A comprehensive research of deep learning approaches for High Definition map construction in autonomous driving

Zimo Wang

School of Computer Science, Beijing Institute of Technology, Beijing, China

1120202440@bit.edu.cn

Abstract. In the realm of autonomous driving, High Definition Maps (HD maps) are indispensable for safe and precise navigation. Traditional HD map construction methods involving point cloud capture and SLAM have proven effective but labor-intensive. This paper addresses the growing interest in leveraging deep learning techniques to streamline HD map creation. This paper presents a systematic exploration of deep learning methodologies for HD map construction. It categorizes these approaches into two core components: Feature Extraction and Feature Decoding. Feature Extraction involves the transformation of input data, comprising images and LiDAR point clouds, into Bird's Eye View (BEV) representations. Feature Decoding is dissected into rasterized map objectives and vector map objectives. Detailed analysis is conducted on prominent methodologies. The paper provides a nuanced evaluation of these deep learning techniques, highlighting their respective strengths and limitations. Factors such as precision, computational efficiency, and the preservation of fine-grained details are considered when selecting the most suitable method. This comprehensive review summarizes and prospects the research in related fields.

Keywords: High Definition Maps, HD Map Construction, Deep Learning, Transformer.

1. Introduction

The notion of a High Definition Map (HD map) was initially introduced during the Mercedes-Benz research planning session in 2010, which subsequently culminated in the Bertha Drive Project [1]. Providing the information of surrounding map elements on road such as lanes, pedestrian crossings, and traffic signs, HD map is an essential part in autonomous driving systems.

Traditionally, such HD semantic maps are often built with point cloud capture, using SLAM to produce globally consistent maps, and then annotate the maps with semantic information. Although this paradigm produces precise HD maps and has been adopted by several organizations that specialize in autonomous driving, it necessitates prohibit human efforts, which also limit the scalability. Furthermore, these approaches are also confronted with accuracy challenges stemming from the process of imprecisely localizing the ego-vehicle for generating local maps from the global reference.

In recent years, with burgeoning popularity of deep learning, an increasing amount of research has been dedicated to the utilization of deep learning techniques for the construction of HD map. Certain deep learning techniques have attained state-of-the-art outcomes within this domain. Existing review literature provides insufficient coverage of this field [2-4]. As a result, the primary objective of this paper is to address this knowledge gap by offering a comprehensive review that systematically consolidates the research accomplishments and progress made in recent years concerning the utilization of deep learning techniques for the construction of HD maps. Through in-depth analysis and synthesis of existing literature, this paper will afford researchers and practitioners the opportunity for a comprehensive understanding of this field while also providing valuable insights for future research directions. Figure 1 is the partition architecture diagram.

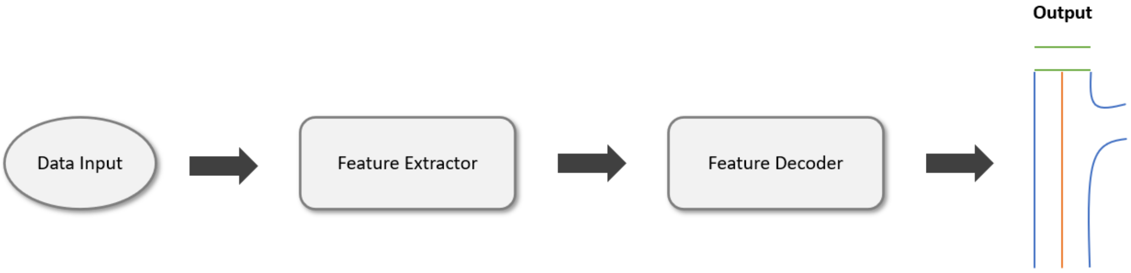

Figure 1. Partition architecture diagram.

2. Overview of the HD map construction method

The process of deep learning techniques for the construction of HD map could be separated into two parts: Feature Extraction and Feature Decoder. In Feature Extraction, input data typically is composed of two components, surrounding images and LiDAR point clouds. The these components undergo a series of preprocessing and feature extraction steps before being transformed into a Bird's Eye View (BEV) representation. Subsequently, more in-depth processing and feature extraction are performed on the information with in the BEV. Within the feature decoder, the extracted features undergo a decoding procedure to produce rasterized or vectorized maps, which are essential for subsequent tasks downstream in the pipeline.

In the following, the algorithms that have emerged in recent years in the field of deep learning-based HD map construction will be systematically introduced. The focus will be on highlighting the different approaches these algorithms take in terms of Feature Extractors and Feature Decoders, and a comparative analysis of their strengths and weaknesses will be provided.

2.1. Feature extractor

The feature extractor in HD map construction plays a crucial role in transforming input data, including images and LiDAR point clouds, into a BEV representation. This transformation is essential to meet the requirements of downstream tasks. The basic idea involves converting both images and LiDAR data into BEV features, which can be subsequently used for various map-related tasks. Additionally, fusion strategies are employed to combine these different sources of information effectively to enhance the overall map generation process.

2.1.1. Convert into BEV. The input of HD map construction task is typically images and LiDAR point cloud which are collected by onboard sensors of autonomous vehicle. The images are taken by several cameras on vehicle which is called perspective view images. In order to meet the semantic map requirements in downstream tasks for the BEV representation, both the perspective view images and LiDAR point cloud should be converted into features in BEV form. To convert Point cloud into BEV, for networks utilizing point cloud input, it is a common practice that all such networks have adopted a variant of PointPillar [5] with dynamic voxelization [6]. The PointPillar method divides the three-dimensional space into multiple pillars and derives feature maps by utilizing the pillar-specific attributes of point clouds associated with each individual pillar. To convert image input into BEV, The HDMapNet [1] directly uses the View-Parsing-Network [4] as conversion. The View-Parsing-Network (VPN) demonstrates strong capabilities in extracting three-dimensional information from two-dimensional images and efficiently achieving the conversion to BEV representation, all while exhibiting excellent scalability. However, it is important to note that this network incurs a significant computational burden and falls short in fully leveraging certain prior information, such as camera angles and intrinsic parameters. Similarly, the InstraGraM [7] used an MLP-based approach to build a unified BEV features map.

VectorMapNet [8] leverages the Inverse Perspective Mapping (IPM) technique to realize such a transformation. IPM is a computer vision technique used to rectify and transform images taken from a perspective or non-planar view into a flat, top-down view. IPM works by reversing the perspective transformation applied to an image, effectively "unwarping" it to remove the distortion caused by the perspective projection. This process allows for more accurate measurements and analysis of objects in the scene, especially when they're supposed to be viewed from a top-down perspective. One significant drawback of IPM is that it assumes a simplified geometric model of the scene, typically assuming planar surfaces. This means that IPM works well for scenarios where objects or surfaces are mostly planar, like road lanes on a flat road. However, it may not work effectively in more complex scenes with non-planar surfaces or when there are significant changes in elevation and perspective.

CenterLineDet [9] integrates the work of both HDMapNet and VectorMapNet. CenterLineDet designed a FusionNet, combining the feature from both IPM and VPN by simply add them together. This approach allows for the integration of the strengths of both methods; however, it comes at the cost of increased time complexity. Moreover, the simple summation of the two methods is equivalent to assigning equal weight to each. Compared to introducing learnable weight variables, this approach may yield slightly inferior results.

In contrast to existing methodologies, SuperFusion [10] proposed depth-aware camera-to-BEV module. This method delves into the utilization of sparse depth priors derived from LiDAR data alongside dense completed depth maps as a form of guidance to enhance the robustness and dependability of image depth prediction. Through this approach, SuperFusion effectively harnesses both a depth prior and precise depth supervision, thereby exhibiting strong generalization capabilities across various challenging environmental scenarios.

MapTR [11] employs the Geometry-guided Kernel Transformer (GKT) [12], an innovative approach within the Transformer architecture for the purpose of learning 2D-to-BEV representations. The GKT method introduces a Bidimensional BEV-to-2D Look-Up Table (LUT) index, which stores the mapping between BEV query indices and image pixel indices. Since the kernel regions remain constant for each BEV network, the LUT can be pre-trained offline, resulting in expedited inference speeds. MapTR has been designed and empirically validated to seamlessly complement various 2D-to-BEV methods while consistently achieving stable performance. By default, MapTR incorporates GKT for its ease of deployment and high computational efficiency.

2.1.2. Fusion. For the works who both use the images and point clouds as input, how to fuse the two kinds of input is also an important question. Most of the concerning works simply concatenate the two kinds of feature together. This method is relatively straightforward yet does not require extensive computational resources. Conversely, SuperFusion devises a sophisticated strategy to harness the mutually enhancing characteristics of these two sensor modalities [10]. SuperFusion divides the fusion strategies into three levels: data-level, feature-level and BEV-level fusion. Under this division, other works only process fusion in feature-level [1][8][9].

In the data-level fusion, SuperFusion undertake the process of projecting LiDAR point clouds onto image planes, resulting in sparse depth images. These sparse depth images, alongside RGB images, are utilized as inputs for the camera-to-BEV transformation module. In feature-level fusion, SuperFusion capitalize on front-view perspective image features to direct the LiDAR BEV features towards long-range predictions. This is accomplished through a cross-attention interaction, facilitating precise long-range HD map prediction. In BEV-level fusion, SuperFusion introduce a BEV alignment module that aligns and merges the perspective image and LiDAR BEV features. The combined BEV features are designed to support a range of tasks, including semantic segmentation, instance embedding, and direction prediction. These fused features are subsequently processed to generate the final HD map prediction.

2.2. Feature decoder

The primary function of the decoder is to generate the final output image based on the BEV features obtained from the feature extractor. Depending on whether the result image is a rasterized map or a vector map, the methods in the feature decoder can be categorized into two types: vector map objectives and rasterized map objectives.

2.2.1. Rasterized map objectives method. A rasterized map is a digital image composed of many small colored blocks called pixels (short for picture elements). In the context of HD map construction, different-colored pixels in a rasterized map are used to represent various road semantic information, such as lane lines, road boundaries, or pedestrian walkways. Given that the generation process of such pixel images closely resembles semantic segmentation in the field of image processing, common methods used in semantic segmentation tasks can be employed to create the aforementioned pixel images. In networks like HDMap [1] and SuperFusion [10], which aim to generate pixel images, Fully Convolutional Network (FCN) is utilized. FCN [13], proposed by Jonathan Long and his colleagues in 2015, is a seminal framework for image semantic segmentation, marking a significant milestone in the application of deep learning to the field of semantic segmentation. To accommodate the requirements of HD map construction tasks, the FCN in both HDMap and SuperFusion incorporates three branches, each specializing in handling semantic prediction, instance embedding, and direction prediction, respectively. The FCN architecture is known for its simplicity, ability to produce relatively fine-grained results, and robustness. However, the results obtained from FCN may not be sufficiently detailed and may lack sensitivity to fine details. This is because FCN does not take into account the relationships between individual pixels, leading to a lack of spatial consistency in its output.

The input of downstream tasks like motion forecasting [14] is typically in the form of vectorized map around ego-vehicle from surrounding cameras or LiDARs. To address this limitation and adapt to the input format required for downstream tasks, the pixel images generated by the aforementioned networks need to undergo preprocessing to be transformed into vector maps. The pre-process is time consuming and strict the model's scalability and performance.

2.2.2. Vector map objective method. To better align with the input format required for downstream tasks, certain networks directly generate vector images as output, without involving dense semantic pixels or intricate post-processing steps. In the vector map, polylines are utilized to represent semantic objects on the road, which are typically represented as a collection of vectors. In order to generate such vector maps, existing networks commonly draw inspiration from Detection Transformer (DETR) [15]. DETR is a deep learning model designed for object detection tasks in images. It was introduced and developed by Facebook AI in 2020. The key innovation of DETR lies in transforming the object detection problem into an end-to-end Transformer model, eliminating the need for traditional object detection steps like anchor boxes and region proposals. This simplifies the entire process and has led to excellent performance, gaining significant attention and application in the field of computer vision.

Although these networks draw inspiration from DETR's principles, the specific implementation process of the Feature Decoder differs among them. VectorNet employs a two-step process for generating vector maps. First, it utilizes a Transformer model similar to DETR, known as the "Map Element Detector," to obtain intermediate results called "Element Keypoints." Subsequently, another Transformer model akin to DETR, referred to as the "Polyline Generator," is used to obtain the final output. The "Map Element Detector" employs Object queries similar to DETR as queries and BEV features as values. On the other hand, the "Polyline Generator" utilizes both Element Keypoints and Object queries as queries, again using BEV features as values. By sequentially connecting the points in the input of the Polyline Generator, a polyline representation of a semantic element in the Vector Map is derived. The introduction of the Keypoint concept is indeed a highlight of VectorNet's work. However, it's important to note that it could potentially introduce issues related to information loss due to the elongation of the pipeline.

Similarly employing a Transformer structure reminiscent of DETR, CenterLineDet's feature decoder concept significantly differs from VectorNet. For each polyline, CenterLineDet creates an agent positioned at the starting point of the polyline. At each time step, the agent moves to another point, serving as the next node of the polyline, or it concludes the prediction of that polyline by determining its current location as the final node. More specifically, when considering a semantic element represented as a polyline with potential branches, CenterLineDet maintains a queue of candidate nodes. At each step, it selects one of these nodes as the starting point for prediction and adds one or more newly generated nodes to the queue. The prediction for that semantic element concludes when the queue becomes empty. Within this DETR-like Transformer structure, is employed to predict the agent's next position. It is evident that CenterLineDet's feature decoder concept aligns more closely with trajectory prediction tasks, where each element in the vector map is forecasted as a trajectory for an agent. In contrast, the approaches of VectorNet and MapTR appear to be more aligned with object detection tasks. In comparison, due to the inability to concurrently output all nodes of the same semantic element like the approach closer to object detection tasks, models based on this trajectory prediction-oriented concept require more time for predictions.

The MapTR framework shares a conceptual basis with VectorNet, but it introduces several significant enhancements to the underlying VectorNet architecture. Firstly, MapTR introduces the concept of equivalent permutations for polylines and polygons. Typically, polylines and polygons are represented as sets of points, and rearranging the order of these points does not change the shape of the polyline or polygon. Consequently, during the training process, specifying a fixed permutation as supervision for point sets is not a reasonable approach. Such fixed permutations conflict with other valid permutations, which can impede the learning process. To address this challenge, MapTR employs a dual-pronged strategy. It computes a set of equivalent permutations that correspond to a given point set. During training, the model's output is compared against this set of permutations to ensure consistency and accuracy. Additionally, MapTR incorporates a hierarchical query embedding scheme, explicitly encoding map elements and enabling hierarchical matching, all rooted in the fundamental concept of modeling permutation equivalence. These innovations, when combined, result in MapTR outperforming VectorNet on the identical dataset.

3. Conclusion

In conclusion, a comprehensive review of recent advancements in the field of deep learning-based HD Map construction for autonomous driving systems has been presented. The two key components of HD map construction, namely Feature Extraction and Feature Decoder, have been explored in detail.

In the Feature Extraction section, the transformation of input data, including surrounding images and LiDAR point clouds, into BEV representations has been discussed. Various approaches have been examined, each with its unique strengths and limitations, with the choice of feature extractor being contingent on specific requirements and constraints. With regard to feature fusion, the paper highlighted the sophisticated strategies employed by SuperFusion to combine image and LiDAR data at data-level, feature-level, and BEV-level fusion, providing a comprehensive solution to maximize the synergy of information between these sensor modalities.

In the Feature Decoder section, two main types of methods were discussed: rasterized map objectives and vector map objectives. Rasterized map objectives, as utilized in HDMap and SuperFusion, employ FCN to generate pixel-based maps. While FCN is robust, it may lack fine-grained detail due to its lack of spatial consistency. Conversely, vector map objective methods, inspired by the DETR model, directly produce vector maps, representing semantic elements as polylines. Notable approaches like VectorNet, CenterLineDet, and MapTR adopt variations of the DETR architecture, with each offering unique advantages in terms of polyline representation and prediction strategies.

In summary, this review has provided valuable insights into the state-of-the-art deep learning techniques employed in HD map construction, enabling researchers and practitioners to gain a comprehensive understanding of this evolving field. As autonomous driving technology continues to advance, the challenges and opportunities in HD map construction will continue to shape the future of self-driving technology.

References

[1] Herrtwich R. The evolution of the HERE HD Live Map at Daimler[J]. HERE Technologies, 2018.

[2] Liu R, Wang J, Zhang B. High definition map for automated driving: Overview and analysis[J]. The Journal of Navigation, 2020, 73(2): 324-341.

[3] Elghazaly G, Frank R, Harvey S, et al. High-Definition Maps: Comprehensive Survey, Challenges and Future Perspectives[J]. IEEE Open Journal of Intelligent Transportation Systems, 2023.

[4] Pan B, Sun J, Leung H Y T, et al. Cross-view semantic segmentation for sensing surroundings[J]. IEEE Robotics and Automation Letters, 2020, 5(3): 4867-4873.

[5] Lang A H, Vora S, Caesar H, et al. Pointpillars: Fast encoders for object detection from point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 12697-12705.

[6] Zhou Y, Sun P, Zhang Y, et al. End-to-end multi-view fusion for 3d object detection in lidar point clouds[C]//Conference on Robot Learning. PMLR, 2020: 923-932.

[7] Shin J, Rameau F, Jeong H, et al. Instagram: Instance-level graph modeling for vectorized hd map learning[J]. arXiv preprint arXiv:2301.04470, 2023.

[8] Liu Y, Yuan T, Wang Y, et al. Vectormapnet: End-to-end vectorized hd map learning[C]//International Conference on Machine Learning. PMLR, 2023: 22352-22369.

[9] Xu Z, Liu Y, Sun Y, et al. CenterLineDet: CenterLine Graph Detection for Road Lanes with Vehicle-mounted Sensors by Transformer for HD Map Generation[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 3553-3559.

[10] Dong H, Zhang X, Jiang X, et al. SuperFusion: Multilevel LiDAR-Camera Fusion for Long-Range HD Map Generation and Prediction[J]. arXiv preprint arXiv:2211.15656, 2022.

[11] Liao B, Chen S, Wang X, et al. Maptr: Structured modeling and learning for online vectorized hd map construction[J]. arXiv preprint arXiv:2208.14437, 2022.

[12] Chen S, Cheng T, Wang X, et al. Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer[J]. arXiv preprint arXiv:2206.04584, 2022.

[13] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

[14] Gao J, Sun C, Zhao H, et al. Vectornet: Encoding hd maps and agent dynamics from vectorized representation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11525-11533.

[15] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]//European conference on computer vision. Cham: Springer International Publishing, 2020: 213-229.