CostDCNet: Cost Volume based Depth Completion for a Single RGB-D Image

Jaewon Kam, Jungeon Kim, Soongjin Kim,
Jaesik Park, and Seungyong Lee

POSTECH
{jwkam95, jungeonkim, kimsj0302, jaesik.park, leesy}@postech.ac.kr

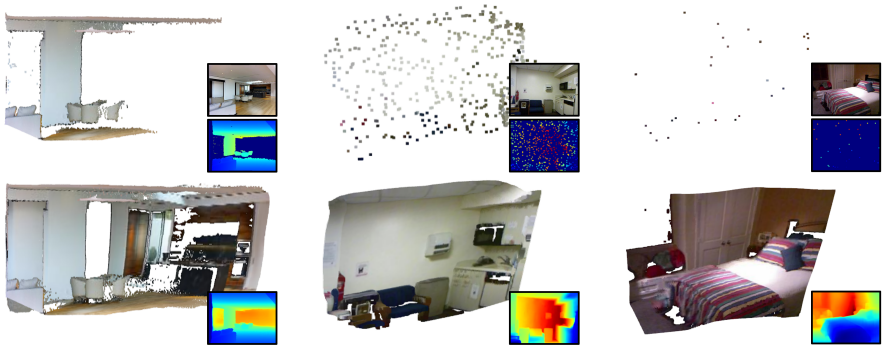
Abstract. Successful depth completion from a single RGB-D image requires both extracting plentiful 2D and 3D features and merging these heterogeneous features appropriately. We propose a novel depth completion framework, *CostDCNet*, based on the cost volume-based depth estimation approach that has been successfully employed for multi-view stereo (MVS). The key to high-quality depth map estimation in the approach is constructing an accurate cost volume. To produce a quality cost volume tailored to single-view depth completion, we present a simple but effective architecture that can fully exploit the 3D information, three options to make an RGB-D feature volume, and per-plane pixel shuffle for efficient volume upsampling. Our *CostDCNet* framework consists of lightweight deep neural networks (∼1.8M parameters), running in real time (∼30ms). Nevertheless, thanks to our simple but effective design, *CostDCNet* demonstrates depth completion results comparable to or better than the state-of-the-art methods.

Keywords: Depth completion, cost volume, 3D convolution, single RGB-D image

1 Introduction

Recently, RGB-D cameras have been widely used in many applications that need 3D geometry information, such as augmented reality (AR), virtual reality (VR), autonomous driving, and robotics. However, various depth sensors, including LiDAR, Kinect, and RealSense, suffer from missing measurements. Depth completion is the task of filling missing areas in depth images obtained from sensors.

Learning-based depth completion methods mainly employ 2D convolutions to extract RGB and depth feature maps from an input RGB-D image, regarding the depth image as a 2D image. Then, they fuse two heterogeneous 2D feature maps in the 2D feature space to infer a completed depth image [18, 2, 26, 31, 25, 33]. However, these 2D convolutions do not directly consider depth-axis information of 3D positions. To fully exploit the 3D geometry information, 3D convolutions could be an alternative, but naïvely applying standard 3D convolutions to input 3D points would be inappropriate due to their sparsity and irregularity.



(a) Semi-dense DC (b) Sparse DC (c) Extremely sparse DC

Fig. 1. Our results of depth completion for three different types of input depth. The top row visualizes the input RGB-D images and their colored point clouds, and the bottom row shows our completed depth images and their colored point clouds.

Multi-view stereo (MVS) and stereo matching methods that use deep neural networks have taken a cost volume concept to infer depth by considering 3D spatial information [14, 13, 45, 37]. These methods have shown compelling depth estimation accuracy. In the methods, a cost volume is constructed in the multiple-depth-plane representation (Figure 3b), and is commonly regarded as containing matching costs between RGB images captured at different viewpoints. Inspired by MVS, some single-view depth completion methods using a cost volume have been proposed [22, 25]. They use 2D convolutional neural networks (CNNs) to directly predict the cost volume from a single RGB-D input. However, the predicted cost volume is produced only using cascaded 2D convolutions that cannot properly utilize 3D information from the input depth.

In this paper, we propose a cost volume based depth completion network (CostDCNet) that can fully exploit 3D information by performing 3D convolutions of heterogeneous 2D and 3D features in the multiple-depth-plane representation. Our method is basically based on the cost volume-based depth estimation framework in the MVS domain. However, unlike aforementioned depth completion methods [22, 25] that use multiple depth planes to represent an inferred depth, our framework uses multiple depth planes as a representation for a volume to be convolved. This approach enables our framework to use 3D convolutions like existing MVS studies. To infer a cost volume, MVS methods need an input feature volume that is commonly constructed by aggregating 2D feature maps of RGB images at multiple viewpoints. Similarly, our framework also requires a feature volume to generate a cost volume. We present three viable options to construct the feature volume, called an RGB-D feature volume, from a single RGB-D image. We experimentally showed that a proper design choice of an RGB-D feature volume is vital to produce a quality cost volume, consequently resulting in the high-quality completed depth. Various methods, including MVS, that rely on 3D convolutions typically suffer from memory and computational

complexity due to the volume data usage. To handle the problem, we adopt an approach to process low-resolution volumes and then to upsample them. For such volume upsampling, we use an adapted version of pixel shuffle [38], which we call per-plane pixel shuffle, that increases volume resolution by rearranging its feature values along spatial dimensions. The upsampling scheme performs only rearrangement operations, so it works in highly memory- and computation-efficient manner.

Our *CostDCNet* framework consists of three parts: (1) construction of RGB-D feature volume from RGB-D image, (2) prediction of cost volume by a modified 3D UNet, (3) final completed depth regression. Due to our effective network design that can exploit 3D geometry and RGB information both fully and collectively, our *CostDCNet*, albeit using the small number of network parameters (∼1.8M), outperforms or is comparable to state-of-the-art (SOTA) methods on both semi-dense and sparse depth datasets. Our codes are publicly available[1].

To summarize, our key contributions are as follows:

- We propose a single-view depth completion method that is based on but adapted from the cost volume-based depth estimation pipeline in the multi-view stereo (MVS) field.
- The proposed method can fully exploit heterogeneous 2D and 3D features of an input RGB-D image due to our scheme for producing RGB-D feature volume from a single RGB-D image, the multiple-depth-plane representation, and 3D convolution operation in the representation.
- We propose the depth completion framework, *CostDCNet*, that is merely composed of lightweight CNNs (∼1.8M parameters) and our efficient volume upsampling module without any complex architecture.
- Due to our highly effective and efficient network design, *CostDCNet* runs in real time (∼30ms) and achieves qualitative and quantitative results that are better than or comparable to SOTA methods in both semi-dense and sparse depth datasets.

2 Related Work

We review representative approaches that are closely related to the technical components in our framework.

Depth completion We classify depth completion studies that use a single RGB-D image as input into two classes according to the sparsity of the input depth that they target: semi-dense depth completion and sparse depth completion. We firstly review studies on semi-dense depth completion.

Zhang et al. [48] predicted surface normals and occlusion boundaries by using two 2D CNNs, then conducted global linear optimization to obtain completed depth. Using the surface normals and occlusion boundaries predicted by Zhang et al., Huang et al. [18] adopted a self-attention mechanism to conserve and

[1] https://github.com/kamse/CostDCNet.

sharpen structures in the inferred depth image. These methods train their models in a supervised manner by using rendered depth from 3D reconstructed scenes as ground truth (GT). However, the poor 3D reconstruction quality leads to inaccurate GT depth images. To avoid this problem, Cao et al. [2] introduced an adaptive sampling strategy for self-supervised training to simulate missing depths in the capturing process. They trained their model with self-supervision but showed comparable results to the supervised methods.

Early sparse depth completion studies that use deep learning regarded a depth image as a 2D image and used 2D CNNs that took a typical encoder-decoder architecture with minor variations [31, 27]. A few advanced methods focused on effectively extracting and fusing multi-modal features from RGB-D image [49, 11, 41]. Several methods that use spatial propagation networks (SPN) [6, 5, 33] iteratively refine initial depth regression by using affinity kernels. These methods that use 2D CNNs show highly encouraging results, but they require a large number of network parameters and might rely on pretrained models from other tasks. Rather than regarding depth images as 2D images, some methods [4, 19] attempted to consider 3D geometry information explicitly, by consolidating 2D and 3D features. To extract 3D features from an input depth, they used point cloud convolutions [42, 1]. However, in their methods, 3D spatial information is not fully utilized because they applied 3D convolutions only to valid pixels in depth. In addition, the methods are not suitable for semi-dense or dense depth, because they search neighbors by using the k-NN algorithm, which slows down as the number of input points increases.

In this paper, our proposed method can process both semi-dense and sparse depth completion classes, and provides comparable or better results than SOTA methods in both classes. It can fully exploit 3D geometry and context information by aggregating 2D and 3D features in a volumetric manner.

Cost volume based depth regression Multi-view depth estimation approaches, such as stereo matching and MVS, have been studied for inferring depth from two or multiple views. They usually build a cost volume, which contains the matching costs of all pixels computed for all possible disparities, to predict the depth. Various matching costs, such as sum of absolute difference (SAD), sum of squared difference (SSD), and normalized cross-correlation (NCC), have been used for building a cost volume. Matching costs have also been computed using deep neural networks [30, 46] due to the robustness. Recent methods [47, 37, 13, 14] aggregate the cost volume with a 3D CNN. However, since 3D convolution requires high amount of computation, it is not easy to operate in real time.

Inspired by MVS works, some single-view depth completion methods attempted to borrow the cost volume concept to regress depth [22, 25]. However, they could not directly calculate cost volume, because no other view is available for matching. Therefore, they inferred the cost volume by using 2D CNNs without matching between multi-views. However, since their neural network still is based on only 2D convolutions, they cannot deal with 3D information appropriately and only generate cost maps rather than a cost volume in 3D space. In

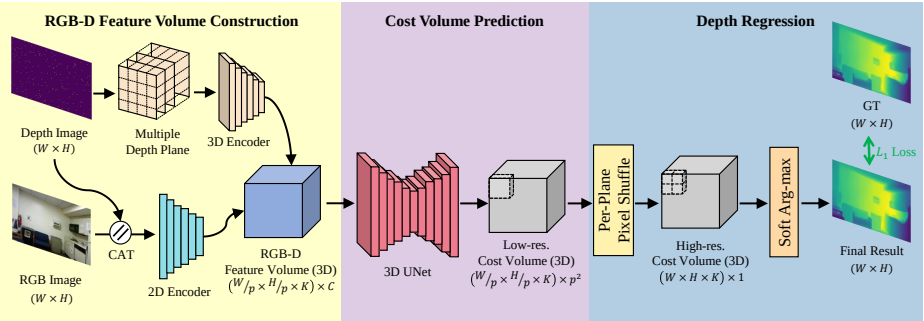


Fig. 2. Overview of *CostDCNet*. Our framework consists of three components. (1) RGB-D feature volume construction, (2) Cost volume prediction, (3) Depth regression.

contrast, we devise *CostDCNet* to effectively handle the 3D geometry information contained in the input depth using 3D convolution.

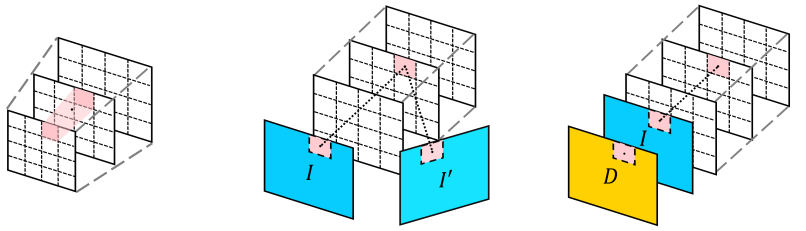
3 Cost Volume based Depth Completion Network

3.1 Overall Process and Network Design

CostDCNet infers a completed depth from a single RGB-D frame by sequentially performing three steps (Figure 2): (1) RGB-D feature volume construction (Section 3.2), (2) Cost volume prediction and (3) Depth regression (Section 3.3). We outline these steps below.

RGB-D feature volume construction To produce an RGB-D feature volume, first of all, we extract 2D and 3D feature maps from an input RGB-D image by using 2D and 3D encoders. Unlike the 2D encoder that directly uses an RGB-D image as input, the 3D encoder requires a depth image to be converted into a multiple-depth-plane representation (Figure 3c). We use 3D sparse convolution [8] to compute the 3D feature only for valid 3D points having non-zero depth. Both 2D and 3D encoders produce feature maps of the p times-reduced width and height to avoid the high memory footprint and heavy computation amount of a 3D CNN. Finally, we consolidate the reduced 2D and 3D feature maps into a single 3D feature volume, which we call an RGB-D feature volume (Section 3.2).

Cost volume prediction and depth regression We feed the produced RGB-D feature volume into our 3D CNN to obtain a cost volume. At this time, the 3D CNN has the architecture of a 3D UNet [9] and additionally adopts pseudo 3D convolutions [35] to relieve the computational overhead of standard 3D convolutions. Because the RGB-D feature volume has the width and height lower than the original image, the predicted cost volume also has a reduced resolution. To recover the reduced resolution to the original one, a naïve option



(a) View frustum (b) Cost volume (c) RGB-D feature volume

Fig. 3. Multiple-depth-plane representation. (a) View frustum in 3D space. (b) and (C) show a cost volume of multi-view stereo and our RGB-D feature volume in the multiple-depth-plane representation, respectively.

is to use 3D deconvolutions. However, feature volume upsampling by such 3D deconvolutions is computationally costly, so some multi-view depth estimation studies [20, 45, 37] commonly employ linear interpolation. We experimentally observed that the linear interpolation tends to cause blurry boundaries in the inferred depth image, as shown in Figure 7. Therefore, we instead opt for the shuffle scheme [38] to upsample the cost volume of reduced size.

Our per-plane pixel shuffle rearranges features of the cost volume to increase its spatial resolution. For example, let the resolution of a target upsampled volume be $W \times H \times D$, and the resolution of the reduced cost volume that the 3D UNet outputs be $\frac{W}{p} \times \frac{H}{p} \times D \times C$. Then, by setting the feature dimension C to p^2 and rearranging a p^2-dimensional feature of a cell of the cost volume to width and height dimensions, we obtain an upsampled cost volume of the resolution $W \times H \times D \times 1$. The method requires only feature vector rearrangement without heavy computation, and shows good visual quality while running fast (Section 4.3, Figure 7). Finally, we estimate the completed depth image by performing the soft-argmax based depth regression like [23] (Section 3.3).

3.2 RGB-D Feature Volume

An RGB-D feature volume is constructed by consolidating 2D and 3D features extracted from a single RGB-D frame, where the consolidation is performed in the multiple-depth-plane representation. In this section, we introduce the multiple-depth-plane representation, how to convert an input depth to the representation, and three different types of methods to merge 2D and 3D features in this representation.

Depth to multiple depth planes A pixel position $\mathbf{u} = (x, y)$ in a depth image $D(\cdot)$ is related to a 3D position $\mathbf{x} \in \mathbb{R}^3$ within a view frustum of the image as $\mathbf{x} = D(\mathbf{u})\mathbb{K}^{-1}\tilde{\mathbf{u}}$, where $\mathbb{K}$ is a 3×3 matrix that includes depth camera intrinsic parameters, $\tilde{\mathbf{u}}$ is homogeneous coordinates $[\mathbf{u}, 1]^T$ of $\mathbf{u}$, and $D(\mathbf{u})$ is the depth value at the pixel position $\mathbf{u}$ (Figure 3a). We can represent 3D position $\mathbf{x}$ as a 3D position in the multiple-depth-plane representation. Then, we quantize the

3D positions in the representation into a voxel grid so that 3D convolutions can be performed (Figure 3c).

To construct the voxel grid, we need to predefine only the number K of uniformly-spaced depth planes $\{d_k\}_{k=1:K}$ because x- and y-axes resolution can be naturally determined to be the same as in the depth image. Maximum and minimum positions of the planes are determined by considering maximum and minimum depth values that a depth sensor can measure. The resolution of the voxel grid then becomes $W \times H \times K$. The 3D position $\mathbf{x} = (x, y, D(x, y))$ of a depth image pixel in the representation is easily quantized to a cell $\mathbf{x}_c$ of the voxel grid as

$$\begin{aligned} \mathbf{x}_c &= (x, y, k') \\ k' &= \arg\min_k |D(x, y) - d_k|. \end{aligned} \tag{1}$$

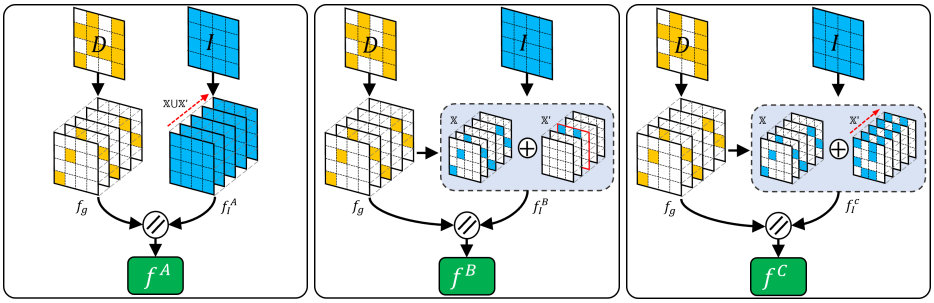
By applying Eq. (1) to all valid pixels having non-zero depths, we obtain a set of valid cells $\{\mathbf{x}_c^n\}_{n=1:N}$ in the voxel grid, where N is the number of valid cells. Then, we construct an input geometry feature volume $f_g^{in} \in \mathbb{R}^{W \times H \times K \times M}$ on the voxel grid as

$$f_g^{in}(\mathbf{x}') = \begin{cases} \mathbf{s}_n, & \mathbf{x}' \in \{\mathbf{x}_c^n\}_{n=1:N} \\ \vec{0}, & \text{otherwise} \end{cases}, \tag{2}$$

where $\mathbf{x}'$ is a cell within f_g^{in}. If $\mathbf{x}'$ corresponds to a valid depth pixel $\mathbf{x}_c^n$, we store $\mathbf{s}_n$ in $\mathbf{x}'$, where $\mathbf{s}_n$ is a M-dimensional feature vector. Otherwise, a M-dimensional zero vector is stored in $\mathbf{x}'$. We set M to one and $\mathbf{s}_n$ to $D(x, y) - d_{k'}$, which is the residual of a depth pixel from the nearest predefined depth plane. Consequently, we can cover a view frustum using a regular voxel grid in the multiple-depth-plane representation, thereby enabling standard 3D convolution operations to be directly applied (Figure 3). Performing standard 3D convolutions on the voxel grid has the effect of adjusting spatial coverage of a 3D convolution kernel accordingly to the distance from a depth camera.

Three types of RGB-D feature volume An RGB-D feature volume is a feature volume in the multiple-depth-plane representation, which is defined as element-wise concatenation of a geometric feature volume f_g and an image feature volume f_I. To obtain f_g, we first convert a depth image to the input geometry feature volume, and then feed it to our 3D encoder (Figure 2). To obtain f_I, we first extract a 2D feature map from the input RGB-D image using our 2D encoder. Then, the 2D feature map is placed into the multiple-depth-plane representation to form f_I in three different ways (Types A, B, C), where the type of f_I determines the type of the final RGB-D feature volume. Below is the description of the three different types.

(i) **Type A**. Several studies for 3D reconstruction, multi-view stereo, and 3D semantic segmentation generate a feature volume by unprojecting multi-view image features into a 3D volume, then integrating them [10, 32, 40]. In these



(a) Type A (b) Type B (c) Type C

Fig. 4. Three designs of RGB-D feature volume. The RGB-D feature volumes are represented by multiple depth planes and are classified into three types according to the structure of the image feature volume.

methods, all the volume cells that are intersected by a ray starting from the camera's center of projection through an image pixel accumulate the same feature of the pixel. Similarly, we generate an image feature volume f_I^A to be filled with the same image features along the depth-axis as follows:

$$f_I^A(x, y, k') = I(x, y),\ \forall k' \in \{k\}_{k=1:K}, \tag{3}$$

where I denotes a 2D image feature map calculated by feeding an input RGB-D image into a 2D encoder (Figure 2).

(ii) **Type B**. Since a Type A feature volume allocates image features to its all cells regardless of whether the corresponding depth pixels are valid or not, it does not consider 3D positional information of image features explicitly. In contrast, Type B considers the valid depth values of pixels for image feature allocation. To be specific, for a pixel with a valid depth, we allocate its 2D image feature to only the corresponding 3D cell in the image feature volume f_I^B. For pixels with invalid depths, we allocate their 2D image features to the corresponding cells in the middle depth plane of f_I^B. Formally,

$$f_I^B(x, y, k) = \begin{cases} I(x, y), & (x, y, k) \in \mathbb{X} \cup \mathbb{X}' \\ \vec{0}, & \text{otherwise} \end{cases}, \tag{4}$$

where $\mathbb{X}$ is the set of cells determined by Eq. (1) from the valid depth pixels. $\mathbb{X}'$ is the set of cells corresponding to pixels (x, y) with invalid depths, where k is set to $K/2$, the center depth plane.

(iii) **Type C**. A Type C feature volume is generated in the same manner as Type B except that the image feature of each pixel with invalid depth is repeatedly allocated to the cells of f_I^C, traversing along the depth-axis.

The three types of RGB-D feature volumes have the same geometric feature volume, but their image feature volumes are defined differently. We experimen-

tally demonstrated that using the Type C feature volume achieves the best performance (Section 4).

3.3 Final Depth Regression and Training Loss

Like the MVS works [23, 20], we can regress a completed depth map $D'(\cdot)$ by applying the softmax operator $\sigma(\,\cdot\,)$ to the upsampled cost volume $V_c \in \mathbb{R}^{W\times H\times K}$ along the depth-axis and using the following equation

$$D'(x,y) = \sum_{k=1}^{K} d_k \times \mathbf{p}^k_{x,y}, \quad \mathbf{p}_{x,y} = \sigma(V_c(x,y,:)), \tag{5}$$

where d_k is the predefined depth value of the k-th plane, (x, y) is an image pixel position, K is the number of depth planes, $V_c(x, y, :)$ is a K-dimensional vector along the depth-axis within the cost volume, and $\mathbf{p}_{x,y}$ is a probability vector obtained by the softmax operator $\sigma(\,\cdot\,)$ for K depth planes at (x, y).

We train our *CostDCNet* in an end-to-end manner by using only the L1 loss function as

$$L = \sum_{(x,y)\in\mathbb{G}} \left| D'(x,y) - D_{gt}(x,y) \right|, \tag{6}$$

where D_{gt} is the GT depth, D' is an inferred depth by Eq. (5), and $\mathbb{G}$ is a set of valid pixels of D_{gt}.

4 Experiments

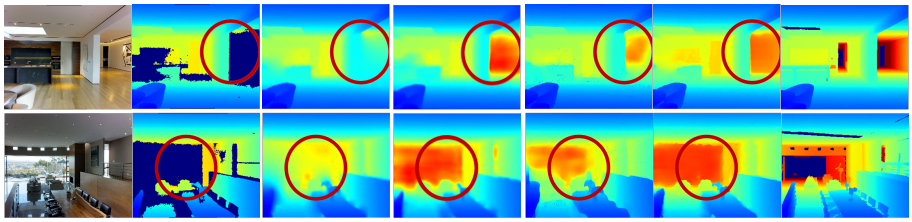
4.1 Experimental Setup

Datasets We evaluated semi-dense depth completion on Matterport3D [3] dataset and sparse depth completion on NYUv2 [39], VOID [43], and KITTI DC [12] datasets.

- Matterport3D is a large-scale indoor dataset, including 194,400 RGB-D images captured by a Matterport camera. With GT depth images provided by Zhang et al. [48], we used training (∼100K images) and test (474 images) sets as in [18]. We use images downsized into size of 320×256 for both training and testing.
- NYUv2 includes RGB-D videos of 464 indoor scenes collected using a Kinect sensor. We randomly sample points to obtain input sparse depth images. We downscaled images into size of 320×240 and then center-cropped them to be size of 304×228. We constructed the training set (∼48K images) and the test set (654 images) as in previous studies [31].
- VOID provides 640 × 480 RGB-D images containing sequences of 56 indoor and outdoor scenes acquired from RealSense D435i camera. We used sparse depth images with about 1500 depth points as input. We train our model on the training set (∼47K images) and test on the test set (800 images) by following the previous protocol of [43].

Table 1. Quantitative comparisons with SOTA semi-dense depth completion methods on the Matterport3D dataset. The numbers are excerpted from each paper except for [26], of which the number was reported by [2].

Method	RMSE(m)↓	MAE(m)↓	SSIM↑	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
MRF [15]	1.675	0.618	0.692	50.6	55.6	65.1	78.0	85.6
AD [28]	1.653	0.610	0.696	50.3	56.0	66.3	79.2	86.1
Zhang et al. [48]	1.316	0.461	0.762	65.7	70.8	78.1	85.1	88.8
Cao et al. [2]	1.187	0.385	0.736	66.5	72.5	79.9	87.1	91.1
Huang et al. [18]	1.092	0.342	0.799	66.1	75.0	85.0	91.1	93.6
MSG-CHN [26]	1.068	0.347	0.778	65.0	73.2	83.3	90.3	93.4
Ours	**1.019**	**0.290**	**0.838**	71.3	78.6	87.1	92.8	94.8



(a) RGB (b) Depth (c) Zhang [48](d) Huang [18] (e) Cao [2] (f) Ours (g) GT

Fig. 5. Qualitative comparisons with SOTA semi-dense depth completion methods on the Matterport3D dataset. The results of [48, 18] were borrowed from the paper [18] and results of [2] were taken from their project page.

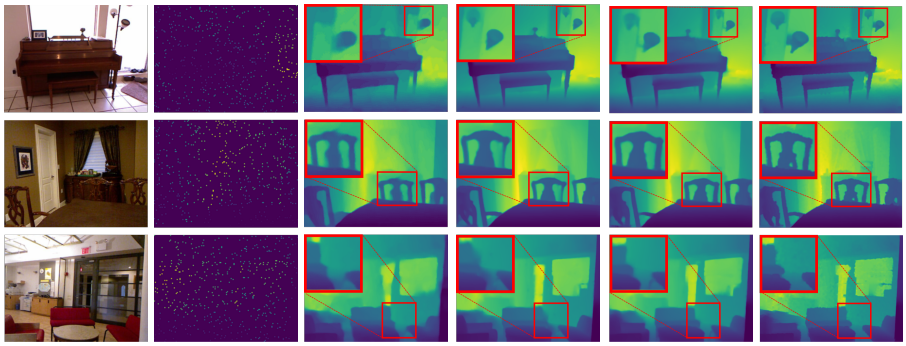
– KITTI DC is an outdoor scene dataset that provides paired RGB and sparse depth images obtained by projecting Velodyne LiDAR sensor measurements onto 2D space. It utilizes 11 consecutive frames to generate denser depth images as GT. For training, we center-cropped the images with size of 1216 × 240 to ignore the regions with no LiDAR measurements. We used ∼93K image pairs for training and 1K pairs for testing as in the previous works.

Evaluation metrics We followed the standard metrics to evaluate the performance: root mean squared error (RMSE), mean absolute error (MAE), structural similarity index map (SSIM), relative mean absolute error (REL), and percentages δ_x of inlier pixels with the less than x meters error.

Implementation detail Our 3D encoder and 2D encoder are composed of 3 and 6 residual blocks [16], respectively. We used the simplified 3D UNet [9] to extract cost volumes. For the detailed network architecture, refer to the supplementary material. We used $K = 16$ depth planes and $p^2 = 16$ feature dimension of a cost volume. The maximum depth d_{max} is set to 5m, 10m, 15m, and 90m for VOID, NYUv2, Matterport3D and KITTI DC datasets, respectively. We used the ADAM optimizer [24] with an initial learning rate of 0.5×10^{-3}, then divided it in half every 20 epochs for network training. We set the batch size to 16 and the training epoch to 50, and trained neural networks using a single NVIDIA RTX 3090 GPU.

Table 2. Quantitative comparisons with SOTA sparse depth completion methods on the NYUv2 dataset. The numbers are excerpted from respective papers.

Method	#points	#params↓	RMSE(m)↓	REL(m)↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
DCoeff [22]	500	45.7M	0.118	0.013	99.4	99.9	-
CSPN [7]		18.5M	0.117	0.016	99.2	99.9	100
CSPN++ [5]		28.8M	0.116	-	-	-	-
DeepLiDAR [34]		53.4M	0.115	0.022	99.3	99.9	100
PRNet [25]		14.3M	0.104	0.014	99.4	99.9	100
GuideNet [41]		63.3M	0.101	0.015	99.5	99.9	100
TWISE [21]		5.8M	0.097	0.013	99.6	99.9	100
NLSPN [33]		25.8M	0.092	**0.012**	99.6	99.9	100
Point-Fusion [19]		8.7M	**0.090**	0.014	99.6	99.9	100
Ours		**1.8M**	0.096	0.013	99.5	99.9	100
Point-Fusion [19]	32	8.7M	0.319	0.057	96.3	99.2	99.8
Ours		**1.8M**	**0.258**	**0.048**	96.4	99.1	99.7



(a) RGB (b) Sparse depth (c) CSPN [7] (d) NLSPN [33] (e) Ours (f) GT

Fig. 6. Qualitative results on NYUv2 dataset. The results of other methods [7, 33] are obtained from the authors' project page.

4.2 Comparisons with SOTA

Semi-dense depth completion We compared our method with previous semi-dense depth completion methods on the Matterport3D dataset. Table 1 shows quantitative comparisons with SOTA methods. Our network outperformed all other methods significantly in all metrics by a large margin. Especially, Hwang et al. [18], which is the previous SOTA method, additionally require the normal and boundary maps predicted by Zhang et al. [48] and their network size (19.8M) is 11 times larger than ours. MSG-CHN [26] is the backbone network used by Cao et al. [2], trained by supervised learning. It has slightly better RMSE compared to [18], but our method outperforms it across all metrics. In qualitative comparisons with other methods (Figure 5), our method better expresses details and has clearer boundaries.

Sparse depth completion We conducted experiments on the NYUv2 dataset to compare our framework with SOTA methods for the sparse depth completion task. Table 2 shows the quantitative results on a few standard metrics. When using 500 depth points as the input, our approach achieved the second and third

Table 3. Quantitative comparisons with SOTA sparse depth completion methods on VOID test set and KITTI DC validation set.

Dataset	Method	Train	#Param↓	Runtime↓	MAE↓	RMSE↓	iMAE↓	iRMSE↓
VOID	KBNet [44]	U	6.9M	**13ms**	39.80	95.86	21.16	49.72
	Ours	U	**1.8M**	30ms	**27.19**	**79.19**	**13.02**	**35.17**
	PENet [17]	S	132M	226ms	34.6	82.01	18.89	40.36
	NLSPN [33]	S	25.8M	122ms	26.7	79.12	12.70	33.88
	Ours	S	**1.8M**	**30ms**	**25.84**	**76.28**	**12.19**	**32.13**
KITTI DC	KBNet [44]	U	6.9M	**16ms**	260.44	1126.85	1.03	3.20
	Ours	U	**1.8M**	34ms	**242.64**	**868.62**	**0.99**	**2.39**

best performance on REL and RMSE metrics, respectively. However, our *Cost-DCNet* uses merely 20% of network parameters of Point-Fusion [19] and 7% of NLSPN [33], respectively. We also notice that our method obtained higher accuracy, compared to PRNet [25] and DCoeff [22] that utilize multiple depth planes as representation for an inferred depth map and perform only 2D convolutions. These results show that our framework exploits 3D information much better than PRNet and DCeoff. In addition, even for extremely sparse input depths (32 points), our method achieves SOTA performance in both Rel and RMSE metrics.

We also compared our method with SOTA unsupervised (KBNet) and supervised (NLSPN [33], PENet [17]) methods on VOID (test set) and KITTI DC (validation set) datasets. Most of unsupervised depth completion methods, including KBNet, do not use GT depths and train networks with additional photometric loss using adjacent views. For fast experiments, we did not use the photometric loss and simply trained our networks with the original input as GT and random samples from the original input as input data. We used L1 + L2 losses for KITTI DC datasets. Table 3 shows experimental results. The numbers of other methods are borrowed from the previous paper [29]. Our method outperforms SOTA unsupervised and supervised methods across all metrics, even running in real time (∼30FPS). While KBNet uses 3D position vectors to leverage 3D information, it is still based on 2D CNN. Therefore, its computational costs are lower than ours, but its inductive bias in 3D space could be weaker than 3D CNNs.

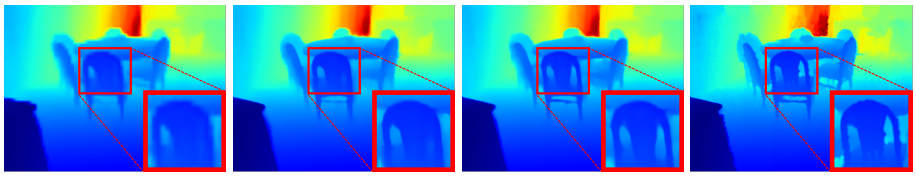
Collectively considering the overall performance, the total number of network parameters, robustness in extreme cases, we believe that our framework could be the best option.

4.3 Ablation Study

In this section, we analyze the role of each component of our framework. For the analysis, we conducted sparse depth completion on the NYUv2 dataset. To speed up the experiments, we set the resolution of input RGB-D image to $\frac{1}{4}$ size of the original before adding the encoders and trained each model up to 20 epochs.

Table 4. Ablation study for each component of our network. The results were evaluated on the NYUv2 dataset.

Method	Network			RGB-D Vol.				Encoder		Upsample			RMSE(m)↓	#param↓
	2D	3D	P3D	f_I^A	A	B	C	2D	3D	BL	BG	PS		
(a)	✓									✓			0.195	17.3M
(b)		✓			✓					✓			0.121	1.0M
(c)			✓		✓					✓			0.123	0.5M
(d)			✓	✓						✓			0.186	0.5M
(e)			✓			✓				✓			0.131	0.5M
(f)			✓				✓			✓			0.122	0.5M
(g)			✓				✓	✓		✓			0.108	1.5M
(h)			✓				✓	✓	✓	✓			0.106	1.8M
(i)			✓				✓	✓	✓		✓		0.102	3.1M
(j)			✓				✓	✓	✓			✓	0.099	1.8M



(a) (b) (c) (d)

Fig. 7. Different upsampling methods. (a) Bilinear upsampling, (b) Bilateral grid learning [45], (c) Per-plane pixel shuffle, and (d) Ground truth.

Network To verify the effect of inferring depth in 3D space, we compare our 3D UNet [9] with 2D UNet [36]. The 2D UNet takes a concatenated RGB-D image as input, and 3D UNet takes as input the RGB-D feature volume. When 2D UNet was replaced with our tiny 3D Unet, even though it has only 2.5% of network parameters of 2D UNet, the accuracy was significantly improved by ∼62% (Table 4a&b). The result implies that considering 3D information with the RGB-D feature volume as an appropriate representation is helpful for inferring high-quality cost volume. We also replaced the 3D convolution of 3D UNet with a pseudo-3D convolution (P3D) [35] to reduce the computational cost. P3D showed similar accuracy to 3D convolution while reducing the number of network parameters by half (Table 4b&c). Therefore, P3D was used in our 3D UNet.

Effectiveness of explicit positional conditioning To construct a RGB-D feature volume, a geometric feature volume f_g is element-wisely concatenated with an image feature volume f_I. We verify that exploiting such explicit positional cues from input depths in RGB-D feature volume construction results in performance improvement. To this end, we compared the results of using only the image feature volume f_I^A (without f_g) (Table 4d) and Type A (with f_g) (Table 4c). f_I^A leads to a poorer accuracy than Type A because it does not use 3D geometry information at all. It suggests that it is necessary to assign geometry features to the proper positions in 3D space for accurate depth completion.

Three designs of RGB-D feature volume We also quantified the effect of the design of RGB-D feature volume. Type A (Table 4c) and Type C (Table 4f) had higher accuracy than Type B (Table 4e). We argue 3D information can be inferred better by assigning 2D features of unknown depth values to all possible depth planes rather than to a specific depth plane. In addition, Type C had slightly higher accuracy than Type A. This result means that ambiguity can be reduced by assigning 2D features of valid depths to the correct depth positions.
2D and 3D encoders We evaluated the effectiveness of the 2D and 3D encoders (Table 4g&h). Compared to a model without encoders (Table 4f), the accuracy was significantly improved when our 2D and 3D encoders are added. These results indicate that the 2D and 3D encoders can enrich the information of our RGB-D feature volumes.
Upsampling We quantified the effect of upsampling layers, using bilinear (BL, Table 4h), bilateral grid learning (BG, Table 4i) [45], and per-plane pixel shuffle (PS, Table 4j) upsampling. BG and PS achieved better accuracy than BL. However, BG requires an additional network, which increases the number of parameters, whereas PS rearranges the volume, which is simple and fast. In qualitative results (Figure 7), BL generally blurred the boundary, BG gave clean boundaries, but caused artifacts that break the structure, whereas PS gives clear boundaries and restores the overall structure well. These results indicate PS can reliably up-scale cost volume to the original resolution at low computational cost.

5 Conclusion

In this paper, we proposed a single-view depth completion framework, *CostDCNet*, that can fully exploit 3D information by performing 3D convolutions of heterogeneous 2D and 3D features in the multiple-depth-plane representation. We introduced three designs of RGB-D feature volume to represent a single RGB-D image into 3D space. Furthermore, we employed the per-plane pixel shuffle to upsample the low-resolution cost volume on which 3D convolutions are performed efficiently. We demonstrated that our system is lightweight (∼1.8M parameters), runs in real time (∼30ms), and achieves qualitatively and quantitatively SOTA or comparable performance in depth completion for semi-dense depths, sparse depths (500 points), and even extremely sparse depths (32 points).
Limitation and future work Although our framework showed impressive results in indoor sparse and semi-dense depth completion, we have a few limitations. Our framework has a fixed number of depth planes, so in outdoor scenes such as the KITTI DC dataset, the distance between the planes becomes wider, and expressive power decreases. To tackle this problem, we plan to closely consolidate the sparse convolution and coarse-to-fine approach with our *CostDCNet*.

Acknowledgements This work was supported by the Ministry of Science and ICT, Korea, through IITP grants (SW Star Lab, 2015-0-00174; AI Innovation Hub, 2021-0-02068; Artificial Intelligence Graduate School Program (POSTECH), 2019-0-01906).

References

1. Boulch, A., Puy, G., Marlet, R.: Fkaconv: Feature-kernel alignment for point cloud convolution. In: Proceedings of Asian Conference on Computer Vision (ACCV) (2020) 4
2. Cao, Z., Li, A., Yuan, Z.: Self-supervised depth completion via adaptive sampling and relative consistency. In: Proceedings of IEEE International Conference on Image Processing (ICIP). pp. 3263–3267 (2021) 1, 4, 10, 11
3. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. arXiv preprint arXiv:1709.06158 (2017) 9
4. Chen, Y., Yang, B., Liang, M., Urtasun, R.: Learning joint 2D-3D representations for depth completion. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 10023–10032 (2019) 4
5. Cheng, X., Wang, P., Guan, C., Yang, R.: Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In: Proceedings of AAAI Conference on Artificial Intelligence. vol. 34, pp. 10615–10622 (2020) 4, 11
6. Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 103–119 (2018) 4
7. Cheng, X., Wang, P., Yang, R.: Learning depth with convolutional spatial propagation network. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **42**(10), 2361–2379 (2019) 11
8. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3075–3084 (2019) 5
9. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D u-net: learning dense volumetric segmentation from sparse annotation. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 424–432. Springer (2016) 5, 10, 13
10. Dai, A., Nießner, M.: 3dmv: Joint 3D-multi-view prediction for 3D semantic scene segmentation. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 452–468 (2018) 7
11. Fu, C., Dong, C., Mertz, C., Dolan, J.M.: Depth completion via inductive fusion of planar lidar and monocular camera. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10843–10848 (2020) 4
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361 (2012) 9
13. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2495–2504 (2020) 2, 4
14. Guo, X., Yang, K., Yang, W., Wang, X., Li, H.: Group-wise correlation stereo network. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3273–3282 (2019) 2, 4
15. Harrison, A., Newman, P.: Image and sparse laser fusion for dense scene reconstruction. In: Proceedings of Field and Service Robotics (FSR). pp. 219–228. Springer (2010) 10

16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016) 10
17. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Penet: Towards precise and efficient image guided depth completion. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 13656–13662 (2021) 12
18. Huang, Y.K., Wu, T.H., Liu, Y.C., Hsu, W.H.: Indoor depth completion with boundary consistency and self-attention. In: Proceedings of IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 0–0 (2019) 1, 3, 9, 10, 11
19. Huynh, L., Nguyen, P., Matas, J., Rahtu, E., Heikkilä, J.: Boosting monocular depth estimation with lightweight 3D point fusion. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 12767–12776 (2021) 4, 11, 12
20. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: End-to-end deep plane sweep stereo. arXiv preprint arXiv:1905.00538 (2019) 6, 9
21. Imran, S., Liu, X., Morris, D.: Depth completion with twin surface extrapolation at occlusion boundaries. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2583–2592 (2021) 11
22. Imran, S., Long, Y., Liu, X., Morris, D.: Depth coefficients for depth completion. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 12438–12447 (2019) 2, 4, 11, 12
23. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 66–75 (2017) 6, 9
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10
25. Lee, B.U., Lee, K., Kweon, I.S.: Depth completion using plane-residual representation. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 13916–13925 (2021) 1, 2, 4, 11, 12
26. Li, A., Yuan, Z., Ling, Y., Chi, W., Zhang, C., et al.: A multi-scale guided cascade hourglass network for depth completion. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 32–40 (2020) 1, 10, 11
27. Liao, Y., Huang, L., Wang, Y., Kodagoda, S., Yu, Y., Liu, Y.: Parse geometry from a line: Monocular depth estimation with partial laser observation. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 5059–5066 (2017) 4
28. Liu, J., Gong, X.: Guided depth enhancement via anisotropic diffusion. In: Proceedings of Pacific-Rim conference on multimedia (PCM). pp. 408–417. Springer (2013) 10
29. Liu, T.Y., Agrawal, P., Chen, A., Hong, B.W., Wong, A.: Monitored distillation for positive congruent depth completion. arXiv preprint arXiv:2203.16034 (2022) 12
30. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 5695–5703 (2016) 4
31. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 4796–4803 (2018) 1, 4, 9

32. Murez, Z., As, T.v., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3D scene reconstruction from posed images. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 414–431. Springer (2020) 7
33. Park, J., Joo, K., Hu, Z., Liu, C.K., So Kweon, I.: Non-local spatial propagation network for depth completion. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 120–136. Springer (2020) 1, 4, 11, 12
34. 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 IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3313–3322 (2019) 11
35. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3D residual networks. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 5533–5541 (2017) 5, 13
36. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241. Springer (2015) 13
37. Shen, Z., Dai, Y., Rao, Z.: Cfnet: Cascade and fused cost volume for robust stereo matching. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 13906–13915 (2021) 2, 4, 6
38. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 1874–1883 (2016) 3, 6
39. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 746–760. Springer (2012) 9
40. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: Neuralrecon: Real-time coherent 3D reconstruction from monocular video. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 15598–15607 (2021) 7
41. Tang, J., Tian, F.P., Feng, W., Li, J., Tan, P.: Learning guided convolutional network for depth completion. IEEE Transactions on Image Processing (TIP) **30**, 1116–1129 (2020) 4, 11
42. Wang, S., Suo, S., Ma, W.C., Pokrovsky, A., Urtasun, R.: Deep parametric continuous convolutional neural networks. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2589–2597 (2018) 4
43. Wong, A., Fei, X., Tsuei, S., Soatto, S.: Unsupervised depth completion from visual inertial odometry. IEEE Robotics and Automation Letters (RA-L) **5**(2), 1899–1906 (2020) 9
44. Wong, A., Soatto, S.: Unsupervised depth completion with calibrated backprojection layers. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 12747–12756 (2021) 12
45. Xu, B., Xu, Y., Yang, X., Jia, W., Guo, Y.: Bilateral grid learning for stereo matching networks. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 12497–12506 (2021) 2, 6, 13, 14
46. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 1592–1599 (2015) 4

47. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 185–194 (2019) 4
48. Zhang, Y., Funkhouser, T.: Deep depth completion of a single RGB-D image. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR). pp. 175–185 (2018) 3, 9, 10, 11
49. Zhong, Y., Wu, C.Y., You, S., Neumann, U.: Deep RGB-D canonical correlation analysis for sparse depth completion. Advances in Neural Information Processing Systems (NeurIPS) **32** (2019) 4