MDPI

Article

Probabilistic Semantic Mapping for Autonomous Driving in Urban Environments

Hengyuan Zhang [1,†] ∗, Shashank Venkatramani [1,†], David Paz [1], Qinru Li [1], Hao Xiang [1] and Henrik I. Christensen [1]

[1] Autonomous Vehicle Laboratory, Contextual Robotics Institute, University of California San Diego; avl@ucsd.edu
∗ Correspondence: hyzhang@ucsd.edu;
† These authors contributed equally to this work.

Abstract: Statistical learning techniques and increased computational power have facilitated the development of self-driving car technology. However, a limiting factor has been the high expense of scaling and maintaining high-definition (HD) maps. These maps are a crucial backbone for many approaches to self-driving technology. In response to this challenge, we present an approach that fuses pre-built point cloud map data with images to automatically and accurately identify static landmarks such as roads, sidewalks, and crosswalks. Our pipeline utilizes semantic segmentation of 2D images, associates semantic labels with points in point cloud maps to pinpoint locations in the physical world, and employs a confusion matrix formulation to generate a probabilistic bird's-eye view semantic map from semantic point clouds. The approach has been tested in an urban area with different segmentation networks to generate a semantic map with road features. The resulting map provides a rich context of the environment that is valuable for downstream tasks such as trajectory generation and intent prediction. Moreover, it has the potential to be extended to automatic generation of HD maps for semantic features. The entire software pipeline is implemented in Robot Operating System (ROS), a widely used robotics framework, and available at: https://github.com/AutonomousVehicleLaboratory/semantic_mapping_v2.

Keywords: Autonomous Vehicles; Semantic Mapping; Semantic Segmentation; Fusion

1. Introduction

Many approaches to design of autonomous vehicles rely on high-definition (HD) maps to model the static parts of the environment. These maps provide crucial information such as centimeter-level definitions of road networks, traffic signs, crosswalks, traffic lights, and speed limits. Due to the dynamic nature of the real world, these maps can quickly become outdated, especially during road network changes or construction. Manually annotating HD maps is a laborious and time-consuming process, and outdated maps can lead to unsafe scenarios when vehicles perform inadequate reference path tracking actions. Extracting semantics and attributes from data are the most challenging aspects of HD map generation [1]. Given this, a method that automates semantic extraction could significantly improve HD map generation, reduce labor costs, and enhance driving safety.

Generating centimeter-level semantic labels for a scene is a cumbersome task. Many efforts approach this problem from the perspective of scene understanding. Prior work has used Conditional Random Fields (CRF) to assign semantic labels [2,3]. More recently, deep learning techniques have shown promising results in retrieving semantic information from images [4–6], point clouds [7] or both [8]. However, semantic scene understanding does not account for stitching together individual observations to generate a map representation.

Some researchers have also explored methods to create semantic maps of the environment, including [9–11]. However, these approaches either rely on aerial imagery / high-cost sensors to extract road information, which can limit the availability of data, or

Citation: Zhang, H.; Venkatramani, S.; Paz, D.; Li, Q.; Xiang, H.; Christensen, H.; Probabilistic Semantic Mapping for Autonomous Driving in Urban Environments. *Sensors* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

they do not explicitly map lane and crosswalk information, which are crucial for HD map generation.

Other work directly generates the lane-level HD map [7,12–15] or topology map [16]. These maps are in sparse vectorized representation that can be valuable for planning. However, these methods are limited by a small set of map elements. For this reason, the generated maps lack the rich context required for urban driving.

Our study addresses gaps in the automatic generation of dense probabilistic semantic maps in urban driving environments. To achieve this, we propose a semantic mapping pipeline that creates a Bird's-Eye View (BEV) semantic map of the environment instead of a single-frame semantic understanding. The pipeline utilizes a confusion matrix to incorporate the uncertainty of the semantic segmentation network into mapping and fuses Light Detection and Ranging (LiDAR) intensity to map lane marks accurately. We leverage dense point maps obtained from a 16-channel LiDAR to reduce the cost and increase data availability. Furthermore, our work builds on state-of-the-art semantic segmentation networks [6,17] that are trained exclusively on publicly available datasets [18], providing rich semantic labels including roads, lane marks, crosswalks, and sidewalks. To evaluate the effectiveness of the proposed model, we compare it with ground truth HD maps generated for our campus, and use data from our autonomous vehicle. The results demonstrate that our model accurately identifies semantic features on the road and can effectively map them with a small error margin.

We augmented our initial work [19] by adding new semantic segmentation models, and adding extensive analysis with modified precision and recall, which are more appropriate for evaluating mapping performance. We additionally open source the code for running our entire pipeline.

The paper is organized with an initial discussion of related work in Section 2. We present the overall methodology in Section 3 and the associated experiments in Section 4. Based on our results, there are a number of issues to consider regarding standard datasets, labeling and evaluation, which are discussed in Section 5 before we summarize in Section 6.

2. Related Work

In this section we will briefly summarize related work across the areas of segmentation (Sub-section 2.1), mapping (Sub-section 2.2), HD map generation (Sub-section 2.3) and probabilistic maps (Sub-section 2.4).

2.1. Semantic Segmentation

There has been significant progress in the field of semantic segmentation, which involves assigning semantic labels to each data point (e.g., pixel or voxel). Large-scale datasets like CityScapes [20], CamVid [21], and Mapillary [18] have accelerated this progress in the domain of road scenes. Semantic segmentation algorithms that provide pixel-level information can be particularly useful for building HD maps, which require fine-grained labeling for scene objects.

2D semantic segmentation approaches, such as those in [4,5,22], use encoder-decoder architectures to interpret global and local information in images. These models, when trained on the aforementioned large datasets, can effectively segment objects on the road. 3D semantic segmentation approaches have utilized Convolutional Neural Networks (CNNs) to classify points in LiDAR point clouds after a transformation into range images, in [23–25] These methods provide promising results, but fail to distinguish objects with textural differences. Full 3D semantic segmentation using voxel-based approaches has also been proposed [26]; however, they require 3D convolutions on dense raw point clouds (32 or 64 channel LiDARs), making real-time operation challenging. New transformer-based approaches have shown improvements in evaluation metrics [27], though they require higher computational capabilities for full self-attention. Recent work also tries to directly generate semantic segmentation in BEV from a single image [28] or using the fusion of LiDAR and camera [8].

2.2. Semantic Mapping

The term semantic mapping has taken various meanings in literature [29]. For our purposes, we have chosen to follow the definition provided in [30], which is a map that contains environmental attributes and occupancy metrics. For the task of autonomous vehicles, this encompasses features such as drivable areas and road features.

There are alternative methods that utilize CRF-based techniques to achieve high-density semantic mapping [3]. In this instance, an associative hierarchical CRF is utilized for semantic segmentation, while a pairwise CRF is used for mapping. The latter strategy ensures that the output remains smooth. Another approach, detailed in [31], involves using a stereo pair to estimate depth reliably. However, this particular method does not account for the explicit mapping of crosswalks and lanes, both of which are necessary for the creation of HD maps.

In a related study, Maturana et al. [9] combine semantic imagery captured by a camera with LiDAR point clouds. They rely on raw point clouds in real-time from a 64-channel LiDAR, which provides more dense real-time information at a higher cost. Our approach, however, can build a map from a relatively cheaper 16-channel LiDAR. Moreover, their research concentrates on off-road environments, whereas our research focuses on urban driving scenarios. In such settings, certain traffic rule-related categories, like crosswalks and lane markings, require higher attention.

2.3. HD Map Generation

The generation of HD maps has been explored from various perspectives including online and offline mapping. Zhou et al. [12] propose to construct lane-level HD maps for urban environments. They first use cameras and LiDARs for 3D semantic reconstruction, then use the OpenStreetMap (OSM) with a semantic particle filter to generate offline lane-level HD maps for the urban environment.

Online methods are gaining popularity. Homayounfar et al. [7] generate a lane-level map for the highway. Facilitated by large-scale open datasets with HD map data such as nuScenes [32], Argoverse 2 [33,34] and OpenLane-V2 [35], a line of work focus on generating online HD map for urban environments. Li et al. [36] propose HDMapNet that generates rasterized maps while Liu et al. [13] propose VectorMapNet to generate vectorized representations directly. MapTR [14] and TopoNet [15] improve mapping performance by using permutation invariant representations and a topology-preserving loss, respectively. Can et al. [16] propose a loss that captures the accuracy in estimating topology. Additionally, HD maps can be built from aerial imagery [37] but the availability of data can present a limitation. These works focus on sparse lane-level representations with predefined map element types. In contrast, our generated dense maps can capture all semantic classes from the semantic segmentation network.

2.4. Probabilistic Map

Probabilistic mapping builds a map that maximizes the likelihood of the map under the data [38]. Thrun et al. [38] build a probabilistic map by modeling the occupancy probability with expectation maximization. Their work and many other works [39,40] address Simultaneous Localization and Mapping (SLAM) while our work focuses only on the mapping of semantic attributes. Semantic maps have been utilized successfully in the areas of localization [41,42] and prediction of pedestrian motion [43]. This approach is advantageous because it enables the representation of inherent distribution information within a discrete space while simultaneously filtering out noise. Our current work builds on this technique by applying it to the creation of semantic maps, while additionally incorporating prior information from LiDAR's intensity channel. As a result of this integration, we can generate semantic maps that are more stable given potentially noisy semantic images.

3. Materials and Methods

Our model consists of three main components: semantic segmentation, semantic association, and semantic mapping. Figure 1 illustrates the overall architecture. To begin, semantic segmentation networks are used to predict semantic labels on 2D images. These labels are then associated with densified 3D point clouds. Finally, a probabilistic mapping process is applied to convert the distribution of observations to a single label on a per-map pixel basis. In the following section, we will provide a detailed description of each component.

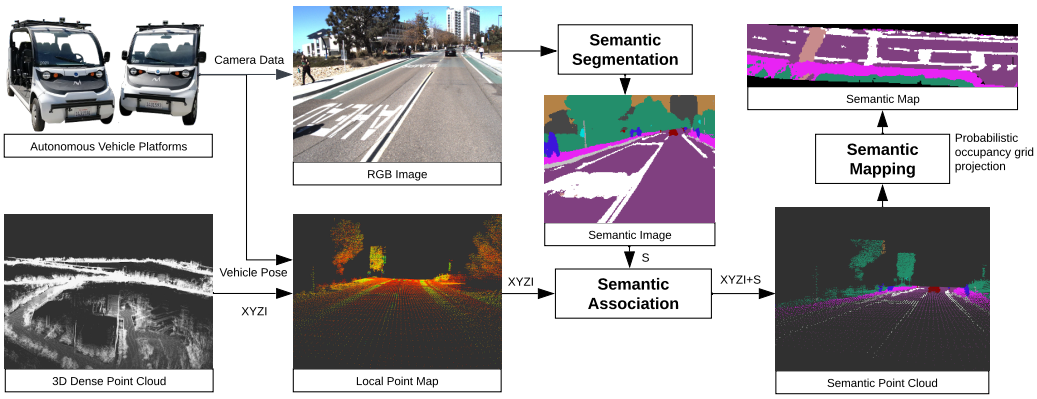

Figure 1. Our semantic mapping pipeline generates semantic labels for images, associates the labels to the local point cloud, and updates the semantic map in Bird's-Eye View (BEV) probabilistically.

3.1. Image Semantic Segmentation

The first component, semantic segmentation, extracts the semantic labels from 2D images using neural networks. For each pixel in an image with shape W by H, the output is a label c from a set of predefined semantic classes C such as road, lane mark, and sidewalk. We offer two different segmentation network options, DeepLabV3Plus [6] and Hierarchical MultiScale Semantic Segmentation with HRNet+OCR (MScale-HRNet) [17]. At inference time DeepLabV3Plus is faster but noisier, and MScale-HRNet is slower and more memory intensive, but provides higher-quality segmentation. We discuss the tradeoffs of both methods in the context of the final generated semantic map in later sections.

For DeepLabV3Plus the feature extraction backbone is a lightweight ResNeXt50 [44] pre-trained on ImageNet [45]. Compared to other backbones like ResNet101 [46], ResNeXt50 achieves the same mean Intersection over Union (mIoU) value with fewer parameters and faster inference times. To further improve inference time while preserving performance, we also employ depth-wise separable convolution in our spatial pyramid and decoder layers, inspired by [6,47].

Our DeepLabV3Plus semantic segmentation network is trained on the Mapillary Vistas dataset [18], which contains a large number of pixel-level semantic segmented images with 66 different labels in autonomous vehicle scenarios. The Mapillary Vistas dataset was at the start of our study the most comprehensive pixel level labeled dataset, and is still considered a viable basis for training. We reduce the labels to 19 essential classes for our driving environment by removing non-essential labels (e.g. snow) and merging labels with similar semantic meanings (e.g. zebra line and crosswalk). This decision is based on the observation that some classes do not appear in our test environment. The details of label merging are described in Sub-section 4.2.

MScale-HRNet uses a much larger HRNet+OCR backbone [48–50] that utilizes object context to achieve higher performance for irregular semantic regions. Additionally, by utilizing multi-scale segmentation with attention [51], the network pulls larger area semantic features from smaller scale images, and more refined semantic features from larger scale images. The fusing of different scales is done in a hierarchical manner, enabling the scales used at inference time to be changed without retraining. As such one can modify

the runtime and memory requirement by lowering or increasing the scales used (this does result in changes in model performance). Overall, it achieves better segmentation than DeepLabV3Plus, at the cost of higher computation requirements.

From a quantitative standpoint on the cityscapes test set DeepLabV3Plus is capable of attaining an mIoU of 82.10% [6] while MScale-HRNet achieves an mIoU of 85.10% [17].

3.2. Point Cloud Semantic Association

The second component, semantic association, reconstructs a 3D scene with semantic labels. Given the semantic images from semantic segmentation, this is achieved by assigning depth to the semantic image. However, the depth information is often not readily available. Depth estimation from multi-view geometry relies on salient features, which can be prone to errors on the road or under challenging lighting conditions. Alternatively, LiDAR sensors can capture depth information, but their sparse resolution, typically with only a few optical channels (e.g., 16), can make it difficult to infer the underlying geometry in real time. To overcome this challenge, our method leverages centimeter-level localization [52] to extract small, dense regions from a previously built dense point cloud map. These regions are then projected into the semantically segmented image to retrieve depth information. Building a dense point map can be automated and only requires driving through the area once, making it much less expensive than human labeling.

Assuming the vehicle is localized with respect to a point cloud map P_g with coordinate X_v. A local point cloud P_l is extracted within a max distance in each dimension in the local coordinates of the vehicle. The transformation from the local point map to the localizer (Velodyne LiDAR) ${}^{l}\mathbf{T}_{m}$ is given by precise centimeter-level localization. We also calibrate the camera with respect to the LiDAR using a non-iterative method solution for the PnP method [53], to estimate their relative transformation ${}^{c}\mathbf{T}_{l}$. Therefore, the extrinsic transformation between the camera and the points map frame ${}^{c}\mathbf{T}_{m}$ is known.

$$ {}^{c}\mathbf{T}_{m} = ({}^{c}\mathbf{T}_{l})({}^{l}\mathbf{T}_{m}). \tag{1} $$

Thus semantic information for a point $\mathbf{X}_m \in P_l$ can be retrieved from the label of its projected points in image coordinates $\mathbf{x}_i$.

$$ \mathbf{x}_i = \mathbf{K}\boldsymbol{\pi}({}^{c}\mathbf{T}_{m})\mathbf{X}_m \tag{2} $$

where $\mathbf{K}$ is the camera intrinsic matrix and $\boldsymbol{\pi} = [\mathbf{I}|\mathbf{0}]$ is the canonical projection matrix.

Finally, we assign the semantic label of pixel $\mathbf{x}_i$ in the semantic image to the point $\mathbf{X}_m$ to form a semantic point cloud.

3.3. Semantic Mapping

A point cloud with semantic labels is a useful representation of a scene's 3D geometry, but it can be affected by sensor measurement noise and small semantic label fluctuations. To address this, we use a local or global probabilistic map, where the former provides dense semantic cues around the ego-vehicle, and the latter automates the process of building HD maps. Both local and global maps use semantic occupancy grids, with the main difference being the reference frame. Our comparisons are performed in the global frame.

A local probabilistic map is a BEV representation in the body frame (rear-axle) of the ego vehicle. We construct it for a given frame using the semantic point cloud and update it when there is a significant change in the ego-vehicle's pose. On the other hand, a global probabilistic map operates directly in the global frame without the need for map transformations. A visual comparison of the two is shown in Figure 2.

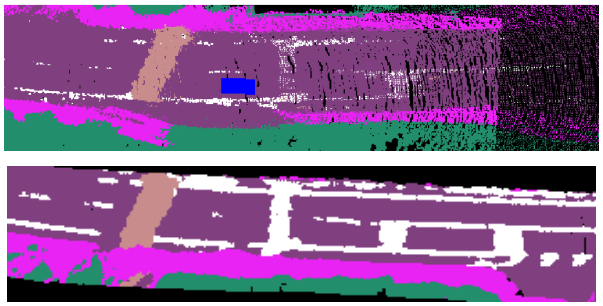

Figure 2. Top to Bottom, local probabilistic map where blue car is ego vehicle, the same region in a final generated global map.

The semantic occupancy grid has height H, width W, and channels C, with each channel corresponding to a semantic class of the scene. The channels for a cell in the BEV map model the semantic class probability distribution. When constructing the semantic point cloud, we project it onto the grid using the x and y components. The point will be associated with the nearest cell c_{ij}, which covers a $d \times d$ square area of the physical world. Then we will update the channels in the cell based on the semantic label of the point.

We enhance the robustness of the semantic occupancy grid estimation using a probabilistic model that incorporates both the semantic and LiDAR intensity information from the point cloud to reduce the prediction error. We denote the semantic label distribution across all the channels as $\mathbf{S}_t$, the observed semantic labels as $\mathbf{z}_t$, and the observed LiDAR intensity as $\mathcal{I}_t$. Thus, the task is to estimate $\mathbf{S}_t$ from past observations, i.e., the probability distribution of $P(\mathbf{S}_t|\mathbf{z}_{1:t}, \mathcal{I}_{1:t})$. We assume that observed semantic labels and LiDAR intensity are conditionally independent given $\mathbf{S}_t$ and follow the Markov assumption to update the semantic probability.

$$P(\mathbf{S}_t|\mathbf{z}_{1:t}, \mathcal{I}_{1:t}) = \frac{1}{\mathbf{Z}} P(\mathbf{z}_t|\mathbf{S}_t) P(\mathcal{I}_t|\mathbf{S}_t) P(\mathbf{S}_{t-1}|\mathbf{z}_{1:t-1}, \mathcal{I}_{1:t-1}) \tag{3}$$

We introduce a normalization factor $\mathbf{Z}$, and assume that $P(\mathbf{S}_t|\mathbf{z}_{1:t-1}, \mathcal{I}_{1:t-1})$ is equivalent to $P(\mathbf{S}_{t-1}|\mathbf{z}_{1:t-1}, \mathcal{I}_{1:t-1})$. To enable a more precise probabilistic update, we use a 2D confusion matrix $\mathbf{M}$ to model $P(\mathbf{z}_t|\mathbf{S}_t)$, where each element in the matrix represents the probability of label i being predicted as label j. Additionally, we model $P(\mathcal{I}_t|\mathbf{S}_t)$ as a prior function of the intensity of each class in the scene.

The confusion matrix models the uncertainty of the model evaluated on a dataset, which describes the prior probability of a label $\mathbf{z}_t$ being observed when the true class is $\mathbf{S}_t$. As a result, for any point projected to the cell, all channels in the cell will be updated according to the confusion matrix. To ensure numerical stability, we use the logarithmic form to update the channels.

The intensity data collected by LiDAR sensors provides valuable information about different materials in the scene. For instance, the top image in Figure 3 shows a BEV intensity map of a road segment where lane markings appear brighter due to their high reflectivity. We use a threshold value k to segment out the lane markings and employ this information as a prior to better understand the layout of the scene. This approach can be especially helpful when semantic segmentation fails to capture the correct label due to poor lighting conditions.

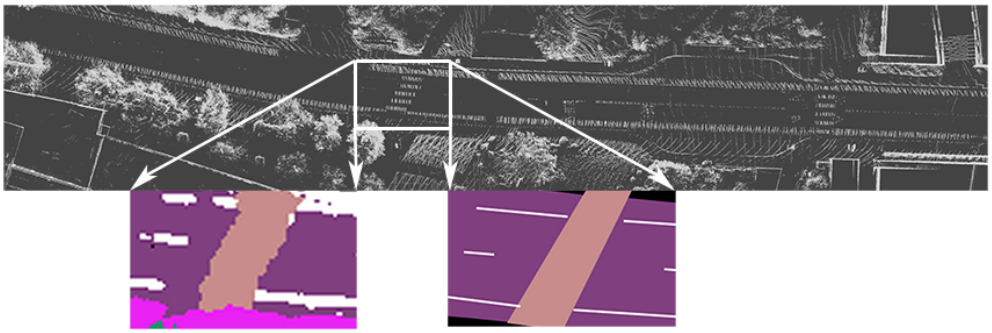

Figure 3. A visualization of our generated map (bottom left), the ground truth label (bottom right), and the intensity thresholded LiDAR point cloud map (top).

4. Experiments

We perform experiments to verify the effectiveness of the proposed semantic mapping pipeline. We introduce our vehicle platform in Sub-section 4.1. Then we discuss the training, hyperparameter, and result comparison of semantic segmentation networks in Sub-section 4.2. The semantic mapping results with ablation study and analysis are presented in Sub-section 4.3. Lastly, we compare different depth association approaches for semantic mapping in Sub-section 4.4.

4.1. Platform

We collected our experimental data using one of our autonomous cars, as described in [52]. This car is equipped with a 16-channel LiDAR and six cameras, arranged with two cameras on the front, one on each side, and two on the back, as depicted in Figure 4. We recorded data from the front left camera, LiDAR, and vehicle position by driving through the UC San Diego campus. The camera data was streamed at approximately 13 Hz, while the LiDAR scans were performed at approximately 10 Hz. By driving through the campus, we were able to gather data for various urban driving scenarios, including challenging situations such as navigating steep hills, intersections, and construction sites.

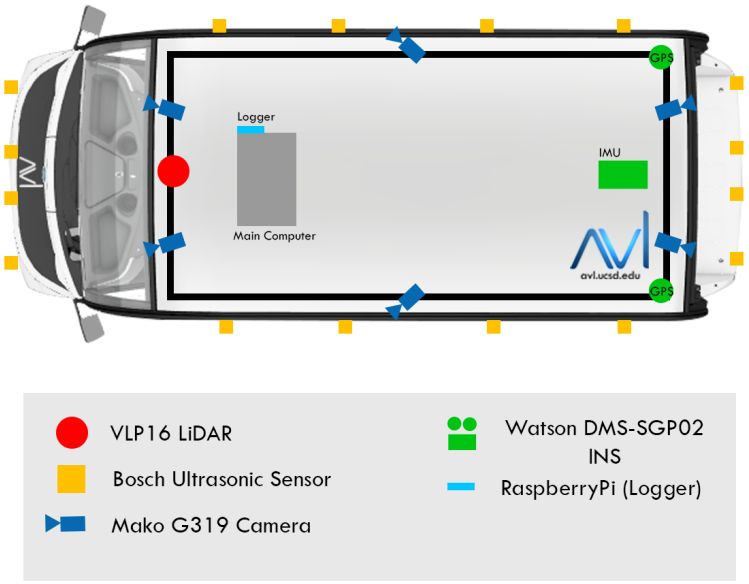

Figure 4. Vehicle Sensor Configuration.

4.2. Image Semantic Segmentation

We have two semantic segmentation networks, MScale-HRNet and DeepLabV3Plus. The MScale-HRNet pre-trained model[1] produces a high-quality semantic mask with clean edges most of the time. For its high-quality results, we use the straight-out-of-the-box pre-trained 65-class model directly. On the other hand, the DeepLabV3Plus produces much noisier results. Therefore, we reduce the total classes from 65 to 19 and retrain the model. In this subsection, we describe the configuration for MScale-HRNet and the training process for DeepLabV3Plus.

4.2.1. MScale HRNet+OCR Configuration

MScale-HRNet allows a flexible scale selection during inference time. We chose three scales at 0.25, 0.5, and 1.0 for our experiments. Typical experiments are done with a scale of 0.5, 1.0, and 2.0 but require more than 11GB Graphics Processing Unit (GPU) memory for input size 1920x1440. We observe that even with the downsized scales, the network was able to produce much cleaner and more accurate results than DeepLabV3Plus. Testing on our own vehicle data showed good generalization.

4.2.2. DeepLabV3Plus Training Dataset

Our training dataset consists of 18,000 images, while our validation dataset has 2,000 images, both of which are obtained from the Mapillary dataset [18]. To optimize our training process, we merged similar categories such as terrain and vegetation, different types of riders and pedestrians into a single human category, and various types of crosswalks into a unified crosswalk class. We also combined traffic-sign-back and traffic-sign-front into a single traffic-sign category, and merged bridge images into the building category.

To further improve the training dataset, we applied several data augmentation techniques, including random horizontal flips with a probability of 0.5, random resizing with a scale ranging from 0.5 to 2, and random cropping. Additionally, we normalized the images to a distribution with a mean of (0.485, 0.456, 0.406) and a standard deviation of (0.229, 0.224, 0.225).

Our experiments indicate that the Mapillary dataset is similar to our driving scenarios, and the extensive data augmentation during the training process helps improve DeepLabV3Plus generalization. We did not observe a significant drop in performance when testing the DeepLabV3Plus model on the UC San Diego campus.

4.2.3. DeepLabV3Plus Hyperparameters

To train our DeepLabV3Plus network, we employ synchronized batch normalization [5] with a batch size of 16. The training process lasts for 200 epochs, utilizing eight 2080Ti GPUs with an input image size of 640x640. The network's output stride is eight.

To optimize the training process, we use the Stochastic Gradient Descent (SGD) optimizer and apply a polynomial learning rate policy [6,54]. Specifically, we set the base learning rate to 0.005 and the power to 0.9, with the learning rate decaying over time according to the formula $base\_lr \times (1 - \frac{epoch}{max\ epoch})^{power}$. We set the momentum to 0.9 and the weight decay to $4e^{-5}$.

4.2.4. Comparison of Semantic Segmentation

We use the mIoU metric to assess a network's performance. In the reduced 19-class Mapillary validation set, ResNeXt50 achieves an mIoU of 68.32%. Although its performance is slightly lower than that of ResNet101, ResNeXt50 requires significantly less memory (from 367MB to 210MB), making it more suitable for our onboard hardware with limited memory. For MScale-HRNet we evaluate it on the 65 class Mapillary validation set, where it achieves an mIoU of 59.71% for 65 classes.

[1] https://github.com/NVIDIA/semantic-segmentation

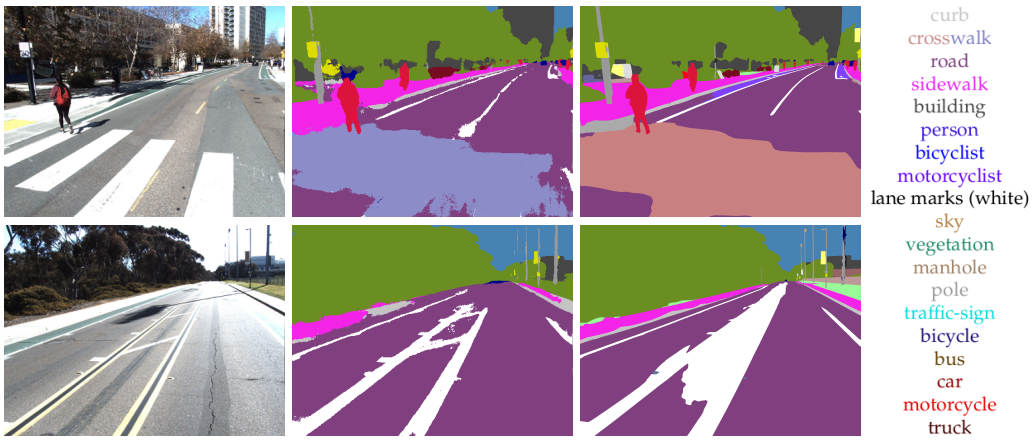

Figure 5. Semantic Segmentation Comparison (Left to right, AVL Dataset Image, DeepLabV3Plus, MSCale-HRNet) with labels colored correspondingly.

Qualitatively, as shown in Figure 5 the two semantic segmentation networks perform similarly in close range with only one major difference. MScale-HRNet fills in gaps on the dash lane while DeepLabV3Plus does not do this as consistently. This stems from irregular labeling in the Mapillary dataset, which we discuss further in Sub-section 5.3. For our ground truth labels, we do not fill in dash lanes and that can lead to a performance drop for the MScale-HRNet approach.

The DeepLabV3Plus generates noisier results on the edges of the segments. MScale-HRNet outputs are cleaner with smooth edges. For areas further away from the camera, MScale-HRNet results give more details. However, these areas are not utilized since we clip the point cloud with a maximum distance to reduce error (see Sub-section 4.3.4).

For an image size of 1920 by 1440, DeepLabV3Plus' inference time is approximately 0.48 s per image and MScale-HRNet's inference time is approximately 1.23 s per image when running on an NVIDIA GeForce RTX 2080Ti graphics card.

4.3. Semantic Mapping

We evaluate the quality of our map generation results by selecting a 1.1 km region of the UC San Diego campus, which has been manually annotated with an HD map containing road information, including crosswalks, sidewalks, and lane marks. The semantic map we generate has five channels - *road, crosswalk, lane marks, vegetation, and sidewalk* - with a resolution of $d = 0.2$ meters. Generating an accurate HD map requires considerable effort, but it demonstrates the value of automating the process.

4.3.1. Metric for Semantic Mapping

In our initial work [19] we used mIoU and pixel accuracy as evaluation metrics. However, a direct comparison on IoU for lane marks is very sensitive to localization error. In Figure 6 we show the generated semantic map, ground truth, and disparity between the lane labels in these two maps. It can be seen that there are relatively consistent detections of the lane lines in the generated semantic map, however, when compared to ground truth they are off by 1 to 2 pixels (0.2–0.4m since 1px = 0.2m). Given that the ground truth lane is about 1 to 2 pixels wide, the offset leads to a very low true positive rate.

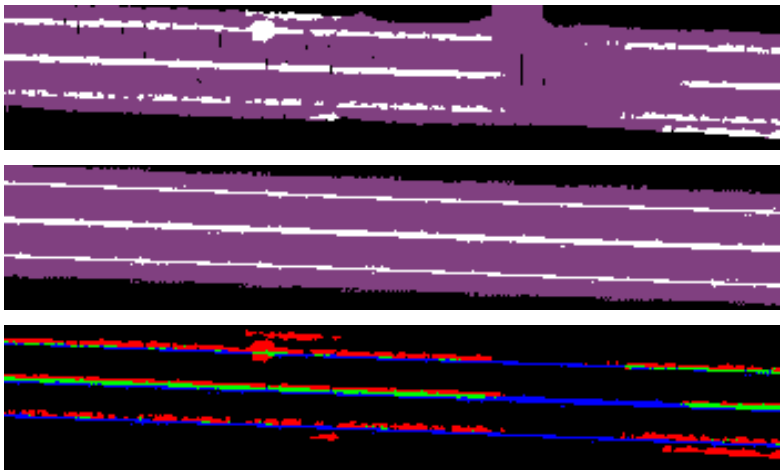

Figure 6. (From top to bottom) Generated semantic map, Groundtruth, Disparity between lane labels. Green represents true positive, red represents false positive, and blue represents false negative.

This offset is also egocentrically consistent across the entire generated map, leading us to believe this is a systematic problem unrelated to the semantic mapping approach. The offset can potentially be caused by an error introduced by the calibration between LiDAR and the camera, the asynchronous camera and LiDAR, the BEV conversion, or a discretization error in mapping.

For generating HD maps, this offset that is present is non-ideal. There are other tasks, however, which are less sensitive to this offset. An example is to use the map as a prior to provide context for scene understanding. The semantic map can be used in downstream tasks such as trajectory generation or motion prediction. In these scenarios, the existence of the semantic information is more important and centimeter-level mapping requirements may be too strict.

Therefore, in addition to IoU, we propose a metric to evaluate the performance of the semantic map that is tolerable to minor offsets. The proposed metric included a modified version of precision and recall. We dilate the ground truth to evaluate the precision of the generated map. We dilate the generated map for each label to evaluate the recall against the original ground truth. Specifically, we used a kernel size of 3, which tolerates a 20 cm error. We notice that these additional metrics match our observation of the performance of the model, thus can better guide our decision in hyperparameter tuning and model comparison.

Additionally, it is worth noting that the sparsity of the LiDAR point cloud may influence these metrics since the output may be accurate, but it may contain unclassified cells (holes). We mitigate this problem by using a smoothing kernel to interpolate the missing labels on our map.

Table 1. Quantitative evaluation on our labeled data for road, crosswalk, and lane mark regions. Refer to Sub-section 4.3 for details.

Network	Config	Road			Crosswalks			Lane marks		
		Precision*	Recall*	IoU	Precision*	Recall*	IoU	Precision*	Recall*	IoU
DeepLabV3+	Vanilla	0.975	**0.786**	**0.678**	**0.990**	0.687	0.567	**0.762**	0.498	0.186
	Vanilla+I	0.975	0.784	0.674	**0.990**	0.677	0.552	0.757	0.576	0.213
	CFN	0.985	0.760	0.641	0.954	**0.745**	**0.622**	0.730	0.833	**0.335**
	CFN+I	0.985	0.759	0.640	0.954	0.741	0.616	0.727	**0.835**	**0.335**
MScale-HRNet	Vanilla	0.983	0.771	0.674	0.911	0.658	0.519	0.725	0.451	0.191
	Vanilla+I	0.984	0.770	0.670	0.909	0.646	0.502	0.720	0.522	0.207
	CFN	**0.989**	0.758	0.647	0.897	0.697	0.547	0.752	0.807	0.320
	CFN+I	**0.989**	0.757	0.645	0.892	0.690	0.537	0.749	0.810	0.321

* The precision and recall are not in common definition. See Sub-section 4.3.1 for details.

4.3.2. Modeling of Observation Uncertainty

To start, we verified the design of the confusion matrix $\mathbf{M}$ to model the uncertainty in the semantic segmentation stage. We explored two approaches for this purpose. The first approach, referred to as Vanilla, is defined by $\mu(\mathbf{I} + \lambda\mathbf{1})$, where λ is a hyper-parameter and μ is a normalization factor. The second approach is CFN, which is the confusion matrix of the semantic segmentation network in the Mapillary validation data set. During inference, we assigned each cell to the label with the highest probability. We present the quantitative results in Table 1. Our findings reveal that CFN significantly outperforms the Vanilla model in terms of IoU and recall, particularly for crosswalks and lane marks. The result is consistent across both backbone networks. This suggests that utilizing the confusion matrix of the network to model the prediction error in semantic segmentation leads to improved map generation results.

4.3.3. Integration with LiDAR Intensity

To take advantage of the varying reflectivity of different road materials, we begin by filtering out all intensity data that falls below the normalized threshold value of $k = 14$, which we manually calibrated for the Velodyne VLP-16 LiDAR (as shown in Figure 3). During the semantic mapping process, when our model predicts the presence of lane marks, we increase the logarithmic probability of that label by a constant factor γ. This suppresses our prediction of other classes and increases our confidence in predicting lane marks. In Table 1, the models that incorporate intensity data are denoted with a "+I" label. Comparing Vanilla+I to Vanilla, we observe improved accuracy and IoU scores for lane marks, but a slight decrease for roads and crosswalks, suggesting the benefit of integrating intensity data for lane mark prediction. However, this trend is not replicated for CFN+I compared to CFN, indicating that a more sophisticated function may be needed to model LiDAR intensity for further improvement.

4.3.4. Effect of Clipping Range

We conduct experiments in Table 1 by clipping the local dense point maps extracted up to 10 m along the longitudinal axis and -15 to 15 m along the lateral axis of the vehicle, as the semantic segmentation performance decreases significantly beyond this range. The effect of range on the final mapping result can be seen in experiments varying the clipping distance, summarized in Table 2.

The result suggests that a shorter distance yields better mapping performance for the most challenging lane mark class. We observed a similar pattern during the hyperparameter tuning for DeepLabV3Plus-based semantic mapping in our initial work [19] and believed that it was caused by a combination of reduced calibration error, and more accurate semantic segmentation for closer ranges. We notice however that MScale-HRNet produces strong

semantic segmentation for longer ranges, but still exhibits the same trend. This leads us to believe that long-range mapping error is mainly related to camera calibration.

Table 2. Ablation Study on point map maximum clipping distance.

Range	Road			Crosswalks			Lane marks		
	Precision*	Recall*	IoU	Precision*	Recall*	IoU	Precision*	Recall*	IoU
30	0.985	**0.847**	0.702	0.695	0.760	0.495	0.555	0.567	0.182
15	**0.989**	0.836	**0.706**	0.823	**0.766**	**0.560**	0.683	0.761	0.270
10	**0.989**	0.757	0.645	**0.892**	0.690	0.537	**0.750**	**0.810**	**0.321**

* The precision and recall are not in common definition. See Sub-section 4.3.1 for details.

4.3.5. Mapping Results

An example of the global map generated by our CFN+I DeepLabV3Plus model for the entire test region is shown in Figure 7. The figure highlights a region of the map, demonstrating our model's ability to clearly capture and map the static elements of the road.

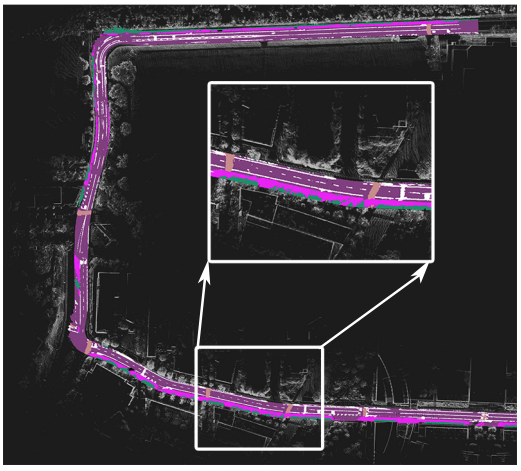

Figure 7. Generated map of testing data set in BEV, displayed on top of the dense point cloud map.

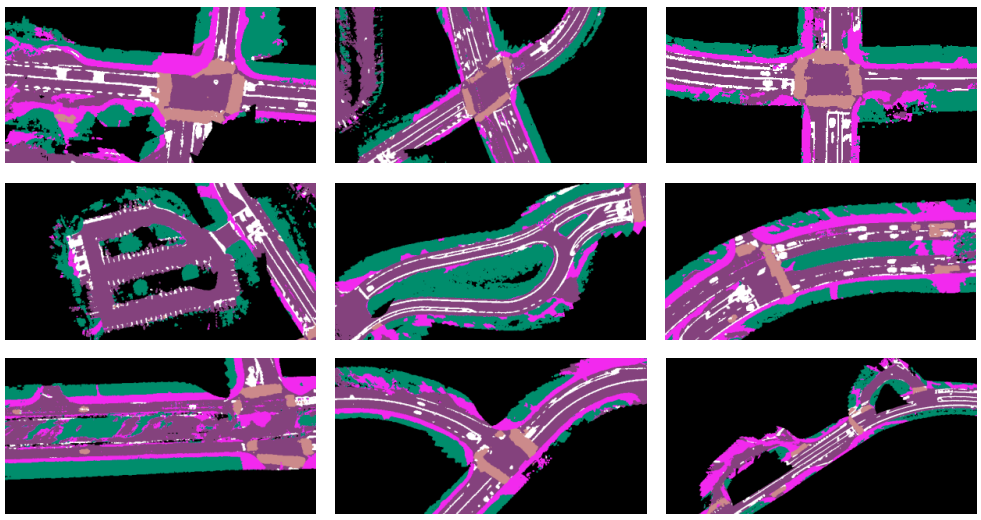

Figure 8. From top to bottom, results in structured environments, unstructured environments, and noisy results.

More examples from testing on the UC San Diego campus are shown in Figure 8. The first row shows results on more common environments such as intersections and road

segments. The second row shows results on less structured environments such as parking lots and curved roads. In these cases, the pipeline can generate visually consistent semantic maps. The last row demonstrates noisy results in a construction zone, intersections with worn road markings, and uncommon road structures. Some of the issues can be addressed by leveraging vehicle-to-infrastructure communication [55].

4.4. Comparison with Different Depth Association Approach

Alternative methods to associate depth exist. In this section, we compare our approach which leverages the dense point cloud map with two approaches to associate depth, using sparse LiDAR scan and planar assumption.

4.4.1. Comparison to Sparse LiDAR Scan

A potential alternative to associating semantic images with depth information is to utilize the real-time point cloud data generated by LiDAR. To accomplish this, we follow a similar mapping approach by projecting the point cloud onto the semantic image frame and constructing the semantic map. The real-time performance of this approach is demonstrated in Figure 9. However, due to the sparsity of point cloud scans from the 16-channel LiDAR used, constructing a semantic map at greater distances is challenging, particularly when the vehicle is moving at higher speeds. Therefore, to enable the creation of semantic maps for longer ranges with a sparse LiDAR, a pre-built dense point cloud map is necessary. With the advances in sensing technology, higher resolution or solid-state LiDARs such as a 128-channel LiDAR can potentially bridge the gap.

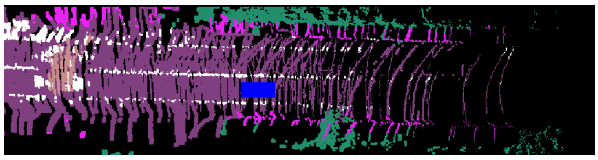

Figure 9. Semantic map generated from real-time LiDAR scan (Black denotes areas not covered by LiDAR).

4.4.2. Comparison to Planar Assumption

We also investigated a different approach, which involves back-projecting the 2D semantic image into 3D space using a homography, assuming a flat ground. This method eliminates black holes in the generated map. However, this approach is not suitable for urban driving scenarios with steep hills or road intersections, as illustrated in Figure 10, since the planar assumption fails under these conditions. Consequently, significant distortion occurs at longer ranges.

Figure 10. Semantic map generated by back-projecting 2D semantic image with a 3D planar assumption.

5. Discussion

We proposed a semantic mapping pipeline that leverages the semantic information from the image and geometric information from the point cloud to generate a probabilistic map in the BEV.

Our experiments highlight the benefits of a probabilistic approach that allows us to capture fine details such as lane marks more accurately, in lieu of semantic segmentation noise. We additionally reviewed the appropriateness of mIoU as a mapping performance metric,

and argue that modified recall and precision better characterize pipeline performance (More details in Sub-section 5.1).

The semantic map generated by our pipeline can provide a rich context for downstream tasks. This includes direct use cases for navigation tasks and behavior prediction that require semantic information to understand the underlying road geometry. For example, in recent work [56,57], a strategy for dynamic trajectory generation for urban driving is proposed. The methods leverage conditional generative models to align coarse global plans to local semantic maps and dynamically regress egocentric trajectories. The semantic features provided by our map can additionally be used as a base for HD Map generation. Combining the dense semantic map from our proposed pipeline with road network topology from approaches such as TopoNet [15] could provide both context and navigation cues respectively. With these aforementioned potentials, the semantic mapping results can still be improved in many aspects.

5.1. IoU and Localization Error

Our analysis suggests that mapping IoU is highly sensitive to localization, and even a minor deviation causes the metric to underrepresent our results. As such, we present the results using metrics that are more tolerant to minor offsets in predictions, which are more consistent with our observations. While localization can be improved, since lane marks are typically 10 cm wide [2] even perfect segmentation with 5 cm localization error drops IoU down to 33%. As such, it is necessary to introduce additional metrics to represent results within an offset tolerance. Recent works [7,13–15] using vectorized representations are evaluated with different metrics that aren't sensitive to localization error.

We notice the mapping offset error is consistent in the egocentric frame, which leads us to believe this is a systematic error of our equipment, and not the semantic mapping pipeline. We believe that better calibration and sensor synchronization can improve mapping results by reducing the offset.

5.2. Semantic Segmentation

Another challenge is the robustness of semantic segmentation. The semantic segmentation model degrades in challenging lighting conditions and unseen environments. For example, as shown in Figure 11, the images may appear over-exposed and trees will cast a shadow on the road on a sunny data. In these scenarios, it is hard to correctly segment the lane marks. Additionally, road constructions and drivable regions that are not well painted compared to the normal road often confuse the network, leading to noisy segmentation.

Figure 11. Semantic Segmentation Degradation from challenging lighting conditions. Left to right, original image, DeepLabV3Plus, MScale-HRNet.

We observe in our ablation study that considering LiDAR intensity values during predictions yields improvements in our performance. VectorMapNet [13] exhibit similar findings, where fusing LiDAR information boosts their performance in challenging environment conditions (puddles on the road). It is clear that multi-sensor approaches increase robustness. In the case of Figure 11 our multi-sensor approach fails to capture the lane

[2] https://safety.fhwa.dot.gov/roadway_dept/night_visib/pavement_marking/ch3.cfm

mark in the final generated global map. Stronger semantic segmentation modules that consider temporal or spatial context are needed. They should be able to handle visual gaps in lane markings, whether due to wear or lighting conditions.

5.3. Mapillary Inconsistency

Another notable issue is the consistency of labels across the dataset used for both Semantic Segmentation Networks. As we mentioned in Sub-section 4.3.1, the Mapillary dataset [18] irregularly fills the dash lanes. For example, as shown in Figure 12 we can see dashed lanes being turned into solid lines in the first example, and in the second example a more complicated zebra-style lane region turned into a fully solid lane label. In the third example, however, the dashed line stays dashed.

Figure 12. On the left we show images from the Mapillary dataset and on the right the visualized labels.

We believe that this inconsistency in training data causes the networks to get confused, and be more biased towards filling in gaps between lane marks when it finds appropriate. As seen in Figure 5 MScale-HRNet is more consistent in filling in the gaps than DeepLabV3Plus. We hypothesize that MScale-HRNet being a more advanced network has a greater ability to learn to fill in (as biased by Mapillary) over DeepLabV3Plus.

This has different ramifications on downstream task performance, as the resulting mapping is affected by the semantic segmentation filling behavior. For navigation tasks, maintaining dashed lanes is important for contextual understanding. Conversely, Zhou et al. [12] use particle filters for road network extraction, where filled lanes would be beneficial.

5.4. Disappearing Lanes and Discretization

The final major issue we observed is under-representation of semantic labels when mapping. Semantic image outputs (especially for MScale-HRNet) show consistent segmentation for lane lines; however, these lines do not necessarily transfer to the final map. By employing a confusion matrix, we account for semantic segmentation error, but we do not account for mapping discretization error.

The experiments ran in this paper were limited to a pixel resolution of 0.2 meters due to memory constraints. This is a large area relative to a lane line's standard width of 0.1 meters. As such, a 0.2×0.2 region that should have been mapped to a lane line may have more road observations that lane line observations. In essence, as a result of our discretization size, this can cause lane line cells to be suppressed by surrounding road observations in the same cell.

An obvious fix would be to increase discretization resolution; however, this comes with multiple problems. In addition to increased memory usage, it requires higher density in observations. Our current maps at 0.2 have holes due to the sparsity of a 16-channel LiDAR point cloud at driving speed. Thus, to counteract this either higher channel LiDARs, slower driving speed, or higher interpolation would be required. Potential exploration could be done by observing distributions of lane line points within a cell, to decide if it represents a lane line or noise. Additionally, discretization can be dropped completely by utilizing vector representations [7,13–15] for lanes instead, that are updated by lane observations.

6. Summary

By incorporating rich information from semantic labels on image frames, our method effectively introduces a statistical approach for identifying road features and mapping them in BEV, as demonstrated by our comparisons to manually annotated maps. This approach can be extended to automate HD map annotation for crosswalks, lane markings, drivable surfaces, and sidewalks, as well as incorporating center lane identifications for path tracking algorithms.

To address the scalability drawbacks of HD maps, future work will involve accounting for road network junctions and forks, allowing for the full automation of road network annotations leveraging graphical methods. While a combination of the proposed techniques may address the scalability and maintenance cost associated with dense point cloud maps for localization, it also opens up new areas of research in high-level dynamic planning. By dynamically estimating drivable surfaces, traffic lanes, lane markings, and other road features, centimeter-level localization may become unnecessary as long as immediate actions can be extracted from a high-level planner. In our future work, we plan to seek solutions for fully automating the HD mapping process while exploring the possibility of dynamic planning without a detailed dense point cloud map.

Author Contributions: Conceptualization, D.P., H.Z., Q.L., H.X. and H.C.; methodology, D.P., H.Z., Q.L., H.X., S.V. and H.C.; software, D.P., H.Z., Q.L., H.X. and S.V.; validation, H.Z., Q.L., H.X. and S.V.; formal analysis, H.X., D.P., H.Z. and Q.L.; investigation, H.Z., S.V., Q.L. and H.X.; resources, H.C.; data curation, D.P. and H.Z.; writing—original draft preparation, D.P., Q.L., H.X., S.V. and H.Z.; writing—review and editing, H.C.; visualization, H.Z., S.V. and D.P.; supervision, H.C.; project administration, H.C. and D.P.; funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by Qualcomm Corporation. The support is gratefully acknowledged.

Data Availability Statement: The data and code presented in this study will be made openly available in https://github.com/AutonomousVehicleLaboratory/semantic_mapping_v2.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiao, J. Machine Learning Assisted High-Definition Map Creation. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC); , 2018; Vol. 01, pp. 367–373. https://doi.org/10.1109/COMPSAC.2018.00058.
2. Douillard, B.; Fox, D.; Ramos, F.; Durrant-Whyte, H. Classification and Semantic Mapping of Urban Environments. *Int. J. Robot. Res.* **2011**, *30*, 5–32. https://doi.org/10.1177/0278364910373409.

3. Sengupta, S.; Sturgess, P.; Ladický, L.; Torr, P.H.S. Automatic dense visual semantic mapping from street-level imagery. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems; , 2012; pp. 857–862. https://doi.org/10.1109/IROS.2012.6385958.
4. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); , 2015; pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.
5. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Honolulu, HI, USA, 2017; pp. 2881–2890. https://doi.org/10.1109/cvpr.2017.660.
6. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision – ECCV 2018; Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y., Eds.; Springer: Cham, 2018; Vol. 11211, pp. 833–851. https://doi.org/10.1007/978-3-030-01234-2_49.
7. Homayounfar, N.; Ma, W.C.; Lakshmikanth, S.K.; Urtasun, R. Hierarchical Recurrent Attention Networks for Structured Online Maps. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; , 2018; pp. 3417–3426. https://doi.org/10.1109/CVPR.2018.00360.
8. Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.; Han, S. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA); , 2023.
9. Maturana, D.; Chou, P.W.; Uenoyama, M.; Scherer, S. Real-Time Semantic Mapping for Autonomous Off-Road Navigation. In Proceedings of the Field and Service Robotics; Hutter, M.; Siegwart, R., Eds.; Springer: Cham, 2018; Vol. 5, pp. 335–350. https://doi.org/10.1007/978-3-319-67361-5_22.
10. Máttyus, G.; Luo, W.; Urtasun, R. Deeproadmapper: Extracting road topology from aerial images. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); , 2017; pp. 3458 – 3466. https://doi.org/10.1109/ICCV.2017.372.
11. Homayounfar, N.; Ma, W.C.; Liang, J.; Wu, X.; Fan, J.; Urtasun, R. DAGMapper: Learning to Map by Discovering Lane Topology. In Proceedings of the IEEE International Conference on Computer Vision; , 2019; pp. 2911–2920. https://doi.org/10.1109/ICCV.2019.00300.
12. Zhou, Y.; Takeda, Y.; Tomizuka, M.; Zhan, W. Automatic Construction of Lane-level HD Maps for Urban Scenes. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); , 2021; pp. 6649–6656. https://doi.org/10.1109/IROS51168.2021.9636205.
13. Liu, Y.; Yuan, T.; Wang, Y.; Wang, Y.; Zhao, H. VectorMapNet: End-to-end Vectorized HD Map Learning, 2023, [arXiv:cs.CV/2206.08920].
14. Liao, B.; Chen, S.; Wang, X.; Cheng, T.; Zhang, Q.; Liu, W.; Huang, C. MapTR: Structured Modeling and Learning for Online Vectorized HD Map Construction, 2023, [arXiv:cs.CV/2208.14437].
15. Li, T.; Chen, L.; Geng, X.; Wang, H.; Li, Y.; Liu, Z.; Jiang, S.; Wang, Y.; Xu, H.; Xu, C.; et al. Topology Reasoning for Driving Scenes, 2023, [arXiv:cs.CV/2304.05277].
16. Can, Y.B.; Liniger, A.; Paudel, D.P.; Van Gool, L. Topology Preserving Local Road Network Estimation from Single Onboard Camera Image. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); , 2022; pp. 17242–17251. https://doi.org/10.1109/CVPR52688.2022.01675.
17. Tao, A.; Sapra, K.; Catanzaro, B. Hierarchical Multi-Scale Attention for Semantic Segmentation, 2020, [arXiv:cs.CV/2005.10821].
18. Neuhold, G.; Ollmann, T.; Rota Bulò, S.; Kontschieder, P. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); , 2017; pp. 5000–5009. https://doi.org/10.1109/ICCV.2017.534.
19. Paz, D.; Zhang, H.; Li, Q.; Xiang, H.; Christensen, H.I. Probabilistic Semantic Mapping for Urban Autonomous Driving Applications. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); , 2020; pp. 2059–2064. https://doi.org/10.1109/IROS45743.2020.9341738.
20. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); , 2016; pp. 3213–3223. https://doi.org/10.1109/CVPR.2016.350.
21. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognit. Lett.* **2008**, *30*, 88–97. https://doi.org/10.1016/j.patrec.2008.04.005.
22. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation, 2017, [arXiv:cs.CV/1706.05587].
23. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA); IEEE: Brisbane, QLD, Australia, 2018; pp. 1887–1893. https://doi.org/10.1109/icra.2018.8462926.
24. Heinzler, R.; Piewak, F.; Schindler, P.; Stork, W. CNN-Based Lidar Point Cloud De-Noising in Adverse Weather. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2514–2521. https://doi.org/10.1109/LRA.2020.2972865.
25. Wang, Y.; Shi, T.; Yun, P.; Tai, L.; Liu, M. PointSeg: Real-Time Semantic Segmentation Based on 3D LiDAR Point Cloud, 2018, [arXiv:cs.CV/1807.06288].

26. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV); IEEE: Qingdao, China, 2017; pp. 537–547. https://doi.org/10.1109/3dv.2017.00067.
27. Jain, J.; Li, J.; Chiu, M.; Hassani, A.; Orlov, N.; Shi, H. OneFormer: One Transformer to Rule Universal Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); , 2023; pp. 2989–2998.
28. Dwivedi, I.; Malla, S.; Chen, Y.T.; Dariush, B. Bird's Eye View Segmentation Using Lifted 2D Semantic Features. In Proceedings of the British Machine Vision Conference (BMVC); , 2021; pp. 6985–6994.
29. Kostavelis, I.; Gasteratos, A. Semantic mapping for mobile robotics tasks: A survey. *Robot. Auton. Syst.* **2015**, *66*, 86–103. https://doi.org/10.1016/j.robot.2014.12.006.
30. Wolf, D.F.; Sukhatme, G.S. Semantic Mapping Using Mobile Robots. *IEEE Trans. Robot.* **2008**, *24*, 245–258. https://doi.org/10.1109/TRO.2008.917001.
31. Sengupta, S.; Greveson, E.; Shahrokni, A.; Torr, P.H.S. Urban 3D semantic modelling using stereo vision. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation; , 2013; pp. 580–585. https://doi.org/10.1109/ICRA.2013.6630632.
32. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); , 2020; pp. 11618–11628. https://doi.org/10.1109/CVPR42600.2020.01164.
33. Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J.K.; et al. Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021); , 2021.
34. Lambert, J.; Hays, J. Trust, but Verify: Cross-Modality Fusion for HD Map Change Detection. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021); , 2021.
35. Wang, H.; Li, T.; Li, Y.; Chen, L.; Sima, C.; Liu, Z.; Wang, Y.; Jiang, S.; Jia, P.; Wang, B.; et al. OpenLane-V2: A Topology Reasoning Benchmark for Scene Understanding in Autonomous Driving, 2023, [arXiv:cs.CV/2304.10440].
36. Li, Q.; Wang, Y.; Wang, Y.; Zhao, H. HDMapNet: An Online HD Map Construction and Evaluation Framework. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA); , 2022; pp. 4628–4634. https://doi.org/10.1109/ICRA46639.2022.9812383.
37. Büchner, M.; Zürn, J.; Todoran, I.G.; Valada, A.; Burgard, W. Learning and Aggregating Lane Graphs for Urban Automated Driving, 2023, [arXiv:cs.CV/2302.06175].
38. Thrun, S.; Fox, D.; Burgard, W. Probabilistic mapping of an environment by a mobile robot. In Proceedings of the Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146); , 1998; Vol. 2, pp. 1546–1551. https://doi.org/10.1109/ROBOT.1998.677346.
39. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. https://doi.org/10.1109/MRA.2006.1638022.
40. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; M. Montiel, J.M.; D. Tardós, J. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. https://doi.org/10.1109/TRO.2021.3075644.
41. Levinson, J.; Thrun, S. Robust vehicle localization in urban environments using probabilistic maps. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation; , 2010; pp. 4372–4378. https://doi.org/10.1109/ROBOT.2010.5509700.
42. Shao, Y.; Toth, C.; Grejner-Brzezinska, D.A.; Strange, L.B. High-Accuracy vehicle localization using a pre-built probability map. In Proceedings of the IGTF 2017 – Imaging & Geospatial Technology Forum 2017, 2017.
43. Wu, J.; Ruenz, J.; Althoff, M. Probabilistic Map-based Pedestrian Motion Prediction Taking Traffic Participants into Consideration. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV); , 2018; pp. 1285–1292. https://doi.org/10.1109/IVS.2018.8500562.
44. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Honolulu, HI, USA, 2017; pp. 5987–5995. https://doi.org/10.1109/cvpr.2017.634.
45. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. https://doi.org/10.1007/s11263-015-0816-y.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Las Vegas, NV, USA, 2016; pp. 770–778. https://doi.org/10.1109/cvpr.2016.90.
47. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); , 2017; pp. 1251–1258. https://doi.org/10.1109/cvpr.2017.195.
48. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); , 2019; pp. 5686–5696. https://doi.org/10.1109/CVPR.2019.00584.
49. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3349–3364. https://doi.org/10.1109/TPAMI.2020.2983686.

50. Yuan, Y.; Chen, X.; Wang, J. Object-Contextual Representations for Semantic Segmentation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August, 2020, Proceedings, Part VI 16; Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J.M., Eds.; Springer: Cham, 2020; Vol. 12351, pp. 173–190. https://doi.org/10.1007/978-3-030-58539-6_11.
51. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May, 2015, Conference Track Proceedings; Bengio, Y.; LeCun, Y., Eds., 2015.
52. Paz, D.; Lai, P.J.; Harish, S.; Zhang, H.; Chan, N.; Hu, C.; Binnani, S.; Christensen, H. Lessons Learned From Deploying Autonomous Vehicles at UC San Diego. In Proceedings of the Field and Service Robotics; Ishigami, G.; Yoshida, K., Eds.; Springer: Singapore, 2019; Vol. 16, pp. 427–441. https://doi.org/10.1007/978-981-15-9460-1_30.
53. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. https://doi.org/10.1007/s11263-008-0152-6.
54. Zhu, Y.; Sapra, K.; Reda, F.A.; Shih, K.J.; Newsam, S.; Tao, A.; Catanzaro, B. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Long Beach, CA, USA, 2019; pp. 8848–8857. https://doi.org/10.1109/cvpr.2019.00906.
55. WU, Q.; SHI, S.; WAN, Z.; FAN, Q.; FAN, P.; ZHANG, C. Towards V2I Age-aware Fairness Access: A DQN Based Intelligent Vehicular Node Training and Test Method. *Chinese Journal of Electronics* **2022**, *32*, 1–15. https://doi.org/10.23919/cje.2022.00.093.
56. Paz, D.; Zhang, H.; Christensen, H.I. TridentNet: A Conditional Generative Model for Dynamic Trajectory Generation. In Proceedings of the Intelligent Autonomous Systems 16; Ang Jr, M.H.; Asama, H.; Lin, W.; Foong, S., Eds.; Springer: Cham, 2022; Vol. 412, pp. 403–416. https://doi.org/10.1007/978-3-030-95892-3_31.
57. Paz, D.; Xiang, H.; Liang, A.; Christensen, H.I. TridentNetV2: Lightweight Graphical Global Plan Representations for Dynamic Trajectory Generation. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA); , 2022; pp. 9265–9271. https://doi.org/10.1109/ICRA46639.2022.9811591.