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

SuperFusion: Multilevel LiDAR-Camera Fusion for Long-Range HD Map Generation

Hao Dong* Weihao Gu* Xianjing Zhang Jintao Xu Rui Ai Huimin Lu Juho Kannala Xieyuanli Chen

***Abstract*— High-definition (HD) semantic map generation of the environment is an essential component of autonomous driving. Existing methods have achieved good performance in this task by fusing different sensor modalities, such as LiDAR and camera. However, current works are based on raw data or network feature-level fusion and only consider short-range HD map generation, limiting their deployment to realistic autonomous driving applications. In this paper, we focus on the task of building the HD maps in both short ranges, i.e., within 30 m, and also predicting long-range HD maps up to 90 m, which is required by downstream path planning and control tasks to improve the smoothness and safety of autonomous driving. To this end, we propose a novel network named SuperFusion, exploiting the fusion of LiDAR and camera data at multiple levels. We use LiDAR depth to improve image depth estimation and use image features to guide long-range LiDAR feature prediction. We benchmark our SuperFusion on the nuScenes dataset and a self-recorded dataset and show that it outperforms the state-of-the-art baseline methods with large margins on all intervals. Additionally, we apply the generated HD map to a downstream path planning task, demonstrating that the long-range HD maps predicted by our method can lead to better path planning for autonomous vehicles. Our code and self-recorded dataset have been released at https://github.com/haomo-ai/SuperFusion.**

I. Introduction

Detecting street lanes and generating semantic high-definition (HD) maps are essential for autonomous vehicles to achieve self-driving. The HD map consists of semantic layers with lane boundaries, road dividers, pedestrian crossings, etc., which provide precise location information about nearby infrastructure, roads, and environments to navigate autonomous vehicles safely [9].

The traditional way builds the HD maps offline by firstly recording point clouds, then creating globally consistent maps using SLAM [26], and finally manually annotating semantics in the maps. Although some autonomous driving companies have created accurate HD maps following such a paradigm, it requires too much human effort and needs continuous updating. Since autonomous vehicles are typically equipped with various sensors, exploiting the onboard sensor data to build local HD maps for online applications attracts much attention. Existing methods usually extract lanes and crossings on the bird's-eye view (BEV) representation of either camera data [28] or LiDAR data [15]. Recently, several methods [15], [18], [22] show advances in fusing multi-sensor modalities. They leverage the complementary information from both sensors to improve the HD map generation performance. Albeit improvements, existing methods fuse LiDAR and camera data in simple ways, either on the raw data level [25], [24], feature level [1], [32], or final BEV level [15], [18], [22], which do not fully exploit the advantages from both modalities. Besides, existing methods only focus on short-range HD map generation due to the limited sensor measurement range, *i.e.*, within 30 m, which limits their usage in downstream applications such as path planning and motion control in real autonomous driving scenarios. As shown in Fig. 1, when the generated HD map is too short, the planning method may create a non-smooth path that requires frequent replanning due to limited perception distances, or even a path that intersects with the sidewalk. This can lead to frustration for users, as rapidly changing controls can degrade their comfort level.

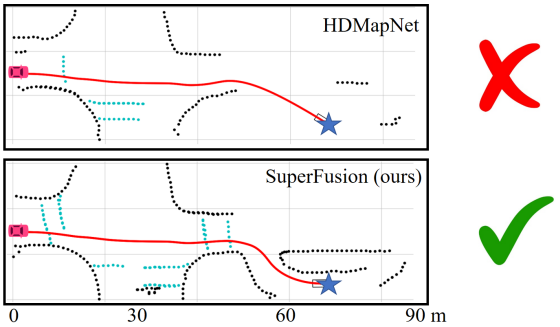


Fig. 1: Long-range HD map generation for path planning. The red car represents the current position of the car, and the blue star is the goal. The upper figure shows that the baseline method only generates short-range HD maps, leading to lousy planning results. The lower one shows that our SuperFusion generates accurate HD maps in both short and long ranges, which serves online path planning well for autonomous driving.

To tackle the problem mentioned above, in this paper, we propose a multilevel LiDAR-camera fusion method, dubbed SuperFusion. It fuses the LiDAR and camera data at three different levels. In the data-level fusion, it combines the projected LiDAR data with images as the input of the camera encoder and uses LiDAR depth to supervise the camera-to-BEV transformation. The feature-level fusion uses camera features to guide the LiDAR features on long-range LiDAR BEV feature prediction using a cross-attention mechanism. In the final BEV-level fusion, our method exploits a BEV alignment module to align and fuse camera and LiDAR BEV

H. Dong is with ETH Zürich. W. Gu, X. Zhang, J. Xu, and R. Ai are with HAOMO.AI. H. Lu and X. Chen are with National University of Defense Technology. J. Kannala is with Aalto University.
Corresponding author: Xieyuanli Chen (xieyuanli.chen@nudt.edu.cn)
*These authors contributed equally.
This work was partially supported by the HAOMO.AI company, Fund for key Laboratory of Space Flight Dynamics Technology (Num 2022-JYAPAF-F1028), and Young Elite Scientists Sponsorship Program by CAST (No. 2023QNRC001).

features. Using our proposed multilevel fusion strategy, SuperFusion generates accurate HD maps in the short range and also predicts accurate semantics in the long-range distances, where the raw LiDAR data may not capture. We thoroughly evaluate our SuperFusion and compare it with the state-of-the-art methods on the publically available nuScenes dataset and our own dataset recorded in real-world self-driving scenarios. The experimental results consistently show that our method outperforms the baseline methods significantly by a large margin on all intervals. Furthermore, we provide the application results of using our generated HD maps for path planning, showing the superiority of our proposed fusion method for long-range HD map generation.

Our contributions can be summarized as: i) our proposed novel multilevel LiDAR-camera fusion network fully leverages the information from both modalities and generates high-quality fused BEV features to support different tasks; ii) our SuperFusion surpasses the state-of-the-art fusion methods in both short-range and long-range HD map generation by a large margin; iii) to the best of our knowledge, our work is the first to achieve long-range HD map generation, *i.e.*, up to 90 m, benefiting the autonomous driving downstream planning task. We will release our code and a new dataset for evaluating long-range HD map generation tasks.

II. Related Work

LiDAR-Camera Fusion. The existing fusion strategies can be divided into three levels: data-level, feature-level, and BEV-level fusion. Data-level fusion methods [25], [24], [31], [17] project LiDAR point clouds to images using the camera projection matrix. The projected sparse depth map can be fed to the network with the image data [25], [24] or decorated with image semantic features [31], [17] to enhance the network inputs. Feature-level fusion methods [1], [32] incorporate different modalities in the feature space using transformers. They first generate LiDAR feature maps, then query image features on those LiDAR features using cross-attention, and finally concatenate them together for downstream tasks. BEV-level fusion methods [15], [18], [22] extract LiDAR and image BEV features separately and then fuse the BEV features by concatenation [15] or fusion modules [18], [22]. For example, HDMapNet [15] uses MLPs to map PV features to BEV features for the camera branch and uses PointPillars [14] to encode BEV features in the LiDAR branch. Recent BEVFusion works [18], [22] use LSS [28] for view transformation in the camera branch and VoxelNet [33] in the LiDAR branch and finally fuse them via a BEV alignment module. Unlike them, our method combines all three-level LiDAR and camera fusion to fully exploit the complementary attributes of these two sensors.

HD Map Generation. The traditional way of reconstructing HD semantic maps is to aggregate LiDAR point clouds using SLAM algorithms [26] and then annotate manually, which is laborious and difficult to update. HDMapNet [15] is a pioneer work on local HD map construction without human annotations. It fuses LiDAR and six surrounding cameras in BEV space for semantic HD map generation. Besides that, VectorMapNet [21] represents map elements as a set of polylines and models these polylines with a set prediction framework, while Image2Map [30] utilizes a transformer to generate HD maps from images in an end-to-end fashion. Several works [8], [10], [3] also detect specific map elements such as lanes. Previous works only segment maps in a short range, usually less than 30 m. Our method is the first work focusing on long-range HD map generation up to 90 m.

III. Methodology

A. Depth-Aware Camera-to-BEV Transformation

We first fuse the LiDAR and camera at the raw data level and leverage the depth information from LiDAR to help the camera lift features to BEV space. To this end, we propose a depth-aware camera-to-BEV transformation module, as shown in Fig. 2. It takes an RGB image $\mathbf{I}$ with the corresponding sparse depth image $\mathbf{D_{sparse}}$ as input. Such sparse depth image $\mathbf{D_{sparse}}$ is obtained by projecting the 3D LiDAR point cloud $\mathbf{P}$ to the image plane using the camera projection matrix. The camera backbone has two branches. The first branch extracts 2D image features $\mathbf{F} \in \mathbb{R}^{W_F \times H_F \times C_F}$, where W_F, H_F and C_F are the width, height and channel numbers. The second branch connects a depth prediction network, which estimates a categorical depth distribution $\mathbf{D} \in \mathbb{R}^{W_F \times H_F \times D}$ for each element in the 2D feature $\mathbf{F}$, where D is the number of discretized depth bins. To better estimate the depth, we use a completion method [13] on $\mathbf{D_{sparse}}$ to generate a dense depth image $\mathbf{D_{dense}}$ and discretize the depth value of each pixel into depth bins, which is finally converted to a one-hot encoding vector to supervise the depth prediction network. The final frustum feature grid $\mathbf{M}$ is generated by the outer product of $\mathbf{D}$ and $\mathbf{F}$ as

$$\mathbf{M}(u, v) = \mathbf{D}(u, v) \otimes \mathbf{F}(u, v), \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{W_F \times H_F \times D \times C_F}$. Finally, each voxel in the frustum is assigned to the nearest pillar and a sum pooling is performed as in LSS [28] to create the camera BEV feature $\mathbf{C} \in \mathbb{R}^{W \times H \times C_F}$.

Our proposed depth-aware camera-to-BEV module differs from the existing depth prediction methods [28], [29]. The depth prediction in LSS [28] is only implicitly supervised by the semantic segmentation loss, which is not enough to generate accurate depth estimation. Different from that, we utilize the depth information from LiDAR as supervision. CaDDN [29] also uses LiDAR depth for supervision but without LiDAR as input, thus unable to generate a robust and reliable depth estimation. Our method uses both the completed dense LiDAR depth image for supervision and also the sparse depth image as an additional channel to the RGB image. In this way, our network exploits both a depth prior and an accurate depth supervision, thus generalizing well to different challenging environments.

B. Image-Guided LiDAR BEV Prediction

In the LiDAR branch, we use PointPillars [14] plus dynamic voxelization [34] as the point cloud encoder to generate LiDAR BEV features $\mathbf{L} \in \mathbb{R}^{W \times H \times C_L}$ for each point

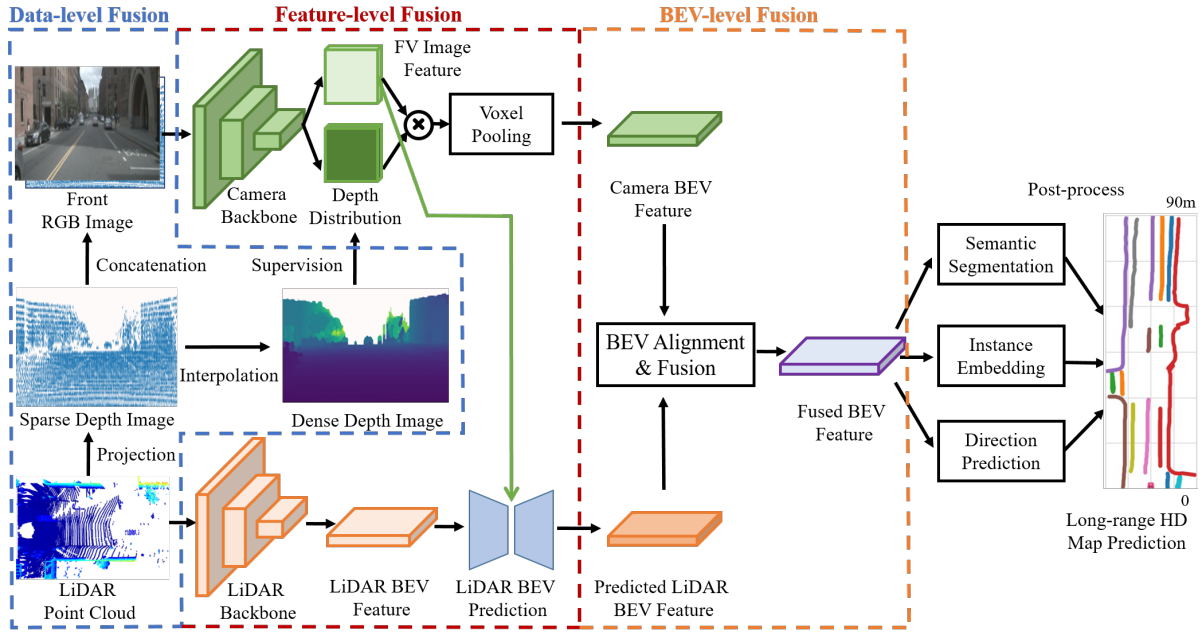
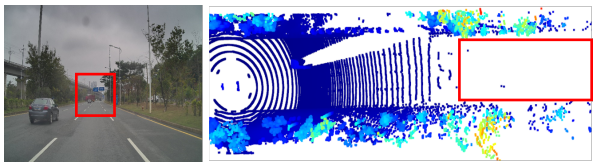
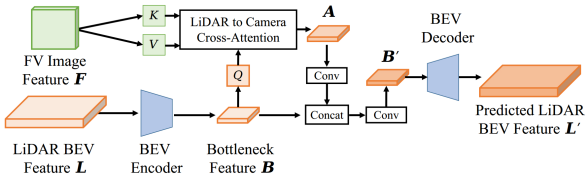


Fig. 2: Pipeline overview of SuperFusion. Our method fuses camera and LiDAR data in three levels: the data-level fusion fuses depth information from LiDAR to improve the accuracy of image depth estimation, the feature-level fusion uses cross-attention for long-range LiDAR BEV feature prediction with the guidance of image features, and the BEV-level fusion aligns two branches to generate high-quality fused BEV features. Finally, the fused BEV features can support different heads, including semantic segmentation, instance embedding, and direction prediction, and finally post-processed to generate the HD map prediction.



(a) The LiDAR usually has a short valid range for the ground plane, while the camera can see a much longer distance.



(b) LiDAR BEV prediction with cross-attention.

Fig. 3: Image-guided LiDAR BEV Prediction.

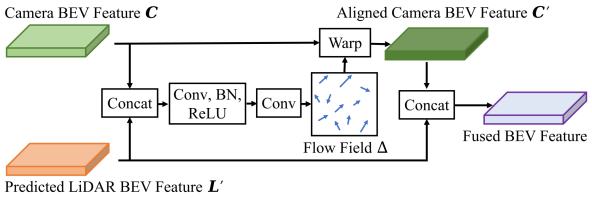


Fig. 4: BEV Alignment and Fusion Module.

cloud $\mathbf{P}$. As shown in Fig. 3a, the LiDAR data only contains a short valid measurement of the ground plane (typically around 30 m for a rotating 32-beam LiDAR), leading many parts of the LiDAR BEV features encoding empty space. Compared to LiDAR, the visible ground area in camera data is usually further. Therefore, we propose a BEV prediction module to predict the unseen areas of the ground for the LiDAR branch with the guidance of image features, as shown in Fig. 3b. The BEV prediction module is an encoder-decoder network. The encoder consists of several convolutional layers to compress the original BEV feature $\mathbf{L}$ to a bottleneck feature $\mathbf{B} \in \mathbb{R}^{W/8 \times H/8 \times C_B}$. We then apply a cross-attention mechanism to dynamically capture the correlations between $\mathbf{B}$ and FV image feature $\mathbf{F}$. Three fully-connected layers are used to transform bottleneck feature $\mathbf{B}$ to query Q and FV image feature $\mathbf{F}$ to key K and value V. The attention affinity matrix is calculated by the inner product between Q and K, which indicates the correlations between each voxel in LiDAR BEV and its corresponding camera features. The matrix is then normalized by a softmax operator and used to weigh and aggregate value V to get the aggregated feature $\mathbf{A}$. This cross-attention mechanism can be formulated as

$$\mathbf{A} = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2)$$

where d_k is the channel dimension used for scaling. We then apply a convolutional layer on the aggregated feature $\mathbf{A}$ to reduce channel, concatenate it with the original bottleneck feature $\mathbf{B}$ and in the end apply another convolutional layer to get the final bottleneck feature $\mathbf{B}'$. Now $\mathbf{B}'$ has the visual guidance from image feature and is fed to the decoder to generate the completed and predicted LiDAR BEV feature $\mathbf{L}'$. By this, we fuse the two modalities at the feature level to better predict the long-range LiDAR BEV features.

C. BEV Alignment and Fusion

So far, we get both the camera and LiDAR BEV features from different branches, which usually have misalignment due to the depth estimation error and inaccurate extrinsic

parameters. Therefore, direct concatenating these two BEV features will result in inferior performance. To better align BEV features, we fuse them at the BEV level and design an alignment and fusion module, as shown in Fig. 4. It takes the camera and LiDAR BEV features as input and outputs a flow field $\mathbf{\Delta} \in \mathbb{R}^{W \times H \times 2}$ for the camera BEV features. The flow field is used to warp the original camera BEV features $\mathbf{C}$ to the aligned BEV features $\mathbf{C}'$ with LiDAR features $\mathbf{L}'$. Following [12], [16], we define the warp function as

$$\begin{aligned}\mathbf{C}'_{wh} = \sum_{w'=1}^{W} \sum_{h'=1}^{H} \mathbf{C}_{w'h'} \cdot \max(0, 1 - |w + \Delta_{1wh} - w'|) \\ \cdot \max(0, 1 - |h + \Delta_{2wh} - h'|),\end{aligned} \tag{3}$$

where a bilinear interpolation kernel is used to sample feature on position $(w + \Delta_{1wh}, h + \Delta_{2wh})$ of $\mathbf{C}$. $\Delta_{1wh}, \Delta_{2wh}$ indicate the learned 2D flow field for position (w, h).

Finally, $\mathbf{C}'$ and $\mathbf{L}'$ are concatenated to generate the fused BEV features, which are the input of the HD map decoder.

D. HD Map Decoder and Training Losses

Following HDMapNet [15], we define the HD map decoder as a fully convolutional network [23] that inputs the fused BEV features and outputs three predictions, including semantic segmentation, instance embedding, and lane direction, which are then used in the post-processing step to vectorize the map.

For training three different heads for three outputs, we use different training losses. We use the cross-entropy loss L_{seg} to supervise the semantic segmentation. For the instance embedding prediction, we define the loss L_{ins} as a variance and a distance loss [5] as

$$L_{var} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{j=1}^{N_c} [\|\mu_c - f_j^{\text{instance}}\| - \delta_v]_+^2, \tag{4}$$

$$L_{dist} = \frac{1}{C(C-1)} \sum_{c_A \neq c_B \in C} [2\delta_d - \|\mu_{c_A} - \mu_{c_B}\|]_+^2, \tag{5}$$

$$L_{ins} = \alpha L_{var} + \beta L_{dist}, \tag{6}$$

where C is the number of clusters, N_c and μ_c are the number of elements in cluster c and mean embedding of c. f_j^{instance} is the embedding of the j th element in c. $\|\cdot\|$ is the L_2 norm, $[x]_+ = max(0, x)$, δ_v and δ_d are margins for the variance and distance loss.

For direction prediction, we discretize the direction into 36 classes uniformly on a circle and define the loss L_{dir} as the cross-entropy loss. We only do backpropagation for those pixels lying on the lanes that have valid directions. During inference, DBSCAN [6] is used to cluster instance embeddings, followed by non-maximum suppression [15] to reduce redundancy. We then use the predicted directions to connect the pixels greedily to get the final vector representations of HD map elements.

We use focal loss [19] with $\gamma = 2.0$ for depth prediction as L_{dep}. The final loss is the combination of the depth estimation, semantic segmentation, instance embedding and lane direction prediction, which is defined as

$$L = \lambda_{dep} L_{dep} + \lambda_{seg} L_{seg} + \lambda_{ins} L_{ins} + \lambda_{dir} L_{dir}, \tag{7}$$

where λ_{dep}, λ_{seg}, λ_{ins}, and λ_{dir} are weighting factors.

IV. Experiments

We evaluate SuperFusion for the long-range HD map generation task on nuScenes [2] and a self-collected dataset.

A. Implementation Details

Model. We use ResNet-101 [11] as our camera branch backbone and PointPillars [14] as our LiDAR branch backbone. For depth estimation, we modify DeepLabV3 [4] to generate pixel-wise probability distribution of depth bins. The camera backbone is initialized using the DeepLabV3 [4] semantic segmentation model pre-trained on the MS-COCO dataset [20]. All other components are randomly initialized. We set the image size to 256×704 and voxelize the LiDAR point cloud with $0.15\,\text{m}$ resolution. We use $[0, 90]\,\text{m} \times [-15, 15]\,\text{m}$ as the range of the BEV HD maps, which results in a size of 600×200. We set the discretized depth bins to $2.0 - 90.0\,\text{m}$ spaced by $1.0\,\text{m}$.

Training Details. We train the model for 30 epochs using stochastic gradient descent with a learning rate of 0.1. For the instance embedding, we set $\alpha = \beta = 1$, $\delta_d = 3.0$, and $\delta_v = 0.5$. We set $\lambda_{dep} = 1.0$, $\lambda_{seg} = 1.0$, $\lambda_{ins} = 1.0$, and $\lambda_{dir} = 0.2$ for different weighting factors.

B. Evaluation Metrics

Intersection over Union. The IoU between the predicted HD map M_1 and ground-truth HD map M_2 is given by

$$\text{IoU}(M_1, M_2) = \frac{|M_1 \cap M_2|}{|M_1 \cup M_2|}. \tag{8}$$

One-way Chamfer Distance. The one-way Chamfer distance (CD) between the predicted curve and ground-truth curve is given by

$$\text{CD} = \frac{1}{C_1} \sum_{x \in C_1} \min_{y \in C_2} \|x - y\|_2, \tag{9}$$

where C_1 and C_2 are sets of points on the predicted curve and ground-truth curve. CD is used to evaluate the spatial distances between two curves. There is a problem when using CD alone for the HD map evaluation. A smaller IoU tends to result in a smaller CD. Here, we combine CD with IoU for selecting true positives as below to better evaluate the HD map generation task.

Average Precision. The average precision (AP) measures the instance detection capability and is defined as

$$\text{AP} = \frac{1}{10} \sum_{r \in \{0.1, 0.2, \ldots, 1.0\}} \text{AP}_r, \tag{10}$$

where AP_r is the precision at recall$=r$. As introduced in [15], they use CD to select the true positive instances. Besides that, here we also add an IoU threshold. The instance is considered as a true positive if and only if the CD is below and the IoU is above the defined thresholds. We set the threshold of IoU as 0.1 and threshold of CD as $1.0\,\text{m}$.

Evaluation on Multiple Intervals. To evaluate the long-range prediction ability of different methods, we split the

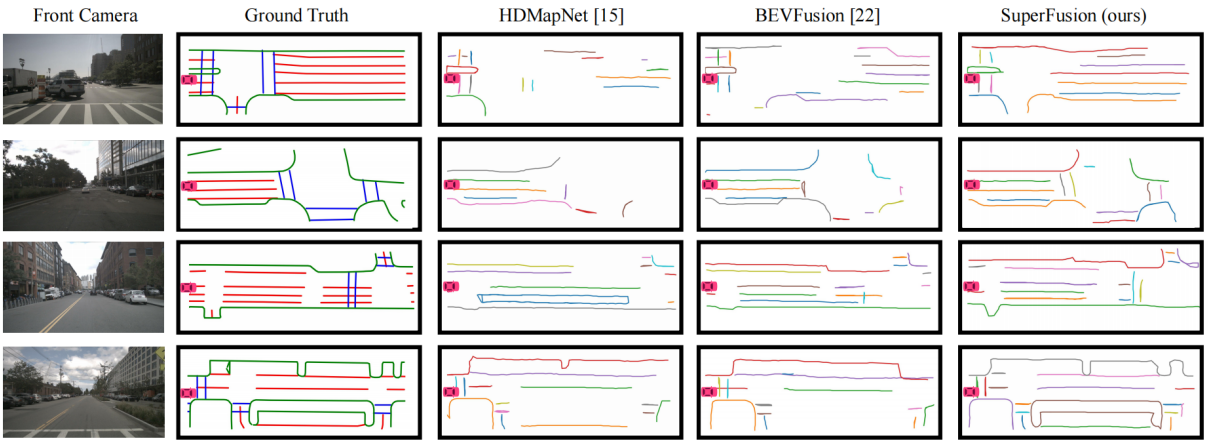


Fig. 5: Qualitative HD map prediction results of different methods. The red car represents the current position of the car. The length of every map is 90 m with respect to the car. Different colors indicate different HD map element instances. For ground truth HD map, green is lane boundary, red is lane divider, and blue is pedestrian crossing. More qualitative results are in the attached demo video.

TABLE I: IoU scores (%) of HD map semantic segmentation on nuScenes dataset. IoU: higher is better. C: camera. L: LiDAR.

Method	Modality	0-30 m			30-60 m			60-90 m			Average IoU		
		Divider	Ped	Boundary	Divider	Ped	Boundary	Divider	Ped	Boundary	Divider	Ped	Boundary
VPN [27]	C	21.1	6.7	20.1	20.9	5.1	20.3	15.9	1.9	14.7	19.4	4.9	18.5
LSS [28]	C	35.1	16.0	33.1	28.5	6.5	26.7	22.2	2.7	20.7	28.9	9.4	27.2
PointPillars [14]	L	41.5	26.4	53.6	18.4	9.1	25.1	4.4	1.7	6.2	23.7	14.5	30.7
HDMapNet [15]	C+L	44.3	28.9	55.4	26.9	10.4	31.0	18.1	5.3	18.3	30.5	16.6	35.7
BEVFusion [18]	C+L	42.0	27.6	52.4	26.8	11.9	30.3	18.1	3.3	15.9	30.0	16.3	34.2
BEVFusion [22]	C+L	45.9	31.2	57.0	30.6	13.7	34.3	22.4	5.0	21.7	33.9	18.8	38.8
SuperFusion (ours)	C+L	**47.9**	**37.4**	**58.4**	**35.6**	**22.8**	**39.4**	**29.2**	**12.2**	**28.1**	**38.0**	**26.2**	**42.7**

TABLE II: Instance detection results on nuScenes dataset. The predefined threshold of Chamfer distance is 1.0 m and the threshold of IoU is 0.1 (*e.g.*a prediction is considered as a true positive if and only if the CD is below and the IoU is above the defined thresholds). AP: higher is better.

Method	Modality	0-30 m			30-60 m			60-90 m			Average AP		
		Divider	Ped	Boundary	Divider	Ped	Boundary	Divider	Ped	Boundary	Divider	Ped	Boundary
VPN [27]	C	16.2	3.4	30.5	17.1	4.1	30.2	13.3	1.5	21.1	15.6	3.1	27.5
LSS [28]	C	24.0	9.9	39.3	23.9	5.7	38.1	19.2	2.2	26.2	22.5	6.2	34.8
PointPillars [14]	L	24.6	18.7	49.3	15.9	7.8	36.8	4.1	1.9	9.2	15.6	10.1	32.7
HDMapNet [15]	C+L	30.5	20.0	54.5	23.7	9.2	46.3	15.2	4.2	26.4	23.6	11.7	43.1
BEVFusion [18]	C+L	25.8	19.1	47.6	20.3	10.2	38.3	12.5	4.0	18.5	20.0	11.6	35.4
BEVFusion [22]	C+L	29.7	22.5	53.6	25.1	11.5	46.1	17.9	4.8	26.9	24.7	13.6	42.8
SuperFusion (ours)	C+L	**33.2**	**26.4**	**58.0**	**30.7**	**18.4**	**52.7**	**24.1**	**10.7**	**38.2**	**29.7**	**19.2**	**50.1**

ground truth into three intervals: $0-30$ m, $30-60$ m, and $60-90$ m. We calculate the IoU and AP of different methods on three intervals to thoroughly evaluate the HD map generation results.

C. Evaluation Results

nuScenes Dataset. We first evaluate our approach on the publicly available nuScenes dataset [2]. We focus on semantic HD map segmentation and instance detection tasks as introduced in [15] and consider three static map elements, including lane boundary, lane divider, and pedestrian crossing. Tab. I shows the comparisons of the IoU scores of semantic map segmentation. Our SuperFusion achieves the best results in all cases and has significant improvements on all intervals (Fig. 5), which shows the superiority of our method. Besides, we can observe that the LiDAR-camera fusion methods are generally better than LiDAR-only or camera-only methods. The performance of the LiDAR-only method drops quickly for long-range distances, especially for $60-90$ m, which reflects the case we analyzed in Fig. 3a. The AP results considering both IoU and CD to decide the true positive shows a more comprehensive evaluation. As shown in Tab. II, our method achieves the best instance detection AP results for all cases with a large margin, verifying the effectiveness of our proposed novel fusion network.

Self-recorded Dataset. To test the good generalization ability of our method, we collect our own dataset in real driving scenes and evaluate all baseline methods on that dataset. Our dataset has a similar setup as nuScenes with a LiDAR and camera sensor configuration. The static map elements are labeled by hand, including the lane boundary and lane divider. There are 21 000 frames of data, with 18 000 for training and 3 000 for testing. Fig. 3a shows sample data from our dataset and we put more examples on GitHub due to page limits. Tab. III shows the comparison results of different baseline methods operating on our dataset. We see consistent superior results of our method in line with those on nuScenes. Our SuperFusion outperforms the state-of-the-art methods for all cases with a large improvement.

TABLE III: The experimental results on the self-recorded dataset.

Method	Modality	Average IoU		Average AP	
		Divider	Boundary	Divider	Boundary
VPN [27]	C	42.9	17.9	33.0	25.4
LSS [28]	C	49.2	20.4	40.4	26.5
PointPillars [14]	L	36.8	15.5	26.1	24.6
HDMapNet [15]	C+L	46.6	18.8	38.3	25.7
BEVFusion [18]	C+L	48.1	21.9	38.8	30.5
BEVFusion [22]	C+L	49.0	18.8	40.5	25.9
SuperFusion (ours)	C+L	**53.0**	**24.7**	**42.4**	**35.0**

TABLE IV: Ablation of the proposed network components.

	Average IoU		
	Divider	Ped	Boundary
w/o Depth Supervision	25.4	13.3	30.8
w/o Depth Prior	34.3	20.5	39.3
w/o LiDAR Prediction	33.4	17.6	38.6
w/o Cross-Attention	32.4	15.2	37.6
w/o BEV Alignment	33.4	21.8	39.1
SuperFusion (ours)	**38.0**	**26.2**	**42.7**

TABLE V: Module alternatives study.

Modules	Alternatives	Average IoU		
		Divider	Ped	Boundary
Alignment module	DynamicAlign [18]	33.8	19.8	38.8
	ConvAlign [22]	33.5	22.9	39.1
	BEVAlign (ours)	**38.0**	**26.2**	**42.7**
Depth prediction module	Depth Encoder (bin)	31.2	18.5	36.1
	Depth Encoder	34.6	20.5	38.5
	Depth Channel (bin)	31.3	16.5	37.0
	Depth Channel (ours)	**38.0**	**26.2**	**42.7**

D. Ablation Studies and Module Analysis

Ablation on Each Module. We conduct ablation studies to validate the effectiveness of each component of our proposed fusion network in Tab. IV. Without depth supervision, the inaccurate depth estimation influences the camera-to-BEV transformation and makes the following alignment module fails, which results in the worst performance. Without the sparse depth map prior from the LiDAR point cloud, the depth estimation is unreliable under challenging environments and thus produces inferior results. Without the prediction module, there is no measurement from LiDAR in the long-range interval, and only camera information is useful, thus deteriorating the overall performance. In the "w/o Cross Attention" setting, we add the encoder-decoder LiDAR BEV prediction structure but remove the cross-attention interaction with camera FV features. In this case, the network tries to learn the LiDAR completion from the data implicitly without guidance from images. The performance drops significantly for this setup, indicating the importance of our proposed image-guided LiDAR prediction module. In the last setting, we remove the BEV alignment module and concatenate the BEV features from the camera and LiDAR directly. As can be seen, due to inaccurate depth estimation and extrinsic parameters, the performance without an alignment is worse than using our proposed BEV aligning module.

Analysis of Module Choices. In the upper part of Tab. V, we show that our BEVAlign module works better than the alignment methods proposed in the previous work [22], [18]. [22] uses a simple convolution-based encoder for alignment, which is not enough when the depth estimation is inaccurate. The dynamic fusion module proposed in [18] works well on 3D object detection tasks but has a limitation on semantic segmentation tasks. In the lower part of Tab. V, we test different ways to add depth prior. One way is to add the sparse depth map as an additional input channel for the image branch. Another way is to use a lightweight encoder separately on RGB image and sparse depth map and concatenate the features from the encoder as the input for the image branch. Besides, the sparse depth map can either store the original depth values or the bin depth values. We see that adding the sparse depth map as an additional input channel with original depth values achieves the best performance.

TABLE VI: Quantitative path planning results.

	HDMapNet [15]	BEVFusion [22]	SuperFusion (ours)
Success rate	45%	49%	**72%**

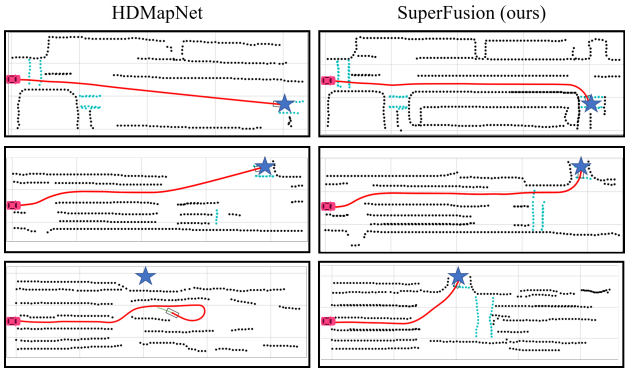


Fig. 6: Path planning results on the generated HD maps.

E. Useful for Path Planning

We use the same dynamic window approach (DWA) [7] for path planning on HD maps generated by HDMapNet [15], BEVFusion [22], and our SuperFusion. We randomly select 100 different scenes and one drivable point between 30 – 90 m as the goal for each scene. The planning is failed if the path intersects with the sidewalk or DWA fails to plan a valid path. Tab. VI shows the planning success rate for different methods. As can be seen, benefiting from accurate prediction for long-range and turning cases, our method has significant improvement compared to the baselines. Fig. 6 shows more visualizations of the planning results.

V. Conclusion

In this paper, we proposed a novel LiDAR-camera fusion network named SuperFusion to tackle the long-range HD map generation task. It exploits the fusion of LiDAR and camera data at multiple levels and generates accurate HD maps in long-range distances up to 90 m. We thoroughly evaluate our SuperFusion on the nuScenes dataset and our self-recorded dataset in autonomous driving environments. The experimental results show that our method outperforms the state-of-the-art methods in HD map generation with large margins. We furthermore showed that the long-range HD maps generated by our method are more beneficial for downstream path planning tasks.

REFERENCES

[1] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.L. Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, and J. Yan. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022.

[4] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation for autonomous driving. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[6] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 226–231, 1996.

[7] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, 1997.

[8] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2019.

[9] F. Ghallabi, F. Nashashibi, G. El-Haj-Shhade, and M.A. Mittet. Lidar-based lane marking detection for vehicle positioning in an hd map. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[10] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang, and T.E. Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Z. Huang, Y. Wei, X. Wang, H. Shi, W. Liu, and T.S. Huang. Alignseg: Feature-aligned segmentation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):550–557, 2021.

[13] J. Ku, A. Harakeh, and S.L. Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *Conference on Computer and Robot Vision (CRV)*, 2018.

[14] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] Q. Li, Y. Wang, Y. Wang, and H. Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[16] X. Li, A. You, Z. Zhu, H. Zhao, M. Yang, K. Yang, and Y. Tong. Semantic flow for fast and accurate scene parsing. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[17] Y. Li, A.W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q.V. Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] T. Liang, H. Xie, K. Yu, Z. Xia, Y.W. Zhiwei Lin, T. Tang, B. Wang, and Z. Tang. BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[19] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[20] T. Lin, M. Maire, S.J. Belongie, L.D. Bourdev, R.B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: common objects in context. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014.

[21] Y. Liu, Y. Wang, Y. Wang, and H. Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022.

[22] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] F. Ma, G.V. Cavalheiro, and S. Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. *arXiv preprint arXiv:1807.00275*, 2018.

[25] F. Ma and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[26] R. Mur-Artal and J.D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[27] B. Pan, J. Sun, H.Y.T. Leung, A. Andonian, and B. Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, Jul 2020.

[28] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[29] C. Reading, A. Harakeh, J. Chae, and S.L. Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] A. Saha, O. Mendez, C. Russell, and R. Bowden. Translating images into maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[31] S. Vora, A.H. Lang, B. Helou, and O. Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] C. Wang, C. Ma, M. Zhu, and X. Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[33] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018.

[34] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, 2020.