GraphCSPN: Geometry-Aware Depth Completion via Dynamic GCNs

Xin Liu[1], Xiaofei Shao[2], Bo Wang[2], Yali Li[1], and Shengjin Wang[1⋆]

[1] Beijing National Research Center for Information Science and Technology (BNRist)
Department of Electronic Engineering, Tsinghua University
xinliu20@mails.tsinghua.edu.cn, {liyali13, wgsgj}@tsinghua.edu.cn
[2] Deptrum Ltd.
{xiaofei.shao, bo.wang}@deptrum.com

Abstract. Image guided depth completion aims to recover per-pixel dense depth maps from sparse depth measurements with the help of aligned color images, which has a wide range of applications from robotics to autonomous driving. However, the 3D nature of sparse-to-dense depth completion has not been fully explored by previous methods. In this work, we propose a **Graph C**onvolution based **S**patial **P**ropagation **N**etwork (**GraphCSPN**) as a general approach for depth completion. First, unlike previous methods, we leverage convolution neural networks as well as graph neural networks in a complementary way for geometric representation learning. In addition, the proposed networks explicitly incorporate learnable geometric constraints to regularize the propagation process performed in three-dimensional space rather than in two-dimensional plane. Furthermore, we construct the graph utilizing sequences of feature patches, and update it dynamically with an edge attention module during propagation, so as to better capture both the local neighboring features and global relationships over long distance. Extensive experiments on both indoor NYU-Depth-v2 and outdoor KITTI datasets demonstrate that our method achieves the state-of-the-art performance, especially when compared in the case of using only a few propagation steps. Code and models are available at the project page [3].

Keywords: Depth completion, Graph neural network, Spatial propagation

1 Introduction

Depth perception plays an important role in various real-world applications of computer vision, such as navigation of robotics [8,27] and autonomous vehicles [1,11], augmented reality [5,6], and 3D face recognition [16,35]. However, it is difficult to directly acquire dense depth maps using depth sensors, including LiDAR, time-of-flight or structure-light-based 3D cameras, either because of the

⋆ Corresponding author
[3] https://github.com/xinliu20/GraphCSPN_ECCV2022

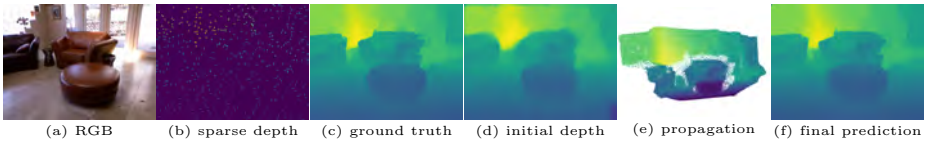


Fig. 1. Illustration of depth completion task using our framework. A backbone model receives the sparse depth map and corresponding RGB image as input and outputs an initial depth prediction. And then the initial depth is iteratively refined by our geometry-aware GraphCSPN in 3D space to produce the final depth prediction. Sparse depth map (b) has less than 1% valid values and is dilated for visualization.

inherent limitations of hardware or due to the interference of surrounding environment. Since depth sensors can only provide sparse depth measurements of object at distance, there has been a growing interest within both academia and industry in reconstructing depth in full resolution with the guidance of corresponding color images.

To address this challenging problem of sparse-to-dense depth completion, a wide variety of methods have been proposed. Early approaches [46,41,36] mainly focus on handcrafted features which often lead to inaccurate results and have poor generalization ability. Recent advance in deep convolutional neural networks (CNN) has demonstrated its promising performance on the task of depth completion [4,37,20]. Although CNN based methods have already achieved impressive results for depth completion, the inherent local connection property of CNN makes it difficult to work on depth map with sparse and irregular distribution, and hence fail to capture 3D geometric features. Inspired by graph neural networks (GNN) that can operate on irregular data represented by a graph, we propose a geometry-aware and dynamically constructed GNN. And it is combined with CNN in a complementary way for geometric representation learning, in order to fully explore the 3D nature of depth prediction.

Among the state-of-the-art methods for depth completion, spatial propagation [32] based models achieve better results and are more efficient and interpretable than direct depth completion models [33]. Convolutional spatial propagation network (CSPN) [4] and other methods built on it [3,37] learn the initial depth prediction and affinity matrix for neighboring pixels, and then iteratively refine the depth prediction through recurrent convolutional operations. Recently, Park *et al.* [37] propose a non-local spatial propagation network (NLSPN) which alleviates the mixed-depth problem on object boundaries. Nevertheless, there are several limitations regarding to such approaches. Firstly, the neighbors and affinity matrix are both fixed during the entire iterative propagation process, which may lead to incorrect predictions because of the propagation of errors in refinement module. In addition, the previous spatial propagation based methods perform propagation in two dimensional plane without geometric constraints, neglecting the 3D nature of depth estimation. Moreover, they suffer from the problem of demanding numerous steps (*e.g.*, 24) of iteration to obtain accu-

rate results. The long iteration process indicates the inefficiency of information propagation and may limit their real-world applications.

To address the limitations stated above, we relax those restrictions and generalize all previous spatial propagation based methods into a unified framework leveraging graph neural networks. The motivation behind our proposed model is not only because GNN is capable of working on irregular data in 3D space, but also the message passing principle [15] of GNN is strongly in accord with the process of spatial propagation. We adopt an encoder-decoder architecture as a simple while effective multi-modality fusion strategy to learn the joint representation of RGB and depth images, which is utilized to construct the graph. Then the graph propagation is performed in 3D space under learnable geometric constraints with neighbors updated dynamically for every step. Furthermore, to facilitate the propagation process, we propose an edge attention module to aggregate information from corresponding position of neighboring patches. In summary, the main contributions of the paper are as follows:

- We propose a graph convolution based spatial propagation network for sparse-to-dense depth completion. It is a generic and propagation-efficient framework and only requires 3 or less propagation steps compared with 18 or more steps used in previous methods.
- We develop a geometry-aware and dynamically constructed graph neural network with an edge attention module. The proposed model provides new insights on how GNN can help to deal with 2D images in 3D perception related tasks.
- Extensive experiments are conducted on both indoor NYU-Depth-v2 and outdoor KITTI datasets which show that our method achieves better results than previous state-of-the-art approaches.

2 Related Work

Depth Completion Image guided depth completion is an important subfield of depth estimation, which aims to predict dense depth maps from various input information with different modalities. However, depth estimation from only a single RGB image often leads to unreliable results due to the inherent ambiguity of depth prediction from images. To attain a robust and accurate estimation, Ma and Karaman [33] proposed a deep regression model for depth completion, which boosts the accuracy of prediction by a large margin compared to using only RGB images. To address the problems of image guided depth completion, various deep learning based methods have been proposed – *e.g.*, sparse invariant convolution [22,9,23], confidence propagation [10,18], multi-modality fusion [43,21], utilizing Bayesian networks [39] and unsupervised learning [48], exploiting semantic segmentation [26,40] and surface normal [38,50] as auxiliary tasks.

Spatial Propagation Network The spatial propagation network (SPN) proposed in [32] can learn semantically-aware affinity matrix for vision tasks including depth completion. The propagation of SPN is performed sequentially in a row-wise and column-wise manner with a three-way connection, which can only

capture limited local features in an inefficient way. Cheng *et al*. [4] applied SPN on the task of depth completion and proposed a convolutional spatial propagation network (CSPN), which performs propagation with a manner of recurrent convolutional operation and alleviates the inefficiency problem of SPN. Later, CSPN++ [3] was proposed to learn context aware and resource aware convolutional spatial propagation networks and improves the accuracy and efficiency of depth completion. Recently, Park *et al*. [37] proposed NLSPN to learn deformable kernels for propagation which is robust to mixed-depth problem on depth boundaries. Following this family of approaches based on spatial propagation, we further propose a graph convolution based spatial propagation network (GraphCSPN) which provides a generic framework for depth completion. Unlike previous methods, GraphCSPN is constructed dynamically by learned patch-wise affinities and performs efficient propagation with geometrically relevant neighbors in three-dimensional space rather than in two-dimensional plane.

Graph Neural Network Graph neural networks (GNNs) receive a set of nodes as input and are invariant to permutations of the node sequence. GNNs work directly on graph-based data and capture dependency of objects via message passing between nodes [52,15,30]. GNNs have been applied in various vision tasks, such as image classification [12,47], object detection [19,17] and visual question answering [44,34]. Unlike previous depth completion methods using GNNs for multi-modality fusion [51], learning dynamic kernel [49], we leverage GNNs as its message passing principle is in accord with spatial propagation. In addition, we develop a geometry-aware and dynamically constructed GCN with edge attention to aggregate and update information from neighboring nodes.

3 Method

In this section, we start by introducing the spatial propagation network (SPN) and previous methods that build on SPN. To address the limitations of those methods, we present our graph convolution based spatial propagation network and show how it extends and generalizes earlier approaches into a unified framework. Then we describe in details every component of the proposed framework, including graph construction, neighborhood estimation and graph propagation. Furthermore, a theoretical analysis of our method from the perspective of anisotropic diffusion is provided in supplementary material.

3.1 Spatial Propagation Network

In the task of sparse-to-dense depth completion, spatial propagation network [32] is designed to be a refine module working on the initial depth prediction in a recursive manner. The initial depth prediction can be the output of an encoder-decoder network or other networks utilizing more complicated multi-modality fusion strategies. After several iteration steps, the final prediction result is obtained with more detailed and accurate structure. We formulate the updating

process of previous methods [32,4,3,37] in each propagation step as follows:

$$d_{p,q}^{s+1}=\mu(\mathbf{D}^s|\mathbf{A}_{p,q},\mathcal{N}_{p,q})=a_{p,q}^{p,q}d_{p,q}^{s}+\sum_{(i,j)\in\mathcal{N}_{p,q}}a_{p,q}^{i,j}d_{i,j}^{s}, \tag{1}$$

where s denotes the iteration step; $d_{p,q}\in\mathbf{D}$ is the depth value at coordinate (p,q) of 2D depth map; $(i,j)\in\mathcal{N}_{p,q}$ indicates the coordinate of neighbors at (p,q); and $a_{p,q}^{i,j}\in\mathbf{A}_{p,q}$ represents the affinity between the pixels at (p,q) and (i,j). During the spatial propagation, each depth value is updated by its neighbors according to the affinity matrix which defines how much information should be passed between neighboring pixels. Previous methods based on SPN construct the neighborhood through simple coordinate shift, which can be summarized as:

$$\mathcal{N}_{p,q}=\{(p+m,q+n)|(m,n)\in\Gamma(\mathbf{D}|p,q),(m,n)\neq(0,0)\}, \tag{2}$$

where (m,n) represents the coordinate shift of neighbors; and Γ denotes the estimation function of neighbors given the initial depth prediction $\mathbf{D}$ and position (p,q) as input. For local spatial propagation methods, the neighbor estimation function Γ is irrelevant to the input and falls to the fixed coordinate set, *e.g.* $\{-1,0,1\}$ for 3×3 kernel. The original SPN [32] performs propagation sequentially in a row-wise and column-wise manner with three-way connections which is inefficient. CSPN [4] updates the depth estimation by making use of all local neighbors simultaneously. CSPN++ [3] is able to learn square kernel of different sizes. The neighborhood estimation function of NLSPN [37] can learn non-local neighbors according to the pixel positions in 2D plane.

3.2 GraphCSPN

After closely examining the updating function and neighborhood construction function of earlier approaches, there are two major limitations that need to be addressed. Firstly, once the affinity matrix $\mathbf{A}_{p,q}$ and neighborhood matrix $\mathcal{N}_{p,q}$ is determined, they will not be changed during the entire process of spatial propagation. So there are only simple iterations and no learning process involved when propagating neighbor observations with corresponding affinities. However, the condition of pixels will be changed after each propagation step, and simple iterations with fixed affinities and neighbors fails to capture the dynamic relationships between pixels. As a result, the fixed configurations slow down the information propagation and demand a large number of iteration steps. And that's why we develop a dynamically constructed graph convolution network for propagation-efficient depth completion.

The second limitation of previous approaches is that the spatial propagation is confined in a small region and performed in two-dimensional plane without any geometric constraints. Structured kernels (*e.g.* 3×3, 5×5) used in CSPN [2] and CSPN++ [3] have a very limited receptive field for every iteration and can easily bring unrelated information into propagation from fixed neighbors, which may result in inaccurate predictions especially on object boundaries. Although

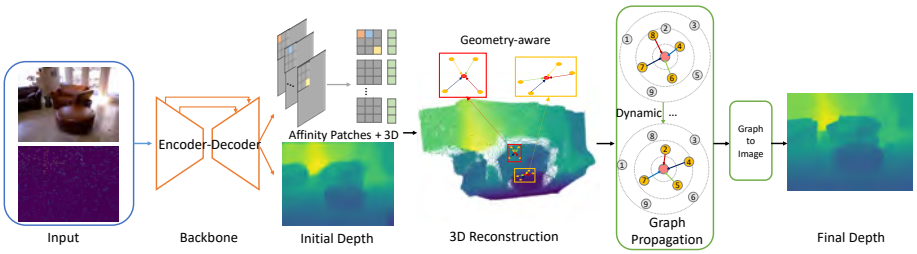


Fig. 2. Overview of the proposed network architecture (Best viewed in color). There are mainly two parts of the network. The first part utilizes an encoder-decoder to jointly learn the initial depth map and affinity matrix, which is sampled and reshaped into sequences of patches and concatenated with 3D position embeddings. The second part of our model estimates the neighbors of different patches on the basis of learned geometric constraints, and performs spatial propagation leveraging dynamic graph convolution networks with self-attention mechanism. The final depth prediction is obtained through a graph-to-image module. As can be seen in the parts of 3D reconstruction and graph propagation, the neighborhood construction in yellow is varying for different red nodes since the geometric structures surrounding red nodes are diverse in 3D space, and the graphs are also changeable because of the dynamic construction of GCN during the propagation process. Note that the different colors of the edges indicate the different attention weights between the red nodes and their yellow neighbors.

NLSPN [37] can predict flexible neighbors, the neighbors are learned merely based on the initial depth estimation without explicit geometric constraints, thus leading to unreliable predictions. To fully explore the 3D nature of depth prediction, the model we propose is geometry-aware and capable of capturing relevant neighbors over long distance, which is critical for depth completion because of the irregular distribution of sparse depth measurements. We now explain the process of GraphCSPN in three consecutive steps: graph construction, neighborhood estimation and graph propagation.

Patch-wise Graph Construction Graph neural networks take as input a set of node features. So the core of graph construction is to convert affinity maps $\mathbf{A} \in \mathbb{R}^{H\times W\times C}$ into a sequence of features $\mathbf{A}^{seq} \in \mathbb{R}^{N\times L}$, where (H, W) is the resolution of the affinity map, C is the number of channels, L is the length of node feature, and N is the total number of nodes. Note that the computation complexity of graph propagation increases quadratically with the number of nodes N. So we need to keep the size of graph small. To do so, we first convert $\mathbf{A}$ into a sequence of patches $\mathbf{A}^{patch} \in \mathbb{R}^{N\times P_h\times P_w}$, where (P_h, P_w) is the resolution of each affinity patch. In this way, the total number of nodes $N = (H\times W)/(P_h\times P_w)$. Otherwise, $N = H\times W$, if we use pixel-wise construction (see more discussions in supplement). For each patch $\mathbf{A}^{patch}$ with shape $P_h\times P_w$, the number of corresponding patches in $\mathbf{A}$ is C, and the pixels of each patch $\mathbf{A}^{patch}$ is from C different channels ($C = P_h\times P_w$). By our patch-wise construction, prior knowledge about local correlations can be established. And then each patch is

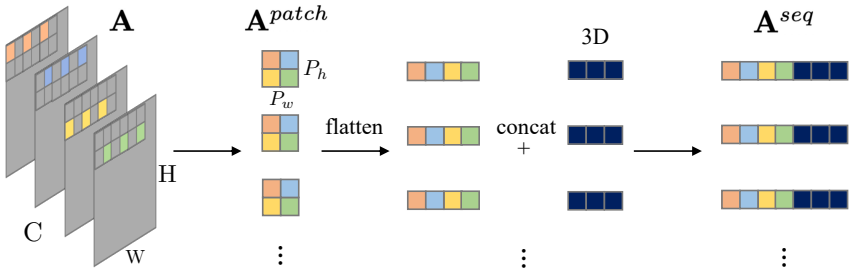


Fig. 3. Illustration of the process of graph construction (Best viewed in color). For better visibility, we set $C = 4, P_h = P_w = 2$, and use this simple example to show how to convert affinity maps into a sequence of features.

flattened and concatenated with 3D position as node features ($L = P_h \times P_w + 3$), so the model is also able to capture global relationships of different patches over long distance, as can be seen in the 3D reconstruction part of Fig.2. To help for a better understanding of graph construction, we use a simple example to illustrate the above process in Fig. 3.

Geometry-aware Neighborhood Estimation The 3D position embedding preserves the original geometry of the local shape. To explicitly incorporate geometric constraints into the graph propagation, the 3D position embeddings are supposed to be added to patches such that the geometric information can be retained during graph propagation. To do so, we project the depth prediction after each graph propagation step into 3D space as follows:

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = d \begin{bmatrix} 1/f_p & 0 & -c_p/f_p & 0 \\ 0 & 1/f_q & -c_q/f_q & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \\ 1/d \end{bmatrix}, \tag{3}$$

where (p, q) and (u, v, w) are the center coordinate of the patch in 2D plane and projected 3D space, respectively; d is the depth predictions during propagations; f_p, f_q, c_p, c_q are the camera intrinsic parameters. After obtaining the 3D position of each patch, k-nearest neighbors algorithm can be applied for neighborhood estimation in 3D space, which can be formulated as follows:

$$\mathcal{N}_{p,q}^{G} = \{(u, v, w) | (u, v, w) = \mathcal{K}(\mathbf{D}|p, q), (u, v, w) \in \mathbb{R}^3\}, \tag{4}$$

where $\mathcal{K}$ represents the projection and estimation function; $\mathcal{N}_{p,q}^{G}$ denotes the neighbors of the patch in 3D space. During the propagation process, the graph is dynamically constructed at each propagation step in line with the neighborhood estimation at that time.

Dynamic Graph Propagation After the neighborhood for each patch is constructed, patches are represented as vertices in a graph, and edges can be established if there are connections between the vertices. Graph propagation is

conducted via feature aggregation and feature update, which is as follows:

$$\mathbf{h}_{v_{s+1}} = \eta\left(\mathbf{h}_{v_s}, \rho\left(\left\{\mathbf{h}_{u_s} \mid u_s \in \mathcal{N}_{s+1}\left(v_s\right)\right\}, \mathbf{h}_{v_s}, \mathcal{W}_\rho\right), \mathcal{W}_\eta\right), \tag{5}$$

where s denotes the propagation step as before in Eqn. 1; ρ and η are the feature aggregation function and update function, $\mathcal{W}_\rho$ and $\mathcal{W}_\eta$ are the learnable parameters for the two functions, respectively; $\mathbf{h}_{v_s}$ represents the features for the vertex v_s; and $\mathcal{N}_{s+1}\left(v_s\right)$ is the set of neighbors for v_s during the step $s+1$. Note that the neighborhood $\mathcal{N}_{s+1}\left(v_s\right)$ varies for different propagation steps, since the graph is dynamically constructed. To further facilitate the graph propagation process, we propose a fine-grained edge attention module as follows:

$$\mathbf{x}_i^{s+1} = \alpha_{i,i}\phi_i(\mathbf{x}_i^s) + \sum_{j \in \mathcal{N}_{s+1}(i)} \alpha_{i,j}\phi_j(\mathbf{x}_i^s || (\mathbf{x}_j^s - \mathbf{x}_i^s)), \tag{6}$$

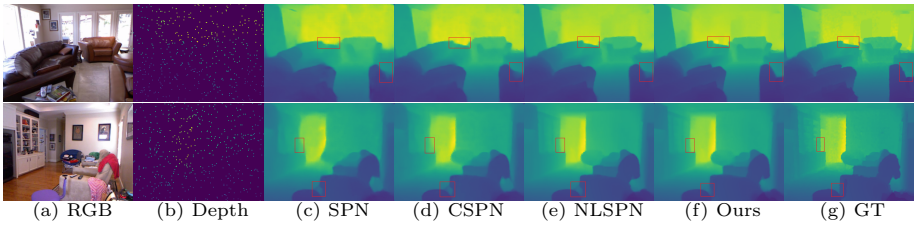
where $\mathbf{x}_i^s$ represents the feature vector for the vertex i at propagtion step s; $\|$ denotes the concatenation operation; ϕ is a neural network which is instantiated as a multi-layer perceptron (MLP) parameterized by $\mathcal{W}_\rho$. The attention is performed in a channel-wise manner and each channel denotes a different position in the original patch, and the attention coefficients $\alpha_{i,j}$ are computed as follows:

$$\alpha_{i,j} = \frac{\exp\left(\psi(\mathbf{x}_i || \mathbf{x}_j)\right)}{\sum_{k \in \mathcal{N}_{s+1}(i) \cup \{i\}} \exp\left(\psi(\mathbf{x}_i || \mathbf{x}_k)\right)}, \tag{7}$$

where ψ denotes a MLP parameterized by $\mathcal{W}_\eta$. Eqn. 6 details the computation process of aggregation function ρ and update function η stated in Eqn. 5. ρ is instantiated as a MLP ϕ which aggregates features of vertices from their neighbors. And η works through the attention coefficient α computed by Eqn. 7. By applying the proposed edge attention module, the different attention weights assigned to neighbors can effectively impede the propagation of incorrect predictions during the iterative refinement. After graph propagation, the graph-to-image module rearranges the sequences of patches to their original position in 2D image to generate the final prediction.

3.3 Overall Architecture

As can be seen in Figure 2, the end-to-end network architecture of our model consists of two parts: an encoder-decoder backbone to learn initial depth prediction and affinity maps, and a graph convolution network for spatial propagation. Follow the common practice of previous work [3,37], we build the encoder upon residual networks and add mirror connections with the decoder to make up the lost spatial information due to the down sampling and pooling operations in the encoder. Since the input of the encoder-decoder comes from different sensing modalities, depth-wise separable convolution is utilized for the first convolution layer which performs convolution on RGB images and sparse depth map independently, followed by a pointwise convolution as a simple while effective multi-modality fusion strategy. The second part of the model is a geometry-aware graph convolution network equipped with edge attention, which is dynamically constructed for iterative refinement of depth predictions.



(a) RGB (b) Depth (c) SPN (d) CSPN (e) NLSPN (f) Ours (g) GT

Fig. 4. Depth completion results on the NYU-Depth-v2 dataset [42]. The area inside the red bounding box reveals that our model recovers more accurate structures. (Better viewed in color and zoom in.)

3.4 Loss Function

The model is trained with masked ℓ_1 loss between the ground truth depth map and predicted depth map, which is the same loss function used in previous methods [33,4,37] and defined as follows:

$$\mathcal{L}(\mathbf{D}^{gt}, \mathbf{D}^{pred}) = \frac{1}{N_v} \sum_{i,j} \mathbb{I}(d^{gt}_{i,j} > 0) \left| \, d^{gt}_{i,j} - d^{pred}_{i,j} \, \right|, \tag{8}$$

where $\mathbb{I}$ is the indication function; $d^{gt}_{i,j} \in \mathbf{D}^{gt}$ and $d^{pred}_{i,j} \in \mathbf{D}^{pred}$ represent the ground truth depth map and the depth map predicted by our model; and N_v denotes the total number of valid depth pixels. We have explored other loss functions, such as ℓ_2 loss and smooth ℓ_1 loss, and also find that the training process can be accelerated by adding extra supervision as weighted auxiliary loss to the output of each graph propagation. More details about the setting of loss functions can be found in ablation.

4 Experiments

We conduct extensive experiments and ablation studies to verify the effectiveness of our model. In this section, we first briefly introduce the datasets and evaluation metrics. Then we present qualitative and quantitative comparisons with state-of-the-arts, and ablation studies of each component of our model are also provided. Additional experiment details can be found in supplementary materials.

4.1 Datasets and Metrics

NYU-Depth-v2 The NYU-Depth-v2 dataset [42] provides RGB and depth images of 464 different indoor scenes with the Kinect sensor. We use the official split of data, where 249 scenes are used for training and the remaining 215 scenes for testing. And we sample about 50K images out of the training set and the original frames of size 640×480 are first downsampled to half and then

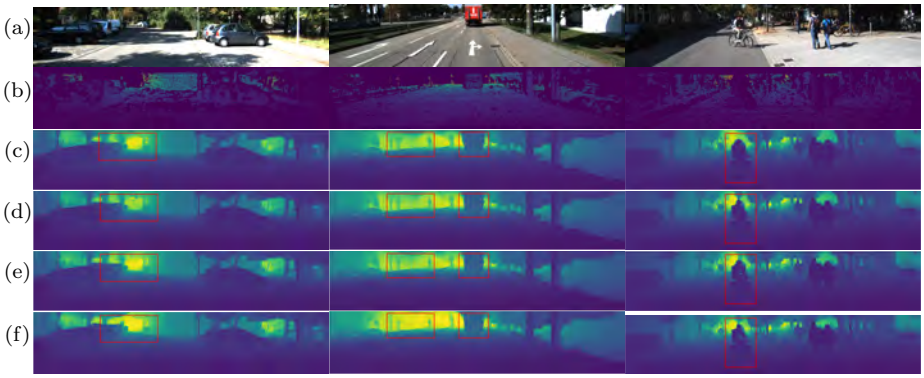


Fig. 5. Depth completion results on the KITTI Depth Completion dataset [45]. (a) RGB, (b) Ground truth, (c) CSPN [4], (d) DepthNormal [50], (e) NLSPN [37], (f) Ours. (Better viewed in color and zoom in.)

304×228 center-cropping is applied, as seen in previous work [28,7,33]. Following the standard setting [33,4], the official test dataset with 654 images is used for evaluating the final performance.

KITTI Dataset KITTI dataset [14,13] is a large self-driving real-world dataset with over 90k paired RGB images and LiDAR depth measurements. We use two versions of KITTI dataset, and one is from [33], which consists of 46k images from the training sequences for training, and a random subset of 3200 images from the test sequences for evaluation. The other is KITTI Depth Completion dataset [45] which provides 86k training, 7k validation and 1k testing depth maps with corresponding raw LiDAR scans and reference images. And we evaluate and test the model on the official selected validation set and test set.

Metrics We adopt the same metrics used in previous work [33,4,45] on NYU-Depth-v2 dataset and KITTI dataset. Given ground truth depth map $D^* = \{d^*\}$ and predicted depth map $D = \{d\}$, the metrics used in the following experiments include:

(1)RMSE: $\sqrt{\frac{1}{|D|}\sum_{d\in D}|d^* - d|^2}$;
(2)MAE: $\frac{1}{|D|}\sum_{d\in D}|d^* - d|$;
(3)iRMSE: $\sqrt{\frac{1}{|D|}\sum_{d\in D}|1/d^* - 1/d|^2}$;
(4)iMAE: $\frac{1}{|D|}\sum_{d\in D}|1/d^* - 1/d|$;
(5)REL: $\frac{1}{|D|}\sum_{d\in D}|d^* - d|/d^*$;
(6)δ_t: percentage of pixels satisfying $max(\frac{d^*}{d}, \frac{d}{d^*}) < t$, where $t \in \{1.25, 1.25^2, 1.25^3\}$.

4.2 Comparison with State-of-the-arts

We evaluate our model on the official test split of NYU-Depth-v2 dataset and 500 depth pixels are randomly sampled from the dense depth map, as in pre-

Methods	Iterations	RMSE ↓	REL ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
S2D [33]	-	0.230	0.044	97.1	99.4	99.8
SPN [32]	3	0.215	0.040	94.2	97.6	98.9
	24	0.172	0.031	98.3	99.7	99.9
CSPN [4]	3	0.135	0.021	97.1	98.1	99.3
	24	0.117	0.016	99.2	99.9	100.0
CSPN++ [3]	24	0.116	-	-	-	-
NLSPN [37]	3	0.119	0.018	98.3	99.4	99.3
	18	0.092	0.012	99.6	99.9	100.0
GraphCSPN	**3**	**0.090**	**0.012**	**99.6**	**99.9**	**100.0**

Table 1. Quantitative evaluation on the NYU-Depth-v2 dataset. We highlight the best results when propagating for only 3 steps.

Methods	Iterations	RMSE ↓	REL ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
S2D [33]	-	3.378	0.073	93.5	97.6	98.9
SPN [32]	3	3.302	0.069	93.7	97.6	99.0
	24	3.243	0.063	94.3	97.8	99.1
CSPN [4]	3	3.125	0.052	94.1	97.8	99.0
	24	2.977	0.044	95.7	98.0	99.1
NLSPN [37]	3	2.697	0.042	95.8	98.0	99.1
	18	2.533	0.038	96.2	98.5	99.3
GraphCSPN	**3**	**2.267**	**0.032**	**97.4**	**98.9**	**99.5**

Table 2. Quantitative evaluation on the KITTI dataset.

vious works [33,4,37]. Quantitative comparison results with methods based on spatial propagation networks are provided in Table 1. S2D [33] is a direct depth completion model which can only obtain blurry results. And spatial propagation networks based approaches are able to generate much more accurate and reliable results by recurrently refining the predictions. However, earlier methods have not taken the geometric constraints into account and fail to capture global relationship over long distance, thus easily leading to unreliable results on object boundaries where a holistic view is need, as can be seen in Figure 4. Moreover, the affinity matrix and neighborhood of previous methods are fixed during propagation, and a large number of iteration steps is required because of the inefficient propagation. The results in Table 1 show that our model achieves significant improvements over previous approaches when compared using only three iteration steps.

The Table 2 shows the comparisons of evaluation results on KITTI dataset [33]. Similar to the result on NYU-Depth-v2 dataset, our model outperforms the previous methods by a large margin which reduces 0.4m in RMSE. And our model also achieves better results compared to other geometry-aware methods [25,38,50] and other methods using GNNs [49,51], as can be seen in Table 3. On the challenging KITTI Depth Completion dataset [45] where the ground truth of depth map is also sparse. Our model still attains competitive results

Method	RMSE	REL	$\delta_{1.25}$	$\delta_{1.25^2}$
DepthCoeff [25]	0.118	0.013	99.4	99.9
DeepLiDAR [38]	0.115	0.022	99.3	99.9
DepthNormal [50]	0.112	0.018	99.5	99.9
GNN [49]	0.106	0.016	99.5	99.9
FCFRNet [31]	0.106	0.015	99.5	99.9
ACMNet [51]	0.105	0.015	99.4	99.9
PRNet [29]	0.104	0.014	99.4	99.9
GuideNet [43]	0.101	0.015	99.5	99.9
TWICE [24]	0.097	0.013	99.6	99.9
GraphCSPN	**0.090**	**0.012**	**99.6**	**99.9**

Table 3. Comparison on NYU-Depth-v2 dataset with other state-of-the-arts.

Method	RMSE	MAE	iRMSE	iMAE
CSPN [4]	1019.64	279.46	2.93	1.15
TWICE [24]	840.20	**195.58**	2.08	**0.82**
DepthNormal [50]	777.05	235.17	2.42	1.13
DeepLiDAR [38]	758.38	226.50	2.56	1.15
CSPN++ [3]	743.69	209.28	2.07	0.90
NLSPN [37]	741.68	199.59	1.99	0.84
GuideNet [43]	736.24	218.83	2.25	0.99
FCFRNet [31]	735.81	217.15	2.20	0.98
PENet [20]	**730.08**	210.55	2.17	0.94
GraphCSPN	738.41	199.31	**1.96**	0.84

Table 4. Comparison on KITTI Depth Completion test dataset.

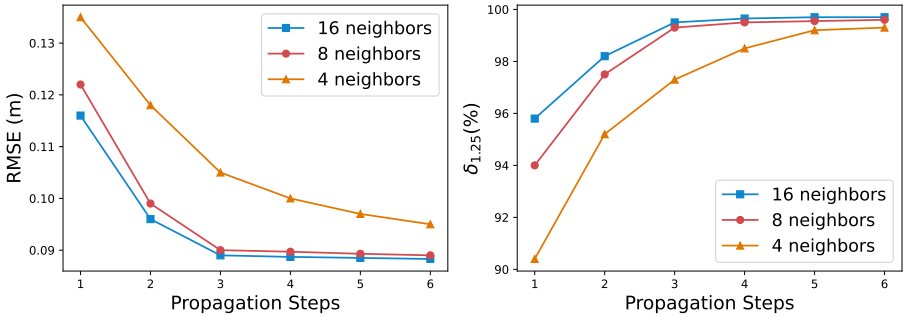


Fig. 6. Impact of number of propagation steps and neighbors on the prediction accuracy on NYU-Depth-v2 dataset.

compared to other state-of-the-arts, as can be seen in Table 4. Qualitative results of the proposed method in comparison with state-of-the-arts are shown in Figure 5. We also provide a video demo of our method using this dataset in supplement.

4.3 Ablation Studies

We conduct extensive experiments to verify the effectiveness of each proposed component. The altered components of our model include: iteration steps, the number of neighbors, the sparsity of depth samples, loss functions, the proposed edge attention and geometry-aware modules, and dynamic graph construction.

Propagation Configurations: There are two important factors in our proposed graph convolution based spatial propagation network, which are the iteration steps and the number of neighbors. To investigate the impact of those factors on final results, we set the iteration steps from 1 to 6 and the number of neighbors to 4, 8, 16. The results in Figure 6 indicate that it is difficult to

loss	epochs	RMSE	REL
ℓ_1	40	0.094	0.013
smooth ℓ_1	42	0.102	0.014
ℓ_2	46	0.104	0.015
auxillary loss	36	0.090	0.012

Table 5. Ablation study on the choices of loss function.

attention	geometry	dynamic	RMSE	REL
	✓	✓	0.101	0.014
✓		✓	0.113	0.017
✓	✓		0.108	0.016
✓	✓	✓	0.090	0.012

Table 6. Ablation study on the configurations of graph construction.

initialization	epochs	RMSE	REL
raw input	50	0.134	0.022

Table 7. Ablation study on the initialization of graph propagation.

sparsity	200	400	500	600	800
RMSE	0.119	0.104	0.090	0.087	0.082

Table 8. Ablation study on the number of sparse depth samples.

aggregate enough information when propagating for only one step, so the results get worse. And when propagating for six steps which is larger than our original model, the model only achieves slightly better results, which implies that a small number of iteration steps is already adequate since our model is propagation-efficient. As for the size of neighborhood, we find that a larger number of neighbors can slightly improve the results while the results degrade when there are only limited neighbors, which also takes more epochs to converge. So a combination of larger number of propagation steps and neighbors can achieves better performance which demonstrate the capacity and potential of our model, but we choose a small number instead for a balance of accuracy and efficiency. Since GNN can directly work on the irregular input data, we also try to apply the graph propagation to the raw input. The results in Table 7 show it needs more steps to converge and leads to poor performance. The learned initial depth map can be viewed as a good starting point for a fast optimization process. Moreover, the encoder-decoder architecture to generate the initial depth is an effective fusion network to learn the joint representation of multi-modality RGB and depth images. And it can work in a complementary way with GNN.

Loss Functions: Our model is driven by ℓ_1 loss. Here we use ℓ_2 loss and smooth ℓ_1 loss to study the effect of different loss functions, as they play a critical role in model training. Although it is expected that a better result in RMSE would be obtained when applying ℓ_2 loss, in experiments we do not see the expected gains and find that it takes more epochs to reach results close to ℓ_1 loss, because ℓ_2 loss is sensitive to outliers and results in very small gradients during the late stage of training. Smooth ℓ_1 loss is a compromise between ℓ_2 loss and ℓ_1 loss, but ℓ_1 loss is a more robust and suitable choice for the task of depth prediction. We also examine the impact of auxillary loss on model training and final results by adding extra supervision to the intermediate outputs of each graph propagation with weight 0.1. The results in Table 5 show that model training benefits from the auxillary loss with a faster speed of convergence, because the intermediate outputs share the same objective with the final output.

Geometry, Attention and Dynamic Construction: In this part, we verify the role of each component of our graph propagation including the geometry-aware module, edge attention, and dynamic construction. Our model explicitly incorporates geometric constraints into the process of propagation. After removing those constraints, the neighborhood estimation can only perform in the feature space of patches with no access to the knowledge of their real locations in 3D space. As a result, the performance drops by a large margin which implies the importance of 3D geometric clues in the task of depth completion. We further remove the edge attention module from our framework and use the mean aggregation function instead to verify the necessity of attention-driven aggregation. As shown in Table 6, the performance of the model without edge attention also decreases with a small degree, because the edge attention module can effectively impede the propagation of errors in refinement module. In addition, when the graph propagation is not constructed dynamically and shares the same neighborhood estimation during all the propagation steps, we can see the performance gets worse and converges at an earlier epoch. Because there is an inherent over-smoothing problem of vanilla graph convolution networks, and the dynamic construction of graph is capable of preventing such problem and helps to achieve more accurate results.

Robustness and Generalization: Following the standard procedure, our original model works on sparse depth map with 500 random samples. To evaluate the robustness of our model to different input sparsity, we test our model using sparse depth map with 200 samples which takes only 0.4% of the dense depth map. Although the performance decreases as expected, the model can still generate reasonable results and when changing the sparse input to 800 samples, the model attains a result of 0.083m evaluated by RMSE metric, which demonstrates our model is robust and generalizes well to different input sparsity. Please refer to the supplement for additional ablation studies and visualizations.

5 Conclusion

In this paper, we have proposed a graph convolution based spatial propagation network for sparse-to-dense depth completion. The proposed method generalizes previous spatial propagation based approaches into a unified framework which is geometry-aware and propagation-efficient. The graph propagation is dynamically constructed and performed with an edge attention module for feature aggregation and update. Extensive experiments demonstrate the effectiveness of the proposed method. Since our model is a generic and effective solution for 3D spatial propagation, it can be further extended to more 3D perception related tasks in the future.

Acknowledgement. This work was supported by the state key development program in 14th Five-Year under Grant Nos.2021QY1702, 2021YFF0602103, 2021YFF0602102. We also thank for the research fund under Grant No. 2019GQG0001 from the Institute for Guo Qiang, Tsinghua University.

References

1. Bai, L., Zhao, Y., Elhousni, M., Huang, X.: Depthnet: Real-time lidar point cloud depth completion for autonomous vehicles. IEEE Access **8**, 227825–227833 (2020)
2. Chen, Y., Yang, B., Liang, M., Urtasun, R.: Learning joint 2d-3d representations for depth completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10023–10032 (2019)
3. Cheng, X., Wang, P., Guan, C., Yang, R.: Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 10615–10622 (2020)
4. Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 103–119 (2018)
5. Diaz, C., Walker, M., Szafir, D.A., Szafir, D.: Designing for depth perceptions in augmented reality. In: 2017 IEEE international symposium on mixed and augmented reality (ISMAR). pp. 111–122. IEEE (2017)
6. Du, R., Turner, E., Dzitsiuk, M., Prasso, L., Duarte, I., Dourgarian, J., Afonso, J., Pascoal, J., Gladstone, J., Cruces, N., et al.: Depthlab: Real-time 3d interaction with depth maps for mobile augmented reality. In: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology. pp. 829–843 (2020)
7. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. arXiv preprint arXiv:1406.2283 (2014)
8. El-laithy, R.A., Huang, J., Yeh, M.: Study on the use of microsoft kinect for robotics applications. In: Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium. pp. 1280–1288. IEEE (2012)
9. Eldesokey, A., Felsberg, M., Khan, F.S.: Propagating confidences through cnns for sparse data regression. arXiv preprint arXiv:1805.11913 (2018)
10. Eldesokey, A., Felsberg, M., Khan, F.S.: Confidence propagation through cnns for guided sparse depth regression. IEEE transactions on pattern analysis and machine intelligence **42**(10), 2423–2436 (2019)
11. Farahnakian, F., Heikkonen, J.: Rgb-depth fusion framework for object detection in autonomous vehicles. In: 2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS). pp. 1–6. IEEE (2020)
12. Garcia, V., Bruna, J.: Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043 (2017)
13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)
15. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International conference on machine learning. pp. 1263–1272. PMLR (2017)
16. Gordon, G.G.: Face recognition based on depth and curvature features. In: Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 808–809. IEEE Computer Society (1992)
17. Gu, J., Hu, H., Wang, L., Wei, Y., Dai, J.: Learning region features for object detection. In: Proceedings of the european conference on computer vision (ECCV). pp. 381–395 (2018)

18. Hekmatian, H., Jin, J., Al-Stouhi, S.: Conf-net: Toward high-confidence dense 3d point-cloud with error-map prediction. arXiv preprint arXiv:1907.10148 (2019)
19. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3588–3597 (2018)
20. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Penet: Towards precise and efficient image guided depth completion. arXiv preprint arXiv:2103.00783 (2021)
21. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Towards precise and efficient image guided depth completion. arXiv preprint arXiv:2103.00783 (2021)
22. Hua, J., Gong, X.: A normalized convolutional neural network for guided sparse depth upsampling. In: IJCAI. pp. 2283–2290. Stockholm, Sweden (2018)
23. Huang, Z., Fan, J., Cheng, S., Yi, S., Wang, X., Li, H.: Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. IEEE Transactions on Image Processing **29**, 3429–3441 (2019)
24. Imran, S., Liu, X., Morris, D.: Depth completion with twin surface extrapolation at occlusion boundaries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2583–2592 (2021)
25. Imran, S., Long, Y., Liu, X., Morris, D.: Depth coefficients for depth completion. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12438–12447. IEEE (2019)
26. Jaritz, M., De Charette, R., Wirbel, E., Perrotton, X., Nashashibi, F.: Sparse and dense data with cnns: Depth completion and semantic segmentation. In: 2018 International Conference on 3D Vision (3DV). pp. 52–60. IEEE (2018)
27. Jing, C., Potgieter, J., Noble, F., Wang, R.: A comparison and analysis of rgb-d cameras' depth performance for robotics application. In: 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). pp. 1–6. IEEE (2017)
28. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV). pp. 239–248. IEEE (2016)
29. Lee, B.U., Lee, K., Kweon, I.S.: Depth completion using plane-residual representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13916–13925 (2021)
30. Li, G., Müller, M., Qian, G., Perez, I.C.D., Abualshour, A., Thabet, A.K., Ghanem, B.: Deepgcns: Making gcns go as deep as cnns. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
31. Liu, L., Song, X., Lyu, X., Diao, J., Wang, M., Liu, Y., Zhang, L.: Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 2136–2144 (2021)
32. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. arXiv preprint arXiv:1710.01020 (2017)
33. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 4796–4803. IEEE (2018)
34. Narasimhan, M., Lazebnik, S., Schwing, A.G.: Out of the box: Reasoning with graph convolution nets for factual visual question answering. arXiv preprint arXiv:1811.00538 (2018)
35. Pan, G., Han, S., Wu, Z., Wang, Y.: 3d face recognition using mapped depth images. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops. pp. 175–175. IEEE (2005)

36. Park, J., Kim, H., Tai, Y.W., Brown, M.S., Kweon, I.S.: High-quality depth map upsampling and completion for rgb-d cameras. IEEE Transactions on Image Processing **23**(12), 5559–5572 (2014)
37. Park, J., Joo, K., Hu, Z., Liu, C.K., Kweon, I.S.: Non-local spatial propagation network for depth completion. arXiv preprint arXiv:2007.10042 **3**(8) (2020)
38. Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., Pollefeys, M.: 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. pp. 3313–3322 (2019)
39. Qu, C., Liu, W., Taylor, C.J.: Bayesian deep basis fitting for depth completion with uncertainty. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16147–16157 (October 2021)
40. Schneider, N., Schneider, L., Pinggera, P., Franke, U., Pollefeys, M., Stiller, C.: Semantically guided depth upsampling. In: German conference on pattern recognition. pp. 37–48. Springer (2016)
41. Shen, J., Cheung, S.C.S.: Layer depth denoising and completion for structured-light rgb-d cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1187–1194 (2013)
42. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European conference on computer vision. pp. 746–760. Springer (2012)
43. Tang, J., Tian, F.P., Feng, W., Li, J., Tan, P.: Learning guided convolutional network for depth completion. IEEE Transactions on Image Processing **30**, 1116–1129 (2020)
44. Teney, D., Liu, L., van Den Hengel, A.: Graph-structured representations for visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2017)
45. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: 2017 international conference on 3D Vision (3DV). pp. 11–20. IEEE (2017)
46. Wang, L., Jin, H., Yang, R., Gong, M.: Stereoscopic inpainting: Joint color and depth completion from stereo images. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
47. Wang, X., Ye, Y., Gupta, A.: Zero-shot recognition via semantic embeddings and knowledge graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6857–6866 (2018)
48. Wong, A., Soatto, S.: Unsupervised depth completion with calibrated backprojection layers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12747–12756 (October 2021)
49. Xiong, X., Xiong, H., Xian, K., Zhao, C., Cao, Z., Li, X.: Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In: European Conference on Computer Vision (ECCV) (2020)
50. Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., Li, H.: Depth completion from sparse lidar data with depth-normal constraints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2811–2820 (2019)
51. Zhao, S., Gong, M., Fu, H., Tao, D.: Adaptive context-aware multi-modal network for depth completion. arXiv preprint arXiv:2008.10833 (2020)
52. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)