Fact Sheet for CVPR 2023 Autonomous Driving Challenge Track 2 Online HD Map Construnction

Mingchao Jiang[1], Yin Cheng[2], Linghai Liu[1]
[1]GAC R&D Center, Guangzhou, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
jiangshaoyu1993@gmail.com, 3175280282@qq.com, liulinghai9@gmail.com

1. Team Name

Team Name: MapSeg.
EvalAI User Name: @FlyEgle.

2. Method

Method Name: Online High-precision Map Construction with Segmentation-guided Structured Modeling and Learning.

Introduction: The development of online high-definition maps is significant since they provide real-time, accurate, and updatable geographic information for location-based applications, such as autonomous driving and intelligent transportation, thus improving the performance and reliability of these applications. Previous works, such as VectorMapNet [4] and MapTR [3], show that direct model generation of vectorized HD maps is a promising solution. However, these methods did not take into account the usage of global semantic information to improve map construction accuracy. To address this limitation, we propose a segmentation-guided structured model (MapSeg) for online HD map construction, as depicted in Figure 1. Specifically, we added a UV segmentation module (USM) and a BEV segmentation module (BSM) based on the MapTR structure, enabling the model to better capture the semantic information. What's more, to further improve the model's vectorization ability, we proposed a semantic guidance module (SGM). More details of USM, BSM and SGM modules are described as follows. The source code are available at https://github.com/FlyEgle/CVPR_hdmap.

2.1. UV Segmentation Module

In order to enhance the semantic capability of the model, we added a UV segmentation module on top of an FPN structure. The segmentation head from DeepLabV3 [1] is exploited directly as our USM and it is only effective during the training phase. This process is formulated as:

$$\mathbf{O_{uv}} = \mathbf{USM}(\mathbf{FPN}(\mathbf{Backbone}(\mathbf{X_{img}}))) \quad (1)$$

where $\mathbf{X_{img}}$ is surround image view. $\mathbf{O_{uv}}$ is uv segmentation feature.

2.2. BEV Segmentation Module

To further improve the vector results generated during the BEV stage by the model, a BEV segmentation module is incorporated after the BEV features, which is also based on DeepLabV3. We call this module BSM. This process is formulated as:

$$\mathbf{O_{bev}} = \mathbf{BSM}(\mathbf{X_{bev}}) \quad (2)$$

where $\mathbf{X_{bev}}$ is the BEV feature after encoding, $\mathbf{O_{bev}}$ is the BEV segmentation feature.

2.3. Semantic Guidance Module

To make the BEV features be more discriminative, we followed the structure of cross attention, so that the BEV segmentation module can better guide the extraction of the BEV feature. To ensure the completeness of the BEV information, we concatenate the guided BEV features with the original BEV features. This process is formulated as:

$$\mathbf{G_{bev}} = \mathbf{softmax}(\frac{f_Q(\mathbf{O_{bev}})f_K(\mathbf{X_{bev}})}{\sqrt{d_k}})f_V(\mathbf{X_{bev}}) \quad (3)$$

$$\mathbf{Y_{bev}} = \mathbf{Cat}[\mathbf{G_{bev}}, \mathbf{X_{bev}}] \quad (4)$$

where f_Q,f_K,f_V is the project function for Q,K,V. $\mathbf{G_{bev}}$ is the guided BEV features and $\mathbf{Y_{bev}}$ is the concatenated BEV features.

2.4. Loss Function

Given the UV segmentation gt $\mathbf{I_{uv}^{gt}}$, the output from the USM module is defined as $\mathbf{I_{uv}^{usm}}$. Given the BEV segmentation gt $\mathbf{I_{bev}^{gt}}$, the output from the BSM module is defined as $\mathbf{I_{bev}^{bsm}}$. We optimize this module with the following loss function:

$$\mathcal{L}_{usm} = \lambda_1\mathcal{L}_{Dice}(\mathbf{I_{uv}^{usm}}, \mathbf{I_{uv}^{gt}}) + \lambda_2\mathcal{L}_{CE}(\mathbf{I_{uv}^{usm}}, \mathbf{I_{uv}^{gt}}) \quad (5)$$

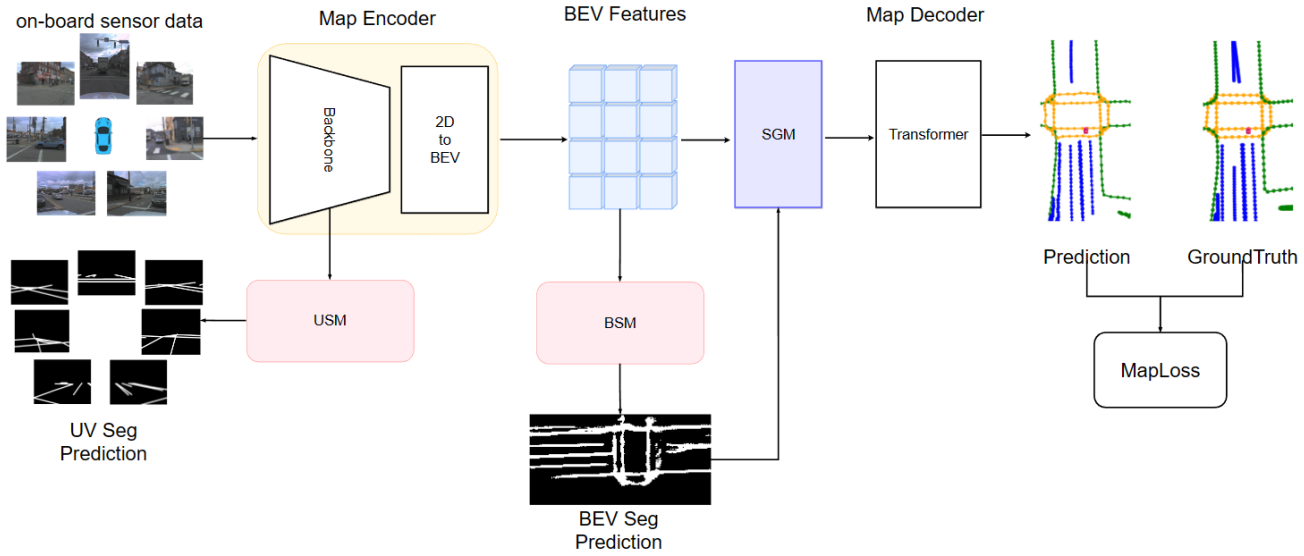

Figure 1. Overview of model architecture.The entire model is consists of by Map Encoder, USM(UV Segmentation Module), BSM(BEV Segmentation Module), SGM(Semantic Guidance Module), and Map Decoder.

$$\mathcal{L}_{bsm} = \lambda_1 \mathcal{L}_{Dice}(\mathbf{I}_{\mathbf{uv}}^{\mathbf{bsm}}, \mathbf{I}_{\mathbf{uv}}^{\mathbf{gt}}) + \lambda_2 \mathcal{L}_{CE}(\mathbf{I}_{\mathbf{uv}}^{\mathbf{bsm}}, \mathbf{I}_{\mathbf{uv}}^{\mathbf{gt}}) \quad (6)$$

$$\mathcal{L}_{seg} = \mathcal{L}_{usm} + \mathcal{L}_{bsm} \quad (7)$$

where $\mathcal{L}_{CE}$ is the Cross-Entropy loss, and $\mathcal{L}_{Dice}$ is the Dice loss:

$$\mathcal{L}_{Dice} = 1 - 2 \cdot \frac{pred \bigcap true}{pred \bigcup true} \quad (8)$$

The total loss of the model can be expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{maptr} + \mathcal{L}_{seg} \quad (9)$$

where $\mathcal{L}_{maptr}$ is the MapTR loss function.

2.5. Training

Model Setting: We generally follow the MapTR-Tiny structure, where the encoder-layer uses one layer and the decoder-layer uses six layers. For DeepLabV3, we follow the structure implemented by the mmsegmentation framework. Considering to reduce the learning difficulty, our semantic segmentation module only predicts two categories, foreground and background.

Data Augmentation: We merge the training data and validation data for training. In the training stage, since the surround image size is not the same, we resized the input images to 1600×2048 and adjusted the camera intrinsic as well. We performed random horizontal flipping, random rotation, and color distortion to augment the training data.

Training Strategy: Our model was optimized with the AdamW [5] method with $\beta_1 = 0.9, \beta_2 = 0.999$ and a batch size of 2. Our model was implemented using PyTorch [6] and trained with 4 NVidia A100 GPUs. The learning rate was initially set to 3×10^{-4} and scheduled with the cosine annealing strategy. ResNet101 [2] was adopted as our backbone, and trained for 100 epochs. To reduce the model inference latency, we use the ResNet50 [2] replace the ResNet101 as the backbone. To ensure the accuracy of the model, we used the ResNet101 best model as the ResNet50 pretrain. Finally, we training 30 epochs with ResNet50 as our results.For $\mathcal{L}_{seg}$ the λ_1 set to 15, λ_2 set to 0.5.

2.6. Testing

We did not perform any model ensembling or test time augmentation (TTA) during the testing phase. In the end, the ResNet50 as backbone was used for testing.

3. Ablation Study

In this section, we presented the performance changes of the proposed different modules on the validation dataset. The baseline is used the ResNet50.

In the table 1, we compared the ablation experiments of different combinations of USM, BSM, and SGM,All module backbone is used the ResNet50.The results showed that all three modules have a significant gain on the original model structure.

In the table 2, we found that the larger resolutions and the larger models can further improve the performance of the model.

Module	size	ped crossing	divider	boundary	map
baseline	800 × 1024	0.5190	0.6162	0.6047	0.5800
USM	800 × 1024	0.5419	0.6185	0.6219	0.5941
BSM	800 × 1024	0.5381	0.6249	0.617	0.5933
USM + BSM	800 × 1024	**0.5442**	0.6405	0.6317	0.6054
USM + BSM + SGM	800 × 1024	0.5397	**0.6512**	**0.6419**	**0.6109**

Table 1. module ablation experiments of USM, BSM, SGM.To boldface indicates the best results.

backbone	size	ped crossing	divider	boundary	map
ResNet50	800 × 1024	0.5397	0.6512	0.6419	0.6109
ResNet50	1600 × 2048	0.5532	0.6685	0.6538	0.6294
ResNet101	1600 × 2048	**0.5714**	**0.6824**	**0.6647**	**0.6452**

Table 2. module ablation experiments for resolution model parameters.

4. Members

Mingchao Jiang (jiangshaoyu1993@gmail.com)
Yin Cheng (3175280282@qq.com)
Linghai Liu (liulinghai9@gmail.com)
The first member will be referred to as the captain of the team.

5. Affliation

GAC R&D Center, Beijing University of Posts and Telecommunications.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[3] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*, 2022. 1

[4] Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022. 1

[5] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 2

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. 2