Tri-Perspective View Decomposition for Geometry-Aware Depth Completion

Zhiqiang Yan[1], Yuankai Lin[2], Kun Wang[1], Yupeng Zheng[3], Yufei Wang[4],
Zhenyu Zhang[5], Jun Li[1]*, and Jian Yang[1]*
[1]PCA Lab†, Nanjing University of Science and Technology, China
[2]Huazhong University of Science and Technology [3]Chinese Academy of Sciences
[4]Northwestern Polytechnical University [5]Nanjing University
{yanzq, kunwang, junli, csjyang}@njust.edu.cn, zhangjesse@foxmail.com
linyuankai@hust.edu.cn, zhengyupeng2022@ia.ac.cn, wangyufei1951@gmail.com

Abstract

Depth completion is a vital task for autonomous driving, as it involves reconstructing the precise 3D geometry of a scene from sparse and noisy depth measurements. However, most existing methods either rely only on 2D depth representations or directly incorporate raw 3D point clouds for compensation, which are still insufficient to capture the fine-grained 3D geometry of the scene. To address this challenge, we introduce ***T****ri-****P****erspective* ***V****iew* ***D****ecomposition (****TPVD****), a novel framework that can explicitly model 3D geometry. In particular, (1) TPVD ingeniously decomposes the original point cloud into three 2D views, one of which corresponds to the sparse depth input. (2) We design TPV Fusion to update the 2D TPV features through recurrent 2D-3D-2D aggregation, where a Distance-Aware Spherical Convolution (DASC) is applied. (3) By adaptively choosing TPV affinitive neighbors, the newly proposed Geometric Spatial Propagation Network (GSPN) further improves the geometric consistency. As a result, our TPVD outperforms existing methods on KITTI, NYUv2, and SUN RGBD. Furthermore, we build a novel depth completion dataset named TOFDC, which is acquired by the time-of-flight (TOF) sensor and the color camera on smartphones. Project page.*

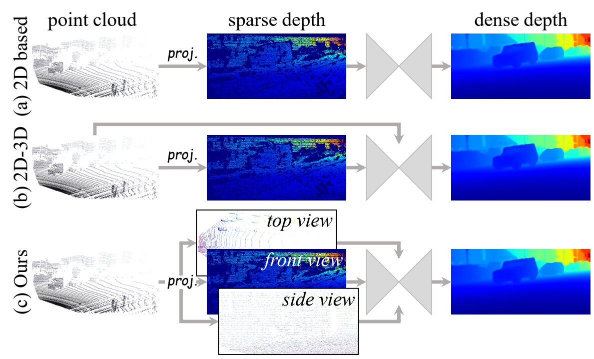

Figure 1. Framework comparison. (a) Previous 2D methods focus on 2D space to recover dense depth, and (b) recent 2D-3D joint approaches introduce 3D point clouds for assistance. Differently, (c) our TPVD decomposes the 3D point clouds into three 2D views to densify the sparse input while preserving the 3D geometry.

1. Introduction

Depth completion [38], the technique of recovering dense depth maps from sparse ones, has a variety of applications in computer vision, such as scene understanding [17, 21, 34, 44, 47, 57, 59], 3D reconstruction [26, 28, 32, 46, 49, 56], and autonomous driving [40, 48, 51, 52, 54, 58, 60]. All of these applications are highly dependent on accurate and reliable depth predictions. However, due to the constraints of hardware development and challenging environment, depth sensors are unable to deliver pixel-wise depth feedback, particularly in outdoor scenarios where the depth density is as low as 5%. Thus, it is essential and worthwhile to complete the void areas of sparse depth for realistic applications.

Most previous depth completion methods [2, 7, 12, 28, 30, 41, 50, 54] focus on 2D feature space to learn depth representations, leading to a severe lack of 3D geometric information. As an alternative, some recent approaches [3, 14, 25, 31, 53, 55, 60] attempt to incorporate 3D geometric priors directly from raw point clouds, rather than relying only on 2D representations. For example, [53, 60] extract point cloud features to embed 3D geometry into their 2D depth generation branches. However, as we known that the point clouds are extremely sparse and their point distributions are varying in different distances, both of which deeply impede the performance of recent models.

*Corresponding authors
†PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Sci. & Tech.

To address the above issues, we propose a novel framework called tri-perspective view decomposition (TPVD). As shown in Fig. 1, unlike existing 2D-3D joint methods [3, 53, 60], TPVD cleverly decomposes the 3D point clouds into three 2D views: top, front, and side. It is worth mentioning that the sparse depth input is exactly included in the front-view map. This decomposition enables TPVD to densify the sparse 3D point clouds in 2D space using 2D convolutions. To leverage the 3D geometric priors more effectively, TPVD employs a recurrent 2D-3D-2D TPV Fusion scheme. In this scheme, the denser 2D TPV features are projected back to 3D space to obtain coarse structural representations. Then, a distance-aware spherical convolution (DASC) is applied to encode the points with varying distributions in a compact spherical space, contributing to refined geometric structures. Next, the 3D spherical features are re-projected into 2D space to update the initial 2D TPV features. That is to say, the 2D process predicts more valid pixels to enrich the 3D process with denser points, while the 3D process captures geometry and feeds it back to the 2D process. These two processes complement each other.

Moreover, TPVD incorporates a plug-and-play geometric spatial propagation network (GSPN) for full-scale 3D geometric refinement. Unlike previous 2D SPN [7, 20, 30, 45] and 3D SPN [25, 60] methods that generate their affinitive neighbors in either a single 2D space or a bird's-eye view space, GSPN constructs the affinity simultaneously in the three decomposed 2D TPV spaces and their joint 3D projection space. Therefore, the affinity preserves both the neighborhood information and the 3D geometric structures.

In addition, since depth information plays a crucial role in accurate 3D reconstruction and human-computer interaction, time-of-flight (TOF) depth sensors are increasingly equipped on edge mobile devices. In this paper, we collect a new depth completion dataset termed TOFDC, with a smartphone that has both TOF lens and color camera.

In summary, our contributions are as follows:

- We introduce a novel framework termed *TPVD*, which densifies the sparse input whilst retaining 3D geometry.
- *TPV Fusion* is proposed to leverage the 3D geometry effectively via recurrent 2D-3D-2D interaction, where *DASC* is applied to handle the varying distributions of LiDAR points. Besides, we design *GSPN* to further produce fine-grained 3D geometric structures.
- We build *TOFDC*, a new smartphone-based depth completion dataset. Moreover, our method consistently outperforms the state-of-the-art approaches on four datasets: KITTI, NYUv2, SUN RGBD, and TOFDC.

2. Related Work

2D Based Depth Completion. Usually, the sparse depth is taken from structured light [35], TOF [11], LiDAR [1, 38], stereo cameras [10, 29], or structure from motion [33, 42].

Recent 2D based image-guided methods [8, 12, 15, 41] focus on RGB-D fusion by direct concatenation or summation. Differently, GuideNet [37] adopts a guided filtering, whose kernel weight is from the guided RGB image. FCFR-Net [22] designs an energy-based fusion to integrate the RGB-D features. RigNet [48] and RigNet++ [50] propose a new guidance unit with low complexity to produce the dynamic kernel. GFormer [32] and CFormer [54] concurrently leverage convolution and transformer to extract both local and long-range representations. Most recently, LRRU [41] presents a large-to-small dynamical kernel scope to capture long-to-short dependencies. However, these 2D based methods deployed in 2D space cannot reserve very precise 3D spatial geometric information.

2D-3D Joint Depth Completion. It is more intuitive and effective to capture geometric structures with 3D representations, such as surface normals [31, 44], graphs [25, 55], point clouds [3, 14, 53], and voxels [60].

For the first time, DLiDAR [31] and DepthNormal [44] introduce surface normals to boost the performance. In view of the effectiveness of the graph neural networks in representing neighborhood relation, ACMNet [55] applies attention-based graph propagation for multi-modal fusion. GraphCSPN [25] leverages convolution neural networks as well as graph neural networks in a complementary way for geometric learning. Lately, FuseNet [3] and PointDC [53] involve LiDAR point cloud branches to model 3D geometry. Moreover, BEV@DC [60] adopts point-voxel architecture based on bird's-eye view for better effectiveness-efficiency trade-off. Different from these 2D-3D joint methods, our TPVD restores dense 2D depth in 2D space while retaining the 3D geometric priors via point cloud decomposition.

Spatial Propagation Network. SPN [7] is increasingly emerging in both 2D based [20, 30, 50] and 2D-3D joint [25, 60] depth completion methods. It digs local or non-local neighbors by 2D and 3D anisotropic filtering kernels.

Initially, 2D SPNs [24] are first proposed to learn pairwise similarity matrix. CSPN [5] conducts recursive convolutions with fixed local neighborhood kernels for improvement, while CSPN++ [7] learns adaptive kernel sizes. PENet [12] further enlarges the receptive fields with dilated convolutions. Differently, NLSPN [30] incorporates non-local neighbors via deformable convolutions. Similarly, DySPN [20] produces dynamic non-linear neighbors by attention mechanism. 3D SPNs [6] are commonly embedded in 2D-3D joint methods to utilize 3D geometry. For example, S3CNet [4] computes key spatial features from LiDAR by a 3D spatial propagation unit. GraphCSPN [25] uses geometric constraints to regularize the 3D propagation. Recently, BEV@DC [60] conducts a point-voxel spatial propagation network for 3D dense supervision. Differently, we aggregate the 2D affinitive neighbors in 2D TPV spaces, resulting in gradual refinement of 3D geometric awareness.

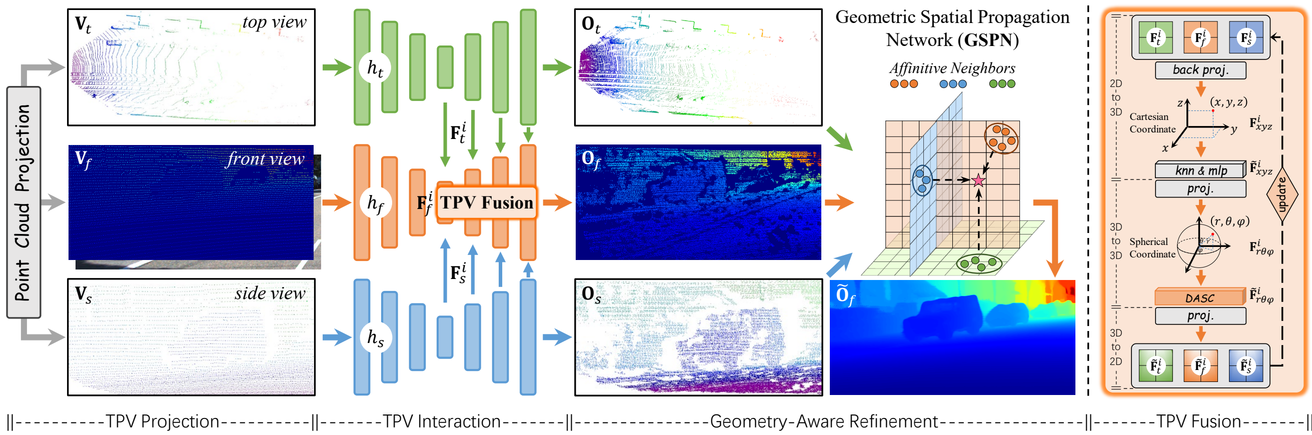

Figure 2. Pipeline of TPVD. The 3D point cloud is first projected into top, side, and front views, where the raw 2D sparse depth input is included in the front view. Then the three views are fed into 2D UNets to produce TPV features that are aggregated by the 2D-3D-2D TPV Fusion, obtaining denser depth with richer geometry. Finally, on the output side, the plug-and-play geometric spatial propagation network (GSPN) generates refined depth results with consistent geometry. *DASC* refers to the distance-aware spherical convolution.

3. TPVD

3.1. Overview

Recent works [3, 53, 60] tend to introduce 3D point clouds to boost the 2D depth completion. Differently, this paper restores dense depth mainly in 2D space, whilst retaining the 3D geometric priors via point cloud decomposition.

Fig. 2 shows our pipeline that consists of ① TPV projection, ② TPV interaction, and ③ geometry-aware refinement. Specifically, in ① the 3D point cloud is first projected into top, side, and front sparse depth views. Then in ② three subnetworks are employed to extract the TPV features, where the TPV Fusion with a distance-aware spherical convolution (DASC) is designed to leverage the 3D geometric priors. Finally, to obtain dense completion with more fine-grained geometry, in ③ the geometric spatial propagation network (GSPN) further improves the geometric consistency.

3.2. TPV Projection

Given a 2D sparse depth map $\mathbf{S} \in \mathbb{R}^{H\times W}$ with the binary mask m, we first transform it into a 3D point cloud, which is then processed by a Multi-layer Perceptron (MLP) and two continuous convolutions [3] to generate the point feature $\mathbf{P} \in \mathbb{R}^{N\times 3}$. Then we employ $\mathcal{P}_{tpv}$ to project the 3D $\mathbf{P}$ into 2D orthogonal top-view $\mathbf{V}_t \in \mathbb{R}^{W\times D}$, side-view $\mathbf{V}_s \in \mathbb{R}^{D\times H}$, and front-view $\mathbf{V}_f \in \mathbb{R}^{H\times W}$. Particularly, we combine $\mathbf{S}$ and $\mathbf{V}_f$ via the mask m to update $\mathbf{V}_f$:

$$\begin{aligned} &\mathbf{V}_t,\ \mathbf{V}_s,\ \mathbf{V}_f = \mathcal{P}_{tpv}(\mathbf{P}), \\ &\text{new } \mathbf{V}_f = \mathbf{S} + (1-m)\mathbf{V}_f. \end{aligned} \tag{1}$$

Unless stated, we use $\mathbf{V}_f$ to represent the new $\mathbf{V}_f$ below.

3.3. TPV Interaction

In Fig. 2, we use h_t, h_s, and h_f subnetworks to encode $\mathbf{V}_t$, $\mathbf{V}_s$, and $\mathbf{V}_f$, as well as the image $\mathbf{I}$ that is aligned with $\mathbf{V}_f$. In each ith layer of the three decoders, their intermediate features are severally denoted as $\mathbf{F}_t^i \in \mathbb{R}^{W_i\times D_i\times C_i}$, $\mathbf{F}_s^i \in \mathbb{R}^{D_i\times H_i\times C_i}$, and $\mathbf{F}_f^i \in \mathbb{R}^{H_i\times W_i\times C_i}$. While $1 \leq i \leq 4$:

$$\mathbf{F}_t^i,\ \mathbf{F}_s^i,\ \mathbf{F}_f^i = h_t(\mathbf{V}_t),\ h_s(\mathbf{V}_s),\ h_f(\mathbf{V}_f,\ \mathbf{I}). \tag{2}$$

TPV Fusion. After obtaining the three 2D TPV features, we introduce TPV Fusion. In Fig. 2 (right), there are three steps in a single iteration of the fusion process:

(1) *2D-to-3D*: To learn 3D geometric priors, the 2D $\mathbf{F}_t^i$, $\mathbf{F}_s^i$, and $\mathbf{F}_f^i$ are jointly projected back to the 3D Cartesian coordinate, yielding $\mathbf{F}_{xyz}^i$. Then, the k-Nearest Neighbor (KNN) computes the k relevant neighbors, while MLP further maps the aggregated features, obtaining the 3D $\tilde{\mathbf{F}}_{xyz}^i$:

$$\mathbf{F}_{xyz}^i = \mathcal{P}_{tpv}^{-1}(\mathbf{F}_t^i,\ \mathbf{F}_s^i,\ \mathbf{F}_f^i), \tag{3}$$

$$\tilde{\mathbf{F}}_{xyz}^i = h_{km}(\mathbf{F}_{xyz}^i), \tag{4}$$

where $h_{km}(\cdot)$ denotes the combined KNN and MLP.

From the blue bars of Fig.3 we observe that, *the point clouds exhibit extreme sparsity that is less than 6%, with their point distributions varying across different distances.* **To weaken the negative impact of the diverse point distributions, a 3D-to-3D strategy is adopted**.

(2) *3D-to-3D*: The 3D cubic $\tilde{\mathbf{F}}_{xyz}^i$ is re-projected into the 3D spherical coordinate by $\mathcal{P}_{sph}$ that produces $\mathbf{F}_{r\theta\varphi}^i$. Then, a distance-aware spherical convolution (DASC) is applied to create the 3D spherical feature $\tilde{\mathbf{F}}_{r\theta\varphi}^i$, which refines the

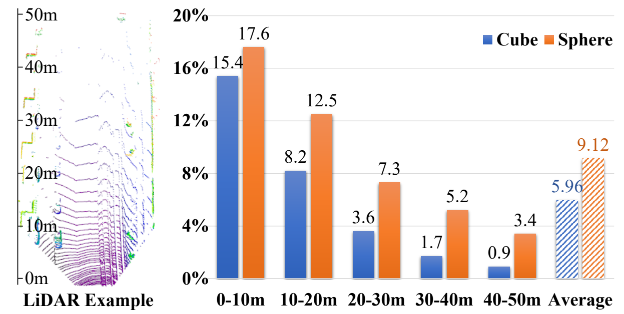

Figure 3. Percentage of non-empty units across different distances between cubic and our spherical transformations.

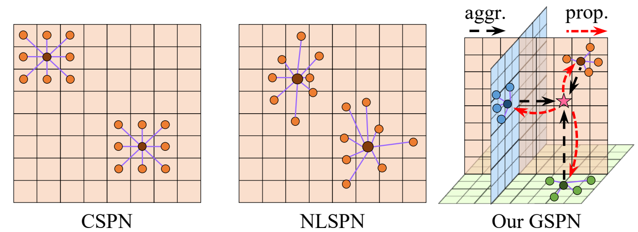

Figure 4. Comparison of SPNs [5, 30] with different neighbor sets. 'aggr.' refers to aggregation while 'prop.' indicates propagation.

geometry in the more compact space:

$$\mathbf{F}^{i}_{r\theta\varphi} = \mathcal{P}_{sph}(\tilde{\mathbf{F}}^{i}_{xyz}), \tag{5}$$

$$\tilde{\mathbf{F}}^{i}_{r\theta\varphi} = h_{dasc}(\mathbf{F}^{i}_{r\theta\varphi}), \tag{6}$$

where $h_{dasc}(\cdot)$ refers to the DASC function (see Eq. 9).

From the orange bars of Fig. 3 we discover that, our 3D-to-3D strategy can better balance the varying point distributions, especially over long distances. After extracting the rich geometric structures in 3D space, **we employ a 3D-to-2D tactic to further densify the sparse depth**.

(3) *3D-to-2D*: The 3D feature $\tilde{\mathbf{F}}^{i}_{r\theta\varphi}$ is projected into 2D space to update the initial 2D $\mathbf{F}^{i}_{t}$, $\mathbf{F}^{i}_{s}$, and $\mathbf{F}^{i}_{f}$ with 2D convolutions h_{2c}, yielding new 2D TPV features:

$$\tilde{\mathbf{F}}^{i}_{t},\ \tilde{\mathbf{F}}^{i}_{s},\ \tilde{\mathbf{F}}^{i}_{f} = h_{2c}(\mathcal{P}_{tpv}(\mathcal{P}^{-1}_{sph}(\tilde{\mathbf{F}}^{i}_{r\theta\varphi}))). \tag{7}$$

In the TPV Fusion process, the 2D decoder layers generate an increased number of valid pixels, which enriches the 3D process with a higher density of points. Concurrently, the 3D process captures geometry and feeds it back into the 2D process. These two processes are complementary.

Particularly, at the output ends of the three TPV subnetworks, we employ three 2D convolutions to predict coarse TPV depth results, obtaining:

$$\mathbf{O}_t,\ \mathbf{O}_s,\ \mathbf{O}_f = h_{2c}(\tilde{\mathbf{F}}^{4}_{t}),\ h_{2c}(\tilde{\mathbf{F}}^{4}_{s}),\ h_{2c}(\tilde{\mathbf{F}}^{4}_{f}). \tag{8}$$

Distance-Aware Spherical Convolution. Given the 3D input $\mathbf{F}^{i}_{r\theta\varphi}$ in Eq. 5, it is sliced by $\mathcal{S}$ into different spherical subareas $\mathbf{A}_{sph} = \{\mathbf{A}^{1}_{sph}, \cdots, \mathbf{A}^{j}_{sph}\}$, each with larger volume $|\mathbf{A}^{j}_{sph}|$ as the distance d increases, *i.e.*, $|\mathbf{A}^{j}| \propto d$. Then, these spherical subareas are flattened by $\mathcal{F}$[1] into cubic shapes $\mathbf{A}_{cub} = \{\mathbf{A}^{1}_{cub}, \cdots, \mathbf{A}^{j}_{cub}\}$ and filtered by h_{3c}, a 3D convolution with kernel $3 \times 3 \times 3$ and stride 1. Thus, Eq. 6 can be written as:

$$\tilde{\mathbf{F}}^{i}_{r\theta\varphi} = \mathcal{F}^{-1}(h_{3c}(\mathcal{F}(\mathcal{S}(\mathbf{F}^{i}_{r\theta\varphi})))). \tag{9}$$

[1]Equirectangular projection (ERP) in DUL [49]

3.4. Geometry-Aware Refinement

Geometric Spatial Propagation Network. SPNs [5, 24] are widely used to recursively refine the coarse depth $\mathbf{O}_f$. Let $\mathbf{O}_{f(a,b)}$ denotes one pixel at (a,b), while $\mathbf{N}_{f(a,b)}$ indicates its neighbors, one of which is located at (m,n). The propagation of $\mathbf{O}_{f(a,b)}$ at step $(l+1)$ is defined as:

$$\mathbf{O}^{l+1}_{f(a,b)} = (1 - \sum_{m,n} \omega^{m,n}_{f(a,b)})\mathbf{O}^{l}_{f(a,b)} + \sum_{m,n} \omega^{m,n}_{f(a,b)} \mathbf{O}^{l}_{f(m,n)}, \tag{10}$$

where $\omega^{m,n}_{f(a,b)}$ is the affinity of pixels at (a,b) and (m,n).

In Fig. 4, the key of SPNs is how to search for the neighbor set $\mathbf{N}_{f(a,b)}$. In 2D space, CSPN [5] constructs $\mathbf{N}^{CS}_{f(a,b)}$ within a fixed square area excluding the centre pixel, while NLSPN [30] deforms it in $\mathbb{R}^{H \times W}$ to build $\mathbf{N}^{NL}_{f(a,b)}$:

$$\mathbf{N}^{CS}_{f(a,b)} = \{\mathbf{O}_{f(a+u,b+v)} \mid u,v \in \{-1,0,1\}\}, \tag{11}$$

$$\mathbf{N}^{NL}_{f(a,b)} = \{\mathbf{O}_{f(a+u,b+v)} \mid u,v \in h_{off}(\mathbf{I},\mathbf{S},a,b)\}, \tag{12}$$

where h_{off} learns the offset based on the RGB-D input.

Differently, given $\mathbf{O}^{l}_{t}$, $\mathbf{O}^{l}_{s}$, and $\mathbf{O}^{l}_{f}$, our GSPN uses the deformable technique $h_{nl}(\cdot)$ in Eqs. 10 and 12 to produce the front-view $\mathbf{O}^{l+1}_{f}$, as well as the top-view $\mathbf{O}^{l+1}_{t}$ and side-view $\mathbf{O}^{l+1}_{s}$ in TPV spaces. Then the three views are aggregated in 3D space [13, 61] via projection and MLP. At last, the 3D feature is propagated back to the TPV spaces for refinement:

$$\tilde{\mathbf{O}}^{l+1}_{t},\ \tilde{\mathbf{O}}^{l+1}_{s},\ \tilde{\mathbf{O}}^{l+1}_{f} = h_{gspn}(\mathbf{O}^{l+1}_{t}, \mathbf{O}^{l+1}_{s}, \mathbf{O}^{l+1}_{f}), \tag{13}$$

where $h_{gspn}(\cdot)$ refers to $\mathcal{P}_{tpv}(h_{mlp}(\mathcal{P}^{-1}_{tpv}(h_{nl}(\cdot))))$.

4. TOFDC

Fig. 5 shows the data acquisition system and data comparison between NYUv2 [35] and our TOFDC. The system consists of the Huawei P30 Pro (for color image and raw depth) and Helios (for ground truth depth). We find that the depth of TOFDC is much denser than NYUv2. Fig. 6 shows the distribution of different scenarios in TOFDC, which stands for texture, flower, light, open space, and video, and we have collected **10,000** RGB-D pairs from these scenarios in total. Please see our appendix for more details.

Method	2D	3D	Params. (M) ↓	RMSE (mm) ↓	MAE (mm) ↓	iRMSE (1/km) ↓	iMAE (1/km) ↓	Publication
CSPN [5]	✓		17.4	1019.64	279.46	2.93	1.15	ECCV 2018
S2D [28]	✓		26.1	814.73	249.95	2.80	1.21	ICRA 2019
NConv [8]	✓		**0.36**	829.98	233.26	2.60	1.03	PAMI 2020
CSPN++ [7]	✓		26.0	743.69	209.28	2.07	0.90	AAAI 2020
NLSPN [30]	✓		25.8	741.68	199.59	1.99	0.84	ECCV 2020
GuideNet [37]	✓		73.5	736.24	218.83	2.25	0.99	TIP 2020
TWISE [15]	✓		1.45	840.20	195.58	2.08	0.82	CVPR 2021
FCFRNet [22]	✓		50.6	735.81	217.15	2.20	0.98	AAAI 2021
PENet [12]	✓		131.5	730.08	210.55	2.17	0.94	ICRA 2021
DySPN [19]	✓		26.3	709.12	192.71	1.88	0.82	AAAI 2022
RigNet [48]	✓		65.2	712.66	203.25	2.08	0.90	ECCV 2022
CFormer [54]	✓		83.5	708.87	203.45	2.01	0.88	CVPR 2023
RigNet++ [50]	✓		19.9	710.85	202.45	2.01	0.89	arXiv 2023
LRRU [41]	✓		21.0	696.51	189.96	1.87	**0.81**	ICCV 2023
DepthNormal [44]	✓	✓	~ 40	777.05	235.17	2.42	1.13	ICCV 2019
FuseNet* [3]	✓	✓	1.9	752.88	221.19	2.34	1.14	ICCV 2019
DLiDAR* [31]	✓	✓	53.4	758.38	226.50	2.56	1.15	CVPR 2019
ACMNet [55]	✓	✓	4.9	744.91	206.09	2.08	0.90	TIP 2021
PointFusion[14]	✓	✓	8.7	741.90	201.10	1.97	0.85	ICCV 2021
GraphCSPN [25]	✓	✓	26.4	738.41	199.31	1.96	0.84	ECCV 2022
BEV@DC [60]	✓	✓	30.8	697.44	189.44	1.83	0.82	CVPR 2023
PointDC [53]	✓	✓	25.1	736.07	201.87	1.97	0.87	ICCV 2023
TPVD (ours)	✓	✓	31.2	**693.97**	**188.60**	**1.82**	**0.81**	CVPR 2024

Table 1. Quantitative results on KITTI online depth completion leaderboard. 2D and 3D refer to models that involve 2D and 3D representations, respectively. * denotes models that involve additional training data. The **best** and the second best metrics are highlighted.

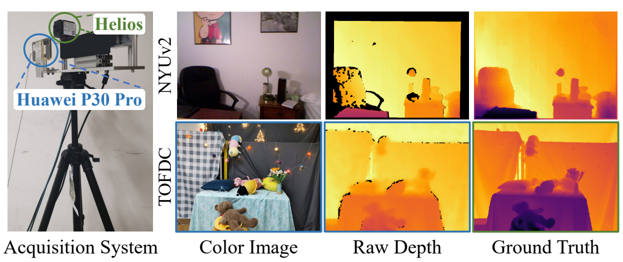

Figure 5. Acquisition system (left) and data comparison (right).

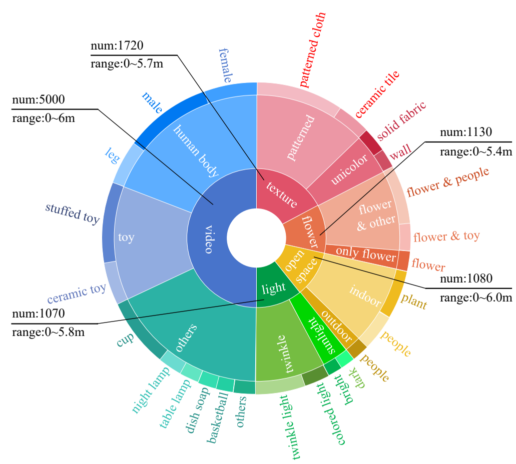

Figure 6. Distribution of different scenarios in our TOFDC.

5. Experiments

5.1. Datasets

TOFDC is collected by the TOF sensor and RGB camera of a Huawei P30 Pro, which covers various scenes such as texture, flower, body, and toy, under different lighting conditions and in open space. It has **10k** 512×384 RGB-D pairs for training and **560** for evaluation. The ground truth depth maps are captured by the Helios TOF camera.

KITTI dataset [38] contains 86k training samples, 1k selected validation samples, and 1k online test samples without ground truths. The depth data is captured by a 64-line LiDAR sensor. Following [20, 37, 51], the RGB-D pairs are bottom center cropped from 1216×352 to 1216×256, as there are no valid LiDAR values near top 100 pixels.

NYUv2 dataset [35] consists of paired RGB-D from 464 indoor scenes, where the depth maps are acquired by Microsoft Kinect. We train our model with 50K samples and test it on the official 654 samples. Following [23, 48, 53, 60], we first downsample the RGB-D pairs from 640×480 to 320×240, and then center crop it to 304×228.

SUN RGBD dataset [36] is selected from several indoor RGB-D datasets [16, 35, 43]. We use 555 samples captured by Kinect V1 and 3,389 samples captured by Asus Xtion camera for cross-dataset evaluation, where we employ the same pre-processing step as that on the NYU2 dataset.

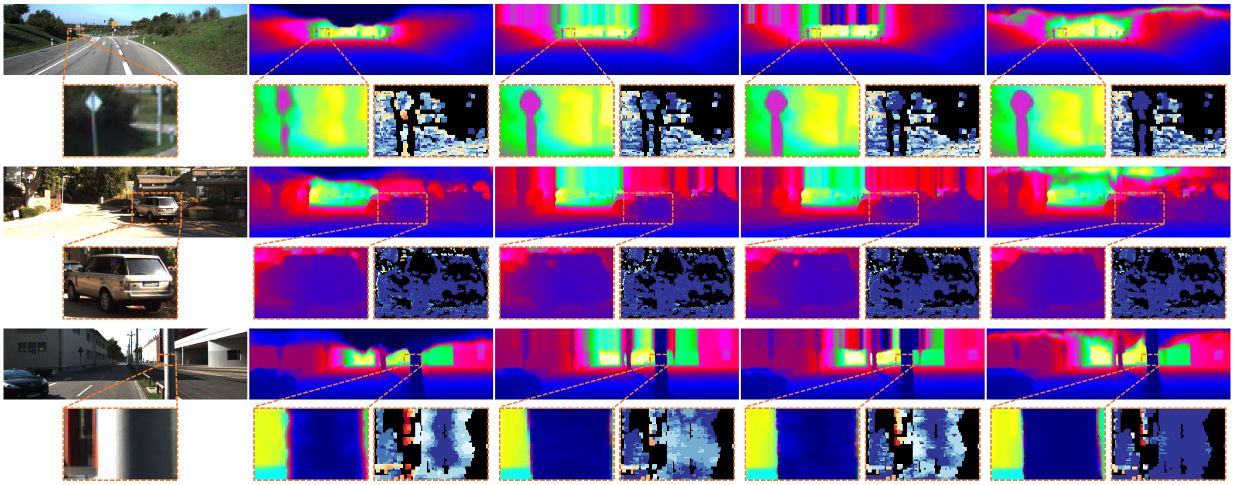

Figure 7. Qualitative results on KITTI depth completion benchmark, including (b) CSPN [5], (c) ACMNet [55], (d) RigNet [48], and (e) our TPVD method. The zoomed-in regions and their corresponding error maps (the darker, the better) show more fine-grained differences.

Method	Params. ↓	FLOPs ↓	Train ↑	Test ↑
ACMNet [55]	**4.9** M	544 G	2.72 FPS	4.20 FPS
BEV@DC [60]	26.9 M	462 G	3.01 FPS	7.87 FPS
TPVD (ours)	31.2 M	**328** G	**3.63** FPS	**8.82** FPS

Table 2. Train & test speed comparison on KITTI validation set.

5.2. Comparison with State-of-the-arts

Outdoor KITTI. We first evaluate the proposed TPVD on KITTI depth completion benchmark that is ranked by RMSE. The top part of Tab. 1 lists the results of 2D based methods while the bottom part reports those of 2D-3D joint approaches. On the whole, TPVD ranks **1st** among all the methods in four evaluation metrics at the time of submission, including RMSE, MAE, iRMSE, and iMAE. For example, TPVD is 15.98 mm superior to the five latest researches on average, *i.e.*, CFormer [54], BEV@DC [60], LRRU [41], PointDC [53], and RigNet++ [50]. Among the 2D-3D joint counterparts, compared with the lightweight FuseNet [3], ACMNet [55], and PointFusion [14], the errors of TPVD are significantly lower, *e.g.*, averagely by 52.59 mm in RMSE and 20.86 mm in MAE. In contrast to those 2D-3D joint methods with similar or larger parameters, TPVD still performs better. Fig. 7 shows the visual comparison with CSPN [5], ACMNet [55], and RigNet [48]. While they produce visually good predictions in general, TPVD can recover more accurate shapes and boundaries. The zoom-in error maps further indicate the superiority.

In addition, Tab. 2 lists the complexity and speed comparison of the 2D-3D joint ACMNet [55], BEV@DC [60], and TPVD. We observe that, despite ACMNet having fewer parameters, its graph model is more complex and requires about twice as many FLOPs as ours. Consequently, ACMNet suffers from slower training and testing speeds. Differently, the LiDAR stream of BEV@DC is removed in testing phase, improving the testing speed from 3.01 FPS to 7.87 FPS. Different from them, our TPV design is computation-friendly though the parameters are slightly higher. The FLOPs is 134 G lower than the second-best BEV@DC, contributing to faster training and testing speeds.

Method	RMSE (m) ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
CSPN [5]	0.117	0.016	99.2	99.9	100.0
FCFRNet [22]	0.106	0.015	99.5	99.9	100.0
GuideNet [37]	0.101	0.015	99.5	99.9	100.0
NLSPN [30]	0.092	0.012	99.6	99.9	100.0
DySPN [19]	0.090	0.012	99.6	99.9	100.0
CFormer [54]	0.091	0.012	99.6	99.9	100.0
RigNet [48]	0.090	0.013	99.6	99.9	100.0
LRRU [41]	0.091	0.011	99.6	99.9	100.0
DLiDAR [31]	0.115	0.022	99.3	99.9	100.0
ACMNet [55]	0.105	0.015	99.4	99.9	100.0
GraphCSPN [25]	0.090	0.012	99.6	99.9	100.0
BEV@DC [60]	0.089	0.012	99.6	99.9	100.0
PointDC [53]	0.089	0.012	99.6	99.9	100.0
TPVD (ours)	**0.086**	**0.010**	**99.7**	**99.9**	**100.0**

Table 3. Quantitative comparison on NYUv2 dataset. The second row shows the results of 2D based methods, whilst the third row illustrates those of 2D-3D joint approaches.

Indoor NYUv2. To verify the effectiveness of TPVD on indoor scenes, following [30, 37, 48], we train TPVD on NYUv2 dataset with 500 sampling depth pixels. As listed in Tab. 3, the top and bottom parts refer to 2D based and 2D-3D joint categories, respectively. We can observe that TPVD still achieve the best performance in all five metrics.

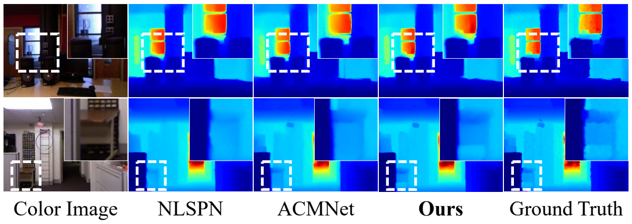

Figure 8. Visual comparison of NLSPN [30], ACMNet [55], and our TPVD method on NYUv2 dataset.

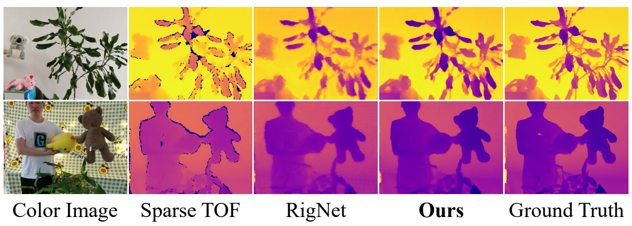

Figure 9. Visual results of RigNet [48] and TPVD on TOFDC.

Method	RMSE (m) ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
CSPN [5]	0.224	0.042	94.5	95.3	96.5
FusionNet [39]	0.116	0.024	98.3	99.4	99.7
GuideNet [37]	0.146	0.030	97.6	98.9	99.5
ENet [12]	0.231	0.061	94.3	95.2	97.4
PENet [12]	0.241	0.043	94.6	95.3	95.5
NLSPN [30]	0.174	0.029	96.4	97.9	98.9
CFormer [54]	0.113	0.029	99.1	99.6	99.9
RigNet [48]	0.133	0.025	97.6	99.1	99.7
GraphCSPN [25]	0.253	0.052	92.0	96.9	98.7
PointDC [53]	0.109	0.021	98.5	99.2	99.6
TPVD (ours)	**0.092**	**0.014**	**99.1**	**99.6**	**99.9**

Table 4. Quantitative comparison on our new TOFDC dataset.

Particularly, compared to previous state-of-the-art methods [41, 53, 54, 60] that are only 1 mm superior in RMSE to concurrent works, our TPVD attains 3 mm improvement again. Meanwhile, the REL is reduced by 20% over the latest 2D-3D joint BEV@DC [60] and PointDC [53]. Fig. 8 shows that TPVD succeeds in restoring detailed structures.

Indoor TOFDC. To further test our TPVD, we implement it on the new TOFDC dataset that is collected by consumptive TOF sensors. As reported in Tab. 4, 2D based and 2D-3D joint methods are divided into the top part and the bottom part, severally. We discover that TPVD outperforms the 2D-3D joint approaches by a large margin. For example, it reduces the RMSE by 15.6% and REL by 33.3% against the second best PointDC [53]. Also, compared with the best 2D based CFormer [54], TPVD is 21 mm superior in RMSE, which is a considerable improvement for indoor scenes. Fig. 9 reveals that TPVD can predict high-quality dense depth results with clearer and sharper structures.

Method	Specialty	RMSE (mm) ↓	MAE (mm) ↓
IP_Basic [18]	params. free	1350.9	305.4
S2D [28]	depth only	985.1	286.5
FusionNet [39]	depth only	995.0	268.0
IR [27]	RGB assisted	**914.7**	297.4
LRRU [41]	depth only	957.4	235.9
TPVD (ours)	depth only	948.6	**231.6**

Table 5. Depth-only comparison on KITTI validation split.

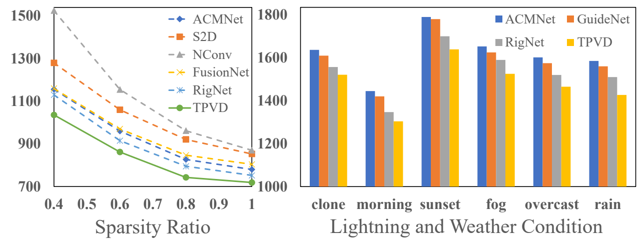

Figure 10. RMSE (mm) comparison under different sparsity ratios on KITTI validation split (left), and diverse lightning and weather conditions on VKITTI dataset [9] (right).

5.3. Generalization Capability

Depth-Only Input. For depth completion task, the auxiliary color images may not always be accessible or dependable, for instance, when the camera malfunctions or when lighting conditions are extremely poor, such as at night. Consequently, we assess our TPVD under a depth-only setting, and compare it with previous methods in Tab. 5. Compared to the depth-only IP_Basic [18], S2D [28], FusionNet [39], and LRRU [41], TPVD achieves the lowest RMSE and MAE, surpassing the second best by 8.8 mm and 4.3 mm, respectively. Furthermore, the MAE of TPVD is significantly superior to that of IR by 65.8 mm though the RMSE is higher. It's noteworthy that TPVD solely takes sparse depth as input, whereas IR uses color images as supervisory signals during training. These analyses indicate that TPVD can work well without image guidance.

Number of Valid Points. We compare the proposed TPVD with five well-known methods with available codes, *i.e.*, S2D [28], NConv [8], FusionNet [39], ACMNet [55], and RigNet [48]. Following [28, 48], we first conduct uniform sampling to produce sparser depth input with ratios (0.4, 0.6, 0.8, 1), where the raw sparsity corresponds to the sampling ratio 1. Then we retrain all the approaches on KITTI and test them on the official validation split. As shown in the left of Fig. 10, our TPVD achieves considerable superiority against other methods under all sparsity ratios. These results demonstrate that the proposed TPVD still can perform well even with complex data input.

Lightning and Weather Condition. KITTI dataset is collected on sunny days [48], whose lightning is almost unchanging and the weather is satisfactory. However, in real-

TPVD	TPV Fusion				GSPN	RMSE (mm)	MAE (mm)
	front	top	side	DASC			
i	✓					763.56	197.82
ii	✓	✓				755.14	194.85
iii	✓	✓	✓			749.38	192.51
iv	✓	✓	✓	✓		735.57	190.26
v	✓	✓	✓	✓	✓	**718.90**	**187.15**

Table 6. Ablation studies of our TPVD on KITTI validation split.

world environments, both factors can be quite complex and pose significant challenges for autonomous driving application. Therefore, we first fine-tune our TPVD (pretrained on KITTI) on "clone" of VKITTI [9] and then test it on the other scenes with various lightning and weather conditions. In Fig. 10 (right), we compare TPVD with GuideNet [37], ACMNet [55], and RigNet [48]. Obviously, our method surpasses the three approaches consistently on morning, sunset, fog, overcast, and rain scenes. It indicates that TPVD can tackle complex lightning and weather conditions.

See Supp. for cross-dataset evaluation on SUN RGBD.

5.4. Ablation Studies

TPVD Designs. Tab. 6 lists the ablation results on KITTI validation split. The baseline model, TPVD-i, solely incorporates the front-view depth. When introducing the top view depth in TPVD-ii, the RMSE decreases from 763.56 mm to 755.15 mm. Building upon TPVD-ii, TPVD-iii integrates the depth of the front, top, and side views, providing comprehensive initial 3D geometry and leading to an improvement of 5.87 mm in RMSE. In TPVD-iv, the application of the proposed DASC further reduces the RMSE by 13.81 mm, marking a significant enhancement. Those improvements in TPVD-ii, iii, and iv over the baseline are primarily attributed to the increased 3D geometric awareness. Lastly, TPVD-v surpasses TPVD-iv by 16.67 mm in RMSE and 3.11 mm in MAE, underscoring the efficacy of GSPN in generating consistent fine-grained geometry through propagation in TPV spaces. In brief, each proposed component contributes positively to the performance of the baseline.

TPV Fusion. The left side of Fig. 11 presents the ablation of TPV Fusion with varying recurrent steps on KITTI validation split. Overall, it can be observed that the error decreases as the recurrent steps increases. For instance, the second step improves upon the first step by approximately 9 mm. However, these limited recurrent steps do not provide sufficient geometric aggregation. Moreover, when the number of steps exceeds 4, the improvement becomes negligible. Consequently, we set the recurrent step to 4 to strike a balance between efficiency and effectiveness.

GSPN. The right side of Fig. 11 ablates GSPN on NYUv2. We find that, (1) a larger number of neighbors leads to lower errors, *e.g.*, the RMSE of 9 neighbors is on average 3.3 mm better than that of 5 neighbors. (2) The performance improves as the iteration increases. When the number is 9 and iteration is 6, GSPN achieves the best result. For efficiency-effectiveness trade-off, we set the neighbor and iteration to 9 and 4, respectively. Fig. 12 shows that with each successive iteration, GSPN progressively produces denser depth with more precise geometry. Furthermore, the receptive fields of the kernels decrease, allowing for a more detailed neighborhood propagation of geometric priors.

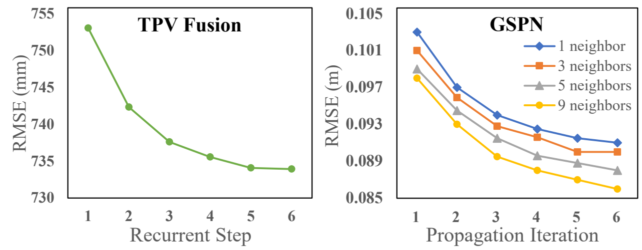

Figure 11. Ablation studies of TPV Fusion on KITTI validation split (left), and GSPN on NYUv2 test set (right).

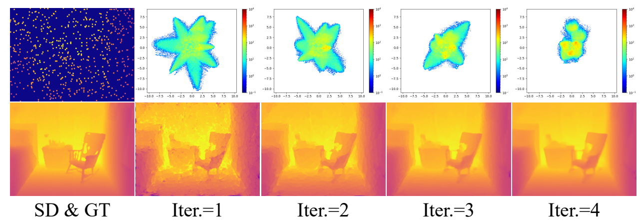

Figure 12. Visual process of GSPN on NYUv2. 1st row: receptive fields of kernels in top-view sparse depth. 2nd row: dense results.

6. Conclusion

In this paper, we proposed the tri-perspective view decomposition (TPVD), a new and novel framework for the 2D depth completion task. It decomposed the raw 3D point cloud into three 2D views to densify sparse measurements, while TPV fusion was designed to learn the 3D geometric priors via recurrent 2D-3D-2D aggregation. In view of the varying LiDAR point distributions, we introduced the distance-aware spherical convolution to refine the geometry in a compact spherical space. Moreover, we presented the geometric spatial propagation network to further improve the geometric consistency. Owing to these designs, TPVD achieves state-of-the-art performance on four benchmarks, including our newly collected dataset, TOFDC.

Acknowledgements

The authors thank the reviewers for their constructive comments. This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX23_0471), and the National Natural Science Fund of China (62361166670, 62072242, and 62376121).

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 2

[2] Dongyue Chen, Tingxuan Huang, Zhimin Song, Shizhuo Deng, and Tong Jia. Agg-net: Attention guided gated-convolutional network for depth image completion. In *ICCV*, pages 8853–8862, 2023. 1

[3] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *ICCV*, pages 10023–10032, 2019. 1, 2, 3, 5, 6

[4] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Liu Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *CoRL*, pages 2148–2161, 2021. 2

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. In *ECCV*, pages 103–119, 2018. 2, 4, 5, 6, 7

[6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2361–2379, 2019. 2

[7] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI*, pages 10615–10622, 2020. 1, 2, 5

[8] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2423–2436, 2020. 2, 5, 7

[9] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, pages 4340–4349, 2016. 7, 8

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 2

[11] Lingzhi He, Hongguang Zhu, Feng Li, Huihui Bai, Runmin Cong, Chunjie Zhang, Chunyu Lin, Meiqin Liu, and Yao Zhao. Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline. In *CVPR*, pages 9229–9238, 2021. 2

[12] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *ICRA*, 2021. 1, 2, 5, 7

[13] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, pages 9223–9232, 2023. 4

[14] Lam Huynh, Phong Nguyen, Jiří Matas, Esa Rahtu, and Janne Heikkilä. Boosting monocular depth estimation with lightweight 3d point fusion. In *ICCV*, pages 12767–12776, 2021. 1, 2, 5, 6

[15] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, pages 2583–2592, 2021. 2, 5

[16] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pages 141–165, 2013. 5

[17] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *3DV*, pages 52–60, 2018. 1

[18] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *CRV*, pages 16–22, 2018. 7

[19] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *AAAI*, pages 1638–1646, 2022. 5, 6

[20] 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. 2, 5

[21] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Single image depth prediction made better: A multivariate gaussian take. In *CVPR*, pages 17346–17356, 2023. 1

[22] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *AAAI*, pages 2136–2144, 2021. 2, 5, 6

[23] Lina Liu, Xibin Song, Jiadai Sun, Xiaoyang Lyu, Lin Li, Yong Liu, and Liangjun Zhang. Mff-net: Towards efficient monocular depth completion with multi-modal feature fusion. *IEEE Robotics and Automation Letters*, 8(2):920–927, 2023. 5

[24] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NeurIPS*, 2017. 2, 4

[25] 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. 1, 2, 5, 6, 7

[26] Xiaoxiao Long, Yuhang Zheng, Yupeng Zheng, Beiwen Tian, Cheng Lin, Lingjie Liu, Hao Zhao, Guyue Zhou, and Wenping Wang. Adaptive surface normal constraint for geometric estimation from monocular images. *arXiv preprint arXiv:2402.05869*, 2024. 1

[27] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *CVPR*, pages 11306–11315, 2020. 7

[28] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, 2019. 1, 5, 7

[29] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A

large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016. 2

[30] 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, 4, 5, 6, 7

[31] 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, 2, 5, 6

[32] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *CVPR*, pages 6250–6259, 2022. 1, 2

[33] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 2

[34] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. Nddepth: Normal-distance assisted monocular depth estimation. In *ICCV*, pages 7931–7940, 2023. 1

[35] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 2, 4, 5

[36] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015. 5

[37] 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. 2, 5, 6, 7, 8

[38] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, pages 11–20, 2017. 1, 2, 5

[39] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *MVA*, pages 1–6, 2019. 7

[40] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark. In *ICCV*, pages 16055–16064, 2021. 1

[41] 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, 2, 5, 6, 7

[42] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2): 1899–1906, 2020. 2

[43] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, pages 1625–1632, 2013. 5

[44] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *ICCV*, pages 2811–2820, 2019. 1, 2, 5

[45] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *ICIP*, pages 913–917. IEEE, 2020. 2

[46] Zhiqiang Yan, Xiang Li, Kun Wang, Zhenyu Zhang, Jun Li, and Jian Yang. Multi-modal masked pre-training for monocular panoramic depth completion. In *ECCV*, pages 378–395, 2022. 1

[47] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Guangyu Li, Jun Li, and Jian Yang. Learning complementary correlations for depth super-resolution with incomplete data in real world. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1

[48] 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, 2, 5, 6, 7, 8

[49] Zhiqiang Yan, Xiang Li, Kun Wang, Shuo Chen, Jun Li, and Jian Yang. Distortion and uncertainty aware loss for panoramic depth completion. In *ICML*, 2023. 1, 4

[50] Zhiqiang Yan, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet++: Efficient repetitive image guided network for depth completion. *arXiv preprint arXiv:2309.00655*, 2023. 1, 2, 5, 6

[51] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Desnet: Decomposed scale-consistent network for unsupervised depth completion. In *AAAI*, pages 3109–3117, 2023. 1, 5

[52] Zhiqiang Yan, Yupeng Zheng, Kun Wang, Xiang Li, Zhenyu Zhang, Shuo Chen, Jun Li, and Jian Yang. Learnable differencing center for nighttime depth perception. *arXiv preprint arXiv:2306.14538*, 2023. 1

[53] 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, 3, 5, 6, 7

[54] 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, 2, 5, 6, 7

[55] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 2021. 1, 2, 5, 6, 7, 8

[56] Zixiang Zhao, Jiangshe Zhang, Shuang Xu, Zudi Lin, and Hanspeter Pfister. Discrete cosine transform network for guided depth map super-resolution. In *CVPR*, pages 5697–5707, 2022. 1

[57] Zixiang Zhao, Jiangshe Zhang, Xiang Gu, Chengli Tan, Shuang Xu, Yulun Zhang, Radu Timofte, and Luc Van Gool. Spherical space feature decomposition for guided depth map super-resolution. In *ICCV*, pages 12547–12558, 2023. 1

[58] Yupeng Zheng, Chengliang Zhong, Pengfei Li, Huan-ang Gao, Yuhang Zheng, Bu Jin, Ling Wang, Hao Zhao, Guyue Zhou, Qichao Zhang, et al. Steps: Joint self-supervised nighttime image enhancement and depth estimation. *arXiv preprint arXiv:2302.01334*, 2023. 1

[59] Yupeng Zheng, Xiang Li, Pengfei Li, Yuhang Zheng, Bu Jin, Chengliang Zhong, Xiaoxiao Long, Hao Zhao, and Qichao

Zhang. Monoocc: Digging into monocular semantic occupancy prediction. *arXiv preprint arXiv:2403.08766*, 2024. 1

[60] 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, 2, 3, 5, 6, 7

[61] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. 4