consistency Loss $\mathcal{L}_{mot}$:** This loss ensures the consistency of rigid body motion between two frames. It is computed by estimating the transformation matrix $T_k \in \mathbb{R}^{4\times4}$ ($k = [1, \cdots, K]$) for each object using the weighted-Kabsch algorithm [35]. The loss minimizes the discrepancy between the transformed points and the estimated scene flow, thereby promoting accurate dynamic motion segmentation.

$$\mathcal{L}_{mot} = \frac{1}{K}\sum_{k=1}^{K}\left\| T_k \cdot (P_t \odot M_t^k) - (\widetilde{P}_{t+1} \odot M_t^k) \right\|_2 \tag{2}$$

where $\odot$ is element-wise product.

**Semantic Scene Flow Consistency Loss $\mathcal{L}_{SC}$:** Enforces scene flow consistency within objects, penalizing discrepancies for uniform motion:

$$\mathcal{L}_{SC} = \frac{1}{K \cdot N_k}\sum_{k=1}^{K}\sum_{i=1}^{N_k}\left( \overrightarrow{S_{t,i}^k} - \frac{1}{N_k}\sum_{j=1}^{N_k}\overrightarrow{S_{t,j}^k} \right)^2 \tag{3}$$

where $N_k = \sum_{i=1}^{N}\delta(M_t, i)$, $\delta(M_t, i) = 1$ when $M_{t,i} = k$ and $\delta(M_t, i) = 0$ otherwise, $\overrightarrow{S_{t,i}^k}$ represents the $i^{\text{th}}$ element of $\overrightarrow{S_t^k}$ and $\overrightarrow{S_t^k} = \overrightarrow{S_t} \odot M_t^k$.

**Rigidity Chamfer Distance Loss $\mathcal{L}_{CD}$:** Ensures spatial consistency by minimizing the Chamfer distance between predicted and scanned points at frame $(t+1)$:

$$\mathcal{L}_{CD} = \sum_{x\in P_{t+1}}\min_{y\in\widetilde{P}_{t+1}}||x-y||_2 + \sum_{y\in\widetilde{P}_{t+1}}\min_{x\in P_{t+1}}||x-y||_2 \tag{4}$$

### B. Iterative Optimization for Scene Flow and Segmentation

The ASF network [8] takes the point cloud $P_t$ and an initialized segmentation mask $M_t^{(0)}$ as inputs. This setup allows the network to iteratively refine the scene flow estimation and dynamic & static segmentation. During each iteration, the latest segmentation mask $M_t$ is concatenated with the point cloud $P_t$ and passed through the network. The process involves *Set Abstraction* (SA) feature extraction, *Scene Flow Set*, and *Up Sampling* (US) modules, which compute a coarse scene flow, $\overrightarrow{S_t}^{(0)}$. This scene flow is utilized for initial motion segmentation, resulting in an updated mask, $M_t$. The iterative refinement process continues until the change $\Delta_{total}$ in the scene flow estimation and motion segmentation converges to a threshold $\epsilon$, initialized to $10^{-3}$,

$$\Delta_{total} = \alpha \cdot \left\| \overrightarrow{S_t}^{(i)} - \overrightarrow{S_t}^{(i-1)} \right\|_2 + \beta \cdot \left\| M_t^{(i)} - M_t^{(i-1)} \right\|_2 \tag{5}$$

where $\alpha$ and $\beta$ are weights that balance the contributions of the scene flow and mask convergence. The system classifies points

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

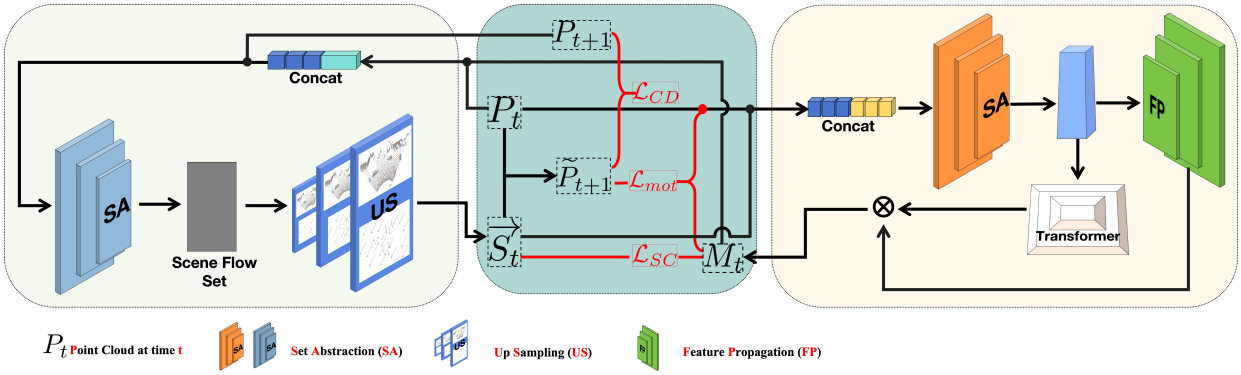



Fig. 3: Illustration of the SSF module, showing a single iteration of joint scene flow estimation and motion segmentation guided by self-supervised loss functions. The module processes two consecutive LiDAR point clouds, estimating scene flow and segmenting moving objects based on flow consistency. The self-supervised losses help refine the predictions within this iterative process, enabling dynamic object detection without manual labels.

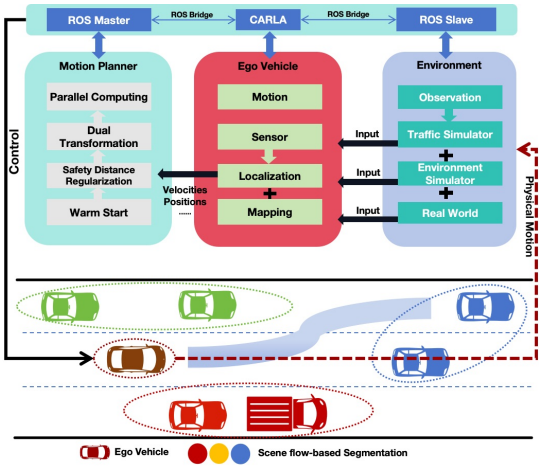



Fig. 4: An illustration of the SSF-based navigation platform developed in the CARLA. This platform integrates our scene flow-based perception system with navigation and control modules, enabling comprehensive testing and evaluation of dynamic scene understanding and SLAM performance in complex, simulated urban scenarios.

as static environment with mask $M_t^{(s)}$ and $(k-1)$ dynamic obstacles with mask $M_t^{(d)}$, that is, $M_t = \{M_t^{(s)}, M_t^{(d)}\}$ : Based on the assumption that dynamic objects are fewer than static background elements in traffic scenarios, it is simple to determine the static set $M_t^{(s)}$ by statistical quantity for each cluster. The maximum cluster size is considered as the static part:

$$M_t^{(s)} = \{M_{t,i} \mid M_{t,i} = \arg\max_k N_k\} \tag{6}$$

### C. Scene Flow-Based Navigation Platform

As depicted in Fig. 4, we setup a SSF-based navigation system testing platform in CARLA [36] that is a high-fidelity simulator built on the Unreal Engine. Our platform consists of the motion planner module, the ego-vehicle module and the environment module. Communication between the two modules is achieved via ROS. The motion planner algorithm integrates scene flow information to enhance obstacle avoidance and path prediction. Specifically, the environment module handles the physical motion data, and then ego-vehicle module can get the observation. After the scene flow of each object is extracted,

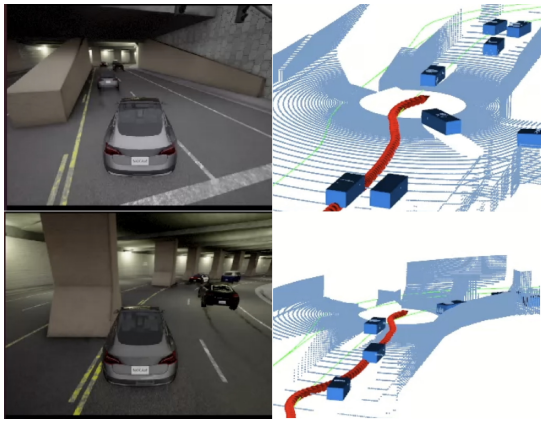


Fig. 5: A real-time snapshot of the SSF navigation system running in the CARLA simulator. The system leverages scene flow estimation to perceive dynamic objects and navigate safely, demonstrating the practical application and effectiveness of our proposed perception framework in a realistic simulated environment.

each obstacle's velocity $V_t^k$ is determined by Eq. (7):

$$V_t^k = \frac{1}{N_k \cdot \Delta t} \sum_{i=1}^{N_k} \left\| \overrightarrow{S_{t,i}^k} \right\| \tag{7}$$

The velocity information, combined with the dynamic object positions computed by $P_t^k$, is fed into the motion planner to control the ego-vehicle. This enables adaptive path adjustments for effective obstacle avoidance and accurate trajectory prediction in dynamic environments. Scene flow-based navigation operates without a map, as localization is directly inferred from scene flow data. As shown in Fig. 5, the goal is to navigate the vehicle from waypoint A to B. Following the planned route, the vehicle encounters and overtakes a slower-moving obstacle due to its higher speed.

## V. Experiments

### A. Validation of SSF-SLAM on Public Datasets

*1) Datasets:* The dataset used for training the ASF [8] and validating the SLAM framework combines SUScape-CARLA[1] (14,472 frames) and KITTI [1] (2,690 frames), totaling 16,352 training and 810 validation frames, each with 8,192 points. To evaluate performance under varying dynamic conditions, the dataset is divided into two subsets based on the number

[1]https://suscape.net/datasets/sceneflow

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

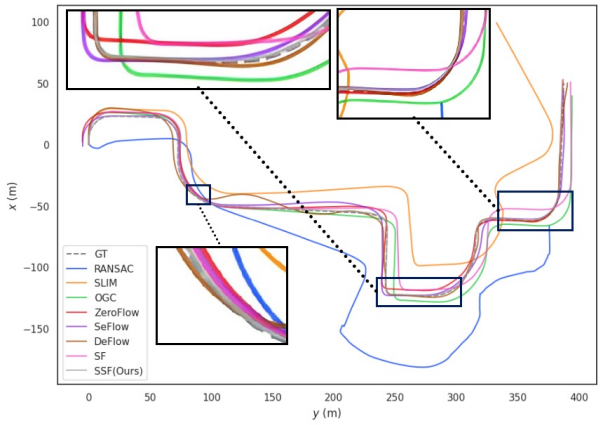
TABLE I: Results of Translational Relative Pose Error (RPE) for SLAM experiments on the $D^H$ and $D^T$ datasets

| Framework | Method | xyz[a] | | | | | | rpy | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $D^H$ | | | $D^T$ | | | $D^H$ | | | $D^T$ | | |
| | | RMSE[b] | SSE | STD | RMSE | SSE | STD | RMSE | SSE | STD | RMSE | SSE | STD |
| A_LOAM | RANSAC [37] | 4.111 | 1352.264 | 2.183 | 5.356 | 1377.533 | 2.225 | 0.244 | 4.562 | 0.192 | 0.385 | 5.127 | 0.202 |
| | SLIM [15]● | 0.585 | 122.794 | 0.431 | 0.586 | 123.356 | 0.432 | 0.106 | 4.040 | 0.090 | 0.110 | 4.307 | 0.086 |
| | OGC [4]⊘ | 0.448 | 72.102 | 0.343 | 0.447 | 71.844 | 0.340 | 0.565 | 1.1469 | 0.050 | 0.605 | 1.1470 | 0.053 |
| | ZeroFlow [11]○ | 0.167 | 9.994 | 0.067 | 0.467 | 78.389 | 0.321 | 0.102 | 0.830 | 0.035 | 0.057 | 1.147 | 0.050 |
| | SeFlow [13]○ | 0.239 | 20.444 | 0.092 | 0.503 | 90.947 | 0.320 | 0.776 | 0.722 | 0.033 | 0.056 | 1.145 | 0.050 |
| | DeFlow [12]○ | 0.385 | 53.315 | 0.186 | 0.569 | 116.210 | 0.310 | 0.134 | 0.645 | 0.051 | 0.056 | 1.141 | 0.050 |
| | SF | 0.455 | 16.553 | 0.378 | 0.544 | 160.653 | 0.547 | 0.063 | 2.417 | 0.084 | 0.104 | 3.154 | 0.096 |
| | SSF(Ours)● | **0.102** | **0.447** | **0.053** | **0.099** | **0.432** | **0.054** | **0.034** | **0.535** | **0.031** | **0.019** | **0.028** | **0.015** |
| S_LOAM | RANSAC [37] | 1.110 | 442.427 | 0.743 | 1.121 | 450.674 | 0.844 | 0.230 | 4.228 | 0.168 | 0.232 | 4.237 | 0.175 |
| | SLIM [15]● | 0.481 | 104.100 | 0.333 | 0.599 | 127.845 | 0.412 | 0.051 | 1.153 | 0.045 | 0.084 | 1.364 | 0.090 |
| | OGC [4]⊘ | 0.405 | 73.832 | 0.343 | 0.271 | 74.874 | 0.359 | 0.034 | 0.535 | 0.031 | 0.059 | 0.637 | 0.373 |
| | ZeroFlow [11]○ | 0.092 | 3.875 | 0.302 | 0.416 | 77.852 | 0.257 | 0.103 | 0.048 | 0.040 | 0.035 | 0.536 | 0.031 |
| | SeFlow [13]○ | 0.123 | 6.831 | 0.048 | 0.428 | 82.650 | 0.253 | 0.098 | 0.043 | 0.035 | 0.047 | 0.546 | 0.021 |
| | DeFlow [12]○ | 0.208 | 19.458 | 0.079 | 0.450 | 91.025 | 0.245 | 0.143 | 0.092 | 0.056 | 0.034 | 0.534 | 0.031 |
| | SF | 0.322 | 8.278 | 0.261 | 0.488 | 85.555 | 0.351 | 0.073 | 2.427 | 0.064 | 0.083 | 3.135 | 0.104 |
| | SSF(Ours)● | **0.057** | **0.380** | **0.043** | **0.069** | **0.376** | **0.047** | **0.014** | **0.016** | **0.011** | **0.020** | **0.032** | **0.017** |

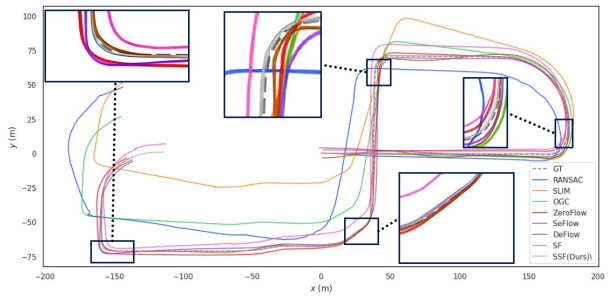
[a] 'xyz' denotes the translational error, indicating the error in camera position along the three axes (x, y, z). 'rpy' represents the rotational error, indicating the error in camera orientation, including the rotations around the camera's fixed coordinate system's x-axis (roll), y-axis (pitch), and z-axis (yaw).

[b] RMSE (Root Mean Square Error) represents the average deviation; SSE (Sum of squares due to error) represents the sum of squared errors between predicted and ground truth values for each sample point, measured in meters; STD (Standard Deviation) represents the dispersion of the sample data, indicating the distribution of sample data around the mean.

[c] ●: uses both segmentation and scene flow predicted by the algorithm; ⊘: uses predicted segmentation but ground-truth scene flow; ○: uses predicted scene flow but ground-truth segmentation.




**(a)** A_LOAM odometry




**(b)** S_LOAM odometry

Fig. 6: Ego-vehicle trajectory estimation (map construction) results based on the point-cloud SLAM with different odometry modules.

of sampled foreground (moving object) points per frame: $D^H$ (Hundred-level) with 100 points per frame, and $D^T$ (Thousand-level) with 4,000 points per frame. The superscripts "H" and "T" indicate the approximate scale of dynamic points, facilitating evaluation across scenes of different dynamic complexity.

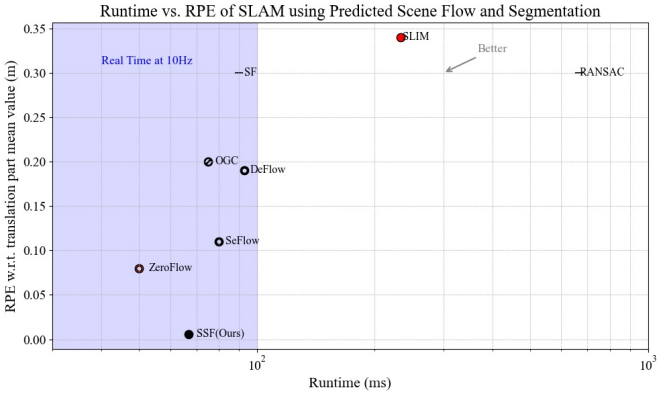



Fig. 7: Runtime vs. RPE of different scene flow-based SLAM methods. The shaded blue region represents real-time capability at 10Hz(<100ms). Our method (SSF) achieves the best accuracy while staying within the real-time regime.The meaning of the markers (●, ⊘, ○) follows the same convention as in Table I.

*2) Comparative Results:* We conducted a comprehensive evaluation of the SLAM system integrated with SSF estimation to validate its localization and mapping performance in dynamic traffic scenarios. Two SLAM frameworks were used: A-LOAM[2] and S-LOAM[3]. A-LOAM is a simplified version of LOAM [38] without IMU, while S-LOAM adds odometry loop closure optimization. Both use RANSAC [37] as the default point cloud registration method. We compared localization accuracy on the $D^H$ and $D^T$ datasets using different odometry modules, as shown in Table I. Here, *SF* denotes scene flow–based odometry without motion segmentation, and *SSF* refers to our segmentation-enhanced method. SSF reduces trajectory errors by over 93% compared to RANSAC. As shown in Fig. 6, our SSF-enhanced SLAM produces trajectories closely matching ground truth (GT), achieving localization accuracy within 10 cm. To evaluate computational

[2]https://github.com/HKUST-Aerial-Robotics/A-LOAM.git
[3]https://github.com/haocaichao/S-LOAM.git

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

TABLE II: Quantitative Results of SSF-PAN Applied in the Navigation Framework

| Moving Obstacles | Success Rate[a]↑ | | | | Navigation Time(s)↓ | | | | Moving Average Speed(m/s)↑ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GT | DBSCAN | PointRCNN | SSF-PAN | GT | DBSCAN | PointRCNN | SSF-PAN | GT | DBSCAN | PointRCNN | SSF-PAN |
| 50 | 0.96 | 0.88 | 0.91 | 0.92 | 308.42 | 466.28 | 453.74 | 430.63 | 11.11 | 8.89 | 10.23 | 10.28 |
| 100 | 0.88 | 0.74 | 0.83 | 0.86 | 353.68 | 536.78 | 438.06 | 418.18 | 9.72 | 8.06 | 8.99 | 9.17 |
| 200 | 0.80 | 0.63 | 0.68 | 0.76 | 630.73 | 690.36 | 679.46 | 677.73 | 9.44 | 7.22 | 7.98 | 8.06 |
| 500 | 0.73 | 0.50 | 0.57 | 0.73 | 722.56 | 777.65 | 750.98 | 741.47 | 9.17 | 5.94 | 6.03 | 6.94 |
| 800 | 0.70 | 0.44 | 0.52 | 0.68 | 739.54 | 839.24 | 820.35 | 804.58 | 7.22 | 5.83 | 5.97 | 6.39 |
| 1000 | 0.65 | 0.33 | 0.46 | 0.60 | 848.43 | 939.57 | 877.35 | 866.73 | 6.67 | 5.56 | 5.65 | 6.29 |

[a] **Success Rate** measures the percentage of successful trials where the vehicle reaches its destination without collisions, **Navigation Time** records the duration required to complete the route, and **Moving Average Speed** reflects the vehicle's average velocity throughout the navigation process.

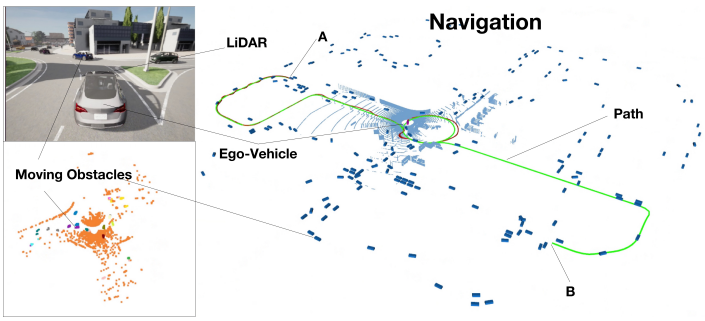



Fig. 8: An illustration of SSF-based autonomous navigation in complex traffic scenes in CARLA.

TABLE III: Clustering/Semantic Segmentation Accuracy on the $D^T$

| Segmentation Method/Accuracy (%) | Only Point Cloud[a] | Only Scene Flow | Point Cloud & Scene Flow(Ours) |
|---|---|---|---|
| OGC [4] | 89.79 | 20.54 | - |
| SLIM [15] | 72.43 | - | - |
| GMM [41] | 60.38 | 87.33 | 90.25 |
| DBSCAN [39] | 42.97/46.62 | OM[b]/ **85.01** | 42.96/46.72 |
| PointNet++ [24] | **87.33** | 30.52 | 85.61 |
| SSF(Ours) | - | - | **93.27** |

[a] **Only Point Cloud** and **Only Scene Flow** use spatial and motion features separately, while **Point Cloud & Scene Flow** represents their concatenation for enhanced segmentation.
[b] OM stands for Out of Memory.

efficiency and real-time feasibility, we compared the runtime and accuracy of various scene flow–based SLAM methods. As shown in Fig. 7, **SSF (Ours)** achieves the lowest RPE (0.006 m) with a runtime of only 67 ms. In contrast, methods like RANSAC [37] and SLIM [15] are slower (over 200 ms) and less accurate. Compared with ZeroFlow [11], SeFlow [13], and DeFlow, our method reduces localization error by approximately 38%–73% while maintaining comparable or better runtime performance, demonstrating the effectiveness of our iterative optimization.

### B. Testing Framework for Vehicle Navigation in CARLA

*1) Experiments Setting:* Navigating vehicles in traffic using only onboard sensors is challenging due to strict accuracy and latency requirements. We test SSF's real-time navigation in CARLA [36] using a 64-line 3D LiDAR for environment perception. As shown in Fig. 8, the ego vehicle uses differential steering to travel between two waypoints while avoiding randomly placed vehicles and pedestrians, which adjust speeds to prevent collisions. Obstacle detection relies on LiDAR, with ground truth orientation, position, and velocity provided. We compare SSF with RDA [34], an optimization-based MPC method for dynamic collision avoidance, and with DBSCAN [39] and PointRCNN [40], two common point cloud segmentation methods. All methods operate at 10 m/s over a 961 m path.

*2) Comparative Results:* The evaluation uses three metrics: success rate, navigation time, and moving average speed. A run is successful if the vehicle reaches its destination without collisions, with success rate computed over 100 trials. Navigation time is recorded in Rviz, and moving average speed is compared across methods. Table II shows results for 100 trials with obstacle counts from 50 to 1,000. SSF outperforms DBSCAN [39] and PointRCNN [40] in all metrics, with a larger margin when obstacle counts exceed 500 due to higher density. For fewer obstacles, SSF's advantage over PointRCNN is smaller, but unlike PointRCNN, SSF directly outputs point clouds for mapping, enabling more accurate speed and pose estimation. By combining scene flow computation, foreground–background segmentation, and dynamic object instance segmentation, SSF improves obstacle avoidance and system reliability.

### C. Ablation Studies

*1) Semantic Segmentation Evaluation:* Table III compares six methods for dynamic-static scene segmentation on $D^T$. Integrating scene flow with point clouds greatly improves accuracy. GMM [41] gains a 29.97% increase, and DBSCAN [39] improves to 46.72%, limited by point cloud distribution. Using scene flow alone, DBSCAN reaches 85.01%. PointNet++ [24] sees a slight drop from 87.33% to 85.61% with scene flow. SLIM [15], focused only on point clouds, scores 72.43%, showing limits despite multi-task design. Our SSF, based on a modified OGC [4], achieves the highest accuracy at 93.27%, showing strong dynamic scene detection.

*2) Scene Flow Evaluation:* This experiment compares two ASF-based strategies: **Implicit Strategy** preprocesses data by assigning different scene flow values to dynamic objects and background, focusing on static motion. **Explicit Strategy** combines semantic and geometric inputs to improve scene flow accuracy with richer context. Table IV shows the explicit strategy outperforms the baseline (point cloud only), improving EPE3D by 0.06 m, AS by 15.44%, and AR by 8.16%. The explicit strategy is used in the final scene flow prediction.

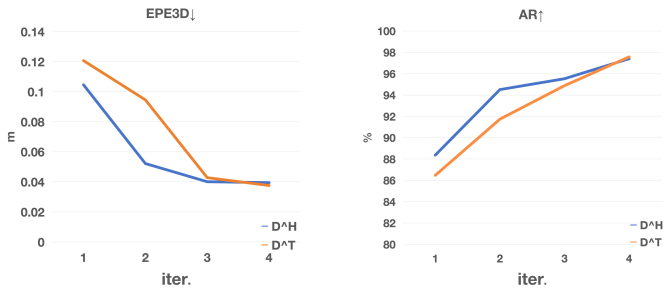
*3) Results of Iterative Framework for Scene Flow Estimation and Motion Segmentation :* Regarding the mutual promotion network, our ablation study compares scene flow accuracy using segmentation and point cloud inputs over iterations. The iterative ASF and OGC framework reduces error and improves accuracy, as shown in Fig. 9.

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

TABLE IV: Scene Flow Estimation Accuracy Averaged over 200 Scenes

| Dataset | Feature Extraction | EPE3D(m)[a]↓ | AS(%) ↑ | AR(%) ↑ | Outliers(%)↓ |
|---|---|---|---|---|---|
| $D^H$ | Only Point Cloud | 0.107400 | 76.8931 | 89.2483 | 69.6648 |
| | Implicit Strategy | 0.073223 | 78.4000 | 89.4864 | 65.7234 |
| | Explicit Strategy | **0.039528** | **92.3349** | **97.4034** | **62.2180** |
| $D^T$ | Only Point Cloud | 0.051007 | 87.1849 | 95.7969 | 63.4523 |
| | Implicit Strategy | 0.086719 | 75.1837 | 86.8626 | 60.1704 |
| | Explicit Strategy | **0.037604** | **92.4589** | **97.5734** | **62.4444** |

[a] EPE3D(End Point Error) represents the error between the predicted and ground truth values for each 3D point, measured in meters. AS (Accuracy Strict) measures the percentage of SSF errors with EPE3D $\leq$ 0.05 meters and relative errors $\leq$ 5%. AR(Accuracy Relaxation) measures the percentage of SSF errors with EPE3D $\leq$ 0.10 meters and relative errors $\leq$ 10%. Outliers represents the percentage of errors in scene flow estimation with EPE3D $>$0.10 meters and relative errors $>$30%.




(a) (b)

Fig. 9: The ablation experiments for the iterative optimization.

TABLE V: SLAM Trajectory Error for Scene Flow Odometry Module

| PC | IMU | Our_Seg | Our_SF | MEAN[a]↓ | RMSE↓ | STD↓ |
|---|---|---|---|---|---|---|
| ✔ | × | × | × | 25.97 | 26.64 | 5.94 |
| ✔ | × | × | ✔ | 5.66 | 6.35 | 2.88 |
| ✔ | ✔ | × | ✔ | 3.99 | 4.26 | 1.51 |
| ✔ | × | ✔ | ✔ | **3.06** | **3.85** | **1.33** |

[a] MEAN represents the average distance between the predicted and ground truth values for each sample point, measured in meters.

TABLE VI: Ablation Study of Semantic Segmentation Accuracy(SSA) and Scene Flow Results with Different Loss Configurations

| Dataset | $\mathcal{L}_{mot}$ | $\mathcal{L}_{SC}$ | $\mathcal{L}_{CD}$ | SSA(%)↑ | EPE3D(m)↓ | AS(%)↑ | AR(%)↑ |
|---|---|---|---|---|---|---|---|
| $D^H$ | ✔ | | | 55.36 | 0.104 | 73.47 | 86.49 |
| | ✔ | ✔ | | 88.41 | 0.058 | 87.45 | 93.33 |
| | ✔ | ✔ | ✔ | **90.85** | **0.040** | **92.33** | **97.40** |
| $D^T$ | ✔ | | | 63.49 | 0.116 | 76.86 | 89.48 |
| | ✔ | ✔ | | 90.34 | 0.068 | 90.48 | 90.47 |
| | ✔ | ✔ | ✔ | **93.27** | **0.038** | **92.46** | **92.57** |

*4) Scene Flow Odometry Ablation Experiment:* Ablation results in Table V evaluate SLAM trajectory errors on the same raw dataset within a fixed map. Methods compared are: (1) PC: point cloud input, (2) IMU: odometry with inertial measurement unit, (3) Our_Seg: OGC using point cloud and scene flow for clustering, (4) Our_SF: scene flow optimization. Results show that adding scene flow and semantic data greatly reduces SLAM trajectory errors.

*5) Different Loss Functions for Semantic Segmentation Accuracy Scene Flow Estimation:* Table VI shows an ablation on loss functions for dynamic-static segmentation and scene flow on $D^H$ and $D^T$. Configurations are: (1) using only the rigid motion consistency loss $\mathcal{L}_{mot}$, (2) incorporating the semantic scene flow consistency loss $\mathcal{L}_{SC}$, and (3) further adding the rigidity chamfer disatance loss $\mathcal{L}_{CD}$. The results show that incorporating $\mathcal{L}_{SC}$ with $\mathcal{L}_{mot}$ improves SSA by up to 33.11% compared to using $\mathcal{L}_{mot}$ alone. Additionally, introducing $\mathcal{L}_{CD}$ further enhances performance, achieving the lowest EPE3D error and the highest SSA, AS, and AR scores across both datasets. These findings demonstrate the effectiveness of our proposed loss functions in refining scene flow estimation.

## VI. Limitations and Real-World Discussion

While simulation offers a safe and controlled environment for developing and evaluating our SSF framework, it cannot fully replicate real-world complexities such as sensor noise, illumination and weather variations, and the presence of dynamic objects. These discrepancies may lead to performance degradation when deploying the system outside of simulation. To address these gaps, we plan to validate our approach on real-world datasets, and explore domain adaptation and sim-to-real transfer techniques to bridge the gap between synthetic and real environments. These steps aim to enhance the robustness and generalization capability of our method for practical deployment.

## VII. Conclusion

This paper has presented a Semantic Scene Flow-based Perception systems for Autonomous Navigation (SSF-PAN) in complex traffic scenarios, which includes a SSF module, an iterative optimization framework and a testing platform. The proposed SSF neural network based on OGC with the improved loss function can enhance the segmentation accuracy of both dynamic objects and the static environment. The proposed iterative optimization framework can improve both scene flow estimation and segmentation performance. The developed SSF-based navigation platform can ensure robust autonomous navigation by continuously assessing the performance of the SSF perception system in simulation environments, enabling map-free navigation in complex traffic scenarios. The experimental findings indicate that our SSF method exhibits superior performance in managing complex traffic conditions compared to current techniques across different navigation tasks, with approximately $1\% \sim 81\%$ improvement in success rate, $2\% \sim 8\%$ reduction in navigation time, and a $10\% \sim 17\%$ increase in the average moving speed. Future research will develop a high-precision multi-task model for scene flow estimation and point cloud segmentation without iterative processing.

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

## REFERENCES


[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[2] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. The slam problem: a survey. *Artificial Intelligence Research and Development*, pages 363–371, 2008.

[3] Pablo F Alcantarilla, José J Yebes, Javier Almazán, and Luis M Bergasa. On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 1290–1297. IEEE, 2012.

[4] Ziyang Song and Bo Yang. Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30798–30812. Curran Associates, Inc., 2022.

[5] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.

[6] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1168–1174. IEEE, 2018.

[7] Xingyu Liu, Charles R. Qi, and Leonidas J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–537, 2019.

[8] Shuaijun Wang, Rui Gao, Ruihua Han, Jianjun Chen, Zirui Zhao, Zhijun Lyu, and Qi Hao. Active scene flow estimation for autonomous driving via real-time scene prediction and optimal decision. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[9] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. *IEEE*, 2019.

[10] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly Supervised Learning of Rigid 3D Scene Flow, 2021.

[11] Kyle Vedder, Neehar Peri, Nathaniel Chodosh, Ishan Khatri, Eric Eaton, Dinesh Jayaraman, Yang Liu, Deva Ramanan, and James Hays. Zeroflow: Scalable scene flow via distillation, 2024.

[12] Qingwen Zhang, Yi Yang, Heng Fang, Ruoyu Geng, and Patric Jensfelt. Deflow: Decoder of scene flow network in autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2105–2111. IEEE, 2024.

[13] Qingwen Zhang, Yi Yang, Peizheng Li, Olov Andersson, and Patric Jensfelt. Seflow: A self-supervised scene flow method in autonomous driving. In *European Conference on Computer Vision*, pages 353–369. Springer, 2024.

[14] Florin Leon and Marius Gavrilescu. A review of tracking, prediction and decision making methods for autonomous driving. *arXiv preprint arXiv:1909.07707*, 2019.

[15] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13126–13136, 2021.

[16] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7108–7118, 2021.

[17] Hugues Thomas, Ben Agro, Mona Gridseth, Jian Zhang, and Timothy D Barfoot. Self-supervised learning of lidar segmentation for autonomous indoor navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14047–14053. IEEE, 2021.

[18] Vladislav Golyanik, Kihwan Kim, Robert Maier, Matthias Nießner, Didier Stricker, and Jan Kautz. Multiframe scene flow with piecewise rigid motion. In *2017 International Conference on 3D Vision (3DV)*, pages 273–281. IEEE, 2017.

[19] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1384, 2013.

[20] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Rigid scene flow for 3d lidar scans. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1765–1770. IEEE, 2016.

[21] Hassan Abu Alhaija, Anita Sellent, Daniel Kondermann, and Carsten Rother. Graphflow–6d large displacement scene flow via graph matching. In *Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings 37*, pages 285–296. Springer, 2015.

[22] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3614–3622, 2019.

[23] 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*, pages 7962–7971, 2019.

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[25] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8160–8171, 2019.

[26] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021.

[27] Philipp Jund, Chris Sweeney, Nichola Abdo, Zhifeng Chen, and Jonathon Shlens. Scalable scene flow from point clouds in the real world, 2021.

[28] Mohamed Aly. Real time detection of lane markers in urban streets. In *2008 IEEE intelligent vehicles symposium*, pages 7–12. IEEE, 2008.

[29] Young-Woo Seo and Ragunathan Raj Rajkumar. Detection and tracking of boundary of unmarked roads. In *17th International Conference on Information Fusion (FUSION)*, pages 1–6. IEEE, 2014.

[30] Teddy Ort, Liam Paull, and Daniela Rus. Autonomous vehicle navigation in rural environments without detailed prior maps. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2040–2047. IEEE, 2018.

[31] Alexander Amini, Guy Rosman, Sertac Karaman, and Daniela Rus. Variational end-to-end navigation and localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8958–8964. IEEE, 2019.

[32] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[33] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, pages 8248–8254. IEEE, 2019.

[34] Ruihua Han, Shuai Wang, Shuaijun Wang, Zeqing Zhang, Qianru Zhang, Yonina C. Eldar, Qi Hao, and Jia Pan. Rda: An accelerated collision free motion planner for autonomous navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 8(3):1715–1722, 2023.

[35] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[36] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[37] 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.

[38] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[39] Henrik Bäcklund, Anders Hedblom, and Niklas Neijman. A density-based spatial clustering of application with noise. *Data Mining TNM033*, 33:11–30, 2011.

[40] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.