A Review on End-to-End High-Definition Map Generation

Jiyong Kwag, Charles Toth

The Ohio State University, Department of Civil, Environmental, and Geodetic Engineering, Columbus, OH, USA
<kwag.3><toth.2>@osu.edu

Keywords: High-Definition Map Generation, Autonomous Driving, Sensor Fusion

Abstract

Autonomous driving offers benefits such as congestion mitigation, increased productivity through the reallocation of driving time, and decreased energy waste. However, achieving Level 4 and 5 autonomous driving remains a significant challenge for both academia and industry. Among the various modules of autonomous driving, High-Definition (HD) maps have become a crucial component due to their high precision in map elements, enabling accurate localization, scene interpretation, navigation, vehicle control and motion forecasting of trajectory of surrounding objects. Several map providers, including TomTom, HERE, Waymo, and NVIDIA, create HD maps for their specific purposes. However, most HD map datasets are not publicly available for individual researchers and companies to investigate the current trends in HD map generation. Furthermore, recent survey papers on HD map generation have tended to focus only on specific aspects, such as road topology or boundary extraction, rather than considering the overall end-to-end HD map generation process. Therefore, we begin with a brief definition, standards, and functionality of HD maps, followed by an exploration of different types of HD maps, including offline and online variants, highlighting their respective advantages and disadvantages. Finally, we will discuss the most recent end-to-end HD map generation architectures, along with various types of open-source HD map datasets and compare their performances.

1. Introduction

1.1 High-Definition Map

In recent years, traditional digital maps have primarily provided two-dimensional information at the basic road level, lacking the necessary accuracy and abundance of environmental attributes essential for autonomous driving, such as road conditions, lanes, and traffic signaling (LETE, 2023). This deficiency is addressed through High-Definition (HD) maps, which involve collecting data from a diverse set of sensors, including lidar, radar, mono/stereo cameras, GNSS, and IMU (Shan, 2020), (Yang, 2018), (Yu, 2015).

HD maps refer to digital maps that contain all critical static properties of the road and surrounding environment essential for autonomous driving. These maps are typically accurate at the centimeter level and provide abundant geometric and semantic information about the road environment (Bao, 2023). The use of HD maps plays a crucial role in improving the precision of vehicle localization and enhancing the robustness and safety of both the vehicle and the surrounding environment. By interacting with different types of sensor modalities, such as cameras, lidar, and radar, an accurate HD map aids in building the perception module of autonomous driving (Bao, 2023), (Liu, 2019). As a result, autonomous driving systems can process downstream tasks, such as motion planning for both long and short-term travel and motion forecasting of surrounding elements (Liu, 2023), such as other vehicles, pedestrians, and cyclists, thereby optimizing vehicle control for safety and driving efficiency. Figure 1 illustrates the autonomous driving pipeline and its individual modules.

Although HD maps are widely used in many companies to deploy safe autonomous driving, each map provider has its own typical standard of HD map, especially concerning the definition of the layers of HD maps. TomTom, HERE, and Bertha Drive are among the most widely known HD map providers (Marchant, 2019), (Joergensen, 2024), (Ziegler, 2014). The most common type of layer division in HD maps is divided into three layers: the road layer, the lane layer, and the feature layer. Table 1 presents different types of map layer divisions.

First, the road layer consists of road topology, direction, and rules that define high-level and basic road characteristics. It is usually designed as a graph and primarily serves navigation purposes. Second, the lane layer consists of lane dividers, centerlines, and stop lines, and it includes more specific road pavement marking elements. Lane layer elements are designed in vectorized map format and communicate with vehicle sensors to build the perception module (Han, 2023), (Liang, 2020), (Zhang, 2022) for environmental modeling, motion planning (Hu, 2021), (Ngiam, 2022), and forecasting (Hu, 2021), (Kamenev, 2022), (Liu, 2021) for vehicle control tasks. Lastly, the feature layer consists of roadside furniture such as traffic signs, signals, and buildings. Feature layers aid localization, especially in urban areas, by detecting benchmarks and roadside elements. Although most map providers define HD maps with three distinct layers, most open-source HD map datasets only consist of road and lane layers. Open-source datasets do not consider the feature layer as a task for HD maps but rather for the perception module.

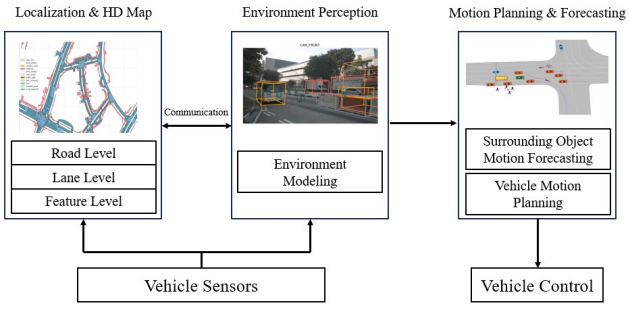

Figure 1. Autonomous driving module pipeline including localization, environment perception modeling, motion planning and forecasting, and vehicle control. Vehicle sensors include a camera, and stereo camera. lidar, radar, GPS/IMU, and more.

Layer	Bertha Drive	TomTom	HERE
1	Open Street Map	Navigation Data	Road Layer
2	Lane Level Map	Planning data	Lane Layer
3	Landmark Map	Road DNA	Localization layer

Table 1. Examples of HD map three-layer division among different HD map providers.

1.2 Offline HD Map Production

Despite its disadvantages, there remains a significant need for offline HD maps due to their high accuracy resulting from professional map production, including human QA/QC. In the domain of autonomous driving, offline HD maps serve as more than just navigation tools; they act as a type of geospatial data, past structured sensor data that assists autonomous driving beyond the usual sensor range, overcoming occlusion problems and harsh environments. Autonomous driving requires high redundancies in each task to minimize the probability of catastrophic accidents, as road accidents not only involve property damages but also human lives. Therefore, human-labeled offline HD maps can provide both redundancy and accuracy in autonomous driving task modules for safety purposes.
n provide both redundancy and accuracy in autonomous driving task modules for safety purposes.

However, offline HD map generation presents challenges. While sensor technologies and data availability have significantly improved, generating offline HD maps requires considerable human effort for annotating and maintaining semantic information, leading to scalability issues. Moreover, many HD map providers increasingly rely on crowdsourcing, a task impractical for individual researchers due to the sheer volume of data and limited access to company data. Lastly, keeping HD maps continuously updated is costly.

1.3 Online HD Map Creation

Online HD mapping refers to the generation of HD maps while an autonomous vehicle is operating on the road. With advancements in sensor technologies and neural networks for object detection in images, online HD map generation has emerged as a preferred solution to reduce human effort in labeling map elements' locations and semantic information. Most online HD map generation models leverage transformer-based methodologies to extract a bird’s eye view (BEV) feature map from the vehicle’s 360-degree surrounding cameras and detect lane locations, types, and directions, road boundaries, and pedestrian crossings. Consequently, generating HD maps on-the-fly is not limited to existing maps for autonomous vehicle localization, potentially alleviating meter-level errors in special circumstances.
rrors in special circumstances.

However, online HD mapping also has its drawbacks. Although the development of neural networks shows promise in generating HD maps while driving, the accuracy of these maps remains questionable. The mean Average Precision (mAP) of state-of-the-art online HD map generation models is shown in Tables 3 and 4. Moreover, online HD map generation models based on open-source HD map datasets have not been evaluated over long-range traveling, as most open-source datasets only consider approximately 20-second intervals for each driving scenario due to dataset size limitations.

2. Open-Source HD Map Datasets

The availability of accurate and precise ground truth datasets plays a crucial role in training neural networks to achieve high performance. In the past few years, there have been limited open-source datasets providing human-labeled HD maps for individual researchers to investigate (Geiger, 2012). However, nowadays, various universities and companies have started releasing open-source datasets that include highly accurate HD maps with human labeling, along with corresponding sensor data such as camera images, lidar, radar, and GPS location data (Wilson, 2023), (Chang, 2019), (Sun, 2020), (Caesar, 2020). This development enables both academic and industry researchers to delve into HD map generation models. In Section 2, we will discuss the most recent and commonly used autonomous driving datasets, covering the following aspects: list of sensors and data collection methods, data annotation and scene selection, and HD map formation.

2.1 nuScenes

2.1.1 Sensors and Data Collections

Following in the spirit of pioneering autonomous driving datasets like KITTI (Geiger, 2012), nuTonomy released their first autonomous driving dataset called nuTonomy Scenes, or nuScenes for short. nuScenes collected their dataset primarily in two cities: Boston (Seaport and South Boston) and Singapore (One North, Holland Village, and Queenstown), known for dense traffic and challenging driving conditions (Caesar, 2020). Their dataset encompasses urban, residential, natural, and industrial areas of the cities, including surrounding vegetation, buildings, vehicles, road markings, and traffic directions (both right and left). To represent diverse driving circumstances, data collection was performed during various times of the day, under different weather conditions such as sunny, rainy, and cloudy days. This makes nuScenes the first dataset to include such a wide range of driving situations.

nuScenes employs a variety of sensor modalities, including 5 radar, 1 lidar, 6 RGB cameras, and 1 GPS/IMU. The 6 RGB cameras are strategically positioned around the vehicle, covering the front center, front right, front left, rear right, rear left, and back, providing a 360-degree field of view. The lidar sensor is mounted on the top of the vehicle, while the 5 radars are distributed at the front, front right, front left, rear right, and rear left. nuScenes claims to be the first and only dataset to include radar data, which offers more ranging data compared to lidar but with higher sparsity in data points. The specific setup locations and axes of each sensor are illustrated in Figure 2.

2.1.2 Data Annotation and Scene Selection

The nuScenes dataset consists of 1000 scenarios, each with a duration of 20 seconds, and is fully annotated with 3D bounding boxes spanning 23 classes, including rare objects. Notably, their dataset includes 7 times more 3D bounding boxes than KITTI, and these bounding boxes are annotated across the entire 360-degree field of view. In contrast, KITTI only annotates objects visible in the frontal view. The annotations in nuScenes are conducted by expert annotators and undergo multiple validations. Each 3D bounding box is marked with the object's x, y, z coordinates; width, length, height; and yaw angle.

The 1000 scenes in the dataset provide approximately 15 hours of driving data within the city, covering a total distance of 242 km at an average speed of 16 km/hr. These scenes encapsulate a wide array of interesting driving scenarios, including high-traffic situations at intersections and construction sites, rare

class objects such as ambulances, potentially dangerous situations on the road like jaywalkers, unusual driving, and pedestrian behaviors, as well as typical driving scenarios such as lane changes, turns, and stops.

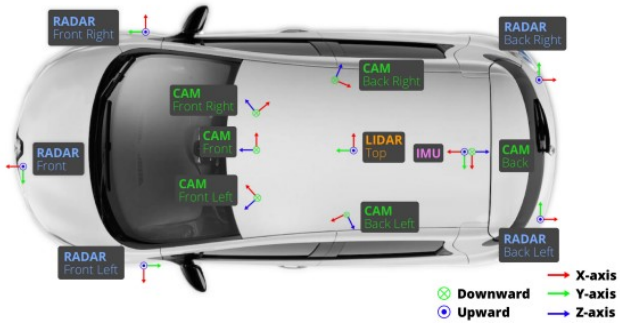

Figure 2. Vehicle sensor setup for nuScenes data collection.

2.1.3 nuScenes HD Map

In addition to sensor data, nuScenes also provides highly accurate and human-labeled semantic HD maps corresponding to each sensor data scenario. The nuScenes HD map is structured around three base primitives: nodes, lines, and polygons. A node represents a basic element with spatial information in terms of x and y coordinates, based on the WGS 84 coordinate system, which corresponds with Google Maps. A line consists of two or more nodes, while a polygon is formed by three or more nodes to shape the external outline of the polygon. Additionally, polygons can include holes, which are also formed by three or more nodes.

The nuScenes HD map is mainly divided into 11 layers: drivable area, road segment, roadblock and divider, lane and divider, pedestrian crossing, sidewalk, stop line, car park, and traffic light. Detailed nuScenes HD map layers are illustrated in Figure 3. The nuScenes HD map includes essential road rules such as intersection points, traffic directions, and traffic light information for drivers. In comparison with the basic three-layer division from HERE, the nuScenes HD map primarily has two layers: road and lane layers. However, nuScenes further subdivides the road and lane layers into smaller segments such as road segments, roadblocks, and lanes.

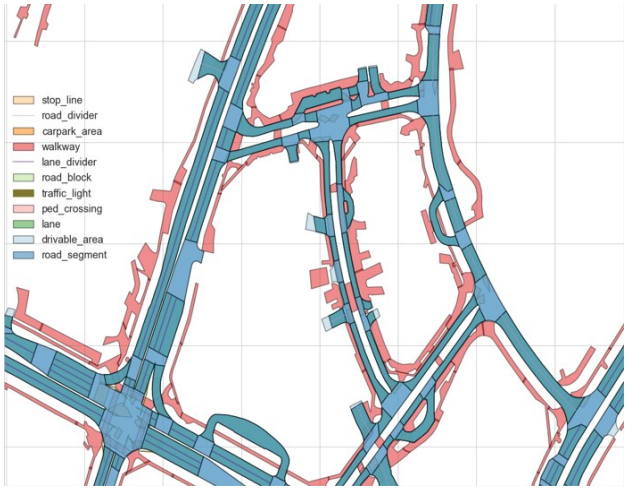

Figure 3. Example of nuScenes 11 semantic layers of local area of Singapore.

It's important to note that unlike the typical understanding of traditional HD maps, nuScenes HD maps do not provide any 3D spatial information regarding surrounding roadside objects or ground height information. Instead, temporary roadside objects such as road barriers and traffic cones are included in the 3D bounding box annotations rather than being marked in the HD map layers.

2.2 Argoverse 2

2.2.1 Sensors and Dat Collection

The Argoverse 2 dataset is one of the most recent autonomous driving datasets and is provided by Argo AI in collaboration with various universities including Georgia Tech, UBC, MIT, and CMU (Wilson, 2023). Argoverse 2 serves as the successor to their previous dataset, Argoverse 1. In comparison to Argoverse 1, Argoverse 2 offers more scenes and is 23 times richer in terms of the number of 3D bounding boxes. The dataset was primarily collected across six different cities in the United States: Austin, Detroit, Miami, Pittsburgh, Washington DC, and Palo Alto, aiming to reflect diverse driving styles and urban structures. Similar to the nuScenes dataset, Argoverse 2 followed similar styles of data collection, including data collection from different seasons such as snowy, sunny, and rainy days, as well as different lighting conditions including night and day driving scenarios.

The sensor modalities in Argoverse 2 include 2 lidars, 7 cameras, 2 stereo cameras, and GPS/IMU. The 7 high-resolution (2048 width x 1550 height) cameras cover a 360-degree field of view, including front, front left and right, side left and right, and rear left and right perspectives. Additionally, 2 stereo cameras (2048 width x 1550 height) are installed at the front left and right positions. The 2 lidar sensors and GPS/IMU are mounted on the top of the vehicle. Argoverse 2 shares a similar sensor configuration with the nuScenes dataset, except for the fact that nuScenes includes 5 radar sensors while Argoverse 2 has 2 stereo cameras instead.

2.2.2 Data Annotation and Scene Selection

Initially, Argoverse 1 contained only 113 scenes, covering a mere 0.6 hours of data collection, making it a smaller dataset compared to other similar datasets released around the same time, such as nuScenes, Waymo Open, and Lyft L5. In response to this, Argoverse 2 was released with 1000 scenes, each lasting 15 seconds, and included annotations for 23 classes. The dataset contains approximately 23 million 3D bounding boxes, surpassing the number in nuScenes. Notably, the 1000 scenes in Argoverse 2 include distinct driving behaviors observed in crowded and dynamic scenarios, such as cut-ins and encounters with jaywalkers. The number of scenes intentionally matches that of Waymo Open and nuScenes.

In their paper, the creators of Argoverse 2 aimed to challenge the notion that bigger datasets are always better. They believed that releasing a benchmark dataset that is excessively large might deter the academic community from engaging with the work, leaving progress solely to well-resourced companies. Moreover, the Argoverse 2 dataset includes annotations for some under-researched objects, such as animals, traffic light signalers, and railed vehicles.

2.2.3 Argoverse HD Map

Argoverse 2 also provides an HD map along with corresponding sensor data. However, unlike nuScenes, which offers a full city-level HD map, Argoverse 2 subdivides the city into local maps for each individual scene. This approach eliminates the need for developers to render or crop the HD map. The data in Argoverse 2 is composed of x, y, z coordinates for their basic primitive node in 3D, utilizing a city-level coordinate system (Chang,

2019). The city-level coordinate system involves generating a local tangent plane from the Earth’s surface centered at a central point within the city of interest. Consequently, city-level coordinates can be easily converted to UTM by adding the origin of the city-level coordinate in UTM to the object’s city-level xyz position, according to user preferences. In contrast to nuScenes HD map, which uses the WGS 84 coordinate system, Argoverse 2 prefers its own city-level coordinate system. This decision is motivated by the fact that leveraging WGS 84 at a city scale causes changes to the xyz coordinates in hundredths decimal place which decreases the interpretability of the location data.

Based on the xyz coordinate primitives, Argoverse 2 provides four distinct layers of HD map: drivable area, lane segment, pedestrian crossing, and ground height. Drivable area illustrates areas where vehicles can drive without damage. Lane segment provides road lane information, such as road boundaries, lane types, intersection information. However, unlike nuScenes, the lane segment layer in Argoverse 2 does not include traffic direction information. Pedestrian crossing identifies areas designated for pedestrians to cross roads safely. Lastly, ground height is a rasterized map with a 30-centimeter resolution to provide high-resolution ground height information. However, Argoverse 2 does not specify whether they used orthometric height or geoid height, nor do they disclose the types of geoid models they employed. Similar to nuScenes, the Argoverse 2 HD map dataset focuses on road and lane layers without including feature layer information, consistent with the basic layer-division principle outlined in Table 1.

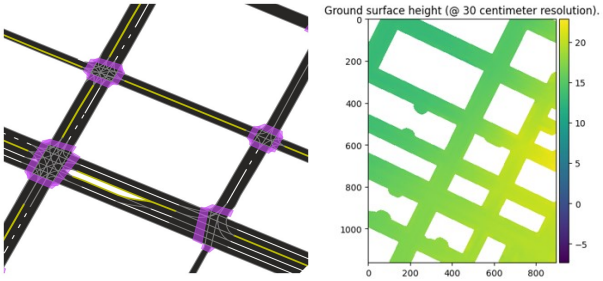

Figure 4. (Left) Example of Argoverse 2 3 semantic layers of local area of Pittsburgh. (Right) Example of Argoverse 2 ground height map of local area of Pittsburgh.

2.3 Comparison among HD Map Datasets

After the release of the KITTI dataset, many autonomous driving datasets began to incorporate 3D box annotation information derived from lidar point clouds. Moreover, the number of classes, scenes, and cuboids increased to reflect advancements in autonomous driving research. However, until recently, there was a notable absence of datasets that included HD maps, which limited research on online HD map generation models. Argoverse 1 was a pioneer in providing an autonomous driving dataset that included HD maps of the collected city data. However, the size of these datasets was limited and not sufficient for training HD map generation models. Following in the footsteps of the KITTI dataset, today's major open-source autonomous driving datasets provide HD maps of the areas where they collected sensor data include Argoverse 2, nuScenes, and Waymo Open.

Table 2 illustrates a comparison among different autonomous driving datasets. Argoverse 2 stands out with its impressive numbers of cuboids and the number of cities included, reflecting the varied road, and driving characteristics of diverse urban environments in the United States. Both nuScenes and Argoverse 2 boast the greatest number of classes, ensuring a comprehensive representation of distinct object types such as ambulances to aid motion planning and control task of the autonomous driving. Lastly, Argoverse 2 and Waymo provide 3D HD maps, which represent an area of research that has not been extensively explored, particularly in the realm of online HD map generation.

	Argoverse2	nuScenes	Waymo	KITTI
Year	**2021**	2019	2019	2012
Cities	**6**	2	3	1
Cameras	**7**	6	5	4
Lidars	2	**5**	1	1
Classes	**23**	**23**	4	8
Cuboids	**~23M**	~1.4M	12M	200K
Scenes	**1K**	**1K**	**1K**	22
Night/Rain	**Yes/Yes**	**Yes/Yes**	**Yes/Yes**	No/No
Map/3DMap	**Yes/Yes**	Yes/No	**Yes/Yes**	No/No

Table 2. Comparison of popular autonomous driving datasets. It is labelled with respect to year that dataset was released. Only Argoverse 2, nuScenes, and Waymo dataset include vector map.

3. End-to-End HD Map Generation Architecture

Following the semantic layers of open-source HD map datasets, HD map is typically constructed in BEV (Xiong, 2023), (Liu, 2023), (Huang, 2022). However, these models are based on rasterized map that has limitation on using autonomous driving downstream tasks, such as motion forecasting and planning. Specifically, these models place emphasis on predicting lane dividers, boundaries, and pedestrian crossings as map elements when evaluating performance in map element prediction tasks. Given that map elements typically consist of simple lines or polygons in 2D, each map element is represented as a set of points or vectors. Furthermore, while predicting the location of map elements is essential, accurately outlining the set of points of the map element is also crucial for achieving accurate map construction. Thus, accurately shaping the outline of map elements is another important task in the HD map construction process.

3.1 HDMapNet

HDMapNet is a pioneering online vectorized HD map generation model that utilizes open-source HD map datasets and corresponding sensor data to generate HD maps in real-time (Li, 2022). Its architecture consists of a BEV feature extractor encoder, based on camera and/or lidar inputs, and a BEV decoder that produces semantic segmentation, instance embedding, and direction prediction maps. Post-processing is then applied to vectorize pixels into a vector map.

The BEV feature encoder comprises two main components: an image encoder and a point cloud encoder. The image encoder utilizes surrounding images captured by cameras, while the point cloud encoder leverages lidar data to extract BEV features. The image encoder consists of a perspective view image encoder, employing an EfficientNet-B0 (EB0) model (Tan, 2020), and a neural view transformer, which converts perspective view feature maps into BEV features in the camera coordinate system via a multi-layer perceptron (MLP). Alternatively, the point cloud encoder, based on a methodology similar to PointNet (Charles, 2017), utilizes dynamic voxelization (Zhou, 2019) to divide the 3D space of the lidar point cloud into multiple pillars, which are then converted into BEV feature maps.

Following the generation of the image and/or BEV feature maps, the BEV decoder employs a ResNet (He, 2016) with three blocks to generate the BEV vector map. This process involves three main branches: semantic prediction, instance embedding, and direction prediction. Semantic prediction employs a fully convolutional network (Long, 2015) and cross-entropy loss to predict the semantic meaning of each part of the embedding, such as lane types, boundaries, and pedestrian crossings. Instance embedding involves clustering elements of the embedding based on DBSCAN to identify necessary line elements of the map in pixels, which are then connected greedily based on direction prediction to construct vector map elements.

Although HDMapNet lays the groundwork for online vectorized HD map generation, it still relies on pixelwise prediction and necessitates post-processing to vectorize pixels for map generation. Additionally, its mean average precision for overall and individual lane dividers, boundaries, and pedestrian crossings requires improvement before it can be used effectively in autonomous driving tasks. Nonetheless, HDMapNet demonstrates the potential of using single or multi-modal inputs to generate HD maps on the fly.

3.2 VectorMapNet

Even though HDMapNet introduced an online vectorized map construction architecture using single and/or multi-modalities, it suffers from the need for post-processing from pixel-generated maps to vector maps. To address this disadvantage, VectorMapNet was introduced as the first end-to-end online HD map generation model without the need for post-processing, generating vector maps directly (Liu, 2023). VectorMapNet follows a similar scheme to HDMapNet, utilizing BEV feature maps to predict map elements, but it enhances mean average precision (mAP) and eliminates post-processing through its unique architecture.

The VectorMapNet architecture is divided into three parts: a BEV feature extractor using images and/or point clouds, a map element detector, and a polyline generator. The BEV feature extractor using images employs ResNet50 (R50) (He, 2016), particularly a shared CNN backbone to obtain perspective view feature maps, which are then converted using Inverse Perspective Mapping (IPM) (Mallot, 1991) to BEV feature maps. The BEV feature extractor using point clouds utilizes PointPillar (Lang, 2019) with dynamic voxelization (Zhou, 2019), similar to HDMapNet, to extract BEV features from the point cloud. BEV features from both images and point clouds are concatenated if both are present.

The map element decoder leverages a variant of DETR (Carion, 2020), which uses object queries to predict an object's location. However, in VectorMapNet, instead of object queries, it employs BEV feature maps and element queries, where each query represents a map element similar to object queries. The objective of the element query is to predict not only the position of the map element but also the position of the constituent x and y points of the predicted map element, referred to as keypoint embedding. The element query also predicts the class label of the predicted map element. The map element decoder adopts a multi-head self/cross-attention module as the attention module of the transformer (Vaswani, 2023). In the prediction head, an MLP decodes the predicted class label and location of the map elements. However, the predicted keypoint does not represent the vertices of the map elements, necessitating further work to construct the vector map element.

To construct vector map elements from keypoints and class labels, the polyline generator adopts a basic transformer architecture (Vaswani, 2023), which has shown superior performance in conditional sequence generation tasks in natural language processing (NLP). The transformer's conditional sequence generation methodology is utilized to capture the complex shape of the map element from keypoints, class labels, and BEV feature maps based on an autoregressive network. The autoregressive network enables each vertex to be predicted based on the previously predicted vertex.

Building on the preceding work of HDMapNet, VectorMapNet introduced the first end-to-end online vectorized HD map construction architecture without the need for post-processing of pixel images. Moreover, VectorMapNet increased mAP by approximately two times compared to HDMapNet. However, the autoregressive network in the polyline generator suffers from cumulative errors and requires significant inference time due to its nature of predicting the current vertex based on the previous vertex. Additionally, the feature introduced by HDMapNet, direction prediction, is not involved in VectorMapNet, indicating further need for improvement in online HD map construction research.

3.3 MapTR

Following the release of VectorMapNet, the utilization of DETR methodology became a key architecture for end-to-end online vector map construction. However, VectorMapNet still lacked a key feature of vector maps, namely direction prediction, and suffered from large inference times. To address this drawback, MapTR introduces perturbation-equivalent modeling. This approach not only predicts direction but also constructs the vector map without the need for a polyline generator, leveraging an autoregressive model to decrease the inference time of the model (Liao, 2023). Moreover, the MapTR architecture only uses image input but still achieves higher mean average precision (mAP) than previous works.

MapTR architecture utilizes images as the sole sensor modality input for HD map construction. It is divided into two parts: the map encoder and the map decoder. The map encoder primarily takes images and processes them into perspective feature maps using R50 (He, 2016). These perspective view feature maps are then converted to BEV features through a geometrically guided transformer (GKT) (Chen, 2022), which leverages geometric information to guide the transformation of perspective features to BEV features.

Similar to VectorMapNet, the map decoder in MapTR uses a DETR-based transformer. It takes BEV features and object queries to predict the class label and location of each point of the map element. In MapTR, the object query is referred to as an instance-level query, representing the map element. Specifically, a fixed number of instance-level queries are employed based on a hierarchical query scheme, where each instance-level query contains a fixed number of point-level queries that form the shape of the map element. Subsequently, permutation-equivalent modeling is employed to perform instance-level matching and point-level matching. This approach adopts a bipartite matching methodology similar to DETR, where predicted objects are matched with ground truth objects and unmatched objects are labeled as no object. Instance-level matching predicts the class level and approximate position of the map element, while point-level matching predicts the shape and direction of the map element at the point level. Through permutation-equivalent modeling, MapTR

achieves higher mAP and direction prediction while maintaining faster inference times without the need for an autoregressive network.

As the name suggests, MapTR shares many similarities with DETR more than VectorMapNet, utilizing a bipartite matching scheme to generate vector map elements without relying on an autoregressive network to reduce the inference time of map element construction. However, the use of a fixed number of point queries presents another drawback. Since the shape of map elements is dynamic and unique, such as round corners, this approach may introduce redundant points and loss of information about the detailed shape of the map elements, potentially leading to a lower mAP for the model.

3.4 PivotNet

MapTR introduced a novel approach by utilizing DETR's bipartite matching scheme to generate map elements, including direction prediction, while maintaining fast inference times. However, due to the fixed number of queries required by DETR, map elements contain redundant points or fail to shape map element correctly. To address this issue, PivotNet introduced the pivot dynamic matching module, which outlines the shape of the map element based on pivotal points (Ding, 2023). Furthermore, PivotNet maintains faster inference times compared to the autoregressive network used in VectorMapNet.

PivotNet is divided into four parts: the camera feature extractor, BEV feature decoder, line-aware point decoder, and pivot point head. The camera feature extractor takes surrounding images and extracts perspective view features using a shared neural network, with PivotNet utilizing SwinT (Liu, 2021) as the transformer-based feature extractor. The BEV feature decoder converts perspective view features into BEV features based on a transformer scheme, similar to BEVFormer (Li, 2022), a transformer encoder-based neural network that employs BEV queries to map perspective views into BEV view features.

After extracting the BEV feature map from the images, the line-aware point decoder employs an MLP and object query scheme. Object queries in PivotNet, referred to as point queries, represent the outline vertices of the map element. The line-aware point decoder utilizes a point-to-line mask module to effectively construct the map element. This module ensures that each point query of the same instance learns and shares a line-aware attention mask. The line-aware attention mask is then combined with cross-attention to encode subordinate and geometric priors into the point queries. The encoded point query, known as the point descriptor in PivotNet, is then utilized by the pivot point head, which employs pivot dynamic matching. This technique utilizes bipartite matching to identify pivotal points, distinguishing the direction of the point and ensuring that multiple points are not placed in the same location within the map element.

By incorporating pivot dynamic matching, PivotNet effectively reduces redundant points and mitigates the loss of points when constructing the map element from the predictions. Additionally, PivotNet maintains higher inference times than VectorMapNet by continuing to use the bipartite matching scheme to predict direction and map elements. However, compared to MapTR, which utilizes a hierarchical query scheme to encode information between map elements and belonging points, PivotNet employs a simpler point query approach. This may result in incomplete map element shapes or accuracy issues (Zhou, 2024).

4. Comparison among HD Map Generation Models

4.1 Comparison Metrics

Based on comparison metrics from image detection (Lin, 2014), the comparison metrics for HD map generation models is average precision (AP), which utilizes the Chamfer distance as the threshold to determine the true positives of the predicted map elements compared to the ground truth. The Chamfer distance measures the dissimilarity between two sets of points by comparing a reconstructed set of points with the ground truth set of points. The threshold values for the Chamfer distance typically used are {0.2, 0.5, 1.0} meters. However, Chamfer distance does not provide any directional difference measurement between two different sets of points.

To compare the specific details of three map elements (lane divider, boundary, and pedestrian crossing), the AP scores for each element are computed individually. The AP is then calculated for each element at different Chamfer distance thresholds. Finally, the mAP is calculated by taking the average of the three AP values. This provides an overall performance metrics for the model across all map elements.

$$D_{dir}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{|\mathcal{S}_1|} \sum_{x \in \mathcal{S}_1} \min_{y \in \mathcal{S}_2} \left|\left|x.y\right|\right|_2, \quad (1)$$

$$CD_{dir}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{2}\left(D_{dir}(\mathcal{S}_1, \mathcal{S}_2) + D_{dir}(\mathcal{S}_2, \mathcal{S}_1)\right), \quad (2)$$

where S_1, S_2 = two sets of points
x, y = coordinate points from set of points
min = minimum values in the set
$||.||_2$ = L2 norm

$$AP = \frac{1}{3} \sum_{\tau \in \{0.2, 0.5, 1.0\}} AP_\tau, \quad (3)$$

Where t = set of Chamfer distance threshold

4.2 Comparisons

Based on the average precision (AP) and mean average precision (mAP) comparison metrics with a Chamfer distance threshold of {0.2, 0.5, 1.0} meters, a comparison among different HD map construction models is performed using only image data as the sensor modality. Among the compared models, PivotNet achieves the highest mAP across both the nuScenes and Argoverse 2 datasets. However, it's important to note that using different types of backbone models may affect the results of the mAP, as PivotNet is the only model in the comparison that uses a transformer-based feature extractor. Despite HDMapNet and VectorMapNet showing lower mAP values, they still achieve high AP scores for boundary detection and lane divider, respectively. Additionally, even with a lower number of training epochs, MapTR demonstrates competitive results on road divider AP. An interesting observation is that the Argoverse 2 dataset yields competitive results with the nuScenes dataset, even with a lower number of training epochs. This suggests that the Argoverse 2 dataset may offer comparable good performance to the nuScenes dataset for HD map construction tasks.

	HDMapNet	VMapNet	MapTR	PivotNet	
BKB	EB0	R50	R50	SwinT	
Epoch	30	110	**24**	30	110
$AP_{divider}$	17.7	27.2	30.7	47.6	**53.6**
AP_{Ped}	13.6	18.2	23.2	38.3	**43.4**
$AP_{boundary}$	32.7	18.4	28.2	43.8	**50.5**
mAP	21.3	21.3	27.3	43.3	**49.2**

	HDMapNet	VMapNet	MapTR	PivotNet
BKB	EB0	R50	R50	SwinT
Epoch	**6**	24	**6**	10
$AP_{divider}$	19.5	33.32	42.2	**51.1**
AP_{Ped}	9.8	18.3	28.3	**36.1**
$AP_{boundary}$	35.9	20.4	33.7	**47.8**
mAP	21.8	24.0	34.8	**45.9**

Table 3, 4. Comparison of SOTA HD map construction models based on different measurements based on nuScenes (Above) Argoverse2 (Bottom) dataset.

5. Conclusion

In this paper, recent autonomous driving datasets with HD map data and end-to-end vectorized HD map construction models are analyzed. HDMapNet introduced the first online vectorized HD map but suffered from post-processing of pixel-wise results to generate the vector map. VectorMapNet addressed the post-processing issue by adopting DETR and an object query-based neural network but has limitations on inference time due to autoregressive modeling. MapTR further developed the DETR architecture using bipartite matching to solve the inference time of vector map construction; however, it has limited mAP. PivotNet resolved the mAP issue by predicting pivotal points of the map element to accurately predict the shape of the map element. Nonetheless, PivotNet misses the interaction between the element and points, still causing accuracy problems.

In conclusion, while recent advancements in autonomous driving datasets and vectorized HD map construction models have addressed various challenges, there remain significant hurdles to overcome. These challenges include ensuring the accuracy of map element predictions, preserving essential 3D information in the generated 2D vector maps, and effectively addressing occlusion issues. Future research efforts should focus on refining existing methodologies to tackle these challenges and pave the way for safer and more efficient autonomous driving systems.

Acknowledgements

This work was supported in part by the U.S. Department of Transportation under Grant 69A3552348327 for the CARMEN+ University Transportation Center.

References

Bao, Zhibin, et al. "A review of high-definition map creation methods for autonomous driving." Engineering Applications of Artificial Intelligence, vol. 122, June 2023, p. 106125, https://doi.org/10.1016/j.engappai.2023.106125.

Caesar, Holger, et al. "NuScenes: A Multimodal dataset for Autonomous Driving." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020, https://doi.org/10.1109/cvpr42600.2020.01164.

Carion, Nicolas, et al. "End-to-end object detection with Transformers." Computer Vision – ECCV 2020, 2020, pp. 213–229, https://doi.org/10.1007/978-3-030-58452-8_13.

Chang, Ming-Fang, et al. "Argoverse: 3D Tracking and forecasting with rich maps." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019, https://doi.org/10.1109/cvpr.2019.00895.

Charles, R. Qi, et al. "PointNet: Deep Learning on point sets for 3D classification and segmentation." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, https://doi.org/10.1109/cvpr.2017.16.

Chen, Shaoyu, et al. "Efficient and Robust 2D-to-BEV Representation Learning via Geometry-Guided Kernel Transformer." arXiv.Org, 9 June 2022, arxiv.org/abs/2206.04584.

Cheng, Bowen, et al. "Masked-attention mask transformer for Universal Image segmentation." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, https://doi.org/10.1109/cvpr52688.2022.00135.

Ding, Wenjie, et al. "PivotNet: Vectorized pivot learning for end-to-end HD map construction." 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 1 Oct. 2023, https://doi.org/10.1109/iccv51070.2023.00340.

Geiger, A., et al. "Are we ready for autonomous driving? The Kitti Vision Benchmark Suite." 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, https://doi.org/10.1109/cvpr.2012.6248074.

Han, Chunrui, et al. "Exploring Recurrent Long-Term Temporal Fusion for Multi-View 3D Perception." arXiv.Org, 13 Mar. 2023, arxiv.org/abs/2303.05970.

He, Kaiming, et al. "Deep residual learning for image recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, https://doi.org/10.1109/cvpr.2016.90.

Hu, Anthony, et al. "Fiery: Future instance prediction in bird's-eye view from surround monocular cameras." 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2021, https://doi.org/10.1109/iccv48922.2021.01499.

Hu, Peiyun, et al. "Safe local motion planning with self-supervised freespace forecasting." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, https://doi.org/10.1109/cvpr46437.2021.01254.

Huang, Junjie, et al. "Bevdet: High-Performance Multi-Camera 3D Object Detection in Bird-Eye-View." arXiv.Org, 16 June 2022, arxiv.org/abs/2112.11790.

Joergensen, René Munk, et al. "Map Data: Static MAP API: Platform." HERE, www.here.com/platform/map-data. Accessed 8 Apr. 2024.

Kamenev, Alexey, et al. "Predictionnet: Real-time joint probabilistic traffic prediction for planning, control, and simulation." 2022 International Conference on Robotics and Automation (ICRA), 23 May 2022, https://doi.org/10.1109/icra46639.2022.9812223.

Lang, Alex H., et al. “Pointpillars: Fast encoders for object detection from point clouds.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019, https://doi.org/10.1109/cvpr.2019.01298.

LEITE, JOÃO PEDRO. “A Brief History of GPS In-Car Navigation.” NDrive, 24 Feb. 2023, ndrive.com/brief-history-gps-car-navigation/.

Li, Qi, et al. “HDMapNet: An online HD map construction and evaluation framework.” 2022 International Conference on Robotics and Automation (ICRA), 23 May 2022, https://doi.org/10.1109/icra46639.2022.9812383.

Li, Zhiqi, et al. “Bevformer: Learning bird’s-eye-view representation from multi-camera images via Spatiotemporal Transformers.” Lecture Notes in Computer Science, 2022, pp. 1–18, https://doi.org/10.1007/978-3-031-20077-9_1.

Liang, Ming, et al. “PnPNet: End-to-End Perception and Prediction with Tracking in the Loop.” arXiv.Org, 27 June 2020, arxiv.org/abs/2005.14711.

Liao, Bencheng, et al. “MAPTR: Structured Modeling and Learning for Online Vectorized HD...” OpenReview, 29 Sept. 2022, openreview.net/forum?id=k7p_YAO7yE.

Lin, Tsung-Yi, et al. “Microsoft Coco: Common Objects in Context.” Computer Vision – ECCV 2014, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48.

Liu, Rong, et al. “High definition map for automated driving: Overview and analysis.” Journal of Navigation, vol. 73, no. 2, 27 Aug. 2019, pp. 324–341, https://doi.org/10.1017/s0373463319000638.

Liu, Yicheng, Jinghuai Zhang, et al. “Multimodal motion prediction with stacked transformers.” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, https://doi.org/10.1109/cvpr46437.2021.00749.

Liu, Yicheng, Tianyuan Yuan, et al. “VectorMapNet: End-to-End Vectorized HD Map Learning.” arXiv.Org, 26 June 2023, arxiv.org/abs/2206.08920.

Liu, Ze, Yutong Lin, et al. “Swin Transformer: Hierarchical vision transformer using shifted windows.” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2021, https://doi.org/10.1109/iccv48922.2021.00986.

Liu, Zhijian, Haotian Tang, et al. “Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation.” 2023 IEEE International Conference on Robotics and Automation (ICRA), 29 May 2023, https://doi.org/10.1109/icra48891.2023.10160968.

Long, Jonathan, et al. “Fully convolutional networks for semantic segmentation.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, https://doi.org/10.1109/cvpr.2015.7298965.

Mallot, H. A., Bulthoff, H. H., Little, J., and Bohrer, S. ¨Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological cybernetics, 64(3):177–185, 1991.

Marchant, Andy. “Behind the Map: How We Keep Our Maps up to Date: Tomtom Newsroom.” TomTom, 22 Oct. 2019, www.tomtom.com/newsroom/behind-the-map/continuous-map-processing/.

Ngiam, Jiquan, et al. “Scene Transformer: A Unified Architecture for Predicting Multiple Agent Trajectories.” arXiv.Org, 4 Mar. 2022, arxiv.org/abs/2106.08417.

Shan, Tixiao, et al. “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping.” 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 24 Oct. 2020, https://doi.org/10.1109/iros45743.2020.9341176.

Sun, Pei, et al. “Scalability in perception for autonomous driving: Waymo Open Dataset.” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020, https://doi.org/10.1109/cvpr42600.2020.00252.

Tan, Mingxing, and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” arXiv.Org, 11 Sept. 2020, arxiv.org/abs/1905.11946.

Vaswani, Ashish, et al. “Attention Is All You Need.” arXiv.Org, 2 Aug. 2023, arxiv.org/abs/1706.03762.

Wilson, Benjamin, et al. “Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting.” arXiv.Org, 2 Jan. 2023, arxiv.org/abs/2301.00493.

Xiong, Xuan, et al. “Neural map prior for autonomous driving.” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023, https://doi.org/10.1109/cvpr52729.2023.01682.

Yang, Sheng, et al. “A robust pose graph approach for city scale Lidar mapping.” 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2018, https://doi.org/10.1109/iros.2018.8593754.

Yu, Fisher, et al. “Semantic alignment of LIDAR data at City Scale.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, https://doi.org/10.1109/cvpr.2015.7298781.

Zhang, Tianyuan, et al. “MUTR3D: A Multi-camera Tracking Framework via 3D-to-2d queries.” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2022, https://doi.org/10.1109/cvprw56347.2022.00500.

Zhou, Yi, Hui Zhang, et al. “HIMap: Hybrid Representation Learning for End-to-End Vectorized HD Map Construction.” arXiv.Org, 26 Mar. 2024, arxiv.org/abs/2403.08639.

Zhou, Yin, Pei Sun, et al. “End-to-End Multi-View Fusion for 3D Object Detection in Lidar Point Clouds.” arXiv.Org, 23 Oct. 2019, arxiv.org/abs/1910.06528.

Ziegler, Julius, et al. “Making Bertha Drive—an autonomous journey on a historic route.” IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 2, 2014, pp. 8–20, https://doi.org/10.1109/mits.2014.2306552.