Tri-Perspective View Decomposition for Geometry-Aware Depth Completion

Supplementary Material

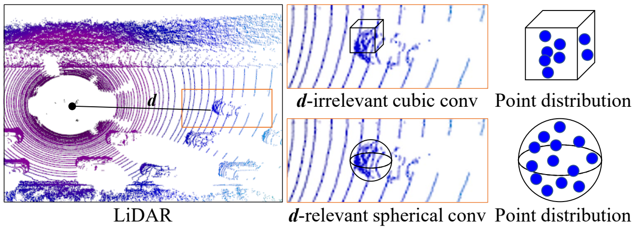

Figure 1. Comparison of the common 3D cubic convolution and our proposed distance-aware spherical convolution.

For one pixel p in the valid pixel set $\mathbb{P}$:	
– REL	$\frac{1}{\lvert\mathbb{P}\rvert}\sum \lvert\mathbf{y}_p - \mathbf{x}_p\rvert / \mathbf{y}_p$
– MAE	$\frac{1}{\lvert\mathbb{P}\rvert}\sum \lvert\mathbf{y}_p - \mathbf{x}_p\rvert$
– iMAE	$\frac{1}{\lvert\mathbb{P}\rvert}\sum \lvert 1/\mathbf{y}_p - 1/\mathbf{x}_p\rvert$
– RMSE	$\sqrt{\frac{1}{\lvert\mathbb{P}\rvert}\sum (\mathbf{y}_p - \mathbf{x}_p)^2}$
– iRMSE	$\sqrt{\frac{1}{\lvert\mathbb{P}\rvert}\sum (1/\mathbf{y}_p - 1/\mathbf{x}_p)^2}$
– RMSELog	$\sqrt{\frac{1}{\lvert\mathbb{P}\rvert}\sum (\log \mathbf{y} - \log \mathbf{x})^2}$
– δ_i	$\frac{\lvert\mathbb{S}\rvert}{\lvert\mathbb{P}\rvert}$, $\mathbb{S}: \max(\mathbf{y}_p/\mathbf{x}_p, \mathbf{x}_p/\mathbf{y}_p) < 1.25^i$

Table 1. Definition of the seven metrics used in the main text.

1. Distance-Aware Spherical Convolution

Fig. 1 illustrates the comparison of our distance-aware spherical convolution (DASC) and the 3D convolution. We observe that the d-relevant DASC involves a higher number of valid points with more balanced distribution.

2. Metric

On KITTI benchmark, we employ RMSE, MAE, iRMSE, and iMAE for evaluation [7, 12, 13, 16]. On NYUv2, TOFDC, and SUN RGBD datasets, RMSE, REL, and δ_i ($i = 1, 2, 3$) are selected for testing [8, 10, 14, 15].

For simplicity, let $\mathbf{x}$ and $\mathbf{y}$ denote the predicted depth and ground truth depth, respectively. Tab. 1 defines the metrics.

3. Loss Function

The total loss function L_{total} consists of three terms, *i.e.*, the front-view L_f, top-view L_t, and side-view L_s. The ground truths of the front, top, and side views are obtained by projecting the annotated point clouds. Following [5, 7, 16], we adopt L_1 and L_2 joint loss functions to denote L_f, L_t, and L_s, *i.e.*, $L_f/L_t/L_s = L_1 + L_2$. As a result, the total loss function L_{total} is defined as:

$$L_{total} = L_f + \alpha L_t + \beta L_s, \tag{1}$$

where α and β are conducted to balance the three terms. Empirically, we set α and β to 0.6 and 0.2, respectively.

4. Implementation Detail

We implement TPVD on Pytorch with four 3090 GPUs. We train it for 50 epochs with Adam [4] optimizer. The initial learning rate is 5×10^{-4} for the first 30 epochs and is reduced to half for every 10 epochs. Following [5, 12], the stochastic depth strategy [2] is used for better training. Also, we employ color jitter and random horizontal flip for data augmentation. The batch size is 3 for each GPU.

5. TOFDC

5.1. Motivation

For depth completion task, the commonly used datasets are KITTI [11] and NYUv2 [9]. Tab. 2 lists the detailed characteristics. KITTI uses LiDAR to collect outdoor scenes, while NYUv2 employs Kinect with time-of-flight (TOF) to capture indoor scenes. However, both LiDAR and Kinect are bulky and inconvenient, especially for ordinary consumers in daily life. Recently, TOF depth sensors have become more common on edge devices (*e.g.*, mobile phones), as depth information is vital for human-computer interaction, such as virtual reality and augmented reality. Therefore, it is important and worthwhile to create a new depth completion dataset on consumer-level edge devices.

5.2. Data Collection

Acquisition System. As illustrated in Fig. 5 (left), the acquisition system consists of the Huawei P30 Pro and Helios, which capture color image and raw depth, and ground truth depth, respectively. The color camera of P30 produces 3648×2736 color images using a 40 megapixel Quad Bayer RYYB sensor, while the TOF camera outputs 240×180 raw depth maps. The industrial-level Helios TOF camera generates higher-resolution depth. Their depth acquisition principle is the same, ensuring consistent depth values.
Data Processing. We calibrate the RGB-D system of the P30 with the Helios TOF camera. We align them on the 640×480 color image coordinate using the intrinsic and extrinsic parameters. The color images and Helios depth maps are cropped to 512×384, while the P30 depth maps to 192×144. Then we conduct nearest interpolation to

Dataset	Outdoor	Indoor	Sensor	Edge Device	Train	Test	Resolution	Real-world
KITTI [11]	✓	×	LiDAR	×	86,898	1,000	$1216 * 352$	✓
NYUv2 [9]	×	✓	Kinect TOF	×	47,584	654	$304 * 228$	×
TOFDC	✓	✓	Phone TOF	✓	10,000	560	$512 * 384$	✓

Table 2. Dataset comparison. Note that these characteristics are calculated according to the **depth completion task**.

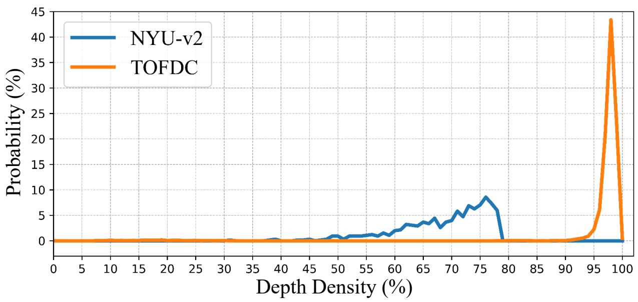

Figure 2. Density-probability comparison of raw depth maps.

upsample the P30 depth maps to 512×384. For the Helios depth maps, there still exist some depth holes caused by environment and object materials (*e.g.*, transparent glass). We use the colorization technique (Levin *et. al*) to fill the holes. Fig. 5 (right) shows the visual result.

Fig. 2 provides the corresponding statistical support. It reveals that the depth density of NYUv2 varies mainly from 60% to 80%, whereas that of TOFDC is highly concentrated between 95% and 100%.

As reported in Tab. 2, we collect the new depth completion dataset TOFDC. It consists of indoor and outdoor scenes, including texture, flower, light, video, and open space in Fig. 3. For the depth completion task, we take the raw depth captured by the P30 TOF lens as input, which is different from NYUv2 where the input depth is sampled from the ground truths.

5.3. Cross-Dataset Evaluation

To validate the generalization on indoor scenes [14], we train TPVD on NYUv2 and test it on SUN RGBD. Comparing Tab. 3-Kinect with Tab. 3, the errors of all methods increase and the accuracy decreases due to different RGB-D sensors. When comparing Tab. 3-Xtion with Tab. 3, since the data is from different Xtion devices, we discover that the performance drops by large margins. However, Tab. 3 reports that our TPVD still achieves the lowest errors and the highest accuracy under Kinect V1 and Xtion splits. For example, under Xtion split, the RMSE of TPVD is 9 mm superior to those of the second best NLSPN [7] and PointDC [14]. These facts evidence the powerful cross-dataset generalization ability of our TPVD.

Method	RMSE (m) ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Collected by Kinect V1					
CSPN [1]	0.729	0.504	69.1	77.8	84.0
NLSPN [7]	0.093	0.020	98.9	99.6	99.7
CostDCNet [3]	0.119	0.033	98.1	99.6	99.7
GraphCSPN [6]	0.094	0.023	98.8	99.6	99.7
PointDC [14]	0.092	0.023	98.9	99.6	99.8
TPVD (ours)	**0.087**	**0.022**	**99.1**	**99.7**	**99.8**
Collected by Xtion					
CSPN [1]	0.490	0.179	84.5	91.5	95.1
NLSPN [7]	0.128	0.015	99.0	99.7	99.9
CostDCNet [3]	0.207	0.028	97.8	99.1	99.5
GraphCSPN [6]	0.131	0.017	99.0	99.7	99.9
PointDC [14]	0.128	0.016	99.1	99.7	99.9
TPVD (ours)	**0.119**	**0.014**	**99.3**	**99.8**	**99.9**

Table 3. Cross-dataset evaluation on SUN RGBD benchmark.

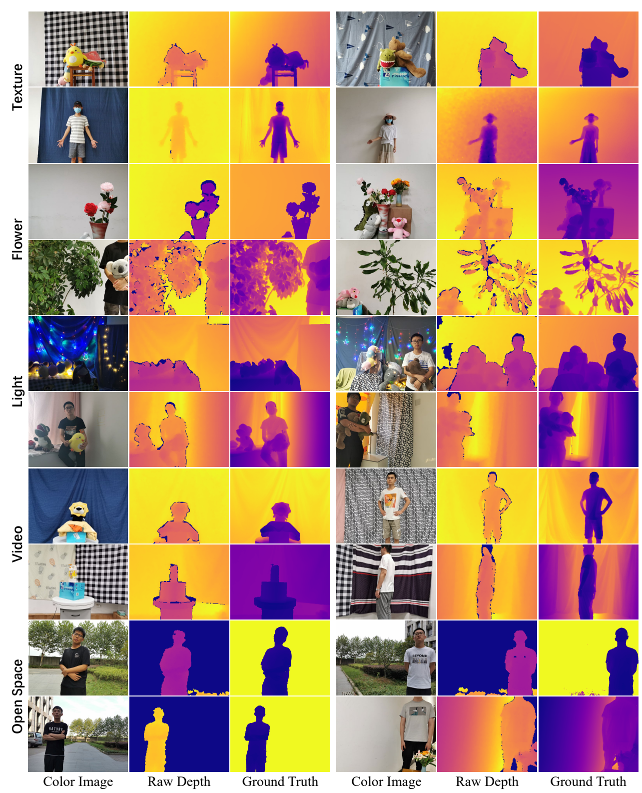

Figure 3. TOFDC examples in different scenarios.

References

[1] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. In *ECCV*, pages 103–119, 2018. 2

[2] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016. 1

[3] Jaewon Kam, Jungeon Kim, Soongjin Kim, Jaesik Park, and Seungyong Lee. Costdcnet: Cost volume based depth completion for a single rgb-d image. In *ECCV*, pages 257–274. Springer, 2022. 2

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Computer Ence*, 2014. 1

[5] Yuankai Lin, Hua Yang, Tao Cheng, Wending Zhou, and Zhouping Yin. Dyspn: Learning dynamic affinity for image-guided depth completion. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. 1

[6] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *ECCV*, pages 90–107. Springer, 2022. 2

[7] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, 2020. 1, 2

[8] 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

[9] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 1, 2

[10] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020. 1

[11] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, pages 11–20, 2017. 1, 2

[12] Yufei Wang, Bo Li, Ge Zhang, Qi Liu, Tao Gao, and Yuchao Dai. Lrru: Long-short range recurrent updating networks for depth completion. In *ICCV*, pages 9422–9432, 2023. 1

[13] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *ECCV*, pages 214–230, 2022. 1

[14] Zhu Yu, Zehua Sheng, Zili Zhou, Lun Luo, Si-Yuan Cao, Hong Gu, Huaqi Zhang, and Hui-Liang Shen. Aggregating feature point cloud for depth completion. In *ICCV*, pages 8732–8743, 2023. 1, 2

[15] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *CVPR*, pages 18527–18536, 2023. 1

[16] Wending Zhou, Xu Yan, Yinghong Liao, Yuankai Lin, Jin Huang, Gangming Zhao, Shuguang Cui, and Zhen Li. Bev@ dc: Bird's-eye view assisted training for depth completion. In *CVPR*, pages 9233–9242, 2023. 1