2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

EgoVM: Achieving Precise Ego-Localization using Lightweight Vectorized Maps

Yuzhe He, Shuang Liang, Xiaofei Rui, Chengying Cai and Guowei Wan*

***Abstract*— Accurate and reliable ego-localization is critical for autonomous driving. In this paper, we present EgoVM, an end-to-end localization network that achieves comparable localization accuracy to prior state-of-the-art methods, but uses lightweight vectorized maps instead of heavy point-based maps. To begin with, we extract BEV features from online multi-view images and LiDAR point cloud. Then, we employ a set of learnable semantic embeddings to encode the semantic types of map elements and supervise them with semantic segmentation, to make their feature representation consistent with BEV features. After that, we feed map queries, composed of learnable semantic embeddings and coordinates of map elements, into a transformer decoder to perform cross-modality matching with BEV features. Finally, we adopt a robust histogram-based pose solver to estimate the optimal pose by searching exhaustively over candidate poses. We comprehensively validate the effectiveness of our method using both the nuScenes dataset and a newly collected dataset. The experimental results show that our method achieves centimeter-level localization accuracy, and outperforms existing methods using vectorized maps by a large margin. Furthermore, our model has been extensively tested in a large fleet of autonomous vehicles under various challenging urban scenes.**

I. INTRODUCTION

Online high-definition (HD) maps derived from bird's eye view (BEV) models have been examined by several studies [1]. However, these maps may suffer from stability and robustness issues due to obstacle occlusions, road abrasion, and inadequate model capabilities, *etc*., which cannot meet the high quality standards of fully autonomous driving. Therefore, HD maps remain an indispensable component of fully autonomous vehicles as they can provide comprehensive and detailed information about road infrastructure. To utilize HD maps as priors, centimeter-level ego-localization is necessary [2].

Several methods have achieved this goal with the aid of 3D light detection and ranging (LiDAR) scanners [3]. In these methods, localization maps are typically represented as points, voxels, or Gaussian distributions on 2D grids, which require enormous storage on the vehicle. This poses significant challenges to deploy the map covering vast areas onto the vehicle. To reduce the map size, several works propose to use pole-like objects, vertical corners, or footprints and surfaces of buildings [4] as map features. Most of the camera-based methods [5] depend on HD maps that contain lane lines, road markings, poles, traffic signs, *etc*.. Despite the significant progress made by these methods over the years, they still fall behind LiDAR-based techniques in terms of performance. End-to-end localization networks [6]–[8] have demonstrated their potential for enhancing localization accuracy, scene generalization, and map compression in recent years. However, these networks also face some limitations, such as the large map size [6], the model-map dependency [8], and the only applicability for simple scenes [9].

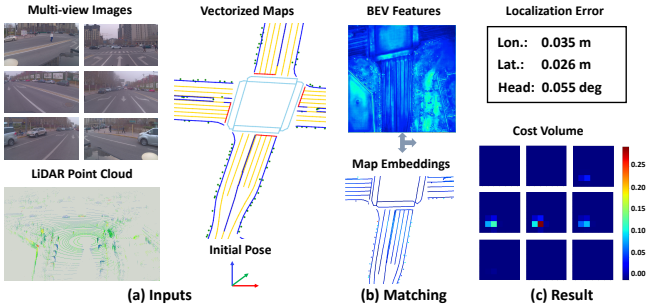

Fig. 1. The illustration of our method. (a) The inputs consist of multi-view images, LiDAR point cloud and vectorized maps. (b) The BEV feature extraction module extracts BEV features, which are then interacted with vectorized maps to generate map embeddings by the cross-modality matching module. (c) The pose solver module takes BEV features and map embeddings as inputs and accomplishes centimeter-level localization.

A practical localization system for fully autonomous driving should meet the following requirements. First, it should achieve **high accuracy**, with the horizontal localization error and the heading error being within a few centimeters and a few tenths of a degree, respectively. Second, it should ensure **high reliability**, with the ability to maintain centimeter-level accuracy in various complex urban scenes. Third, it should have **good scalability**, with the capability to generate and update the localization map efficiently and massively. Moreover, the localization map should have low storage demand to enable onboard deployment, and the model should be upgradeable without updating the map.

To achieve the above goals, we propose a novel ego-localization network, dubbed **EgoVM** (**Ego**-localization using **V**ectorized **M**aps). EgoVM takes multi-view images from surrounding cameras, 3D points from LiDAR sensor, and offline vectorized maps as inputs, and estimates the pose offset relative to the initial pose in an end-to-end manner. Fig. 1 shows the illustration of our method. In our method, we only estimate 2D position and heading offsets of a 6-Dof pose as in [10]. Generating BEV features from multi-view images, 3D LiDAR points or both has been applied in 3D object detection [11] and semantic segmentation

This work is supported by Baidu Autonomous Driving Technology Department (ADT).

The authors are with Baidu ADT, Beijing 100094, P. R. China, {heyuzhe, liangshuang18, ruixiaofei, caichengying, wanguowei}@baidu.com.

*Author to whom correspondence should be addressed, E-mail: wanguowei@baidu.com

[12]. Therefore, we attempt to perform matching between vectorized maps and BEV features to estimate 3-DoF pose (2D position and heading) offset in BEV space. However, they are still different modalities and difficult to compare, even if they are transformed into a canonical view. To address this issue, we employ a set of learnable embeddings to describe the *semantic types* of vectorized map elements, which include *lane line*, *pedestrian crossing*, *road marking*, *pole* and *traffic sign*, *etc*.. The learnable embeddings are supervised by semantic segmentation to make their features consistent with BEV features. Then, vectorized map elements, represented by learnable semantic embeddings and geometric coordinates, interact with BEV features through a transformer decoder for cross-modality matching. Finally, we apply a robust histogram-based pose solver [6] to determine the optimal pose offset by searching exhaustively over 3-DoF candidate poses.

In summary, our main contributions are:

- An end-to-end localization network that uses light vectorized maps, which achieves centimeter-level localization accuracy comparable to those methods that use heavy point-based maps and is far superior to existing methods that use vectorized maps.
- A novel design for cross-modality matching, which adopts a set of learnable semantic embeddings and a transformer decoder to bridge the representation gap between vectorized maps and BEV features.
- Comprehensive tests and detailed ablation analysis on real-world datasets to verify the effectiveness of the proposed method.
- Integration with other sensors (GNSS, IMU) for a multi-sensor fusion localization system that has been extensively tested in various challenging urban scenes.

II. RELATED WORK

A. Localization Using Point-based Maps.

Methods that align multiple passes of LiDAR point clouds over the same area to construct accurate maps and match the online sensory input with the map have been widely used in fully autonomous vehicles. The pioneering works [10] represent maps using LiDAR intensities, which could provide texture information about the environment. Subsequent works [13] combine altitude information with intensities to achieve more robust and accurate localization. The work of R. Wolcott *et al*. [3] employs Gaussian Mixture Model (GMM) to describe intensity and altitude information, which works well in snowy scenes and has the potential to handle overpass situations. H. Liu *et al*. [14] proposes to extract low-level semantic segmentation-based features, including ground, road-curb, surface, and edge. These methods have achieved impressive performance, but their high storage requirements limit their applicability. In contrast, our approach requires only lightweight vectorized maps with very little storage.

B. Localization Using Vectorized or Semantic Maps.

A compact representation can be achieved by utilizing lightweight vectorized maps, which contain geometric and semantic information of the scene, such as lane lines, road markings, poles, traffic signs, and their combinations [15]. In the case of lane lines and road markings, the matching process can be performed either from perspective or from bird's eye view. The majority of methods perform matching from perspective view when poles or traffic signs are involved. Several recent methods propose using coarse-to-fine strategy [16], distance transform [17], or robust data association [18] to improve the robustness and availability of localization systems. Our method aims to incorporate vectorized maps to achieve accurate and reliable centimeter-level localization for fully autonomous driving due to their compactness. In contrast to prior work, we combine the lightweight vectorized map and BEV perception to achieve an high-precision end-to-end localization framework.

C. End-to-end Localization Networks.

End-to-end localization networks that compute similarity between online LiDAR sweeps and intensity map [19] or keypoint map [6] have been explored. They both apply a 3D cost volume to estimate horizontal and heading offsets in an exhaustive searching way. A similar solution [8] exploits long-term salient, distinctive, and stable features to achieve centimeter-level visual localization. X. Wei *et al*. [7] proposes to learn to compress the map without loss of localization accuracy. However, due to deep features residing in the map, the map must be updated for model upgrades, which does not facilitate map deployment and model iteration. Differently, we depend on only the vectorized map independent of the model. W. Ma *et al*. [9] exploits lanes and traffic signs to localize against a sparse semantic map that requires orders of magnitude less storage than previous approaches. However, different from us, their method depends on detection network and is not end-to-end trained. Zhang *et al*. [20] proposes an end-to-end visual localization method based on HD Map and BEV representation. They encode the entire landmark instance and estimate pose via regression, whereas we encode each vectorized segment part, estimate pose based on a cost volume, and adopt semantic embedding and semantic supervision, which is more interpretable.

D. BEV Feature Extraction.

Since prediction and planning tasks operate under bird's eye view, many methods have attempted to generate BEV features from multi-view images and perform perception task. One series of methods first performs monocular depth estimation, and then lifts 2D image features to 3D space and splats to BEV [21]. Another series of methods employs transformer [22] to perform view transformation [11]. There are also several LiDAR-camera fusion strategies. One is point decoration fusion, which acquires image features or semantic scores for LiDAR points and then generates BEV features, *e.g*., PointPainting [23]. A second approach is to fuse the LiDAR BEV features and image BEV features directly under

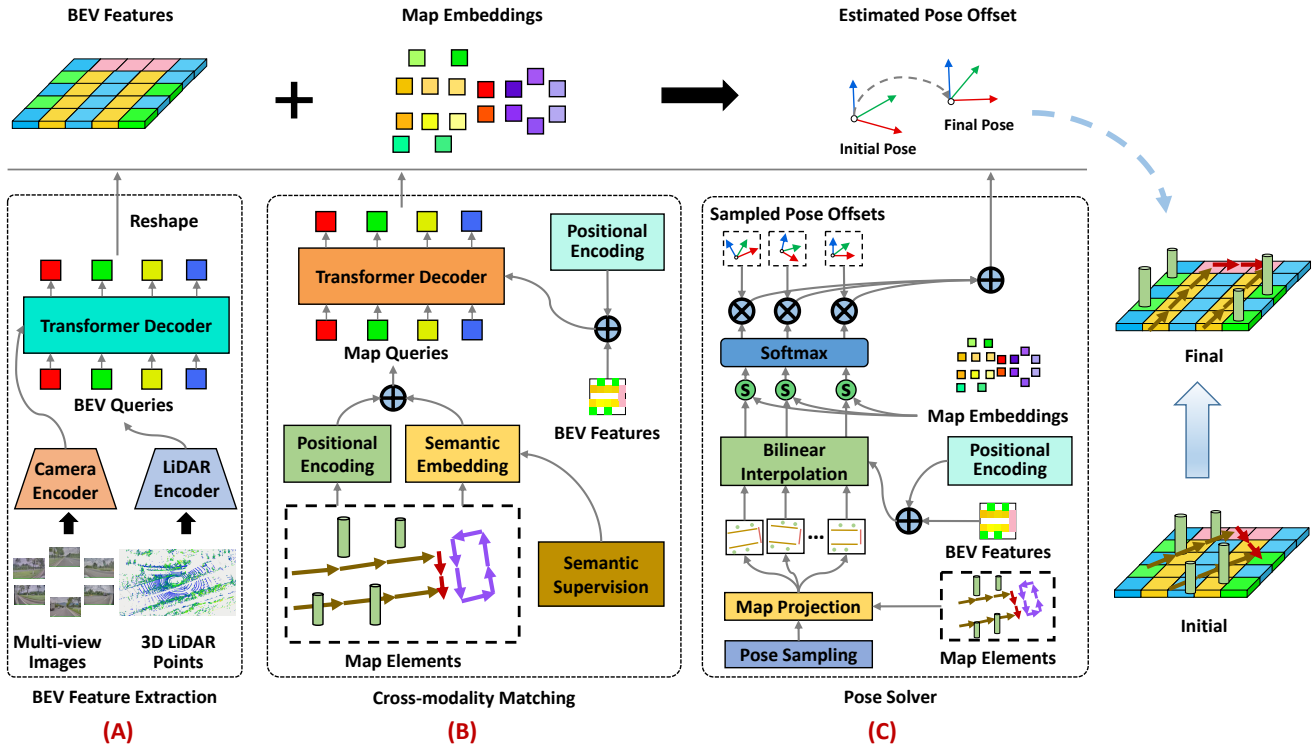

Fig. 2. The network architecture of EgoVM. First, the multi-view images and 3D LiDAR points are fed into camera encoder and LiDAR encoder respectively, and fused to obtain BEV features. Second, vectorized map elements and BEV features are used as the queries and keys / values of the transformer decoder for cross-modality matching, thus generating map embeddings. Third, map embeddings are compared with interpolated map features by candidate poses to calculate their similarities, so as to estimate the optimal pose offset.

the BEV representation, *e.g.*, BEVFusion [24]. The last kind adopts transformer to perform multi-modality fusion, *e.g.*, DeepFusion [25].

III. PROBLEM FORMULATION

Our goal is to estimate an optimal pose offset given an online point cloud, multi-view images, pre-built vectorized maps, and an initial pose. The pre-built maps contain vectorized elements such as lane lines, road boundaries, pedestrian crossings, stop lines, road markings, traffic signs, and poles, denoted as $\mathbf{M} = \{M_i | i = 1, 2, \cdots, K\}$, where M_i is the i-th vectorized map element. Specifically, lane line, road boundary and stop line are described as horizontal segment on the BEV plane, expressed as endpoints $(x_s, y_s, x_e, y_e) \in \mathbb{R}^4$. The pedestrian crossing is represented by segments of adjacent endpoints of a polygon with $(x_i, y_i, x_{i+1}, y_{i+1}) \in \mathbb{R}^4$. The traffic sign and pole are vectorized as $(x, y, 0, h) \in \mathbb{R}^4$, where (x, y) and h are the center point and height, respectively. Our model also takes an initial pose as an input, which can be provided by a multi-sensor fusion localization system. The estimated pose offset only consists of the 2D horizontal and heading offsets represented as $\Delta T = (\Delta x, \Delta y, \Delta \psi)$, following classical LiDAR-based localization methods [3], [10], [13].

IV. METHOD

Fig. 2 shows three parts of EgoVM. In the first step, denoted as (A) in the figure, the multi-view images and 3D LiDAR points are fed into a camera encoder and LiDAR encoder respectively, and fused using a transformer decoder to obtain BEV features. Next, a set of learnable embeddings encoding map element types, supervised by semantic segmentation, interact with BEV features through another transformer decoder to perform cross-modality matching in (B), thus obtaining map embeddings. Third, the pose solver samples several candidate poses to project map elements to BEV plane as shown in (C), obtaining corresponding features by bilinear interpolation, and then compares them with map embeddings to estimate the optimal pose offset.

A. BEV Feature Extraction

We adopt a transformer decoder for fusing image features and LiDAR BEV features. First, multi-view images and LiDAR points are fed into a camera encoder and a LiDAR encoder to extract image features and LiDAR BEV features respectively. Then, the transformer decoder takes LiDAR BEV features to initialize BEV queries and interact with image features, thus obtaining fused BEV features.

Camera Encoder. The multi-view images $\{I_i | i = 1, 2, \cdots, N\}$ are fed into a shared backbone network (*e.g.*, ResNet [26]), followed by FPN [27] to extract multi-scale features, denoted as F^I, where $F^I_{ij} \in \mathbb{R}^{H_j \times W_j \times C}$ is the j-th level feature of i-th camera, and H_j, W_j are the height and width of j-th level feature, C is the feature dimension.

LiDAR Encoder. The 3D LiDAR points $\{P_i | i = 1, 2, \cdots, n\}$ are first fed into a pillar-based feature extractor

(*e.g.*, Pillar Feature Net of PointPillars [28]) to extract pseudo image features. Then, a group of 2D convolutional layers is applied to obtain LiDAR BEV features $F^L \in \mathbb{R}^{H \times W \times C}$ from the pseudo image features, where H and W are the height and width of BEV space.

BEV Fusion. To generate unified LiDAR-camera BEV features, we adopt a transformer decoder to fuse LiDAR BEV features and multi-view image features based on BEVFormer [11]. Specifically, we use LiDAR BEV features to initialize BEV queries, perform self-attention on them, and then apply cross-attention to aggregate multi-view image features. The self-attention and cross-attention layers are implemented based on deformable attention [29] for efficiency. The fused BEV features are denoted as $F^B \in \mathbb{R}^{H \times W \times C}$.

B. Cross-modality Matching

Vectorized map elements differ significantly from BEV features in terms of representation. To match them, we employ a set of learnable embeddings and a transformer decoder to bridge the representation gap. The learnable embeddings encode the semantic types of the map elements and then act as queries for the transformer decoder to interact with BEV features and generate unified features for map elements, denoted as *map embeddings*.

Semantic Embedding. Vectorized map elements have different semantic types, such as *lane line*, *road boundary*, *stop line*, *pedestrian crossing*, *road marking*, *pole* and *traffic sign*. We use a set of learnable embeddings E^{sem} to describe N_e semantic types, where $E^{sem}_j \in \mathbb{R}^C$ is expected to represent j-th semantic type. Correspondingly, the semantic embedding of map element M_i is denoted as $E^{sem}_{s_i}$.

Positional Encoding. Vectorized map elements $\mathbf{M}$ are commonly represented in a global coordinate system, such as Universal Transverse Mercator (UTM) coordinate. Thus, we first normalize M_i to $\widehat{M_i}$ by subtracting XY coordinates of initial pose and dividing by the size of the BEV space. Then, we feed $\widehat{M_i}$ into a shared Multi-Layer Perceptron (MLP) to obtain positional encodings $E^{pos}_i \in \mathbb{R}^C$.

Matching. We apply a transformer decoder to match map elements and BEV features, resulting in map embedding M^{emb} of K map elements. We first initialize the map query $Q_i \in \mathbb{R}^C$ of the map element M_i by adding its semantic embedding and positional encoding:

$$Q_i = E^{sem}_{s_i} + E^{pos}_i. \tag{1}$$

Then, self-attention and cross-attention are applied among map queries and BEV features. Specifically, the self-attention module is formulated as:

$$\mathrm{SA}(Q_i) = \sum_{m=1}^{A} W_m \sum_{i'=1}^{K} A_m(Q_i, Q_{i'}) W'_m Q_{i'}, \tag{2}$$

where A is the number of heads, W_m and W'_m are learnable projection matrices, $A_m(Q_i, Q_{i'})$ is attention weight between map queries Q_i and $Q_{i'}$. The cross-attention module is defined as:

$$\mathrm{CA}(Q_i, F^B) = \mathrm{DA}(Q_i, r^B_i, F^B + B^{pos}), \tag{3}$$

where DA represents deformable attention, $r^B_i \in \mathbb{R}^2$ is a reference point that is acquired by projecting the endpoint of map element M_i to BEV space via the inital pose, and $B^{pos} \in \mathbb{R}^{H \times W \times C}$ is positional encodings of BEV grids by feeding 2D BEV coordinates into a MLP.

Semantic Supervision. To better learn the correspondence between semantic embeddings E^{sem} and BEV features F^B, we predict BEV semantic probabilities of different semantic types using E^{sem} and F^B, which are supervised via ground truth semantic probabilities. Specifically, we take the j-th semantic type as example. The predicted semantic probabilities $P_j \in \mathbb{R}^{H \times W}$ of j-th semantic type is formulated as:

$$P_j(h, w) = \mathrm{sigmoid}\left(F^B(h, w) \odot E^{sem}_j\right), \tag{4}$$

where h and w are the indices of BEV grid, and $\odot$ represents dot product.

To generate the ground truth semantic probabilities for each semantic type j, we do the following steps. First, we project the map elements $\{M_i | s_i = j\}$ to the BEV plane using the ground truth pose. Second, we divide the BEV plane into a grid of size $H \times W$ and assign each cell a value of 0 or 1 depending on whether it is occupied by a map element or not. This gives us a binary matrix $P^{GT}_j \in \mathbb{R}^{H \times W}$ that represents the ground truth semantic probabilities for semantic type j.

C. Pose Solver

Following [6], we utilize a histogram-based pose solver to estimate the optimal pose offset.

Candidate Poses. We sample candidate pose offsets along x, y and yaw dimensions by grid searching, denoted as $\{\Delta T_{pqr} = (\Delta x_p, \Delta y_q, \Delta \psi_r) | 1 \le p, q, r \le N_s\}$, and then generate the candidate poses T_{pqr} by composing the initial pose and them.

Optimal Pose Offset. We project map elements M_i into the BEV plane using a specific candidate pose T_{pqr} to obtain their BEV features $M^{bev}_i(T_{pqr})$ by bilinear interpolation on fused BEV features F^B. Then we calculate the similarity score of F^B and M^{emb} under candidate pose T_{pqr} by:

$$S(T_{pqr}) = \frac{1}{K} \sum_{i=1}^{K} h(M^{bev}_i(T_{pqr}) \odot M^{emb}_i), \tag{5}$$

where $M^{emb}_i \in M^{emb}$ is the map embedding of M_i, and h is a shared MLP. Then, we normalize the similarity scores $\{S(T_{pqr}) | 1 \le p, q, r \le N_s\}$ of all candidate poses by *softmax* to obtain the posterior probability $p(T_{pqr}|X)$ (also denoted as cost volume as shown in Fig.1), where $X = \{F^B, M^{emb}\}$. Finally, we estimate pose offset $\Delta T = E(T|X)$ and covariance $\Sigma = Var(T|X)$ as follows:

$$\Delta T = \sum_{1 \le p,q,r \le N_s} p(T_{pqr}|X) \Delta T_{pqr}. \tag{6}$$

$$\Sigma = \sum_{1 \le p,q,r \le N_s} p(T_{pqr}|X)(\Delta T_{pqr} - \Delta T)(\Delta T_{pqr} - \Delta T)^T. \tag{7}$$

D. Loss Function

RMSE Loss. We define the first loss as the root mean square error (RMSE) between predicted pose offset ΔT and ground truth offset ΔT_{gt}:

$$\mathcal{L}_{rmse} = \|\Lambda^{\frac{1}{2}} U^T (\Delta T - \Delta T_{gt})\|_2, \tag{8}$$

where $\Sigma = U\xi U^T$, and $\Lambda \in \mathbb{R}^{3\times3}$ is a diagonal matrix obtained by normalizing the diagonal elements of ξ^{-1}.

Pose Solver KL loss. The second loss is derived from Kullback-Leibler (KL) divergence $D_{KL}(t(T)||p(T|X))$, which aims to regularize the posterior probability distribution. After dropping constant terms of KL divergence, the KL loss can be obtained:

$$\mathcal{L}_{KL} = -\int t(T) \log p(X|T) dT + \log \int p(X|T) dT, \tag{9}$$

where $t(T)$ and $p(X|T)$ denote the target probability distribution and likelihood function. Note $p(X|T) \propto p(T|X)$, which is defined in the Sec. IV-C.

Following [30], we define $t(T) = \delta(T - T_{gt})$, and $\delta(\cdot)$ is the Dirac delta function. Equation 9 is rewritten as:

$$\mathcal{L}_{KL} = -\log \frac{\exp(S(T_{gt}))}{\int \exp(S(T)) dT}. \tag{10}$$

Based on this, the pose solver KL loss is obtained by Monte Carlo integration:

$$\mathcal{L}^{ps}_{KL} = -\log \frac{\exp(S(T_{gt}))}{\sum_{1\le p,q,r\le N_s} \exp(S(T_{pqr}))}. \tag{11}$$

Random Pose KL Loss. We use a random pose sampling strategy to enhance the supervision further. The sampled poses $T_j (1 \le j \le N_r)$ are drawn from a pose distribution $q(T)$, and the KL loss is calculated as follows:

$$\mathcal{L}^{rp}_{KL} = -\log \frac{\exp(S(T_{gt}))}{\frac{1}{N_r} \sum_{1\le j\le N_r} \frac{1}{q(T_j)} \exp(S(T_j))}, \tag{12}$$

where $q(T)$ is a combination of a 2-DoF multivariate t-distribution on x and y dimensions, and a mixture of von Mises and uniform distribution on yaw dimension.

Semantic Segmentation Loss. We use a semantic segmentation loss function that better supervises semantic embeddings and BEV features by summing up the focal losses (FC) of all semantic classes:

$$\mathcal{L}_{ss} = \sum_{j=1}^{N_e} \mathrm{FC}(P_j, P_j^{GT}). \tag{13}$$

V. MAP EXTENSION

Vectorized maps consist of appearance features, such as lane lines, road markings, stop lines, and pedestrian crossings, and geometric features, such as poles and traffic signs. Localization that relies only on appearance features is prone to degradation in low-light conditions. Geometric features can help to improve the localization performance, but they are sparse and not available in every road section. Therefore, we propose to use *surfels* [31], which are planar features that are rich in the scene, to enhance the geometric features in the map. We represent surfels as $\{(p, n, r) \in \mathbb{R}^7\}$, where $p \in \mathbb{R}^2$ is the center point, $n \in \mathbb{R}^3$ is the norm vector, $r = (\lambda_1/\lambda_2, \lambda_1/\lambda_3)$, and $\lambda_1 < \lambda_2 < \lambda_3$ are the eigenvalues of the covariance matrices of surfels. Surfels are abundant and beneficial for localization, but processing all of them is inefficient. Therefore, we first split the global static point cloud into 3D voxels with size of $0.2m \times 0.2m \times 0.2m$, and extract a single surfel in each voxel. Then we apply eigenvalue and grid sampling to reduce the number of surfels, by discarding surfels with $\lambda_1/\lambda_2 > 0.1$ and retaining only one surfel with the smallest λ_1/λ_2 in each $1m\times1m$ horizonal grid. Like the other vectorized map elements, surfels are encoded as positional encodings by a MLP and then added to corresponding semantic embedding E^{sem}_{surfel} to obtain their map queries. After that, surfel map queries are processed as the same of other kinds of map queries.

VI. EXPERIMENTS

A. Datasets

We evaluate the performance of the proposed network on two datasets: nuScenes [32] and a self-collected dataset. The nuScenes dataset contains more than 28,000 frames for training and 6,000 frames for validation. The self-collected dataset consists of 203,645 frames for training and 81,877 frames for validation. Each frame includes six camera images and one LiDAR point cloud, which are time-aligned, as well as intrinsics and extrinsics, ground truth pose and surrounding map elements. The self-collected dataset was gathered from a road network spanning over 1,000 kilometers and featuring various complex urban scenes. The training data was carefully selected from over 4 million frames of vehicle data collected between February and September 2022, while the validation data was randomly chosen from service vehicle data in December 2022. The dataset comprises diverse scenarios, including day, night, and peak hours. The ground truth poses are obtained via point cloud registration and GNSS RTK/INS post-processing.

Implementation Details. The range of BEV space is $[-40m, 40m]$ in XY dimensions and the resolution is 0.5m, thus $H = W = 160$. The feature and embedding size $C = 256$ and the image feature level $L = 2$, where $H_1 = 28, W_1 = 40, H_2 = 14, W_2 = 20$. The map elements number is $K = 896$. The layer of transformer decoder is 4 and the head number is 4. The number of sampled pose in each dimension $N_s = 7$. The initial pose is obtained by adding uniformly sampled noisy pose offsets within $[-3m, 3m]$ in the X and Y dimensions and $[-3°, 3°]$ in the yaw dimension to the ground truth pose. Besides, we use iterative multi-resolution pose solver to balance the accuracy and efficiency.

B. Performance

Comparison Methods. We compare our method with several state-of-the-art methods, namely MSF-LiDAR [13], DA4AD [8], a structure-based method, a fusion method and BEV-Locator [20]. MSF-LiDAR is a LiDAR-based localization method that models the map as Gaussian distributions.

TABLE I

ACCURACY COMPARISON ON THE SELF-COLLECTED DATASET.

Method	Longitudinal Error			Lateral Error			Yaw Error		
	MAE(m)	RMSE(m)	<0.1m/0.2m/0.3m(%)	MAE(m)	RMSE(m)	<0.1m/0.2m/0.3m(%)	MAE(°)	RMSE(°)	<0.1°/0.3°/0.6°(%)
MSF-LiDAR	0.041	0.052	94.31/99.86/**99.99**	0.045	0.058	91.22/99.84/**99.99**	0.092	0.122	64.02/97.22/99.96
DA4AD	0.089	0.200	76.20/94.14/96.89	0.085	0.167	74.85/93.95/97.06	0.137	0.231	53.56/92.76/98.14
Structure-based	0.318	0.372	14.66/31.26/49.72	0.143	0.191	47.85/74.51/87.74	0.206	0.267	35.33/72.15/97.92
Fusion	0.038	**0.049**	95.17/**99.89**/**99.99**	0.042	0.055	92.55/98.87/**99.99**	0.085	0.114	67.89/97.75/99.97
Ours (Visual)	0.149	0.241	45.91/77.43/91.30	0.073	0.098	72.94/95.85/99.52	0.168	0.215	35.75/86.11/99.02
Ours (LiDAR with surfel)	0.045	0.076	93.10/99.28/99.64	0.048	0.083	90.78/99.67/99.84	0.099	0.139	56.11/99.16/99.89
Ours (LiDAR with surfel and pole)	0.037	0.063	95.72/99.52/99.74	0.041	0.053	94.28/98.89/**99.99**	0.094	0.116	58.10/**99.24**/**99.99**
Ours	**0.035**	0.087	**96.94**/99.70/99.78	**0.033**	**0.043**	**97.44**/**99.93**/**99.99**	**0.080**	**0.106**	**70.27**/98.91/99.89

DA4AD is an end-to-end visual localization method based on a dense 3D feature point map. The structure-based method is a visual localization method that matches the 3D landmarks in the HD Map with the perceived 2D landmarks in the online images. The fusion method is post-fusion of MSF-LiDAR and DA4AD based on covariance weighted average. BEV-Locator is an end-to-end visual localization method that shares some similarities with ours. However, our method differs from it in several aspects such as map query type, semantic supervision, pose solver and so on. Our method has four modes: visual, LiDAR with surfel, LiDAR with surfel and pole and full mode. The visual mode only uses camera images as the input. Two LiDAR modes use surfel and both surfel and pole in the vectorized map, respectively. The full mode uses both camera images and LiDAR point cloud.

Self-Collected Dataset. We conduct experiments on the self-collected dataset and compare our method with MSF-LiDAR, DA4AD, a structure-based method and a fusion method. The comparison results are shown in Table I. The localization accuracy is measured by mean absolute error (MAE), root mean square error (RMSE) and the percentage of localization error within a certain threshold (e.g., 0.1m/0.1°). The best and second best results are highlighted in bold and underline, respectively. Our method achieves the best MAEs of 0.035m, 0.033m and 0.080° in the longitudinal, lateral and yaw dimensions, respectively, which are superior to the other methods. In addition, the percentages of longitudinal, lateral and yaw errors less than 0.3m/0.3m/0.6° are above 99.7%, which indicates the high stability of our method. The post-fusion of MSF-LiDAR and DA4AD is better than two single-modality methods, however, our method is slightly better than the fusion method due to adopting uniform BEV fusion. The accuracy of LiDAR mode with surfel and pole is higher than that of LiDAR mode with only surfel, but lower than that of full mode. Notice that the longitudinal accuracy of our full mode is relatively low. The reason is that in some scenarios, there are few landmarks providing longitudinal constraints in the map, and then our method degrades in the longitudinal dimension. We intend to further explore some new geometric and texture features in our future work.

NuScenes Dataset. As shown in Table II, we compare our visual mode, full mode and BEV-Locator on the nuScenes dataset. The visual mode of our method achieves lower errors of 0.151m, 0.047m, and 0.092° in the longitudinal, lateral and yaw dimensions, respectively, than those of BEV-Locator, which demonstrates the effectiveness of our visual localization method based on HD Map and images. Furthermore, our full mode improves the localization accuracy by incorporating LiDAR point cloud.

TABLE II

ACCURACY COMPARISON ON THE NUSCENES DATASET.

Method	Longitudinal Error		Lateral Error		Yaw Error	
	MAE(m)	<0.3m(%)	MAE(m)	<0.3m(%)	MAE(°)	<0.6°(%)
BEV-Locator	0.178	-	0.076	-	0.510	-
Ours (Visual)	0.151	86.96	0.047	99.64	0.092	99.46
Ours	**0.109**	**94.55**	**0.034**	**99.86**	**0.089**	**99.50**

TABLE III

THE MAP SIZES OF COMPARISON METHODS.

Localization Map Size	MSF-LiDAR	DA4AD	Ours
MB/km	8.36	5.92	**0.35**

Map Size. Table III reports the map sizes of MSF-LiDAR, DA4AD and our method, which are 8.36MB/km, 5.92MB/km, and 0.35MB/km, respectively. Compared with MSF-LiDAR and DA4AD, our method achieves significant map size reduction by 95.8% and 94.1%, respectively, which demonstrates the compactness of our map.

C. Ablations

To evaluate the effectiveness of each component of our method, we perform several ablation studies on our self-collected dataset. To verify the robustness of our method, we increase the difficulty of the dataset by randomly removing some types of landmarks from the map in this section. The results are shown in Table IV.

Surfel. In experiment A1, we remove the surfel landmarks from the map. The errors of A1 are higher than those of our full mode, especially in the longitudinal direction. This is because after removing the pole features randomly in some frames, the longitudinal constraints can only rely on the surfel features, and the removal of surfel features will reduce the longitudinal localization accuracy. This demonstrates the effectiveness of the extended surfel features.

Transformer Decoder. Based on A1, we remove the transformer decoder and directly use the map queries as the map embeddings in the experiment A2. The results show that A2 has larger errors than A1, especially in the lateral direction. This indicates that the self-attention of the map queries and the cross-attention of the map queries and the BEV features in the transformer decoder are crucial for our network.

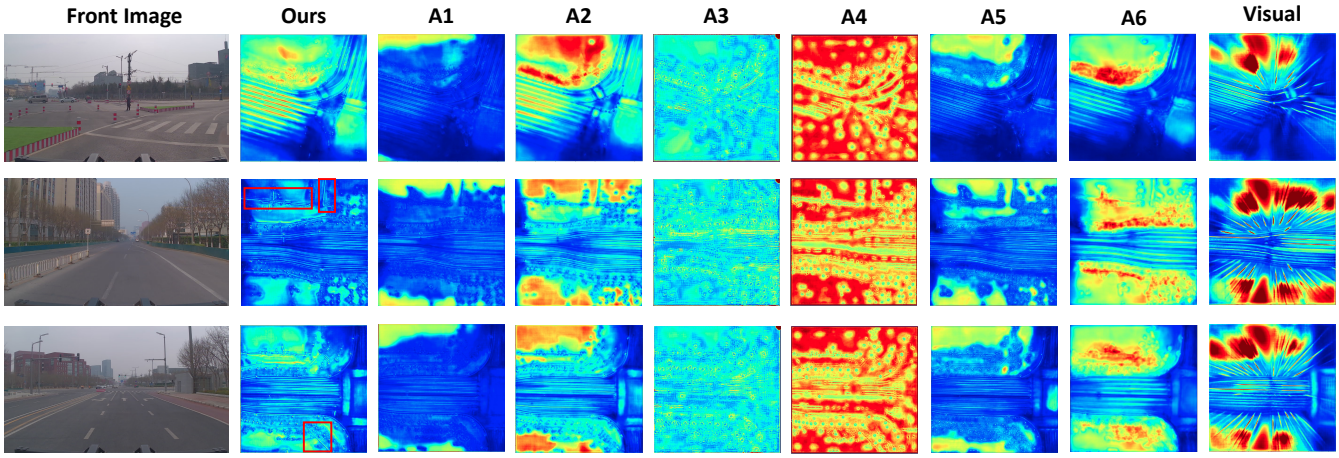

Fig. 3. The three rows illustrate the visualization results of three distinct scenes. The leftmost column displays the front images of different scenes, and the remaining columns depict the BEV features of the ablation experiments and the two modes of our method.

TABLE IV

ABLATION EXPERIMENTS OF THE KEY COMPONENTS OF OUR METHOD ON THE AUGMENTED SELF-COLLECTED DATASET.

	Surfel	Decoder	SemSup	SemEmb	$\mathcal{L}_{KL}^{rp}$	Histogram	Longitudinal Error MAE(m)	Longitudinal Error RMSE(m)	Longitudinal Error <0.1m(%)	Lateral Error MAE(m)	Lateral Error RMSE(m)	Lateral Error <0.1m(%)	Yaw Error MAE(°)	Yaw Error RMSE(°)	Yaw Error <0.1°(%)
Ours	✓	✓	✓	✓	✓	✓	**0.045**	**0.121**	**93.68**	**0.040**	**0.059**	**94.45**	**0.108**	**0.136**	**53.44**
A1	✗	✓	✓	✓	✓	✓	0.092	0.298	90.09	0.044	0.102	93.10	0.109	0.144	53.36
A2	✗	✗	✓	✓	✓	✓	0.093	0.317	83.63	0.068	0.223	86.64	0.115	0.176	52.80
A3	✗	✓	✗	✓	✓	✓	0.094	0.345	89.23	0.057	0.227	92.81	0.123	0.168	49.83
A4	✗	✓	✗	✗	✓	✓	0.099	0.384	89.91	0.050	0.157	92.80	0.157	0.225	44.84
A5	✗	✓	✓	✓	✗	✓	0.093	0.350	89.35	0.049	0.132	91.74	0.121	0.156	48.93
A6	✗	✓	✓	✓	✓	✗	0.116	0.352	77.70	0.085	0.266	83.09	0.161	0.310	45.58

TABLE V

ROAD TESTING EVALUATION OVER 16,500 KM IN A TEST AREA WITH ROAD NETWORK EXCEEDING 1000 KM.

Method	AR	Longitudinal Error MAE(m)	Longitudinal Error RMSE(m)	Longitudinal Error <0.1m/0.2m/0.3m(%)	Lateral Error MAE(m)	Lateral Error RMSE(m)	Lateral Error <0.1m/0.2m/0.3m(%)	Yaw Error MAE(°)	Yaw Error RMSE(°)	Yaw Error <0.1°/0.3°/0.6°(%)
GNSS-RTK	93.67%	0.132	1.082	91.41/93.33/94.12	0.122	0.950	91.26/93.63/94.54	-	-	-
EgoVM	99.82%	0.037	0.104	96.99/99.66/99.77	**0.026**	**0.035**	**98.54/99.93**/99.99	0.131	0.506	42.49/96.29/**99.75**
Integration	**100.0%**	**0.030**	**0.040**	**97.80/99.84/99.96**	**0.026**	0.036	97.90/99.89/**100.0**	**0.122**	**0.192**	**44.41/97.31**/99.74

Semantic Ablations. We remove the semantic supervision (SemSup) from A1 in the experiment A3, and we further replace the semantic embeddings (SemEmb) with semantic encodings as in BEV-Locator in the experiment A4. The results show that A3 and A4 both have larger errors than A1, particularly in the lateral direction. This shows that our network benefits from the combination of semantic embeddings and semantic supervision.

Random Pose KL Loss. The purpose of experiment A5 is to test the effectiveness of our proposed random pose KL loss $\mathcal{L}_{KL}^{rp}$. Compared with A1, the localization errors of A5 increase for all evaluation metrics, which shows that the random pose KL loss can improve the performance.

Pose Solver. We conduct the experiment A6 to examine the effect of the histogram-based pose solver, in which we use a regression-based pose solver instead of the histogram-based one in A1. The accuracy of the regression-based approach is lower than that of A1. Moreover, the regression-based pose solver lacks interpretability and is hard to debug.

BEV Feature Maps Visualization. To better understand the contribution of each key component, we visualize the BEV features of different modes and ablations of our method. Fig. 3 shows the front image and the BEV features. It can be observed that the BEV features of ours, A1, A2, A5, A6 and visual mode with semantic segmentation supervision have clear semantic elements such as lanes, curbs, crosswalks and poles, while the BEV features of A3 and A4 are blurry. This indicates that the semantic supervision enhances the quality and interpretability of the BEV features. Moreover, surfel features (e.g. building surfaces) are also distinctly learned in the BEV features of our full mode marked by red rectangular boxes. In contrast, the BEV features of the visual mode have inaccurate pole positions due to the erroneous depth estimation, which accounts for the large longitudinal error of the visual mode.

D. Road Testing Evaluation

We build a multi-sensor fusion localization system based on error-state kalman filter (ESKF) that integrates EgoVM, GNSS-RTK, and inertial navigation, and deploy it to a fleet of RoboTaxi vehicles for road testing evaluation. We built a road testing dataset over 16,500 kilometers using autonomous driving data recorded in a complex urban area with over 1000 kilometers of road network. Table V compares the localization accuracy of GNSS-RTK, EgoVM and integration methods. We use four metrics: MAE, RMSE, percentage of errors within certain thresholds, and available ratio (AR). AR

is a new metric that measures the percentage of cases where the longitudinal, lateral and yaw errors are simultaneously less than some thresholds. Considering the accuracy requirement of autonomous driving, we select thresholds of 0.6m, 0.3m and 1° for longitudinal, lateral and yaw respectively. The integration method achieves 100% AR by combining the measurements of GNSS-RTK and inertial navigation, while EgoVM sometimes fails to provide accurate longitudinal localization due to the lack of landmarks in some degraded scenes. Moreover, by the time of paper submission, our system has been tested in autonomous close-loop mode for more than 5,000,000 kilometers.

VII. CONCLUSION

We have proposed EgoVM, a novel end-to-end localization network that can improve localization accuracy to the centimeter level with lightweight vectorized maps in various challenging urban scenes. We design a cross-modality matching module comprising learnable semantic embeddings supervised by semantic segmentation and a transformer decoder, which enhances the matching performance by transforming the two input modalities into a unified representation. Moreover, we further improve the localization performance by incorporating LiDAR geometric features, which compensate for the deficiency of appearance features in certain scenes. We have integrated our model with GNSS and IMU sensors to form a multi-sensor fusion localization system and have deployed it to a large fleet of autonomous vehicles, demonstrating its commercial viability.

REFERENCES

[1] Y. Liu, Y. Yuantian, Y. Wang, *et al.*, "VectorMapNet: End-to-end vectorized hd map learning," *arXiv preprint arXiv:2206.08920*, 2022.

[2] C. Badue, R. Guidolini, R. V. Carneiro, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[3] R. W. Wolcott and R. M. Eustice, "Robust LiDAR localization using multiresolution gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 3, pp. 292–319, 2017.

[4] E. Javanmardi, Y. Gu, M. Javanmardi, *et al.*, "Autonomous vehicle self-localization based on abstract map and multi-channel lidar in urban area," *IATSS Research*, vol. 43, no. 1, pp. 1–13, 2019.

[5] K. Petek, K. Sirohi, D. Büscher, *et al.*, "Robust monocular localization in sparse hd maps leveraging multi-task uncertainty estimation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4163–4169.

[6] W. Lu, Y. Zhou, G. Wan, *et al.*, "L3-Net: Towards learning based LiDAR localization for autonomous driving," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019.

[7] X. Wei, I. A. Bârsan, S. Wang, *et al.*, "Learning to localize through compressed binary maps," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 308–10 316.

[8] Y. Zhou, G. Wan, S. Hou, *et al.*, "DA4AD: End-to-end deep attention-based visual localization for autonomous driving," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020, pp. 271–289.

[9] W.-C. Ma, I. Tartavull, I. A. Bârsan, *et al.*, "Exploiting sparse semantic hd maps for self-driving vehicle localization," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5304–5311.

[10] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 4372–4378.

[11] Z. Li, W. Wang, H. Li, *et al.*, "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.

[12] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] G. Wan, X. Yang, R. Cai, *et al.*, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2018, pp. 4670–4677.

[14] H. Liu, Q. Ye, H. Wang, *et al.*, "A precise and robust segmentation-based lidar localization system for automated urban driving," *Remote Sensing*, vol. 11, no. 11, 2019.

[15] R. P. D. Vivacqua, M. Bertozzi, P. Cerri, *et al.*, "Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 19, no. 2, p. 582–597, feb 2018.

[16] C. Guo, M. Lin, H. Guo, *et al.*, "Coarse-to-fine semantic localization with hd map for autonomous driving in structural scenes," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1146–1153.

[17] J.-H. Pauls, K. Petek, F. Poggenhans, *et al.*, "Monocular localization in hd maps by combining semantic segmentation and distance transform," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4595–4601.

[18] W. Cheng, S. Yang, M. Zhou, *et al.*, "Road mapping and localization using sparse semantic visual features," *IEEE Robotics and Automation Letters (RAL)*, vol. 6, no. 4, pp. 8118–8125, 2021.

[19] I. A. Bârsan, S. Wang, A. Pokrovsky, *et al.*, "Learning to localize using a lidar intensity map," in *Proc. of the 2nd Conference on Robot Learning (CoRL)*, 2018.

[20] Z. Zhang, M. Xu, W. Zhou, *et al.*, "BEV-Locator: An end-to-end visual semantic localization network using multi-view images," *arXiv preprint arXiv:2211.14927*, 2022.

[21] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.

[22] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Proc. of the 31nd International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[23] S. Vora, A. H. Lang, B. Helou, *et al.*, "PointPainting: Sequential fusion for 3d object detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] Z. Liu, H. Tang, A. Amini, *et al.*, "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," *CoRR*, vol. abs/2205.13542, 2022.

[25] Y. Li, A. W. Yu, T. Meng, *et al.*, "DeepFusion: Lidar-camera deep fusion for multi-modal 3d object detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 182–17 191.

[26] K. He, X. Zhang, S. Ren, *et al.*, "Deep residual learning for image recognition," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[27] T.-Y. Lin, P. Dollár, R. Girshick, *et al.*, "Feature pyramid networks for object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[28] A. H. Lang, S. Vora, H. Caesar, *et al.*, "PointPillars: Fast encoders for object detection from point clouds," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 689–12 697, 2019.

[29] X. Zhu, W. Su, L. Lu, *et al.*, "Deformable DETR: deformable transformers for end-to-end object detection," in *Proc. of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[30] H. Chen, P. Wang, F. Wang, *et al.*, "EPro-PnP: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2771–2780.

[31] H. Pfister, M. Zwicker, J. Van Baar, *et al.*, "Surfels: Surface elements as rendering primitives," in *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 335–342.

[32] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.