SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2025, Vol. 68, Iss. 2, 122106:1–122106:17
https://doi.org/10.1007/s11432-023-4114-6

BEV-Locator: an end-to-end visual semantic localization network using multi-view images

Zhihuang ZHANG[2†], Meng XU[1*†], Wenqiang ZHOU[3], Tao PENG[3], Liang LI[2] & Stefan POSLAD[4]

[1]*School of Information Technology & Management, University of International Business and Economics, Beijing 100029, China*
[2]*School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China*
[3]*Qcraft Inc., Beijing 100054, China*
[4]*School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK*

Received 27 September 2023/Revised 25 January 2024/Accepted 25 March 2024/Published online 17 January 2025

Abstract Accurate localization ability is fundamental in autonomous driving. Traditional visual localization frameworks approach the semantic map-matching problem with geometric models, which rely on complex parameter tuning and thus hinder large-scale deployment. In this paper, we propose BEV-Locator: an end-to-end visual semantic localization neural network using multi-view camera images. Specifically, a visual BEV (bird-eye-view) encoder extracts and flattens the multi-view images into BEV space. While the semantic map features are structurally embedded as map query sequences. Then a cross-model transformer associates the BEV features and semantic map queries. The localization information of ego-car is recursively queried out by cross-attention modules. Finally, the ego pose can be inferred by decoding the transformer outputs. This end-to-end model speaks to its broad applicability across different driving environments, including high-speed scenarios. We evaluate the proposed method in large-scale nuScenes and Qcraft datasets. The experimental results show that the BEV-Locator is capable of estimating the vehicle poses under versatile scenarios, which effectively associates the cross-model information from multi-view images and global semantic maps. The experiments report satisfactory accuracy with mean absolute errors of 0.052 m, 0.135 m and 0.251° in lateral, longitudinal translation and heading angle degree.

Keywords visual localization, semantic map, bird-eye-view, transformer, pose estimation

Citation Zhang Z H, Xu M, Zhou W Q, et al. BEV-Locator: an end-to-end visual semantic localization network using multi-view images. Sci China Inf Sci, 2025, 68(2): 122106, https://doi.org/10.1007/s11432-023-4114-6

1 Introduction

Recently, the research on intelligent driving has attracted considerable attention both in academia and industries [1, 2]. Accurate and robust vehicle localization is one of the crucial modules of autonomous driving and advanced driver assistance systems (ADAS) [3]. As shown in Figure 1, founded on the accurate pose, the perceptual range and capabilities of an intelligent vehicle can be boosted from a pre-built high-definition (HD) map. Besides, the map also provides rich fundamental prior information for vehicle navigation [4], planning [5], and control [6]. Therefore, the task of the localization module is to estimate the precise position and orientation of the vehicle in a known scene through the sensors launched on itself.

The problem of vehicle localization has been previously explored intensively by exploiting the features of geometry and visual appearance (e.g., scale-invariant feature transform (SIFT) features, LiDAR intensity [7], LiDAR point clouds [8], etc.), which overcomes the limitations of GPS and IMU when signal drift and blocking occur. Traditional handcrafted features (e.g., SIFT, speeded up robust features (SURF) [9], oriented FAST and rotated brIEF (ORB) [10] features) show good performance without much variance of environment conditions, in the meanwhile, the robustness lacks under varying environments (e.g., dynamic objects, motion blur or changes in lighting). As an alternative, onboard LiDAR could acquire rich landmark information [11]. However, the high cost and expensive computation, as well as the

* Corresponding author (email: xumeng@uibe.edu.cn)
† These authors contributed equally to this work.

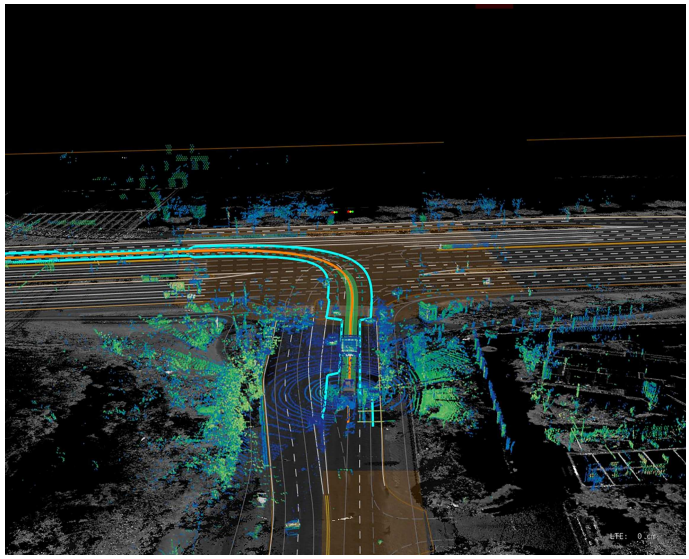

Figure 1 (Color online) Perceptual range and capabilities of the intelligent vehicle are boosted from the HD map.

fewer detected features and greater noises in rainy and snowy weather limit its wide application. Recent studies [12–16] indicate that the semantic map with location and type information of landmarks could help to improve the robustness in localization task with a reasonable cost, Besides, the semantic description is more robust against environmental variance caused by weather change, light condition, and pavement wear.

Previous studies [17–20] have engaged in semantic map information based localization, which tightly links semantic and visual features. The mainstream approach implies three steps for localizing the vehicle pose: semantic feature extracting through convolutional neural networks (CNN), semantic features association (e.g., RANSAC, KD-tree with semantic projection), and pose optimization [21] or filtering [22]. While these model-based methods are fairly effective, the algorithm relies on plane assumption and hand-crafted features, which may bring projection offset and inconsistent perception ability of features from different distances and scales. To tackle these problems, previous studies design complex constraints and strategies based on prior knowledge, which is effort-costly and time-consuming when processing the cross-model problem with uncertain scales.

With these challenges in mind, in this paper, we propose BEV-Locator: an end-to-end localization framework that requires little hand-engineering features extraction and parameter tuning. The primary motivation behind this work is not merely to enhance location precision but to integrate an end-to-end localization technique, combining surround-view images and semantic maps, into larger models addressing autonomous driving tasks. In the realm of autonomous driving, the majority of mass-produced techniques lean heavily on semantic maps. The adaptability of semantic maps, coupled with our multi-view images, supports the incorporation of diverse semantic categories. This adaptability underscores its wide-ranging effectiveness in varied driving conditions, notably in high-speed settings. BEV-Locator learns to predict the optimal pose of the ego-vehicle through supervised learning. We believe this data-driven manner may significantly simplify the visual semantic localization problem. Specifically, we encode visual features by transforming surrounding images into bird-eye-view (BEV) feature space. In the meanwhile, the semantic map is encoded to form map queries. A transformer structure is adopted to associate map queries and BEV features. Finally, the network decodes ego-pose through transformer outputs.

To the best knowledge of the authors, BEV-Locator is the first work that formulates the visual semantic localization problem as an end-to-end learning scheme. The major contributions of this research are as follows:

- We propose a novel end-to-end architecture for visual semantic localization from multi-view images and semantic environment, allowing accurate pose estimation of the ego-vehicle. The data-driven manner avoids geometry optimization strategy design and parameter tuning.
- By adopting the transformer structure in cross-modal feature association, querying, and encoding-decoding. we address the key challenge of the cross-modality matching between semantic map elements and camera images.

• We utilize the surrounding images to enhance the perceptual capabilities of the images through a unified BEV feature space. The feasibility of the visual semantic localization problem to be a subtask of BEV feature-based large model is validated.

• Through a series of experiments on the large-scale nuScenes and Qcraft datasets, we show the validity of our proposed model, which achieves the state-of-the-art performance on both datasets. We also verify the necessity and performance of the BEV grid setting, transformer encoder strategy and positional embedding strategy by ablation study.

2 Related work

2.1 Geometric information based localization

Geometric features have been explored to apply in large-scale visual localization with many attempts. Traditional local features (e.g., SIFT, SURF, ORB, etc.) play an important role to create 2D-3D correspondences from points in images and structure from motion (SfM) model point sets, and then the camera pose is retrieved using the matches. For instance, Ref. [23] employed random forest to directly predict correspondences between RGB-D images and 3D scene points. Ref. [24] proposed a bidirectional matching with SIFT features and geo-registered 3D point cloud. However, suffering from the environment with variance or repetitive conditions, the matching accuracy of local features drastically decreases especially in long-term localization. While global features (e.g., vector of locally aggregated descriptors (VLAD)-like feature [25] and DenseVLAD [26]) show impressive performance and robustness in terms of long-term localization, they still need training data for each scene with less scalability.

With the development of deep learning, learnable features have recently been integrated into image matching tasks instead of hand-crafted features. The learnable descriptors (e.g., group invariant feature transform (GIFT) [27], HardNet [28], and SOSNet [29], etc.) are applied after the local detection features extraction process. Furtherly, the detection and description steps are replaced into end-to-end networks, such as detect-then-describe (e.g., SuperPoint [30]), detect-and-describe (e.g., R2D2 [31], D2-Net [32]), and describe-then-detect (e.g., DELF [33]) strategies. With the filtered matches calculated from CNN feature based image matching algorithms, the localization information is then computed by RANSAC or SfM, which is computation-consuming in a multi-stage manner. Localization from the information fusion is formulated into optimization [21] or filtering problem [22], which relies on the plane assumption, prior knowledge, and multiple parameters in a time-consuming multi-stage manner.

Our proposed approach proposes an end-to-end framework that can be easily augmented to extract image features in changing environmental conditions (e.g., day and night, varying illumination, etc.). It also incorporates semantic information auxiliary approach, unlike some approaches that use multi-stage procedures with optimization or filtering to fuse the information, which allows the network to learn robust and accurate localization in a concise and environment-insensitive way.

2.2 Cross-view encoding surrounding images

Due to the great need for facilitating the cross-view sensing ability of vehicles, many approaches have tried to encode surrounding images into the BEV feature space. In the past few years, four main types of view transformation module schemes have emerged for BEV perception. Methods (e.g., Cam2BEV [34] and VectorMapNet [35]) based on inverse perspective mapping (IPM) inversely map the features of the perspective space to the BEV space through the plane assumption. However, the IPM based BEV encoding method is usually used for lane detection or free space estimation because of its strict plane assumption. Another series of methods was first proposed by Lift-Splat [36], which uses monocular depth estimation, lifts 2D image features into frustum features of each camera, and "splat" on BEV. Following work includes BEV-Seg [37], CaDDN [38], FIERY [39], BEVDet [40], and BEVDepth [41]. Although improvements are considered in different aspects, the Lift-Splat-based method consumes a lot of video memory due to the use of additional network estimation depth and limits the size of other modules, which affects the overall performance.

Since 2020, transformers [42] have become popular in computer vision, attention-based transformer shows attractiveness for modelling view transformation, where each location in the target domain has the same distance to visit any location in the source domain, and overcomes the limited local receptive field of the convolutional layer in CNN. PYVA [43] is the first method to propose that a cross-attention

decoder can be used for view transformation to lift image features to the BEV space. BEVFormer [44] interacts spatio-temporal information through pre-defined grid-shaped BEV query, spatial cross attention (SCA) and temporal self attention. Although the data dependencies of transformers make them more expressive, they are also difficult to train. Additionally, deploying a transformer module in embedded systems with limited resources for autonomous vehicles can also be a significant challenge.

The MLP series of methods learn the mapping between perspective space and BEV space by modelling view transitions. VPN [45] stretches the 2D physical extent of the BEV into a 1-dimensional vector and then performs a full join operation on it. But it ignores strong geometric priors, each camera must learn a network with many parameters, with a certain risk of overfitting. Considering the prior knowledge, PON [46] proposed a semantic Bayesian occupancy grid framework, which accumulates multi-cameras information cross-image scales and timestamps. The multi-layer perceptron (MLP) method adopts a data-driven approach, and could easily employ on vehicles. Our work integrated MLP-based BEV generating mechanism [47] into the visual semantic localization model to show the effectiveness of unifying the surrounding images features and connecting the BEV space, which could model 3D environments implicitly and consider the camera extrinsic explicitly, thus could be easily fused with semantic information to further improve the accuracy and robustness.

2.3 Semantic map based localization

Maps could provide powerful cues for tasks such as scene understanding [48] and localization [49, 50] in robotics using semantic labels (traffic lights, lane lines, pedestrian crossings, etc.). The pose of the vehicle could be computed by matching sensor input and a prior map, Refs. [51, 52] represented maps as LiDAR intensities, Ref. [53] constructed dense semantic maps from image segmentation and localize by matching semantic and geometric information, Refs. [54, 55] followed a coarse-to-fine way, firstly the traffic sign object detection is used to retrieve the in the geometric reference database, and then the position of the vehicle is estimated through bundle adjustment. The proposal of VectorNet [56] provides inspiration for our network, using the vector to encode different features of the semantic map information, which enables a structured representation of semantic maps and can be adapted to the input of convolutional networks.

2.4 Cross-model semantic and visual features association

The semantic map-based visual localization task searches for the best matching vehicle pose by combining the current visual input and map information. The work of semantic map matching through data association and pose estimation process is beneficial to large-scale deployment with a small storage consumption, but repeatedly associating local and online semantic features brings problems such as false and missing matching. To address this problem, later studies applied filtering or optimization algorithms to estimate pose, Ref. [57] used particle filters to update matching features, Ref. [12] reprojected map features and minimized line and point residuals to optimize the pose. However, with inconsistent perception ability of features at different distances and scales, these methods need prior knowledge and multiple parameters.

With the impressive continuous development of deep learning technology, transformers have investigated data association. Compared to traditional geometric feature-based methods or semantic map-based methods, learning-based localization methods combining semantic and visual features could encode useful features through neural networks without extracting flexible parameter designing and multi-stage work.

We apply the transformer structure to associate visual and semantic features by cross-model querying, and further decode the vehicle pose from the query features. With the supervision of the transformer network, the model could match the semantic and visual information in an end-to-end fashion.

3 Methods

This section introduces the details of BEV-Locator: an end-to-end visual localization neural network architecture to locate the vehicle poses based on surrounding images and the semantic map. The visual semantic localization problem can be formulated as: given the surrounding multi-view images $\mathcal{I}_{i=1,\ldots,n}$ of the current state (n indicates the number of cameras), the initial pose $[\check{x}, \check{y}, \check{\psi}]$ (x, y, ψ are the 2D position and the yaw angle under local navigation coordinate system) of the ego-vehicle, and the corresponding semantic map (including the position and semantic type of boundaries, dividers, markings, poles, etc.) from online map-database, determine the optimal pose $[\hat{x}, \hat{y}, \hat{\psi}]$ of the ego-vehicle. Specifically, the inputs

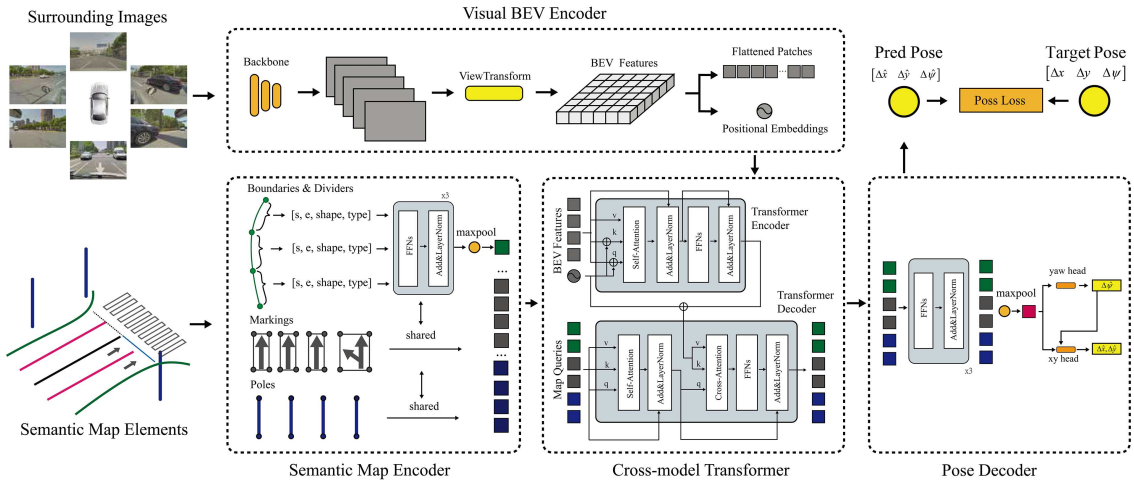

Figure 2 (Color online) Overview of our proposed BEV-Locator framework, consisting of BEV encoder (extracts features from surrounding images and projects to BEV space), semantic map encoder (encodes semantic map information to structural vectors, which are also considered as map queries for transformer module), cross-model transformer module (computes the attention and query ego-pose information based on map queries and BEV feature), and pose decoder (maps the query vectors into vehicle pose).

of BEV-Locator are the surrounding camera images and the semantic map that is projected to the initial pose. The output is the delta pose $[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}]$ between initial pose and the predicted pose. In other words, we obtain the optimal pose as follows:

$$[\hat{x}, \hat{y}, \hat{\psi}]^{\mathrm{T}} = [\check{x}, \check{y}, \check{\psi}]^{\mathrm{T}} + [\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}]^{\mathrm{T}}. \tag{1}$$

Figure 2 illustrates a modular overview of the proposed framework, consisting of a visual BEV encoder module, a semantic map encoder module, a cross-model transformer module, and a pose decoder module. The BEV feature of the surrounding images is transferred into a rasterized representation by the visual BEV encoder. The semantic map is instance-wise encoded as several compact vectors (also regarded as map queries) through the semantic map encoder. Conditioned on the BEV features and map queries, the cross-model transformer module computes the self-attention and cross-attention to query out pose information of ego-vehicle. Based on the queried-out information, the pose decoder module further infers the ego-pose where the map features have an optimal matching relationship with the corresponding images.

3.1 Visual BEV encoder

The visual BEV encoder serves to extract features of images from surrounding views and project to BEV feature space, which is serially parameterized by three components, namely image feature extractor $\phi_{\mathcal{I}}$, view transformer ϕ_V and BEV feature dimensionality reduction module ϕ_R (see Algorithm 1).

Image feature extractor $\phi_{\mathcal{I}}$ takes the surrounding images $\{\mathcal{I}_i \in \mathbb{R}^{C\times H_i\times W_i}\}_{i=1,\ldots,n}$ from n cameras as input, where H_i, W_i and C are the dimension of height, width and channel of each image I_i. The feature map $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathcal{I}}} \subseteq \mathbb{R}^{C_{\phi_{\mathcal{I}}}\times H_{\phi_{\mathcal{I}}}\times W_{\phi_{\mathcal{I}}}}$ of each image $\mathcal{I}_i$ is generated through a shared backbone, where $H_{\phi_{\mathcal{I}}}$, $W_{\phi_{\mathcal{I}}}$ and $C_{\phi_{\mathcal{I}}}$ represent the feature map dimension.

$$\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathcal{I}}} = f_{\phi_{\mathcal{I}}}(\mathcal{I}_i). \tag{2}$$

Inspired by VPN [45], we transform the extracted image features into BEV space by view transformer ϕ_V, which contains view relation module (VRM) ϕ_{VRM} and view fusion module (VFM) ϕ_{VFM}. VRM transfers the extracted image features from the image coordinate to the camera coordinate by MLP ϕ_{VRM}, then with the given corresponding extrinsic matrix $\mathcal{E}_i$, the BEV feature $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VFM}}}$ of each image $\mathcal{I}_i$ is projected from $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VRM}}}$, then the BEV features $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VRM}}}$ of n cameras are merged into the unified BEV space $\mathcal{F}_{\mathcal{I}}^{\mathrm{BEV}} \subseteq \mathbb{R}^{C_{\mathrm{BEV}}\times H_{\mathrm{BEV}}\times W_{\mathrm{BEV}}}$.

$$\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VRM}}} = f_{\phi_{\mathrm{VRM}}}(\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathcal{I}}}), \tag{3}$$

$$\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VFM}}} = f_{\phi_{\mathrm{VFM}}}(\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VRM}}}, \mathcal{E}_i), \tag{4}$$

Algorithm 1 Visual BEV encoder.

Input:
n: number of surrounding cameras;
$\{\mathcal{I}_i\}_n$: images from n cameras;
$\mathcal{E}_i$: the corresponding extrinsic matrix for each image;
Output:
$\mathcal{F}_{\mathcal{I}}^t$: surrounding images flattened features in BEV space;
PE: positional embedding of the BEV features;
1. $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathcal{I}}} \leftarrow$Feature_extraction$(\mathcal{I}_i)$;
2. $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VRM}}} \leftarrow$View_Relation_Module$(\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathcal{I}}})$;
3. $\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VFM}}} \leftarrow$View_Fusion_Module$(\mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VRM}}}, \mathcal{E}_i)$;
4. $\mathcal{F}_{\mathcal{I}}^{\mathrm{BEV}} \leftarrow$ Merge_into_BEV_space $({\mathcal{I}_i}^{\phi_{\mathrm{VFM}}}, N)$;
5. $\mathcal{F}_{\mathcal{I}}^{R} \leftarrow$Dimension_reduction_network $(\mathcal{F}_{\mathcal{I}}^{\mathrm{BEV}})$;
6. $(\mathcal{F}_{\mathcal{I}}^t, \mathrm{PE}) \leftarrow$Flatten_into_patches$(\mathcal{F}_{\mathcal{I}}^R)$.

Algorithm 2 Semantic map encoder.

Input:
$\mathcal{M}$: the semantic map, which includes:
X: number of boundaries or dividers;
$B = \{B_i\}_X$: elements of boundaries or dividers;
Y: number of markings;
$M = \{M_i\}_Y$: elements of markings;
K: number of poles;
$P = \{P_i\}_K$: elements of poles;
Output:
$\mathcal{MQ}$: map queries which is fixed-sized representation;
1. $\mathcal{M}^t \leftarrow$Structured_represent$(\mathcal{M})$;
2. $\mathcal{MQ} \leftarrow$ Multilayer_Perceptron_Max_Pooling$(\mathcal{M}^t)$.

$$\mathcal{F}_{\mathcal{I}}^{\mathrm{BEV}} = \frac{1}{N}\sum_{i=1}^{i=N} \mathcal{F}_{\mathcal{I}_i}^{\phi_{\mathrm{VFM}}}. \tag{5}$$

After feature extraction, perspective transformation and external parameter transformation, we obtained dense images in the BEV space. In order to be better suitable for subsequent transformer-based network training, we then apply ResNet network ϕ_R to reduce the dense BEV images dimension, in which the features are reduced into a lower-resolution map $\mathcal{F}_{\mathcal{I}}^R$.

$$\mathcal{F}_{\mathcal{I}}^R = f_{\phi_R}(\mathcal{F}_{\mathcal{I}}^{\mathrm{BEV}}). \tag{6}$$

Inspired by the transformer structure in DETR [58], the BEV feature map $\mathcal{F}_{\mathcal{I}}^R$ are flattened into sequence as $\mathcal{F}_{\mathcal{I}}^t$. Besides, the model supplements the BEV features with positional embedding to preserve spatial order and enhance the perception ability.

3.2 Semantic map encoder

Semantic maps, including the elements of boundaries, dividers, road markings or poles are usually represented in the form of lines, polygons or points. However these elements lack a unified structure, therefore they could not be fed directly into neural networks. Inspired by VectorNet [56], we encode the map elements from discrete points into structured vectors. Specifically, a semantic map $\mathcal{M}$ consists of a set of road elements. Each element can be represented as a set of discrete points. For example, a road divider $B_i = \{v_i \in \mathbb{R}^2 | i = 1, \ldots, N_b\}$ consists of N_b points. The proposed solution is detailed in Algorithm 2. Following the VectorNet we can denote the vector as follows:

$$v_i = [p_i^s, p_i^e, s_i, t_i], \tag{7}$$

where p_i^s and p_i^e represent the 2D position of adjacent point inside a map element; s_i stands for the shape of the map element (point, line, polygon, etc.), t_i is the semantic label of the vector (road curb, road divider, pole, marking, etc.).

To form fixed-size tensors for feeding semantic map elements into network training, we design a three-dimensional structure to store semantic map element information. The first dimension size is the maximum number D_1 of map elements. The second dimension size D_2 is the maximum number of vectors (the number of discrete points in each map element). The third dimension is to represent the vector

Algorithm 3 Cross-model transformer.

Input:
$\mathcal{F}_{\mathcal{I}}^{t}$: surrounding images flattened features in BEV space;
$\mathcal{MQ}$: fixed-sized structured semantic map queries;
PE: positional embedding of the BEV features;
Output:
$\mathcal{TQ}$: transformer guided queried feature;
1. $\mathcal{F}_{\mathcal{I},\text{encoder}}^{t} \leftarrow$Transformer_encoder($\mathcal{F}_{\mathcal{I}}^{t}$);
2. $\mathcal{TQ} \leftarrow$Transformer_decoder($\mathcal{MQ}, \mathcal{F}_{\mathcal{I},\text{encoder}}^{t}$, PE).

attributes. Following this pattern, we load the unstructured semantic map $\mathcal{M}$ into a fixed-size structured representation, where $\mathcal{M}^t \subseteq D_1 \times D_2 \times 8$. We pad the blank elements with 0 and prepare a map mask to indicate existing elements.

The semantic map encoder encodes map elements into map queries. Each node of the semantic element is first mapped to a high-dimensional space through a shared MLP. And a max pooling layer extracts the global information inside the element. In practice, the MLP and max pooling operations are repeated to increase encoder capacity. We denote the global information as a map query, which meets the concept in the transformer structure. The overall map queries is represented as $\mathcal{MQ} \subseteq \mathbb{R}^{D_1 \times \text{dim}_{\text{emb}}}$, dim_{emb} is the encoded query dimension.

3.3 Cross-model transformer module

Our cross-model transformer module is built on the basic structure of the transformer [42], which associates the map queries and BEV features to query ego-pose information. The module takes the input from the visual BEV encoder module and semantic map encoder module and contains an encoder-decoder structure. The solution of this module is depicted in Algorithm 3.

Transformer encoder. The encoder takes a flattened BEV feature patches sequence $\mathcal{F}_{\mathcal{I}}^{t}$ as input. Each encoder layer contains a multi-head self-attention module and a position-wise fully connected feed forward network (FFN), each followed by layer normalization (LN) [59] and the residual connection (RC) [60].

Transformer decoder. The decoder transforms the map queries $\mathcal{MQ}$ with D_1 embeddings of size dim_{emb}. Each decoder layer consists of a multi-head self-attention module, a cross-attention module and the FFN module, each followed by LN and RC. Finally the predicted query embeddings $\mathcal{TQ} \subseteq \mathbb{R}^{D_1 \times \text{emb\_dim}}$ are independently decoded by the FFN. Using self-attention and cross-attention mechanisms makes the model globally map the pair-wise relations between permutation-invariant map queries and BEV features, while embedding the local position information to assist the querying. Varying from the traditional transformer decoder in detection task [58], the position information of BEV features would benefit the localization task. Thus the positional embedding is also applied to the value input in the cross-attention module. This slight modification of transformer structure is major in the final accuracy, which will be discussed in the ablation study.

3.4 Pose decoder and pose loss function

Conditioned on the information queried out by the transformer, the pose of the ego-vehicle can be decoded by the pose decoder. We consider each map query contains pose information or constraint offered by the corresponding map element. Therefore, the pose decoder is designed to aggregate information from each map query and predict pose from a global perspective. We adopt a shared MLP to further encode the map queries and a max pooling layer to aggregate global information. The max pooling layer merges the map queries into a global permutation-invariant vector. Finally, an MLP maps the global information to the offset $[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}]$ between the current estimated pose and initial pose (see Algorithm 4):

$$[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}] = \mathcal{PD}(\mathcal{TQ}). \tag{8}$$

The supervision of the BEV-Locator is the ground-truth pose offset, which can be manually generated or retrieved from a more precise localization module. Given the supervision $[\Delta x, \Delta y, \Delta\psi]$ and network prediction $[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}]$, the BEV-Locator can be optimized by the Smooth L1 Loss via the following loss function:

$$\mathcal{L} = \alpha \times (||\Delta\hat{x}, \Delta x||_{S1} + ||\Delta\hat{y}, \Delta y||_{S1}) + ||\Delta\hat{\psi}, \Delta\psi||_{S1}, \tag{9}$$

where α is the balance weight for the position loss and rotation loss. $||\cdot||_{S1}$ denotes the Smooth L1 loss.

Algorithm 4 Pose decoder.

Input:
$\mathcal{TQ}$: transformer guided query feature;
Output:
$[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}]$: predicted pose offset;
$[\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\psi}] \leftarrow$ Multilayer_Perceptron_Max_Pooling($\mathcal{TQ}$);
Optimization:
$\mathcal{L} = \alpha \times (||\Delta\hat{x}, \Delta x||_{S1} + ||\Delta\hat{y}, \Delta y||_{S1}) + ||\Delta\hat{\psi}, \Delta\psi||_{S1}$.

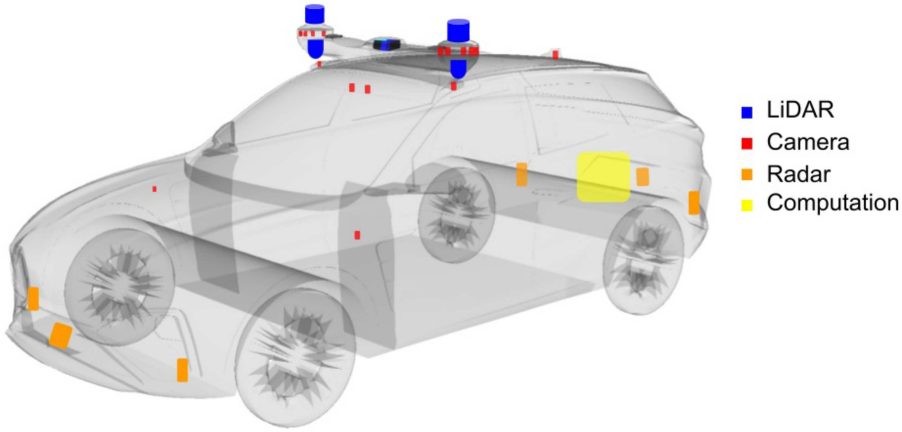

Figure 3 (Color online) Experiment platform of Qcraft dataset equipped with onboard sensors and reference devices on MarvelR car.

4 Experiments and discussions

4.1 Experimental settings

4.1.1 *Datasets*

nuScenes dataset [61] is a well-known large-scale dataset for multiple autonomous driving tasks (e.g., 3D object detection task, object tracking task, etc.), which consists of 1000 scenes covering 242 km collected in Boston and Singapore, each scene contains a full sensor-suite consisting of 1 LiDAR, 5 Radar, 6 cameras, IMU, and GNSS receivers, and 11 semantic layers (crosswalk, sidewalk, traffic lights, stop lines, lanes, etc.). 1.4 M images were captured in RGB format with a frequency of 12 Hz and a resolution of 1900×900 pixels in a diverse set of challenging locations, changing times and weather conditions (e.g., nighttime and rainy environments). The corresponding ground truth vehicle poses were calculated through a Monte Carlo Localization scheme from LiDAR and odometry information [62] in an offline HD LiDAR points map.

Qcraft dataset is recorded by an autonomous MarvelR car [63] in Suzhou, China over 400 km, with the speed varying from 3 to 80 km/h. Qcraft dataset was collected by 7 surrounding cameras, 5 LiDAR, 5 Radar, Figure 3 shows the layout of the sensors. The corresponding semantic elements (lane boundary, lane divider, light pole, etc.) with positions in the global coordinate system are also supplied. The dataset includes 7 trajectories, each trajectory consists of 7 sets of images from the surrounding views. During the dataset generation, we crop and resize the images into a uniform resolution of 1920×1080 pixels. The corresponding ground truth ego poses were obtained from the trajectory of RTK and post-processing.

4.1.2 *Task and metrics*

Visual semantic localization task takes vectorized semantic map features and surrounding images as input to estimate the vehicle localization. We measure our proposed approach with the localization metric.

Localization metric is computed in the ego-vehicle coordinate system and measures the 3-DoF pose differences. We use the mean absolute error (MAE) and 90% percentile values of the $[\Delta x, \Delta y, \Delta\psi]$ error to evaluate our 3-DoF localization model, which describes the localization performance of lateral, longitudinal, and yaw estimation.

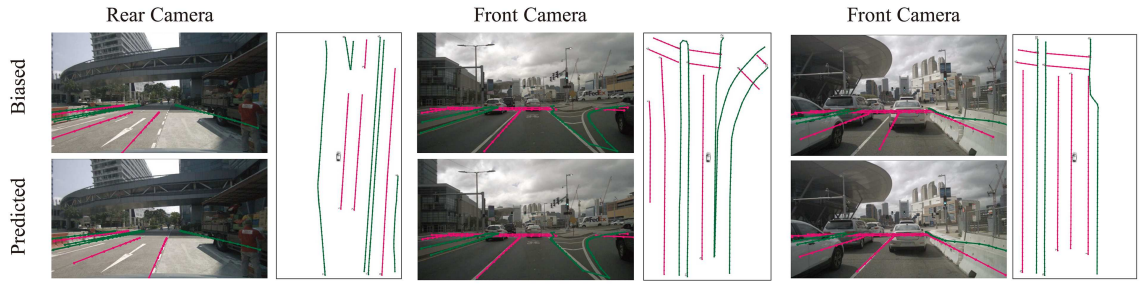

Figure 4 (Color online) Results on nuScenes dataset, where the semantic maps are reprojected onto camera images. The upper row shows the biased poses (also known as the initial guesses) and the lower row indicates that the network predicts optimal poses where the semantic map features coincide with images.

4.1.3 *Implementation details*

For the map encoder, the semantic map is loaded into $32 \times 128 \times 8$ tensor to represent all the semantic features (e.g., crosswalk, lane boundary, light pole, traffic lights, cross-walks, etc.). And the map is further encoded as $32 \times \dim_{\text{emb}}$. We generate the map mask to prevent blank embedding (all set as 0).

Inspired by [36], in the surrounding images encoding period, we use EfficientNet-B0 [64] pre-trained on ImageNet [65] as the visual backbone to extract the image features. Then, a series of MLPs are utilized to map the correspondences of the camera and ground plane. Finally, the features in the BEV are merged into dense features, which equals a physical range of [60 m $\times$ 30 m]. To reduce the dimension of BEV raster map, ResNet18 convolution network with 4 blocks is adopted to reduce images from $[C_{\text{BEV}} \times H_{\text{BEV}} \times W_{\text{BEV}}]$ to $[\dim_{\text{emb}} \times H_{\text{BEV}}/32 \times W_{\text{BEV}}/32]$.

Our proposed framework is implemented with PyTorch [66] using the ADAM [67] solver with an initial learning rate of 1e$-$5 and weight_decay of 1e$-$7 on 4 NVIDIA Tesla V100 GPU using DataParallel. The SmoothL1Loss is used for calculating translation and rotation offsets, with the weight coefficient of translation error $\alpha = 0.04$. The hyperparameters are as follows: cropped image size is 270×480, batch size = 8 (total batch size = 32), max_grad_norm = 5.0, and $\dim_{\text{emb}} = 256$.

In the training process, we generate random pose deviations. Specifically, we sample the random longitudinal deviation in [$-$2, 2] m, lateral deviation in [$-$1, 1] m and yaw deviation in [$-2°$, $2°$]. The semantic maps are first projected to the vehicle coordinate system and then translated by these deviations. The task of the network is to predict the deviation from the biased map and the surrounding images. Therefore, the deviations act as supervision, allowing the network to be trained in an end-to-end manner.

4.2 nuScenes dataset results

nuScenes dataset contains 700 train scenes and 150 test scenes in urban areas with images captured by 6 surrounding cameras. We conduct experiments on the nuScenes dataset to validate the effectiveness BEV-Locator (trained with 35 epochs).

We extract map elements from the map interface. The element types include road boundaries, lane dividers, and pedestrian crossings. All of the 6 camera images are combined to form the BEV features. Figure 4 visualizes the localization process. Based on the provided semantic map, the initial pose, and the camera parameters, the map elements can be reprojected to the image perspective view. The upper pictures show the biased pose and the lower row pictures present the pose predicted by BEV-Locator. By comparing the upper and lower pictures, it can be observed that the map elements coincide with the elements in the camera views, which indicates the ego-vehicle present in the correct position and validates the effectiveness of the BEV-Locator.

Figure 5(a) illustrates the error distribution of BEV-Locator on the nuScenes dataset. The error curves indicate BEV-Locator generates excellent pose accuracy. The position errors in the lateral and longitudinal directions are less than 20 and 60 cm. It means the position in both lateral and longitudinal directions is well constrained by map elements in most cases. Besides, the heading directions can be predicted under $1°$ error. Through the investigation, the effectiveness of BEV-Locator in the nuScenes dataset is validated.

The performance on the nuScenes dataset is not as expected primarily due to two reasons. First, the camera setup of nuScenes led to inadequate height information, making the system presume all features to be on the ground. This constrained the exploitation of 3D spatial configurations. Second, the nuScenes

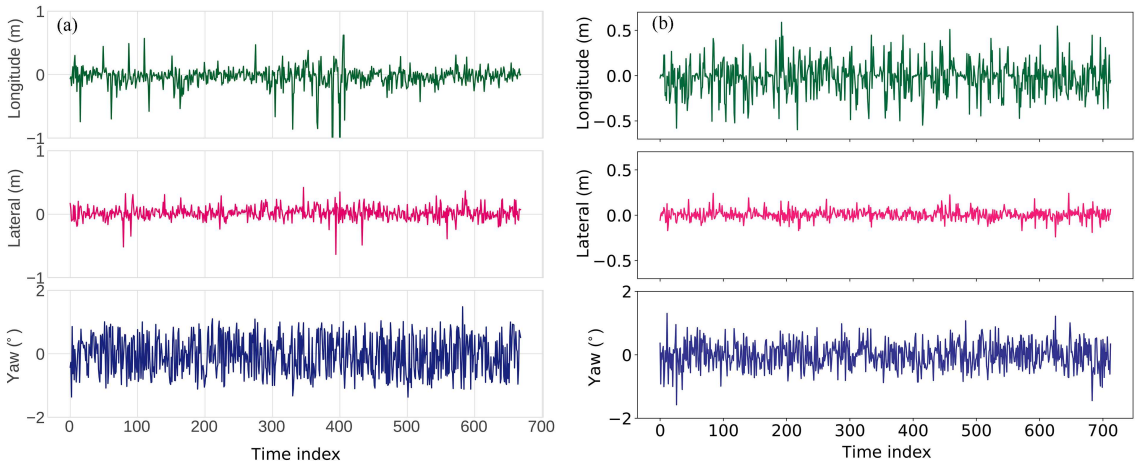

Figure 5 (Color online) (a) nuScenes dataset quantitative results; (b) Qcraft dataset quantitative results.

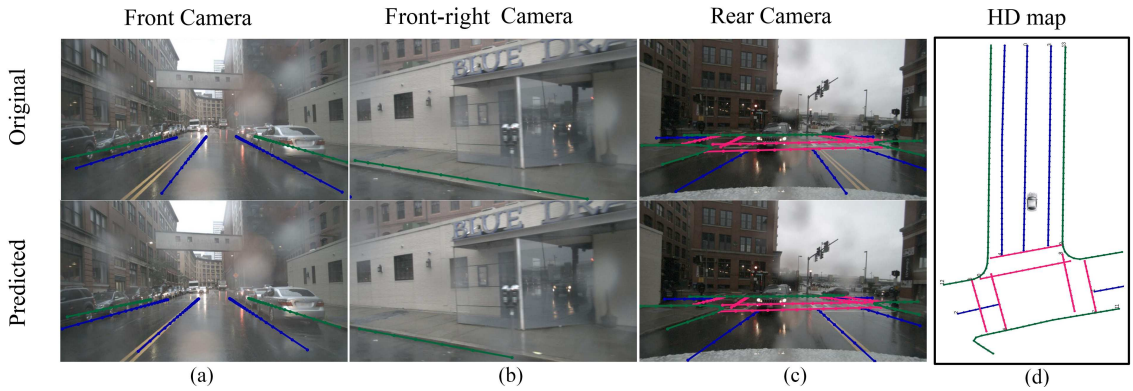

Figure 6 (Color online) Comparison of semantic feature visibility during rainy conditions in the nuScenes dataset. This figure illustrates the challenge of semantic occlusion, with specific cases highlighted: (a) front camera view demonstrates blurred lane markings and obscured road signs, (b) front-right camera view shows reflection and scattering effects on semantic cues, (c) rear camera view exhibits occlusion of traffic lines due to rain streaks, and (d) HD map presents the ground truth for reference. These scenarios underscore the importance of robust feature detection in adverse weather conditions for accurate localization.

map's quality was not tailored for localization tasks, affecting its suitability. We initially used nuScenes for preliminary validation. For a more detailed evaluation, we introduced the Qdataset, which comes with a camera extrinsic for better height data capture. All further analyses and tests pivot on results from Qdataset, ensuring insights into the model's real-world performance.

In addressing the impact of semantic occlusion on localization, our experiments demonstrate that occlusion does affect the accuracy of localization. Figure 6 illustrates the performance of our localization system under rain conditions from the nuScenes dataset, where critical features are partially obscured. The tests confirm that robust feature detection is essential, especially in adverse weather, to maintain localization reliability.

4.3 Qcraft dataset results

We further validate BEV-Locator with the Qcraft dataset, which contains urban roads and expressways with clearer lane lines and road markings. The semantic map consists of road curbs, lane dividers, road marks, and traffic light poles. For a fair comparison, 6 cameras are selected out of 7 to form the BEV features. All the training parameters are the same as those in the nuScenes dataset. Similarly, we show the reprojected semantic maps from three different views in Figure 7. The semantic map describes the road markings with enclosed polygons and the traffic poles are presented with the contact points to the ground. It can also be concluded that the BEV-Locator successfully predicts the optimal pose of the ego-pose in scenarios of the Qcraft dataset. Combining the constraints of map elements, the position and heading of the vehicle are correctly predicted by the network.

The error curves of a segmented trajectory are illustrated in Figure 5(b). Most of the lateral and

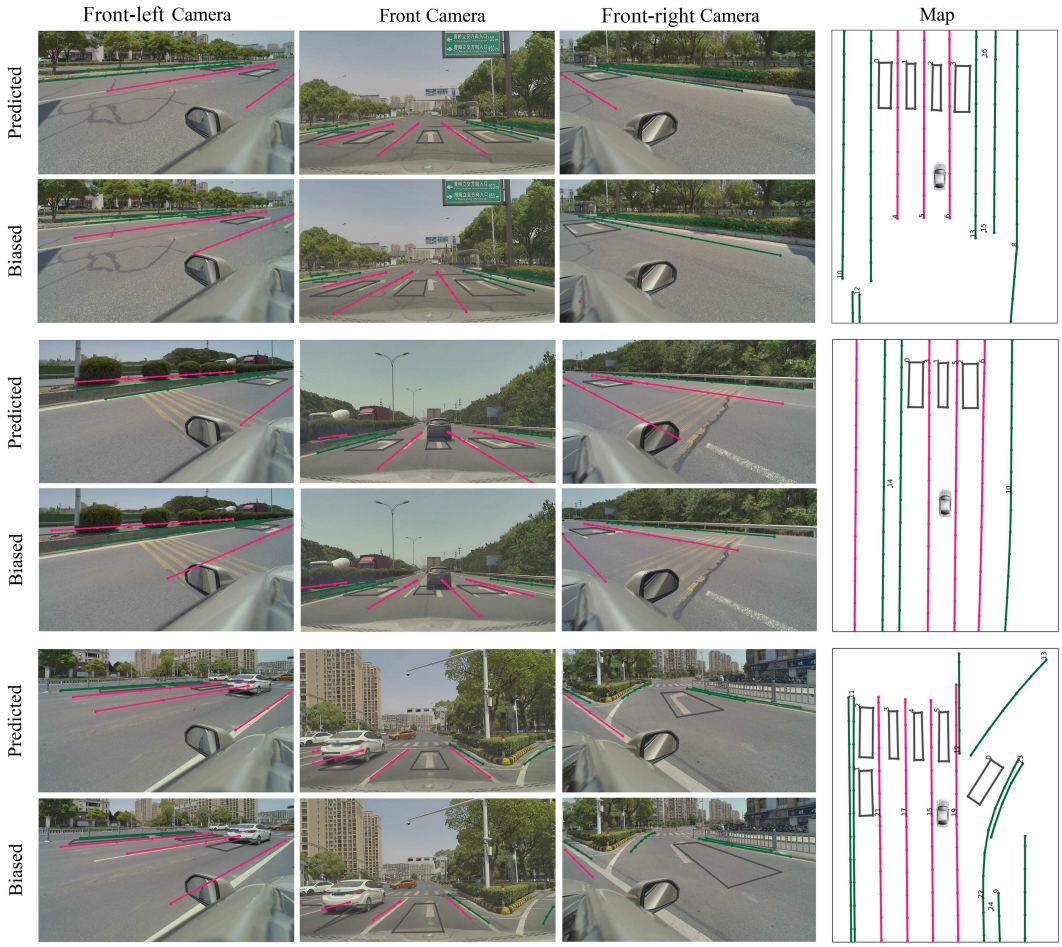

Figure 7 (Color online) Results on Qcraft dataset, where the semantic maps are reprojected onto camera images. The lower rows show the biased poses (also known as the initial guesses) and the upper rows indicate the network predicts optimal poses where the semantic map features coincide with images.

longitudinal errors are under 10 and 40 cm. Compared with the nuScenes dataset, the BEV-Locator delivers superior accuracy in the Qcraft dataset and we ascribe this to clearer road elements and higher map quality. Especially in high-speed scenarios, road curvatures tend to be minimal with clear lane markings and traffic signs, making localization comparatively easier than in lower-speed conditions. Next, the quantitative analysis and comparison with other methods would be discussed.

Regarding the effect of semantic occlusion on localization, similar experiments were performed using the Qcraft dataset. As shown in Figure 8, the system's ability to localize is challenged when key semantic features are obstructed by environmental elements such as heavy rain. The results indicate that robust algorithms could enable accurate localization under diverse conditions.

4.4 Comparison with existing methods

Table 1 [12, 14, 61, 68–70] illustrates the performance comparison of our proposed BEV-Locator with other existing localization techniques. It is imperative to note at the outset that visual localization research invariably involves disparate hardware configurations, scenarios, and map utilities, engendering a considerable diversity in the experimental setups and consequently, the results. Due to this, we have endeavoured to present a balanced comparison focusing on the overall localization accuracy obtained through various approaches.

As a pioneering method, the comparable results lack from existing localization methods on the nuScenes dataset. We assess the efficacy of QDataset with a baseline. Findings underscore the robustness of our method, which, when tested across both nuScenes and QDataset, displayed a localization error reduction by an order of magnitude compared to other methods.

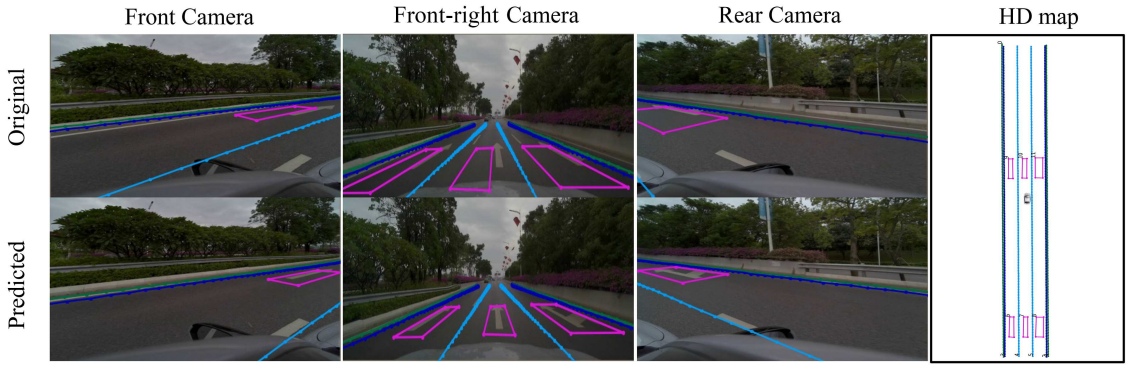

Figure 8 (Color online) Comparison of semantic feature visibility during rainy conditions in the Qcraft dataset.

Table 1 Performance comparison between existing localization methods. To validate the effectiveness and performance of the proposed framework, we compare our localization results with the following methods: Choi et al. [68], Pauls et al. [69], Xiao et al. [12], Wang et al. [70] and Zhang et al. [14]. The bold numbers represent the lowest localization errors among the compared methods.

Method	Sensors	Descriptions	Dataset	Lat. (m)	Lon. (m)	Yaw (°)
Choi et al. [68]	Mono Camera+WSS +GNSS (spp)+IMU	Project image feature to ground plane for particle filtering	Highway [68]	0.10	0.25	–
Pauls et al. [69]	Camera (mono) + WSS	Semantic segmentation as detection front end, implicit association, pose graph optimization	Karlsruhe [69]	0.11	0.90	0.56
Xiao et al. [12]	Camera (mono)	Reprojecting map elements to image plane, optimizing the pixel distance errors	MLVHM [12]	0.29		–
Wang et al. [70]	Camera (mono) + WSS + GNSS	Novel association method and sliding window factor graph optimization	Urban road [70]	0.12	0.43	0.11
Zhang et al. [14]	Camera (mono)+WSS +GNSS+IMU	Reconstructing local semantic map, matching through neural network	Highway [14]	0.054	0.204	–
			nuScenes [61]	0.096	0.219	–
Imegery	LiDAR + GNSS	Histogram filter based LiDAR localization using intensity and elevation grid map	nuScenes [61]	0.125	0.185	0.653
			QDataset	0.097	0.161	0.213
BEV-Locator	Multi-view cameras	Using single time frame images; end-to-end prediction	Highway	0.063	0.158	0.382
			nuScenes [61]	**0.076**	**0.178**	0.510
			QDataset	**0.052**	**0.135**	0.251

It can be seen that BEV-Locator possesses the best position accuracy both on the nuScenes dataset and the Qcraft dataset. Compared with the other approaches using multi-sensors fusion based input, our method is based on camera-only input at a single time. In other words, our method achieves remarkable performance on the visual localization problem. Besides, since the BEV-Locator can only be trained with the supervision of the pose offset, this end-to-end manner significantly simplifies the process of building a visual semantic localization system without complex strategies or parameter fine-tuning. Moreover, since the transformer structure holds a larger learning capacity that allows for large-scale data training, the BEV-Locator could be hopefully deployed to a wide range of scenarios.

We now investigate the reason that the lateral error is smaller than the longitudinal error found in the experiment results and other visual semantic localization methods. Intuitively, the semantic elements, lane lines, road marks, and light poles provide lateral constraints simultaneously, while longitudinal position could only be constrained by light poles or road marks. The amount of longitudinal constraint elements is often less than the number of lateral constraint elements. In addition, these elements may exist at more distant distances compared with adjacent lane lines. In summary, the longitudinal accuracy is incomparable to the lateral one. Fortunately, the downstream modules also require less accuracy for longitudinal positioning, which somewhat makes up for this problem.

4.5 Efficiency evaluation

In order to evaluate the computational efficiency of the BEV-Locator, a semantic localization network, we conducted a series of experiments to quantitatively assess its inference speed. The configuration of the

Table 2 Runtime comparison between existing localization methods. To validate the runtime with different platforms, we compare our localization runtime with the following methods: Xiao et al. [12] and Zhang et al. [14]. Bold values represent the minimum processing time for the localization module and whole system across all tested methods and platforms.

Method	Platform	Modules	Time (ms) [percentage(%)]	Localization module Time (ms)	Whole system Time (ms)
Xiao et al. [12]	Intel Core i7-7700 CPU	Detect lane markings	–	4.34	–
		Detect lane endpoints	–		
		Detect road signs	–		
		Vehicle localization	4.34 [–]		
Zhang et al. [14]	GTX 1080	Map-matching inference	8.9 [35.25]	15.5	**24.55**
		Visual semantic localization process	15.5 [63.14]		
		Filter	0.15 [0.61]		
BEV-Locator	GTX 1080	BEV encoder	28.114 [88.88]	**0.520**	31.631
		Semantic map encoder	0.553 [1.75]		
		Cross-model transformer	2.444 [7.73]		
		Pose decoder	0.520 [1.64]		
BEV-Locator	RTX 3060	BEV encoder	45.846 [91.77]	0.563	49.975
		Semantic map encoder	1.114 [2.23]		
		Cross-model transformer	2.434 [4.87]		
		Pose decoder	0.563 [1.13]		

network model and inputs for these experiments align with those detailed in Subsection 4.3. We utilized the CUDA Event API within the PyTorch framework to record the runtime of each individual module in the network. To ensure the accuracy of our timing statistics, the GPU was pre-warmed with a preliminary run of 1000 cycles before the commencement of each test. Table 2 presents a comprehensive comparison of runtime performances between our method and existing localization techniques. This comparison spans different GPU platforms and encompasses the average inference time and corresponding percentages for each method. Notably, the table is structured to clearly delineate the time consumed by individual modules and the overall system for each approach, thereby offering insights into the efficiency and resource allocation of these localization methods.

As can be observed, a single pose estimation operation takes about 31.63 ms on the RTX 3060 platform and is slightly slower on the GTX 1080, averaging around 50 ms. The generation of BEV features by the BEV encoder occupies nearly 90% of the runtime, indicating its central role in the system. Excluding the time consumed for BEV feature generation, the other modules collectively take approximately 3 to 4 ms.

Under the BEV perspective of the autonomous driving perception paradigm where multiple tasks share a single BEV space feature, the BEV-Locator's localization approach eliminates common steps seen in traditional localization methods including semantic element extraction, feature association matching, and pose optimization. Thanks to the high efficiency of GPU parallel computation, integrating semantic map matching localization as a sub-task in the BEV perception model entails a small additional computational burden. Generally speaking, given the camera sampling frequency ranging between 10 and 36 Hz, it can be concluded that the BEV-Locator meets the real-time requirements for deployment. Moreover, with ongoing research aimed at efficient BEV feature generation and model deployment optimization, there is potential for further enhancing the inference efficiency of the entire model in the future.

4.6 Ablation studies

To better understand the effectiveness of each module in our framework, we conduct the ablation study to validate through a series of comparison experiments with the Qcraft dataset.

Effectiveness of different BEV grid sizes. To investigate the effects of different BEV grid sizes, in Table 3, we test the impact of different BEV grid sizes on vehicle localization performance. We observe that a smaller BEV grid size contributes to higher pose accuracy. This can be explained by the fact that higher resolution allows for better encoding pose information of the map elements. However, higher resolution also brings computational burdens, posing challenges in terms of both computation time and graphics memories.

Effectiveness of transformer encoder. Table 4 exhibits the accuracy of the BEV-Locator with or without the transformer encoder. Without encoder layers, the longitudinal error and lateral error drop

Table 3 Effectiveness of different BEV grid sizes.

BEV grid size (m)	Longitudinal (m)	Lateral (m)	Yaw (°)
0.50	0.200 [0.459]	0.072 [0.115]	0.269 [0.499]
0.25	0.177 [0.400]	0.057 [0.118]	0.262 [0.427]
0.15	0.135 [0.309]	0.052 [0.107]	0.251 [0.395]

Table 4 Effectiveness of the transformer encoder strategy.

Transformer encoder	Longitudinal (m)	Lateral (m)	Yaw (°)
Without encoder	0.214 [0.443]	0.057 [0.122]	0.273 [0.538]
With encoder	0.135 [0.309]	0.052 [0.107]	0.251 [0.395]

Table 5 Effectiveness of the positional embedding strategy in the transformer decoder.

Transformer decoder	Longitudinal (m)	Lateral (m)	Yaw (°)
With pos_embedding in value	0.135 [0.309]	0.052 [0.107]	0.251 [0.395]
Without pos_embedding in value	1.212 [1.874]	0.122 [0.343]	0.619 [0.798]

0.0789 and 0.005 m, respectively. We hypothesize that self-attention performs information interaction between BEV grids. This enables global scene awareness for road elements.

Effectiveness of positional embedding in transformer decoder. In Table 5, we evaluate the influence of different transformer strategies in the transformer decoder module. Based on our experiments, we find that the BEV-Locator converges hardly when the conventional transformer structure is adopted, especially in the longitudinal direction. The problem was solved by a slight change in the transformer decoder. We add positional embedding to the value term in the cross-attention operation. Intuitively, each map query contains both semantic information and position information of the map element. Through the transformer, the map query is meant to query out its relative position information under BEV space. Therefore, the position information (contained in positional embedding) of each grid needs to be retrieved as a value. This small change contributes significantly to the performance of the BEV-Locator.

4.7 Discussions

To sum up, we evaluated the availability of BEV-Locator through the above experiments, from which we could conclude that our method achieves state-of-the-art performance in visual semantic localization. Conditioned on the results, we summarize the following findings:

(i) We demonstrated that the semantic map elements can be encoded as queries. With the transformer structure, the pose information of the ego-vehicle can be queried from the BEV feature space. The effectiveness of the transformer for cross-modality matching between semantic map elements and visual images is verified.

(ii) We formulate the visual semantic localization problem as an end-to-end learning task. The neural network requires simple supervision generated by pose offset. Simply using vehicle trajectories with raw images and the semantic map is sufficient to generate the training dataset for the BEV-Locator.

(iii) We validate the performance and accuracy of the BEV-Locator on the nuScense dataset and Qcraft dataset. Compared with the existing visual localization methods, BEV-Locator achieves state-of-the-art performance with only the images in a timestamp. Besides, since BEV-Locator is a data-driven method, we avoid geometry optimization strategy design and parameter tuning.

(iv) We evaluate the runtime efficiency of the BEV-Locator in different GPU environments, showcasing a commendable speed with the majority of the computational time attributed to the BEV encoder module, and the system affirms its readiness for real-time applications.

(v) BEV-Locator explores the feasibility of the visual semantic localization problem as a subtask of the BEV feature-based large model. Our future work aims to integrate the BEV-Locator with other perception subtasks in a large uniform BEV model. Benefiting from the BEV and transformer structure, we hypothesize that the BEV-Locator has the potential to cope with large-scale scenarios.

5 Conclusion

We presented BEV-Locator, a new design for the visual semantic localization system based on map encoding, BEV features, and transformers for direct pose estimation of ego-vehicle. The introduced networks could efficiently encode the images and semantic maps, and further query the pose information through cross-model transformer structure. BEV-Locator is straightforward to implement following an end-to-end data-driven manner, without complex optimization strategies or complex parameter tuning. Our approach achieves state-of-the-art performance based on the nuScenes dataset and the Qcraft dataset. Our work demonstrates the effectiveness of estimating ego-pose in the BEV space. This allows visual semantic localization to be one of the subtasks of the BEV-based large model for autonomous vehicle design.

Acknowledgements This work was supported by Beijing Higher Education Society under the 2024 General Project Scheme (Grant No. MS2024128). Furthermore, the research received funding from the Ningbo Philosophy and Social Science Planning Project, as part of the "Ningbo Development Blue Book 2025" Initiative (Grant No. GL24-16). We would like to extend our gratitude to our colleagues at Qcraft for their invaluable insights and expertise, which significantly contributed to the progress of this research.

References

1 Meiring G A M, Myburgh H C. A review of intelligent driving style analysis systems and related artificial intelligence algorithms. Sensors, 2015, 15: 30653–30682
2 Ahmed A H, Elmokashfi A. ICRAN: intelligent control for self-driving RAN based on deep reinforcement learning. IEEE Trans Netw Serv Manage, 2022, 19: 2751–2766
3 Greenwood P M, Lenneman J K, Baldwin C L. Advanced driver assistance systems (ADAS): demographics, preferred sources of information, and accuracy of ADAS knowledge. Transp Res Part F, 2022, 86: 131–150
4 Alkendi Y, Seneviratne L, Zweiri Y. State of the art in vision-based localization techniques for autonomous navigation systems. IEEE Access, 2021, 9: 76847–76874
5 Artunedo A, Villagra J, Godoy J, et al. Motion planning approach considering localization uncertainty. IEEE Trans Veh Technol, 2020, 69: 5983–5994
6 Patel R H, Härri J, Bonnet C. Impact of localization errors on automated vehicle control strategies. In: Proceedings of the IEEE Vehicular Networking Conference (VNC), 2017. 61–68
7 Barsan I A, Wang S, Pokrovsky A, et al. Learning to localize using a LiDAR intensity map. 2020. ArXiv:2012.10902
8 Yin H, Tang L, Ding X, et al. LocNet: global localization in 3D point clouds for mobile vehicles. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2018. 728–733
9 Bay H, Tuytelaars T, Gool L V. SURF: speeded up robust features. In: Proceedings of the European Conference on Computer Vision, 2006. 404–417
10 Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the International Conference on Computer Vision, 2011. 2564–2571
11 Lu W, Zhou Y, Wan G, et al. L3-Net: towards learning based LiDAR localization for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 6389–6398
12 Xiao Z, Yang D, Wen T, et al. Monocular localization with vector HD map (MLVHM): a low-cost method for commercial IVs. Sensors, 2020, 20: 1870
13 Qin T, Zheng Y, Chen T, et al. A light-weight semantic map for visual localization towards autonomous driving. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2021. 11248–11254
14 Zhang Z, Zhao J, Huang C, et al. Learning visual semantic map-matching for loosely multi-sensor fusion localization of autonomous vehicles. IEEE Trans Intell Veh, 2023, 8: 358–367
15 Ren Y, Liu B, Cheng R, et al. Lightweight semantic-aided localization with spinning LiDAR sensor. IEEE Trans Intell Veh, 2023, 8: 605–615
16 Zhao Z, Zhang W, Gu J, et al. LiDAR mapping optimization based on lightweight semantic segmentation. IEEE Trans Intell Veh, 2019, 4: 353–362
17 Toft C, Stenborg E, Hammarstrand L, et al. Semantic match consistency for long-term visual localization. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 383–399
18 Karkus P, Hsu D, Lee W S. Particle filter networks with application to visual localization. In: Proceedings of the Conference on Robot Learning, 2018. 169–178
19 Xiao L, Wang J, Qiu X, et al. Dynamic-SLAM: semantic monocular visual localization and mapping based on deep learning in dynamic environment. Robot Auton Syst, 2019, 117: 1–16
20 Wu J, Shi Q, Lu Q, et al. Learning invariant semantic representation for long-term robust visual localization. Eng Appl Artif Intell, 2022, 111: 104793
21 Cho S, Kim C, Sunwoo M, et al. Robust localization in map changing environments based on hierarchical approach of sliding window optimization and filtering. IEEE Trans Intell Transp Syst, 2022, 23: 3783–3789

22 Djuric P M, Kotecha J H, Zhang J, et al. Particle filtering. IEEE Signal Process Mag, 2003, 20: 19–38

23 Shotton J, Glocker B, Zach C, et al. Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013. 2930–2937

24 Li Y, Snavely N, Huttenlocher D, et al. Worldwide pose estimation using 3D point clouds. In: Proceedings of the European Conference on Computer Vision, 2012. 15–29

25 Jégou H, Douze M, Schmid C, et al. Aggregating local descriptors into a compact image representation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. 3304–3311

26 Torii A, Arandjelovic R, Sivic J, et al. 24/7 place recognition by view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1808–1817

27 Liu Y, Shen Z, Lin Z, et al. GIFT: learning transformation-invariant dense visual descriptors via group CNNs. In: Proceedings of the Advances in Neural Information Processing Systems, 2019

28 Mishchuk A, Mishkin D, Radenovic F, et al. Working hard to know your neighbor's margins: local descriptor learning loss. In: Proceedings of the Advances in Neural Information Processing Systems, 2017

29 Tian Y, Yu X, Fan B, et al. SOSNet: second order similarity regularization for local descriptor learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 11016–11025

30 DeTone D, Malisiewicz T, Rabinovich A. Superpoint: self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018. 224–236

31 Revaud J, Weinzaepfel P, De Souza C, et al. R2D2: repeatable and reliable detector and descriptor. 2019. ArXiv:1906.06195

32 Dusmanu M, Rocco I, Pajdla T, et al. D2-Net: a trainable cnn for joint description and detection of local features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 8092–8101

33 Noh H, Araujo A, Sim J, et al. Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 3456–3465

34 Reiher L, Lampe B, Eckstein L. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In: Proceedings of the 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020. 1–7

35 Liu Y, Wang Y, Wang Y, et al. VectorMapNet: end-to-end vectorized HD map learning. 2022. ArXiv:2206.08920

36 Philion J, Fidler S. Lift, splat, shoot: encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In: Proceedings of the European Conference on Computer Vision, 2020. 194–210

37 Ng M H, Radia K, Chen J, et al. BEV-Seg: bird's eye view semantic segmentation using geometry and semantic point cloud. 2020. ArXiv:2006.11436

38 Reading C, Harakeh A, Chae J, et al. Categorical depth distribution network for monocular 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 8555–8564

39 Hu A, Murez Z, Mohan N, et al. FIERY: future instance prediction in bird's-eye view from surround monocular cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 15273–15282

40 Huang J, Huang G, Zhu Z, et al. BEVDet: high-performance multi-camera 3D object detection in bird-eye-view. 2021. ArXiv:2112.11790

41 Li Y, Ge Z, Yu G, et al. BEVDepth: acquisition of reliable depth for multi-view 3D object detection. 2022. ArXiv:2206.10092

42 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017

43 Yang W, Li Q, Liu W, et al. Projecting your view attentively: monocular road scene layout estimation via cross-view transformation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 15536–15545

44 Li Z, Wang W, Li H, et al. BEVFormer: learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. 2022. ArXiv:2203.17270

45 Pan B, Sun J, Leung H Y T, et al. Cross-view semantic segmentation for sensing surroundings. IEEE Robot Autom Lett, 2020, 5: 4867–4873

46 Roddick T, Cipolla R. Predicting semantic map representations from images using pyramid occupancy networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 11138–11147

47 Li Q, Wang Y, Wang Y, et al. HDMapNet: a local semantic map learning and evaluation framework. 2021. ArXiv:2107.06307

48 Wang S, Fidler S, Urtasun R. Holistic 3D scene understanding from a single geo-tagged image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3964–3972

49 Brubaker M A, Geiger A, Urtasun R. Lost! Leveraging the crowd for probabilistic visual self-localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013. 3057–3064

50 Ma W-C, Wang S, Brubaker M A, et al. Find your way by observing the sun and other semantic cues. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2017. 6292–6299

51 Levinson J, Montemerlo M, Thrun S. Map-based precision vehicle localization in urban environments. In: Robotics: Science and Systems. Cambridge: MIT Press, 2007

52 Levinson J, Thrun S. Robust vehicle localization in urban environments using probabilistic maps. In: Proceedings of the

IEEE International Conference on Robotics and Automation, 2010. 4372–4378

53 Schönberger J L, Pollefeys M, Geiger A, et al. Semantic visual localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 6896–6906

54 Welzel A, Reisdorf P, Wanielik G. Improving urban vehicle localization with traffic sign recognition. In: Proceedings of the 18th International Conference on Intelligent Transportation Systems, 2015. 2728–2732

55 Qu X, Soheilian B, Paparoditis N. Vehicle localization using mono-camera and geo-referenced traffic signs. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2015. 605–610

56 Gao J, Sun C, Zhao H, et al. VectorNet: encoding HD maps and agent dynamics from vectorized representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 11525–11533

57 Suhr J K, Jang J, Min D, et al. Sensor fusion-based low-cost vehicle localization system for complex urban environments. IEEE Trans Intell Transp Syst, 2016, 18: 1078–1086

58 Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision, 2020. 213–229

59 Ba J L, Kiros J R, Hinton G E. Layer normalization. 2016. ArXiv:1607.06450

60 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

61 Caesar H, Bankiti V, Lang A H, et al. nuScenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 11621–11631

62 Chong Z J, Qin B, Bandyopadhyay T, et al. Synthetic 2D LiDAR for precise vehicle localization in 3D urban environment. In: Proceedings of the IEEE International Conference on Robotics and Automation, 2013. 1554–1559

63 Morris Garages. The MG MARVEL R Electric. 2022. https://www.mgmotor.eu/configurator/marvel-r

64 Tan M, Le Q. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning, 2019. 6105–6114

65 Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis, 2015, 115: 211–252

66 Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019

67 Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980

68 Choi M J, Suhr J K, Choi K, et al. Low-cost precise vehicle localization using lane endpoints and road signs for highway situations. IEEE Access, 2019, 7: 149846

69 Pauls J-H, Petek K, Poggenhans F, et al. Monocular localization in HD maps by combining semantic segmentation and distance transform. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020. 4595–4601

70 Wang H, Xue C, Zhou Y, et al. Visual semantic localization based on HD map for autonomous vehicles in urban scenarios. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2021. 11255–11261