Deep Cost Ray Fusion for Sparse Depth Video Completion

Jungeon Kim[1], Soongjin Kim[1], Jaesik Park[2], and Seungyong Lee[1]

[1] POSTECH, South Korea
[2] Seoul National University, South Korea
{jungeonkim,kimsj0302,leesy}@postech.ac.kr
jaesik.park@snu.ac.kr

Abstract. In this paper, we present a learning-based framework for sparse depth video completion. Given a sparse depth map and a color image at a certain viewpoint, our approach makes a cost volume that is constructed on depth hypothesis planes. To effectively fuse sequential cost volumes of the multiple viewpoints for improved depth completion, we introduce a learning-based cost volume fusion framework, namely *RayFusion*, that effectively leverages the attention mechanism for each pair of overlapped rays in adjacent cost volumes. As a result of leveraging feature statistics accumulated over time, our proposed framework consistently outperforms or rivals state-of-the-art approaches on diverse indoor and outdoor datasets, including the KITTI Depth Completion benchmark, VOID Depth Completion benchmark, and ScanNetV2 dataset, using much fewer network parameters.

Keywords: Depth completion · Cost volume fusion · RGB-D video

1 Introduction

With the benefit of capturing the actual distance, range-sensing devices such as Microsoft Kinect, LiDARs, and Intel RealSense have become increasingly popular. Notably, recent releases of high-end mobile devices like the iPhone are equipped with LiDARs, reflecting this trend. However, these depth sensors often suffer from missing or insufficient depth measurements. To address the challenge, a variety of learning-based depth completion methods have been proposed [11,26, 29,47,60,70]. Most of the state-of-the-art depth completion methods use merely a single-view RGB-D image to fill in missing depth values, predominantly focusing on extracting informative multimodal features from an input RGB-D image.

Given the accessibility of RGB-D video data, self-supervised depth completion studies utilized multiple RGB frames for auxiliary photometric loss [59, 60] for the network training. For more direct utilization of temporal information for enhanced depth completion at test time, a few recent studies [24, 37] tried to fuse feature maps of individual frames using a convolutional long short-term memory (ConvLSTM) [43] or spatio-temporal convolution [50]. These feature fusion methods basically need warping the previous feature map to align it with

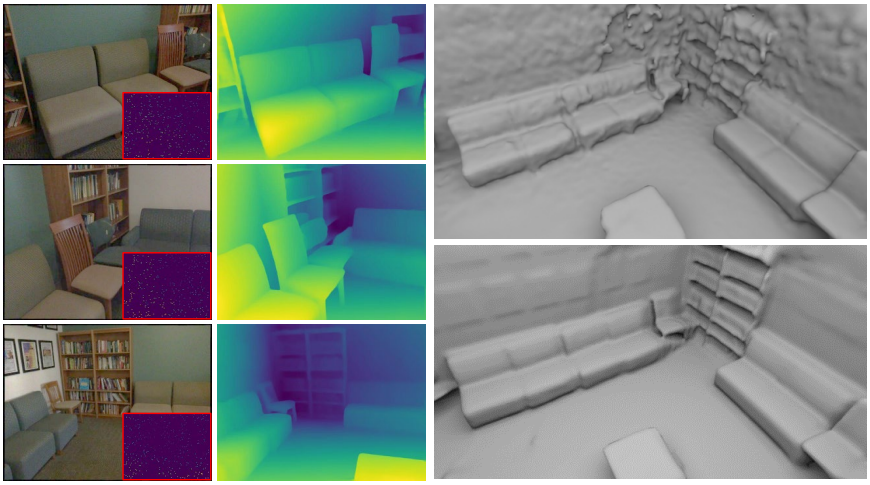

Fig. 1: Depth video completion result of our RayFusion framework. The framework takes RGB and sparse depth (0.1% density) video pairs as input (left) and infers completed depth maps (middle). Additionally, we show 3D reconstructions using raw sparse depths (top right) and the completed depths (bottom right). See the supplementary video for various video depth completion results.

the current feature map. However, the alignment is error-prone because such a warping depends on the predicted depths of the previous frame. Although achieving better temporal smoothness, these methods [24, 37] tend to exhibit inferior accuracy compared to the single-view completion methods.

In this paper, instead of the feature map alignment approach, we utilize a cost volume [18,22,64,66], which has been widely adopted for multi-view stereo, for temporal fusion (Figure 1). A cost volume is computed with hypothesis depth planes and contains information about probability distributions used for subsequent depth regression. It spatially spans the viewing frustum in the Euclidean space (Figure 2 (a)), enabling fusion for cost volumes to be directly performed in the 3D overlapped region of viewing frustums through volume resampling. Therefore, unlike feature image fusion methods, the approach remains unaffected by erroneous depth predictions.

To effectively fuse cost volumes obtained from an RGB-D video, a potential approach is to apply a recurrent neural network (RNN) [4, 13] to overlapped voxels in the cost volume. However, this fusion approach can overlook global attributes contained in the cost volumes (Table 4 (v)). The attention mechanism [52] could be a good alternative, but applying a global attention scheme to the entire cost volume requires a huge memory footprint and computation resource (Figure 2 (d)).

This paper introduces a framework that utilizes a *ray-based cost volume fusion scheme*. Consider a ray that penetrates two cost volumes of different viewpoints (Figure 2 (a)). Our fusion scheme is basically motivated by observation

that the features along the ray within cost volumes contain information about probability distributions on hypothesis depth planes. We make the ray-wise features (Figure 2 (b)) from two views become a minimal unit for the volume fusion, avoiding a heavy memory footprint, unlike whole volume attention. Our fusion procedure for ray-wise features comprises two sequential stages: self-attention for refining a current-view depth hypothesis and cross-attention for fusing current-view and previous-view cost volumes. We employ the cross entropy (CE) loss to effectively train the fusion module using pseudo ground truth probability distributions [34].

We validate the proposed framework through comprehensive experiments on diverse indoor and outdoor datasets, including the KITTI Depth Completion benchmark [51], VOID Depth Completion benchmark [59], and ScanNetV2 dataset [5]. As a result, we demonstrate outperforming performance over state-of-the-art (SOTA) depth completion methods in both depth and 3D reconstruction metrics and generalization ability despite utilizing significantly fewer network parameters (1.15M parameters - 94.5% smaller than LRRU [55]) thanks to our effective ray-wise attention design. More interestingly, we demonstrate that the proposed framework, despite not utilizing multiview information (i.e., only using the self-attention stage), still achieves SOTA performance on VOID and ScanNetV2 datasets. This achievement is attributed to our self-attention stage, which refines the cost volume by considering intrinsic properties such as the entropy of probability distributions within the cost volumes.

To summarize, our contributions are as follows:

- We propose an end-to-end deep learning-based framework, *RayFusion*, that effectively utilizes temporal information from an input RGB-D video to enhance sparse depth completion.
- We propose a novel *ray-based cost volume fusion* scheme that leverages the attention mechanism of the Transformer [52] to consider attributes of probability distributions within cost volumes.
- Our *RayFusion* consistently outperforms or competes with previous SOTA depth completion methods on various indoor and outdoor datasets with significantly fewer network parameters.

2 Related Work

Our framework is closely related to multi-view stereo and depth completion research. We review representative methods of those fields.

Multi-view stereo. With the advent of deep learning, many multi-view stereo (MVS) methods using deep neural networks have been proposed to replace traditional MVS approaches [42]. Inspired by the plane sweep stereo, the mainstream deep learning-based methodology in MVS is basically composed of three main stages [66]; image feature extraction, cost volume creation, and cost regularization. Numerous studies have tried to improve those stages or craft novel loss functions to predict accurate depths [3, 12, 18, 44, 53, 64, 64, 67]. Recently, a few

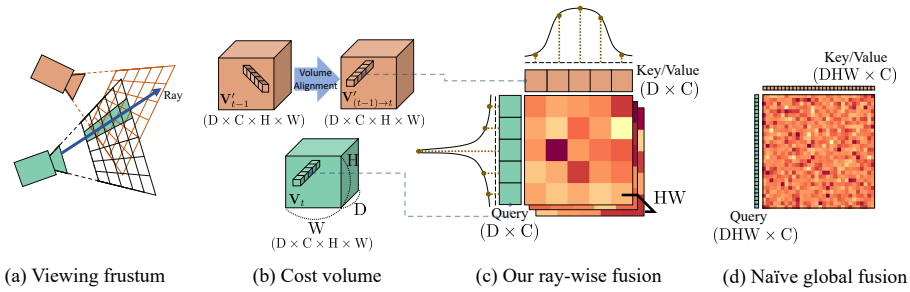

Fig. 2: Illustration of the proposed cost volume fusion scheme. A cost volume is constructed on D depth hypothesis planes and each voxel contains a C-dimensional feature vector. When fusing two aligned cost volumes $(\mathbf{V}'_{(t-1)\to t}, \mathbf{V}_t)$ (b), the proposed scheme (c) applies the attention mechanism into feature sequences corresponding to rays. It is computationally- and memory-efficient than the naive approach (d) of calculating the attention for all features in cost volumes.

studies [6,54,61] have employed the attention approach of Transformer for global feature matching on epipolar lines in the image space.

As another line of research, methods that use a monocular RGB video as the input [7, 14, 28] have been proposed. Unlike the common MVS studies that use multi-view RGB images with proper baselines as the input, they fully leverage the RGB sequence by the temporal fusion of various representations, including feature images [7], a latent vector without spatial information [14], and a probability volume [28] using different techniques (ConvLSTM [7]; the nonparametric Gaussian process [14]; Bayesian filtering implemented as a naive 3D CNN [28]).

Depth completion. Early studies in this field achieve depth completion by considering a depth image as an additional RGB image channel and concatenating it along the channel dimension. The resulting image is then fed into 2D convolutional networks [26,33]. Follow-up studies deal with depth images using separate networks for late fusion with RGB features [11,29,47,60,70]. A few studies utilize networks that estimate a surface normal [16,38,68], uncertainty [8,15,46,49], or edge [39,48] as a local property related to depth information.

Recent methods consider depth images in 3D space to properly use 3D positional information [1,2,17,21]. They back-projected a depth map to obtain a point cloud and extract features on the point cloud. The point cloud features are projected onto the image space and concatenated with RGB features. A few methods adopt Vision Transformer as the backbone to fully leverage the global context on the image domain at the expense of huge network parameters [40,69]. Unlike aforementioned approaches that focus on extracting better multimodal (2D and 3D) image features, CostDCNet [22] forms a multi-modal feature volume in 3D space, called a cost volume, from a single RGB-D image and infers a completed depth from the cost volume.

The depth completion methods with state-of-the-art (SOTA) performances mainly focus on using only a single-view RGB-D image. While a few stud-

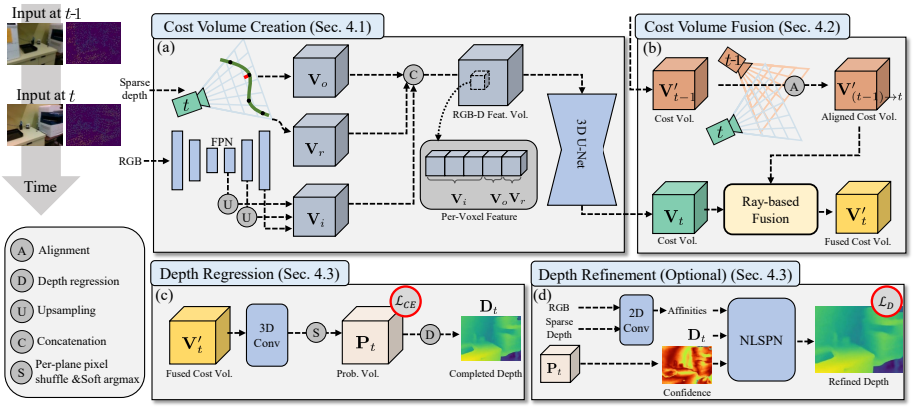

Fig. 3: Overall pipeline of our framework. For each frame, our framework infers a cost volume from a single-view RGB-D image (Section 4.1) and then fuses the cost volume with the cost volume updated up to the previous frame (Section 4.2). The fused cost volume is used for completed depth regression (Section 4.3) and becomes the cost volume for fusion at the next frame. Finally, the completed depth is refined by non-local spatial propagation networks (NLSPN).

ies [24, 37] tried to utilize multiple RGB-D frames by temporally fusing image features via ConvLSTM or spatio-temporal convolution, their results may exhibit inferior accuracy than single-view approaches since their fusion scheme does not adequately address misalignments of adjacent images.

In this paper, we propose an effective framework that fuses two temporally adjacent cost volumes of different viewpoints to infer a more accurate completed depth map. Unlike existing RGB-D video-based methods, our framework performs cost volume fusion in 3D space, which does not rely on the previous erroneous depth prediction. For efficient and effective fusion, we propose a ray-based fusion scheme that leverages the attention mechanism of Transformer [52].

3 Overview

For depth video completion with a calibrated camera, we formulate the supervised learning problem as follows:

$$\boldsymbol{\theta}^{\star} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{D}_t, \mathbf{D}_{gt}), \tag{1}$$
$$\mathbf{D}_t = f_{\boldsymbol{\theta}}((\mathbf{I}_t, \mathbf{S}_t), ..., (\mathbf{I}_1, \mathbf{S}_1), \mathbf{T}_t, ..., \mathbf{T}_1, \mathbf{K}),$$

where $f_{\boldsymbol{\theta}}$ is a predictor with learnable parameters $\boldsymbol{\theta}$ that uses color $\mathbf{I}_t$ and sparse depth $\mathbf{S}_t$ images, camera poses $\mathbf{T}_t \in SE(3)$, and camera intrinsic parameters $\mathbf{K}$ to infer a completed depth image $\mathbf{D}_t$ at the current frame t, $\mathbf{D}_{gt}$ is the ground truth completed depth, and $\mathcal{L}(\cdot, \cdot)$ is a loss function. To implement the

predictor $f_{\boldsymbol{\theta}}$, we propose an incremental cost volume update approach with ray-wise attention, called *RayFusion*, and reformulate the problem as follows:

$$\begin{aligned} \boldsymbol{\theta}^{\star} &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{D}'_t, \mathbf{D}_{gt}, \mathbf{P}_t, \mathbf{P}_{gt}), \\ \mathbf{D}'_t &= H_{\boldsymbol{\theta}}(\mathbf{D}_t, \mathbf{P}_t, \mathbf{I}_t, \mathbf{S}_t), \quad \mathbf{D}_t, \mathbf{P}_t = R_{\boldsymbol{\theta}}(\mathbf{V}'_t), \\ \mathbf{V}'_t &= F_{\boldsymbol{\theta}}(\mathbf{V}'_{t-1}, \mathbf{V}_t, \mathbf{T}_t, \mathbf{T}_{t-1}, \mathbf{K}), \quad \mathbf{V}_t = C_{\boldsymbol{\theta}}(\mathbf{I}_t, \mathbf{S}_t, \mathbf{K}), \end{aligned} \tag{2}$$

where $C_{\boldsymbol{\theta}}$, $F_{\boldsymbol{\theta}}$, $R_{\boldsymbol{\theta}}$, and $H_{\boldsymbol{\theta}}$ are neural networks for cost volume creation, cost volume fusion, depth regression, and depth refinement, respectively.

For each frame, $C_{\boldsymbol{\theta}}$ predicts a cost volume $\mathbf{V}_t$ using the current RGB $\mathbf{I}_t$ and sparse depth $\mathbf{S}_t$ images, and camera intrinsics $\mathbf{K}$ as the input. Then, the predicted cost volume $\mathbf{V}_t$ is fused with the cost volume $\mathbf{V}'_{t-1}$ updated up to the previous frame by $F_{\boldsymbol{\theta}}$. The fused cost volume $\mathbf{V}'_t$ at the current frame is used for regressing a completed depth image $\mathbf{D}_t$ via a probability volume $\mathbf{P}_t$ computed by $R_{\boldsymbol{\theta}}$. Lastly, the depth refinement module $H_{\boldsymbol{\theta}}$ improves the completed depth on the image domain using non-local spatial propagation (NLSPN) [35] to obtain the final completed depth $\mathbf{D}'_t$.

Figure 3 shows the overall pipeline of our framework. In the following sections, we elaborate on the main components of our framework, RGB-D cost volume creation (Section 4.1), ray-based cost volume fusion (Section 4.2), and completed depth regression and refinement (Section 4.3).

4 Deep Cost Ray Fusion

4.1 Cost Volume Creation

Given an input RGB image and sparse depth samples at each frame, our cost volume creation module $C_{\boldsymbol{\theta}}$ forms occupancy $\mathbf{V}_o$, residual volumes $\mathbf{V}_r$ from sparse depth samples, and RGB feature volume $\mathbf{V}_i$ from multi-scale image features (Figure 3 (a)). We concatenate these input feature volumes along channel dimensions to obtain an RGB-D feature volume. Then, we feed the RGB-D feature volume into a 3D convolutional U-Net to infer a cost volume at the current frame. We utilize a modified version of CostDCNet [22] for cost volume creation that does not use a separate geometric feature extractor and uses multi-scale image features.

The feature volume ($\mathbf{V}_o$,$\mathbf{V}_r$,$\mathbf{V}_i$) is constructed on uniformly spaced hypothesis depth planes [18,66]. The number of depth planes D and the minimum $d_{\min}$ and maximum $d_{\max}$ depth values of these planes are hyperparameters. When the image spatial resolution is H $\times$ W and the feature dimension is C, we have a feature volume $\mathbf{V} \in \mathbb{R}^{D \times C \times H \times W}$. Note that the spatial coverage of the feature volume in 3D Euclidean space corresponds to the viewing frustum at the current frame. More details are described in the supplementary document.

4.2 Ray-based Fusion

In this section, we explain our ray-based cost volume fusion scheme that effectively considers intrinsic attributes of probability distributions within cost volumes in a memory-efficient manner. Let us assume that we have a cost volume $\mathbf{V}_t$ from the current frame (Section 4.1) and a cost volume $\mathbf{V}'_{t-1}$ updated until the previous frame $t-1$, as shown in Figure 3 (b).

Aligning cost volumes. We first align two cost volumes ($\mathbf{V}'_{t-1}$, $\mathbf{V}_t$) of different viewpoints for the fusion. We utilize the relative camera pose between t and $t-1$ and employ inverse mapping to obtain an aligned cost volume $\mathbf{V}'_{(t-1)\to t}$. This inverse mapping makes the coordinates of $\mathbf{V}'_{t-1}$ to be aligned with the coordinates at the current viewpoint, and it allows easy ray-wise computation in the aligned coordinates. More details can be found in the supplementary document.

Fusion. We now introduce our approach to fuse two volumes $\mathbf{V}_t$ and $\mathbf{V}'_{(t-1)\to t}$ using attention mechanism [52]. The naïve approach is to linearize all voxel features of each cost volume and then compute the cross-attention. However, it requires $\mathrm{D}^2\mathrm{H}^2\mathrm{W}^2$ entries for attention weight calculation, which is impractical due to the huge memory footprint and computation complexity (Figure 2 (d)).

Instead, we propose the *ray-wise fusion scheme* that calculates attention for only extracted ray-wise features of the aligned cost volumes $\mathbf{V}_t$ and $\mathbf{V}'_{(t-1)\to t}$ (Figures 2 and 4). For an arbitrary pixel position (h, w), we can obtain a ray-wise feature $\mathbf{F}_t = \mathbf{V}(:, :, h, w) \in \mathbb{R}^{D\times C}$ from a cost volume, where each row of the matrix $\mathbf{F}_t$ indicates a C-dimensional feature vector for a certain depth plane hypothesis. We then regard $\mathbf{F}_t$ as D tokens, where each token is a C-dimensional feature, and apply the attention mechanism to fuse two ray-wise features $\mathbf{F}_t = \mathbf{V}_t(:, :, h, w)$ and $\mathbf{F}_{t-1} = \mathbf{V}'_{(t-1)\to t}(:, :, h, w)$.

A straightforward option for the fusion is to compute cross-attention between $\mathbf{F}_t$ and $\mathbf{F}_{t-1}$. However, the cross-attention does not consider the intrinsic properties of individual ray-wise features. Inspired by the stereo matching approach [57] that utilizes entropy of a probability distribution as an uncertainty prior, we expect the network to consider the intrinsic characteristics of each ray-wise feature. In addition, this independent fusion does not consider spatially adjacent features within cost volumes.

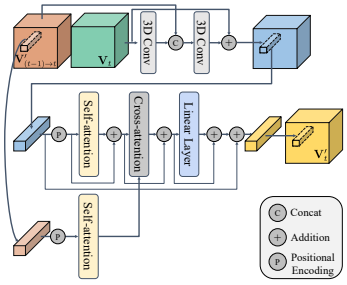

Fig. 4: Our ray fusion module.

To compensate for such deficiency, we apply two 3D convolutional layers to the volumes before applying our fusion scheme, as shown in Figure 4. We then compute the self-attention $\mathbf{SA}_t = Attn(\mathbf{F}_t, \mathbf{F}_t, \mathbf{F}_t)$[3] and com-

[3] denoted as $Attn(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \big(softmax(\frac{\mathbf{Q}\mathbf{W}_Q(\mathbf{K}\mathbf{W}_K)^\mathsf{T}}{\sqrt{d}})\mathbf{V}\mathbf{W}_V\big)\mathbf{W}_O$, where $\{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O\}$ are learnable linear projection parameters.

pute $\mathbf{SA}_{t-1}$ similarly. Finally, the fused feature is calculated using cross-attention $\mathbf{CA}_t = Attn(\mathbf{SA}_t, \mathbf{SA}_{t-1}, \mathbf{SA}_{t-1})$. To inject information about relative positions among D tokens in $\mathbf{F}$, we add the sinusoidal positional encodings [52] of depth plane indices before the fusion.

We repeat the process for all ray-wise feature pairs to make the fused cost volume ($\mathbf{V}'_t$) as depicted in Figure 4). Note that the proposed method needs only $\mathrm{D}^2\mathrm{HW}$ entries for constructing attention maps, and it is much more memory-efficient than naïve approach consuming $\mathrm{D}^2\mathrm{H}^2\mathrm{W}^2$ entries.

4.3 Depth Regression

To regress a completed depth map from the fused cost volume $\mathbf{V}'_t \in \mathbb{R}^{D\times C\times H\times W}$, we firstly convert the fused cost volume to an unnormalized probability volume $\mathbf{P}'_t$ ($\mathrm{D}\times\mathrm{H}\times\mathrm{W}$) via a single 3D convolutional layer and per-plane pixel shuffle [22]. Then, we apply the softmax operator $\sigma(\cdot)$ [23] to regress a completed depth $\mathbf{D}_t$ as follows [12, 18]:

$$\mathbf{D}_t(h,w) = \sum_{i=1}^{D} d_i \times \mathbf{p}^i_{h,w}, \quad \mathbf{p}_{h,w} = \mathbf{P}_t(:,h,w) = \sigma(\mathbf{P}'_t(:,h,w)), \tag{3}$$

where d_i is the pre-defined depth value of the i-th hypothesis depth plane, (h, w) is an image pixel position, D is the number of hypothesis depth planes, $\mathbf{P}_t$ is a probability volume, and $\mathbf{p}_{h,w}$ is a D-dimensional probability vector for depth planes at (h, w).

To further refine the regressed depth $\mathbf{D}_t$ on the image domain, we adapt non-local spatial propagation networks (NLSPN) [35] with minor modification. NLSPN takes as the input an affinity map, a confidence map, and a depth to be refined. In our case, the regressed depth is accurate enough, so we utilize shallow 2D convolutional networks for estimating an affinity map, and we directly compute a confidence map from $\mathbf{P}_t(d, h, w)$. For more details, we refer the readers to the supplementary document.

4.4 Loss Function

Our framework is fully differentiable, and it can be trained in an end-to-end manner. We use the L_1 depth loss and a cross-entropy loss for probability volume supervision (Figure 3). L_1 depth regression loss is defined as follows:

$$\mathcal{L}_{L1} = \frac{1}{|\mathbb{P}|} \sum_{(h,w)\in\mathbb{P}} \left|\mathbf{D}_t(h,w) - \mathbf{D}_{gt}(h,w)\right|, \tag{4}$$

where $\mathbb{P}$ is the set of sparse GT depth pixels, $\mathbf{D}_t$ and $\mathbf{D}_{gt}$ are the completed depth and a ground truth depth. The cross-entropy loss is defined as follows:

$$\mathcal{L}_{CE} = \frac{1}{|\mathbb{P}|} \sum_{(h,w)\in\mathbb{P}} -\mathbf{p}_{gt}^{\mathsf{T}} \log \mathbf{p}, \tag{5}$$

where $\mathbf{p}_{gt}$ is a ground truth probability vector over the hypothesis depth plane, and $\mathbf{p}$ is a predicted probability vector obtained from $\mathbf{P}_t(:, h, w)$ (Eq. (3)).

We found that using hard labels (one-hot vector) for $\mathbf{p}_{gt}$ is not effective for the performance, similar to observations in [23]. We instead make soft labels from ground truth depths using the idea of Nuanes et al. [34]. We find two nearest depth planes for a given ground truth depth and then compute normalized weights for respective planes. This computation results in a probability distribution vector where all elements are zero except for two elements containing the respective normalized values.

Finally, our total loss $\mathcal{L}_{total}$ is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{L1} + \mathcal{L}_{CE}. \tag{6}$$

5 Experiments

5.1 Implementation Details

We implement our framework using PyTorch [9,36,56]. We train our model with a batch size of four on three NVIDIA GeForce RTX 3090 GPUs. We employ the AdamW optimizer [31] with a weight decay of 0.0001 and an initial learning rate of 0.001. The learning rate is reduced by a factor of 0.5 at a predefined epoch schedule. The average inference time of our model on the ScanNetv2 test set is 77ms. The number of hypothesis depth planes is set to 16. A cost volume as a four-dimensional volume requires a high memory footprint. In practice, we utilize downscaled images with a factor of 4. The supplementary document provides additional visual results and detailed information, including network architecture, error metrics, GPU memory consumption, and inference times.

5.2 Datasets

We use the following indoor/outdoor datasets to demonstrate our approach.

ScanNetV2 [5] dataset is a large-scale RGB-D dataset for indoor scenes, comprising 1,201 training, 312 validation, and 100 testing scenes (≈211,000 frames) captured with a handheld RGB-D sensor. This dataset includes accurate camera calibration parameters and ground truth depth images. In this case, we randomly obtain 300 depth samples from a ground truth depth image to create input sparse depth images. We set $d_{\min} = 10^{-3}m$ and $d_{\max} = 10m$. We randomly cropped images to 512×384 pixels during training. The learning rate schedule is $\{10, 15, 20, 25\}$ over 30 epochs. We test depths less than 10m for performance evaluation.

VOID [59] dataset contains synchronized 640×480 RGB images and sparse depth maps of indoor (e.g., laboratories and classrooms) and outdoor (gardens) scenes. The depth maps contain about 150, 500, or 1500 sparse depth samples (corresponding to 0.05%, 0.15%, and 0.5%) for each scene. Depth samples are obtained from a set of image features tracked by XIVO [10], a visual-inertial odometry system. The dense ground-truth depth maps are acquired by active

Table 1: Quantitative comparison on the ScanNetV2 [5] test set. 'SPN' and 'S' denote our depth refinement module and single-view setting. 'R' means that GT depths are obtained by rendering GT meshes. The unit for all metrics, except for **iMAE** (1/m) and **iRMSE** (1/m), is meter.

Method	#Param.	Depth Error				3D Reconstruction Error					
		MAE↓	RMSE↓	iMAE↓	iRMSE↓	Acc↓	Compl↓	Chamfer↓	Prec↑	Recall↑	F-score↑
SimpleRecon [41]	49.1M	0.0887	0.1448	0.0315	0.0499	0.0681	0.0557	0.0619	0.6694	0.6494	0.6572
ComplFormer [69]	83.5M	0.0276	0.0868	0.0091	0.0295	0.0442	0.0223	0.0332	0.8596	0.9152	0.8838
NLSPN [35]	25.8M	0.0266	0.0859	0.0086	0.0290	0.0412	0.0223	0.0318	0.8642	0.9147	0.8862
CostDCNet [22]	1.8M	0.0244	0.0759	0.0086	0.0255	0.0339	0.0204	0.0272	0.8778	0.9269	0.8998
Ours w/o SPN+S	1.11M	0.0208	0.0681	0.0068	0.0230	-	-	-	-	-	-
Ours w/o SPN	1.11M	0.0159	0.0553	0.0053	0.0192	0.0294	0.0181	0.0237	0.8908	0.9467	0.9163
Ours	1.15M	0.0160	0.0554	0.0053	0.0193	0.0295	0.0181	0.0238	0.8908	0.9465	0.9161
DeepSmooth+R	20.4M	0.043	0.142	-	-	-	-	-	-	-	-
Ours+S+R	1.15M	0.036	0.114	0.0199	0.0624	-	-	-	-	-	-

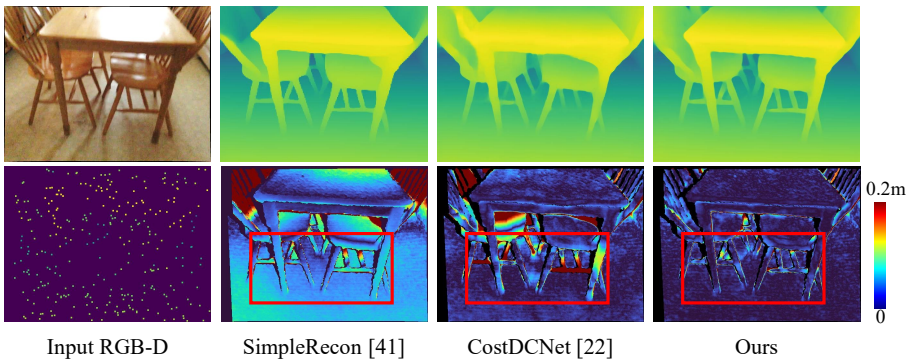

Fig. 5: Visual comparison of completed depths (top) on the ScanNetV2 test set. Error maps of completed depths are also presented (bottom).

stereo. The VOID dataset contains 56 sequences with challenging camera motions. Among the 56 sequences, 48 sequences (≈45,000 frames) are designated for training, and eight sequences (800 frames) are assigned for testing. We set $d_{\min} = 10^{-3}m$ and $d_{\max} = 6m$. During training, randomly cropped images of 512×384 pixels are used, and the learning rate schedule is $\{30, 40, 50, 60, 70\}$ over 80 epochs. We follow the evaluation protocol of [59] and evaluate depths within [0.2, 5.0]m.

KITTI [51] depth completion (DC) benchmark dataset contains about 86,000 1242×375 RGB-D pairs that capture diverse road scenes. The sparse depth samples are obtained using a Velodyne LiDAR sensor, and it accounts for approximately 5% of the image space. As the 1,000 test set frames of KITTI are not captured sequentially, our temporal fusion module cannot be applied. Thus, we evaluate our method on scenes in the validation set with sequential frames. We set $d_{\min} = 10^{-1}m$ and $d_{\max} = 90m$. During training, we use randomly cropped images of 1216×240 pixels, and the learning rate decay schedule is $\{30, 40, 50, 60, 70\}$ over the total 100 epochs.

Table 2: Quantitative comparison on VOID [59] test set. 'SPN' and 'S' denote our depth refinement module and single-view setting.

Method	#Param.	Density		MAE↓ (mm)	RMSE↓ (mm)	iMAE↓ (1/km)	iRMSE↓ (1/km)
		Training	Testing				
SS-S2D [32]	27.8M			178.85	243.84	80.12	107.69
DDP [65]	18.8M			151.86	222.36	74.59	112.36
VOICED [59]	9.7M			85.05	169.79	48.92	104.02
ScaffNet [58]	7.8M			59.53	119.14	35.72	68.36
MSG-CHN [25]	0.36M			43.57	109.94	23.44	52.09
KBNet [60]	6.9M	0.50%	0.50%	39.80	95.86	21.16	49.72
PENet [15]	132.0M			34.61	82.01	18.89	40.36
Mondi [30]	5.3M			29.67	79.78	14.84	37.88
NLSPN [35]	25.8M			26.74	79.12	12.70	33.88
ComplFormer [69]	83.5M			49.61	141.40	21.08	51.53
LRRU [55]	21.0M			47.20	118.00	22.00	48.30
CostDCNet [22]	1.8M			25.84	76.28	12.19	32.13
Ours w/o SPN+S	1.11M			25.53	68.83	12.37	31.52
Ours w/o SPN	1.11M	0.50%	0.50%	24.57	65.46	12.03	30.26
Ours	1.15M			24.51	65.46	11.98	30.20
LRRU [55]	21.0M	0.50%	0.15%	115.30	262.60	44.70	86.30
			0.05%	207.80	409.90	78.20	127.30
ComplFormer [69]	83.5M	0.50%	0.15%	228.49	449.51	62.71	119.92
			0.05%	395.42	639.05	107.52	174.05
NLSPN [35]	25.8M	0.50%	0.15%	65.91	160.76	27.79	63.26
			0.05%	118.19	245.41	52.57	99.36
Mondi [30]	5.3M	0.50%	0.15%	61.37	146.57	27.96	64.36
			0.05%	104.97	225.60	48.44	96.79
Ours w/o SPN+S	1.11M	0.50%	0.15%	52.80	121.65	24.80	56.83
			0.05%	84.45	174.46	41.75	83.64
Ours w/o SPN	1.11M	0.50%	0.15%	48.75	110.68	23.17	52.43
			0.05%	78.65	162.32	39.22	77.65
Ours	1.15M	0.50%	0.15%	48.67	110.56	23.10	52.35
			0.05%	78.55	162.17	39.15	77.55

5.3 Comparison

We compare our method with state-of-the-art (SOTA) single-view and video depth completion models qualitatively and quantitatively. For extensive comparison, we also present a comparison with a recent multiview stereo method [41] using RGB videos.

In the ScanNetV2 experiment, we choose recent single-view sparse depth completion approaches (NLSPN [35], CostDCNet [22], CompletionFormer [69]), a video depth completion method (DeepSmooth [24]), and a multiview stereo method (SimpleRecon [41]). We used the pretrained model of SimpleRecon provided by the authors, and we trained other models from scratch using the official source codes. Since DeepSmooth is a framework designed for semi-dense depth completion and its official source code is not available, we trained our framework under the same conditions as DeepSmooth for comparison. As a result, our method shows the best performance in both depth and 3D error metrics (Table 1 and Figure 5). While SimpleRecon often produces visually pleasing depth maps, its estimated depth values result in inferior quantitative performance because it does not utilize sparse depth samples, which serve as a strong prior.

We observe similar trends in the VOID test set, even if the depth samples provided by the VOID dataset are not from range-based sensors. Our model outperforms the state-of-the-art approaches [15, 22, 25, 30, 32, 35, 58–60, 65, 69] in most error metrics in this VOID dataset. Compared to CostDCNet, the second-

Table 3: Quantitative comparison on the KITTI [51] validation set. 'SPN' and 'S' denote our depth refinement module and single-view setting.

Method	#Param.	MAE↓ (mm)	RMSE↓ (mm)	iMAE↓ (1/km)	iRMSE↓ (1/km)
SS-S2D [32]	27.8M	269.20	878.50	1.34	3.25
Depth-normal [63]	29.0M	236.67	811.07	1.11	2.45
MSG-CHN [25]	0.36M	227.94	821.94	0.98	2.47
Mondi [30]	5.3M	218.22	815.16	0.91	2.18
Uber-FuseNet [2]	1.9M	217.00	785.00	1.08	2.36
DeepLidar [38]	53.4M	215.38	687.00	1.10	2.51
DC-3co [20]	27.0M	215.04	1011.30	0.94	2.50
3DepthNet [62]	-	208.96	693.23	0.98	2.37
PENet [15]	131.5M	208.81	753.75	0.91	2.16
NLSPN [35]	25.8M	198.64	771.80	0.83	2.03
CompletionFormer [69]	83.5M	198.63	748.07	0.85	2.00
TWISE [19]	1.5M	193.40	879.40	0.81	2.19
DySPN [27]	26.3M	192.50	745.80	-	-
LRRU [55]	21.0M	188.80	729.50	0.80	1.90
Ours w/o SPN+S	1.11M	193.05	793.03	0.76	1.74
Ours w/o SPN	1.11M	182.65	726.95	0.74	1.68
Ours	1.15M	176.23	720.63	0.72	1.68

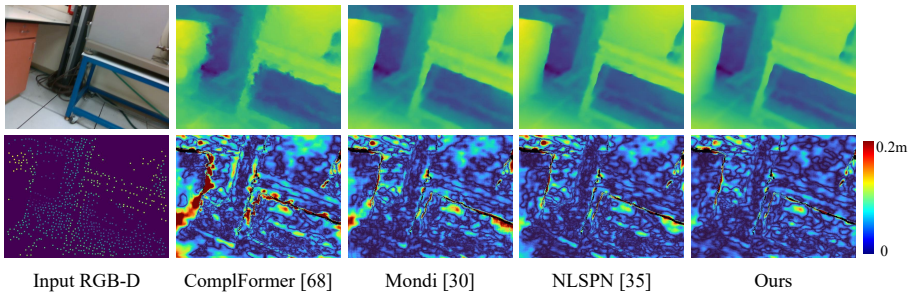

Fig. 6: Visual comparison of our method with state-of-the-art depth completion methods [30, 35, 69] on the VOID test set (0.5% sparsity) [59]. Completed depths (top) and error maps (bottom) are presented.

best model, our approach shows better performance with the smaller network parameters (Table 2 and Figure 6).

We achieved the best score among the baseline approaches [2, 15, 19, 20, 25, 27, 30, 32, 35, 38, 55, 62, 63, 69] in the validation set of the KITTI depth completion benchmark in most of the error metrics. Note that modern approaches such as DySPN [27], NLSPN [35], CompletionFormer [69], and LRRU [55] have much larger number of network parameters than ours to achieve competitive performance (Table 3 and Figure 7). We believe that our ray-based cost fusion scheme using self-/cross-attention is effective on various datasets.

5.4 Analysis

Effect of ray-based cost volume fusion. Our ray-based fusion improves the performance in most of the metrics with a small number of additional parameters

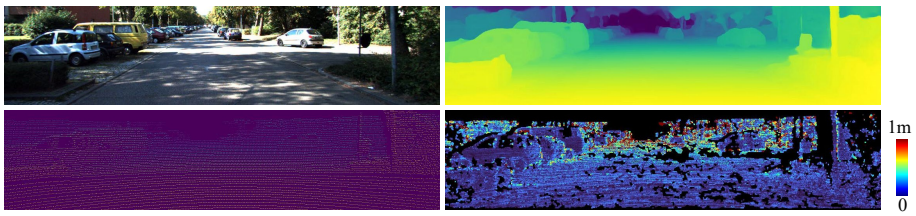

Fig. 7: Our completion result on the KITTI depth completion benchmark. Please refer to the supplementary document for a comparison with other approaches.

(0.08M) (Table 4 (iii, vii)). To compare the local cost volume fusion and our ray-based fusion, we adopt a convolutional variant of gated recurrent unit (GRU) used in [45] to fuse two aligned cost volumes. As a result, while the local fusion achieves slight performance improvement in the RMSE metric, its MAE metric rather deteriorates (Table 4 (v)).

Effect of cross entropy loss. We observed that using $\mathcal{L}_{CE}$ loss for direct distribution supervision contributes to stable learning. It also results in a balanced performance between mean absolute difference (MAE) and root mean square error (RMSE). This is a different observation to existing depth completion works [27, 35, 69] that combine L_1 and L_2 depth loss terms (Table 4 (i) and (ii)). A combination of three losses (L_1 + L_2 + CE loss term) does not lead to performance gain compared to using only the CE loss term (Table 4 (ii), (iv)).

Single RGB-D image as input. To evaluate the performance of our framework given a single RGB-D image as the input, we disable the cross-attention part of our fusion module and use only the self-attention part. Inevitably, the performance degrades in this single-view setting as it cannot leverage information from previous RGB-D frames. However, it is noteworthy that even in the single-view scenario, our framework is competitive to state-of-the-art depth completion methods [22, 27, 30, 35, 55, 69] on three different types of datasets (VOID, KITTI, and ScanNet) (Tables 3, 2, 1).

Effect of depth refinement module. Our depth refinement module, which has small network parameters (0.04M), improves the MAE metric on the KITTI dataset. However, we observed that the impact of the NLSPN on performance in the VOID dataset is relatively marginal, and the addition of NLSPN even leads to a slight decrease in performance on the ScanNetV2 dataset (Tables 3, 2, 1).

Robustness to various depth sparsity. To assess the sparsity-agnostic capability of our model and other approaches, we train the models on the VOID dataset with 0.5% depth sparsity and evaluate them on the VOID test set having 0.15% and 0.05% sparsity. As shown in Table 2, our approach is less affected by the changed depth sparsity compared with other approaches [30, 35, 69], despite using smaller network parameters.

Cross-dataset generalization. To evaluate the cross-dataset generalization ability, we train our model and baseline approaches (NLSPN [35] and CompletionFormer [69]) on the ScanNetv2 training set and evaluate them on the VOID

Table 4: Ablation study on the KITTI validation set. 'A' denotes our framework without the proposed ray-based fusion 'B1' and depth refinement 'C'. 'B2' denotes the convolutional GRU-based cost volume fusion [45]. 'L_1', 'L_2', and 'L_{CE}' denote L_1 and L_2 depth losses, and cross entropy loss, respectively. We use 20% of the KITTI training set for this experiment.

Network / Loss	#Param.	MAE↓	RMSE↓	iMAE↓	iRMSE↓
i A / L_1	1.03M	203.73	855.07	0.79	1.85
ii A / L_{CE}	1.03M	223.59	812.49	0.91	1.98
iii A / L_{CE}+L_1	1.03M	203.05	834.09	0.79	1.83
iv A / L_{CE}+L_1+L_2	1.03M	220.22	817.94	0.89	1.98
v A+B2 / L_{CE}	1.09M	228.85	799.03	1.03	2.04
vi A+B1 / L_{CE}	1.11M	215.18	774.10	0.88	1.87
vii A+B1 / L_{CE}+L_1	1.11M	198.51	777.37	0.82	1.82
viii A+B1+C / L_{CE}, SPN L_1	1.15M	188.88	768.11	0.77	1.78

Table 5: Quantitative comparison for cross-dataset generalization ability. All models are trained on the ScanNetv2 training set and tested on the VOID test set of 0.5% depth density.

Method	Params	MAE↓	RMSE↓	iMAE↓	iRMSE↓
NLSPN [35]	25.8M	158.60	571.80	22.00	57.50
CompletionFormer [69]	83.5M	65.90	190.01	20.66	49.83
Ours	1.15M	29.08	78.34	12.79	31.93

test set with 0.5% density. While CompletionFormer performs slightly worse than NLSPN on the ScanNetV2 test set, it demonstrates better generalization ability on the VOID test set (see Tables 1 and 5). Our method exhibits better generalizability than other methods. We speculate that our approach, which directly works with ray-wise cost slice, is a more generic approach across different dataset domains.

6 Conclusion

In this paper, we proposed a learning-based depth completion framework that effectively utilizes temporal information from an RGB-D video. We introduced the ray-based cost volume fusion scheme that leverages the attention mechanism. The fusion module effectively fuses cost volume predictions over time to infer a more accurate cost volume which is used for completed depth regression subsequently. We demonstrate that our framework, *RayFusion*, consistently beats or rivals state-of-the-art (SOTA) depth completion methods on diverse indoor and outdoor datasets, despite utilizing significantly fewer network parameters.

Limitation and future work. Our framework, relying on cost volumes with 3D convolutions and computing attention maps between them, suffers from a high memory footprint. Additionally, persistent poor depth predictions over time pose challenges that our method cannot fully resolve. Future work could involve designing a more computationally efficient network architecture [41] that eliminates fully 3D convolutions.

Acknowledgements

This work was supported by the NRF grant (RS-2023-00280400) and IITP grants (ICT Research Center, RS-2024-00437866; RS-2023-00227993; AI Innovation Hub, RS-2021-II212068; AI Graduate School Programs at POSTECH and SNU, RS-2019-II191906 and RS-2021-II211343) funded by Korea government (MSIT).

References

1. Chen, H., Yang, H., Zhang, Y., et al.: Depth completion using geometry-aware embedding. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 8680–8686. IEEE (2022)
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, S., Xu, Z., Zhu, S., Li, Z., Li, L.E., Ramamoorthi, R., Su, H.: Deep stereo using adaptive thin volume representation with uncertainty awareness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2524–2534 (2020)
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
5. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
6. Ding, Y., Yuan, W., Zhu, Q., Zhang, H., Liu, X., Wang, Y., Liu, X.: Transmvsnet: Global context-aware multi-view stereo network with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8585–8594 (2022)
7. Duzceker, A., Galliani, S., Vogel, C., Speciale, P., Dusmanu, M., Pollefeys, M.: Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15324–15333 (June 2021)
8. Eldesokey, A., Felsberg, M., Holmquist, K., Persson, M.: Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12014–12023 (2020)
9. Falcon, W., The PyTorch Lightning team: PyTorch Lightning (3 2019). `https://doi.org/10.5281/zenodo.3828935`, `https://github.com/Lightning-AI/lightning`
10. Fei, X., Wong, A., Soatto, S.: Geo-supervised visual depth prediction. IEEE Robotics and Automation Letters **4**(2), 1661–1668 (2019)
11. Fu, C., Dong, C., Mertz, C., Dolan, J.M.: Depth completion via inductive fusion of planar lidar and monocular camera. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10843–10848. IEEE (2020)
12. 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 the IEEE/CVF conference on computer vision and pattern recognition. pp. 2495–2504 (2020)

13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
14. Hou, Y., Kannala, J., Solin, A.: Multi-view stereo by temporal nonparametric fusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2651–2660 (2019)
15. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Towards precise and efficient image guided depth completion (2021)
16. Huang, Y.K., Wu, T.H., Liu, Y.C., Hsu, W.H.: Indoor depth completion with boundary consistency and self-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
17. Huynh, L., Nguyen, P., Matas, J., Rahtu, E., Heikkilä, J.: Boosting monocular depth estimation with lightweight 3d point fusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12767–12776 (2021)
18. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: End-to-end deep plane sweep stereo. arXiv preprint arXiv:1905.00538 (2019)
19. 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)
20. 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)
21. Jeon, Y., Kim, H., Seo, S.W.: Abcd: Attentive bilateral convolutional network for robust depth completion. IEEE Robotics and Automation Letters **7**(1), 81–87 (2021)
22. Kam, J., Kim, J., Kim, S., Park, J., Lee, S.: Costdcnet: Cost volume based depth completion for a single rgb-d image. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II. pp. 257–274. Springer (2022)
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 the IEEE International Conference on Computer Vision. pp. 66–75 (2017)
24. Krishna, S., Vandrotti, B.S.: Deepsmooth: Efficient and smooth depth completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 3358–3367 (June 2023)
25. Li, A., Yuan, Z., Ling, Y., Chi, W., Zhang, C., et al.: A multi-scale guided cascade hourglass network for depth completion. In: The IEEE Winter Conference on Applications of Computer Vision. pp. 32–40 (2020)
26. 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: 2017 IEEE international conference on robotics and automation (ICRA). pp. 5059–5066. IEEE (2017)
27. Lin, Y., Cheng, T., Zhong, Q., Zhou, W., Yang, H.: Dynamic spatial propagation network for depth completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1638–1646 (2022)
28. Liu, C., Gu, J., Kim, K., Narasimhan, S.G., Kautz, J.: Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10986–10995 (2019)
29. Liu, L., Song, X., Sun, J., Lyu, X., Li, L., Liu, Y., Zhang, L.: Mff-net: Towards efficient monocular depth completion with multi-modal feature fusion. IEEE Robotics and Automation Letters (2023)

30. Liu, T.Y., Agrawal, P., Chen, A., Hong, B.W., Wong, A.: Monitored distillation for positive congruent depth completion. In: European Conference on Computer Vision. pp. 35–53. Springer (2022)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
32. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera (2019)
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. Nuanes, T., Elsey, M., Sankaranarayanan, A., Shen, J.: Soft cross entropy loss and bottleneck tri-cost volume for efficient stereo depth prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2846–2854 (June 2021)
35. Park, J., Joo, K., Hu, Z., Liu, C.K., Kweon, I.S.: Non-local spatial propagation network for depth completion. In: Proc. of European Conference on Computer Vision (ECCV) (2020)
36. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
37. Patil, V., Van Gansbeke, W., Dai, D., Van Gool, L.: Don't forget the past: Recurrent depth estimation from monocular video. IEEE Robotics and Automation Letters **5**(4), 6813–6820 (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. Ramesh, A.N., Giovanneschi, F., González-Huici, M.A.: Siunet: Sparsity invariant u-net for edge-aware depth completion. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5818–5827 (2023)
40. Rho, K., Ha, J., Kim, Y.: Guideformer: Transformers for image guided depth completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6250–6259 (June 2022)
41. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplerecon: 3d reconstruction without 3d convolutions. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII. pp. 1–19. Springer (2022)
42. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
43. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems **28** (2015)
44. Sormann, C., Santellani, E., Rossi, M., Kuhn, A., Fraundorfer, F.: Dels-mvs: Deep epipolar line search for multi-view stereo. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3087–3096 (2023)
45. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: NeuralRecon: Real-time coherent 3D reconstruction from monocular video. CVPR (2021)

46. Taguchi, K., Morita, S., Hayashi, Y., Imaeda, W., Fujiyoshi, H.: Uncertainty-aware interactive lidar sampling for deep depth completion. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3028–3036 (2023)
47. 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)
48. Tao, Z., Shuguo, P., Hui, Z., Yingchun, S.: Dilated u-block for lightweight indoor depth completion with sobel edge. IEEE Signal Processing Letters **28**, 1615–1619 (2021)
49. Teixeira, L., Oswald, M.R., Pollefeys, M., Chli, M.: Aerial single-view depth completion with image-guided uncertainty estimation. IEEE Robotics and Automation Letters **5**(2), 1055–1062 (2020)
50. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)
51. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: International Conference on 3D Vision (3DV) (2017)
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
53. Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: Patchmatchnet: Learned multi-view patchmatch stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14194–14203 (2021)
54. Wang, X., Zhu, Z., Huang, G., Qin, F., Ye, Y., He, Y., Chi, X., Wang, X.: Mvster: epipolar transformer for efficient multi-view stereo. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI. pp. 573–591. Springer (2022)
55. Wang, Y., Li, B., Zhang, G., Liu, Q., Tao, G., Dai, Y.: Lrru: Long-short range recurrent updating networks for depth completion. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2023)
56. Wightman, R.: Pytorch image models. `https://github.com/rwightman/pytorch-image-models` (2019). `https://doi.org/10.5281/zenodo.4414861`
57. Won, C., Ryu, J., Lim, J.: End-to-end learning for omnidirectional stereo matching with uncertainty prior. IEEE transactions on pattern analysis and machine intelligence **43**(11), 3850–3862 (2020)
58. Wong, A., Cicek, S., Soatto, S.: Learning topology from synthetic data for unsupervised depth completion. IEEE Robotics and Automation Letters **6**(2), 1495–1502 (2021)
59. Wong, A., Fei, X., Tsuei, S., Soatto, S.: Unsupervised depth completion from visual inertial odometry. IEEE Robotics and Automation Letters **5**(2), 1899–1906 (2020)
60. Wong, A., Soatto, S.: Unsupervised depth completion with calibrated backprojection layers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12747–12756 (2021)
61. Xi, J., Shi, Y., Wang, Y., Guo, Y., Xu, K.: Raymvsnet: Learning ray-based 1d implicit fields for accurate multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8595–8605 (2022)
62. Xiang, R., Zheng, F., Su, H., Zhang, Z.: 3ddepthnet: Point cloud guided depth completion network for sparse depth and single color image. arXiv preprint arXiv:2003.09175 (2020)

63. 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)
64. Yang, J., Mao, W., Alvarez, J.M., Liu, M.: Cost volume pyramid based depth inference for multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4877–4886 (2020)
65. Yang, Y., Wong, A., Soatto, S.: Dense depth posterior (ddp) from single image and sparse range. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3353–3362 (2019)
66. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
67. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5525–5534 (2019)
68. Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 175–185 (2018)
69. Zhang, Y., Guo, X., Poggi, M., Zhu, Z., Huang, G., Mattoccia, S.: Completionformer: Depth completion with convolutions and vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18527–18536 (2023)
70. 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 **32** (2019)