BEV@DC: Bird's-Eye View Assisted Training for Depth Completion
Supplementary Material

1. Overview

This supplementary material provides more technical details and experimental results about the proposed BEV@DC. Specifically, the algorithm written in pseudo-code for the PV-SPN is introduced in section 2. The definitions of our experiment metrics are shown in section 3. The detailed architecture and the training configuration of our network are presented in section 4. Lastly, we conduct more experiments and demonstrate more visualization results in section 5.

2. Algorithm for PV-SPN

The detailed algorithm for the Point-Voxel Spatial Propagation Network (PV-SPN) is provided in the Algorithm 1.

Algorithm 1: PV Spatial Propagation Network

Input: BEV completion prediction D^{Bev}
Output: Fine-grained 3D completion result
Generate coarse 3D completion via interpolating height dimension $V^{3D} = MLP(D^{Bev})$
for $l \leftarrow 0$ **to** *iterations* L **do**
  Voxel inflation: $V^{3D} \leftarrow \mathrm{V}^{3D_{inf}}$;
  Transform voxel to LiDAR: $V \leftarrow$ Voxel2LiDAR($\mathrm{V}^{3D_{inf}}$);
  K-nearest neighbors search S^{3D} for $v_i \in V$ in the original LiDAR point cloud P to obtain $N(v_i) \leftarrow \{p_i \in \mathcal{S}^{3D}(v_i, P)\}$;
  Geometric-aware aggregation A^{3D} on $N(v_i)$: $s_i^l \leftarrow \mathcal{A}^{3D}(V^{l-1}|P, \mathcal{N}(v_i))$;
  Update $v_i^l \in V^l$ based on s_i^l and threshold k: $v_i^l \leftarrow \{1_{s_i^l \geq k}, 0_{s_i^l < k}\}$;

3. Evaluation Metrics

We choose six different metrics for our experimental evaluation, including root mean square error (RMSE), mean absolute relative error (REL), percentage of pixels satisfying δ_τ, mean absolute error (MAE), inverse RMSE (iRMSE), and inverse MAE (iMAE), which is consistent with all the previous works. [1, 5–7] The definitions of all the metrics are shown as follows.

$$\text{RMSE (mm)} : \sqrt{\frac{1}{n}\sum_x (\hat{d}_x - d_x)^2} \tag{1}$$

$$\text{MAE (mm)} : \frac{1}{n}\sum_x |\hat{d}_x - d_x| \tag{2}$$

$$\text{iRMSE (1/km)} : \sqrt{\frac{1}{n}\sum_x (\frac{1}{\hat{d}_x} - \frac{1}{d_x})^2} \tag{3}$$

$$\text{iMAE (1/km)} : \frac{1}{n}\sum_x |\frac{1}{\hat{d}_x} - \frac{1}{d_x}| \tag{4}$$

$$\text{REL} : \frac{1}{n}\sum_x |\frac{\hat{d}_x - d_x}{d_x}| \tag{5}$$

$$\delta_\tau : max(\frac{d_x}{\hat{d}_x} - \frac{\hat{d}_x}{d_x}) < \tau, \tau \in \{1.25, 1.25^2, 1.25^3\} \tag{6}$$

where d_x is the ground truth depth at valid pixel x (with non-zero depth value), $\hat{d}_x$ is the predicted depth at pixel x, n is the total number of the valid pixels.

4. Implementation Details.

Network Setup. Our proposed method consists of two streams: (1) A Image branch that takes the RGB and depth as input and generates a coarse depth map with the guidance affinity and the confidence map for the following 2D SPN module to refine. (2) A LiDAR branch that intakes the LiDAR voxel grid and produces the BEV prediction for the PV-SPN module to generate the fine-grained 3D voxel completion. The detailed network architecture for the 2D and 3D branches is illustrated in Fig 1. In the camera branch, we apply a U-Net [8] encoder-decoder architecture with ResNet-34 [3] as the backbone for a fair comparison with the previous works. We adopt dynamic SPN [5] in the camera branch as it yields better performance by dynamically generating affinity matrices during iterations. In the LiDAR branch, we apply sparse convolution proposed in MinkowskiNet [2] and construct the similar architecture as

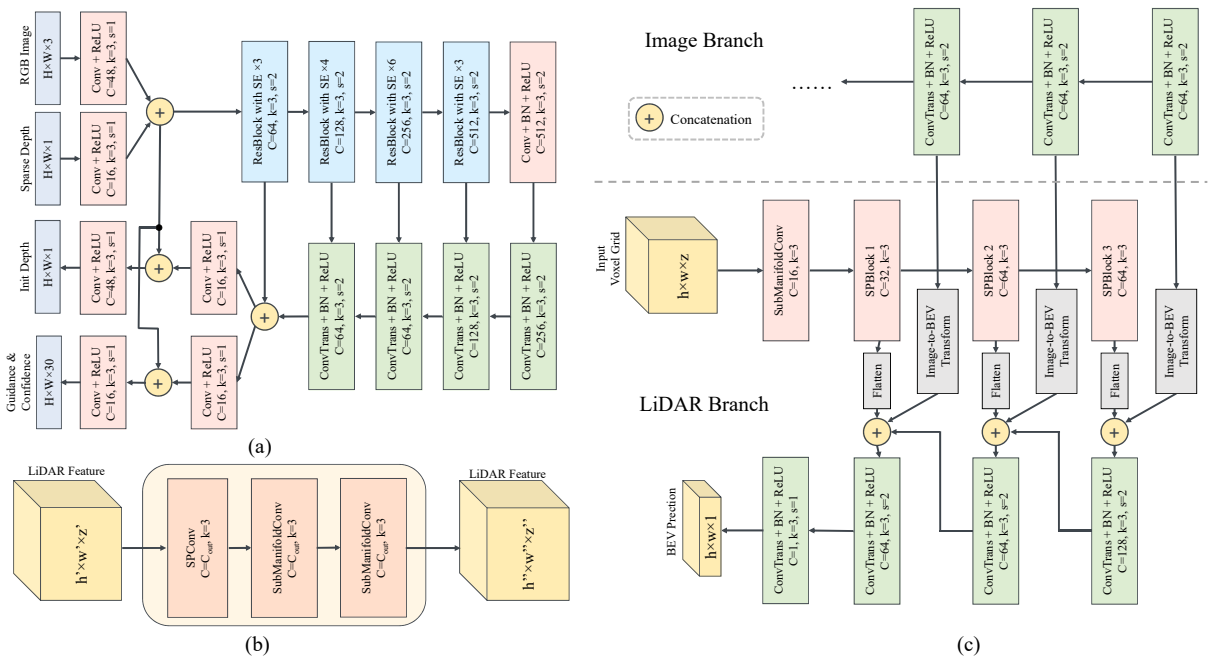

Figure 1. Detailed architecture of the individual components. Part (a) is the image branch that adopts 2D Unet [8] to process the RGB image and corresponding sparse depth measurements. Part (c) is the LiDAR branch that generates BEV prediction from the LiDAR voxel grid using a 3D Unet structure. Part (b) presents a SPBlock in the LiDAR branch. In the above figure, H,W represents the image height and width while h,w,z denotes the voxel spatial shape. ResBlock with SE is a ResNet block [3] with stochastic depth [4]. c,k,s indicate the output channel, kernel sizes, and stride.

the camera branch. The CRBD shares the same structure as the image decoder while using a wider channel dimension since it fuses the features from both LiDAR and camera branches. The number of neighborhood k in PV-SPN is 16, and the hidden channels for the MLPs are identically 64. The propagation steps for both SPN and PV-SPN are 3.

Training Details. In our experiments, we use Adam optimizer with β_1=0.9, β_2=0.999. We first train the camera stream for 30 epochs with the initial learning rate of 1×10^{-3}, and reduce the learning rate by half at 10, 15, 20 and 25, respectively. And we train our whole architecture for 50 epochs with the initial learning rate of 1×10^{-3}, which is decayed by half at 30, 35, 40 and 45, respectively. All the experiments are on 4 NVIDIA V100 GPUs with a batch size of 4.

5. Additional Experiments

Variations of 3D SPN. To get a fine-grained prediction of 3D completion, PV-SPN is proposed to refine the coarse voxel grids from the LiDAR branch. We extend the previous 2D Spatial Propagation Networks (SPN), including Convolutional Spatial Propagation Network (CSPN), Non-Local Spatial Propagation Network (NLSPN), and Dynamic Spatial Propagation Network (DySPN) [1, 5, 6] that refine a 2D coarse depth map to 3D space and compare them with our PV-SPN to demonstrate the effeteness of this module. As shown in Table 1, PV-SPN achieves the best performance among all 3D SPN modules with the least number of iterations.

Table 1. Comparison of four variations of 3D SPN on the KITTI DC validation set.

3D SPN Modules	Iterations	RMSE (mm)	MAE (mm)
3D CSPN [1]	12	733.4	190.8
3D NLSPN [6]	12	727.2	190.1
3D DySPN [5]	6	723.6	188.4
PV-SPN (ours)	3	**719.6**	**187.1**

LiDAR Branch in Other Camera-based Methods. Our plug-and-play solution can be incorporated into different camera-based methods. We extend our modules to one classic camera-based method, *i.e.*, NLSPN. The results on KITTI are shown below, which demonstrates the benefits of the LiDAR branch.

Model	RMSE	MAE	iRMSE	iMAE
NLSPN	783.3	228.9	2.7	1.2
NLSPN + ours	773.9	223.6	2.4	1.0

Visualization. Figure 2 demonstrates more visualization results on KITTI dataset.

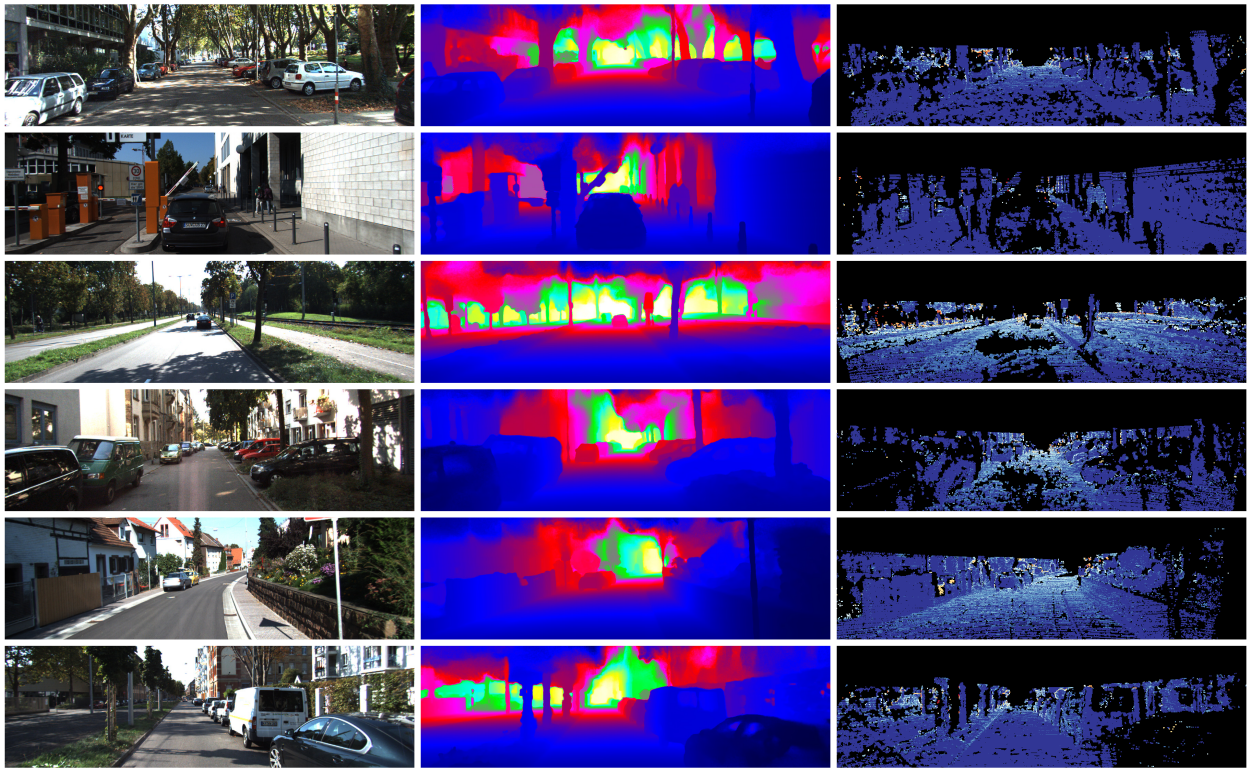

Figure 2. Visualization results on KITTI DC Online Benchmark. From left to right is RGB, our depth prediction, and error map.

References

[1] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network. In *ECCV*, pages 108–125, 2018. 1, 2

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, pages 3075–3084, 2019. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016. 1, 2

[4] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–661, 2016. 2

[5] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic Spatial Propagation Network for Depth Completion. In *AAAI*, 2022. 1, 2

[6] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-kuei Liu, and In So Kweon. Non-Local Spatial Propagation Network for Depth Completion. In *ECCV*, pages 120–136, 2020. 1, 2

[7] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene from Sparse LiDAR Data and Single Color Image. In *CVPR*, pages 3313–3322, 2019. 1

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015. 1, 2