2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Hybrid Bird's-Eye Edge Based Semantic Visual SLAM for Automated Valet Parking

Zhenzhen Xiang, Anbo Bao and Jianbo Su

***Abstract*— Vision-based localization and mapping solution is promising to be adopted in the automated valet parking task. In this paper, a semantic SLAM framework that leverages the hybrid edge information on bird's-eye view images is presented. To extract useful edges from the synthesized bird's-eye view image and the free-space contours for the SLAM task, different segmentation methods are designed to remove the noisy glare edges and distorted object edges caused by the inverse perspective mapping in view synthesis. Since only the free-space segmentation model needs training, our methods can dramatically reduce the labeling burden compared with previous road marking based methods. Those incorrect and incomplete edges are further cleaned and recovered by a temporal fusion of consecutive edges in a local map, respectively. Both a semantic edge point cloud map and an occupancy grid map can be built simultaneously in real time. Experiments in a parking garage demonstrate that the proposed framework can achieve higher accuracy and perform more robustly than previous point feature based methods.**

I. Introduction

Automated valet parking (AVP) is one of the most promising applications in autonomous driving. For a standard AVP task, the simultaneous localization and mapping (SLAM) system is essential for an intelligent vehicle to autonomously navigate in a parking lot or garage [1]. In recent years, as vehicles are equipped with more and more cameras, the vision-based SLAM systems have attracted extensive attention [2].

Different from traditional visual SLAM frameworks that directly process multiple raw images, in this paper, we mainly focus on the bird's-eye view image synthesized from four surrounding fisheye images by inverse perspective mapping (IPM). The main advantage of taking the bird's-eye view image as input is to dramatically decrease the computation load, which is very important in real-time applications.

As shown in Fig. 1, the bird's-eye view image can preserve most of the information on the ground surface around the vehicle with a 360° field of view. Since there are usually complex lighting conditions (e.g., glares reflected by the ground) and plenty of dynamic objects (e.g., moving cars and pedestrians) in parking scenarios, it is a very challenging task to perform the SLAM task. To handle these difficult situations, with the development of deep learning, recent work attempts to train a deep neural network in order to extract semantic information for more accurate and robust localization and mapping. Although a significant progress has been made with such methods, a huge amount of labeled images are needed for training the models, especially when the system is adapted to new unseen environments.

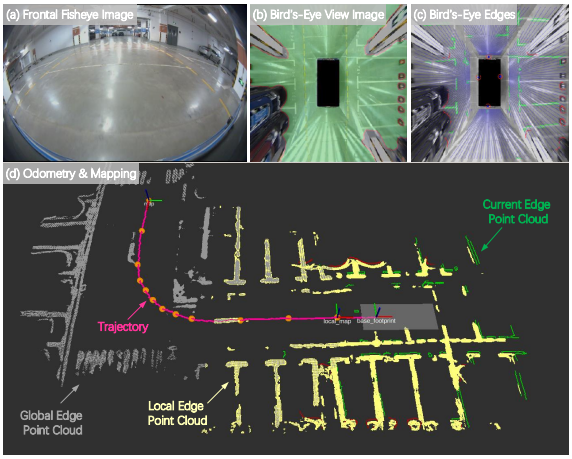


Fig. 1. An overview of our solution for automated valet parking, including (a) a sample image captured by the fisheye cameras on our autonomous vehicle, (b) the bird's-eye view image generated by view synthesis with free-space segmentation result, (c) the extracted edges for semantic visual SLAM system, and (d) the built edge point cloud map as well as the trajectory.

In this paper, we proposed a hybrid semantic information extraction method which combines a classical unsupervised edge detector and a group of IPM-based edge segmentation methods. As shown in Fig. 1(b), our framework only requires the coarse segmentation of free space instead of the fine segmentation of road markings, which can greatly reduce the burden on data annotation. With our edge segmentation methods, most of the noisy and distorted edges, e.g., the edge of glares and the contour of objects over the ground, can be filtered by leveraging the prior knowledge of IPM (Fig. 1(c)). Therefore, both the bird's-eye edges inside the free space and the contours of free space can be combined as hybrid edges and input to the SLAM system for localization and mapping. We can find that the generated edge point cloud map in Fig. 1(d) clearly reflects the road markings on the ground surface.

II. Related Work

A. Visual SLAM for Multi-Camera Systems

Increasing the number of cameras in visual SLAM system can significantly enhance the accuracy and robustness of the system. Therefore, more and more state-of-the-art visual SLAM frameworks such as SVO [3], VINS-Fusion [4] and

This work was partially financially supported by the projects of National Natural Science Foundation of China under grant 61533012, 91748120 and 52041502.

Zhenzhen Xiang, Anbo Bao and Jianbo Su are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China. Emails: wwfzs1990@gmail.com

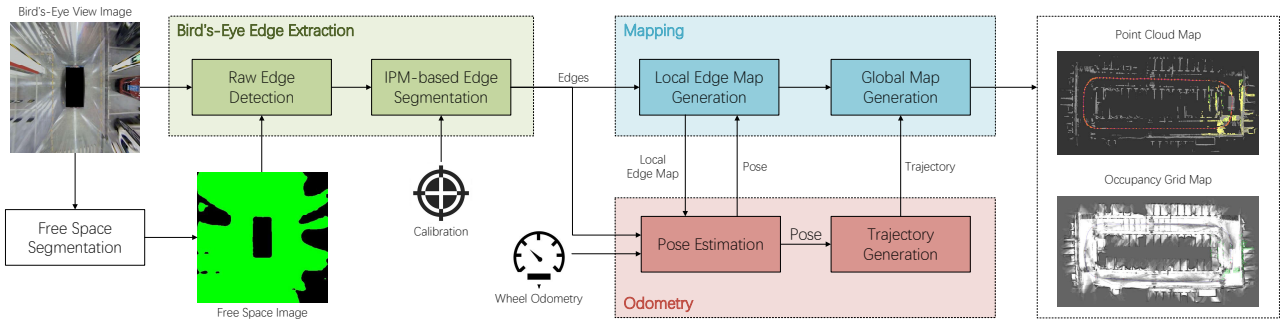


Fig. 2. The proposed bird's-eye edge based semantic visual SLAM framework. The bird's-eye view image is generated from four fisheye images by view synthesis. The free space is segmented to provide a mask for edge detection. The bird's-eye edge extraction module processes the raw edges detected from both the bird's-eye view image and free-space image, followed by an IPM-based edge segmentation procedure to remove noisy and distorted edges. The local map is generated by fusing consecutive edge images, which is used for relative pose estimation and global mapping. Both the edge point cloud map and occupancy grid map are simultaneously built and optimized.

OpenVINS [5] have supported multi-camera configurations. Different camera models [6] and various camera setups [7], [8] have been extensively investigated. Recently, Kuo et al. redesigned the visual SLAM framework for arbitrary multi-camera systems which could adapt to various camera configurations [9].

Although directly processing the raw images from multiple cameras can maximize the use of original information, it also requires a powerful computation platform and may not be suitable for real-time applications. As an alternative, recent work also investigates using the synthesized bird's-eye view or surrounding view images, which can efficiently perform the SLAM task with decent accuracy [10], [11], including our previous work [12].

B. Semantic Visual SLAM in Autonomous Driving

Semantics-aided visual SLAM has been extensively investigated for several years in the field of autonomous driving, especially with the rapid development of deep learning. General semantic visual SLAM mainly focuses on the semantics extraction [13] and data association [14], [15]. Semantic segmentation and object detection models are widely used for extracting point-level [16]–[18] and object-level [19]–[21] semantics, which have brought additional semantic and geometric constraints to SLAM formulations for more stable and accurate performance. However, since the deep learning based models usually need plenty manually labeled data for training, it may take a long time before these methods are transferred to new scenarios. Therefore, we consider to design a hybrid edge extraction method only requiring coarse free-space segmentation results, which can be about $10\times$ faster than previous methods for labeling a frame.

C. Automated Valet Parking Applications

Vision-based solutions for AVP applications are the trend of recent investigations as the cameras are much cheaper than LiDAR sensors and can capture rich semantic information. The V-Charge project which aimed to provide AVP services with close-to-market sensors has made a significant progress on building a vision-based navigation system for AVP tasks [22]–[24]. Both the metric and semantic information were adopted in [23] to build a parking space map for navigation, which has been followed by [25] with a tightly-coupled multi-sensor fusion based framework. With pixel-level segmentation results of the environment, the authors of [26] and [27] similarly took the road markings as the input and built a complete visual SLAM system for AVP tasks. In our previous work [12], we have demonstrated the effectiveness of incorporating the free-space contour to a feature-based visual odometry system. To further leverage the semantics in bird's-eye view images, we attempt to extract the edges on the ground surface which is much denser and more stable for visual SLAM systems.

III. Framework

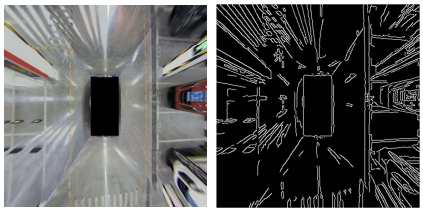
In general, our framework is a hybrid edge based semantic visual SLAM system using bird's-eye view images as input. It mainly consists of three parts as shown in Fig. 2:

- **Bird's-eye edge extraction:** The synthesized bird's-eye view image as well as the segmented free-space image are taken as the input of our system. Raw hybrid edges including those inside the free space and the contour of free space are detected, followed by an IPM-based segmentation module which is able to remove most of noisy edges caused by the glares on the ground as well as distorted object edges above the ground.
- **Mapping:** With the extracted hybrid edges, the mapping module accumulates and generates a probabilistic local edge map for relative pose estimation. A global edge point cloud map as well as an occupancy grid map can be derived with the recorded trajectory and pose graph optimization.
- **Odometry:** The current pose in local map is estimated by semantic point cloud registration, given the initial transformation from wheel odometry. The pose of each frame is further accumulated to generate the vehicle's trajectory in global map.

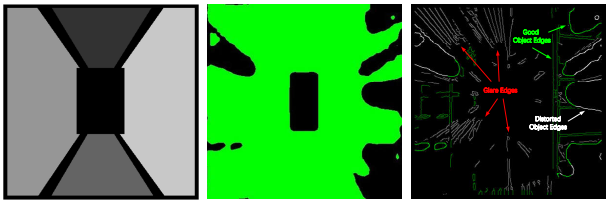
In the following two sections, details of our framework will be illustrated.

IV. Bird's-Eye Edge Extraction

As we can find from sample bird's-eye view image in Fig. 3(a), there are plenty of road markings on the ground surface, which are good landmarks for a SLAM system. However, the image is also polluted by the glares from the



(a) A bird's-eye view image (left) and the detected edges by Canny edge detector (right)



(b) The predefined view mask image (left), segmented free-space image (middle) and the masked edges (right)

Fig. 3. The detected raw edges on a bird's-eye view image and the masked edges considering the view synthesis and free-space segmentation. Note that different types of edges as pointed out.

lights reflected by the ground. In addition, the contours of those objects over the ground are dramatically distorted in IPM for view synthesis. Thus, the role of our bird's-eye edge extraction module is to detect and preserve high quality edges from the input bird's-eye view and free-space images, which is fundamental for the whole SLAM system.

Two submodules are concatenated in the edge extraction procedure. Firstly, raw edges are detected from the input images. Then these edges are segmented to remove noisy and distorted parts by considering the distortion effect of IPM in view synthesis.

A. Raw Edge Detection

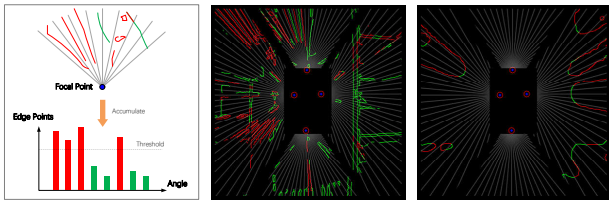
The raw edges on the input images can be detected by either traditional edge detection algorithms (like Canny edge detector [28]). As shown in Fig. 3(a), on the one hand, the edges of road markings as well as object footprints are successfully detected on the whole; on the other hand, such detectors normally are unable to distinguish the useful edges for SLAM task from those useless and noisy edges of the surrounding vehicles, pillars, or glares.

With the help of free-space segmentation, the edges inside the objects over the ground can be removed. However, the remaining parts still contains a significant amount of disturbing edges from the glares and objects distorted by the IPM as depicted in Fig. 3(b).

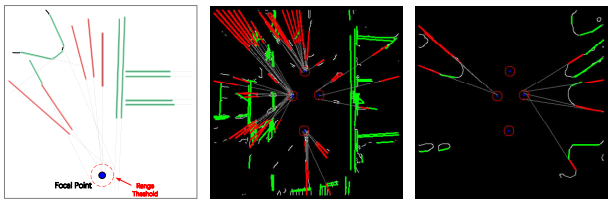
Therefore, the masked edges need further processing before sending to the mapping and odometry modules. Otherwise, the performance of following procedures will be dramatically decreased.

B. IPM-based Edge Segmentation

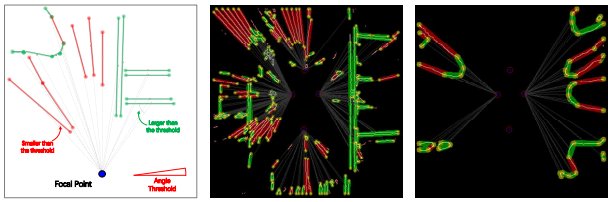
The basic idea of cleaning the edges is to leverage the characteristic of distortion effect in IPM. As we can find in Fig. 3, the edges of glares and objects are stretched in the view synthesis. Most of these edges are consistently radial, which approximately pass the focal point of each



(a) Ray-based segmentation (left) and its output edges for bird's-eye view image (middle) and free-space image (right)



(b) Line-based segmentation (left) and its output edges for bird's-eye view image (middle) and free-space image (right)



(c) Polyline-based segmentation (left) and its output edges for bird's-eye view image (middle) and free-space image (right)

Fig. 4. Illustrations of different edge segmentation methods and their results. The red edges are segmented as those distorted by IPM and will be removed. The green edges are preserved for the mapping and odometry modules.

camera. Such phenomenon motivates us to design various segmentation methods which are depicted in Fig. 4.

Ray-based segmentation. A naïve and straightforward thought is to consider the detected edges as the results of a 2D LiDAR sensor like our previous work [12]. As shown in Fig. 4(a), the view of a camera sensor is divided evenly with a specific angle step. For each angle bin, the edge points that fall in the same bin are accumulated. Then the edge points in those bins with larger number of points than a predefined threshold are removed.

Although the ray-based segmentation method is simple, fast and able to remove a certain part of the distorted edges, its drawbacks are also obvious. Firstly, the dividing lines of angle bins are predefined and can separate a long distorted edge into two parts. This is because the distorted edges are usually not that straight. Secondly, the structure of edges is not considered and is broken in the accumulation, which will make those small and dense edges be incorrectly removed. Finally, tuning the parameters of ray-based segmentation can also be difficult as the angle step and threshold are coupled.

Line-based segmentation. To consider the geometric distribution of edges, we further attempt to detect line segments on the edge image. Specifically, as shown in Fig. 4(b), line segments are detected by a line segment detector (e.g., a Hough transform based detector [29]). Then the distance between the focal point of camera and each line segment is computed and compared to a distance threshold. Those

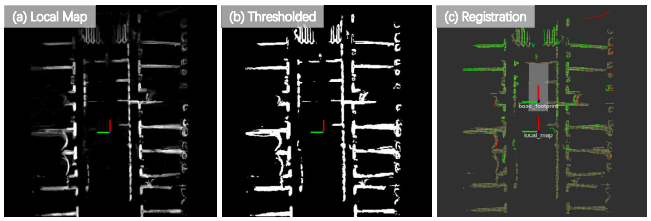


Fig. 5. The pipeline of local map generation for pose estimation, including (a) the accumulated probabilistic local map, (b) the thresholded local map and (c) the extracted edges (bird's-eye edges in green and free-space edges in red) registered to the local map.

line segments that are sufficiently close to the focal point will be marked as distorted edges.

The main advantage of line-based segmentation is reducing the possibility of falsely removing dense edges along a specific radial direction. Also, the parameters of detection and segmentation threshold are clearly divided. However, since the line segment detection is generally performed at the level of edge image, the points from unconnected edges can be wrongly considered as the same part. In addition, it can be difficult for those edges far away from the camera to fulfill the condition for removing if they are not straight enough.

Polyline-based segmentation. To further consider the fine edge structures, we firstly distinguish different edge instances before the segmentation. Then for each edge instance, the Douglas-Peucker algorithm [30] is applied to simplify the edge and generate a polyline to represent it. As shown in Fig. 4(c), the focal point is connected to the center of each line segment in each polyline. The angle distance between the line segment and the connected line is evaluated with an angle threshold. We can find that those line segments on distorted edges are commonly with smaller angles.

The major superiority of polyline-based method is that it simplifies the line estimation and restricts it to the level of edge instance. Also, it is less sensible to evaluate the angle between two line direction vectors than the distance from the focal point to the line segment, especially for those edges far from the vehicle.

It should be noticed that, there still exist some small edges from the glares that are not completely removed by the polyline-based method. In the meanwhile, some useful road marking edges can be wrongly removed when they are just at the same radial direction of a camera. For these segmentation errors, we will further deal with them in the local mapping by leveraging the temporal fusion of consecutive observations.

V. Semantic Odometry and Mapping

After the segmentation of bird's-eye edges, we can convert them to point clouds in Euclidean space with the calibrated scale factor. With such hybrid point clouds which have two classes of edges (i.e., *bird's-eye edge* and *free-space edge*), a semantic SLAM system can be established.

A. Local Map Generation

As previously discussed in Sec. IV-B, the segmentation module is unable to completely remove the noisy edges, for example, some parts of the glare edges, due to the limitation of its mechanism. Meanwhile, some road markings or parking spot edges can be incorrectly removed when they temporarily locate on the ray direction from the focal point of a camera. Thus, the extracted edges on current image are commonly incomplete and unstable, which will make it difficult to acquire robust relative pose estimation with a key frame based strategy.

To overcome the limitation of single frame based estimation, we alternatively accumulate the extracted edges in consecutive frames and build a local edge map for more stable motion estimation. The local map can be easily initialized with the edges on the first frame. Then by iteratively estimating the poses of following frames, new edges are transformed and added to the local map frame by frame. A sample of derived local map is shown in Fig. 5, where the fusion of different frames is probabilistic. To update the local map smoothly and fill small gaps, the edges are processed by a Gaussian filter before adding them to the map.

In order to remove the incorrectly segmented edges which have been added to the local map, a sliding window fusion is also applied. The frame buffer length for the local map is restricted to a fixed number. When it is full, the oldest edges will be subtracted from the local map. It should be noticed that, those edge points with sufficient high probabilities will not be modified in the subtraction, which can keep those significant edges in the local map as much as possible.

When the vehicle's translation or rotation exceed predefined thresholds, the local map will be transformed to current pose and those edges outside the range of local map will not be maintained and updated.

B. Pose Estimation

The fused local edge map is thresholded and converted to point cloud in world coordinates. To estimate the current vehicle's pose in local map, the edge points in current frame are first projected to the local map by the transformation from wheel odometry. Then the data association is established by the nearest neighbor searching. Finally, the current pose in local map $\mathbf{T}_{vehicle}^{local}$ can be estimated by solving the following problem:

$$\min_{\mathbf{T}_{vehicle}^{local}} \underbrace{\sum_i \left\| \mathbf{p}_i^{local} - \mathbf{T}_{vehicle}^{local} \mathbf{p}_i^{vehicle} \right\|_2}_{\text{bird's-eye edge point distances}} + \underbrace{w \sum_j \left\| \mathbf{p}_j^{local} - \mathbf{T}_{vehicle}^{local} \mathbf{p}_j^{vehicle} \right\|_2}_{\text{free-space edge point distances}}, \tag{1}$$

where $\mathbf{p}_i^{local}$ and $\mathbf{p}_i^{vehicle}$ are the i-th point correspondence for bird's-eye edge clouds between current frame and local map, while $\mathbf{p}_j^{local}$ and $\mathbf{p}_j^{vehicle}$ represent the j-th free-space edge point correspondence. w is the weight factor to balance the costs between the above two different classes of edges.

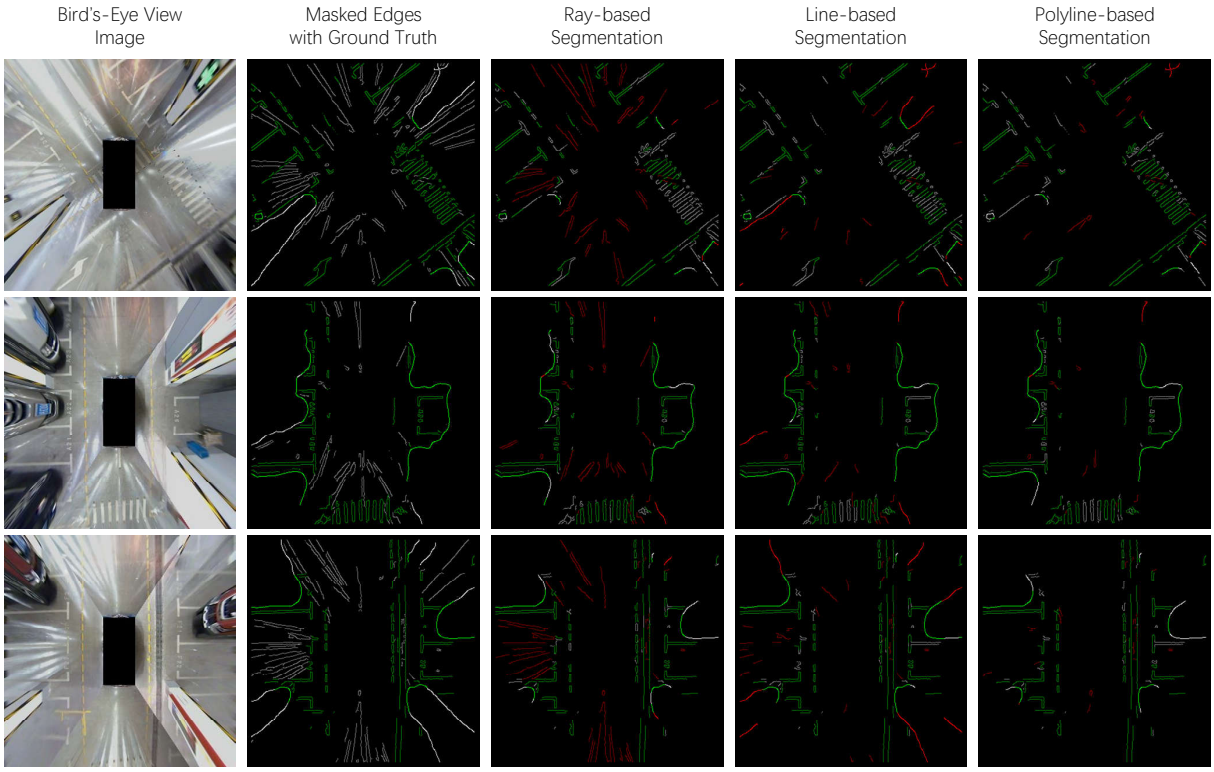


Fig. 6. The extracted bird's-eye edges by different segmentation methods. The manually labeled ground-truth edges for evaluation are drawn in green on the masked edges. For each column of segmented edges, the colors represent whether the edges are correctly preserved (green) or not (red), while the white edges are those missed by the proposed methods. In addition, The color intensity means whether the edges are inside the free space (brighter) or on the contour of free space (darker).

C. Global Mapping

The global map can be derived from the the local map and the global trajectory of the vehicle. Since the extracted free-space edges can also be considered as 2D LiDAR measurements, not only a semantic point cloud map but also an occupancy grid map can be simultaneously built, as shown in Fig. 2. When a loop closure is detected between local maps, which is similar to the pose estimation problem, a normal pose graph optimization is performed to correct the drift.

VI. Experiments

A. Data Collection

The data sequence used to evaluate our framework was collected in an underground garage. Detailed information of the sequence is listed in Table I. The size of synthesized bird's-eye view image is 384 $\times$ 384, which covers a region of 15.3 m $\times$ 15.3 m around the vehicle. A modified ICNet model [31] was applied to get the free-space segmentation results. The ground truth trajectory was computed with the fusion of wheel odometry and high-precision IMU measurements since GPS information was not available underground.

TABLE I
Data Sequence Information

Scenario	Length [m]	Duration [s]	Mean Speed [m/s]	Max. Speed [m/s]	Bird's-Eye Images
Underground Garage	128.38	201	0.64	0.96	1336

B. Bird's-Eye Edge Extraction

We first evaluate the performance of different segmentation methods for extracting hybrid bird's-eye edges. Both qualitative and quantitative results are shown to demonstrate their capabilities of filtering noisy and distorted edges.

1) Qualitative results: Several sample images and their corresponding edge extraction results are listed in Fig. 6. For each sample, we manually labeled on the edge images masked by the view range and free space (see Fig. 3(b)) to derive the ground truth. Then the final edges output by each method were compared with the ground truth and visualized in different colors.

From the figure we can find that, for the ray-based segmentation, too many edges of glares are wrongly preserved. This situation may happen when those edges are just divided into two bins or they are not that long enough compared with the threshold. For the line-based segmentation, most parts of the glares' edges are successfully removed. However, those useless edges that are far away and not that straight (especially the contour of free space) may fail to satisfy the range threshold to the focal point, thus, they will not be filtered. Finally, the polyline-based segmentation method can successfully remove most of the noisy and distorted edges at the expense of missing a small part of useful edges.

2) Quantitative results: To quantitatively analyze the performance of each method, we manually labeled 30 edge images to compute the metrics of *Precision* and *Recall* for each method. For a fair evaluation, we tuned the thresholds of each method to make them at the same level of *Recall*

TABLE II

PRECISION AND RECALL FOR EACH EDGE SEGMENTATION METHOD

Method	Precision	Recall
Ray-based Edge Segmentation	0.631	0.729
Line-based Edge Segmentation	0.751	0.730
Polyline-based Edge Segmentation	**0.863**	**0.732**

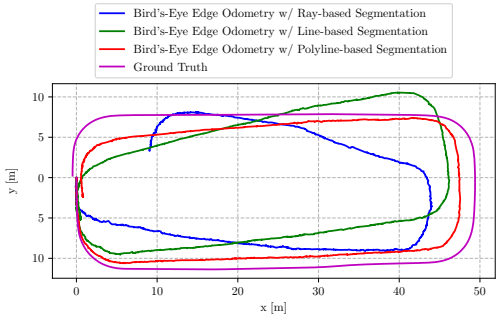


Fig. 7. The estimated trajectories with different bird's-eye edge segmentation methods.

TABLE III

ABSOLUTE TRAJECTORY ERROR OF PROPOSED FRAMEWORK WITH DIFFERENT EDGE SEGMENTATION METHODS

Edge Segmentation	RMSE [m]	Mean [m]	Max [m]
Ray-based	6.493	5.975	10.772
Line-based	4.148	3.797	6.137
Polyline-based	**1.993**	**1.845**	**3.169**

rate. The threshold for the number of edge points in ray-based segmentation was set to 110 with the ray angle step of 4°. The range threshold to the focal point in line-based segmentation was tuned to 18.0. The angle threshold in polyline-based segmentation was configured to 8.0°.

As shown in Table II, when the *Recall* rate of each method is all around 0.73, the polyline-based edge segmentation method achieves the highest precision, which is much better than the ray-based method (+23.2%) and line-based method (+11.2%).

C. Odometry & Mapping

1) Trajectories with Different Edge Segmentation Methods: Since the performance of a SLAM system is significantly influenced by the quality of input data, we recorded and evaluated the bird's-eye odometry with the edges extracted by different segmentation methods. The estimated trajectories are plotted in Fig. 7 and the root mean square (RMS) errors to the ground truth are listed in Table III, which are computed by the evaluation tool presented in [32]. The results suggest that the bird's-eye edges extracted by polyline-based method can dramatically improve the performance of visual odometry system compared with the other two methods.

2) Trajectories with Different Frameworks: To demonstrate the effectiveness of bird's-eye edges, the proposed framework was compared with the ORB feature based and free-space scan based method in our previous work [12]. In addition, since the extracted edges can be represented with point clouds, it is very convenient to input such data to a general LiDAR based SLAM system like *Cartographer* [33] in our experiments.

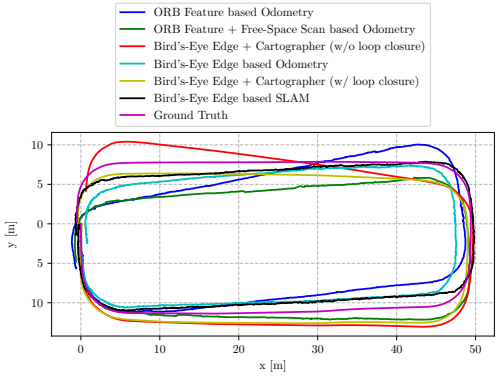


Fig. 8. The estimated trajectories with different SLAM frameworks.

TABLE IV

ABSOLUTE TRAJECTORY ERROR OF DIFFERENT FRAMEWORKS

Method	RMSE [m]	Mean [m]	Max [m]
ORB Feature	3.264	2.702	5.977
ORB Feature + Free Space Scan	2.776	2.245	5.114
Bird's-Eye Edge Odometry	2.069	1.859	3.560
Bird's-Eye Edge + Cartographer (no loop closure)	2.651	2.414	4.136
Bird's-Eye Edge + Cartographer (loop closure)	1.397	**1.173**	2.253
Bird's-Eye Edge SLAM	**1.310**	1.225	**1.918**

As shown in Fig. 8 and Table IV, on the one hand, compared with the ORB features and free-space scans, the bird's-eye edge based odometry has achieved better results, which suggests the hybrid edges are more effective and stable than the point features in our task; on the other hand, when inputting the bird's-eye edges to *Cartographer*, the pure odometry without pose graph optimization has an obvious drift, which can be eliminated after the loop closure.

VII. CONCLUSIONS

In this paper, we proposed a semantic visual SLAM framework based on hybrid edges extracted from the bird's-eye view images, which can be applied to the AVP tasks. On the one hand, compared with traditional sparse point features, the edges are much denser and more robust. On the other hand, compared with recent semantic road marking point clouds, our segmentation methods only need coarse free-space annotations for training, instead of the time-consuming and labor-intensive labeling for road markings. By leveraging the IPM distortion effect, most noisy glare edges and distorted object edges can be filtered by the proposed segmentation methods. With the input of extracted edges, a visual SLAM system was built and evaluated on the practical data in an underground garage. Experimental results demonstrated the effectiveness of our method and its potential to be integrated into other general SLAM frameworks. In the future, we will further develop a robust and hierarchical semantic visual SLAM framework that combines multimodal semantics and multiple sensors for the AVP applications.

REFERENCES

[1] H. Banzhaf, D. Nienhüser *et al.*, "The Future of Parking: A Survey on Automated Valet Parking with an Outlook on High Density Parking," in *Proceedings of 2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1827–1834.

[2] G. Bresson, Z. Alsayed *et al.*, "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[3] C. Forster, Z. Zhang *et al.*, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[4] T. Qin, J. Pan *et al.*, "A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors," *ArXiv190103638 Cs*, Jan. 2019.

[5] P. Geneva, K. Eckenhoff *et al.*, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 4666–4672.

[6] G. H. Lee, F. Faundorfer, and M. Pollefeys, "Motion Estimation for Self-Driving Cars with a Generalized Camera," in *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 2746–2753.

[7] P. Liu, M. Geppert *et al.*, "Towards Robust Visual Odometry with a Multi-Camera System," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1154–1161.

[8] H. Seok and J. Lim, "ROVO: Robust Omnidirectional Visual Odometry for Wide-baseline Wide-FOV Camera Systems," in *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6344–6350.

[9] J. Kuo, M. Muglikar *et al.*, "Redesigning SLAM for Arbitrary Multi-Camera Systems," in *Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 2116–2122.

[10] E. S. Lee, W. Choi, and D. Kum, "Bird's Eye View Localization of Surrounding Vehicles: Longitudinal and Lateral Distance Estimation with Partial Appearance," *Robotics and Autonomous Systems*, vol. 112, pp. 178–189, Feb. 2019.

[11] L. Deng, M. Yang *et al.*, "Semantic Segmentation-Based Lane-Level Localization Using Around View Monitoring System," *IEEE Sens. J.*, vol. 19, no. 21, pp. 10 077–10 086, Nov. 2019.

[12] Z. Xiang, J. Yu *et al.*, "ViLiVO: Virtual LiDAR-Visual Odometry for an Autonomous Vehicle with a Multi-Camera System," in *Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 2486–2492.

[13] S. Grigorescu, B. Trasnea *et al.*, "A Survey of Deep Learning Techniques for Autonomous Driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, 2020.

[14] S. L. Bowman, N. Atanasov *et al.*, "Probabilistic Data Association for Semantic SLAM," in *Proceedings of 2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1722–1729.

[15] K. Doherty, D. Fourie, and J. Leonard, "Multimodal Semantic SLAM with Probabilistic Data Association," in *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2419–2425.

[16] J. Jeong, Y. Cho, and A. Kim, "Road-SLAM : Road Marking Based SLAM with Lane-Level Accuracy," in *Proceedings of 2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1736–1473.

[17] K.-N. Lianos, J. L. Schonberger *et al.*, "VSO: Visual Semantic Odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018, pp. 234–250.

[18] X. Wu, A. Benbihi *et al.*, "Semantic Nearest Neighbor Fields Monocular Edge Visual-Odometry," *ArXiv190400738 Cs*, Apr. 2019.

[19] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[20] C. Rubino, M. Crocco, and A. Del Bue, "3D Object Localisation from Multi-View Image Detections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1281–1294, Jun. 2018.

[21] Y. Zhang, H. Zhang *et al.*, "Bundle Adjustment for Monocular Visual Odometry Based on Detections of Traffic Signs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 151–162, Jan. 2020.

[22] P. Furgale, U. Schwesinger *et al.*, "Toward Automated Driving in Cities using Close-to-Market Sensors: An Overview of the V-Charge Project," in *Proceedings of 2013 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2013, pp. 809–816.

[23] H. Grimmett, M. Buerki *et al.*, "Integrating Metric and Semantic Maps for Vision-Only Automated Parking," in *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2159–2166.

[24] U. Schwesinger, M. Bürki *et al.*, "Automated Valet Parking and Charging for E-Mobility," in *Proceedings of 2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 157–164.

[25] X. Shao, L. Zhang *et al.*, "A Tightly-coupled Semantic SLAM System with Visual, Inertial and Surround-view Sensors for Autonomous Indoor Parking," in *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 2691–2699.

[26] J. Hu, M. Yang *et al.*, "Mapping and Localization Using Semantic Road Marking with Centimeter-level Accuracy in Indoor Parking Lots," in *Proceedings of 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 4068–4073.

[27] T. Qin, T. Chen *et al.*, "AVP-SLAM: Semantic Visual Mapping and Localization for Autonomous Vehicles in the Parking Lot," *ArXiv200701813 Cs*, Jul. 2020.

[28] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[29] C. Galamhos, J. Matas, and J. Kittler, "Progressive Probabilistic Hough Transform for Line Detection," in *Proceedings of 1999 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Jun. 1999, pp. 554–560 Vol. 1.

[30] D. H. Douglas and T. K. Peucker, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature," *Cartographica*, vol. 10, no. 2, pp. 112–122, Dec. 1973.

[31] H. Zhao *et al.*, "ICNet for Real-Time Semantic Segmentation on High-Resolution Images," in *Proceedings of 2018 European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420.

[32] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," in *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 7244–7251.

[33] W. Hess, D. Kohler *et al.*, "Real-Time Loop Closure in 2D LIDAR SLAM," in *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1271–1278.

[34] A. V. Savkin, "Decentralised Coordination of Groups of Autonomous Mobile Robots," *Int. J. Syst., Control Commun.*, vol. 1, no. 1, pp. 72–81, Jul. 2008.