Aggregating Feature Point Cloud for Depth Completion

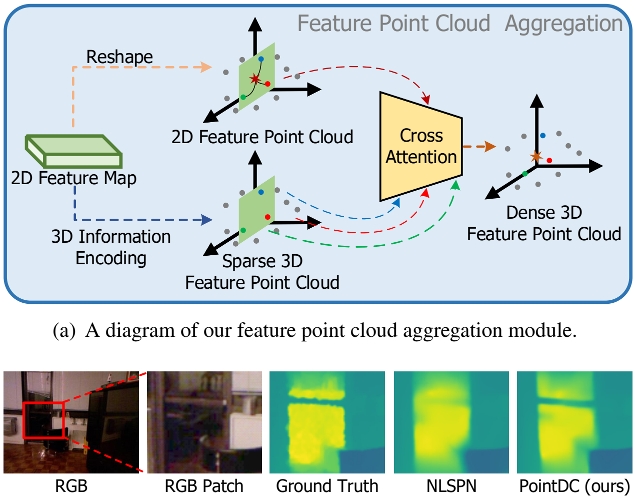
Zhu Yu[1], Zehua Sheng[1], Zili Zhou[1], Lun Luo[1],
Si-Yuan Cao[2,1*] Hong Gu[4], Huaqi Zhang[4], Hui-Liang Shen[1,3*]
[1]College of Information Science and Electronic Engineering, Zhejiang University
[2]Ningbo Innovation Center, Zhejiang University
[3]Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province
[4]vivo Mobile Communication Company Ltd.
{yu_zhu, shengzehua, zhou_zili, luolun, cao_siyuan, shenhl}@zju.edu.cn
{guhong, zhanghuaqi}@vivo.com

Abstract

Guided depth completion aims to recover dense depth maps by propagating depth information from the given pixels to the remaining ones under the guidance of RGB images. However, most of the existing methods achieve this using a large number of iterative refinements or stacking repetitive blocks. Due to the limited receptive field of conventional convolution, the generalizability with respect to different sparsity levels of input depth maps is impeded. To tackle these problems, we propose a feature point cloud aggregation framework to directly propagate 3D depth information between the given points and the missing ones. We extract 2D feature map from images and transform the sparse depth map to point cloud to extract sparse 3D features. By regarding the extracted features as two sets of feature point clouds, the depth information for a target location can be reconstructed by aggregating adjacent sparse 3D features from the known points using cross attention. Based on this, we design a neural network, called as PointDC, to complete the entire depth information reconstruction process. Experimental results show that, our PointDC achieves superior or competitive results on the KITTI benchmark and NYUv2 dataset. In addition, the proposed PointDC demonstrates its higher generalizability to different sparsity levels of the input depth maps and cross-dataset evaluation.

1. Introduction

In recent years, dense depth maps have shown great importance in various computer vision tasks, including autonomous driving [39, 7], 3D object detection [48, 47], augmented reality[22, 9, 55] and 3D reconstruction [11, 35].



(a) A diagram of our feature point cloud aggregation module.

(b) An example scene for comparison.

Figure 1. The diagram of feature point cloud aggregation module of our PointDC. It reconstructs 3D information for the 2D feature map by aggregating from the 3D features, where the 2D and 3D features are viewed as two sets of feature point clouds. Compared with the state-of-the-art depth completion approach NLSPN [31], our PointDC can still achieve better results although the details in the RGB image are hard to discriminate by human eyes.

However, commercially available depth sensors, such as LiDARs or RGB-D cameras, typically produce highly sparse depth maps that cannot accurately capture the full 3D information of the scene. To address this limitation, recent researches [20, 42, 10] have focused on directly reconstructing dense depth maps from sparse observations. Despite significant progress, this approach remains challenging due to the ill-posed nature of the problem, which often leads to unsatisfactory accuracy. In comparison, a more promising solution is to incorporate an additional RGB image captured

[*] Corresponding author.

in the same scene. Based on the auxiliary structural information, it is much easier to complete the sparse depth map. This approach, known as *guided depth completion*, has become one of the important steps for the aforementioned vision applications.

Given a sparse depth image, guided depth completion essentially aims to propagate depth information from known pixels to the remaining ones under the guidance of an RGB image [5, 31]. Generally, it can be classified into two categories. The first one [5, 6, 31, 27, 16, 54, 41, 52] treats sparse depth maps as ordinary images and formulates guided depth completion as a guided image restoration task, where depth values are regarded as pixel intensities. In this case, the information is propagated by learning various types of affinities among neighboring pixels from RGB images [5, 6, 31, 27] or constructing content-adaptive neural networks [54, 41, 52]. However, these methods are primarily designed to operate in 2D space and therefore struggle to fully exploit the 3D geometry information that has been demonstrated to be beneficial for depth estimation in both multi-view stereo (MVS) and stereo matching methods [14, 13]. To explicitly consider 3D geometry information, the second category of methods [23, 21, 4, 17] extracts 3D features using point cloud convolutions [46, 1] or by interpreting depth information with plane-residual representation [23, 21]. This category of methods propagate information by employing either 2D or 3D convolution. In a word, both types of methods propagate depth information in a progressive or an iterative manner due to limited receptive field. Consequently, they may be less robust in cases where there are varying levels of point sparsity, as it becomes increasingly difficult to propagate information between distant pixels when the densities of sparse points decrease.

In this work, we propose a feature point cloud aggregation framework to directly propagate the given sparse depth information to the entire image. In this way, our framework can overcome the limited receptive field of conventional convolutions and generalize well to different sparsity levels of the input depth maps. Given the inputs, we transform the depth map to point cloud using the camera intrinsic matrix. Then we extract sparse 2D features from the images and 3D features from the point cloud. We hypothesize that the 2D features only give visual descriptions of the scene while the 3D features contain the extra 3D information. Generally, similar visual contents tend to have similar depth values within neighboring regions. Therefore, the 3D depth information of a target location can be reconstructed from the adjacent sparse 3D features using cross-attention strategy. By referring the 2D and sparse 3D features as the *2D* and the *sparse 3D feature point clouds*, the reconstruction process can be achieved in a cross-attention manner with higher flexibility.

Based on the above analysis, we design a neural network, called as PointDC, to handle the depth completion task. First, PointDC generates the 2D and 3D feature point clouds with a UNet [36] and several stacked local self-attention transformer blocks, respectively. Then, for a target location, its 3D depth information is reconstructed based on its neighboring 3D feature points by the feature point cloud aggregation module which is mainly a local cross-attention transformer block. A diagram of this module is shown in Fig. 1(a). Finally, from the reconstructed *dense 3D feature point cloud*, PointDC generates the final dense depth map.

In summary, the main contributions of this work are as follows:

- We propose a feature point cloud aggregation framework which extracts both 2D and sparse 3D features for depth completion. It reconstructs the depth information for a target location by the adjacent sparse 3D feature points, in which each location can capture 3D information from the sparse 3D features directly regardless of the sparsity level of the input depth maps.
- We design a novel local transformer by regarding the extracted features as two sets of point clouds, which is used to exploit 3D geometry information and reconstruct the depth information for each target location.
- Experimental results show that our PointDC achieves better or comparable results compared to state-of-the-art depth completion methods. In addition, our PointDC demonstrates its higher generalizability to different sparsity levels of the input depth maps and cross-dataset evaluation.

2. Related Work

Depth Completion. Depth completion restores dense depth maps by propagating information from the observed pixels to unobserved ones [20]. Early depth-only methods [20, 42, 10] generate dense depth maps using only one single sparse image by designing appropriate operators (*e.g*. sparse invariant CNN [42], normalized convolutional neural network (NCNN) [10]). However, the information propagation of these methods depends on pixel locations, whose performance is limited when the input depth maps are highly sparse.

To attain higher performance, guided depth completion introduces an additional RGB image to assist the completion process. In this case, the information propagation can be guided by auxiliary structural information of the RGB image, significantly boosting the results compared to those depth-only methods. Existing guided depth completion approaches can be roughly classified into two categories. The first one regards it as a guided image restoration task, which propagates information by regular or dynamic convolution. S2D [30] directly concatenate RGB and depth images and then feed them to a simple U-Net [36]. Following the spatial propagation network (SPN) [26], SPN-based methods [5, 6, 31, 27] first estimate a rough result, and then refine it by local, non-local, or other modified types of affinities. A few advanced methods [52, 41, 54] fuse multi-modal features by constructing content-adaptive neural networks. Auxiliary tasks [20, 32, 51] are also adopted to better supervise the learning process. However, these methods are unable to capture 3D geometry information which has shown to be useful for depth estimation in [14, 13].

Different from the above methods which mainly conduct completion in the 2D image space, the other type of works attempt to consider 3D geometry information explicitly. Base on the plane-residual representation [23], some methods [23, 21] borrow the cost volume concept to extract 3D information from the sparse depth maps and formulate depth prediction as a classification-regression problem. FuseNet [4] and Point-Fusion [17] extract 3D features using point cloud convolution [46, 1] and directly consolidate 2D and 3D features. Similar to the first category, the 3D information propagation of the second is fulfilled by 2D or 3D convolution. To sum up, to propagate information to entire image, both types of methods need to achieve this in a progressive or an iterative manner due to limited receptive field. Therefore, these methods may be less robust to different levels of points sparsity.

Vision Transformer. In recent studies, ViTs [8, 28] have demonstrated huge potential in various vision tasks due to larger receptive field, including image classification [3, 24], image segmentation [40, 25], dense prediction [33, 50], *etc.* DPT [33] adopts ViT[8] as a backbone for encoding global information at multiple stages for depth estimation and semantic segmentation. Based on the long-range modeling property of the attention mechanism, GMFlow [50] reformulates optical flow estimation as a global matching problem. RHWF [2] employs the attention focusing mechanism, which captures the intra/inter correspondence information in a global→nonlocal→local manner. Guideformer [34] firstly introduces transformer into depth completion, which enlarges the receptive field for propagating information in the long range. However, this method also regards guided depth completion as a guided restoration task, which can't exploit 3D geometry information. In this work, we devise the transformer-based PointDC to effectively extract and propagate the 3D geometry information contained in the input sparse depth maps.

3. Problem Definition

Given a sparse depth image $\mathbf{S} \in \mathbb{R}^{H\times W}$ and an RGB image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, guided depth completion aims to restore a dense depth map $\mathbf{D} \in \mathbb{R}^{H\times W}$. The main purpose of this task is to propagate the depth information from the given points to the entire image under the guidance of the RGB image. Based on the assumption that similar visual contents tend to have similar depth values within neighboring regions [41], we restore the depth value of a target location by aggregating the known depth information using the visual similarity between the target point and the given points.

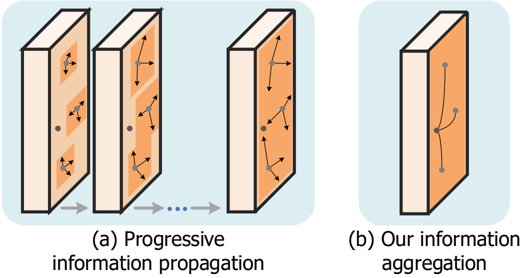

(a) Progressive information propagation
(b) Our information aggregation

Figure 2. Comparison between the information propagation process of previous methods and our PointDC. Previous methods require to propagate information in a progressive or an iterative manner while our PointDC can directly propagate the depth information from the given points to the entire image.

Formally, the 2D feature map $\mathbf{F}$ can be obtained as follows

$$\mathbf{F} = f\left(\mathbf{S}, \mathbf{I}\right) \in \mathbb{R}^{H\times W\times C}, \tag{1}$$

where $f\left(\cdot\right)$ denotes the 2D feature extraction function. As $\mathbf{F}$ is computed in the 2D space, we hypothesize that it mainly represents the 2D visual information. Then, we index $\mathbf{F}$ using the positions where the values in $\mathbf{S}$ are given to obtain $\mathbf{H}_i \in \mathbb{R}^{M\times C}$ and transform $\mathbf{S}$ to point cloud $\mathbf{P} \in \mathbb{R}^{M\times C}$ via the camera intrinsic matrix. M is the number of given points. Both $\mathbf{H}_i$ and $\mathbf{P}$ serve as the inputs of the the 3D geometry information encoding function $g\left(\cdot\right)$ to further extract the extra 3D information contained in the point cloud. This process can be formally denoted as follows

$$\mathbf{H} = g\left(\mathbf{P}, \mathbf{H}_i\right) \in \mathbb{R}^{M\times C}. \tag{2}$$

As $\mathbf{H}_i$ is indexed from the 2D feature map and encoded in the 3D space with $\mathbf{P}$ to capture 3D geometry information, we hypothesize that $\mathbf{H}$ contains both 2D visual and 3D depth information. For a specific pixel of $\mathbf{F}$, estimating its depth value equals reconstructing its 3D depth information. Similar to point cloud completion [53], the reconstruction process can be fulfilled by aggregating from the points in $\mathbf{H}$.

By reshaping $\mathbf{F}$ to a shape of $HW\times C$ and denoting the reshaped result as $\mathbf{F}_l$, $\mathbf{H}$ and $\mathbf{F}_l$ can be viewed as two sets of feature point clouds, where the former contains only 2D visual information and the latter contains full information. We denote the set of the points whose depth information has been reconstructed as $\hat{\mathbf{F}}_l \in \mathbb{R}^{HW\times C}$. For simplicity, we name $\mathbf{F}_l$ as *2D feature point cloud*, $\mathbf{H}$ as *sparse 3D feature point cloud*, and $\hat{\mathbf{F}}_l$ as *dense 3D feature point cloud*. Let denote the index of $\mathbf{F}_l$ and $\hat{\mathbf{F}}_l$ as $\mathbf{x}$. To obtain $\hat{\mathbf{F}}_l\left(\mathbf{x}\right) \in \mathbb{R}^{C\times 1}$,

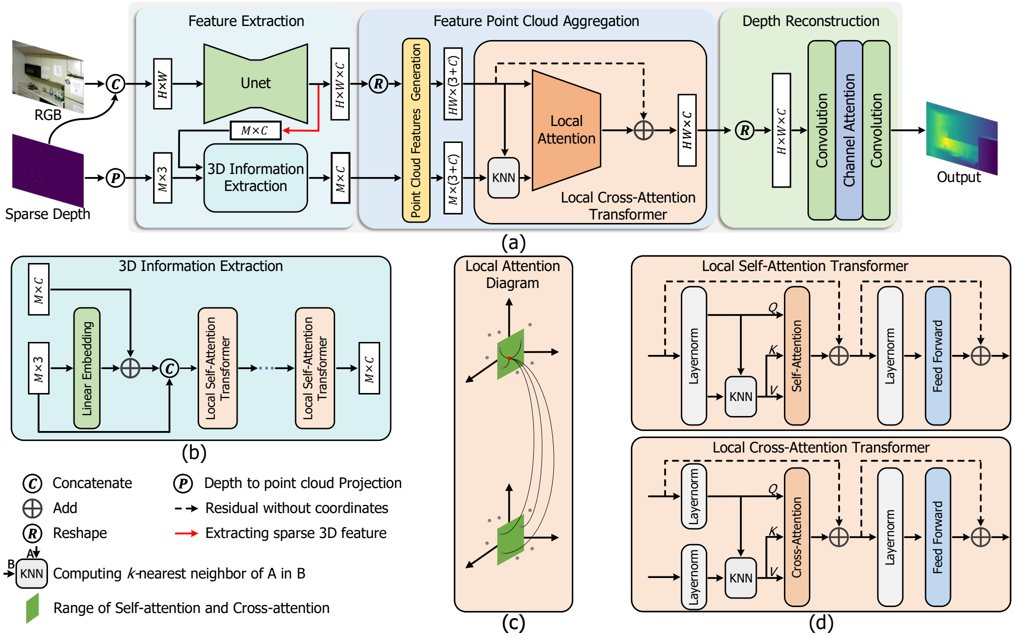

Feature Extraction
Feature Point Cloud Aggregation
Depth Reconstruction
RGB
Sparse Depth
C
P
H×W
M×3
Unet
M×C
3D Information Extraction
H×W×C
M×C
R
Point Cloud Features Generation
HW×(3+C)
M×(3+C)
KNN
Local Attention
HW×C
Local Cross-Attention Transformer
R
H×W×C
Convolution
Channel Attention
Convolution
Output
(a)
3D Information Extraction
M×C
M×3
Linear Embedding
C
Local Self-Attention Transformer
Local Self-Attention Transformer
M×C
(b)
Local Attention Diagram
(c)
Local Self-Attention Transformer
Layernorm
Q
K
V
KNN
Self-Attention
Layernorm
Feed Forward
Local Cross-Attention Transformer
Layernorm
Layernorm
Q
K
V
KNN
Cross-Attention
Layernorm
Feed Forward
(d)
C Concatenate
Add
R Reshape
P Depth to point cloud Projection
Residual without coordinates
Extracting sparse 3D feature
A
B
KNN Computing k-nearest neighbor of A in B
Range of Self-attention and Cross-attention

Figure 3. Schematics and detailed architectures of PointDC. (a) Overall architecture of our PointDC. (b) Detailed structure of the 3D information extraction block. (c) Diagram of the local attention mechanism. (d) Details of the local self-attention transformer and the local cross-attention transformer.

we first compare the feature similarity of $\mathbf{F}_l\left(\mathbf{x}\right) \in \mathbb{R}^{C\times 1}$ with respect to all sparse 3D feature points of $\mathbf{H}$ by computing their correlations. This can be implemented efficiently with a simple matrix multiplication

$$\mathbf{W} = \frac{\mathbf{H}\mathbf{F}_l\left(\mathbf{x}\right)}{\sqrt{C}} \in \mathbb{R}^{M\times 1}, \tag{3}$$

where $\mathbf{W}$ represents the correlation matrix. Each element of $\mathbf{W}$ measures the visual similarity between the target point and one of the sparse 3D feature points. Then, we normalize $\mathbf{W}$ with the softmax operation

$$\hat{\mathbf{W}} = \mathrm{softmax}\left(\mathbf{W}\right) \in \mathbb{R}^{M\times 1}. \tag{4}$$

Finally, $\hat{\mathbf{F}}_l\left(\mathbf{x}\right)$ can be computed by

$$\hat{\mathbf{F}}_l\left(\mathbf{x}\right) = \mathbf{H}^{\top}\hat{\mathbf{W}} \in \mathbb{R}^{C\times 1}, \tag{5}$$

To sum up, the overall reconstruction process can be formulated as follows

$$\hat{\mathbf{F}}_l\left(\mathbf{x}\right) = \mathbf{H}^{\top}\,\mathrm{softmax}\left(\frac{\mathbf{H}\mathbf{F}_l\left(\mathbf{x}\right)}{\sqrt{C}}\right). \tag{6}$$

In this way, each 2D feature point $\mathbf{F}_l\left(\mathbf{x}\right)$ can directly captures 3D geometry information from the sparse 3D feature point cloud $\mathbf{H}$ based on their visual similarity regardless of the sparsity level of the input depth maps. A simple diagram of this process is shown in Fig. 2(b). In comparison, previous methods require to propagate information in a progressive or an iterative manner, as shown in Fig. 2(a). Besides, only the feature points within neighboring regions contribute to the result mostly, so it's unnecessary to compute the correlation globally. Therefore, we reconstruct depth information of $\mathbf{F}_l\left(\mathbf{x}\right)$ by using its k neighboring points in $\mathbf{H}$, where the distance between two points is measured by the euclidean distance of their coordinates. Then, Eq. 6 is reformulated as

$$\hat{\mathbf{F}}_l\left(\mathbf{x}\right) = \mathbf{H}\left(\mathcal{N}\left(\mathbf{x}\right)\right)^{\top}\,\mathrm{softmax}\left(\frac{\mathbf{H}\left(\mathcal{N}\left(\mathbf{x}\right)\right)\mathbf{F}_l\left(\mathbf{x}\right)}{\sqrt{C}}\right), \tag{7}$$

where $\mathcal{N}\left(\mathbf{x}\right)$ denotes the indices of the k neighboring points of $\mathbf{F}_l\left(\mathbf{x}\right)$ in $\mathbf{H}$.

4. PointDC

The schematics of proposed PointDC is shown in Fig. 3. It mainly consists of three modules, *i.e.*, *feature extraction*, *feature point cloud aggregation* and *depth reconstruction*. Given an RGB image and a sparse depth image, the feature extraction module extracts the 2D feature map $\mathbf{F}$ and the sparse 3D features $\mathbf{H}$. Then the feature point cloud aggregation module reconstructs the 3D information of the points of $\mathbf{F}_l$ by aggregating the points of $\mathbf{H}$ to generate dense 3D features $\hat{\mathbf{F}}_l$. Finally, the depth reconstruction module regresses the final dense depth map $\hat{\mathbf{D}}$ from $\hat{\mathbf{F}}$.

4.1. Feature Extraction

As shown in Fig. 3(a), the feature extraction module consists of two branches, a 2D branch that extracts 2D feature map $\mathbf{F}$ and a 3D branch that extracts sparse 3D features $\mathbf{H}$. In the 2D branch, following most of the existing methods [31, 27], the 2D feature extraction network is a UNet [36] which adopts ResNet-34 [15] as the backbone. First, we concatenate the RGB and sparse depth images, and then feed them into the 2D feature extraction network to generate the 2D feature map $\mathbf{F}$. Next, we obtain sparse point clouds $\mathbf{P} \in \mathbb{R}^{M \times 3}$ from the sparse depth maps using the camera intrinsic matrix and extract the initial sparse 3D features $\mathbf{H}_i$ from $\mathbf{F}$. In the 3D branch, the 3D information extraction block takes both $\mathbf{P}$ and $\mathbf{H}_i$ as inputs, and outputs the 3D features $\mathbf{H}$. The detailed architecture of this block is shown in Fig. 3(b). Linear embedding is used to extract features $\mathbf{F}_P \in \mathbb{R}^{M \times C}$ from $\mathbf{P}$. Then we sum $\mathbf{F}_P$ to $\mathbf{H}_i$ to get $\mathbf{H}_P$. Finally, N stacked local self-attention transformer (LST) blocks are used to encode $\mathbf{H}_P$ for better exploiting 3D geometry information and generate $\mathbf{H}$. In this work, N is empirically set to 4.

The details of LST are shown at the top of Fig. 3(d). We do not simply compute the global attention considering the computational complexity. Therefore, before sending $\mathbf{H}_P$ to the transformer blocks, we first concatenate it with $\mathbf{P}$ which serve as coordinates of $\mathbf{H}_P$ to measure the distances between two feature points so that each point only requires k_1 of its neighboring points for attention computing. A simple diagram of the attention mechanism is shown in Fig. 3(c). Let denote $\mathbf{y} = (x, y, z)$ the index of $\mathbf{H}$, the local self attention mechanism can be formulated as

$$\mathbf{H}(\mathbf{y}) = \mathbf{H}_P(\mathcal{N}(\mathbf{y}))^\top \operatorname{softmax}\left(\frac{\mathbf{H}_P(\mathcal{N}(\mathbf{y}))\,\mathbf{H}_P(\mathbf{y})}{\sqrt{C}}\right), \tag{8}$$

where $\mathcal{N}(\mathbf{y})$ denotes the set of k_1 nearest neighboring indices of $\mathbf{y}$. In this work, we set k_1 to 9 following [4, 17].

4.2. Feature Point Cloud Aggregation

After obtaining $\mathbf{F}$ and $\mathbf{H}$, we reconstruct the 3D information of the elements in $\mathbf{F}$ based on $\mathbf{H}$ using a local cross attention transformer (LCT) block. The detailed architecture of this block is shown at the bottom of Fig. 3(d). Before sending $\mathbf{F}$ and $\mathbf{H}$ to the LCT block, we first reshape $\mathbf{F}$ to the 2D feature point cloud $\mathbf{F}_l$. We concatenate coordinates for $\mathbf{F}_l$. To keep the same dimension of coordinates with LST, instead of simply using 2D image plane coordinates $\mathbf{x} = (i, j)$, we add an additional dimension to $\mathbf{x}$. We denote $\hat{\mathbf{x}} = (i, j, d^*)$ as the new coordinate, where d^* a constant value. The forms of coordinates are the same for $\mathbf{H}$. Then the 3D information of the 2D feature point cloud is reconstructed by the LCT blocks. The cross attention mechanism can be formulated as

$$\hat{\mathbf{F}}_l(\hat{\mathbf{x}}) = \mathbf{H}(\mathcal{N}(\hat{\mathbf{x}}))^\top \operatorname{softmax}\left(\frac{\mathbf{H}(\mathcal{N}(\hat{\mathbf{x}}))\,\mathbf{F}_l(\hat{\mathbf{x}})}{\sqrt{C}}\right), \tag{9}$$

where $\mathcal{N}(\hat{\mathbf{x}})$ denotes the set of k_2 nearest neighboring indices of $\hat{\mathbf{x}}$ in the 3D feature point cloud. In this work, k_2 is empirically set to 9.

4.3. Depth Reconstruction

In the preceding subsections, we have discussed the process of reconstructing dense 3D feature point cloud $\hat{\mathbf{F}}_l$. To generate the final result, we first reshape $\hat{\mathbf{F}}_l$ to a map $\hat{\mathbf{F}} \in \mathbb{R}^{H \times W \times C}$. Then we suppress redundant channels of $\hat{\mathbf{F}}$ by a channel attention block and finally generate the recovered dense depth map $\hat{\mathbf{D}}$ with a convolution layer.

4.4. Loss Function

We train our network with a combination of L_1 loss, L_2 loss and gradient loss L_{grad}:

$$\mathcal{L} = L_1 + \gamma L_2 + \mu L_{\text{grad}}, \tag{10}$$

where L_1 and L_2 are L1 norm and L2 norm between the estimated result $\hat{\mathbf{D}}$ and the ground truth $\mathbf{D}$, respectively. L_{grad} penalizes the errors on edges. γ and μ are the coefficients to control the trade-off between the three losses. γ is set to 0 for NYU Depth v2 dataset [37] and 1 for KITTI DC dataset [43]. μ is empirically set to 0.7 for all datasets.

5. Experiments

5.1. Datasets and Metrics

NYU Depth v2 dataset. The NYU-Depth-v2 dataset [37] is captured by Microsoft Kinect sensor, containing both RGB and depth sequences of 464 indoor scenes. Following previous work [31, 52, 45], we adopt a subset of 50K images as training set and evaluate on the official labeled test set. For training and testing, we first down-sample images to 320 × 240 and then center-crop them to 304 × 228 to remove the invalid regions.

KITTI Depth Completion Dataset. The KITTI Depth completion dataset [12, 43] is a large outdoor dataset captured by a driving vehicle. It provides 86K RGB and LiDAR pairs for training, 1K pairs for validation and the remaining 1K pairs for testing. As the depth maps are captured by HDL-64 LiDAR sensor, each single depth map contains less than 6% valid values and the ground truth depth maps are generated by aggregating multiple consecutive frames, whose density is about 14%. Since there are nearly no valid points at the top regions of depth images, the input images are bottom center cropped to 1216 × 240.

SUN RGBD Dataset. The SUN RGBD dataset [38] is an indoor dataset containing RGB-D images constructed based on several existing datasets [37, 19, 49]. We use it only for cross-dataset evaluation. 555 frames captured by Kinect V1 and 3389 captured by Asus Xtion camera are used to evaluate our model, where we conduct the same preprocessing as in the NYUv2 dataset.

Method	RMSE ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Bilateral [37]	0.479	0.084	92.4	97.6	98.9
S2D [30]	0.204	0.043	97.8	99.6	99.9
CSPN [5]	0.117	0.016	99.2	99.9	100.0
DeepLiDAR [32]	0.115	0.022	99.3	99.9	100.0
DepthNormal [51]	0.112	0.018	99.5	99.9	100.0
ACMNet [54]	0.105	0.015	99.4	99.9	100.0
GuideNet [41]	0.101	0.015	99.5	99.9	100.0
TWICE [18]	0.097	0.013	**99.6**	99.9	100.0
NLSPN [31]	0.092	**0.012**	**99.6**	99.9	100.0
RigNet [52]	0.090	0.013	**99.6**	99.9	100.0
GraphCSPN [27]	0.090	**0.012**	**99.6**	99.9	100.0
PRNet [23]	0.104	0.014	99.4	99.9	100.0
CostDCNet [21]	0.096	0.013	99.5	99.9	100.0
Point-Fusion [17]	0.090	0.014	**99.6**	99.9	100.0
PointDC (ours)	**0.089**	**0.012**	**99.6**	99.9	100.0

Table 1. Quantitative comparisons on the NYU Depth V2 dataset [37]. The metrics RMSE and REL are presented in meters (m). Algorithms of the upper block regard depth completion as a guided image restoration task while the ones of the lower block exploit 3D geometry information and fuse with 2D features.

Metrics. Following existing methods [31, 27, 52], we use five metrics for NYUv2 dataset, including RMSE, REL, and $\delta_i (i = 1.25, 1.25^2, 1.25^3)$. For the KITTI depth completion dataset, we use four metrics, including RMSE, MAE, iRMSE and iMAE.

5.2. Implementation Details

PointDC is implemented with the Pytorch framework. We adopt the AdamW optimizer [29] with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and set the maximum learning rate to 5×10^{-4}. For the NYU-Depth-v2 dataset, we train the model for 150000 iterations, where the batch size is set to 16 and 500 depth pixels are randomly sampled from the ground truth to generate the input sparse depth map. For the KITTI DC dataset, the model is trained for 300,000 iterations with a batch size of 8. We randomly sample 10, 000 points for training following [4]. The cosine annealing learning rate strategy is adopted for the learning rate decay where the cosine warm-up strategy is applied for the first 5% iterations.

5.3. Evaluation on NYU Depth v2 Dataset

We first evaluate PointDC on the official test split of NYU-Depth-v2 dataset [37]. For quantitative comparison, we list the results in Table 1. We divide various depth completion algorithms into two categories: one category regards the guided depth completion as a guided image restoration task, the other category learns both 2D and 3D information. As shown in Table 1, PointDC achieves the best accuracy measured by all evaluation metrics.

To conduct qualitative comparisons, we display three examples in Fig. 4. In the simple scenes of the first two rows, PointDC generates more details than other methods, for example, the kettle in the first row and the chair in the second row. A more challenging example is shown in the third row of Fig. 4. Although the details in color image are hard to discriminate by human eyes, PointDC can still effectively recover good results while the other methods fail to achieve this. The above quantitative and qualitative comparisons demonstrate the excellent performance of our model.

Components	RMSE (m)↓	REL (m) ↓	#param (M)↓
w/o feature point cloud aggregation	0.093	0.013	25.07
w/o 3D information Extraction	0.091	**0.012**	24.967
w/o channel attention	0.090	**0.012**	25.097
full model	**0.089**	**0.012**	25.098

Table 2. Ablation studies on the feature point cloud aggregation, 3D information extraction and channel attention modules.

K nearest neighbors	3	6	9	12	15	18
RMSE (m)↓	0.089	0.090	0.089	0.090	0.090	0.089

Table 3. Ablation studies on the number of k-nearest neighbors in the 3D information extraction module and the feature point cloud aggregation module, which influence the model's receptive field.

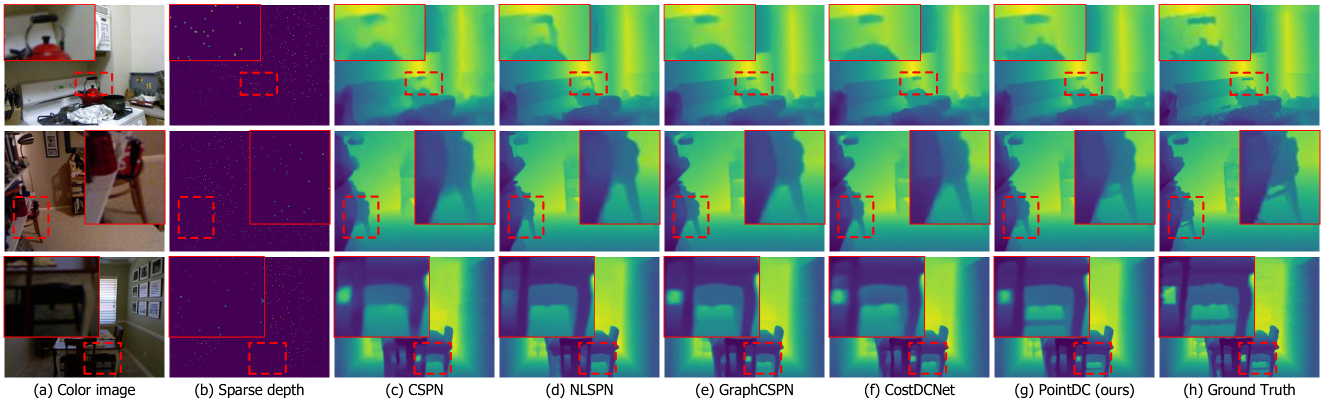
number of blocks	1	2	3	4
RMSE (mm)↓	90.50	90.33	89.11	88.83

Table 4. Ablation studies on the number of local self-attention transformer blocks in the 3D information extraction module.

5.4. Ablation Study

To verify the effectiveness of the components in PointDC, including feature point cloud aggregation, 3D information extraction and channel attention, we conduct ablation studies on the NYU Depth v2 dataset [37]. Additionally, we validate the influence of the number of k-nearest neighbors in the feature point cloud aggregation module.

Components in PointDC. We respectively deactivate each of the three components to validate their effectiveness and list the test results in Table 2. Without feature point cloud aggregation module, the extracted 3D information is directly added to the image feauture map. In this case, the propagation of 3D information is similar to [4, 17] which mainly depends on the convolutions. We can observe that the accuracy drops the most without the feature point cloud aggregation module. Without the 3D information extraction module, the information during propagation is mainly extracted at 2D space, which do not fully exploit 3D geometry information. Without channel attention, some unnecessary channels are not suppressed and they will bring adverse effects to the performance. The results demonstrate that the accuracy of the full network decreases when any of the three components is deactivated. We also evaluate the influence of each component on the generalizability of the model under different sparsity levels of the input depth maps. We display the results in Fig. 5. We can observe that the RMSE



(a) Color image (b) Sparse depth (c) CSPN (d) NLSPN (e) GraphCSPN (f) CostDCNet (g) PointDC (ours) (h) Ground Truth

Figure 4. **Qualitative depth completion results on the NYU Depth V2 dataset.** (a) Color image, (b) Sparse depth, (c) CSPN [5], (d) NLSPN [31], (e) GraphCSPN [27], (f) CostDCNet [21], (e) PointDC (ours), (h) Ground truth. All the results are generated under the same 500 samples.

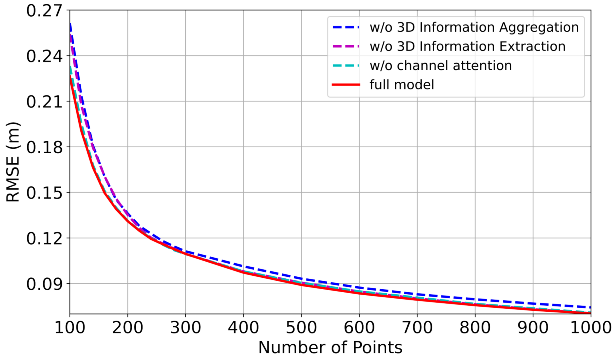


Figure 5. Generalization comparison of completion results obtained by our PointDC in four cases: (1) without feature point cloud aggregation, (2) without 3D information extraction, (3) without channel attention, (4) full model.

value drops quickly with the decrease of the number of the sampled points when removing the feature point cloud aggregation and 3D information extraction modules. To sum up, the above ablation studies demonstrate the effectiveness of these components.

The number of k-nearest neighbors in feature point cloud aggregation module. We conduct experiments to validate the influence of k_2 in the feature point cloud aggregation module and list the results in Table 3. We can observe that our model's performance is stable in terms of the RMSE, which demonstrates that our model is quite robust to this hyper-parameter. The reason is that the neighboring feature points within local regions mostly contribute to the results, as mentioned in Section 3. For simplicity, we set this number to 9 for fair comparisons, which proves that the improvements of accuracy are mainly contributed by our long-range information propagation strategy.

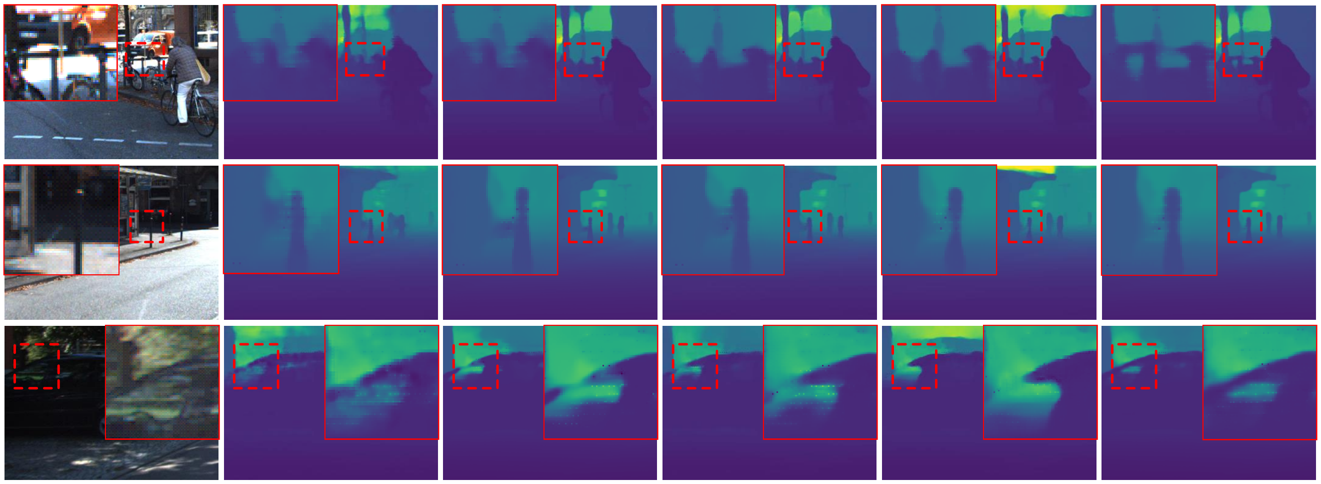
The number of local self-attention transformer blocks in the 3D information extraction module. To validate how the number of the local self-attention transformer blocks influence the final accuracy, we change the number of N and list the results in Table 4. As shown in the table, with the increasing of N, the RMSE of our PointDC decreases. And it tends to saturate when N is larger than 2. The results prove that exploiting 3D geometry information within a number of neighboring points plays an important role. In this work, we set N to 4.

Method	RMSE ↓	MAE ↓	iRMSE↓	iMAE↓
CSPN [5]	1019.64	279.46	2.93	1.15
TWICE [18]	840.20	**195.58**	2.08	**0.82**
DepthNormal [51]	777.05	235.17	2.42	1.13
DeepLiDAR [32]	758.38	226.50	2.56	1.15
FuseNet [4]	752.88	221.19	2.34	1.14
ACMNet [54]	744.91	206.09	2.08	0.90
NLSPN [31]	741.68	199.59	1.99	0.84
GraphCSPN [27]	738.41	199.31	**1.96**	0.84
GuideNet [41]	736.24	218.83	2.25	0.99
PENet [16]	730.08	210.55	2.17	0.94
Guideformer [34]	721.48	207.76	2.14	0.97
RigNet [52]	**712.66**	203.25	2.08	0.90
FuseNet [4]	752.88	221.19	2.34	1.14
Point-Fusion [17]	741.9	201.10	1.97	0.85
PointDC (ours)	736.07	201.87	1.97	0.87

Table 5. Quantitative comparisons on the KITTI Depth Completion test dataset [43]. The metrics RMSE and REL are presented in millimeter (mm), while iRMSE and iMAR are presented in 1/kilometer (1/km). Following Tab 1. Algorithms on the upper block mainly reason in the 2D image space while the lower ones exploit 3D geometry information.

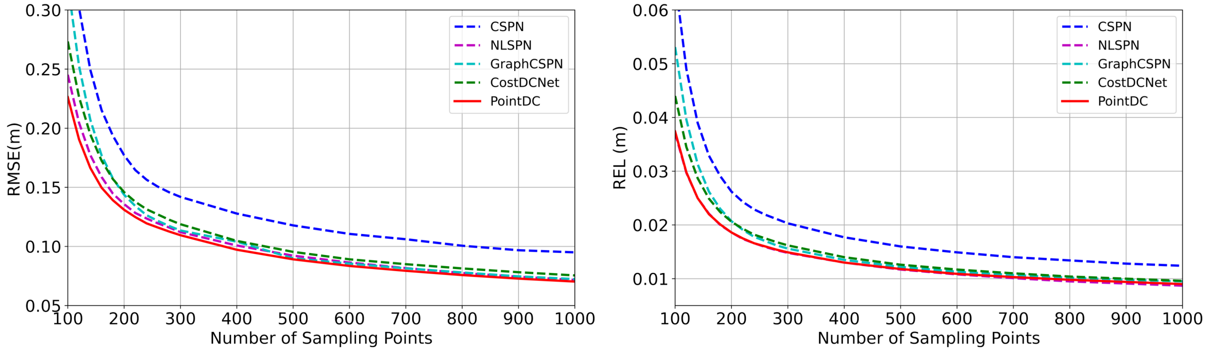
5.5. Evaluation on KITTI DC Dataset

To demonstrate the versatility of our model, we evaluate PointDC with the KITTI Depth Completion dataset [43] and list the qualitative results Table 5. Our model ranks 4th in terms of RMSE, but we excel all the methods which exploit 3D information in this metric. In the upper block,



(a) Color Image (b) GuideNet (C) NLSPN (d) PENet (e) ACMNet (f) PointDC (ours)

Figure 6. **Qualitative depth completion results on the KITTI DC Dataset [43].** (a) Color image, (b) GuideNet [41], (d) NLSPN [31], (d) PENet [16], (e) ACMNet [54], (f) PointDC (ours).



(a) RMSE on different number of sampling points

(b) REL on different number of sampling points.

Figure 7. Comparison with existing methods under different number of sampled points on NYU depth v2 [37], including CSPN [5], NLSPN [31], GraphCSPN [27] and CostDCNet [21].

Guideformer [34] is the first method which introduces transformer [44] for depth completion. Compare to Guideformer [34], our method achieves better performance in terms of MAE, iRMSE, and iMAE metrics.

We display three examples in Fig. 6 for qualitative comparisons. In the first and second row, PointDC recovers clearer details such as the bicycle and the bars. The example in the third row is more challenging, but PointDC still achieves better results, especially around the car window. Both qualitative and quantitative analyses demonstrate that PointDC attains competitive results compared to other state-of-the-arts.

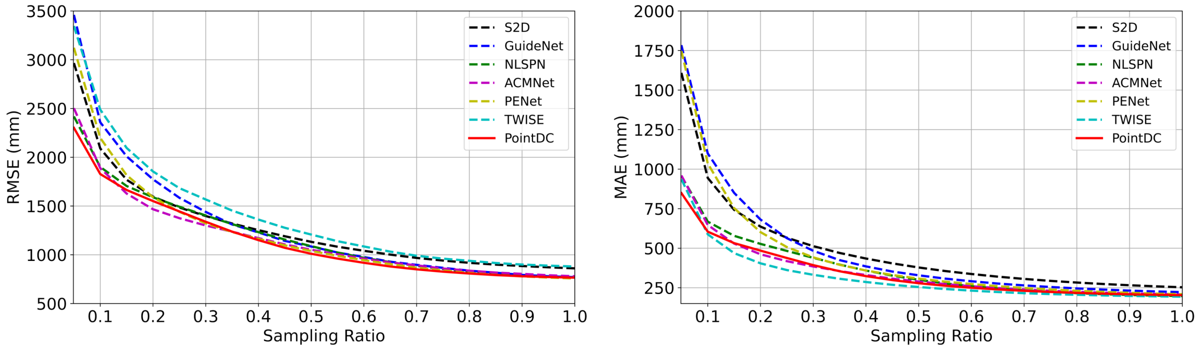
5.6. Generalization Capability

To validate the generalizability of PointDC, we carry out extensive experiments: (1) different sparsity levels of the input depth map. (2) cross-dataset evaluation.

Different sparsity levels. In practice, the number of sparse points is different for various scenarios. To compare the performance under different sparsity levels of the input depth map, we train PointDC on a certain setting and then evaluate on other sparsity levels. For the indoor NYU depth v2 dataset [37], we change the number of sampled points from 100 to 1000 with a step size of 100. We compare our PointDC with CSPN [5], NLSPN [31], GraphCSPN [27], and CostDCNet [21]. The results are displayed in Fig. 7. In terms of RMSE and REL metrics, it is observed that PointDC exceeds other methods on all the spar-

Method	RMSE ↓	REL ↓	$\delta_1 \downarrow$	$\delta_2 \downarrow$	$\delta_3 \downarrow$
CSPN [5]	0.729	0.504	69.1	77.8	84.0
NLSPN [31]	0.093	**0.020**	**98.9**	**99.6**	99.7
CostDCNet [21]	0.119	0.033	98.1	99.3	99.6
GraphCSPN [27]	0.094	0.023	98.8	99.6	99.7
PointDC	**0.092**	0.023	**98.9**	**99.6**	**99.8**

Table 6. Cross-dataset evaluation performance on the SUN RGBD Dataset collected by Kinect V1. The metrics RMSE and REL are presented in meter (m)



(a) RMSE under different sampling ratios.

(b) MAE under different sampling ratios.

Figure 8. Comparison with existing methods under different sample ratios on the valiation set of KITTI Depth Completion [37], including CSPN [5], NLSPN [31], GraphCSPN [27] and CostDCNet [21].

Method	RMSE ↓	REL ↓	δ_1 ↓	δ_2 ↓	δ_3 ↓
CSPN [5]	0.490	0.179	84.5	91.5	95.1
NLSPN [31]	**0.128**	**0.015**	99.0	**99.7**	**99.9**
CostDCNet [21]	0.207	0.028	97.8	99.1	99.5
GraphCSPN [27]	0.131	0.017	99.0	**99.7**	**99.9**
PointDC	**0.128**	0.016	**99.1**	**99.7**	**99.9**

Table 7. Cross-dataset evaluation performance on the SUN RGBD Dataset collected by Xtion. The metrics RMSE and REL are presented in meter (m)

sity levels. This demonstrates the generalizability of our PointDC in indoor scenes. For the KITTI Depth Completion Dataset [43], we uniformly sub-sample the raw LiDAR depth by different ratios from 1 to 0.1 with a step size of 0.1 and display the results in Fig 8. The minimum value of the x axis is 0.05. The comparison mathods include S2D [30], GuideNet [41], NLSPN [31], ACMNet [54], PENet [16], and TWISE [18]. In Fig. 8, at the beginning, PointDC is inferior to PENet [16], however, our model achieves better results when the sample ratio decreases, especially at 0.1 and 0.05. Although the accuracy of PointDC is a little inferior to CMNet [54] when sample ratio is 0.2 and 0.3, the overall performance of PointDC is better than all the methods. These results demonstrate PointDC is robust to different sparsity levels of the input depth maps.

Cross-dataset Evaluation. To validate cross-dataset performance, we train PointDC on the NYUv2 dataset and then test on the SUN RGBD dataset [38] directly. The evaluation results on the dataset captured by Kinect V1 are listed in Table 6 and the dataset captured by Xtion are listed in Table 7.Compared to Table 1, the accuracy of all the methods drops due to different camera and depth sensors. As the datas captured by Xtion come from a different device, the accuracy decreases more. However, we can observe from the results that our model still achieves the best performance in terms of RMSE. For REL, our model achieves competitive results with NLSPN [31] and outperforms all the other methods. These above analyses demonstrate the strong cross-dataset generalizability of our PointDC.

Method	Parameters (M)↓	FLOPs (G) ↓
PointDC	25.098	108.89
NLSPN [31]	26.4	542.2

Table 8. Computational analysis, which is measured with inputs of resolution 228 × 304.

5.7. Computational Cost

We list the parameters and FLOPs of PointDC in Table 8, with these of NLSPN for comparison. Although the parameters of PointDC are similar to NLSPN, PointDC requires much less FLOPs (G). The reason is that NLSPN finishes the information propagation by numerous iterations while out PointDC can achieve it directly.

6. Conclusion

We propose a feature point cloud aggregation framework that extracts both 2D and sparse 3D features for depth completion. It reconstructs the depth information for a target location by the adjacent sparse 3D feature points, in which each location can capture 3D information from the sparse 3D features directly regardless of the sparsity levels of the input depth maps. Based on this, we build a neural network called as PointDC. We experimentally show that our PointDC achieves better or comparable results compared to state-of-the-art depth completion methods. In addition, our PointDC demonstrates strong generalization performance with respect to the different sparsity levels of the input depth and cross-data evaluation.

Acknowledgement. This work was supported in part by the Ten Thousand Talents Program of Zhejiang Province under grant 2020R52003 and in part by the National Natural Science Foundation of China under grant 62002323.

References

[1] Alexandre Boulch, Gilles Puy, and Renaud Marlet. FKAConv: Feature-Kernel Alignment for Point Cloud Convolution. In *Asian Conference on Computer Vision*, 2020. 2, 3

[2] Si-Yuan Cao, Runmin Zhang, Lun Luo, Beinan Yu, Zehua Sheng, Junwei Li, and Hui-Liang Shen. Recurrent homography estimation using homography-guided image warping and focus transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9833–9842, 2023. 3

[3] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 357–366. IEEE, 2021. 3

[4] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10023–10032, 2019. 2, 3, 5, 6, 7

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision*, pages 103–119, 2018. 2, 3, 6, 7, 8, 9

[6] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2361–2379, 2019. 2, 3

[7] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, 2021. 1

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[9] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, et al. Depthlab: Real-time 3d interaction with depth maps for mobile augmented reality. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, pages 829–843, 2020. 1

[10] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. *arXiv preprint arXiv:1805.11913*, 2018. 1, 2

[11] Yanping Fu, Qingan Yan, Long Yang, Jie Liao, and Chunxia Xiao. Texture mapping for 3d reconstruction with rgb-d sensor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4645–4653, 2018. 1

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 5

[13] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020. 2, 3

[14] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 2, 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. 5

[16] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Towards precise and efficient image guided depth completion. In *IEEE International Conference on Robotics and Automation*. IEEE, 2021. 2, 7, 8, 9

[17] Lam Huynh, Phong Nguyen, Jiří Matas, Esa Rahtu, and Janne Heikkilä. Boosting monocular depth estimation with lightweight 3d point fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12767–12776, 2021. 2, 3, 5, 6, 7

[18] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2021. 6, 7, 9

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

[20] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *International Conference on 3D Vision*, pages 52–60, 2018. 1, 2, 3

[21] Jaewon Kam, Jungeon Kim, Soongjin Kim, Jaesik Park, and Seungyong Lee. Costdcnet: Cost volume based depth completion for a single rgb-d image. In *Proceedings of the European Conference on Computer Vision*, pages 257–274, 2022. 2, 3, 6, 7, 8, 9

[22] Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. Optimizing depth perception in virtual and augmented reality through gaze-contingent stereo rendering. *ACM Transactions on Graphics*, 39(6):1–10, 2020. 1

[23] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13916–13925, 2021. 2, 3, 6

[24] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 4804–4814. IEEE, 2022. 3

[25] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9131–9140. IEEE, 2020. 3

[26] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[27] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *Proceedings of the European Conference on Computer Vision*, pages 90–107, 2022. 2, 3, 5, 6, 7, 8, 9

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022. IEEE, 2021. 3

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[30] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation*, 2018. 2, 6, 9

[31] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Proceedings of the European Conference on Computer Vision*, pages 120–136, 2020. 1, 2, 3, 5, 6, 7, 8, 9

[32] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019. 3, 6, 7

[33] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 3

[34] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6250–6259, 2022. 3, 7, 8

[35] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 1

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 2, 3, 5

[37] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *Proceedings of the European Conference on Computer Vision*, 7576:746–760, 2012. 5, 6, 8, 9

[38] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 5, 9

[39] Zhenbo Song, Jianfeng Lu, Yazhou Yao, and Jian Zhang. Self-supervised depth completion from direct visual-lidar odometry in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11654–11665, 2021. 1

[40] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272. IEEE, 2021. 3

[41] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020. 2, 3, 6, 7, 8, 9

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

[43] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision*, 2017. 5, 7, 8, 9

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 8

[45] Haowen Wang, Mingyuan Wang, Zhengping Che, Zhiyuan Xu, Xiuquan Qiao, Mengshi Qi, Feifei Feng, and Jian Tang. Rgb-depth fusion gan for indoor depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6209–6218, 2022. 5

[46] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018. 2, 3

[47] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 1

[48] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5418–5427, 2022. 1

[49] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1625–1632, 2013. 5

[50] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130. IEEE, 2022. 3

[51] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2811–2820, 2019. 3, 6, 7

[52] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *Proceedings of the European Conference on Computer Vision*, pages 214–230, 2022. 2, 3, 5, 6, 7

[53] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15890–15899, 2021. 3

[54] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021. 2, 3, 6, 7, 8, 9

[55] Lingting Zhu, Xian Liu, Xuanyu Liu, Rui Qian, Ziwei Liu, and Lequan Yu. Taming diffusion models for audio-driven co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10544–10553, June 2023. 1