Monocular Mapping and Localization of Urban Road Scenes Based on Parameterized Semantic Representation

Fengsheng Ding[1,2], Xinchun Ji[2], Dongyan Wei[1,2], Jingyu Zhang[3], Kai Li[2] and Hong Yuan[2]

[1] *University of Chinese Academy of Sciences, No.19A Yuquan Road, Shijingshan District, Beijing 100049, China*
[2] *Aerospace Information Research Institute Chinese Academy of Sciences, No.9 Dengzhuang South Road, Haidian District, Beijing 100094, China*
[3] *Shanghai Astronomical Observatory, Chinese Academy of Sciences, No.80 Nandan Road, Shanghai 200030, China*

Abstract

The semantic map plays an essential role in unmanned tasks with limited resources for urban road scenes, especially for precise localization in the GNSS-blocked area. With the advantages of low cost, rich information acquisition and wide depth range, monocular cameras have received much attention and research in semantic mapping. Most of the current monocular mapping methods use dense point clouds to represent semantic features, which is insufficient of the compactness in feature representation, faces the pressure from storage and computational resources for common autonomous vehicles, and lacks multi-dimensional scene elements. Based on a low-cost monocular camera, this paper proposes a parametric semantic mapping algorithm for multi-dimensional features in road scenes. Meanwhile, a semantic map-based monocular camera matching localization algorithm is proposed. The experiment results on KAIST Urban Dataset show that the root mean square error of localization on x and y is about 0.3m, and the map size is compressed to 8.8 kB/km. Achieve map lightweight while meeting the positioning accuracy requirements for autonomous vehicle tasks under urban road environments.

Keywords

semantic mapping, parameterized features, map-based localization, autonomous vehicles

1. Introduction

Recently, autonomous vehicles (AVs) based services such as Robo-taxi and autonomous valet parking (AVP) have been developing rapidly, in which self-localization is a primary component. High-definition map (HD map) [1] and dense point cloud map [2] can achieve centimeter-level positioning for these services, but the resource required to build, update and use prevent them from being widely used in AVs. Benefit from the richness, long-term existence and insensitivity to viewpoint, lighting and weather conditions of semantic features in the urban road scenes [3], semantic map can help achieve accurate localization in GNSS-block area and realize the lightweight of the map to some extent, friendly to the storage and computing resources of consumer-grade vehicles, which is conducive to the promotion of use. Crowdsourced mapping in large urban scale scenarios is an efficient way to build semantic map that can extend the mapping area and provide map updates at any time through local semantic mapping on consumer-grade vehicles [4].

A monocular camera is a low-cost sensor that can be deployed on consumer-grade vehicles rapidly and widely, offering the advantage of rich information acquisition and a wide depth range for use in local semantic mapping [5]. However, in the work of monocular semantic mapping for localization tasks, semantic features are currently still presented in the map as dense point clouds, while point cloud stitching is one of the necessary procedures [5][6]. These methods' first disadvantage is that it pressures storage and computational resources. On the other hand, the compactness of the feature representation needs to be improved, and blurred feature edges are the most direct manifestation, which makes map-based localization face higher uncertainty. Meanwhile, many studies now use various semantic features in maps for monocular localization [7-10], most of the current mapping

Proceedings of the Work-in-Progress Papers at the 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN-WiP 2023), September 25 - 28, 2023, Nuremberg, Germany

dingfengsheng21@mails.ucas.ac.cn (Fengsheng Ding); jixc@aircas.ac.cn (Xinchun Ji); weidy@aircas.ac.cn (Dongyan Wei); zjy8712@126.com (Jingyu Zhang); kukay@163.com (Kai Li)

studies are confined to single or partial features on the road surface, which limits the constraint capability provided during localization. For example, Zhou et al. [11] and Cheng et al. [12] investigated different ways of feature representation in maps, such as deep key points and partial parameterization, with applicability limitations and lack of reconstruction of semantics at the partial road because of reconstruction difficulty. Wen et al. [13] reconstruct the roadside poles but neglect the road surface features. Mapping with multidimensional semantic features in urban road scenarios needs to be addressed further.

To address the above problems, we propose a novel monocular semantic mapping method for urban road scenes: multidimensional semantic mapping based on parameterized features. The proposed monocular mapping system can be performed on consumer-grade vehicles focused on road markings and spatial pole features. The contributions of this paper are as follows:

- A method for parametrically characterizing the semantics of urban road scenes is proposed, and feature representation satisfies a stronger compactness while realizing the lightweight of the map.
- A monocular mapping method that minimizes the semantic observation error is proposed to realize the construction of parametric semantic maps with multidimensional features.
- A localization verification system based on the constructed semantic maps to evaluate the accuracy and usability of the maps.

The paper is organized as follows: Section II presents the system flowchart and a brief overview of the monocular mapping and localization algorithms; Section III introduces parametric characterization and mapping algorithm; Section IV is an experimental validation of the proposed method; Section V summaries the study and discusses future work.

2. System overview

The system flowchart of the proposed algorithm is shown in Figure 1, including two parts: monocular mapping and localization verification. The monocular mapping includes three modules of feature extraction and preprocessing, local reconstruction and global mapping. Semantic features are first extracted and preprocessed; the local reconstruction module implements feature parameter initialization, 2D association and optimal recovery of spatial location. Localization verification implements a map matching positioning system for map evaluation.

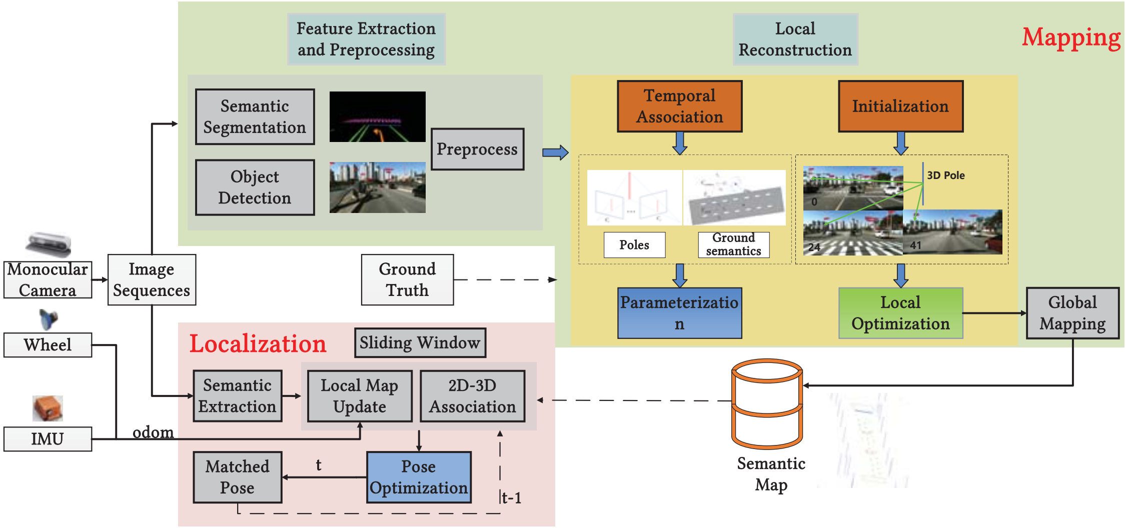


Figure 1: System pipeline for the algorithm. The whole system includes two parts: offline mapping and online localization. The input is provided by a monocular camera, IMU and wheel speedometer, and the output is a lightweight environmental semantic map and localization results. Some schematic diagrams are provided in the figure, including examples of environmental semantic feature extraction results and feature initialization and association.

3. Methods

3.1. Feature extraction and preprocessing

The semantic features are divided into two categories: road surface features and roadside pole features. They are extracted through semantic segmentation and object detection network, respectively. DeepLabv3+ [14] segments the image at the pixel level for road surface features for road surface features. We segment six semantics on road surfaces: slowdowns; arrows; crosswalks; road lines (both solid and broken lines); stop lines; numbers; texts. Sem-LSD [15] is used to detect the poles in the image. Sem-LSD encodes advanced semantic information, which is more robust in matching associations in complex urban environments. Sem-LSD detects a pole as a Bounding Box, which is vertical and 2 to 3 pixels wide in the image, and its centerline is used to represent the poles.

Due to the generalization ability of the model, there may be false detections in the feature extraction. We use post-processing based on the results of geometric attributes to reduce the impact of segmentation errors on the mapping accuracy.

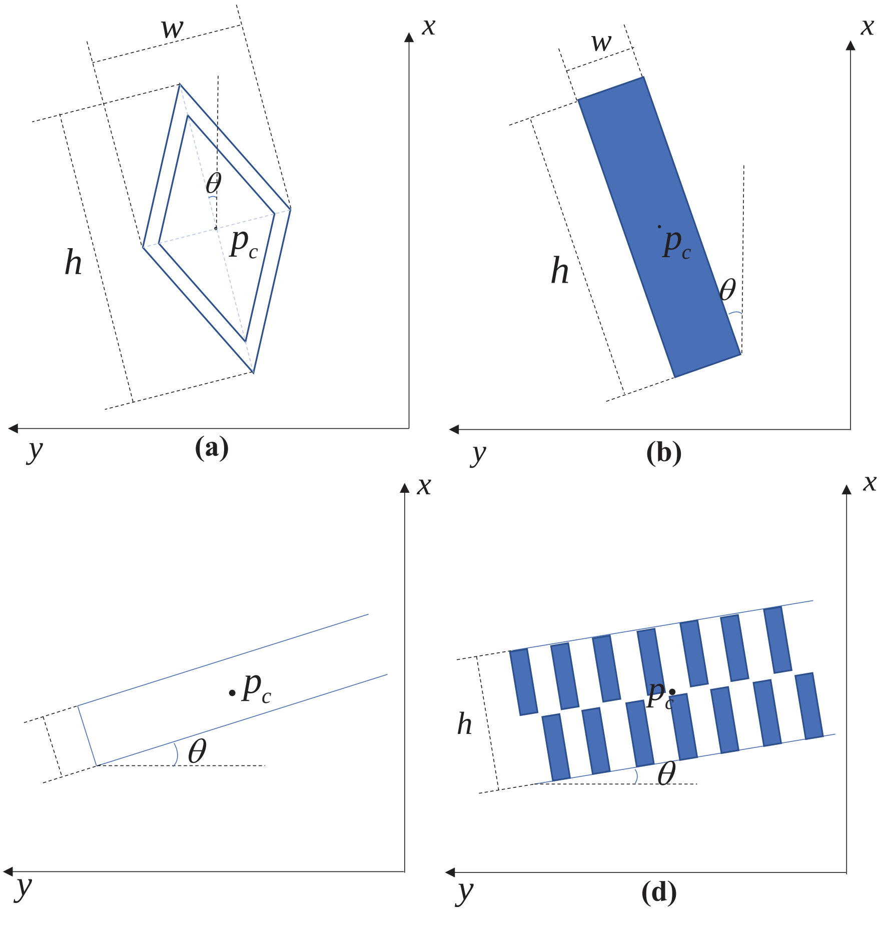


Figure 2: The parameterizations of the semantic features on the road surface are presented on x-y plane under the vehicle coordinate system. (a) and (b) represent the parameterization of road features 1, which include two categories, slowdown and broken lines; (c) and (d) represent the parameterization of road features 2, including two categories, stoplines and crosswalks.

3.2. Feature parameterization

We classify the extracted semantics into three categories based on their shape and geometric properties:

- Road features 1 (slowdowns and broken lines).
- Road features 2 (stop lines and crosswalks).
- Pole features (poles).

The road lines obtained by semantic segmentation are divided into broken lines and solid lines according to length, only the former being used. Next, the semantics of the different priors were parameterized separately.

Road features 1 It can be observed entirely for slowdowns and broken lines so that the parameterization can represent the whole shape. As shown in Figure 2(a) and (b), the minimum enclosing box is first fitted according to the feature point cloud in the XOY plane. Then the centroid $p_c(x_c, y_c, 0)$, the size (w,h), and the yaw angle θ of the box are estimated in the local vehicle frame. In summary, six parameters are used to represent the road features 1 under local vehicle frame, denoted as:

$$\mathbf{P} = [x_c, y_c, \theta, w, h] \tag{1}$$

Together with the local pose $\mathbf{T} \in SE(3)$ stored in the map.

Road features 2 The lateral range is vast for stop lines and crosswalks, which generally exist at road junctions. The full view is usually unavailable due to the limited camera view and vehicle occlusion. Therefore, only the longitudinal attributes are usually stored as map elements, as shown in Figure 2(c) and (d). The width h, centroid $p_c = (x_c,\ *,\ 0)$ and yaw angle θ of the feature in the longitudinal direction are first calculated based on the feature point cloud in the XOY plane. As a result, the properties of road features 2 are recorded using three parameters, denoted as:

$$\mathbf{P}_2 = [x_c,\ \theta,\ h] \tag{2}$$

also include local pose $\mathbf{T}$.

Pole features Considering the computational unfriendliness of the orthogonal representation, we use Plücker coordinates to represent a straight line in the space for poles [16], as shown in Figure 3(a). For a straight pole line $\mathcal{L}$ in the camera frame, its Plücker coordinate is

$$\mathcal{L} = \left(\mathbf{n}^\top, \mathbf{v}^\top\right)^\top,\ \mathbf{n}^\top \mathbf{v} = 0 \tag{3}$$

where $\mathbf{n}$ is the normal on the plane formed by the pole and the origin, and $\mathbf{v}$ is the direction vector of the pole. The distance from the pole to the camera's optical center can be calculated from the Plücker coordinates as $d = \|\mathbf{n}\| / \|\mathbf{v}\|$. For spatial poles, there is a geometric prior perpendicular to the XOZ plane in the camera frame, so the original Plücker coordinates can be expressed in a simple way. First, the direction vector can be simplified to $\mathbf{v} = (0,1,0)$, and the normal vector in Plücker coordinates is obtained by:

$$\begin{aligned} &\mathbf{n} = d(\cos\theta,\ 0,\ \sin\theta) \\ &\mathcal{L} = \left(\mathbf{n}^\top, \mathbf{v}^\top\right)^\top = (d\cos\theta,\ 0,\ d\sin\theta,\ 0,\ 1,\ 0) \end{aligned} \tag{4}$$

Therefore, the parameters

$$\mathbf{P}_3 = [d,\ \theta] \tag{5}$$

are used to represent the pole in the local camera frame, combined with the local pose to represent poles in the map.

3.3. Temporal association and initialization

The reconstruction of each 2D semantic feature needs three processes: temporal association, initialization, and local optimization. The temporal association establishes the associations of each semantic feature between image sequences. The initialization completes a transformation of the feature from 2D to 3D space and obtains the initial values for parameterization and local reconstruction. Pole features perform the 2D-2D temporal associations first, then initialized by multi-view reconstruction. On the contrary, the road features can obtain rough world coordinates by inverse perspective mapping (IPM) [17], so they can be initialized directly.

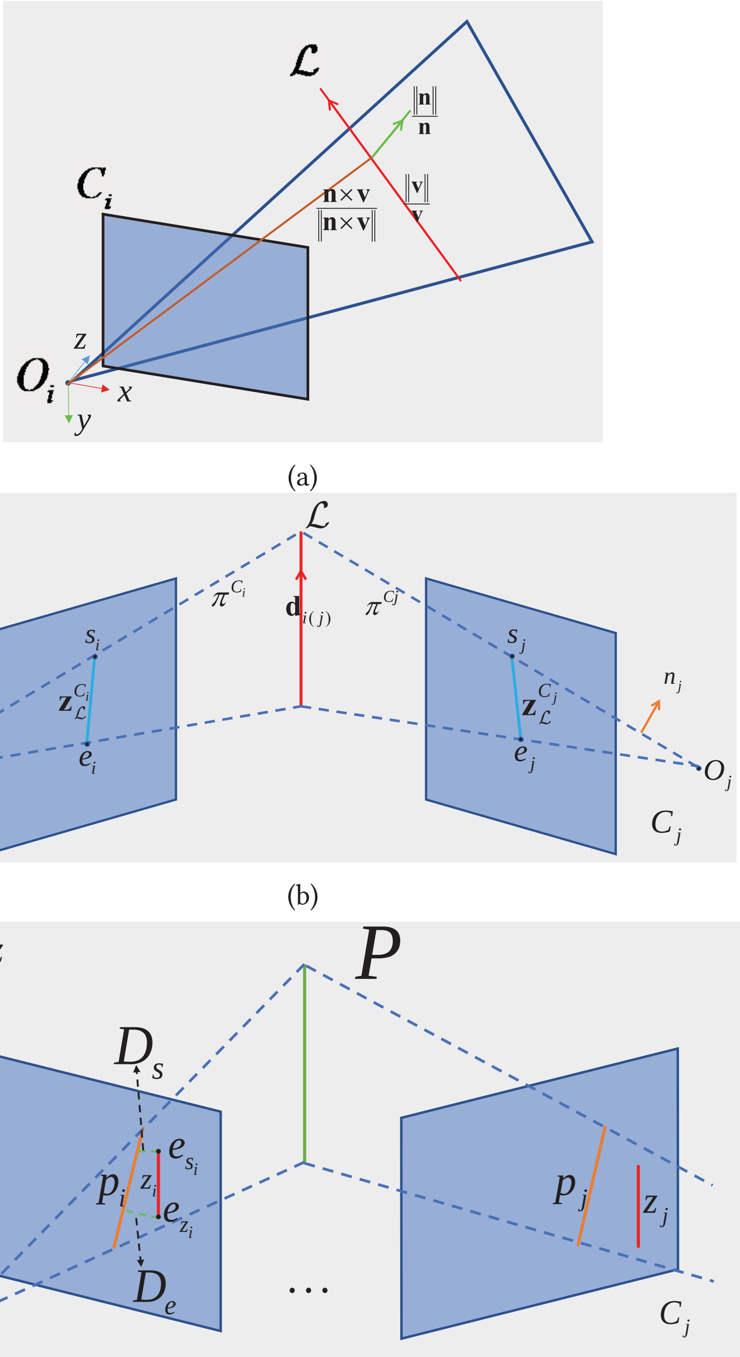
3.3.1. Initialization

Initialization obtains the initial values of the parameters in 3D space for each observed 2D feature. **Road surface features** include road feature 1 and 2. The best reference frame is selected in the observation sequence, then the point cloud of the semantic features in the vehicle frame is obtained by IPM, as shown in Figure 4. With the assumption that the ground in front of the vehicle is a plane:

$$\mathbf{p} = \mathbf{MP} = \mathbf{K}[\mathbf{R}|\ \mathbf{t}]\mathbf{P} \tag{6}$$

$\mathbf{p} = KT_c^v\mathbf{P}_w$

where $\mathbf{p} = (u, v)$ and $\mathbf{P} = (X_v, Y_v, 0, 1)$ represent corresponding point in the image and 3D space, $\mathbf{K}$ is the camera intrinsic matrix, $\mathbf{R}$ and $\mathbf{t}$ form the camera extrinsic matrix with respect to the vehicle's center, and $\mathbf{M}$ represents the projection matrix from the vehicle frame to the image plane. After the inverse projection, all pixels of each semantic feature are mapped to the corresponding 3D spatial location, and a dense point cloud representation of the semantic feature in the local vehicle frame is obtained. Then the initial values of the feature parameters are obtained from the dense point cloud as described in Section 3.2.



(a)

(b)

(c)

Figure 3: Parametric representation of pole feature. (a) Plücker coordinates of pole lines in camera frame; (b) Parameterization initialization of poles; (c) Reprojection error of pole feature.

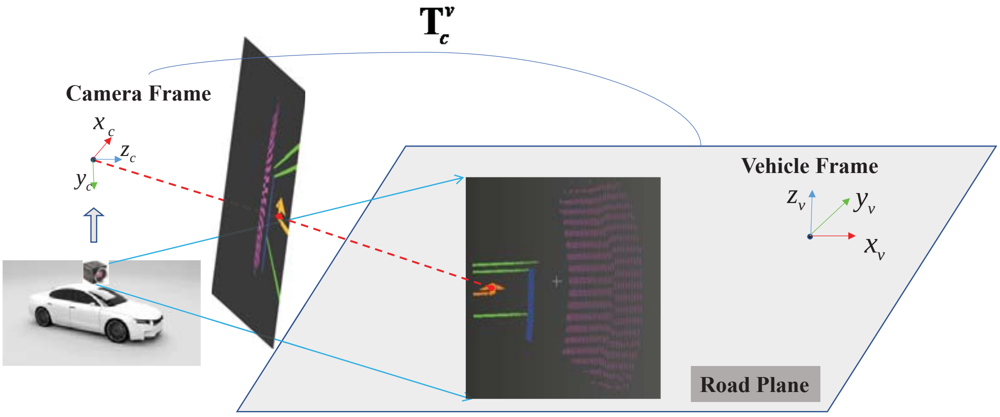


Figure 4: The principle of inverse perspective transformation. The image segmented by the camera is associated with the corresponding position on the ground in front of the vehicle, and the red dot indicates the correspondence of the road arrow. $\mathbf{T}_c^v$ is the extrinsic parameter from the camera to the vehicle coordinate system. The subscript c represents the camera coordinate system, and v represents the vehicle coordinate system.

Pole feature Limited by the working mechanism of the monocular camera, we cannot recover the spatial position of the pole from a single frame. Two suitable reference frames in the observation sequence are selected for initialization, and the initial values of the pole parameters in local camera frame are obtained by calculating the intersection line of two planes. As shown in Figure 3(b), the pole is observed in the reference $\mathbf{z}_{\mathcal{L}}^{C_i}$ and $\mathbf{z}_{\mathcal{L}}^{C_j}$. Using the reference frame C_i as the local camera frame, the endpoints of the observation $\mathbf{z}_{\mathcal{L}}^{C_i}$ on image are $\mathbf{s}_i = \left[u_S, v_S, 1\right]^{\mathrm{T}}$ and $\mathbf{e}_i = \left[u_e, v_e, 1\right]^{\mathrm{T}}$. With the coordinate origin $O = (x_o, y_o, z_o)^{\mathrm{T}}$, we can determine a plane $\pi^{C_i} = \left[\pi_x, \pi_y, \pi_z, \pi_w\right]^{\mathrm{T}}$:

$$\pi_x\left(x - x_o\right) + \pi_y\left(y - y_o\right) + \pi_z\left(z - z_o\right) = 0 \tag{7}$$

$$\begin{bmatrix}\pi_x \\ \pi_y \\ \pi_z\end{bmatrix} = \left[\mathbf{s}_i\right]\times\mathbf{e}_i, \quad \pi_w = \pi_x x_o + \pi_y y_o + \pi_z z_o \tag{8}$$

The initial values of the pole feature are obtained by intersecting the plane under reference frames π^{C_i} and π^{C_j} :

$$\mathbf{L}^* = \begin{bmatrix}[\mathbf{d}]_\times & \mathbf{v} \\ -\mathbf{v}^\top & 0\end{bmatrix} = \pi^{C_i}(\pi^{C_j})^{\mathrm{T}} - \pi^{C_j}(\pi^{C_i})^{\mathrm{T}} \in \mathbb{R}^{4\times 4} \tag{9}$$

The $\mathbf{v}$ and $\mathbf{d}$ obtained from $\mathbf{L}^*$ are used as the initial values of the parameters of pole features.

3.3.2. Temporal association

Temporal association establishes 2D correspondences of the same feature under different observations. We consider different schemes for the two types of semantics.

Road surface features Semantic point clouds of road features under the local frame are obtained by IPM and converted to global coordinates with reference pose. The same feature observed at different moments locates at the approximate position under the world frame, so the feature association between different moments is achieved directly by comparing the distance in the world frame. For slowdowns, stop lines and crosswalks, which occur infrequently and have independent signatures in a short period, the temporal association can be accomplished by Euclidean distance discrimination in the global frame. For broken lines, which appear frequently and are close to each other in a multi-lane environment with high confusion. Firstly, all broken lines point clouds under the global frame are segmented by DBSCAN clustering algorithm [18] to get different broken line instances. Next, the geometric and spatial constraints are combined to determine whether there is a possibility of establishing an association between each 2D observed broken lines and all segmented instances using multiple conditions, including angle, length and overlap rate.

Pole feature We use classical object tracking algorithm SORT (Simple Online and Realtime Tracking) [19] in computer vision to track the pole in temporal order for 2D associations.

3.4. Local optimization

Known the initial parameters and 2D observations of features, the optimal parameters are obtained by minimizing the reprojection error between feature in the image and spatial, which is used as the final parameters in the map. The cost function $\mathbf{P}_m$ of the nonlinear optimization for the feature parameter in the corresponding local frame is:

$$\begin{aligned}\hat{\mathbf{P}}_m = \sum_{j=1}^{N_z} \arg\min_{P_i} (r_m^{\text{dis}})^T(\mathbf{z}_j^m, \mathbf{P}_m)(\boldsymbol{\Sigma}_m^{\text{dis}})^{-1} r_m^{\text{dis}}(\mathbf{z}_j^m, \mathbf{P}_m) \\ + (r_m^{\text{ang}})^T(\mathbf{z}_j^m, \mathbf{P}_m)(\boldsymbol{\Sigma}_m^{\text{ang}})^{-1} r_m^{\text{ang}}(\mathbf{z}_j^m, \mathbf{P}_m)\end{aligned} \tag{10}$$

where m is the semantic categories with the value 1, 2 or 3. $\mathbf{P}_m$ represents the parameters of the semantic features. $\mathbf{z}_j^m$ is the jth 2D observation of feature m, N_z is the number of 2D observations, $\boldsymbol{\Sigma}_m$ is the covariance. $r_m^{\text{dis}}(\mathbf{z}_j^m, \mathbf{P}_m)$ is the distance residual, while $r_m^{\text{ang}}(\mathbf{z}_j^m, \mathbf{P}_m)$ is the angular residual.

For road features 1, the distance error between the feature and the observation is designed as the distance of the four vertices of the feature in the image plane. For road features 2, limited by the camera view, only the offset on the y-direction of the image is used:

$$\begin{aligned} r_i^{\text{dis}}(\mathbf{z}_j, \mathbf{P}) &= D_i(\mathbf{z}_j, p_j), \quad i = 1, 2 \\ p^j &= \mathbf{K}\mathbf{T}_{ref}^i \mathbf{P}\end{aligned} \tag{11}$$

where $\mathbf{P}$ is the parameter of the road features, $D_1(\mathbf{z}_j, p_j)$ is the distance between the observation of j frame and the four vertices of the projected image plane, $D_2(\mathbf{z}_j, p_j)$ is the coordinate difference and width difference between the observation of the j frame and the projection of the center point y in the vertical direction of the image plane. $\mathbf{T}_{ref}^i$ is the relative pose of the reference frame and observation frame, $\mathbf{K}$ is the camera intrinsic matrix.

For roadside pole features, the feature is projected to the observed camera frame firstly. The distance error function is designed as the distance from the endpoint of the 2D observation line in the normal plane to the projection line, as shown in Figure 3(c).

$$\begin{aligned} r_3^{dis}(\mathbf{z}_i, \mathbf{P}) &= D_s(s_{\mathbf{z}_i}, p_i) + D_e(e_{\mathbf{z}_i}, p_i) \\ p_i &= \mathbf{K}'\mathbf{T}_v^c \mathbf{T}_{ref}^i \mathbf{T}_c^v \mathbf{P}\end{aligned} \tag{12}$$

where $D_s(s_{\mathbf{z}^j}, p^j)$ and $D_e(e_{\mathbf{z}^j}, p^j)$ denote the distances from 2D line endpoints to 3D line, respectively.

$$\begin{aligned} D_s(s_{\mathbf{z}_i}, p_i) &= \frac{s_{\mathbf{z}_i}\mathbf{l}_{p_i}}{s_3\sqrt{l_1^2 + l_2^2}} \in \mathbb{R}^1 \\ D_e(e_{\mathbf{z}^i}, p^i) &= \frac{e_{\mathbf{z}_i}\mathbf{l}_{p_i}}{e_3\sqrt{l_1^2 + l_2^2}} \in \mathbb{R}^1 \end{aligned} \tag{13}$$

$s_{\mathbf{z}_i} = (s_1, s_2, s_3)$ and $e_{\mathbf{z}_i} = (e_1, e_2, e_3)$ denote the two endpoints of the line observations, $\mathbf{l}_{p_j} = (l_1, l_2, l_3)$ denotes the coordinates of the line features in the normal image plane, $\mathbf{T}_v^c$ denotes the transformation matrix from the vehicle to the camera frame, and $\mathbf{K}'$ is identity matrix here.

After minimizing the reprojection error of the semantic features, the optimized parameters are considered as the final feature parameters, which are stored in the map and used as matching primitives for localization.

3.5. Localization verification

Based on the a priori semantic point cloud map, we evaluate the accuracy and usability of the map by matching localization. Considering the single-frame semantic sparsity and perspective distortion filtering some features, to ensure the richness of semantic features and robustly optimize the pose, we use sliding window strategy for localization, the initial pose $\mathbf{T}_t$ is obtained by combining the estimated pose at time $t-1$ with the odometry. Then the local map is obtained and matching pairs are established by the projection distance according to (11) and (12). According to the parameterization of different features, 2D-3D matching pairs are established using Euclidean distance and point-to-line distance under the vehicle frame. Based on the matched pairs, the current frame's pose to be optimized can be recurred to the remaining frames in the window, then minimize the error between the features in the map and in each window frame to obtain the optimal poses $\hat{\mathbf{T}}_t$.

$$\mathbf{T}_t = \hat{\mathbf{T}}_t \mathbf{T}_{t-1}^t \tag{14}$$

$$\hat{\mathbf{T}}_t = \arg\min \sum_{i=t-N_w}^{t} r((\mathbf{T}_t \mathbf{T}_i^t)^{-1}, M_i) \tag{15}$$

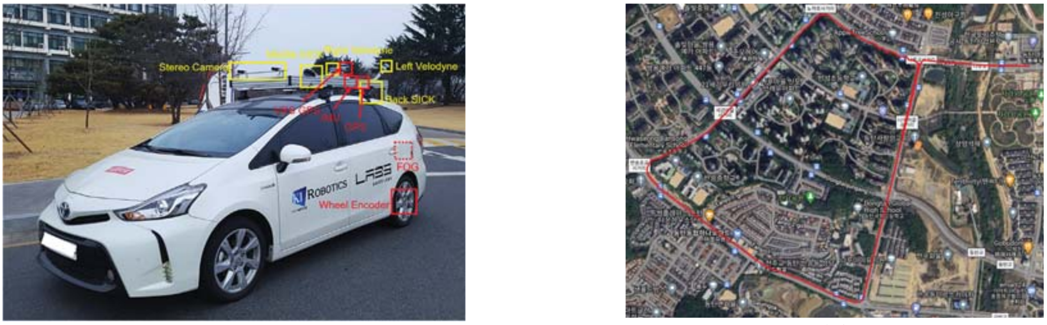
$$M_i = \left\{ \left(S_j^c, s_j^c \right) \right\} \quad j = 1, \cdots, N_i, c = 1, \cdots, N_c \tag{16}$$

where N_c, N_w and N_i are the number of semantic categories, the sliding window size, and 2D-3D matching pairs in window i. Residual function r refers to Section 3.4. S_j^c and s_j^c are the map primitives and 2D features in the matching pairs, and $\mathbf{T}_{t-1}^t$ is the pose provided by odometry.

4. Experimental results

4.1. Setup

The KAIST Urban Dataset [20] is a publicly available dataset of urban road scenarios, and we conduct experiments on urban26 sequence with a total length of 3.98 km. In this paper, the images acquired by the left camera and ground truth are used for mapping. The semantic mapping process is implemented offline, and localization is done in real-time.



(a) (b)

Figure 5: KAIST Complex Urban dataset. (a) is the collection car of the data set, (b) is the route of the data used in the experiment in Google Map.

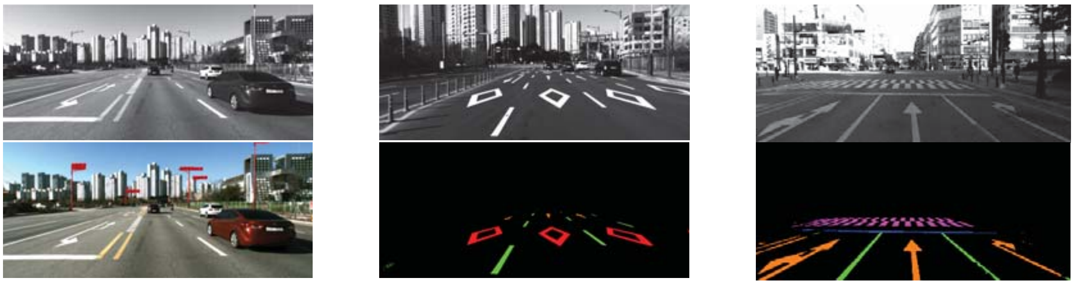


Figure 6: Feature extraction. From left to right, the feature extraction results are shown at frames 57, 303 and 356, including rod features, slowdowns and broken lines of the pavement and crosswalks, arrows and stop lines of the pavement, respectively.

4.2. Semantic Mapping

Extracting 2D semantic features from raw images is the first step of mapping. DeepLabv3+ and Sem-LSD are used to extract road semantics and roadside pole semantics, respectively. Due to the lack of annotated data, DeepLabv3+ is first pre-trained on Cityscapes, and then trained on SeRM dataset [6] to fine-tune the network to obtain an accurate semantic inference model based on the MMSegmentation [21] platform. The labels trained using DeepLabv3+ are slowdowns, arrows, crosswalks, road lines, stop lines, numbers and texts. Considering the accuracy and other issues, only slowdowns, crosswalks, road lines and stop lines are used for mapping. For roadside pole features, use the model labeled and trained on KAIST urban dataset to detect poles.

Considering the influence of perspective effect, the distant ground is more distorted in the image, so we preserve semantic features in the ROI region from 2 to 20m relative to the vehicle center. When parameterization, considering that the roads are generally in a straight line when slowdowns, crosswalks and stop lines appear, parameter θ of these three types of features is set to 0° in this paper for the convenience of reconstruction. While Broken lines, due to the large number and common at curves, this parameter cannot be neglected and has some influence on localization, so the angle is estimated during initialization. In the temporal association, we expand the width of the target box by 20 pixels before tracking.

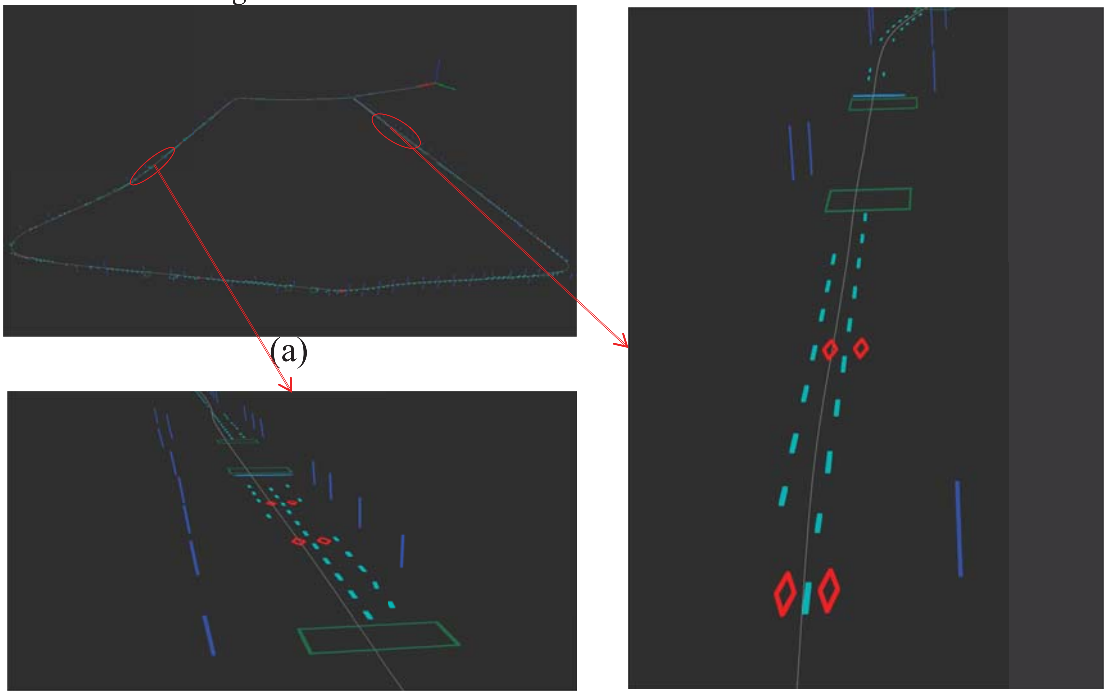


Figure 7: Global semantic mapping result on KAIST urban26 sequence. The categories of semantic elements are distinguished by colors, e.g., slowdown signs are red and poles are blue. (a) is the global semantic map; (b) and (c) are local enlarged images of the global semantic map.

The final semantic map is shown in Figure 7, the size of the map built on a road of length 3.98km is only 35.2kB. The feature parameterization is completed with a lightweight map, while there are five semantic categories in the map, including slowdowns, crosswalks, stop lines, broken lines and poles, with numbers of 36, 23, 21, 506, and 153, respectively.

4.3. Localization Verification

Two matching methods are used to verify the accuracy and usability of the maps: using ground features with slowdowns and broke lines (match-G: Ground); using multi-dimensional features with round semantic features and poles (match-GP: Ground and Pole). The accuracy of the map is evaluated by the error between matching optimized pose and ground truth. We focused on the localization error in x, y and yaw angle. At the same time, to verify the efficiency and accuracy of our map, we compare it with the semantic point cloud ICP localization method used in [5]. [5] uses the semantic point cloud map and the local point cloud generated by IPM for ICP matching, and combined with the odometry to achieve 6 DOF localization. For the convenience of verification, based on the fact that the change of roll and pitch angle is much smaller than the yaw angle when driving in the road scene, we evaluate accuracy according to the error between matching results and ground truth on the translation and yaw angle.

Figure 8 shows localization errors distribution of match-GP with trajectory. From the whole map, we can see that the localization error is about 0.3m, while some positioning points have larger errors, usually at intersections that lack road semantic features. Figure 9 shows the results of match-GP in the semantic map, we can find localization results always keep a small jump around the real trajectory.

With the blessing of multidimensional features, our map can also provide pole features constraints in addition to ground semantics. As can be seen in Table 1, compared with match-G, the positioning error of match-GP on x and y is about 0.3m, and the yaw angle error is less than 0.5°. After adding pole constraints, there are 5% and 12% improvements in x and y, and a small improvement in yaw. As shown in the Figure 10, the positioning frequency is greatly increased while the positioning accuracy is improved, which is concentrated in a lower region. The number of positioning points in match-GP is 3.4 times that of match-G, which provides smoother positioning results. In summary, we parameterize the multidimensional features in the environment to build a lightweight map. We conducted experiments using point cloud map localization using the method in [5]. Compared to that which can achieve ten-centimeter-level localization accuracy for autonomous vehicle services, the demand for network bandwidth, storage, and computing resources is remarkably relaxed. The real-time performance of the algorithm is analyzed, and compared with many time-consuming operations on point clouds in [5], our method is faster and less resource-dependent. At the same time, it can still achieve decimeter-level positioning, which satisfies the localization accuracy requirements for autonomous vehicle services (e.g., Robo-taxi, unmanned delivery vehicles, etc.) in urban road scenes.

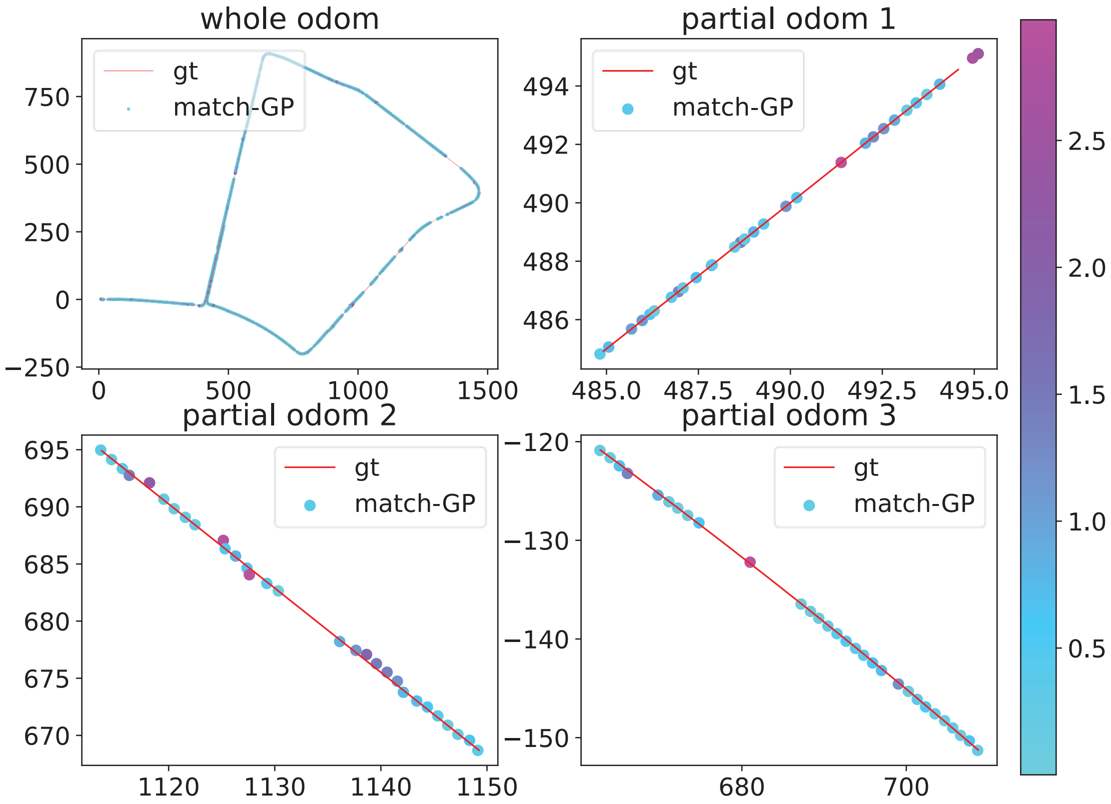


Figure 8: The match-GP localization error dispersion diagram. The top left image shows the difference between match localization results and ground truth. The red line indicates the real trajectory, the point set represents the localization points, and the localization error is reflected by the color mapping with colorbar. The remaining three images are partial results of the specified interval interception. The units of all numbers in the figure are m.

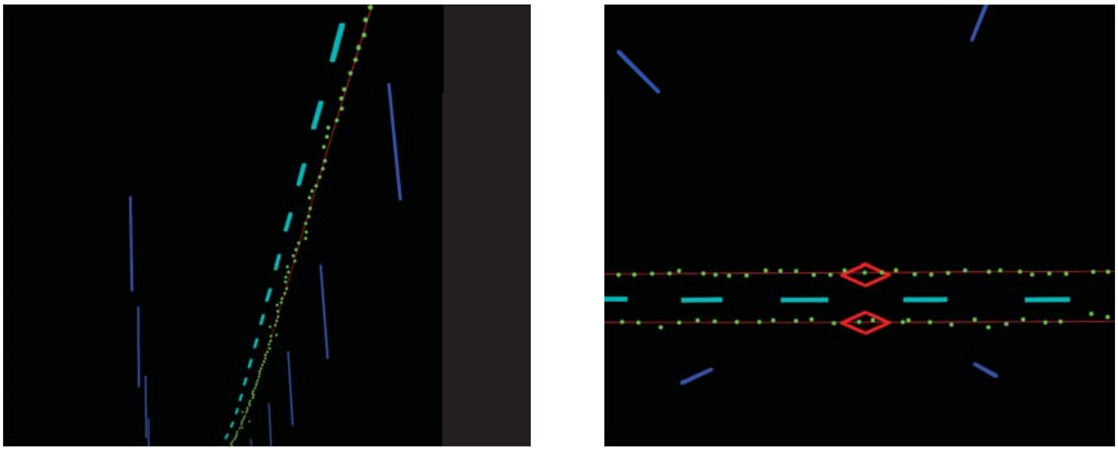


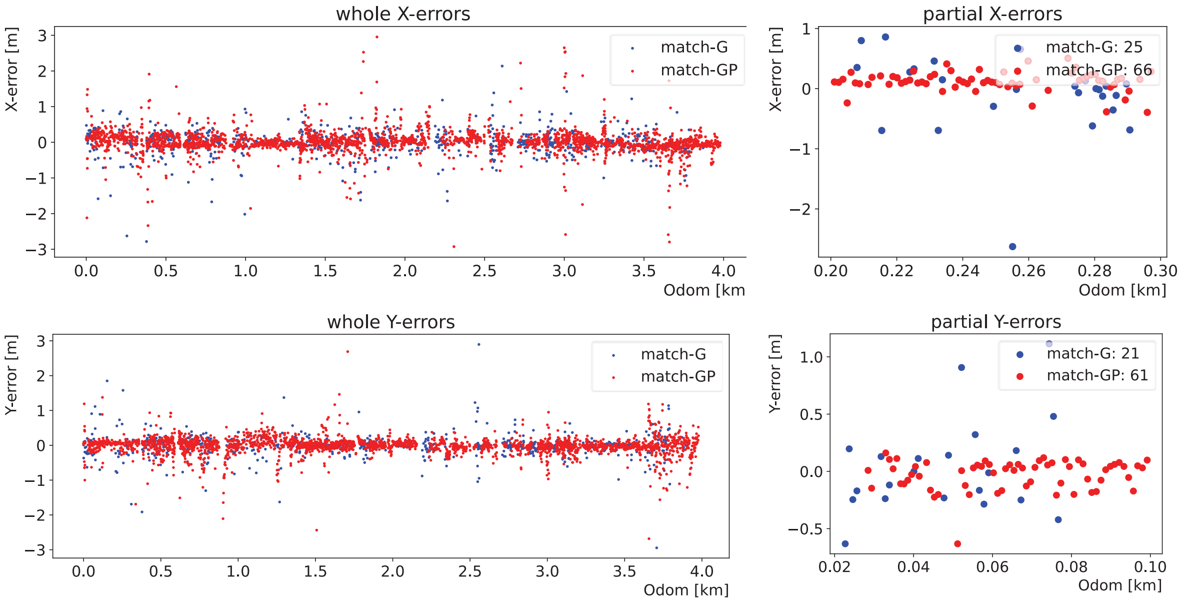
Figure 9: The localization results of match-GP method. Green dots indicate localization points, red lines are real trajectories, and different semantic features in the map are represented in different colors.

Table 1
The localization error results in the directions of x, y and yaw. The localization point numbers and map size are counted respectively. X, Y (Unit: m), Yaw (Unit: degree), Average time (Unit: ms).

method	X	Y	YAW	Localization Numbers	Average time	Map size
match-G	0.347	0.283	0.493	802	6.807	19.4 kB
match-GP	0.328	0.246	0.458	2720	8.741	35.2kB
[5]	0.104	0.107	0.143	3374	58.516	293.6MB

5. Conclusion

In this paper, we propose a parameterized multidimensional semantic mapping method that relies on a monocular camera to achieve lightweight mapping in urban road environments. Decimeter-level localization accuracy using multidimensional features is achieved, while the kB level storage per kilometer makes it possible to apply to AV tasks with limited hardware resources. In future work, we hope to make better use of the map by applying other map semantics and boundary consistency in localization.



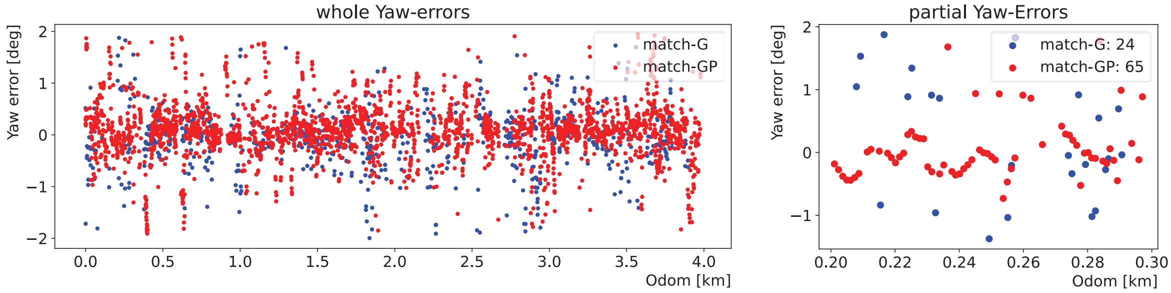


Figure 10: Localization error quantizations for match-G and match-GP on x, y and yaw. Whole and partial paths results on the left and right side respectively.

References

[1] R. Liu, J. Wang, B. Zhang, High definition map for automated driving: overview and analysis, J. Navig. 73.2 (2019) 324–341. doi:10.1017/s0373463319000638.

[2] C. Xia, Y. Shen, Y. Yang, X. Deng, S. Chen, J. Xin, N. Zheng, Onboard sensors-based self-localization for autonomous vehicle with hierarchical map, IEEE Trans. Cybern. (2022) 1 – 14. doi:10.1109/tcyb.2022.3155724.

[3] J. L. Schonberger, M. Pollefeys, A. Geiger, T. Sattler, Semantic visual localization, in: 2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR), IEEE, 2018. doi:10.1109/cvpr.2018.00721.

[4] B. Wijaya, K. Jiang, M. Yang, T. Wen, X. Tang, D. Yang, Crowdsourced road semantics mapping based on pixel-wise confidence level, Automot. Innov. 5.1 (2022) 43 – 56. doi:10.1007/s42154-021-00173-x.

[5] T. Qin, Y. Zheng, T. Chen, Y. Chen, Q. Su, A light-weight semantic map for visual localization towards autonomous driving, in: 2021 IEEE international conference on robotics and automation (ICRA), IEEE, 2021. doi:10.1109/icra48506.2021.9561663.

[6] W. Jang, J. Hyun, J. An, M. Cho, E. Kim, A lane-level road marking map using a monocular camera, IEEE/CAA J. Autom. SIn. 9.1 (2022) 187 – 204. doi:10.1109/jas.2021.1004293.

[7] H. Li, C. Xue, F. Wen, H. Zhang, W. Gao, BSP-MonoLoc: basic semantic primitives based monocular localization on roads, in: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2021. doi:10.1109/iros51168.2021.9636321.

[8] T. Wen, K. Jiang, B. Wijaya, H. Li, M. Yang, D. Yang, TM^3 Loc: tightly-coupled monocular map matching for high precision vehicle localization, IEEE Trans. Intell. Transp. Syst. (2022) 1 – 14. doi:10.1109/tits.2022.3176914.

[9] H. Wang, C. Xue, Y. Tang, W. Li, F. Wen, H. Zhang, LTSR: long-term semantic relocalization based on HD map for autonomous vehicles, in: 2022 IEEE international conference on robotics and automation (ICRA), IEEE, 2022. doi:10.1109/icra46639.2022.9811855.

[10] C. Guo, M. Lin, H. Guo, P. Liang, E. Cheng, Coarse-to-fine semantic localization with HD map for autonomous driving in structural scenes, in: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2021. doi:10.1109/iros51168.2021.9635923.

[11] Y. Zhou, X. Li, S. Li, X. Wang, Visual mapping and localization system based on compact instance-level road markings with spatial uncertainty, IEEE Robot. Autom. Lett. 7.4 (2022) 10802 – 10809. doi:10.1109/lra.2022.3196470.

[12] W. Cheng, S. Yang, M. Zhou, Z. Liu, Y. Chen, M. Li, Road mapping and localization using sparse semantic visual features, IEEE Robot. Autom. Lett. 6.4 (2021) 8118 – 8125. doi:10.1109/lra.2021.3068948.

[13] T. Wen, K. Jiang, J. Miao, B. Wijaya, P. Jia, M. Yang, D. Yang, Roadside HD map object reconstruction using monocular camera, IEEE Robot. Autom. Lett. (2022) 1 – 8. doi:10.1109/lra.2022.3185367.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with atrous separable convolution for semantic image segmentation, in: Computer vision – ECCV 2018, Springer International Publishing, Cham, 2018, p. 833 – 851. doi:10.1007/978-3-030-01234-2_49.

[15] Y. Sun, X. Han, K. Sun, B. Li, Y. Chen, M. Li, Sem-LSD: A Learning-based Semantic Line Segment Detector, arXiv preprint arXiv:1909.06591 (2019).

[16] Y. He, J. Zhao, Y. Guo, W. He, K. Yuan, PL-VIO: tightly-coupled monocular visual – inertial odometry using point and line features, Sensors 18.4 (2018) 1159. doi:10.3390/s18041159.

[17] M. Bertozz, A. Broggi, A. Fascioli, Stereo inverse perspective mapping: theory and applications, Image Vis. Comput. 16.8 (1998) 585 – 590. doi:10.1016/s0262-8856(97)00093-0.

[18] M. Ester, H. Kriegel, J. Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, in: 1996 knowledge discovery and data mining(kdd), AAAI Press, 1996, pp. 226 – 231.

[19] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE international conference on image processing (ICIP), IEEE, 2016. doi:10.1109/icip.2016.7533003.

[20] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, A. Kim, Complex urban dataset with multi-level sensors from highly diverse urban environments, Int. J. Robot. Res. 38.6 (2019) 642 – 657. doi:10.1177/0278364919843996.

[21] MMSegmentation Contributors. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark, 2020. URL: https://github.com/open-mmlab/mmsegmentation.