Tri-Perspective View Decomposition for Geometry Aware Depth Completion and Super-Resolution

Zhiqiang Yan, Kun Wang, Xiang Li, Guangwei Gao, *Senior Member, IEEE*, Jun Li, *Member, IEEE*, and Jian Yang

***Abstract*—Depth completion and super-resolution are crucial tasks for comprehensive RGB-D scene understanding, as they involve reconstructing the precise 3D geometry of a scene from sparse or low-resolution depth measurements. However, most existing methods either rely solely 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. In this paper, we introduce Tri-Perspective View Decomposition (TPVD) frameworks that can explicitly model 3D geometry. To this end, (1) TPVD ingeniously decomposes the original 3D point cloud into three 2D views, one of which corresponds to the sparse or low-resolution depth input. (2) For sufficient geometric interaction, TPV Fusion is designed to update the 2D TPV features through recurrent 2D-3D-2D aggregation. (3) By adaptively searching for TPV affinitive neighbors, two additional refinement heads are developed for these two tasks to further improve the geometric consistency. Meanwhile, we build novel datasets named TOFDC for depth completion and TOFDSR for depth super-resolution. Both datasets are acquired using time-of-flight (TOF) sensors and color cameras on smartphones. Extensive experiments on TOFDC, KITTI, NYUv2, SUN RGBD, VKITTI, TOFDSR, RGB-D-D, Lu, and Middlebury datasets indicate that our TPVD outperforms previous depth completion and super-resolution methods, reaching the state of the art.**

***Index Terms*—Depth completion, depth super-resolution, view decomposition, geometry propagation, TOF RGB-D dataset.**

I. Introduction

OVER the past decade, the rapid development of software and hardware technologies has led to the widespread adoption of RGB-D devices in daily life. Notably, common depth sensors paired with color cameras include Kinect, RealSense, TOF, and LiDAR. These RGB-D data capture systems have extensive applications in the field of computer vision, such as scene understanding [1], [2], [3], [4], [5], [6], [7], [8], [9], 3D reconstruction [10], [11], [12], [13], [14], [15], and autonomous driving [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. All of these applications are highly dependent on accurate and reliable depth predictions. However, due to the inherent constraints of the hardware and challenging environments, depth sensors are unable to provide pixel-wise depth feedback, particularly in outdoor scenarios where the depth density can be as low as 5% . Moreover, the disparity in development between color cameras and depth sensors results in existing color cameras easily capturing ultra-high-definition (e.g., 3648 × 2736) color images, while the corresponding depth sensors are limited to low-resolution (240 × 180) depth maps. Therefore, for realistic applications, it is essential to complete sparse depth data and enhance the resolution, leading to the emergence of depth completion and depth super-resolution tasks.

Depth completion aims to recover dense depth maps from noisy sparse depth measurements. As illustrated in Fig. 1(a), most previous depth completion methods [10], [18], [24], [26], [27], [28], [29], [30], [31] focus on 2D feature space to learn depth representations, leading to a severe lack of 3D geometric information. As an alternative, some recent approaches [21], [32], [33], [34], [35], [36], [37] attempt to incorporate 3D geometric priors directly from raw point clouds, rather than relying only on 2D representations. For example, Zhou et al. [21] and Yu et al. [37] propose extracting point cloud features to incorporate 3D geometry into their 2D depth generation branches. However, as we known that the point clouds in 3D space are extremely sparse and their point distributions are varying in different distances, both of which deeply impede the performance of recent models.

Different from depth completion, the depth source for depth super-resolution is dense but low-resolution. As depicted in Fig. 1(b), mainstream depth super-resolution models [2], [6], [9], [38], [39], [40], [41], [42], [43] pay more attention to 2D feature space to restore geometric structures. For instance, the filtering based series [43], [44], [45] generate filtering kernels from color images to guide the adaptive recovery of depth details. Tang et al. [42] and De et al. [3] present implicit interpolation and graph regularisation for further depth refinement near boundaries and other non-smooth areas. However, modeling the geometry solely from 2D RGB-D pairs in 2D space is insufficient, as it overlooks the richer and more fine-grained geometric priors present in raw 3D point clouds. Notably, all

Received 14 October 2024; revised 31 July 2025; accepted 3 August 2025. Date of publication 6 August 2025; date of current version 5 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant U24A20330, Grant 62361166670, and Grant 62206134. Recommended for acceptance by G. Gallego. *(Corresponding authors: Jun Li; Jian Yang.)*

Zhiqiang Yan, Kun Wang, Jun Li, and Jian Yang are with the PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: yanzq@njust.edu.cn; kunwang@njust.edu.cn; junli@njust.edu.cn; csjyang@njust.edu.cn).

Xiang Li is with the PCA Lab, VCIP, College of Computer Science, Nankai University, Tianjin 300071, China (e-mail: xiang.li.implus@nankai.edu.cn).

Guangwei Gao is with the Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing 210046, China (e-mail: csggao@gmail.com).

Digital Object Identifier 10.1109/TPAMI.2025.3596391

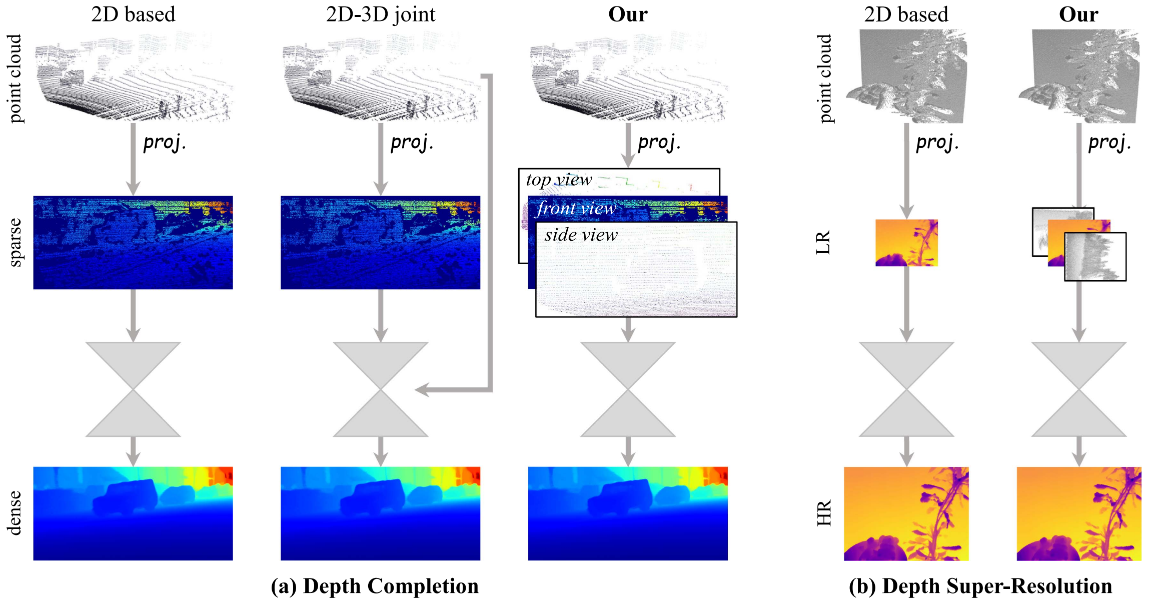

Fig. 1. Framework comparison. Previous 2D methods focus on 2D space to recover dense or high-resolution depth, while recent 2D-3D joint approaches introduce 3D point clouds for assistance. Differently, our TPVD decomposes the 3D point clouds into three 2D views to densify the sparse or enhance the low-resolution input while preserving the 3D geometry. LR: low-resolution, HR: high-resolution.

super-resolution processes involve increasing resolution from low to high, resulting in the depth source with fixed pixels, as well as the point cloud, becoming sparse.

To address the above challenges, we propose novel tri-perspective view decomposition (TPVD) frameworks for depth completion and super-resolution. As shown in Fig. 1(a) and (b), unlike existing 2D based approaches [3], [9], [17], [24], [43], [46] or 2D-3D joint methods [21], [32], [37], our TPVD cleverly decomposes 3D point clouds into three 2D views: top, front, and side. It is worth mentioning that the sparse or low-resolution depth input corresponds exactly to the front-view map. This decomposition enables TPVD to densify sparse or enhance the low-resolution 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.

Furthermore, TPVD incorporates two plug-and-play refinement heads for these two tasks. To be specific, for depth completion, TPVD designs a geometric spatial propagation network (GSPN) for full-scale 3D geometric refinement. Unlike previous 2D SPN [27], [28], [47], [48] and 3D SPN [21], [36] 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. For depth super-resolution, numerous studies have demonstrated the necessity and effectiveness of residual learning. However, we find that the GSPN without any residual connections is not suitable for this task. Instead, TPVD introduces a geometric sparse-pair transform (GSPT), which constructs multi-scale geometric affinity by integrating highly compressed top and side views. It updates the front-view coarse depth through a recurrent residual unit that performs lookups on the cross-view affinity.

In addition, since depth cues play a crucial role in precise 3D reconstruction and human-computer interaction, TOF depth sensors are increasingly deployed on edge mobile devices. This paper introduces a depth completion dataset, termed TOFDC, and a depth super-resolution dataset, termed TOFDSR, collected using a smartphone that has both a TOF lens and a color camera. It is worth noting that TOFDC is the first mobile TOF-based depth completion benchmark, while TOFDSR is the largest real-world super-resolution dataset.

This paper builds upon the initial version of our CVPR 2024 conference paper (Oral presentation) [23]. In this extended version, we expand its application from depth completion to depth super-resolution, both of which are RGB-D based depth perception tasks. Unlike previous approaches that depend solely on 2D depth representations, we introduce 3D point clouds to explicitly model 3D geometry through TPV decomposition, fusion, and GSPT refinement. Additionally, we have developed a novel large-scale dataset named TOFDSR. It is the second real-world depth super-resolution dataset and contains an order of magnitude more data than the first.

In summary, our contributions are listed as follows:

- We introduce novel TPVD frameworks for tasks of depth completion and super-resolution, which are capable of densifying the sparse input or enhancing the resolution whilst retaining rich 3D geometry.

- We propose TPV Fusion to leverage the 3D geometry effectively via recurrent 2D-3D-2D interaction, where DASC is applied to handle the distance-varying distributions of depth points. Besides, two refinement heads, i.e., GSPN and GSPT, are designed to further produce fine-grained 3D geometric structures.
- We build two novel datasets, i.e., TOFDC for depth completion and TOFDSR for depth super-resolution, both making significant contributions to their respective fields.
- Extensive experiments verify the consistent superiority of our method. TPVD surpasses previous state-of-the-art depth completion and super-resolution approaches on nine datasets, i.e., TOFDC, KITTI, NYUv2, SUN RGBD, VKITTI, TOFDSR, RGB-D-D, Lu, and Middlebury.

II. Related Work

A. Depth Completion

1) 2D Based: Usually, the sparse depth is taken from structured light [49], TOF [50], LiDAR [26], stereo cameras [51], or structure from motion [52]. Recent 2D based image-guided methods [29], [31], [53], [54] focus on RGB-D fusion by direct concatenation or summation. Differently, GuideNet [55] adopts a guided filtering, whose kernel weight is from the guided RGB image. FCFRNet [56] designs an energy-based fusion to integrate the RGB-D features. RigNet [17] and RigNet++ [30] propose a new guidance unit with low complexity to produce the dynamic kernel. GFormer [13] and CFormer [18] concurrently leverage convolution and transformer to extract both local and long-range representations. Most recently, LRRU [31] 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 geometry.

2) 2D-3D Joint: It is more intuitive and effective to capture geometric structures with 3D representations, such as surface normals [1], [33], graphs [34], [36], point clouds [32], [35], [37], and voxels [21]. For the first time, DLiDAR [33] and DepthNormal [1] introduce surface normals to boost the performance. In view of the effectiveness of the graph neural networks in representing neighborhood relation, ACMNet [34] applies attention-based graph propagation for multi-modal fusion. GraphCSPN [36] leverages convolution neural networks as well as graph neural networks in a complementary way for geometric learning. Lately, FuseNet [32] and PointDC [37] involve LiDAR point cloud branches to model 3D geometry. Moreover, BEV@DC [21] 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 through point cloud decomposition.

3) Spatial Propagation Network: SPN [27] is increasingly emerging in both 2D based [28], [30], [48] and 2D-3D joint [21], [36] depth completion methods. It digs local or non-local neighbors by 2D and 3D anisotropic filtering kernels. Initially, 2D SPNs [57] are first proposed to learn pairwise similarity matrix. CSPN [58] conducts recursive convolutions with fixed local neighborhood kernels for improvement, while CSPN++ [27] learns adaptive kernel sizes. PENet [29] further enlarges the receptive fields with dilated convolutions. Differently, NLSPN [28] incorporates non-local neighbors via deformable convolutions. Similarly, DySPN [48] produces dynamic non-linear neighbors by attention mechanism. 3D SPNs [59] are commonly embedded in 2D-3D joint methods to utilize 3D geometry. For example, S3CNet [60] computes key spatial features from LiDAR by a 3D spatial propagation unit. GraphCSPN [36] uses geometric constraints to regularize the 3D propagation. Recently, BEV@DC [21] 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 geometry.

B. Depth Super-Resolution

Research on depth super-resolution predominantly focuses on 2D-based approaches. Leveraging the rich structure of color images, guided methods [11], [14], [61], [62], [62], [63], [64], [65], [66], [67], [68] have attracted considerable attention. For example, [69] presents a symmetric uncertainty method to select effective RGB information for high-resolution depth recovery while avoiding detrimental textures. [43] introduces a joint image filtering technique that adaptively determines the neighbors and their weights for each pixel. [70] proposes a multi-modal convolutional sparse coding approach to automatically separate common and private features across different modalities. Similarly, [14] develops a discrete cosine network to extract both shared and specific multi-modal information.

Besides, some methods leverage multi-task learning frameworks to harness complementary knowledge, including depth estimation, depth completion, and depth super-resolution. For instance, [2] introduces an auxiliary depth completion branch to propagate the correlation of dense depth into the depth super-resolution branch. [71] transforms RGB data into a space closer to the depth space via depth estimation, facilitating RGB-D fusion. Furthermore, [72] develops cross-task knowledge distillation to exchange correlations between their depth super-resolution and depth estimation branches.

Most recently, [73] designs a structure flow-guided network to learn edge-focused guidance features for depth structure enhancement. Concurrently, graph regularization [3] and anisotropic diffusion [65] are applied to enhance depth structure recovery. Moreover, [74] significantly improves video depth super-resolution using consistency constraints and direct TOF data. [45] leverages scene priors from large vision models for high-quality depth super-resolution. [68] presents a novel approach that preserves intrinsic phase information, leading to more accurate and detailed depth reconstruction.

Different from these 2D based depth super-resolution methods, our TPVD enhances depth resolution in 2D space while preserving 3D geometric priors via point cloud decomposition.

C. Geometric Decomposition for Depth

Depth information plays a vital role in 2D-to-3D conversion, as it provides essential geometric cues that aid in understanding the spatial structure of a scene. Recently, an increasing number

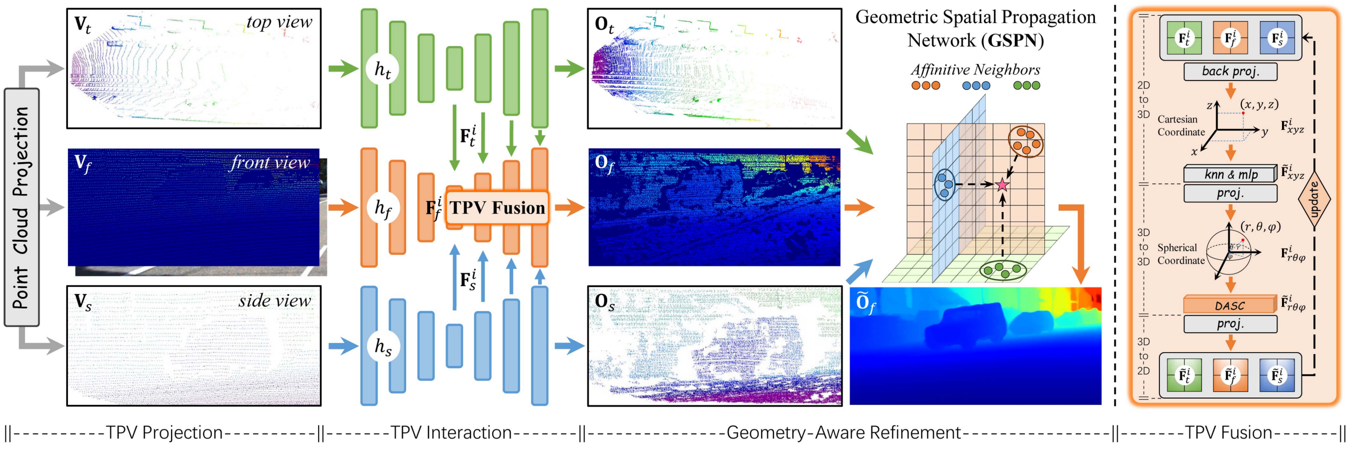

Fig. 2. Pipeline of TPVD for depth completion. The 3D point cloud is first projected into top, side, and front views, where the raw 2D sparse depth input corresponds to the front view. Then the three views are fed into 2D UNets to produce TPV features that are aggregated by the 2D-3D-2D TPV Fusion, obtaining 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.

of studies have focused on decomposing depth to simplify tasks and achieve accurate prediction results. For the first time, DORN [75] discretizes depth and reformulates depth network learning as an ordinal regression problem. Subsequently, several methods [76], [77], [78] are proposed to further enhance DORN by reinterpreting this discretization as bins. Recently, Liu et al. [5] utilizes multivariate Gaussian distribution to model per-pixel scene depth. NDDepth [7] parametrizes plane representation by assuming that 3D scenes are composed of piece-wise planes. TPVFormer [79] decouples 3D voxels into three 2D planes via pooling operations. OccNeRF [80] decouples the 3D occupancy task into 2D depth estimation and 2D semantic segmentation. These innovative approaches have significantly inspired us to advance depth completion and super-resolution using geometric decomposition.

D. Feature Plane Decomposition

Plane-based decomposition has emerged as an efficient paradigm for representing high-dimensional scene information. K-Planes [81] factorize radiance fields into a set of learnable, axis-aligned 2D planes, enabling compact modeling across spatial, temporal, and appearance domains. TK-Planes [82] extend this idea with tiered high-dimensional features to better capture dynamics in UAV scenes. A similar strategy is employed in event-based video reconstruction [83], where interpolated multi-plane features support temporally coherent video synthesis from sparse RGB-event inputs. Although originally developed for rendering and synthesis, these methods inspire our depth completion approach. We adopt a view-decoupled design that projects 3D points onto three orthogonal planes, enabling structured reasoning and efficient RGB-depth fusion under sparse inputs.

III. Methodology

The pipelines of TPVD for depth completion and super-resolution are illustrated in Figs. 2 and 6, respectively. These two pipelines share a very similar architecture, with slight differences in the backbone details and the task-specific refinement modules. Therefore, we use the TPVD for depth completion as an example to elaborate on from Sections III-A to III-E, followed by a description of the different design for depth super-resolution in Section III-F.

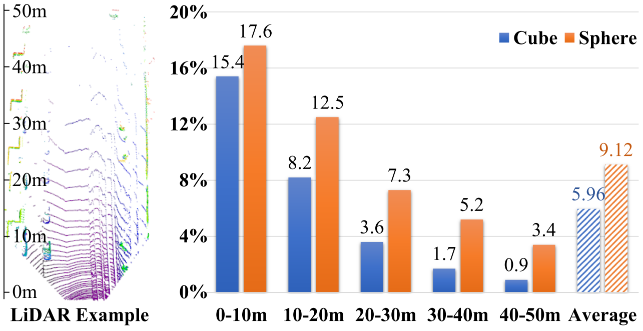

Fig. 3. Percentage of non-empty units across different distances between cubic and our spherical transformations.

A. Network Architecture

Fig. 2 shows our pipeline that consists of TPV projection, TPV interaction, and geometry-aware refinement. Specifically, the 3D point cloud is first projected into top, side, and front sparse depth views. Then three symmetric subnetworks [55], [84] 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, the geometric spatial propagation network (GSPN) further improves the geometric consistency.

B. 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 [32] 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 $\mathbf{V}_f$:

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

Unless stated, we use $\mathbf{V}_f$ to represent $\tilde{\mathbf{V}}_f$ for simplicity below.

C. 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 $1 \leq i \leq 4$:

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

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 $\tilde{\mathbf{F}}_{xyz}^i$:

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

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

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 5%, 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 geometry in the more compact space:

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

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

where $h_{dasc}(\cdot)$ refers to the DASC function (see (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 $\tilde{\mathbf{F}}_{r\theta\varphi}^i$ is projected into 2D space to update the initial 2D $\mathbf{F}_t^i$, $\mathbf{F}_s^i$, and $\mathbf{F}_f^i$ with 2D convolutions h_{2c}, yielding new 2D TPV features:

$$\left(\tilde{\mathbf{F}}_t^i,\ \tilde{\mathbf{F}}_s^i,\ \tilde{\mathbf{F}}_f^i\right) = h_{2c}\left(\mathcal{P}_{tpv}\left(\mathcal{P}_{sph}^{-1}\left(\tilde{\mathbf{F}}_{r\theta\varphi}^i\right)\right)\right). \tag{7}$$

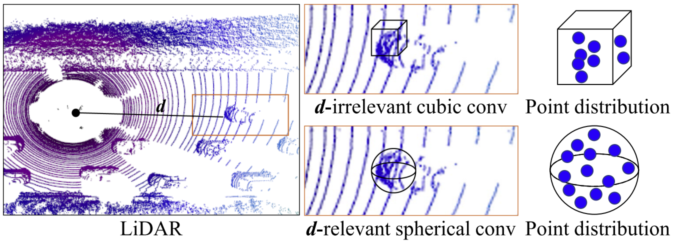

Fig. 4. Comparison of common 3D cubic convolutions and our proposed distance-aware spherical convolutions.

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:

$$\mathbf{O}_t = h_{2c}\left(\tilde{\mathbf{F}}_t^4\right),\ \mathbf{O}_s = h_{2c}\left(\tilde{\mathbf{F}}_s^4\right),\ \mathbf{O}_f = h_{2c}\left(\tilde{\mathbf{F}}_f^4\right). \tag{8}$$

Distance-Aware Spherical Convolution: Given the 3D input $\mathbf{F}_{r\theta\varphi}^i$ in (5), it is sliced by $\mathcal{S}$ into different spherical subareas $\mathbf{A}_{sph} = \{\mathbf{A}_{sph}^1, \ldots, \mathbf{A}_{sph}^j\}$, each with larger volume $|\mathbf{A}_{sph}^j|$ 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 $\mathbf{A}_{cub} = \{\mathbf{A}_{cub}^1, \ldots, \mathbf{A}_{cub}^j\}$ and filtered by h_{3c}, a 3D convolution with kernel $3\times 3\times 3$ and stride 1. Consequently, (6) can be rewritten as:

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

Fig. 4 indicates that the d-relevant DASC involves a higher number of valid points with more balanced distribution.

D. Geometry-Aware Refinement

Geometric Spatial Propagation Network: SPNs [57], [58] 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:

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

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

In Fig. 5, the key of SPNs is how to search for the neighbor set $\mathbf{N}_{f(a,b)}$. In 2D space, CSPN [58] constructs $\mathbf{N}_{f(a,b)}^{CS}$ within a fixed square area excluding the centre pixel, while NLSPN [28]

[1] Equirectangular projection (ERP) utilized in DUL [15]

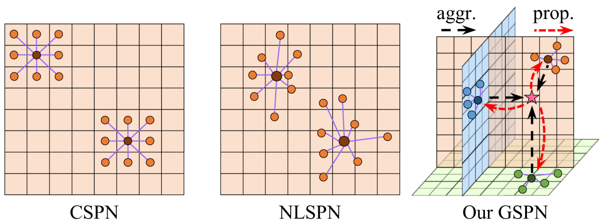

Fig. 5. Comparison of SPNs with different neighbor sets. ‘aggr.’ refers to aggregation while ‘prop.’ indicates propagation.

deforms it in $\mathbb{R}^{H\times W}$ to build $\mathbf{N}^{NL}_{f(a,b)}$:

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

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

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 (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 [79], [85] via projection and MLP. At last, the 3D feature is propagated back to the TPV spaces for refinement:

$$\left(\tilde{\mathbf{O}}^{l+1}_t,\ \tilde{\mathbf{O}}^{l+1}_s,\ \tilde{\mathbf{O}}^{l+1}_f\right) = h_{gspn}\left(\mathbf{O}^{l+1}_t, \mathbf{O}^{l+1}_s, \mathbf{O}^{l+1}_f\right), \tag{13}$$

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

E. Loss Function

The total loss function $\mathcal{L}^{dc}_{total}$ for depth completion consists of three terms, i.e., the front-view $\mathcal{L}_f$, top-view $\mathcal{L}_t$, and side-view $\mathcal{L}_s$. The ground truths of the front, top, and side views are obtained by projecting the annotated point clouds. Following [21], [28], [48], we adopt $\mathcal{L}_1$ and $\mathcal{L}_2$ joint loss functions to denote $\mathcal{L}_f$, $\mathcal{L}_t$, and $\mathcal{L}_s$. For example, $\mathcal{L}_f = \mathcal{L}_1 + \mathcal{L}_2$. As a result, the total loss function $\mathcal{L}^{dc}_{total}$ is defined as:

$$\mathcal{L}^{dc}_{total} = \mathcal{L}_f + \alpha\mathcal{L}_t + \beta\mathcal{L}_s, \tag{14}$$

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

F. Specific Designs for Depth Super-Resolution

Network Architecture: Overall, as presented in Fig. 6, the three subnetworks of depth super-resolution share similar structures with those of depth completion, differing only in their basic units. Due to the sparsity of depth input, most depth completion methods [10], [17], [23], [24], [28] employ UNet-like networks as backbones to gradually densify the sparse data. In contrast, the depth input for depth super-resolution is dense. The mainstream solutions combine simple convolutions and residual techniques, which have proven highly suitable for the depth super-resolution task [9], [11], [14], [43], [50], [62]. Hence, following [9], [45], [69], [87], we adopt the residual group [86] as the basic unit of the subnetworks in Fig. 6, with each basic unit containing two residual groups. Note that the low-resolution depth input is first upsampled to high-resolution using bicubic interpolation, and a residual connection is conducted at the ends of the subnetworks, following popular depth super-resolution approaches. Finally, we propose a geometric sparse-pair transform (GSPT) module for further high-resolution depth refinement.

Geometric Sparse-Pair Transform: As previously discussed, the residual strategy is essential for the performance of depth super-resolution models. However, we observe that SPN techniques [23], [28], [58], including our GSPN, are not particularly suitable without any residuals. Therefore, we develop GSPT as an alternative. Inspired by [88], as shown in Fig. 6, our GSPT performs recurrent residual updates via a series of convolutional gated recurrent units (Conv GRU) [89] and utilizes lookups from the crossing affinity between the top view $\mathbf{O}_t$ and the side view $\mathbf{O}_s$.

Specifically, GSPT first leverages the horizontal strip average pooling $h^r_{hp}(\cdot)$ and the vertical strip average pooling $h^r_{vp}(\cdot)$ [2], [19] to squeeze $\mathbf{O}_s$ and $\mathbf{O}_t$, yielding:

$$\hat{\mathbf{O}}^r_s = h^r_{hp}(\mathbf{O}_s),\ \hat{\mathbf{O}}^r_t = h^r_{vp}(\mathbf{O}_t), \tag{15}$$

where $\hat{\mathbf{O}}^r_s \in \mathbb{R}^{C\times H\times r}$ and $\hat{\mathbf{O}}^r_t \in \mathbb{R}^{C\times r\times W}$, with $r \in \{1,2,4,8\}$ representing the adaptive output size.

Then, GSPT conducts a dot product on the squeezed sparse pairs to construct a lightweight, geometry-aware cross-view affinity pyramid, resulting in:

$$\mathbf{Q} = \left\{\hat{\mathbf{O}}^r_s \cdot \hat{\mathbf{O}}^r_t, | r \in \{1,2,4,8\}\right\}, \tag{16}$$

where $\mathbf{Q} \in \mathbb{R}^{4C\times H\times W}$. Actually, s can be an arbitrary number. By eliminating the pooling step, the complexity of the affinity becomes $(HW)^2$. Although this results in a slight performance sacrifice, our strategy remains more efficient.

Finally, GSPT employs Conv GRUs to gradually refine the coarse front view $\mathbf{O}_f$. Concretely, it performs lookups $h_{lkp}(\cdot)$ on the cross-view affinity pyramid $\mathbf{Q}$, generating the hidden state input $\mathbf{H}$ for the Conv GRUs:

$$\mathbf{H} = h_{lkp}(\mathbf{Q}). \tag{17}$$

For simplicity, we replace the lookup function with convolutions, instead of using the complex indexing method proposed by [88]. The outputs of the Conv GRUs are denoted as:

$$\begin{aligned} \mathbf{G}^1 &= f^1_{gru}\left(\mathbf{O}_f,\ \mathbf{H}^1\right), \\ \mathbf{G}^{l+1} &= f^{l+1}_{gru}\left(\mathbf{G}^l,\ \mathbf{H}^l\right), \end{aligned} \tag{18}$$

where $f_{gru}(\cdot)$ refers to the Conv GRU function. As a result, the refined front-view depth can be described as:

$$\begin{aligned} \tilde{\mathbf{O}}^1_f &= \mathbf{V}_f + \mathbf{G}^1, \\ \tilde{\mathbf{O}}^{l+1}_f &= \tilde{\mathbf{O}}^l_f + \mathbf{G}^l. \end{aligned} \tag{19}$$

Loss Function: The total loss function $\mathcal{L}^{dsr}_{total}$ for depth super-resolution has the same form as (14). But differently, following prior depth super-resolution studies [9], [14], [50], we employ the $\mathcal{L}_1$ loss function for $\mathcal{L}_f$, $\mathcal{L}_t$, and $\mathcal{L}_s$. Consequently, the total loss function $\mathcal{L}^{dsr}_{total}$ is defined as:

$$\mathcal{L}^{dsr}_{total} = \mathcal{L}_f + \lambda\mathcal{L}_t + \mu\mathcal{L}_s, \tag{20}$$

where λ and μ are balanced coefficients, both set to 0.1.

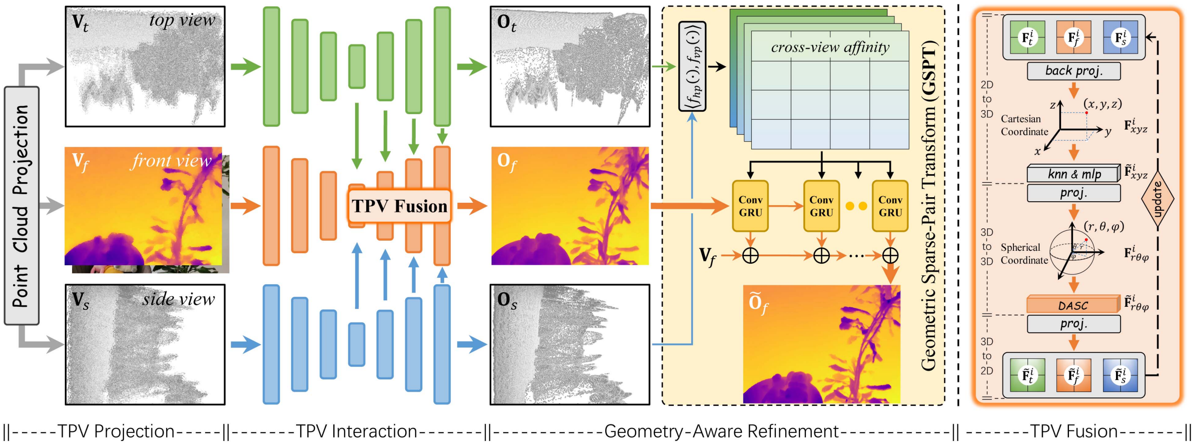

Fig. 6. Pipeline of TPVD for depth super-resolution. Unlike depth completion, each layer of the three subnetworks here consists of two residual groups [86]. In addition, we design a geometric sparse-pair transform (GSPT) specifically for the depth super-resolution task.

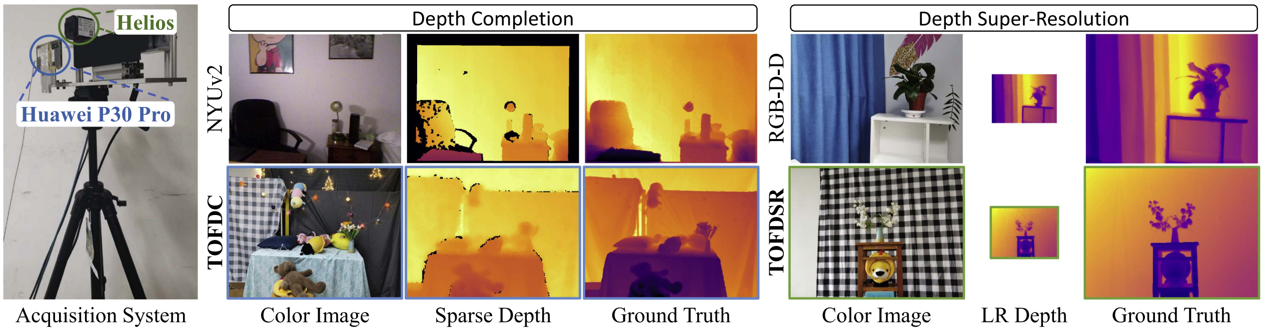

Fig. 7. Acquisition system and data comparison of depth completion and super-resolution. These RGB-D examples are sourced from official releases.

TABLE I
DATASET COMPARISON OF DEPTH COMPLETION AND SUPER-RESOLUTION

Dataset	Year	Outdoor	Indoor	Sensor	Edge Device	Train	Test	Resolution	Real-World
Depth Completion									
KITTI [26]	2017	✓		LiDAR		86,898	1,000	1216 × 352	✓
NYUv2† [49]	2012		✓	Kinect TOF		47,584	654	304 × 228	
TOFDC	2024	✓	✓	phone TOF	✓	10,000	560	512 × 384	✓
Depth Super-Resolution									
Lu [91]	2014		✓	structure light		-	6	640 × 480	
Middlebury [92], [93]	2007		✓	stereo camera		-	30	multi-scale	
NYUv2 [49]	2012		✓	Kinect TOF		1,000	449	640 × 480	
RGB-D-D [50]	2021		✓	phone TOF	✓	2,215	405	512 × 384	✓
TOFDSR	2024	✓	✓	phone TOF	✓	10,000	560	512 × 384	✓

† indicates the post-processed version by [10] that is widely used for the depth completion task. For depth super-resolution, the resolution refers to the size of ground truth depth.

IV. TOFDC AND TOFDSR DATASETS

A. Overview

Fig. 7 illustrates the data acquisition system and data comparison of depth completion and super-resolution. The system comprises a Huawei P30 Pro (for capturing color image and raw depth) and a Helios sensor (for obtaining ground truth depth). We find that the depth data from TOFDC is much denser compared to NYUv2. Fig. 8 reveals that the depth density in NYUv2 predominantly ranges from 60% to 80%, whereas TOFDC exhibits a high concentration between 95% and 100% . Fig. 9 presents the distribution of different scenarios in TOFDC and TOFDSR, encompassing categories such as texture, flower, light, open space, and video. As shown at the bottom of Fig. 8, our TOFDSR has a larger amount of data with many more scenarios. In total, we have collected **10,000** RGB-D pairs from these scenarios. Table I provides a comparative analysis of datasets for depth completion and super-resolution, underscoring that

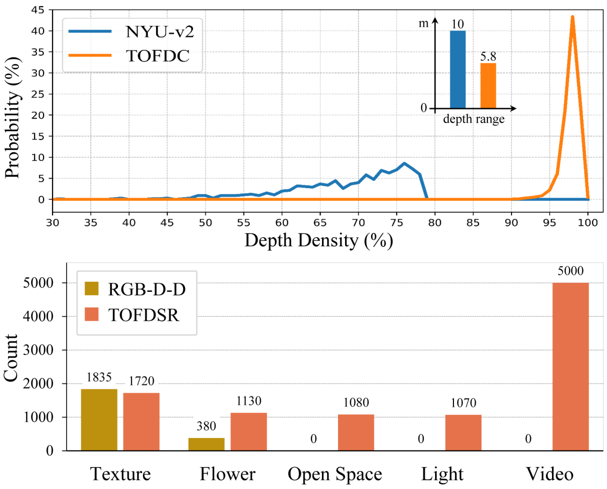

Fig. 8. Comparison of characteristics with widely used RGBD datasets.

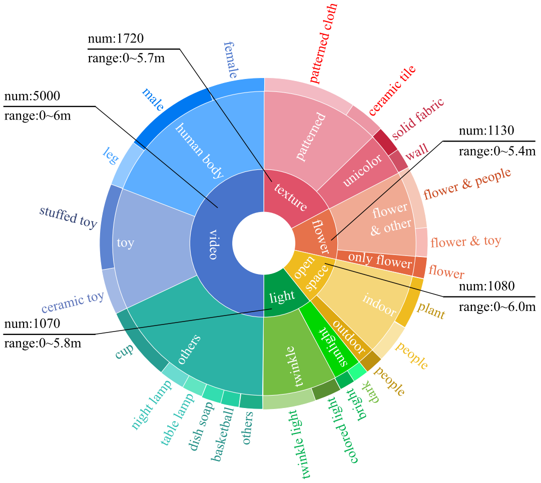

Fig. 9. Distribution of different scenarios in our TOFDC and TOFDSR.

our TOFDC and TOFDSR datasets are valuable contributions to these research domains.

B. Data Collection

As shown in Fig. 7 (left), 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 640 × 480 higher-resolution depth, using 0.3-megapixel Sony DepthSense IMX556 sensor. Their depth acquisition principle is the same, ensuring consistent depth values.

We calibrate the RGB-D system of the P30 with the Helios TOF camera, aligning them on the 640 × 480 color image coordinate using the intrinsic (599.9, 2837.9, 1816, 1394.9) and relative extrinsic parameters. The color images and Helios depth maps are cropped to 512 × 384, while the P30 depth maps to 192 × 144. To yield dense low-resolution depth input for depth super-resolution [50], we employ the colorization technique [90] to fill holes in the P30 depth maps. Additionally, to produce high-resolution sparse depth input for depth completion task, we conduct nearest interpolation to upsample the P30 depth maps to 512 × 384. Finally, the Helios depth maps are also processed using colorization to produce dense ground truth depth maps.

V. Experiments

Section V-A provides a brief introduction to the related datasets. Section V-B describes the metrics and implementation details. Section V-C discusses the performance of TPVD on depth completion, including the comparison with state-of-the-art methods (Section V-C1), generalization capability (Section V-C2), cross-dataset evaluation (Section V-C3), and ablation studies (Section V-C4). Additionally, Section V-D further validates the performance of TPVD on depth super-resolution, including the comparison with previous approaches on real-world and synthetic datasets (Section V-D1) and ablation studies (Section V-D2).

A. Datasets

1) Depth Completion: 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 10 k 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 [26] contains 86 k training samples, 1 k selected validation samples, and 1 k online test samples without ground truths. The depth data is captured by a 64-line LiDAR sensor. Following [19], [48], [55], the RGB-D pairs are bottom center cropped from 1216 × 352 to 1216 × 256, as there are no valid LiDAR values near the top 100pixels.

NYUv2 dataset [49] consists of paired RGB-D from 464 indoor scenes, where the depth maps are acquired by Microsoft Kinect. We train our model with 50 K samples and test it on the official 654 samples. Following [17], [21], [37], [94], we first downsample the RGB-D pairs from 640 × 480 to 320 × 240, and then center crop them to 304 × 228.

SUN RGBD dataset [95] is selected from several indoor RGB-D datasets [49], [96]. 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.

VKITTI [97] is synthesized from KITTI video sequences. It produces color images under various lighting (such as sunset and morning) and weather (such as rain and fog) conditions. Following [55], we use the masks generated from KITTI sparse depth to create sparse samples, closely mimicking real-world sparse depth distribution. Sequences 0001, 0002, 0006, and 0018 are used for training, while sequence 0020 is used for testing. This results in 1,289 frames for fine-tuning and 837 frames for evaluation under each condition.

2) Depth Super-Resolution: TOFDSR is a counterpart to the TOFDC dataset with rich lighting and weak textures. The training set comprises 10 K triples, while the testing split includes 560 triples. Each triple consists of a 512 × 384 color image,

TABLE II
METRIC DEFINITION

For one pixel p in the valid pixel set $\mathbb{P}$ of $\mathbf{y}$:	
– 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	$\sqrt{\frac{1}{\lvert\mathbb{P}\rvert}\sum(\log\mathbf{y}-\log\mathbf{x})^2}$
– $\delta_i \mid i=1,2,3$	$\frac{\lvert\mathbb{S}\rvert}{\lvert\mathbb{P}\rvert}$, $\mathbb{S}:\max(\mathbf{y}_p/\mathbf{x}_p, \mathbf{x}_p/\mathbf{y}_p) < 1.25^i$

x: prediction, y: ground truth.

a 512×384 ground truth depth map, and a 192×144 low-resolution depth map.

RGB-D-D [50] is the first real-world dataset for depth super-resolution. It includes 2,215 RGB-D pairs for training and 405 for testing, where the low-resolution depth is obtained via the TOF camera of a Huawei P30 Pro.

NYUv2 [49] consists of video sequences from various indoor scenes as recorded by the RGB-D cameras from Microsoft Kinect. For depth super-resolution, the dataset provides 1,449 densely labeled pairs (640×480) of aligned RGB and depth images, with 1,000 pairs designated for training and 449 pairs for testing. The low-resolution depth is downsampled from the ground truth using bicubic interpolation [9], [14], [50].

Lu [91] and *Middlebury* [92], [93] are typically used for testing, with models trained on the synthetic NYUv2. Lu has 6 RGB-D pairs with a resolution of 640×480, while Middlebury comprises 30 samples with multi-scale resolutions.

B. Metrics and Implementation Details

On KITTI benchmark, we employ RMSE, MAE, iRMSE, and iMAE for evaluation [17], [21], [28], [31]. On TOFDC, NYUv2, and SUN RGBD datasets, RMSE, REL, and δ_i are used for testing [18], [33], [37], [55]. See Table II for details.

We implement TPVD on Pytorch with four 3090 GPUs. For depth completion, we train it for 50 epochs with Adam [98] optimizer. The initial learning rate is 5×10^{-4} for the first 30 epochs and is reduced to half for every 10 epochs. Following [31], [48], the stochastic depth strategy [99] is used for better training. Also, we employ color jitter and random horizontal flip for data augmentation. For depth super-resolution, the total epoch is 65 and the initial learning rate is set to 1×10^{-4}.

C. Analysis on Depth Completion

1) 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 Table III 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 [18], BEV@DC [21], LRRU [31], PointDC [37], and RigNet++ [30]. Among the 2D-3D joint counterparts, compared with the lightweight FuseNet [32], ACMNet [34], and PointFusion [35], 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. 10 shows the visual comparison with CSPN [58], ACMNet [34], and RigNet [17]. 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, Table IV lists the complexity and speed comparison of the 2D-3D joint ACMNet [34], BEV@DC [21], 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 the 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 are 134 G lower than the second-best BEV@DC, contributing to faster training and testing speeds.

Indoor NYUv2: To verify the effectiveness of TPVD on indoor scenes, following [17], [28], [55], we train TPVD on NYUv2 dataset with 500 sampling depth pixels. As listed in Table V, the top and bottom parts refer to 2D-based and 2D-3D joint categories, respectively. We can observe that TPVD still achieves the best performance in all five metrics. Particularly, compared to previous state-of-the-art methods [18], [21], [31], [37] 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 [21] and PointDC [37]. Fig. 11 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 Table VI, 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 [37]. Also, compared with the best 2D based CFormer [18], TPVD is 21 mm superior in RMSE, which is a considerable improvement for indoor scenes. Fig. 12 reveals that TPVD can predict high-quality dense depth results with clearer and sharper structures.

2) Generalization Capability: Depth-Only Input: For depth completion task, the auxiliary color images may not always be accessible or dependable. For example, 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

TABLE III
QUANTITATIVE RESULTS ON KITTI ONLINE DEPTH COMPLETION LEADERBOARD

Method	2D	3D	Params. (M) ↓	RMSE (mm) ↓	MAE (mm) ↓	iRMSE (1/km) ↓	iMAE (1/km) ↓	Publication
CSPN [58]	✓		17.4	1019.64	279.46	2.93	1.15	ECCV 2018
S2D [10]	✓		26.1	814.73	249.95	2.80	1.21	ICRA 2019
NConv [53]	✓		**0.36**	829.98	233.26	2.60	1.03	PAMI 2020
CSPN++ [27]	✓		26.0	743.69	209.28	2.07	0.90	AAAI 2020
NLSPN [28]	✓		25.8	741.68	199.59	1.99	0.84	ECCV 2020
GuideNet [55]	✓		73.5	736.24	218.83	2.25	0.99	TIP 2020
TWISE [54]	✓		1.45	840.20	195.58	2.08	0.82	CVPR 2021
FCFRNet [56]	✓		50.6	735.81	217.15	2.20	0.98	AAAI 2021
PENet [29]	✓		131.5	730.08	210.55	2.17	0.94	ICRA 2021
DySPN [100]	✓		26.3	709.12	192.71	1.88	0.82	AAAI 2022
RigNet [17]	✓		65.2	712.66	203.25	2.08	0.90	ECCV 2022
CFormer [18]	✓		83.5	708.87	203.45	2.01	0.88	CVPR 2023
RigNet++ [30]	✓		19.9	710.85	202.45	2.01	0.89	arXiv 2023
LRRU [31]	✓		21.0	696.51	189.96	1.87	**0.81**	ICCV 2023
DepthNormal [1]	✓	✓	∼ 40	777.05	235.17	2.42	1.13	ICCV 2019
FuseNet* [32]	✓	✓	1.9	752.88	221.19	2.34	1.14	ICCV 2019
DLiDAR* [33]	✓	✓	53.4	758.38	226.50	2.56	1.15	CVPR 2019
ACMNet [34]	✓	✓	4.9	744.91	206.09	2.08	0.90	TIP 2021
PointFusion [35]	✓	✓	8.7	741.90	201.10	1.97	0.85	ICCV 2021
GraphCSPN [36]	✓	✓	26.4	738.41	199.31	1.96	0.84	ECCV 2022
BEV@DC [21]	✓	✓	30.8	697.44	189.44	1.83	0.82	CVPR 2023
PointDC [37]	✓	✓	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

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 results are highlighted.

TABLE IV
SPEED COMPARISON ON KITTI VALIDATION SET

Method	Params. ↓	FLOPs ↓	Train ↑	Test ↑
ACMNet [34]	**4.9** M	544 G	2.72 FPS	4.20 FPS
BEV@DC [21]	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 V
QUANTITATIVE COMPARISON ON NYUV2 DATASET

Method	RMSE (m) ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
CSPN [58]	0.117	0.016	99.2	99.9	100.0
FCFRNet [56]	0.106	0.015	99.5	99.9	100.0
GuideNet [55]	0.101	0.015	99.5	99.9	100.0
NLSPN [28]	0.092	0.012	99.6	99.9	100.0
DySPN [100]	0.090	0.012	99.6	99.9	100.0
CFormer [18]	0.091	0.012	99.6	99.9	100.0
RigNet [17]	0.090	0.013	99.6	99.9	100.0
LRRU [31]	0.091	0.011	99.6	99.9	100.0
DLiDAR [33]	0.115	0.022	99.3	99.9	100.0
ACMNet [34]	0.105	0.015	99.4	99.9	100.0
GraphCSPN [36]	0.090	0.012	99.6	99.9	100.0
BEV@DC [21]	0.089	0.012	99.6	99.9	100.0
PointDC [37]	0.089	0.012	99.6	99.9	100.0
TPVD (ours)	**0.086**	**0.010**	**99.7**	**99.9**	**100.0**

The second row shows the results of 2D based methods, whilst the third row illustrates those of 2D-3D joint approaches.

TABLE VI
QUANTITATIVE COMPARISON ON OUR NEW TOFDC DATASET

Method	RMSE (m) ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
CSPN [58]	0.224	0.042	94.5	95.3	96.5
FusionNet [101]	0.116	0.024	98.3	99.4	99.7
GuideNet [55]	0.146	0.030	97.6	98.9	99.5
ENet [29]	0.231	0.061	94.3	95.2	97.4
PENet [29]	0.241	0.043	94.6	95.3	95.5
NLSPN [28]	0.174	0.029	96.4	97.9	98.9
CFormer [18]	0.113	0.029	99.1	99.6	99.9
RigNet [17]	0.133	0.025	97.6	99.1	99.7
GraphCSPN [36]	0.253	0.052	92.0	96.9	98.7
PointDC [37]	0.109	0.021	98.5	99.2	99.6
TPVD (ours)	**0.092**	**0.014**	**99.1**	**99.6**	**99.9**

TABLE VII
DEPTH-ONLY COMPARISON ON KITTI VALIDATION SPLIT

Method	Specialty	RMSE (mm) ↓	MAE (mm) ↓
IP_Basic [102]	params. free	1350.9	305.4
S2D [10]	depth only	985.1	286.5
FusionNet [101]	depth only	995.0	268.0
IR [103]	RGB assisted	**914.7**	297.4
LRRU [31]	depth only	957.4	235.9
TPVD (ours)	depth only	948.6	**231.6**

in Table VII. Compared to the depth-only IP_Basic [102], S2D [10], FusionNet [101], and LRRU [31], 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 the training process. These analyses indicate that the proposed 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 [10],

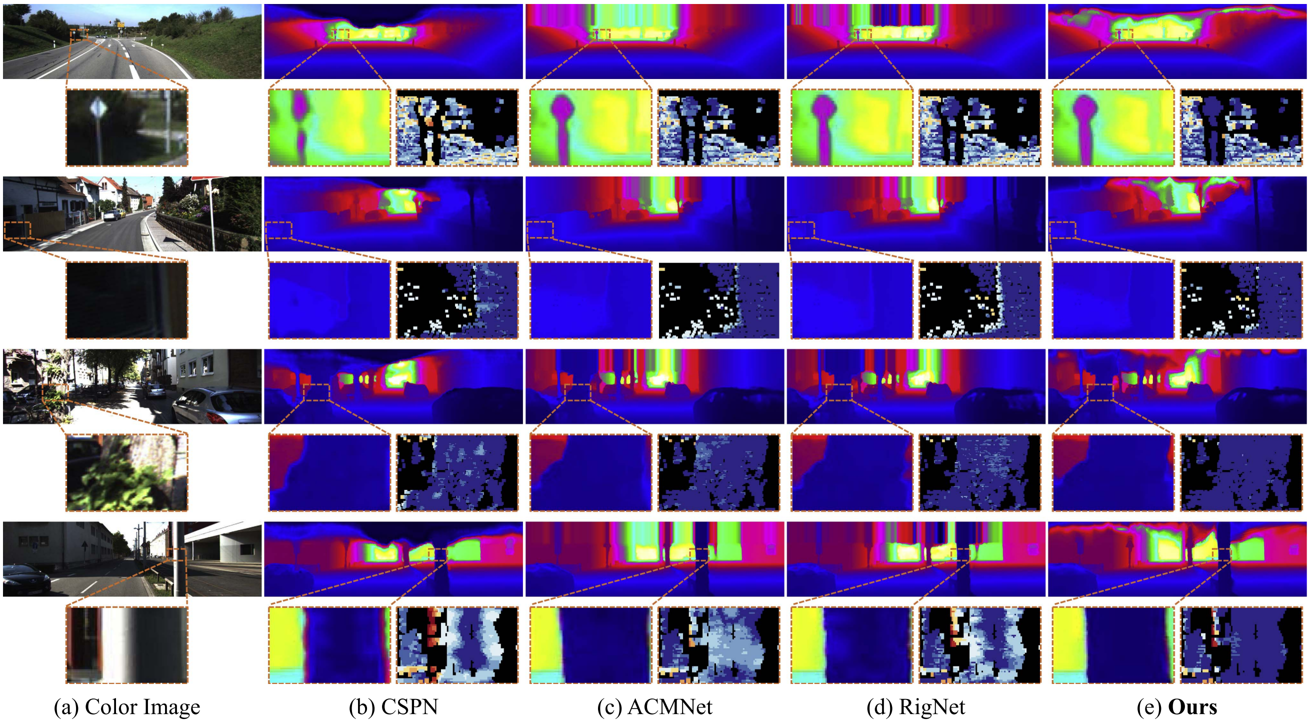

Fig. 10. Qualitative results on KITTI depth completion benchmark. The methods include (b) CSPN [58], (c) ACMNet [34], (d) RigNet [17], and (e) our TPVD method. The zoomed-in regions and their corresponding error maps (the darker, the better) show more fine-grained differences.

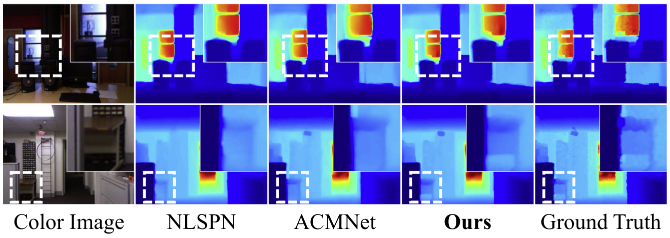

Fig. 11. Visual comparison on NYUv2 dataset.

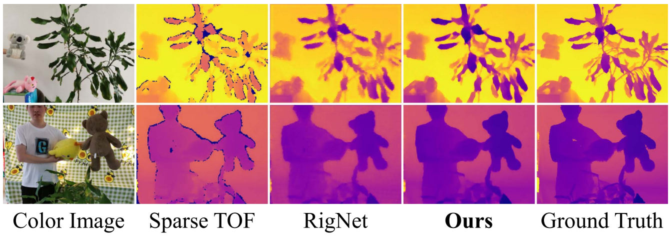

Fig. 12. Visual comparison on TOFDC dataset.

NConv [53], FusionNet [101], ACMNet [34], and RigNet [17]. Following [10], [17], 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 on the left of Fig. 13, our TPVD achieves considerable superiority over other methods under all sparsity ratios. These results demonstrate that the proposed TPVD still can perform well even with complex data input.

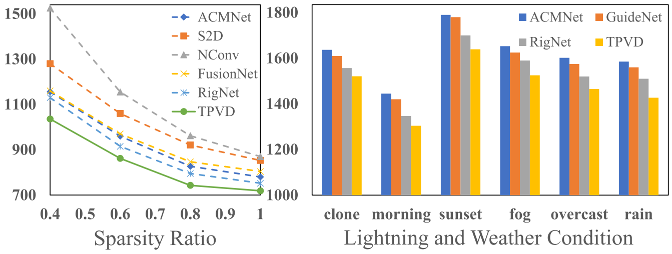

Fig. 13. Comparison of generalization capability. Left: performance under different sparsity ratios on KITTI validation split. Right: performance under diverse lightning and weather conditions on VKITTI. The metric is RMSE.

Lighting and Weather Condition: KITTI dataset is collected on sunny days [17], whose lighting is almost unchanging and the weather is satisfactory. However, in real-world environments, both factors can be quite complex and pose significant challenges for autonomous driving applications. Therefore, we first fine-tune our TPVD (pretrained on KITTI) on the "clone" of VKITTI [97] and then test it on the other scenes with various lighting and weather conditions. In Fig. 13 (right), we compare TPVD with GuideNet [55], ACMNet [34], and RigNet [17]. Obviously, our method surpasses the three approaches consistently in morning, sunset, fog, overcast, and rain scenes. It indicates that TPVD can tackle complex lighting and weather conditions.

3) Cross-Dataset Evaluation: To validate the generalization on indoor scenes [37], we train TPVD on NYUv2 and test it on SUN RGBD. Comparing Table VIII-Kinect with Table V, the

TABLE VIII
Cross-Dataset Evaluation on SUN RGBD Benchmark

Method	RMSE (m) ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Collected by Kinect V1					
CSPN [58]	0.729	0.504	69.1	77.8	84.0
NLSPN [28]	0.093	0.020	98.9	99.6	99.7
CostDCNet [104]	0.119	0.033	98.1	99.6	99.7
GraphCSPN [36]	0.094	<u>0.023</u>	98.8	99.6	99.7
PointDC [37]	<u>0.092</u>	<u>0.023</u>	98.9	99.6	99.8
TPVD (ours)	**0.087**	**0.022**	**99.1**	**99.7**	**99.8**
Collected by Xtion					
CSPN [58]	0.490	0.179	84.5	91.5	95.1
NLSPN [28]	<u>0.128</u>	0.015	99.0	99.7	99.9
CostDCNet [104]	0.207	0.028	97.8	99.1	99.5
GraphCSPN [36]	0.131	0.017	99.0	99.7	99.9
PointDC [37]	<u>0.128</u>	<u>0.016</u>	99.1	99.7	99.9
TPVD (ours)	**0.119**	**0.014**	**99.3**	**99.8**	**99.9**

TABLE IX
Ablation Studies of Our TPVD on KITTI Validation Split

TPVD (DC)	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**

errors of all methods increase and the accuracy decreases due to different RGB-D sensors. When comparing Table VIII-Xtion with Table V, since the data is from different Xtion devices, we discover that the performance drops by large margins. However, Table VIII reports that 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 [28] and PointDC [37]. These facts evidence the powerful cross-dataset generalization ability of TPVD.

4) Ablation Studies: TPVD Designs: Table IX 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 baseline.

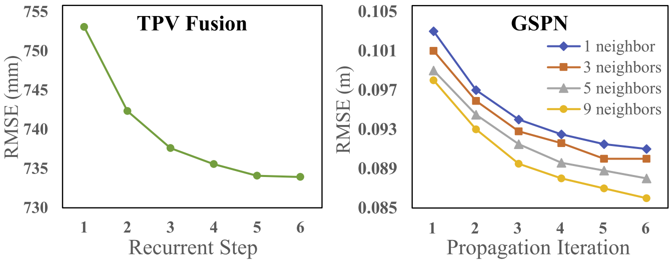

Fig. 14. Ablation studies of TPV Fusion (KITTI) and GSPN (NYUv2).

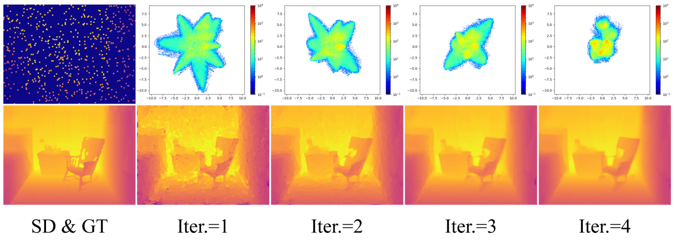

Fig. 15. Visual process of GSPN on NYUv2. 1st row: receptive fields of kernels in top-view sparse depth. 2nd row: dense results.

TPV Fusion: The left side of Fig. 14 presents the ablation of the TPV Fusion with varying recurrent steps on KITTI validation split. Overall, it can be observed that the error decreases as the recurrent step 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. 14 ablates the 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. 15 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.

D. Analysis on Depth Super-Resolution

1) Comparison With State-of-the-Arts: Synthetic: Table X lists the depth super-resolution results on synthetic datasets, where the low-resolution depth data is generated from ground truth using bicubic interpolation [2], [9], [14], [50]. As can be seen, TPVD achieves superior or competitive performance compared to other methods across four benchmarks under ×4, ×8, and ×16 settings. For instance, TPVD surpasses the second-best SGNet [9] by an average of 8.7% on the NYUv2 dataset, which is a notable improvement for the RGB-D based indoor scene understanding task. Additionally, among larger models with more than

TABLE X
QUANTITATIVE COMPARISON ON SYNTHETIC DEPTH SUPER-RESOLUTION DATASETS

Method	Params. (M) ↓	NYUv2			RGB-D-D			Lu			Middlebury			Publication
		×4	×8	×16	×4	×8	×16	×4	×8	×16	×4	×8	×16	
DJF [38]	**0.08**	2.80	5.33	9.46	3.41	5.57	8.15	1.65	3.96	6.75	1.68	3.24	5.62	ECCV 2016
DJFR [41]	**0.08**	2.38	4.94	9.18	3.35	5.57	7.99	1.15	3.57	6.77	1.32	3.19	5.57	PAMI 2019
PAC [105]	-	1.89	3.33	6.78	1.25	1.98	3.49	1.20	2.33	5.19	1.32	2.62	4.58	CVPR 2019
CUNet [70]	0.21	1.92	3.70	6.78	1.18	1.95	3.45	0.91	2.23	4.99	1.10	2.17	4.33	PAMI 2020
DSRNet [40]	-	3.00	5.16	8.41	-	-	-	1.77	3.10	5.11	1.77	3.05	4.96	TIP 2018
DKN [43]	1.16	1.62	3.26	6.51	1.30	1.96	3.42	0.96	2.16	5.11	1.23	2.12	4.24	IJCV 2021
FDKN [43]	0.69	1.86	3.58	6.96	1.18	1.91	3.41	0.82	2.10	5.05	1.08	2.17	4.50	IJCV 2021
FDSR [50]	0.6	1.61	3.18	5.86	1.16	1.82	3.06	1.29	2.19	5.00	1.13	2.08	4.39	CVPR 2021
DAGF [44]	2.44	1.36	2.87	6.06	1.14	1.76	2.82	0.83	1.93	4.80	1.15	1.80	3.70	TNNLS 2023
GraphSR [3]	32.53	1.79	3.17	6.02	1.30	1.83	3.12	0.92	2.05	5.15	1.11	2.12	4.43	CVPR 2022
SUFT [69]	22.01	1.12	2.51	4.86	1.10	1.69	2.71	1.10	1.74	3.92	1.07	1.75	3.18	MM 2022
DCTNet [14]	0.48	1.59	3.16	5.84	1.08	1.74	3.05	0.88	1.85	4.39	1.10	2.05	4.19	CVPR 2022
RSAG [106]	11.99	1.23	2.51	5.27	1.14	1.75	2.96	**0.79**	1.67	4.30	1.13	2.74	3.55	AAAI 2023
DADA [65]	32.53	1.54	2.74	4.80	1.20	1.83	2.80	0.96	1.87	4.01	1.20	2.03	4.18	CVPR 2023
SSDNet [6]	-	1.60	3.14	5.86	**1.04**	1.72	2.92	0.80	1.82	4.77	**1.02**	1.91	4.02	ICCV 2023
SGNet [9]	35.42	1.10	2.44	4.77	1.10	1.64	2.55	1.03	1.61	3.55	1.15	1.64	2.95	AAAI 2024
TPVD (ours)	4.6	**1.08**	**2.35**	**4.62**	1.06	**1.62**	**2.52**	0.82	**1.58**	**3.29**	**1.02**	**1.61**	**2.88**	-

The evaluation metric is RMSE (centimeter).

TABLE XI
QUANTITATIVE COMPARISON ON REAL-WORLD DEPTH SUPER-RESOLUTION DATASETS

Train Set	Test Set	DJF [38]	DJFR [41]	DKN [43]	FDSR [50]	DCTNet [14]	SUFT [69]	SSDNet [6]	SGNet [9]	**TPVD (ours)**
NYUv2	RGB-D-D	7.90	8.01	7.38	7.50	7.37	7.22	7.32	7.22	**7.19**
RGB-D-D	RGB-D-D	5.54	5.52	5.08	5.49	5.43	5.41	5.38	5.32	**4.28**
NYUv2	TOFDSR	7.20	7.26	7.30	7.46	7.16	7.11	-	7.10	**7.08**
TOFDSR	TOFDSR	5.84	5.72	5.50	5.03	5.16	4.37	-	4.33	**4.25**

30 M parameters, TPVD demonstrates average improvements of 16.1%, 9.8%, 16.1%, and 16.6% on NYUv2, RGB-D-D, Lu, and Middlebury, respectively. Furthermore, Fig. 16 provides a visual comparison, illustrating that TPVD can recover more accurate depth results with sharper structures.

Real-World: Table XI presents the depth super-resolution results on real-world depth super-resolution datasets, where the low-resolution and ground truth depth maps are captured by two different sensors. In the third and fifth rows of Table XI, we train and test these methods using real-world data. The results indicate that TPVD significantly outperforms other methods, surpassing the second-best by 15.7% on RGB-D-D and 2.3% on TOFDSR. However, as presented in the second and fourth rows of Table XI, when evaluating models pretrained on NYUv2 on the two real-world datasets, we observe a performance drop across all approaches due to the domain gap between synthetic and real-world data. Despite this, our approach still achieves the best results among them. Fig. 16 further demonstrates the visual superiority of the TPVD.

2) Ablation Studies: Table XII reports the comprehensive ablation studies of TPVD on the real-world depth super-resolution TOFDSR dataset. The findings clearly indicate that the TPV Fusion design significantly enhances model performance. Specifically, the inclusion of auxiliary top-view and side-view depth maps leads to a gradual reduction in RMSE from 4.60 cm to 4.48 cm. Additionally, the incorporation of the DASC module further benefits the model. Moreover, embedding the GSPT module results in an additional error reduction of 0.18 cm, thereby confirming the efficacy of GSPT in geometric refinement. Each component contributes positively to the baseline, collectively demonstrating their substantial impact on significantly improving model accuracy and robustness.

TABLE XII
ABLATION STUDIES OF OUR TPVD ON TOFDSR DATASET

TPVD (DSR)	TPV Fusion				GSPT	Swin-T	Params. (M)	FLOPs (G)	RMSE (cm)
	front	top	side	DASC					
i	✓						0.9	95	4.60
ii	✓	✓					2.1	208	4.53
iii	✓	✓	✓				3.5	312	4.48
iv	✓	✓	✓	✓			4.1	364	4.43
v	✓	✓	✓	✓	✓		4.6	397	**4.23**
vi	✓	✓	✓	✓	✓	✓	5.8	182	4.36

TPVD-Vi replaces the resnet backbone of TPVD-V with swin-tiny [107].

Finally, to ablate the backbone, we replace the encoder composed of residual groups [86] with a Swin-Tiny [107] variant, where each stage is configured with 48 channels, consistent with the original residual-based encoder to ensure a fair comparison. This modification increases the parameter count by 1.2 M, reduces the FLOPs by half, but results in a performance drop of 1.3 cm. These results suggest that while the Swin-based encoder is more efficient in terms of computation, it may sacrifice some accuracy under comparable architectural settings.

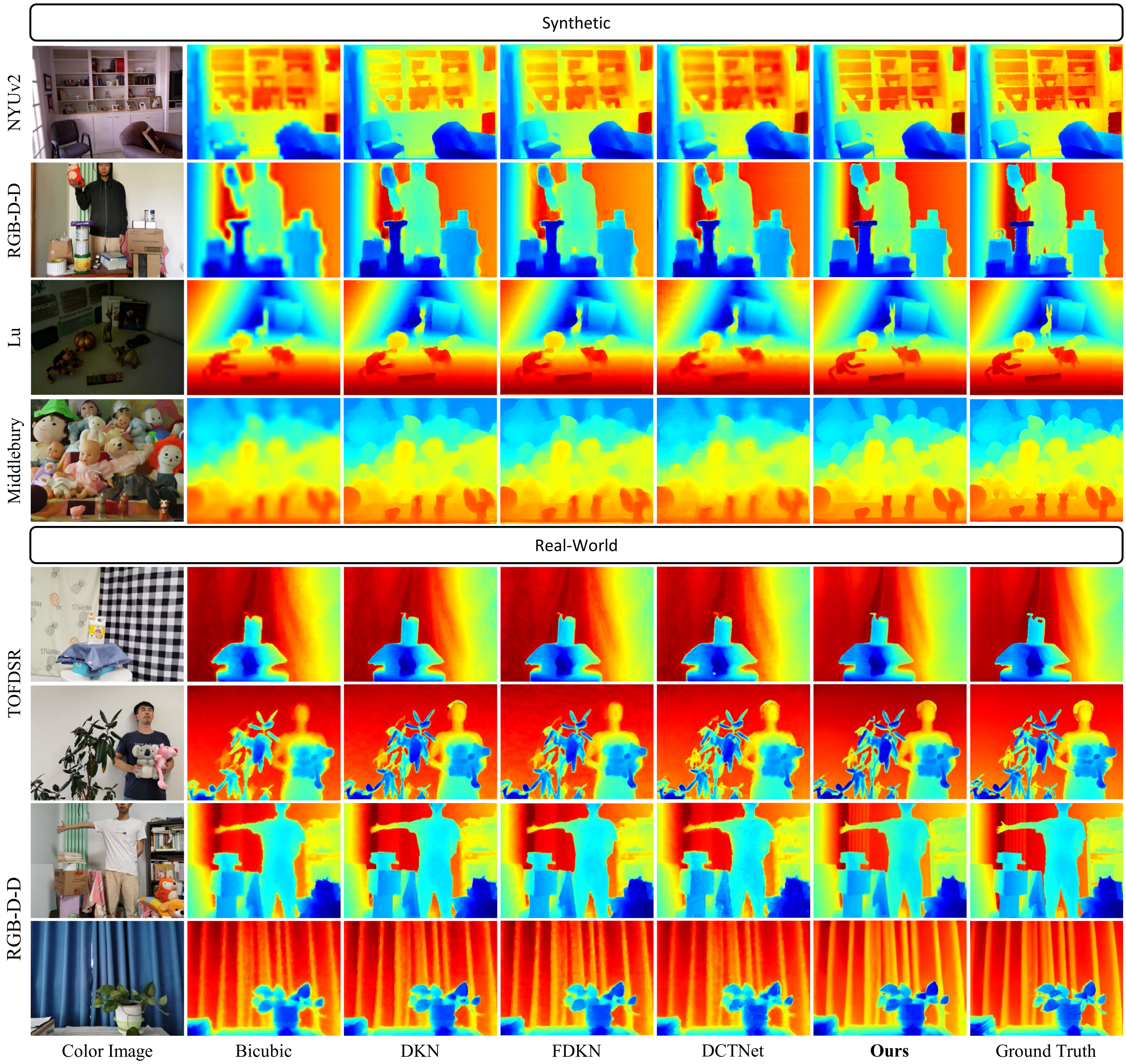

Fig. 16. Quantitative results on four synthetic and two real-world depth super-resolution datasets.

E. Failure Case

As shown in Fig. 17, our method has difficulty accurately estimating depth near transparent objects such as car windows and glass doors. It tends to predict the depth of the background behind the glass, rather than the transparent surface itself.

The main cause of this failure lies in the limitations of both input data and supervision. For depth completion, sparse and ground truth depths usually lack valid values in transparent regions. For depth super-resolution, low-resolution and ground truth depths often contain incorrect values corresponding to the background. Moreover, color images can be misleading, as they tend to reflect background content rather than the transparent surface itself. Due to these quality deficiencies in the training data around transparent objects, our view-decoupled approach fails to handle them effectively.

A potential solution could be to integrate transparent object detection/segmentation technologies [108], [109] to generate masks, which can then guide and compensate the depth predictions using the surrounding valid depth values.

VI. Conclusion

In this paper, we propose novel tri-perspective view decomposition (TPVD) frameworks for the tasks of depth completion and

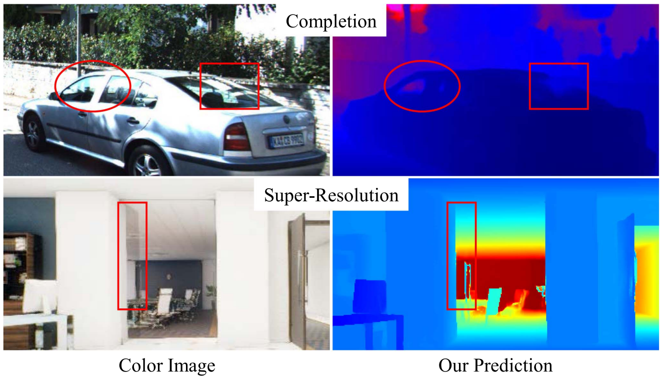

Fig. 17. Failure cases in depth completion and super-resolution around transparent objects, such as car windows and glass doors.

super-resolution. The core concept involves decomposing the raw 3D point cloud into three 2D views to densify sparse depth or enhance low-resolution measurements. We design the TPV fusion to learn 3D geometric priors through recurrent 2D-3D-2D aggregation, incorporating distance-aware spherical convolution to refine geometry within a compact spherical space. Additionally, for these two tasks, we introduce the geometric spatial propagation network and the geometric sparse-pair transform to further enhance the geometric consistency. Owing to these designs, TPVD outperforms previous depth completion and super-resolution methods across nine benchmarks, including our newly collected TOFDC and TOFDSR.

Limitation and future work: (i) Although our method achieves state-of-the-art performance across various datasets, it suffers from relatively high computational complexity. This is mainly due to the three-branch architecture introduced by the view-decoupling strategy and the design of the recurrent fusion modules. (ii) Our method struggles to recover accurate depth values in scenes containing special materials, such as transparent glass. Consequently, two key improvements are planned for future work: (i) Investigating more lightweight backbone architectures and simplifying the recurrent fusion modules to obtain better results with fewer iterations. (ii) Incorporating segmentation masks for special materials to enhance depth estimation accuracy in these challenging regions. Moreover, we believe that event-driven methods [83], [110], [111], [112], [113], [114] hold great potential to advance multiple facets of multi-modal monocular depth perception in the future.

References

[1] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, "Depth completion from sparse LiDAR data with depth-normal constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2811–2820.

[2] Z. Yan et al., "Learning complementary correlations for depth super-resolution with incomplete data in real world," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 5616–5626, Apr. 2024.

[3] R. De Lutio, A. Becker, S. D'Aronco, S. Russo, J. D. Wegner, and K. Schindler, "Learning graph regularisation for guided super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1979–1988.

[4] H. Wang, M. Yang, X. Lan, C. Zhu, and N. Zheng, "Depth map recovery based on a unified depth boundary distortion model," *IEEE Trans. Image Process.*, vol. 31, pp. 7020–7035, 2022.

[5] C. Liu, S. Kumar, S. Gu, R. Timofte, and L. Van Gool, "Single image depth prediction made better: A multivariate gaussian take," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17346–17356.

[6] Z. Zhao et al., "Spherical space feature decomposition for guided depth map super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 12547–12558.

[7] S. Shao, Z. Pei, W. Chen, X. Wu, and Z. Li, "NDDepth: Normal-distance assisted monocular depth estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 7931–7940.

[8] H. Wang, M. Yang, C. Zhu, and N. Zheng, "RGB-guided depth map recovery by two-stage coarse-to-fine dense CRF models," *IEEE Trans. Image Process.*, vol. 32, pp. 1315–1328, 2023.

[9] Z. Wang, Z. Yan, and J. Yang, "SGNet: Structure guided network via gradient-frequency awareness for depth map super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 5823–5831.

[10] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3288–3295.

[11] X. Song et al., "Channel attention based iterative residual learning for depth map super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5631–5640.

[12] Z. Yan, X. Li, K. Wang, Z. Zhang, J. Li, and J. Yang, "Multi-modal masked pre-training for monocular panoramic depth completion," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 378–395.

[13] K. Rho, J. Ha, and Y. Kim, "GuideFormer: Transformers for image guided depth completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6250–6259.

[14] Z. Zhao, J. Zhang, S. Xu, Z. Lin, and H. Pfister, "Discrete cosine transform network for guided depth map super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5697–5707.

[15] Z. Yan, X. Li, K. Wang, S. Chen, J. Li, and J. Yang, "Distortion and uncertainty aware loss for panoramic depth completion," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 39099–39109.

[16] K. Wang et al., "Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16055–16064.

[17] Z. Yan, K. Wang, X. Li, Z. Zhang, J. Li, and J. Yang, "RigNet: Repetitive image guided network for depth completion," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 214–230.

[18] Y. Zhang, X. Guo, M. Poggi, Z. Zhu, G. Huang, and S. Mattoccia, "CompletionFormer: Depth completion with convolutions and vision transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18527–18536.

[19] Z. Yan, K. Wang, X. Li, Z. Zhang, J. Li, and J. Yang, "DesNet: Decomposed scale-consistent network for unsupervised depth completion," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 3109–3117.

[20] Z. Yan, Y. Zheng, D.-P. Fan, X. Li, J. Li, and J. Yang, "Learnable differencing center for nighttime depth perception," *Vis. Intell.*, vol. 2, no. 1, 2024, Art. no. 15.

[21] W. Zhou et al., "BEV, DC: Bird's-eye view assisted training for depth completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9233–9242.

[22] Y. Wang et al., "Improving depth completion via depth feature upsampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21104–21113.

[23] Z. Yan et al., "Tri-perspective view decomposition for geometry-aware depth completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 4874–4884.

[24] J. Tang, F.-P. Tian, B. An, J. Li, and P. Tan, "Bilateral propagation network for depth completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 9763–9772.

[25] S. Shao, Z. Pei, W. Chen, P. C. Y. Chen, and Z. Li, "NDDepth: Normal-distance assisted monocular depth estimation and completion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 8883–8899, Dec. 2024.

[26] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 11–20.

[27] X. Cheng, P. Wang, C. Guan, and R. Yang, "CSPN : Learning context and resource aware convolutional spatial propagation networks for depth completion," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10615–10622.

[28] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 120–136.

[29] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "PENet: Towards precise and efficient image guided depth completion," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13656–13662.

[30] Z. Yan, X. Li, Z. Zhang, J. Li, and J. Yang, "RigNet : Efficient repetitive image guided network for depth completion," 2023, *arXiv:2107.13802*.

[31] Y. Wang, B. Li, G. Zhang, Q. Liu, T. Gao, and Y. Dai, "LRRU: Long-short range recurrent updating networks for depth completion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 9422–9432.

[32] Y. Chen, B. Yang, M. Liang, and R. Urtasun, "Learning joint 2D-3D representations for depth completion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 10023–10032.

[33] J. Qiu et al., "DeepLiDAR: Deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3313–3322.

[34] S. Zhao, M. Gong, H. Fu, and D. Tao, "Adaptive context-aware multi-modal network for depth completion," *IEEE Trans. Image Process.*, vol. 30, pp. 5264–5276, 2021.

[35] L. Huynh, P. Nguyen, J. Matas, E. Rahtu, and J. Heikkilä, "Boosting monocular depth estimation with lightweight 3D point fusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12767–12776.

[36] X. Liu, X. Shao, B. Wang, Y. Li, and S. Wang, "GraphCSPN: Geometry-aware depth completion via dynamic GCNs," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 90–107.

[37] Z. Yu et al., "Aggregating feature point cloud for depth completion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 8732–8743.

[38] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 154–169.

[39] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 353–369.

[40] C. Guo, C. Li, J. Guo, R. Cong, H. Fu, and P. Han, "Hierarchical features driven residual learning for depth map super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2545–2557, May 2019.

[41] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Joint image filtering with deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1909–1923, Aug. 2019.

[42] J. Tang, X. Chen, and G. Zeng, "Joint implicit image function for guided depth super-resolution," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 4390–4399.

[43] B. Kim, J. Ponce, and B. Ham, "Deformable kernel networks for joint image filtering," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 579–600, 2021.

[44] Z. Zhong, X. Liu, J. Jiang, D. Zhao, and X. Ji, "Deep attentional guided image filtering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12236–12250, Sep. 2024.

[45] Z. Wang et al., "Scene prior filtering for depth map super-resolution," 2024, *arXiv:2402.13876*.

[46] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4796–4803.

[47] Z. Xu, H. Yin, and J. Yao, "Deformable spatial propagation networks for depth completion," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 913–917.

[48] Y. Lin, H. Yang, T. Cheng, W. Zhou, and Z. Yin, "DySPN: Learning dynamic affinity for image-guided depth completion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 6, pp. 4596–4609, Jun. 2024.

[49] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2012, pp. 746–760.

[50] L. He et al., "Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9229–9238.

[51] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[52] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[53] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through CNNs for guided sparse depth regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2423–2436, Oct. 2020.

[54] S. Imran, X. Liu, and D. Morris, "Depth completion with twin surface extrapolation at occlusion boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2583–2592.

[55] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *IEEE Trans. Image Process.*, vol. 30, pp. 1116–1129, 2020.

[56] L. Liu et al., "FCFR-Net: Feature fusion based coarse-to-fine residual learning for depth completion," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2136–2144.

[57] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, "Learning affinity via spatial propagation networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1519–1529.

[58] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 103–119.

[59] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2361–2379, Oct. 2020.

[60] R. Cheng, C. Agia, Y. Ren, X. Li, and L. Bingbing, "S3CNet: A sparse semantic scene completion network for LiDAR point clouds," in *Proc. 4th Conf. Robot Learn.*, 2021, pp. 2148–2161.

[61] S. Gu et al., "Learned dynamic guidance for depth image reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2437–2452, Oct. 2020.

[62] Y. Yang, Q. Cao, J. Zhang, and D. Tao, "Codon: On orchestrating cross-domain attentions for depth super-resolution," *Int. J. Comput. Vis.*, vol. 130, no. 2, pp. 267–284, 2022.

[63] Z. Zhong, X. Liu, J. Jiang, D. Zhao, Z. Chen, and X. Ji, "High-resolution depth maps imaging via attention-based hierarchical multi-modal fusion," *IEEE Trans. Image Process.*, vol. 31, pp. 648–663, 2022.

[64] Z. Zhong, X. Liu, J. Jiang, D. Zhao, and X. Ji, "Guided depth map super-resolution: A survey," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–36, 2023.

[65] N. Metzger, R. C. Daudt, and K. Schindler, "Guided depth super-resolution by deep anisotropic diffusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18237–18246.

[66] X. Qiao, C. Ge, C. Zhao, F. Tosi, M. Poggi, and S. Mattoccia, "Self-supervised depth super-resolution with contrastive multiview pre-training," *Neural Netw.*, vol. 168, pp. 223–236, 2023.

[67] F. Zhang, N. Liu, and F. Duan, "Coarse-to-fine depth super-resolution with adaptive RGB-D feature attention," *IEEE Trans. Multimedia*, vol. 26, pp. 2621–2633, 2024.

[68] X. Chen et al., "Intrinsic phase-preserving networks for depth super resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 1210–1218.

[69] W. Shi, M. Ye, and B. Du, "Symmetric uncertainty-aware feature transmission for depth super-resolution," in *Proc. ACM Int. Conf. Multimedia*, 2022, pp. 3867–3876.

[70] X. Deng and P. L. Dragotti, "Deep convolutional neural network for multi-modal image restoration and fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3333–3348, Oct. 2021.

[71] Q. Tang et al., "BridgeNet: A joint learning network of depth map super-resolution and monocular depth estimation," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 2148–2157.

[72] B. Sun, X. Ye, B. Li, H. Li, Z. Wang, and R. Xu, "Learning scene structure guidance via cross-task knowledge transfer for single depth super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7792–7801.

[73] J. Yuan, H. Jiang, X. Li, J. Qian, J. Li, and J. Yang, "Structure flow-guided network for real depth super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 3340–3348.

[74] Z. Sun et al., "Consistent direct time-of-flight video depth super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5075–5085.

[75] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2002–2011.

[76] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4009–4018.

[77] Z. Li, X. Wang, X. Liu, and J. Jiang, "BinsFormer: Revisiting adaptive bins for monocular depth estimation," *IEEE Trans. Image Process.*, vol. 33, pp. 3964–3976, 2024.

[78] S. Shao, Z. Pei, X. Wu, Z. Liu, W. Chen, and Z. Li, "IEBins: Iterative elastic bins for monocular depth estimation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 53025–53037.

[79] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3D semantic occupancy prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9223–9232.

[80] C. Zhang et al., "OccNeRF: Advancing 3D occupancy prediction in LiDAR-free environments," 2023, *arXiv:2312.09243*.

[81] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 12479–12488.

[82] C. Maxey, J. Choi, Y. Lee, H. Lee, D. Manocha, and H. Kwon, "TK-planes: Tiered K-planes with high dimensional feature vectors for dynamic UAV-based scenes," 2024, *arXiv:2405.02762*.

[83] Z. Wang, F. Hamann, K. Chaney, W. Jiang, G. Gallego, and K. Daniilidis, "Event-based continuous color video decompression from single frames," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 4968–4978.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[85] S. Zuo, W. Zheng, Y. Huang, J. Zhou, and J. Lu, "Pointocc: Cylindrical tri-perspective view for point-based 3D semantic occupancy prediction," 2023, *arXiv:2308.16896*.

[86] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.

[87] X. Gu, Y. Guo, F. Deligianni, and G.-Z. Yang, "Coupled real-synthetic domain adaptation for real-world deep depth enhancement," *IEEE Trans. Image Process.*, vol. 29, pp. 6343–6356, 2020.

[88] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 402–419.

[89] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[90] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH*, 2004, pp. 689–694.

[91] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3390–3397.

[92] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[93] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[94] L. Liu et al., "MFF-net: Towards efficient monocular depth completion with multi-modal feature fusion," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 920–927, Feb. 2023.

[95] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.

[96] J. Xiao, A. Owens, and A. Torralba, "Sun3D: A database of big spaces reconstructed using SFM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1625–1632.

[97] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4340–4349.

[98] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–13.

[99] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 646–661.

[100] Y. Lin, T. Cheng, Q. Zhong, W. Zhou, and H. Yang, "Dynamic spatial propagation network for depth completion," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1638–1646.

[101] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy LiDAR completion with RGB guidance and uncertainty," in *Proc. 16th Int. Conf. Mach. Vis. Appl.*, 2019, pp. 1–6.

[102] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the CPU," in *Proc. 15th Conf. Comput. Robot Vis.*, 2018, pp. 16–22.

[103] K. Lu, N. Barnes, S. Anwar, and L. Zheng, "From depth what can you see? depth completion via auxiliary image reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11306–11315.

[104] J. Kam, J. Kim, S. Kim, J. Park, and S. Lee, "CostDCNet: Cost volume based depth completion for a single RGB-D image," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 257–274.

[105] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, "Pixel-adaptive convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11166–11175.

[106] J. Yuan, H. Jiang, X. Li, J. Qian, J. Li, and J. Yang, "Recurrent structure attention guidance for depth super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 3331–3339.

[107] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[108] H. Mei et al., "Glass segmentation using intensity and spectral polarization cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12622–12631.

[109] L. Zhu et al., "RGB-D local implicit function for depth completion of transparent objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4649–4658.

[110] D. Gehrig and D. Scaramuzza, "Low-latency automotive vision with event cameras," *Nature*, vol. 629, no. 8014, pp. 1034–1040, 2024.

[111] Z. Wang, K. Chaney, and K. Daniilidis, "Evac3D: From event-based apparent contours to 3D models via continuous visual hulls," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 284–299.

[112] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[113] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3867–3876.

[114] Z. Yan, J. Jiao, Z. Wang, and G. H. Lee, "Event-driven dynamic scene depth completion," 2025, *arXiv:2505.13279*.

Zhiqiang Yan received the PhD degree from the Nanjing University of Science and Technology, supervised by Prof. Jian Yang, in 2024. He is currently a postdoctoral researcher with the National University of Singapore. His research focuses on 3D computer vision, especially depth estimation, depth completion, and depth super-resolution, which are important for 3D reconstruction, autonomous driving, and 3D perception. He has published more than 10 papers in top conferences such as CVPR, ICCV, and ECCV.

Kun Wang received the PhD degree from the Nanjing University of Science and Technology, in 2025, supervised by Prof. Jian Yang. He is currently a research fellow with the Singapore University of Technology and Design. His research interests include computer vision and machine learning, especially in depth estimation, depth completion, 3D reconstruction, and related topics. He has published 10+ papers in top conferences such as ICCV and NeurIPS.

Xiang Li received the PhD degree from the Nanjing University of Science and Technology, Jiangsu, China, in 2020. He is an associate professor with the College of Computer Science, Nankai University. His research interests include CNN/Transformer backbones, object detection, knowledge distillation, and self-supervised learning. He has published 30+ papers in top journals and conferences such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, CVPR, NeurIPS, etc.

Guangwei Gao (Senior Member, IEEE) received the PhD degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, in 2014. He was a visiting student with the Department of Computing, The Hong Kong Polytechnic University, in 2011 and 2013, respectively. He was also a project researcher with the National Institute of Informatics, Japan, in 2019. He is currently a professor with the Nanjing University of Posts and Telecommunications. His research interests include pattern recognition and computer vision. He has published more than 70 scientific papers in *IEEE Transactions on Image Processing*/*IEEE Transactions on Circuits and Systems for Video Technology*/*IEEE Transactions on Intelligent Transportation Systems*/*IEEE Transactions on Multimedia*/*IEEE Transactions on Information Forensics and Security*, *ACM Transactions on Internet Technology*/*ACM Transactions on Multimedia Computing, Communications, and Applications*, *Pattern Recognition*, AAAI, IJCAI, etc.

Jun Li (Member, IEEE) received the PhD degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2015. From 2012 to 2013, he was a visiting student with the Department of Statistics, Rutgers University, Piscataway, NJ, USA. From 2015 to 2018, he was a post-doctoral associate with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. From 2018 to 2019, he was a post-doctoral associate with the Institute of Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, USA. He has been a professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, since 2019. His research interests are computer vision and creative learning. He has served as an AC/SPC for NeurIPS/ACM MM/AAAI.

Jian Yang received the PhD degree from the Nanjing University of Science and Technology (NUST), in 2002, majoring in pattern recognition and intelligence systems. From 2003 to 2007, he was a postdoctoral fellow with the University of Zaragoza, Hong Kong Polytechnic University and New Jersey Institute of Technology, respectively. From 2007 to present, he is a professor with the School of Computer Science and Technology of NUST. His papers have been cited more than 50000 times in the Scholar Google. His research interests include pattern recognition and computer vision. Currently, he is/was an associate editor of *Pattern Recognition*, *Pattern Recognition Letters*, *IEEE Trans. Neural Networks and Learning Systems*, and *Neurocomputing*. He is a fellow of IAPR.