VI-Map: Infrastructure-Assisted Real-Time HD Mapping for Autonomous Driving

Yuze He[†], Chen Bian[†], Jingfei Xia[†], Shuyao Shi[†], Zhenyu Yan[†*], Qun Song[$], Guoliang Xing[†]

[†]The Chinese University of Hong Kong, Hong Kong SAR, China
[$]Delft University of Technology, Delft, The Netherlands

ABSTRACT

HD map is a key enabling technology towards fully autonomous driving. We propose VI-Map, the first system that leverages roadside infrastructure to enhance real-time HD mapping for autonomous driving. The core concept of VI-Map is to exploit the unique cumulative observations made by roadside infrastructure to build and maintain an accurate and current HD map. This HD map is then fused with on-vehicle HD maps in real time, resulting in a more comprehensive and up-to-date HD map. By extracting concise bird-eye-view features from infrastructure observations and utilizing vectorized map representations, VI-Map incurs low compute and communication overhead. We conducted end-to-end evaluations of VI-Map on a real-world testbed and a simulator. Experiment results show that VI-Map can construct decentimeter-level (up to 0.3 m) HD maps and achieve real-time (up to a delay of 42 ms) map fusion between driving vehicles and roadside infrastructure. This represents a significant improvement of 2.8× and 3× in map accuracy and coverage compared to the state-of-the-art online HD mapping approaches. A video demo of VI-Map on our real-world testbed is available at https://youtu.be/p2RO65R5Ezg.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**;
• **Computing methodologies** → **Vision for robotics**;

*Corresponding author.

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3613280

KEYWORDS

Online HD Map, Scene Understanding, Vehicle-Infrastructure Information Fusion, Autonomous Driving

ACM Reference Format:
Yuze He[†], Chen Bian[†], Jingfei Xia[†], Shuyao Shi[†], Zhenyu Yan[†*], Qun Song[$], Guoliang Xing[†]. 2023. VI-Map: Infrastructure-Assisted Real-Time HD Mapping for Autonomous Driving. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3570361.3613280

1 INTRODUCTION

Autonomous driving systems are poised to revolutionize the transportation industry. HD map is an essential component for autonomous vehicles to perceive and navigate their surroundings. The HD map comprises two core components: geometry and topology[28]. Geometry encompasses the locations and semantics of stationary physical assets related to roadways, such as lanes, road boundaries, lane dividers, and crosswalks. We note that the HD map referred to in this study is vectorized, which utilizes geometric primitives like lines, curves, and polygons to depict road geometry, instead of utilizing raw point clouds. Topology describes lane connectivity, illustrating how lanes or groups of lanes interconnect, influenced by predetermined traffic rules and real-time road conditions. By providing a comprehensive and meticulous representation of the environment, HD map equips autonomous vehicles to comprehend scenes, chart optimal routes, and make context-aware decisions.

Existing approaches to constructing HD maps can be categorized into two primary schemes: offline and online. Offline construction typically involves labor-intensive data collection using specialized survey vans (above $200,000 USD apiece) equipped with a combination of high-end sensors like cameras, LiDAR, GPS, IMU, and radars [3, 42]. The use of SLAM technologies [43, 53] then facilitates the creation of globally consistent maps, followed by either manual or semi-automatical annotation of the maps. Major players in the autonomous driving industry, such as TomTom [47], HERE

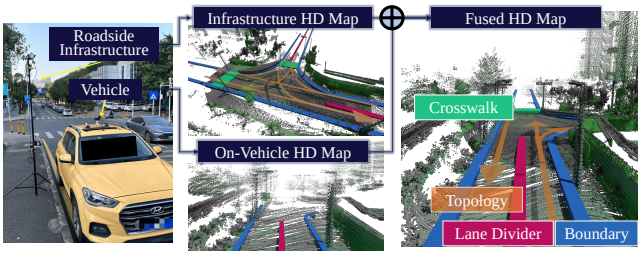


Figure 1: Infrastructure-assisted HD mapping.

[46], and Baidu [3], adopt this approach for its ability to generate highly detailed and comprehensive map information. However, this existing practice suffers from excessive costs. Moreover, the constructed maps can easily become outdated between two mapping iterations [24, 38]. This is because maintaining up-to-date HD maps of large areas can be impractical given the sheer size of road networks and frequent changes of lane connectivity [37]. Consequently, most current offline HD maps are built for highways, leaving city roads largely uncovered [6].

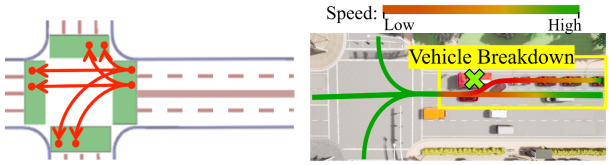
To address the challenges inherent in the offline scheme, recent advancements [9, 22, 27, 29, 30] explore the concept of online HD map learning. This approach aims to estimate the local HD map on-the-fly based on onboard sensor observations such as point clouds of LiDARs and/or images of the surrounding cameras. While this strategy reduces reliance on global offline HD maps and offers the potential for more cost-effective and scalable HD mapping, it is not without limitations. These methods encounter inherent limitations stemming from on-vehicle sensors and dynamic road conditions. These challenges encompass limited sensor field-of-view and variations in sensor data quality attributed to factors like occlusion and swift movement. Additionally, online HD maps typically lack road topology due to the complexities of inferring such logical information in real time. Consequently, current online map construction schemes can be fragile and incomplete.

Interestingly, we discover that intelligent roadside infrastructure or roadside unit (RSU), equipped with sensors and compute units, presents an ideal solution for HD map construction. Notably, a distinctive and crucial attribute unique to roadside infrastructure is its ability to continuously observe road sections while stationary. This capability addresses the shortcomings of both offline and online methods. Firstly, compared to offline approaches, uninterrupted and continuous observation allows infrastructure to promptly update the dynamically evolving HD map. Secondly, in contrast to online techniques, the broad field of view and static, accumulated observations empower infrastructure with comprehensive, unobstructed, and high-quality sensor data.

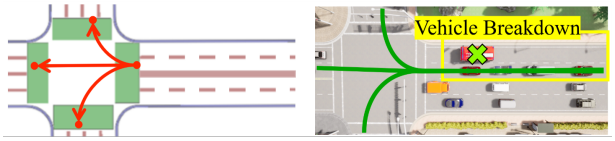
Motivated by the challenges encountered by current methods and the potential of intelligent roadside infrastructure in HD map construction, this paper introduces VI-Map, the pioneering system that harnesses roadside infrastructure to create and maintain HD maps for autonomous driving. In our design, infrastructure employs its own sensors, like LiDARs or 2D/3D cameras, to construct and refresh the HD map. The vehicle then integrates this map with its own HD map in real-time, enhancing/updating the vehicle's scene understanding (as depicted in Fig. 1). The core idea behind VI-Map is to capitalize on the distinct, stationary, and cumulative observations of roadside infrastructure to facilitate accurate and current HD mapping. Specifically, VI-Map first extracts 5 carefully designed concise bird-eye-view (BEV) features from the dense point clouds and accurate vehicle trajectories captured by infrastructure, and then employs them for efficient map geometry construction. VI-Map then leverages the latest vehicle trajectories to estimate and update the current map topology. Finally, VI-Map adopts a new three-stage map fusion algorithm to merge the infrastructure's HD map with the on-vehicle one. We note that VI-Map does not aim to replace existing HD mapping methods. Instead, it offers a critical complementary HD mapping paradigm for autonomous driving, by leveraging increasingly available intelligent roadside infrastructure.

VI-Map offers several key advantages. (i) It transforms massive and unstructured cumulative 3D data into structured, compact, and concise 2D bird-eye-view (BEV) features. These features can be processed with a highly efficient 2D CNN, greatly reducing the compute overhead and enabling its practical deployment on edge devices on infrastructure. (ii) VI-Map generates vectorized HD maps on the infrastructure. Such vectorized representation is highly lightweight, minimizing the communication overhead between infrastructure and vehicles. Additionally, the representation is also compatible with OpenDRIVE [36], a widely adopted industry-standard HD Map data format. (iii) VI-Map does not require the precise location of roadside infrastructures or time synchronization between vehicles and infrastructure. This significantly lowers the barriers to the wider deployment of roadside infrastructure and the adoption of our solution. (iv) VI-Map allows a simple, fast, and flexible deployment, as it allows roadside infrastructure nodes to build their own HD maps independently, which are then fused into the continuous global on-vehicle HD map. As a result, VI-Map can work with mobile RSUs that are deployed on complex and rapidly changing road sections where there is a critical demand for fresh HD map updates. VI-Map can thus offer patches for updating the global HD map at important road sections, complementary to the existing offline and online map construction schemes. This leads to a highly scalable architecture for infrastructure-assisted HD mapping for autonomous driving.

We have implemented VI-Map on a real-world testbed consisting of a modified passenger vehicle, and a mobile RSU



(a) Left: The topology of an outdated HD map cannot reflect the vehicle breakdown. Right: Vehicles fail to make the right steering decision and cause traffic congestion.



(b) Left: The topology of a fresh HD map depicts the blocked and accessible lanes. Right: Vehicles change to the accessible lane in advance.

Figure 2: Vehicle behaviors with the topology extracted from outdated HD map and fresh HD map.

that is deployed at 18 different road sections in two cities, covering up to 5 types of roads. We collected two new datasets using the testbed and a leading simulation platform for autonomous driving [16], respectively. The results demonstrate that compared with the online map construction approaches, VI-Map can extend the local HD map range of a vehicle by 3× and improve the map accuracy by 2.8×, while only incurring 42 ms map fusion delay on the vehicle. At the application level, VI-Map increases the traffic efficiency of problematic road sections by over 5× and improves the ride comfort index by 3.9×. A video demo of VI-Map on our real-world testbed is available at https://youtu.be/p2RO65R5Ezg.

2 MOTIVATING CASE STUDY

This section begins with a case study to understand the limitations of current HD maps on autonomous vehicles (Sec. 2.1). Then, we provide the key insights into the advantages offered by infrastructure in generating timely and high-quality HD maps, underscoring the potential of infrastructure-assisted HD mapping (Sec. 2.2).

2.1 Limitations of on-vehicle HD maps

Offline HD Maps. Autonomous vehicles rely on up-to-date HD maps for both global route planning as well as local behavior decisions and motion planning. However, various road situations such as road construction, congestion, and accidents make road geometry and topology (i.e., lane connectivity) ever-changing, necessitating frequent HD map updates for safe and efficient autonomous driving [24, 38].

Vehicles with outdated HD maps may make wrong behavior and motion planning decisions that do not comply with current road conditions, leading to sub-optimal or even hazardous driving performance. We illustrate an example using CARLA [16], a popular driving simulator that has been used in developing industrial autonomous vehicle systems such as Apollo [2] and Autoware [1].

In this case study, a broken-down vehicle blocks the rightmost lane, leading to a change in road topology where the right lane is no longer connected to other lanes. Fig. 2(a) and Fig. 2(b) illustrate the behavior of vehicles with an obsolete HD map and a fresh HD map, respectively. With the outdated HD map, vehicles suffer from sudden braking and sharp turns, which results in a slower speed or even congestion. In contrast, a fresh HD map allows vehicles to be aware of the lane states in advance and make better decisions, such as choosing the unimpeded left lane, leading to smoother and faster passage through the impacted road section (average speed 1.6 m/s vs. 8.1 m/s). Note that the decisions made by CARLA's autonomous driving agents in the case study may not be optimal. Nevertheless, we still observe considerable benefits of maintaining a real-time HD map, even with the off-the-shelf autonomous driving agent implementations.

Online HD Maps. Online map construction methods use the onboard sensor to obtain up-to-date HD maps. However, such maps are susceptible to the vehicle's limited sensor range and uncertain sensor data quality. For an intuitive illustration, we run the online map construction method [27] on the real vehicle LiDAR data collected from a city street. A detailed result/visualization can be found in the video in the abstract. In particular, the occlusion of LiDAR by surrounding vehicles makes the constructed map incomplete and fragmented. Our evaluation reveals that the integrity of online-generated maps could fall below 25% when half of the road is occluded, which cannot meet the stringent safety requirements for autonomous driving.

2.2 Benefits of Roadside Infrastructure

As discussed in Sec. 2.1, offline HD maps provide a complete but outdated perception, while up-to-date online HD maps constructed by the vehicle only can be inaccurate. This work addresses these issues by exploiting infrastructure-assisted HD map construction and update. Roadside infrastructure can achieve continuous observation of road sections at a standstill, which is ideal for local HD map construction as it offers two key advantages: higher perception quality and the ability to capture real-time topology changes. Roadside infrastructure, including off-the-shelf RSUs [12, 13, 20], is increasingly available. This trend provides an opportunity to leverage roadside infrastructure to improve the HD mapping of autonomous vehicles.

Complete and clear perception. The sensors installed on the infrastructure provide a broader field of view and a longer perception range, and are less likely to be obstructed compared with sensors mounted on vehicles. Moreover, by accumulating the sensor data over time, we can obtain a much more detailed and precise observation of the road. The

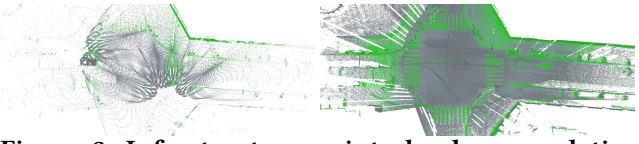


Figure 3: Infrastructure point cloud accumulation brings clearer perception.

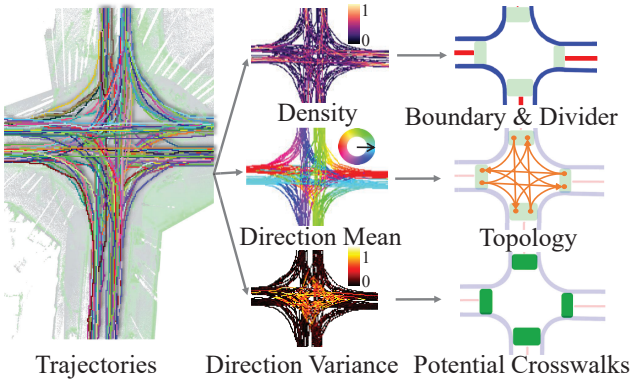


Figure 4: An illustration of what we can extract from vehicle trajectories (Left to right: vehicle trajectories, trajectory attributes, and corresponding map elements.).

left and right images in Fig. 3 illustrate the point clouds of the infrastructure from one frame and from a 10-second superimposition, respectively. The accumulated point cloud from the infrastructure can provide much more details and accurate information like lane lines and road boundaries, which are essential for autonomous driving.

Precise Trajectory Observations. By using LiDAR for vehicle detection and tracking, infrastructure can obtain continuous and accurate trajectories (decimeter accuracy [33]), which is exclusive to infrastructure and cannot be obtained through alternative methods like the vehicle's GPS. Our key observation is that these fresh and precise trajectories provide valuable information for reasoning about the geometry and topology of the road. We demonstrate our findings using a typical real-world intersection (Fig. 4). Trajectory density differentiates lanes and helps find lane dividers and road boundaries. Trajectory direction is consistent with map topology, indicating directed connectivity between lanes. Trajectory direction variance reflects differences in driving direction at the same location, which is high near intersections. Therefore, it can be used for inferring potential crosswalks. These observations can be incorporated together to achieve real-time yet highly lightweight map construction and update.

3 DESIGN OF VI-MAP

3.1 System Overview

We propose *VI-Map*, the first system that exploits roadside infrastructure for real-time HD mapping for autonomous vehicles. Fig. 5 shows the overview of VI-Map. Specifically, VI-Map harnesses distinct data collected by roadside infrastructure, including the accumulated point cloud and precise vehicle trajectories, to build and maintain HD maps. Notably, these two types of data are unique to roadside infrastructure. To the best of our knowledge, our work represents the first attempt to discover and leverage such specific data sources for the purpose of HD map construction.

VI-Map consists of three key components. Firstly, to effectively handle the massive, unstructured, and heterogeneous 3D point cloud and trajectory data, the *geometry construction* (Sec.3.2) module projects these data types into the BEV space. This yields a streamlined, structured, and unified 2D BEV representation. Then, it extracts specific features from both data types, distilling valuable insights tailored for generating map geometry. The geometry module utilizes less fresh but massive point cloud and trajectory data for high-precision geometry prediction. In contrast, the *topology estimation* (Sec.3.3) uses newly arriving trajectories for topology reasoning. An update strategy is designed to identify trajectory changes and trigger the topology update. The resulting precise vectorized map geometry, along with the fresh topology, forms a concise infrastructure HD map. Finally, the *map fusion* (Sec.3.4) executed within the vehicle receives the infrastructure HD map and merges it with the on-vehicle HD map. This module adeptly employs the semantic, proximity, and direction attributes of the vectorized HD map, facilitating the swift integration of infrastructure and vehicle HD maps, thereby providing real-time HD map support for autonomous driving. Furthermore, the vectorized HD map itself is extremely lightweight, leading to minimal communication overhead and robust adaptability to varying communication conditions between infrastructure and vehicles.

3.2 Geometry construction

This module aims to generate the geometry component of the HD map on the infrastructure, making use of two key data sources obtained from the infrastructure: accumulated point clouds and vehicle trajectories. In our design, the map geometry is defined as vectorized representations of four road element types: road boundary, lane divider, lane, and crosswalk. These elements cover the most common elements in HD maps and are consistent with the existing online map construction methods [27, 29]. Specifically, line-based elements such as road boundaries and lane dividers are represented as spline curves, while region-based elements like lanes and crosswalks are represented as polygons.

The geometry construction poses the following challenges: (i) Point cloud and trajectory are two heterogeneous data types and can be hard to deal with simultaneously. (ii) Processing the multi-frame accumulated 3D point clouds on the infrastructure edge device with limited computing resources is a challenge. We approach these challenges by projecting

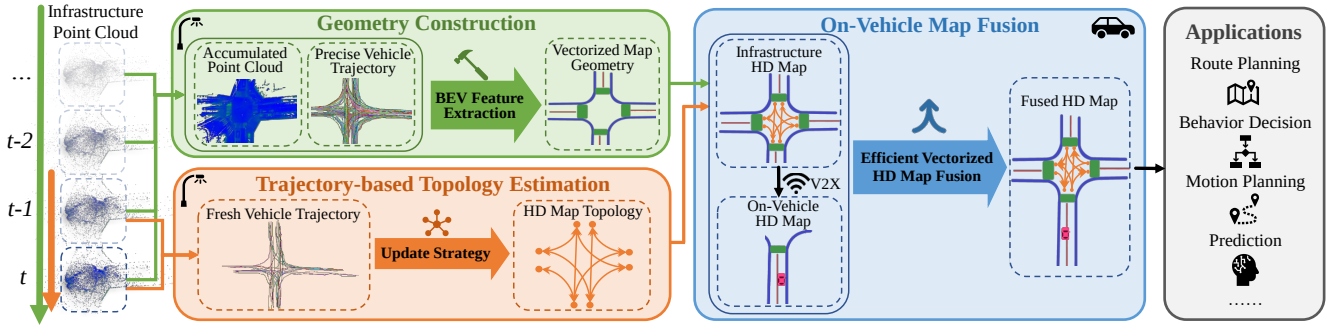


Figure 5: System architecture of VI-Map.

the two types of data to a bird-eye view (BEV) and rasterizing them. As a result, (i) unstructured LiDAR points and trajectories are both converted to a regular grid, enabling simultaneously learning features from both inputs to infer geometry; (ii) such image-like BEV representation can be efficiently processed by a highly lightweight 2D CNN, eliminating the need for resource-intensive 3D point cloud DNNs. This approach enables the construction of maps on a mobile GPU with high efficiency.

Fig. 6 shows the pipeline of the geometry construction module, which includes four steps: BEV projection, feature extraction, instance segmentation, and vectorization. In the initial step, the two types of accumulated data are projected into the BEV space. From this projection, we extract five meticulously crafted features, which are tailored for the inference of map geometry. These features are subsequently inputted into a 2D CNN for instance segmentation. This process yields map element instances represented as grid maps. Ultimately, the grid-based representation of map elements undergoes vectorization, culminating in the creation of a vectorized HD map.

BEV projection. First, we describe the process of obtaining the accumulated point cloud and accurate vehicle trajectories. For each infrastructure point cloud frame, we use a state-of-the-art 3D multi-object tracking method (AB3DMOT [51]) to detect and track vehicles. Frames with no vehicle detection or tracking are accumulated as a static point cloud, while trajectories are saved for tracked vehicles. Then, we project the point cloud and trajectories onto the ground plane in the BEV view. Points in the point cloud with distances greater than 0.5 m from the ground are filtered out, as they may include points of trees or buildings that are irrelevant for map construction. Both the LiDAR points and trajectories are projected onto the ground plane, resulting in a set of 2D points (x, y), where (x, y) represents the coordinates of each point. Finally, we rasterize the points to generate 2D grids. In particular, we scatter the point into pixel location (u, v) with the grid size of (H, W). The rasterization is denoted by $u = [(x - x_{\min})/\alpha]$, $v = [(y - y_{\min})/\alpha]$. The grid height and width are denoted by $H = [(x_{max} - x_{\min})/\alpha]$, $W = [(y_{max} - y_{\min})/\alpha]$, respectively, where $[\cdot]$ is the rounding operation and α indicates the resolution of the rasterization. For instance, when $\alpha = 10$, the cell size is $0.1\,\text{m} \times 0.1\,\text{m}$. Following the previous step, we obtain BEV grid representations for the point cloud and trajectory, respectively, where each cell may contain varying numbers of points. Each point contains some features, we design and extract these features in the next step.

Feature extraction. We design five features from the LiDAR points and trajectory points for HD map construction, i.e., *height*, *intensity*, *density*, *direction mean*, and *direction variance*. For LiDAR points, we calculate the maximum distance to the ground plane in each cell as the *height* feature ($\in \mathbb{R}^1$). It can help us infer the road boundaries since curbs are usually $0.2 \sim 0.3\,\text{m}$ higher than the ground. We also compute the average *intensity* ($\in \mathbb{R}^1$) of points in each cell. High intensity indicates the existence of ground markings, such as lane dividers and crosswalks, because the paints used for ground marking printing are usually reflective. For the trajectory points, we compute three features as motivated in Sec. 2.2, i.e., *density* ($\in \mathbb{R}^1$), *direction mean* ($\in \mathbb{R}^2$), and *direction variance* ($\in \mathbb{R}^1$). The density is computed by counting the number of trajectory points in each cell. We use a 2D direction vector instead of an angle to represent the direction to improve the smoothness of the representation space. The direction variance is defined as $\sigma^2 = 1 - \|\mathbf{R}\|_2$, where $R = \sum_i \mathbf{v}_i/n$ is the average of the n direction vectors $\{\mathbf{v}_i\}$ and $\|\cdot\|_2$ is the L2 norm of the average direction. To summarize, we extract a total of five features for each cell and concatenate them to form a feature map with shape $(H, W, 6)$, where the trajectory direction vector occupies two channels.

BEV instance segmentation. We employ a 2D CNN with a UNet-like structure [41] to perform semantic instance segmentation with the extracted feature map in BEV. The prediction is made for four road element types: lane, road boundary, lane divider, and crosswalk, We train the CNN using combined weighted cross-entropy loss [39] and instance clustering loss [15], which is denoted by $L = \alpha L_{semantic} + \beta L_{instance}$. The CNN takes the feature map of shape $(H, W, 6)$ as input and generates a pixel-wise mask for each individual road

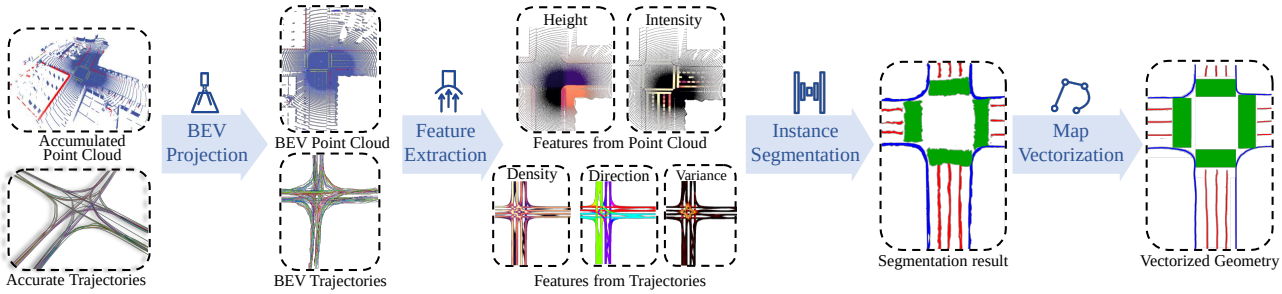


Figure 6: Design of the HD map geometry construction on roadside infrastructure.

element, resulting in an instance mask output. The instance segmentation results also act as inputs for the topology estimation module.

Map vectorization. The instance segmentation maps for each category of road elements can not be directly used by vehicles, as they are not compatible with most autonomous driving frameworks, which typically employ vectorized representation for HD maps. Therefore, we further vectorize the map, generating sparse and compact representations of the HD map. This step also minimizes the map's data volume, hence decreasing the communication overhead during the transmission of the infrastructure HD map to vehicles.

We follow the standard HD map specifications in ASAM OpenDRIVE to model the road elements as splines (for boundaries and dividers) or polygons (for lanes and crosswalks). We first extract each instance as a set of pixel locations $\{(u_i, v_i); i = 1, ..., n\}$ on the segmentation map. To fit each boundary and divider, we seek a cubic function $g(u)$ that minimizes the mean square error: $1/n \sum_i (g(u_i) - v_i)^2$. The solution can be easily found by least squares regression. We fit the minimum enclosing rectangle of $\{(u_i, v_i)\}$ as the geometry representation for lanes and crosswalks.

3.3 Topology Estimation

This module builds and updates the map topology by leveraging the precise vehicle trajectories and instance segmentation results of lanes from Sec. 3.2. We adopt a graph to represent the topology, following the definition in OpenDRIVE. In this context, the graph's nodes and edges correspond to lanes and the connections between them, respectively. The design of this module is based on our key observation that precise vehicle trajectories can be used for inferring map topology and identifying topology changes. This is due to the strong correlation between lane connectivity and trajectory, allowing the inference of one from the other. Additionally, since trajectories are continuous, utilizing fresh trajectory data enables timely estimation and update of the map topology.

The principle of topology construction is as follows: if two lanes are crossed by the same trajectory, they are considered to be connected. The connectivity is directed, which is described by the direction of the trajectory. As discussed in Sec. 2.1, the road topology is dynamic. Thus, we design an update strategy to decide when to trigger an update upon topology changes. The update strategy detects topology changes from both temporal and spatial dimensions and cross-validates them to reduce erroneous updates. Specifically, for the temporal dimension, we assume that, for each lane i, the arrival time t for k vehicle trajectories follows a Poisson distribution. We discretize the continuous arrival time and define it as the number of time units. Each time unit has a duration of 1 seconds. The probability mass function for each lane i is defined as: $P_i(t, k) = \frac{(\lambda_i t)^k e^{-\lambda_i t}}{k!}$. The rate parameter λ_i for each lane i will be dynamically updated based on the set of most recent historical observations on the arrival time for k trajectories within a predefined time interval. In our experiments, we set the time interval to be one-hour long, which is a common update interval in many traffic flow monitoring and forecasting works [40, 45]. We then use a χ^2 test with a significance level of 0.05 to determine for each lane i if the historical observations follow the current Poisson distribution. In the spatial dimension, we compute the three trajectory features, trajectory density, direction mean, and direction variance for the most recent k trajectories. Then we calculate the cross-entropy loss l between the current trajectory features and those in the one-hour time interval. The map topology is updated only if $p > 0.05$ and l exceeds a specified threshold.

3.4 Map fusion

This module runs on the vehicle and aims to merge the received HD map from the infrastructure with the on-vehicle HD map. This involves addressing the map fusion problem, i.e., find the coordinate transformation between the two maps and use this transformation to integrate them. Existing map fusion techniques are primarily developed for scenarios like multi-robot cooperative SLAM, where the maps being fused are occupancy grid maps. However, these existing methods cannot be directly adapted to our context due to the distinction in map types — we aim to fuse two vectorized HD maps. To the best of our knowledge, there is currently no map fusion method with the exact same settings as ours.

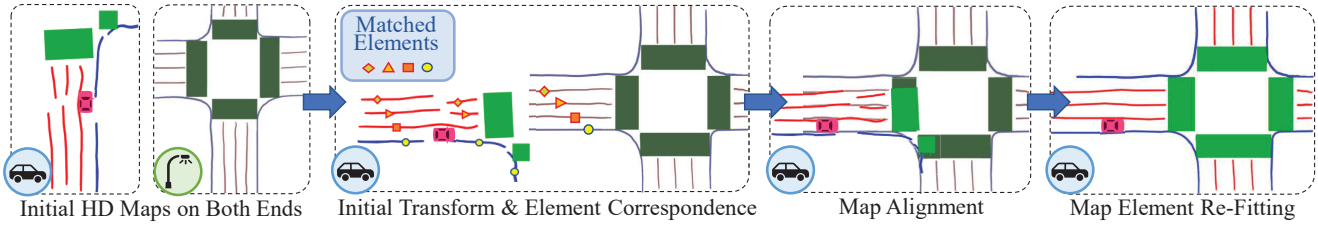


Figure 7: Design of HD map fusion on the vehicle.

To this end, we propose a three-stage map fusion algorithm to merge the infrastructure HD map and the vehicle HD map. Fig. 7 shows a pipeline of this module. Firstly, we exploit the semantic attributes and spatial proximity of the map elements to establish their correspondence. Subsequently, we utilize the directional characteristics to determine the transformation between the two maps, relying on their established correspondences. This transformation is then employed to align the two HD maps accurately. Lastly, we refit each pair of corresponding map elements into a unified representation, culminating in the creation of the final merged HD map.

Stage 1: Map element correspondence.. To merge two HD maps, each having multiple road elements, we need to first find element correspondence between the infrastructure and vehicle. We commence by transforming the infrastructure HD map from its infrastructure coordinate to the vehicle coordinate using an initial transformation, which can be easily derived based on the infrastructure and vehicle poses obtained from GPS and IMU data. This preliminary transformation can be inaccurate due to GPS errors, which will be refined in the next stage. Then, within the vehicle's HD map, we adopt a straightforward approach whereby we match each map element with its counterpart in the infrastructure HD map. This matching is established based on shared semantic label and proximity, with the corresponding element being the one in the infrastructure HD map that holds the same semantic label and exhibits the shortest distance. The distance between two elements is defined as the distance between their nearest endpoints. One fact is that an element in the infrastructure HD map might correspond to multiple elements within the vehicle HD map, particularly when occlusions in the vehicle's perspective lead to the fragmentation of map elements. Notably, we leverage the semantic labels of map elements to enhance matching accuracy and efficiency. Specifically, we exclusively select elements belonging to the same semantic category as potential corresponding pairs. This simple yet effective approach has demonstrated commendable results in establishing correspondence, even in cases where the initial transformation is inaccurate.

Stage 2: Map alignment.. After we find all the corresponding element pairs, we eliminate the error in the initial transformation by refining the alignment between all of the element pairs. Note that we do the alignment only using line-type elements (splines) since it reveals more positional information. We find the optimal rigid transformation by minimizing a novel direction-aware chamfer distance between two spline curves. Specifically, we first convert the splines back to points by sampling with a fixed interval. We then compute the tangent directions (d_u, d_v) of each point. Together with position (u, v), we assign a tuple $\psi = (u, v, d_u, d_v)$ for each point, and we got two tuple sets from infrastructure and vehicle $\{\psi_i\}$, $\{\psi_v\}$. Next, in each iteration, we first find tuple correspondences by searching for the closest tuple in $\{\psi_i\}$ for each ψ_v. The distance is defined as the Euler distance between two tuples. We then estimate a rigid transformation using a weighted root mean square, $w = \cos(\theta_v - \theta_i)^\gamma$, where γ is a hyperparameter that controls how much bias the estimation should lean towards points that have similar directions. We average all the transformations estimated from each corresponding element pair. The mean value is then used to transform the splines on the infrastructure. We then repeat the above process for several iterations. We find that five iterations can already lead to convergence. Finally, we get a refined transformation that can align the map element from the infrastructure to the vehicle.

Stage 3: Map element re-fitting.. Finally, we merge each corresponding pair into one union map element. For line type element, we densely sample points on the two elements, and repeat the fitting process as illustrated in Sec. 3.2. For regional features, We simply take the union of a pair of corresponding elements. We now have an integrated HD map that is aligned with the vehicle perspective, which can be used in the downstream tasks.

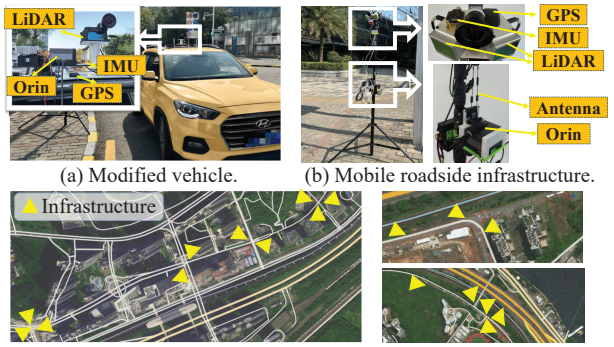
4 TESTBED AND DATASET

In this section, we will present the testbeds of VI-Map and datasets collected for evaluations. The research has been granted IRB approval.

Testbed. We implement VI-Map on both a real-world setup and CARLA simulator for extensive data collection and evaluation. Fig. 8 shows our real-world setup, which includes a modified vehicle and a customized mobile pole as the roadside infrastructure unit. Each pole equips two Livox AVIA LiDARs [31] at the height of 5 m. A Livox HAP LiDAR [32] is mounted on top of a vehicle at about 1.7 m. The mobile

Table 1: Comparisons between the hardware and costs of the RSUs in existing works [25, 48] and in VI-Map.

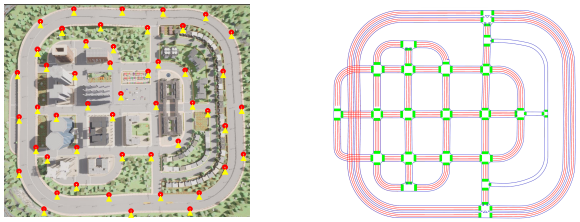
Works	Sensors		Compute units		Communication units		Total cost per RSU
[25]	UMRR-0C radar Basler acA1920-50gc camera	×4 ×4	INTEL Xeon E5-2630v4 2.2GHz CPU NVIDIA Tesla V100 SXM2 GPU	×2 ×2	✗		~USD $20,000
[48]	Velodyne VLP-16 LiDAR	×1	Laptop PC (CPU Core i7 8-cores) with 16GB RAM	×1	Intel Wi-Fi 6 AX200	×1	~USD $4,700
VI-Map	Livox AVIA LiDAR NEO-M8T GPS HWT9052-485 IMU	×2 ×1 ×1	Nvidia Jetson Orin	×1	IoTwrt AR350 switch	×1	~USD $5,600



(a) Modified vehicle. (b) Mobile roadside infrastructure.

(c) 18 tested locations in two cities.

Figure 8: A real-world testbed deployed for VI-Map data collection and system evaluation.



(a) Locations of roadside infrastructure. **(b) HD map parsed from OpenDRIVE file.**

Figure 9: Town 5 map used in CARLA dataset.

pole is installed with an NVIDIA Jetson Orin and an 802.11ac WiFi router for wireless communication with vehicles. The test vehicle (see Fig. 8) carries another Orin to run an online HD map construction baseline [27] and VI-Map's map fusion code. Both vehicle and mobile pole install a NEO-M8T GPS [49] and an HWT9052-485 IMU [52] to estimate the pose.

Here we present a cost analysis for the mobile RSU, providing valuable insights into its practical implementation in real-world scenarios. Table 1 presents comparisons between the hardware and costs of the RSUs used in existing works [25, 48] and in VI-Map. Each RSU primarily comprises several components: sensors, compute units and communication units. It is worth noting that the cost of RSUs can potentially be further curtailed, particularly as LiDAR prices continue to decrease over time. In order to ensure comprehensive road coverage, the recommended deployment distance between two adjacent RSUs is approximately 50 meters.

Existing autonomous driving public datasets [8, 18] only provide vehicle-centric point cloud data, which cannot be used for the evaluation of VI-Map. Therefore, we collect two new datasets, one from our real-world testbed and one generated by the CARLA simulator [16]. Both datasets contain a constant period of point clouds from both the infrastructure and the vehicle. The real-world dataset is collected from 18 road sections in two cities, covering a variety of road types (T-junctions, crossroads, bends, straight roads, etc.) and varying lane numbers ranging from one to six lanes. Table 2 summarizes the two datasets. The details are as follows.

Table 2: Summary of two new datasets, where "infr." represents infrastructure and "veh." represents vehicle.

Dataset	#road sections	#road types	#GT HD maps (infr./veh.)	#point cloud frames (infr./veh.)
Real-world dataset	18	5	18/4586	16752/4586
CARLA dataset	50	8	50/25300	89761/25300

Real-world Dataset. During the collection of the dataset, the average collection time of the infrastructure point cloud is 3 minutes (up to around 10 minutes). The vehicle speed is 25 km/h on average (up to around 30 km/h). To obtain the ground-truth HD maps, we first register each point cloud pair of infrastructure and vehicle, and then project the fused point clouds to BEV and rasterize it to images, with a resolution of 0.15 m/pixel. Then, we manually annotate the map geometry (polylines and polygons) in the images using the CVAT tool [14]. We annotate an HD map for each pair of infrastructure-vehicle point clouds and individual vehicle point clouds if it has little overlap with the infrastructure point cloud. Separate ground-truth HD maps of infrastructure and vehicles are obtained by cropping the fused HD maps. The map topology for each road section is manually annotated as an $n \times n$ binary array, where n denotes the number of lanes in a road section, and values of 1 and 0 indicate the connection and disconnection of two lanes, respectively. As a result, we obtain one infrastructure HD map and multiple vehicle HD maps for each road section, which are used as the training data of our geometry construction module and the online map construction baseline [27], respectively. We also annotate the vehicles in the infrastructure point cloud for the training data of tracking algorithm AB3DMOT [51].

CARLA Dataset. We also render a dataset in CARLA [16]. We configure roadside infrastructures at different locations of Town 5 in the CARLA ecosystem. Each infrastructure and

the ego vehicle is installed with a LiDAR with 32 channels and 360° field of view, which is consistent with mainstream autonomous driving datasets [8, 18]. We generate 200 vehicles wandering around the whole town to simulate the traffic flow. We parse the map file (in .xodr file format defined by OpenDRIVE) provided by CARLA to generate the ground-truth HD map of the whole town. Then, the local HD map of each infrastructure and each frame of the vehicle point cloud is obtained by cropping the town map according to the pose of each subject. Fig. 9 shows all the infrastructure locations and the global HD map. The CARLA dataset covers 50 road sections in a town of 439 m × 509 m. Moreover, CARLA enables the simulation of events that causes changes in road topology, such as vehicle breakdown or accidents, which can be dangerous to simulate in the real world. We simulate vehicle breakdown in road sections with three different road types, i.e., T-junctions, intersections, and two-way two-lane straight roads. We also simulate different degrees of road-marking blurriness, i.e., mild (< 20%), moderate (50% − 60%), and severe (> 90%).

5 EVALUATION

5.1 Evaluation Setup and Metrics

5.1.1 Evaluation setup.. We set the height H and width W of the BEV grid (r.f. Section 3.2) to 300, according to the road coverage of the infrastructure point cloud. The geometry construction model of VI-Map, the vehicle tracking model [51], and the baseline model [27] are trained or fine-tuned using the two datasets introduced in Sec. 4. The training is performed on a server equipped with Intel Xeon Silver 4210 CPU and one NVIDIA RTX 2080Ti GPU. We adopt 10-fold validation for the three models.

5.1.2 IoU, CD_P, CD_L, CD, P, R.. We adhere to the evaluation methodologies in [19, 27, 29] and employ these widely recognized evaluation metrics to assess the precision of map geometry. Intersection-over-union (*IoU*) and Chamfer distance (*CD*) are used as metrics for evaluating line-type elements such as boundary and divider. For the regional element crosswalk, we use *IoU*, precision (*P*), and recall (*R*) as metrics. Specifically, *IoU* is an Eulerian metric that measures the pixel semantic differences between the predicted map and the ground truth, which is denoted by $\text{IoU}\left(\mathcal{D}_P, \mathcal{D}_G\right) = \frac{|\mathcal{D}_P \cap \mathcal{D}_G|}{|\mathcal{D}_P \cup \mathcal{D}_G|}$, in which $\mathcal{D}_P, \mathcal{D}_G \subseteq \mathbb{R}^{H \times W \times D}$ are dense representations of map elements (curves and polygons rasterized on the BEV grid), $\mathcal{D}_P$ is the prediction and $\mathcal{D}_G$ represents the ground truth, H and W are the height and width of the BEV grid, D is the number of map element categories and $|\cdot|$ denotes the size of the set. *CD* is a Lagrangian metric that measures the spatial distances of vector geometric shapes (curves or splines). CD_{Dir} is the directional Chamfer distance and *CD* is the bi-directional Chamfer distance. Specifically, CD_{Dir} is defined as $\text{CD}_{\text{Dir}}\left(\mathcal{S}_1, \mathcal{S}_2\right) = \frac{1}{\mathcal{S}_1} \sum_{x \in \mathcal{S}_1} \min_{y \in \mathcal{S}_2} \|x - y\|_2$, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are the two sets of points sampled on the predicted curve and the ground-truth curve. CD_P denotes the *CD* from prediction to label (equivalent to precision), while CD_L denotes the *CD* from label to prediction (equivalent to recall). *CD* is defined as $\text{CD}\left(\mathcal{S}_1, \mathcal{S}_2\right) = \text{CD}_{\text{Dir}}\left(\mathcal{S}_1, \mathcal{S}_2\right) + \text{CD}_{\text{Dir}}\left(\mathcal{S}_2, \mathcal{S}_1\right)$. The precision and recall metrics used for regional elements are defined as $P = \frac{|\mathcal{D}_P \cap \mathcal{D}_G|}{|\mathcal{D}_P|}$ and $R = \frac{|\mathcal{D}_P \cap \mathcal{D}_G|}{|\mathcal{D}_G|}$, respectively. CD_P and CD_L reflect the accuracy and completeness of the predicted map, respectively.

5.1.3 Ride comfort, average passing time, traffic throughput.. We evaluate the benefits VI-Map can bring using these three metrics related to user experience. The ride comfort is quantified using longitudinal acceleration a_x and lateral acceleration a_y of vehicles, widely acknowledged as key indicators for evaluating ride comfort [23, 35]. Lower acceleration value corresponds to better ride comfort. The average passing time is the average time for a vehicle to traverse a road section. The traffic throughput is the number of vehicles passing through a road section during a fixed period.

5.1.4 Response time, success time, success rate.. We use these three metrics to evaluate the performance of the map topology update strategy. Response time is defined as $t_r - t_c$, where t_c and t_r are the moment when the road topology changes and the moment the infrastructure map topology is updated, respectively. Success time is given by $t_s - t_c$, where t_s is the moment when the map topology is updated correctly. t_s can be different from t_r as the update can be wrong. The success rate is the number of correct topology updates divided by the number of all updates.

5.2 End-to-end Evaluation

We evaluate the end-to-end system performance of VI-Map on a real-world road with four road types, i.e., two-way single crossroad, T junction, four-lane straight road, and bend. Fig. 10 shows the trace that the vehicle enters and exits a crossroad with a mobile pole on the side. The gray dots are the vehicle point clouds along the trace. The colored dots denote the vehicle's GPS positions, where different colors represent the *IoU* of the constructed HD map. At the beginning and end of the trace, the *IoU* is 40% with only the on-vehicle HD map available. When the vehicle enters the coverage of the infrastructure, the *IoU* increases to 80% with the fused HD map. The result shows that VI-Map can benefit the vehicle with the precise infrastructure map and thus assist the vehicle in going through complicated road sections safely. We also measure the end-to-end latency of VI-Map. VI-Map achieves an end-to-end latency of 37 *ms* on average and 42 ms maximum. This means that VI-Map can work

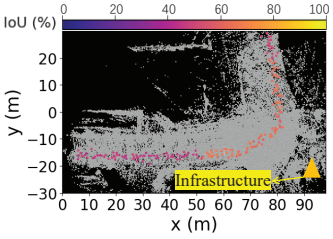


Figure 10: HD map *IoU* scores of a vehicle passing a road section where an infrastructure is located.

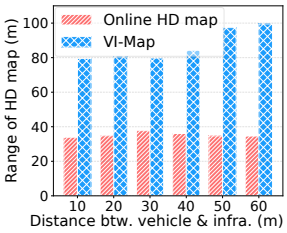


Figure 11: VI-Map extends online HD map range.

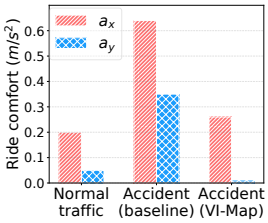


Figure 12: VI-Map improves ride comfort.

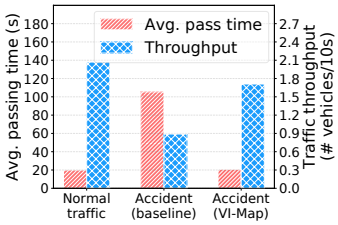


Figure 13: VI-Map improves traffic efficiency.

in real-time since most autonomous vehicles have a sensor frame rate of 10 Hz. Additionally, we observe gaps between the IoU at the beginning and end of the trace, even though there are both vehicle HD maps only. This is because the road markings at the start place are badly worn, resulting in a very low IoU of the online map. We evaluate the influence of the road marking incompleteness in Section 5.6.

5.3 Benefit of VI-Map

In this section, we evaluate the benefits VI-Map can bring to autonomous vehicles. The key observations from this section are: (i) The precise map geometry of VI-Map complements and extends the vehicle's online HD map beyond the limitations of single-vehicle perception. (ii) The fresh map topology of VI-Map enables vehicles to make better behavior decisions under the current road conditions, benefiting passengers, vehicles, and transportation systems.

5.3.1 Extend online HD map range. We evaluate the extended range brought by VI-Map when the distance between the vehicle and infrastructure changes. Fig. 11 shows that the fused HD map by VI-Map can provide double the range than the online HD map only, even when the vehicles are very close to the infrastructure (i.e., < 10 m). This is because the infrastructure can provide a wider field of view thanks to its high altitude, and is less prone to occlusions than vehicles. The extended range of HD map is the basis that VI-Map can benefit many downstream tasks and improve safety in autonomous driving.

5.3.2 Improve ride experience and traffic efficiency. VI-Map provides vehicles with up-to-date map topology to support downstream tasks such as decision making and motion planning. We evaluate how the map freshness affects the behavior of autonomous vehicles, and what impacts and outcomes these behaviors lead to. We construct a crossroad in CARLA, in which vehicles with hard-coded AI use HD map and surrounding information for autonomous driving. We perform the following steps. First, all autopilot vehicles driving in the town are equipped with offline HD maps provided by CARLA. Second, we set a vehicle to stop suddenly to simulate a lane-blocking scenario, which causes the road topology to change and the offline HD map to become obsolete. We record the data traces of the road section before and after the topology change, including the infrastructure point clouds and the timestamps, accelerations, and positions of all vehicles passing through the road section. Third, we feed the infrastructure point cloud into VI-Map to generate an updated HD map. We manually update the topology of the corresponding road section in the CARLA map and load the new map to all autopilot vehicles. Lastly, we let vehicles drive with the new map and record the new data traces.

We calculate the ride comfort, average passing time, and traffic throughput. Fig. 12 shows the average ride comfort under the three situations. It shows that driving with an obsolete map can lead to a bad ride experience due to frequent stop-and-go and sharp turns. VI-Map significantly improves ride comfort by 3.9× compared with obsolete map, which even matches the comfort level in non-accident scenarios. This is due to the VI-Map's ability to update the map in a timely manner, allowing vehicles to make better decisions and resulting in a smoother and more comfortable ride experience. Fig. 13 demonstrates the average vehicle passing time and traffic throughput of the road section. Compared with the outdated map, VI-Map reduces the time for vehicles passing through problematic road sections to one-fifth and improves traffic throughput by 2×.

5.4 Performance of VI-Map

5.4.1 Geometry Construction. We evaluate the geometry construction of VI-Map on the real-world testbed. We compare VI-Map with two baselines: (i) a state-of-the-art online HD map construction method called HDMapNet with vehicle data only[27]; (ii) A modified method based on HDMapNet, in which we fuse the raw point cloud of the infrastructure to the vehicle. Fig. 14 shows an example of map construction results of an intersection using two baselines and VI-Map. The blue and red splines as well as the green rectangles directly stem from the output of VI-Map. The orange arrows are manually annotated to display the system's generated map topology, whose original form is a graph (Sec. 3.3). We can find that the HD map generated by VI-Map provides a wider range compared to vehicles only. Moreover, VI-Map

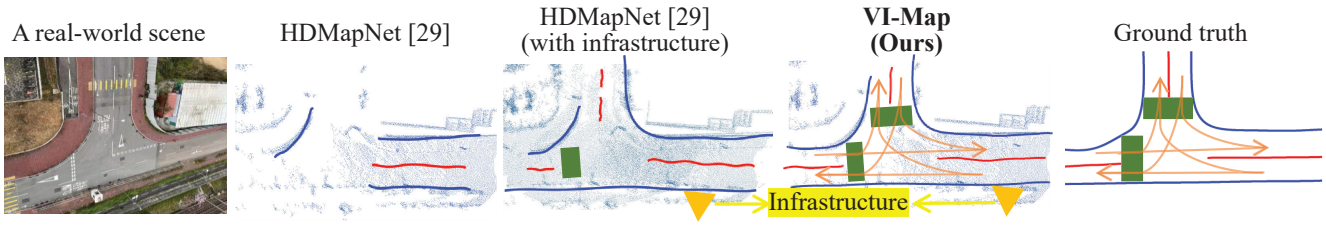


Figure 14: HD mapping results of different methods in a real-world scene.

Table 3: HD map geometry accuracy on the real-world dataset. A higher *IoU* (%) is better. A lower *CD* (*m*) is better.

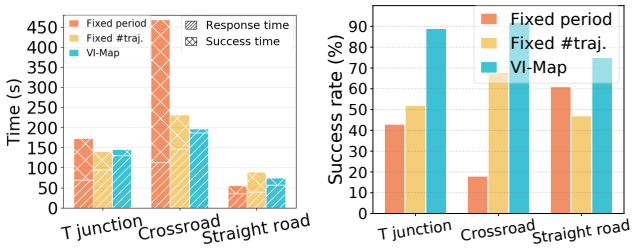
Method	Road boundary				Lane divider				Crosswalks			All classes				Topology
	IoU	CD_P	CD_L	*CD*	*IoU*	CD_P	CD_L	*CD*	*IoU*	*P*	*R*	*IoU*	CD_P	CD_L	*CD*	
HDMapNet [27]	52.73	0.71	0.96	0.80	47.16	0.62	1.17	0.89	58.21	62.56	49.37	52.70	0.67	1.07	0.85	No
HDMapNet(with infr.)	63.52	0.53	0.86	0.77	79.65	0.46	0.87	0.72	70.23	82.26	74.94	71.13	0.50	0.87	0.75	No
VI-Map (Ours)	**86.35**	**0.37**	**0.35**	**0.36**	**90.49**	**0.28**	**0.21**	**0.23**	**85.71**	**92.82**	**95.36**	**87.52**	**0.33**	**0.28**	**0.30**	Yes
VI-Map w/o traj.	80.73	0.41	0.47	0.45	82.32	0.39	0.48	0.44	79.89	88.62	80.75	80.98	0.40	0.48	0.45	No

provides topology and a more accurate HD map than the two baselines.

Then, we evaluate VI-Map on the real-world dataset. Table 3 shows the results using the metrics defined in Sec. 5.1.2. VI-Map outperforms the baselines in all metrics. In particular, VI-Map delivers 16%-35% higher *IoU* compared with HDMapNet. VI-Map achieves decimeter-level map precision (i.e., *CD*) with an average error of 0.3 m. The results also show that HDMapNet can be benefited from the infrastructure's point cloud (20% improvement in *IoU*), thanks to its unobscured field of view. However, VI-Map exhibit much better performance than HDMapNet with infrastructure data, which is around 60% improvement in *CD*. This is because the overlay of two single-frame point clouds from the infrastructure and vehicle is still too sparse to predict the location of map elements precisely. VI-Map exploits accumulated point clouds and precise trajectories, which reveal more details and clues to locate map elements precisely.

We also observe that the infrastructure point cloud contributes more significantly to the improvement of HDMapNet for lane dividers compared to road boundaries and crosswalks. This is because the lane divider occupies a smaller area and is mainly inferred by point intensity, overlaying infrastructure point cloud makes the intensity more pronounced in the particular small area. Further, we find that the gap between CD_P and CD_L of VI-Map is much smaller ($\sim$ 13%) than the baselines. This is because the HD maps generated by the two baselines can be incomplete due to sparse point clouds and potential occlusions, while the complete HD map of VI-Map can ensure consistency between the two metrics. Additionally, we find that compared with Euclidean metric *IoU*, VI-Map improves the two baselines more on Lagrangian metric *CD* (35%-26% improvement vs. 65%-60% improvement). This is because VI-Map generates continuous map elements and fits them into vectorized shapes. In addition, we conduct an ablation study to evaluate the effectiveness of trajectories in map geometry construction. Table 3 also shows VI-Map without the trajectory input, in which it exhibits 7% less in *IoU* and 50% increase in *CD*. The result supports our design that the accurate trajectories imply features for inferring map geometry, and combining it with point cloud features leads to better performance.

5.4.2 Topology Estimation. We evaluate the performance of our update strategy of HD map topology by examining response time, success time, and success rate as defined in Sec. 5.1.4. We compare our update strategy in Sec. 3.3 with two baselines: (i) Fixed period of time: The topology is updated with a fixed time period T. (ii) Fixed number of trajectories: The topology is updated after observing a fixed number of additional trajectories n. For the former method, we set T to the traffic light cycle duration for intersections and 60 seconds for straight roads. For the fixed number of trajectories method, we set n to $3 \times$ *Number of Lanes*. It is important to note that these values were chosen after an extensive search to identify the best-performing parameters for each baseline approach. As the performance of the map update can vary with different road types and traffic conditions, we evaluate under three different road conditions: a town center crossroad, a T-junction on the outskirts of town, and a four-lane straight road. Specifically, we simulate the same situation as described in Sec. 5.3.2 under these three different road conditions, five times each. Fig. 15 shows the performance of VI-Map and the two baselines. Fig. 15(a) shows the response time and the success time of the three update approaches. VI-Map delivers the shortest success time under all road conditions, though its response is not the fastest. The two baselines respond quickly but take a much longer time to get a correct topology update. VI-Map considers the topology update per lane level, while the two baselines do not. Thus, the trajectories collected under the two baseline strategies



(a) Response & success time. **(b) Success rate.**

Figure 15: Topology estimation performance of different update strategies at different road types.

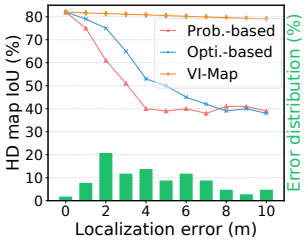


Figure 16: HD map *IoU* scores of fused maps generated by different map merging methods over vehicle localization errors.

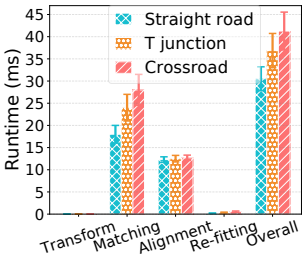


Figure 17: Run-time latency of each step of VI-Map on the vehicle at different road types.

can be unevenly distributed among lanes, with some lanes not having any trajectories, which can result in update failure. Fig. 15(b) shows the success rate. VI-Map achieves an average success rate of 84%, surpasses the two baselines by a large extent of 30%-44%. Our update strategy integrates both temporal and spatial changes to detect the change in topology, and avoid false updates with cross-validation.

5.4.3 Map Fusion. We then experiment on the map fusion method using the real-world dataset. This module runs on the vehicle and is responsible for merging the infrastructure map with the vehicle map. We implement two baseline methods by adapting the ideas of typical map merging methods in collaborative SLAM to our scenario: the probability method [5, 26, 54] and the optimization method [4, 10, 34]. We rasterize our vectorized map to a grid map and follow the implementation of these methods. Fig. 16 shows the performance of the fused map obtained by applying these fusion methods. The curves in Fig. 16 show the map IoU of the three fusion methods under different vehicle localization errors, respectively. The green bars in Fig. 16 present the distribution of GPS localization errors. The result shows that VI-Map achieves more than 40% *IoU* improvement over baselines when the localization error is greater than 8 m. As the localization error increases, the performance of the two baselines decreases sharply, whereas VI-Map maintains a map IoU of over 78% for all tested localization errors. This is due to the fact that VI-Map views the map as individual map elements and leverages instance-level correspondence of these elements to accurately align and merge the two maps. In contrast, the two baseline approaches view the map as a whole and rely solely on the initial transformation to align the two maps.

5.5 System Overhead

5.5.1 Each step delay on vehicle. We measure the run-time latency of VI-Map's individual steps on the vehicle and the whole map fusion process. Since the runtime of map fusion may be affected by the road complexity, i.e. the number of map elements, we test on three different road types in the real world. Fig. 17 shows the runtime of each step, where the error bars indicate the lowest 5% and 95% of the runtime across all frames with the same setting. The results show that the overall on-vehicle compute time of VI-Map is less than 50 ms under all road types, which can meet the stringent real-time requirement of autonomous driving applications (prediction [17], decision making [11], and motion planning [50]). Therefore, VI-Map enables real-time HD mapping on vehicles with the assistance of the roadside infrastructure. The matching step takes half of the total compute time, as it involves distance calculation between pairs of map elements. We note that this runtime can be further accelerated by computing on GPU. The runtime of the transform and align steps do not grow with the road complexity as they are independent of the number of map elements.

5.5.2 Communication overhead. We compare the communication overhead of VI-Map with approaches that transmit raw point cloud data (e.g., HDMapNet with infrastructure) or rasterized map data [4, 10, 26, 54]. We measure the data transmission time traces using the real-world testbed in Sec. 4, in which the vehicle and infrastructure communicate using an 802.11ac Wi-FI network with 80 MHz bandwidth. The average data volume and the transmission time are shown in Table 4. VI-Map transmits 71.2 KB of vectorized map representation (control points of splines, vertices of polygons, and topology graph) in an average of 13.6 ms. VI-Map reduces the data transmission volume and time by about 42× and 40× compared with transmitting raw point cloud data, and 22× compared with transmitting rasterized map data. Therefore, VI-Map can be deployed with a wide range of V2X networks that even have low communication bandwidth.

5.6 System Robustness

As observed in Sec. 5.2, online HD maps can be significantly affected by worn ground markings. In this section, we evaluate VI-Map under different incompleteness of road markings. Specifically, we mask different ratios of ground marking points in the point cloud using CARLA. Table 5 shows that as the incompleteness of ground markings increases, all metrics of VI-Map decreases slightly and steadily, maintaining

Table 4: The average size of shared data and the transmission time via an 802.11ac network.

Shared data type	Shared data size	Transmission time
Raw point cloud	2.92 MB	544.6 ms
Rasterized map	1.54 MB	304.3 ms
Vectorized map of VI-Map	**71.2 KB**	**13.6 ms**

Table 5: Accuracy of VI-Map's HD map geometry with different degrees of ground marking incompleteness.

Incompleteness of road markings	Boundary+Divider				Crosswalks			All classes
	IoU	CD_P	CD_L	*CD*	*IoU*	*P*	*R*	*IoU*
Mild (< 20%)	87.70	0.34	0.36	0.36	84.96	92.23	89.01	86.79
Moderate (~ 50%)	84.75	0.38	0.45	0.41	79.38	89.61	86.12	82.96
Severe (> 90%)	82.65	0.40	0.46	0.43	75.62	85.98	78.57	80.31

an acceptable performance even under the most severe occlusion, which still outperforms the baselines in Table 3 that are without severe occlusion. Different from online map construction methods that rely solely on sensor data (point clouds), VI-Map additionally leverages the precise trajectory observations of infrastructure and extracts unique features valuable to map construction. Such a design makes VI-Map resilient to various ground marking conditions as it is not affecting trajectory features.

6 RELATED WORK

Online HD map construction methods. HDMapNet [27] is the first work that introduces the problem of HD semantic map learning. It encodes features from a single frame of LiDAR point cloud and/or images from surrounding cameras, and predicts semantic map elements in the bird's-eye view. STSU [9] proposes an end-to-end method that extracts local road network graphs and detects objects simultaneously, given only a front-facing camera image. VectorMapNet [29] uses transformer modules to predict a sparse set of polylines in the bird's-eye view to model the geometry of HD maps. These works all take onboard sensor observations as inputs and thus are limited due to physical barriers such as obstacle occlusion and limited sensing range.

Infrastructure-assisted vehicle perception. Recent works have shown the potential of infrastructure to enhance perception in autonomous driving. VI-Eye [21] proposes a semantic-based point cloud registration method for merging vehicle point cloud with infrastructure point cloud. VIPS [44] fuses the object detection results of the infrastructure and vehicle by leveraging a graph matching algorithm. Michael *et al.* [7] leverage the infrastructure to detect, track and predict the motion of vehicles to build an environment model, which is transmitted to the vehicle for motion planning. Among these works, infrastructures are primarily utilized for dynamic object perception. VI-Map stands apart from these endeavors, as its objective is to harness infrastructure to construct precise and current HD maps on the infrastructure side, thereby assisting the generation of on-vehicle HD maps.

7 DISCUSSION

Scalability of VI-Map. VI-Map's adaptability extends to a diverse range of road types and traffic scenarios. First, the BEV features used for inferring HD maps are low-level features derived directly from raw data, without any assumptions made regarding road structures or lane numbers. Second, heavy or light traffic condition primarily affects the accumulation time of static point clouds and vehicle trajectories. However, it does not harm the accuracy of the constructed HD map.However, the scalability of VI-Map is limited for inferring complex-shaped 3D map elements such as traffic lights, fire hydrants, or bus stops. This limitation is primarily attributed to the projection of 3D data onto the 2D BEV space and the subsequent processing within that domain, which results in the loss of information during the dimensionality reduction process. Consequently, this approach may not yield optimal performance when dealing with intricate 3D map elements, as the finer details of their 3D geometry are lost during the projection.

Limitations and failure cases. VI-Map can achieve unsatisfactory performance in adverse weather conditions, such as rain, snow, and fog, due to the significant noises of data from LiDARs. Furthermore, VI-Map may also encounter challenges in high-speed driving scenarios, as the increased velocity leads to a significant drop in the quality of the point cloud, thereby affecting the performance of VI-Map. Nevertheless, the proposed Geometry Construction, Topology Estimation, and Map Fusion can be modified and applied to other sensing modalities like 3D cameras.

8 CONCLUSION

In conclusion, we present VI-Map, the first system that utilizes the unique advantages of roadside infrastructure to enhance on-vehicle HD maps by providing accurate and timely infrastructure HD maps. We have implemented VI-Map end-to-end and the experimental results show that VI-Map enhances existing HD mapping methods in terms of map geometry accuracy, map topology freshness, system robustness, and efficiency.

ACKNOWLEDGEMENT

This work is supported in part by Research Grants Council (RGC) of Hong Kong under General Research Fund #14222222, #14211121, and #14214022, and National Natural Science Foundation of China under Grant No. 62202407.

APPENDIX

The research artifact accompanying this paper is available via https://github.com/yuzehh/VI-Map.

REFERENCES

[1] Autoware. 2023. The Autoware Foundation - open source for autonomous driving. https://www.autoware.org/. (2023).

[2] Baidu. 2023. Apollo. https://apollo.auto/. (2023).

[3] Zhibin Bao, Sabir Hossain, Haoxiang Lang, and Xianke Lin. 2022. High-definition map generation technologies for autonomous driving: a review. *arXiv preprint arXiv:2206.05400* (2022).

[4] Andreas Birk and Stefano Carpin. 2006. Merging occupancy grid maps from multiple robots. *Proc. IEEE* 94, 7 (2006), 1384–1397.

[5] Jose-Luis Blanco, Javier González-Jiménez, and Juan-Antonio Fernández-Madrigal. 2013. A robust, multi-hypothesis approach to matching occupancy grid maps. *Robotica* 31, 5 (2013), 687–701.

[6] The Lidar News Blog. 2022. HD Map Database Coverage Doubled. https://blog.lidarnews.com/hd-map-database-coverage-doubled/. (2022).

[7] Michael Buchholz, Johannes Müller, Martin Herrmann, Jan Strohbeck, Benjamin Völz, Matthias Maier, Jonas Paczia, Oliver Stein, Hubert Rehborn, and Rüdiger-Walter Henn. 2021. Handling occlusions in automated driving using a multiaccess edge computing server-based environment model from infrastructure sensors. *IEEE Intelligent Transportation Systems Magazine* 14, 3 (2021), 106–120.

[8] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.

[9] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. 2021. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15661–15670.

[10] Stefano Carpin. 2008. Fast and accurate map merging for multi-robot systems. *Autonomous robots* 25 (2008), 305–316.

[11] Xue-Mei Chen, Min Jin, Yi-song Miao, and Qiang Zhang. 2017. Driving decision-making analysis of car-following for autonomous vehicle under complex urban environment. *Journal of central south university* 24 (2017), 1476–1482.

[12] Cohda. 2023. Cohda Wireless MK6C EVK RSU. https://www.cohdawireless.com/solutions/hardware/mk6c-evk/. (2023).

[13] Commsignia. 2023. Commsignia Roadside Unit. https://www.commsignia.com/products/. (2023).

[14] CVAT. 2023. CVAT Annotation tool. https://www.cvat.ai/. (2023).

[15] Bert De Brabandere, Davy Neven, and Luc Van Gool. 2017. Semantic instance segmentation for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 7–9.

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[17] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. 2021. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9710–9719.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3354–3361.

[19] Nikhil Gosala, Kürsat Petek, Paulo L. J. Drews-Jr, Wolfram Burgard, and Abhinav Valada. 2023. SkyEye: Self-Supervised Bird's-Eye-View Semantic Mapping Using Monocular Frontal View Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14901–14910.

[20] HARMAN. 2023. HARMAN Savari StreetWAVE RSU. https://car.harman.com/solutions/connectivity/harman-savari-streetwave. (2023).

[21] Yuze He, Li Ma, Zhehao Jiang, Yi Tang, and Guoliang Xing. 2021. VI-eye: semantic-based 3D point cloud registration for infrastructure-assisted autonomous driving. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 573–586.

[22] Namdar Homayounfar, Wei-Chiu Ma, Justin Liang, Xinyu Wu, Jack Fan, and Raquel Urtasun. 2019. Dagmapper: Learning to map by discovering lane topology. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2911–2920.

[23] Zhiyu Huang, Jingda Wu, and Chen Lv. 2021. Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 10239–10251.

[24] Kitae Kim, Soohyun Cho, and Woojin Chung. 2021. Hd map update for autonomous driving with crowdsourced data. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1895–1901.

[25] Annkathrin Krämmer, Christoph Schöller, Dhiraj Gulati, Venkatnarayanan Lakshminarasimhan, Franz Kurz, Dominik Rosenbaum, Claus Lenz, and Alois Knoll. 2019. Providentia–A Large-Scale Sensor System for the Assistance of Autonomous Vehicles and Its Evaluation. *arXiv preprint arXiv:1906.06789* (2019).

[26] Heon-Cheol Lee, Seung-Hwan Lee, Myoung Hwan Choi, and Beom-Hee Lee. 2012. Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses. *Robotica* 30, 2 (2012), 205–220.

[27] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. 2022. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 4628–4634.

[28] Rong Liu, Jinling Wang, and Bingqi Zhang. 2020. High definition map for automated driving: Overview and analysis. *The Journal of Navigation* 73, 2 (2020), 324–341.

[29] Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. 2022. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920* (2022).

[30] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. 2022. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. *arXiv preprint arXiv:2205.13542* (2022).

[31] Livox. 2023. Livox AVIA. https://www.livoxtech.com/avia. (2023).

[32] Livox. 2023. Livox HAP. https://www.livoxtech.com/hap. (2023).

[33] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. 2021. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision* 129 (2021), 548–578.

[34] Xin Ma, Rui Guo, Yibin Li, and Weidong Chen. 2008. Adaptive genetic algorithm for occupancy grid maps merging. In *2008 7th World Congress on Intelligent Control and Automation*. IEEE, 5716–5720.

[35] Maximilian Naumann, Liting Sun, Wei Zhan, and Masayoshi Tomizuka. 2020. Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior based on Inverse Reinforcement Learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 5481–5487. https://doi.org/10.1109/ICRA40945.2020.9196795

[36] ASAM OpenDRIVE. 2023. OpenDRIVE Format Specification. https://www.asam.net/standards/detail/opendrive/. (2023).

[37] Teddy Ort, Liam Paull, and Daniela Rus. 2018. Autonomous vehicle navigation in rural environments without detailed prior maps. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2040–2047.

[38] David Pannen, Martin Liebner, Wolfgang Hempel, and Wolfram Burgard. 2020. How to keep HD maps for automated driving up to date. In

2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2288–2294.

[39] Vasyl Pihur, Susmita Datta, and Somnath Datta. 2007. Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach. *Bioinformatics* 23, 13 (2007), 1607–1615.

[40] Nicholas G Polson and Vadim O Sokolov. 2017. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies* 79 (2017), 1–17.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[42] Heiko G Seif and Xiaolong Hu. 2016. Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry. *Engineering* 2, 2 (2016), 159–162.

[43] Tixiao Shan and Brendan Englot. 2018. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4758–4765.

[44] Shuyao Shi, Jiahe Cui, Zhehao Jiang, Zhenyu Yan, Guoliang Xing, Jianwei Niu, and Zhenchao Ouyang. 2022. VIPS: real-time perception fusion for infrastructure-assisted autonomous driving. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 133–146.

[45] Brian L Smith and Michael J Demetsky. 1994. Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches. In *Proceedings of IEEE international conference on systems, man and cybernetics*, Vol. 2. IEEE, 1706–1709.

[46] HERE Technologies. 2018. HERE HD LiveMap, The Most Intelligent Sensor for Autonomous Driving. https://bit.ly/2Woss4K. (2018).

[47] TomTom. 2018. HD Maps - Highly Accurate Border-to-border Model of the Road. https://bit.ly/2WrI1sd. (2018).

[48] Manabu Tsukada, Takaharu Oi, Masahiro Kitazawa, and Hiroshi Esaki. 2020. Networked roadside perception units for autonomous driving. *Sensors* 20, 18 (2020), 5320.

[49] U-BLOX. 2023. NEO-M8T GPS. https://www.u-blox.com/en/product/neolea-m8t-series. (2023).

[50] Jingke Wang, Yue Wang, Dongkun Zhang, Yezhou Yang, and Rong Xiong. 2020. Learning hierarchical behavior and motion planning for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2235–2242.

[51] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 2020. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv preprint arXiv:2008.08063* (2020).

[52] WIT-MOTION. 2023. HWT9052-485 IMU. https://www.wit-motion.cn/#/witmotion/product/detail?id=0e24e2a59ac94b14b3034a92e4338b2c. (2023).

[53] Ji Zhang and Sanjiv Singh. 2014. LOAM: Lidar odometry and mapping in real-time.. In *Robotics: Science and Systems*, Vol. 2. Berkeley, CA, 1–9.

[54] Xun S Zhou and Stergios I Roumeliotis. 2006. Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 1785–1792.