2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Monocular Localization with Semantics Map for Autonomous Vehicles

Jixiang Wan[1,2,*,†], Xudong Zhang[1,†], Shuzhou Dong[1], Yuwei Zhang[1],
Yuchen Yang[1], Ruoxi Wu[1], Ye Jiang[1], Jijunnan Li[1], Jinquan Lin[1], Ming Yang[2]

***Abstract*— Accurate and robust localization remains a significant challenge for autonomous vehicles. The cost of sensors and limitations in local computational efficiency make it difficult to scale to large commercial applications. Traditional vision-based approaches focus on texture features that are susceptible to changes in lighting, season, perspective, and appearance. Additionally, the large storage size of maps with descriptors and complex optimization processes hinder system performance. To balance efficiency and accuracy, we propose a novel lightweight visual semantic localization algorithm that employs stable semantic features instead of low-level texture features. First, semantic maps are constructed offline by detecting semantic objects, such as ground markers, lane lines, and poles, using cameras or LiDAR sensors. Then, online visual localization is performed through data association of semantic features and map objects. We evaluated our proposed localization framework in the publicly available KAIST Urban dataset and in scenarios recorded by ourselves. The experimental results demonstrate that our method is a reliable and practical localization solution in various autonomous driving localization tasks.**

I. INTRODUCTION

In recent times, autonomous vehicles have received increasing attention from both academia and industry. Accurate and robust self-localization is critical for autonomous driving and serves as the foundation for subsequent applications, which include path planning, cooperative driving, map updating, and more. Although centimeter-level localization accuracy is now achievable in many scenarios by utilizing high-precision sensors like GPS-RTK and LiDAR, their expensive hardware costs create obstacles for their widespread utilization. In contrast, vision sensors such as cameras with their mature processes and low expense are gaining significant attention in the realm of commercial autonomous driving solutions.

To achieve visual global localization, one popular approach is to solve the PnP problem which creates associations between the 2D features tracked in the current image and the 3D features in the pre-constructed Structure From Motion (SFM) map. To ensure success under varying viewpoints and lighting, it is crucial that the extracted visual features are highly repeatable and consistent. Recent studies like [1], [2] have implemented learnable descriptors and matching strategies based on deep learning for good performance. Other researchers such as in [3], [4] proposed end-to-end pose regression models, achieving outstanding results in their experiments. However, the generalization capability of these methods to new environments has not been demonstrated. Furthermore, the generating of complex descriptors can significantly increase map memory usage, which affects the computational efficacy, particularly in city-scale localization tasks.

The integration of semantic information has been shown to significantly enhance the accuracy and robustness of location estimation. Recent studies [5]–[7] have demonstrated the benefits of utilizing semantic information in the environment to improve the representation of visual features and simplify the computational requirements. Especially in autonomous driving scenarios, lightweight localization can be achieved by detecting semantic objects such as ground markers, lane lines, crosswalks and pole-like objects [8]–[10]. Compared with traditional visual features, these semantic features are widely available on urban roads and have long-term stability and robustness in the face of weather changes, light fluctuations, perspective changes, and occlusions caused by dynamic obstacles [11], [12]. Furthermore, producing a semantic map using semantic objects instead of dense points can further reduce the cost of map distribution and storage.

Associating semantic cues from current observations and elements in a semantic map offers a promising solution for monocular visual localization in autonomous vehicles. However, there are several challenges to consider. On the one hand, standard vector High-Definition (HD) maps usually require specialized data-acquisition equipment and significant manpower for labeling. On the other hand, correctly transforming targets in 2D images to 3D real shapes presents a challenging problem due to dimension degradation defects. Therefore, this paper proposes a lightweight visual localization pipeline for autonomous driving, consisting of a semantic map constructor without manual annotation and a localization module using low-cost cameras and IMU devices. The main contributions of this paper are summarized as follows:

- We propose an enhanced inverse perspective mapping model that considers the rotation of camera, allowing for the accurate computation of bird's-eye view images during motion.
- We propose an algorithm that facilitates the construction of global semantic maps using conventional LiDAR with minimal annotation assistance or supervision.
- We present a monocular localization algorithm based on common road visual semantic features and validate its effectiveness in real traffic scenarios.

[1]OPPO Research Institute, Shanghai, China.
[2]Department of Automation, Shanghai Jiao Tong University, Shanghai, China.
† Authors contributed equally to this work.
∗ indicates corresponding author. Contact: wanjixiang@oppo.com

II. RELATED WORKS

A. Visual Localization

VINS [13] and ORB-SLAM [14]–[16] are commonly used visual SLAM frameworks to achieve accurate trajectory measurements, integrating modules with feature point extraction and matching, keyframe bundle adjustment, loop closure detection, and map registration. Hloc [1] constructs a global visual localization framework that includes image retrieval, local feature matching, and pose regression. Nonetheless, The real-time localization at city-scale presents a challenge for them.

LaneLoc [17] is one of the pioneers that utilizes lane lines in combination with a prior semantic maps. TM3Loc [18] propose a tightly-coupled vehicle localization framework using semantic landmark matching in a HD Map. RSCM [19] attempts to resolve the underdetermined problem in registration methods by dividing lane segments into shapes and curves. Dt-loc [20] proposes distance transforms of the semantic detection to enable the differentiable data association process to achieve high localization precision. LAVIL [21] explores the limit of visual semantic localization with the aid of LiDAR odometry. Improving the accuracy and reducing the cost of manual production of the prefabricated maps are effective ways to advance semantic localization approaches.

B. LiDAR SALM

LiDAR has the ability to detect the real scale and location information of objects, which can significantly enhance the creation of high-precision semantic maps. Most existing LiDAR SLAM works can be traced back to the LOAM algorithm [22], which proposes approaches for extracting valid feature points and registering a global map. The subsequent Lego-LOAM [23] method disregards ground points to expedite computation and incorporates a loop closure detection module to reduce the long-term drift. LIOM [24] introduces IMU pre-integration into odometry and proposes a LiDAR-imu tightly coupled SLAM method. FAST-LIO [25] addresses the issue of motion distortion during point cloud scanning by utilizing IMU measurements to compensate for motion distortion, while improving the Kalman gain formula formulation to reduce the computational effort of iterative optimization. FAST-LIO2 [26] proposes an incremental k-d tree data structure, ikd-Tree, to improve the search efficiency. It makes large-scale dense point cloud computing possible.

III. PROPOSED APPROACH

In this work, we present a visual localization method based on semantic map, as shown in Fig. 1. The system consists two parts: (1) Global semantic map generation. The data collected from roads by vehicles equipped with LIDAR, GPS-RTK and IMU, or other navigation sensors, is utilized in creating point cloud maps using LiDAR SLAM. The semantic features such as lane lines, lane signs, and pole-like objects are extracted from the point cloud to construct a semantic map. (2) Localization module. We use CNN to extract semantic information from the image captured by camera. Ground pixels (e.g. landmarks, crosswalks, lane lines) are used to construct a local map using inverse perspective mapping (IPM), and aligned with the global map. Pole-like objects (e.g. trunks, streetlights, poles of traffic lights and billboards) on the semantic map are projected onto the image to create line matching. The vehicle's 6-DOF pose can be obtained by minimizing the global reprojection error.

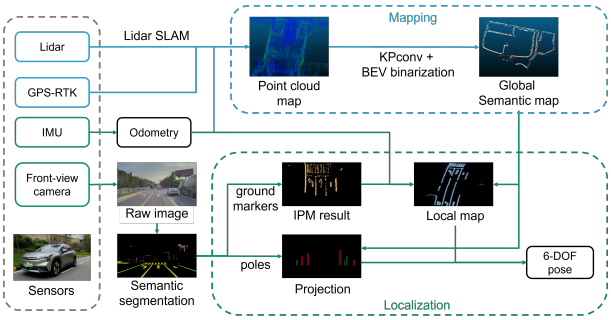


Fig. 1. Illustration of the system structure. The upper part illustrates the construction of global semantic map, and the lower part is the vehicle localization process is through the monocular camera.

A. Semantic Map

With the improved FAST-LIO2 algorithm [26] by fusing GPS-RTK information in the pose graph optimization module to ensure global location accuracy, the data collected by LiDAR is registered as a high-precision point cloud map. From which, we segment the pole-like semantic, and extract the two endpoints of each pole using Euclidean clustering and RANSAC linear fitting. Ground point clouds are extracted from pre-trained KPConv models [27] and plane-growth method. To accurately segment the ground marks, we project KPConv segmentation results onto the BEV plane, with the road surface point cloud's reflectivity treated as pixel values. Here, we employ the OTSU algorithm [28] to further binarize reflectivity values, enabling the isolation of clear lane markings and road surfaces. Finaly, We apply the mapping relationship between 3D point cloud and the BEV image to back-project the segmentation results into the 3D point cloud, enabling the 3D spatial semantic segmentation of the relevant elements, as depicted in Fig. 2.

B. Image Segmentation

The first step of localization is the semantic segmentation of images. We divide all the semantics into three categories: ground markers, poles, and background. A lightweight model, BiSeNetV2 [29], is selected to segment the necessary semantic features. To improve the computational efficiency of pixel projection, OpenCV [30] is used to extract all ground markings contours instead of using the entire semantic masks. This approach is favored because the position information of the contour can provide equivalent spatial constraints as whole marking pixels. Each pole instance is fitted as a straight line using the least squares method, which facilitates calculating the distance from the map point to the fitted pole. Fig. 3 illustrates a visualization of image segmentation in real traffic scenes.

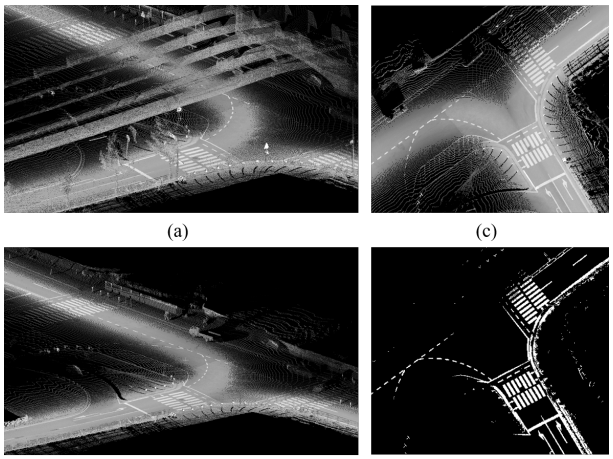
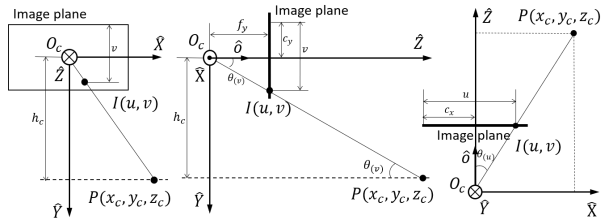


Fig. 2. Point cloud map generation and BEV segmentation. (a) shows the original point cloud map. (b) is the ground point cloud produced by LiDAR SLAM. (c) provides an example of BEV image, where each pixel corresponds to a 10 cm voxel. (d) displays the OTSU binarization results, which preserves high-contrast features on roads, including lane lines and markers.

Fig. 3. Image segmentation. (a) is the raw image captured by front-view camera. (b) is the semantic segmentation result. The orange and gray pixels indicate ground markers and poles, respectively. Green pixels highlight the outline of the ground markers and red pixels indicate the fitted straight lines of the poles. Note that short poles are discarded to avoid bringing in noise.

C. Inverse Perspective Transformation

After segmentation, the ground markers are transformed from the image plane to the vehicle coordinate system. This process can be executed through the IPM algorithm. Fig. 4 provides the conventional IPM model using physical parameters of the pinhole camera. The projection of point P in the ground plane to point I in the image plane is shown from three views.



(a) Front view (b) Side view (c) Top view

Fig. 4. The schematic of basic IPM model.

Based on the principles of projection for the pinhole camera, the translation from the point $[x_c, y_c, z_c]^T \in R^3$ to the pixel $[u, v]^T \in I^2$ can be described as Eq. 1.

$$Z \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \times \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (1)$$

where K is the intrinsic matrix of camera, f_x and f_y represent the focal length, s is the skewness factor, and (c_x, c_y) denotes the optical center of camera.

It is a reasonable assumption that a vehicle's wheels remain in contact with the ground while it is being driven on the road. This implies that the vertical height h between the ground and the optical center of the camera mounted on the vehicle remains constant. As shown in Fig. 4(b), the tilt angle between line $\overline{O_cP}$ and optical axis $\hat{o}$ is determined by the vertical coordinate v of point I. This angle is represented as $\theta(v)$, and the geometric relation can be expressed via the equation $\tan(\theta(v)) = \frac{z_c}{y_c}$. From Eq. 1, $\theta(v)$ can be derived as:

$$\theta(v) = \arctan(\frac{f_y}{v - c_y}) \quad (2)$$

Similarly, from the geometric relations illustrated in 4(c), we can deduce $\theta(u)$.

$$\theta(u) = \arctan(\frac{x_c}{z_c}) = \arctan(\frac{f_y(u - c_x) - s(v - c_y)}{f_x f_y}) \quad (3)$$

Eq. 2 and 3 define the fixed mapping relationship between the position of the point $P = [x_c, y_c, z_c]^T$ and $I = [u, v]^T$.

However, this basic IPM model is limited to the ideal situation where the ground surface is perfectly horizontal and the camera optical axis $\hat{o}$ is strictly parallel to the ground. In reality, vehicle motion induces camera rotation along the $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ axes, denoted as θ_{roll}, θ_{pitch} and θ_{yaw}. The enhanced IPM model with rotation angle compensation is shown in Fig. 5. The projection position P in the ground plane shifts to $P^{'} = [x_c^{'}, y_c^{'}, z_c^{'}]^T$.

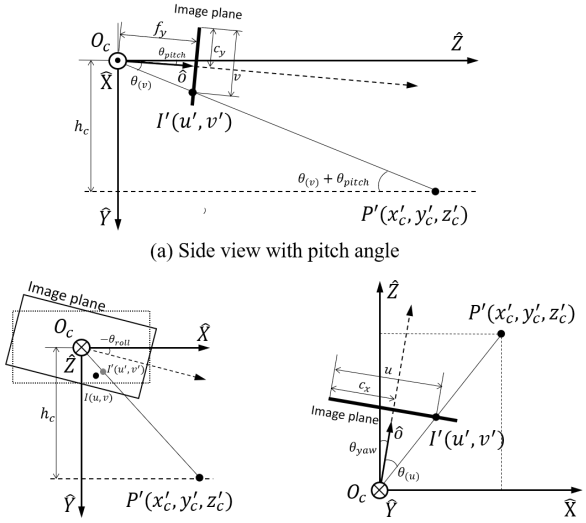
The effect of θ_{roll} can be visualized as rotating the image plane along the optical axis $\hat{o}$, as shown in Fig. 5(b). Therefore, the equivalent image mapped point can be obtained by rotating $I^{'} = [u^{'}, v^{'}]^T$ with an angle equal to $-\theta_{roll}$. The transformation from I to $I^{'}$ can be described as follows.

$$\begin{bmatrix} u^{'} \\ v^{'} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{roll}) & \sin(\theta_{roll}) \\ -\sin(\theta_{roll}) & \cos(\theta_{roll}) \end{bmatrix} \times \begin{bmatrix} u \\ v \end{bmatrix} \quad (4)$$

As shown in the side view from Fig. 5(a), θ_{pitch} causes an inclination with axis $\hat{Z}$ of camera coordinate system. This expressions of $z_c^{'}$ is deduced in Eq. 5.

$$\begin{aligned} z_c^{'} &= y_c^{'} \cdot \cot(\theta(v^{'}) + \theta_{pitch}) \\ &= h \cdot \frac{1 - \tan(\theta(v^{'}))\tan(\theta_{pitch})}{\tan(\theta(v^{'})) + \tan(\theta_{pitch})} \end{aligned} \quad (5)$$

Note that $z_c^{'}$ depends only on the variables $v^{'}$ of the pixel point $I^{'} = [u^{'}, v^{'}]^T$ and θ_{pitch}. We derive $x_c^{'}$ using



(a) Side view with pitch angle

(b) Front view with roll angle

(c) Top view with yaw angle

Fig. 5. The schematic of the enhanced IPM model with roll, pith, and yaw angles compensation.

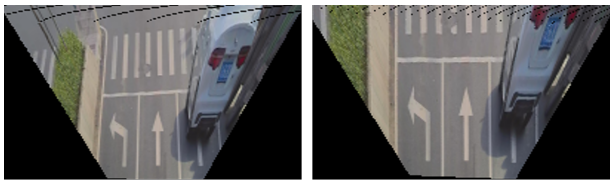
a proportional expression between $x_c^{'}$ and $z_c^{'}$ as illustrated in Fig. 5(c).

$$\begin{aligned} x_c^{'} &= z_c^{'} \cdot \tan(\theta(u^{'}) + \theta_{yaw}) \\ &= z_c^{'} \cdot \frac{\tan(\theta(u^{'})) + \tan(\theta_{yaw})}{1 - \tan(\theta(u^{'}))\tan(\theta_{yaw})} \end{aligned} \tag{6}$$

Furthermore, the installation of the camera results in an initial deviations $\theta_{roll,0}$, $\theta_{pitch,0}$, and $\theta_{yaw,0}$. These deviations are fixed and can be obtained through factory calibration. As a result, the compensation equations of the enhanced IPM can be derived from Eq. 4, 5 and 6.

$$\begin{cases} u^{'} = u \cdot \cos(\theta_{roll,0} + \theta_{roll}) + v \cdot \sin(\theta_{roll,0} + \theta_{roll}) \\ v^{'} = -u \cdot \sin(\theta_{roll,0} + \theta_{roll}) + v \cdot \cos(\theta_{roll,0} + \theta_{roll}) \\ \tan(\theta(u^{'})) = \dfrac{f_y(u^{'} - c_x) - s(v^{'} - c_y)}{f_x \cdot f_y} \\ z_c^{'} = h \cdot \dfrac{1 - \tan(\theta_{pitch,0} + \theta_{pitch})\dfrac{f_y}{v^{'} - c_y}}{\tan(\theta_{pitch,0} + \theta_{pitch}) + \dfrac{f_y}{v^{'} - c_y}} \\ x_c^{'} = z_c^{'} \cdot \dfrac{\tan(\theta_{yaw,0} + \theta_{yaw}) + \tan(\theta(u^{'}))}{1 - \tan(\theta_{yaw,0} + \theta_{yaw})\tan(\theta(u^{'}))} \\ y_c^{'} = h \end{cases} \tag{7}$$

In the real-world driving scenario, The deflection angles $(\theta_{roll}, \theta_{pitch}, \theta_{yaw})$ of the moving vehicle are computed via the integration of IMU data. Subsequently, the IPM model with rotation compensation is used to compute the projected coordinates of specific pixels and accurately restore the their 3D position in space. Fig. 6(a) shows the distorted BEV image of the vanilla IPM model. On the other hand, Fig. 6(b) presents the result of the enhanced IPM model with angle compensation. This illustrates the substantial distortion in the BEV image from even considerably small variations in angle during motion.



(a) (b)

Fig. 6. (a) is the BEV image transformed by the vanilla IPM. (b) is the enhanced IPM result with deflection angle compensation $roll, pitch, yaw = (0.8°, -1.9°, -1.2°)$.

D. Optimization Solver

Before optimizing pose from the k-th image frame, the vehicle state must be prepared, including the prior pose as well as the position of the ground markers and pole features. The iterative nonlinear optimization method is then used to match the current features with the global semantic map, resulting in the current pose of the vehicle.

Prior Pose. Vins-mono [13] propose an visual inertial odometry (VIO) method, which offers the vehicle's relative position and rotation. To improve the accuracy of the prior pose, the relative pose transformation between frame k and $k-1$ is computed, and integrated to the semantic localization result of the previous frame. This helps to minimize the cumulative error caused by IMU integration. The prior pose for k-th frame, denoted as T_k^*, is expressed in the following formula.

$$T_k^* = T_{k-1}(\hat{T}_{k-1})^{-1}\hat{T}_k \tag{8}$$

where T_{k-1} indicates the localization result of the previous frame obtained by our semantic localization algorithm. $\hat{T}_{k-1}$ and $\hat{T}_k$ are the VIO results for the corresponding frames respectively.

Ground Markers Representation. In the k-th image frame, we preserve the contoured pixels of ground markers. we designate the positions of m points in lane marking contours as $P_{lane,k}^I = \{\langle p_i, lane\rangle\}_{i=1}^m$, where $p_i = [u_i, v_i]^T$ is pixel coordinate. Therefore, the lane marking points in the vehicle coordinate system $\{\mathscr{V}\}$ can be represented as:

$$P_{lane,k}^V = T_C^V \mathscr{M}_{ipm}(P_{lane,k}^I) \tag{9}$$

where The matrix T_C^V is the external parameters from $\{\mathscr{C}\}$ to $\{\mathscr{V}\}$ and remains constant. $\mathscr{M}_{ipm}(\cdot)$ represents the IPM model.

Due to the limited field of view and segmentation noise in a single image, we accumulate several frames of lane data by employing a sliding window. We generate a local semantic map that composed of the ground features from the most recent c frames, while limiting its size to less than 50 meters. Subsequently, the local map can be transformed to the world coordinate system $\{\mathscr{W}\}$ with the prior pose $T_{V,k}^{W,*}$. We search

the nearby points $\bar{P}^W_{lane}$ by building a KD-tree of the global semantic map, as formulated in Eq. 10.

$$\bar{P}^W_{lane} \simeq P^W_{lane,k} = T^{W,*}_{V,k} \sum_{i=0}^{c} [T^V_C \mathscr{M}_{ipm}(p^I_{lane,k-i})] \quad (10)$$

Finally, we will only consider nearby points whose distance is less than a certain threshold (e.g. 1m). The loss is computed as follows:

$$\mathscr{L}_{lane} = \sum \| P^W_{lane,k} - \bar{P}^W_{lane} \|^2 + \sum D(P^W_{lane,k}, \bar{L}^W_{lane}) \quad (11)$$

where $\bar{L}^W_{lane}$ denotes the fitted line using the 5 nearest points in the semantic map, and $D(\cdot)$ is used to measure the distance from a point to the line.

Pole-like Objects Representation. When ground markings are not visible, relying solely on parallel lane lines fails to provide effective restraint in the forward direction of the vehicle. Pole-like objects (e.g. poles, lamp posts, tree trunks, etc.) are straight and perpendicular to the ground, which can be utilized to address this issue.

We use pairs of endpoints to denote n poles in the semantic map as $\bar{P}^W_{pole} = \{\langle p_{1i}, p_{2i} \rangle\}^n_{i=1}$, where each pole i is represented by two endpoints $p_{1i} = [x_i, y_i, z_{1i}]^T$ and $p_{2i} = [x_i, y_i, z_{2i}]^T$. Furthermore, the poles are projected into k-th image frame as $\bar{P}^I_{pole}$ with prior pose and projection function.

$$\bar{P}^I_{pole,i} = \frac{1}{z^C_i} K (T^V_C)^{-1} (T^{W,*}_{V,k})^{-1} \bar{P}^W_{pole,i} \quad (12)$$

where z^C_i is the z-coordinate of point i of the poles in camera coordinate $\{\mathscr{C}\}$.

For each endpoint projected onto the image, we find the closest straight line fitted by the segmentation result of pole-like objects. The distance from the endpoints $\bar{P}^I_{pole}$ to the corresponding fitted lines $L^I_{pole,i}$ is calculated as the residual.

$$\mathscr{L}_{pole} = \sum_{i=0}^{n} [D(\bar{P}^I_{pole,i}, L^I_{pole,i})] \quad (13)$$

Finally, the optimal global consistency matching is a nonlinear least squares problem, and the Ceres-Slover [31] with the Levenberg-Marquardt (LM) algorithm is employed to solve the pose of the vehicle.

$$T^W_{V,k} = \underset{T^{W,*}_{V,k}}{\arg\min} (\mathscr{L}_{lane} + \mathscr{L}_{pole}) \quad (14)$$

IV. EXPERIMENTAL EVALUATION

A. Datasets

The public KAIST dataset [32] provides a variety of sensor data acquired from complex urban environments. We select some typical scenes of autonomous driving (i.e. suburban, urban, and highway) from sequences 26, 38, and 39. Among them, the given point cloud data from LiDAR is used to construct a global semantic map, while the left camera and IMU measurements are used for localization.

In addition, we record a dataset covering a entire industrial park and several surrounding public roads, which add up to an approximately 6km-long road network in Chongqing, China. Fig. 7(a) shows the satellite map of the selected area. The dataset is collected by our self-driving cars equipped with a front-view camera, LiDAR, GPS-RTK, and IMU. We use LiDAR data to construct the point cloud map, and treat the GPS-RTK as the ground truth of localization.

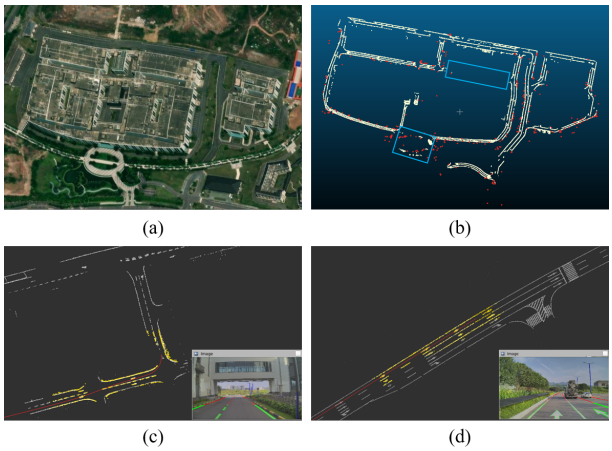


Fig. 7. An illumination of the qualitative results. (a) shown the satellite map of industrial park area of our self-recorded dataset. (b) is the global semantic map of industrial park. The ground markers are drawn in yellow and the endpoints of poles are drawn in red. The blue boxes indicating the areas without sufficient lane semantic information. (c) is a visual example of real-time pose optimization of in the scenes of industrial park. (d) public roads. In (c) and (d), the white points denote the lane marking map dynamically loaded with grid zones, and the yellow points indicate the current local lane markers map, which is projected to world coordinate frame by optimized pose. The green pixels in images indicate the lane marking feature used in the current frame during the pose estimation. Due to perspective noise, the pixels that are too far from the camera are discarded during the optimization process and marked as red. The blue lines represent the fitted pole-like features.

B. Visual localization accuracy

To evaluate the performance of our system, we compared it against other semantic localization algorithms, including CL+PA [33], PC semantic [34], and fusion SFM [9] on the KAIST dataset. Following the benchmark, we consider the localization accuracy on the x, y directions, as well as the heading (yaw) angle. We used the root mean squared error of absolute trajectory error (ATE) as the evaluation metrics, which includes RMSE Trans (m) and RMSE Rot (deg). Table I displays the results of the comparison of our algorithm to the baselines in various scenarios, indicating that our proposed algorithm achieved comparable localization accuracy to the baselines.

To further evaluate the effectiveness and generalizability of our system, we conduct an experiment base on our self-recorded dataset and compared our algorithm to the state-of-the-art visual localization toolbox Hloc [1]. We follow the standard evaluation method proposed in [35] for outdoor localization: $(0.25m, 2°)$, $(0.5m, 5°)$ and $(5m, 10°)$. Detailed results of the comparison with Hloc are presented in Table II. Notably, for the park dataset, Hloc requires an

TABLE I

RMSE RESULTS OF KAIST URBAN DATASET.

dataset Trans (m) Rot (deg)	Suburban	Urban	Highway
CL+PA [33]	0.604 0.882	0.580 1.080	1.806 0.935
PC Semantic [34]	1.798 0.464	0.893 0.91	2.494 0.907
fusion SFM [9]	0.573 0.510	0.54 0.68	1.964 0.853
Ours	**0.525** **0.507**	**0.472** **0.701**	**1.673** **0.822**

additional storage of about 4.5G of map data in colmap [36] format, while our system only needs to keep about 2M semantic point cloud maps. Despite the much smaller prior map, our proposed system achieves higher translation and rotation accuracy than the baseline. In addition, we observe that the overall localization accuracy in the industrial park are not as good as the public road due to the incomplete and scarce lane markings, as shown in Fig. 7(b). In contrast, Hloc can achieve higher accuracy than vacant public roads with the help of features such as dense buildings. Fig. 7(c) and (d) illustrate visual examples of our localization algorithm running in real time based on the park and the public road dataset.

TABLE II

PERFORMANCE COMPARISON OF PROPOSED ALGORITHM FOR SELF-RECORDED DATASET.

dataset m deg	Park 0.25/0.5/5.0 2/5/10		Public Road 0.25/0.5/5.0 2/5/10	
		Trans Rot		Trans Rot
Hloc [1]	30.47/63.49/95.51	1.25 0.59	26.33/56.32/93.58	1.43 0.71
Ours	38.38/77.23/97.72	0.52 0.63	32.86/80.16/98.21	0.49 0.65

Fig. 8 illustrates the distribution of vertical and horizontal position error in the vehicle frame and the heading angle error of our system. In comparison, the horizontal error distribution is more concentrated and closer to zero, which confirms that lane markings particularly the prevailing lane line feature, have stronger constraints on the horizontal direction. Poor accuracy in the vertical direction and heading angle may result from a lack of pole supervision in certain cases.

To evaluate the effectiveness of each proposed features in detail, we conduct an ablation study on our public road dataset. To ensure a fair comparison with VIO results, we also consider the relative pose error (RPE) metric using EVO [37], as shown in Table III. Our method eliminates the cumulative drift error of VIO by incorporating the global map, leading to a translational RMSE of 0.492 m, which is acceptable for autonomous driving tasks. Interestingly, both the lane markings and pole features of the semantic

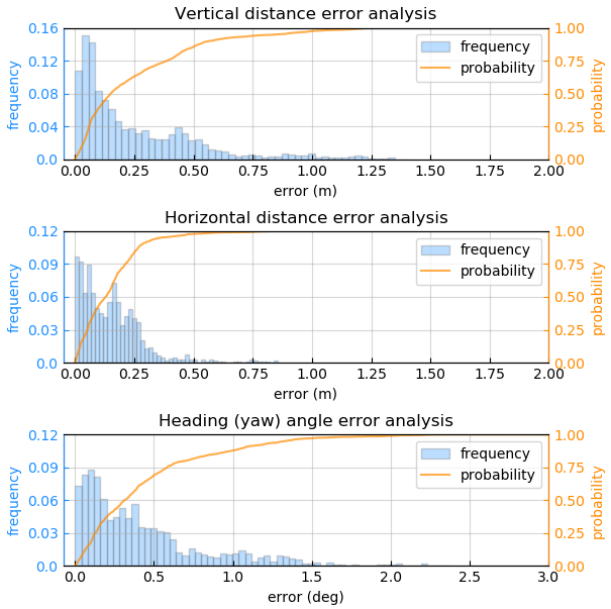


Fig. 8. The probability distribution plot of localization error in vertical and horizontal direction, and heading angle respectively.

map outperform the baseline in terms of RPE, indicating that visual features contribute to more efficient and robust localization accuracy.

TABLE III

VALIDATION RESULTS OF DIFFERENT METHODS ON PUBLIC ROAD DATASET.

VIO	lane markers	poles	ATE Trans m	RPE Trans m
✓			152.52	0.096
✓	✓		0.513	0.041
✓		✓	0.546	0.043
✓	✓	✓	0.492	0.038

"✓" means the corresponding feature is selected.

V. CONCLUSIONS

In this paper, we propose a visual localization system for autonomous vehicles based on stable visual semantic features, such as ground markers, lane lines, and poles. In our framework, we first construct the semantic map offline using LiDAR, and then optimize the matching of semantic features and corresponding information from the map to estimate the current position and direction of the vehicle. We validate our proposed system in a variety of challenging real-world traffic scenarios, and the results show that our proposed approach achieves better translation and rotation accuracy than the baseline. In future work, we consider integrating more kinds of low-cost sensors, such as GPS, to further extend the application of robust localization of autonomous vehicles in more complex traffic scenarios.

REFERENCES

[1] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[2] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, "Dxslam: A robust and efficient visual slam system with deep features," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4958–4965, IEEE, 2020.

[3] Y. Zhou, G. Wan, S. Hou, L. Yu, G. Wang, X. Rui, and S. Song, "Da4ad: End-to-end deep attention-based visual localization for autonomous driving," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pp. 271–289, Springer, 2020.

[4] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3247–3257, 2021.

[5] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[6] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "Dynaslam ii: Tightly-coupled multi-object tracking and slam," *IEEE robotics and automation letters*, vol. 6, no. 3, pp. 5191–5198, 2021.

[7] S. Liang, Y. Zhang, R. Tian, D. Zhu, L. Yang, and Z. Cao, "Semloc: Accurate and robust visual localization with semantic and structural constraints from prior maps," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4135–4141, IEEE, 2022.

[8] J. Jeong, Y. Cho, and A. Kim, "Road-slam: Road marking based slam with lane-level accuracy," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1736–1473, IEEE, 2017.

[9] K. Li, X. Zhang, L. Kun, and S. Zhang, "Vision global localization with semantic segmentation and interest feature points," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4581–4587, IEEE, 2020.

[10] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, "A light-weight semantic map for visual localization towards autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11248–11254, IEEE, 2021.

[11] Y. Lu, J. Huang, Y.-T. Chen, and B. Heisele, "Monocular localization in urban environments using road markings," in *2017 IEEE intelligent vehicles symposium (IV)*, pp. 468–474, IEEE, 2017.

[12] H. Wang, C. Xue, Y. Zhou, F. Wen, and H. Zhang, "Visual semantic localization based on hd map for autonomous vehicles in urban scenarios," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11255–11261, IEEE, 2021.

[13] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[15] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[16] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[17] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 449–454, IEEE, 2013.

[18] T. Wen, K. Jiang, B. Wijaya, H. Li, M. Yang, and D. Yang, "Tm 3 loc: Tightly-coupled monocular map matching for high precision vehicle localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 20268–20281, 2022.

[19] S. Kim, S. Kim, J. Seok, C. Ryu, D. Hwang, and K. Jo, "Road shape classification-based matching between lane detection and hd map for robust localization of autonomous cars," *IEEE Transactions on Intelligent Vehicles*, 2022.

[20] C. Zhang, H. Liu, H. Li, K. Guo, K. Yang, R. Cai, and Z. Li, "Dt-loc: Monocular visual localization on hd vector map using distance transforms of 2d semantic detections," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1531–1538, IEEE, 2021.

[21] H. Li, L. Pan, and J. Zhao, "Lidar-aided visual-inertial localization with semantic maps," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 910–916, IEEE, 2022.

[22] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014.

[23] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, IEEE, 2018.

[24] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3144–3150, IEEE, 2019.

[25] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[26] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[27] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

[28] None, "A threshold selection method from gray-level histograms," *Systems Man & Cybernetics IEEE Transactions on*, vol. 9, no. 1, pp. 62–66, 1979.

[29] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, vol. 129, pp. 3051–3068, 2021.

[30] G. Bradski, "The opencv library," *dr dobbs journal of software tools*, 2000.

[31] S. Agarwal, K. Mierle, *et al.*, "Ceres solver," 2012.

[32] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.

[33] Z. Liao, J. Shi, X. Qi, X. Zhang, W. Wang, Y. He, X. Liu, and R. Wei, "Coarse-to-fine visual localization using semantic compact map," in *2020 3rd International Conference on Control and Robots (ICCR)*, pp. 30–37, IEEE, 2020.

[34] E. Stenborg, C. Toft, and L. Hammarstrand, "Long-term visual localization using semantically segmented images," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6484–6490, IEEE, 2018.

[35] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8601–8610, 2018.

[36] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] M. Grupp, "evo: Python package for the evaluation of odometry and slam.." `https://github.com/MichaelGrupp/evo`, 2017.