Deep Cost Ray Fusion for Sparse Depth Video Completion

Supplementary Material

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

A Network Architecture

Cost volume creation. For the RGB image feature extractor, we adopt the image feature extractor used in NeuralRecon [12]. The feature extractor is a lightweight variant of MnasNet [13] initialized with pretrained weights obtained from ImageNet [5] and utilizes the architecture of Feature Pyramid Network [6] to extract multi-scale features. For 3D convolutional U-Net, we employ the network architecture used in CostDCNet [4].

Ray-based cost volume fusion module. Figure 8 illustrates the architecture of our ray-based cost volume fusion module, where additional details (not shown in Figure 4), including normalization, activation functions, and linear embedding of voxel features within cost volumes, are presented. We utilize pseudo-3D convolutions [10] to reduce computational costs when considering spatially adjacent voxel features.

B Training Details

Our network training procedure consists of two phases. In the first phase, we make batches where each one has *randomly picked* RGB-D images (four images in our setting) from the dataset. We use these batches to train all learnable parts except for the cross-attention module.

In the second phase, we make batches where each one has four *consecutive* RGB-D images to learn temporal fusion. Here, the first sample in the batch is used to train learnable parts except for cross-attention. Such batch configuration is necessary because the first frame does not have the cost volume of the previous frame to fuse. The second to last samples in the batch are used to train all parts, including cross-attention, that learn to fuse volumes incrementally.

Once the training finishes, we can utilize the network to fuse cost volumes incrementally. In addition, to handle the single-view settings, where only a single RGB-D image was given, we disabled only the cross-attention module in the trained network. With this scheme, we report all experimental results in the single-view setting.

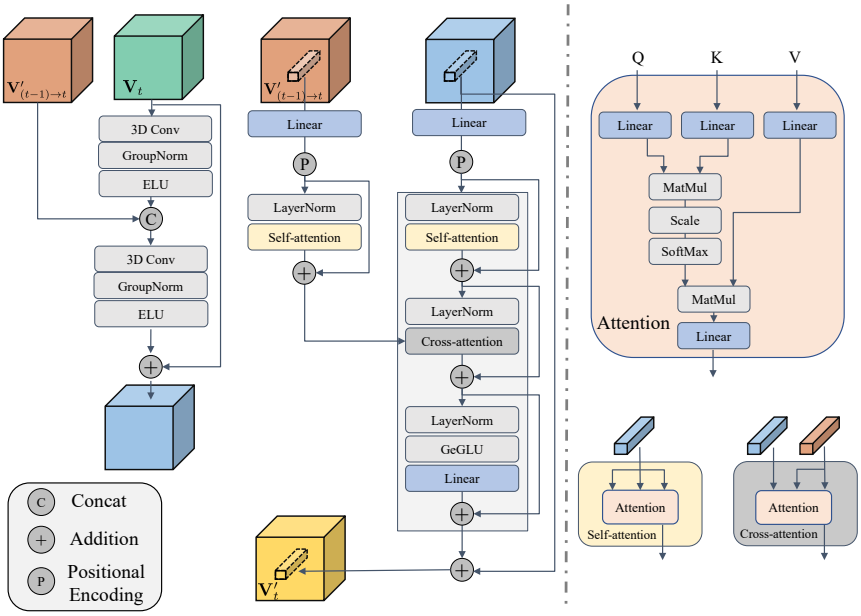

Fig. 8: Detailed architecture of our distribution-aware cost volume fusion module. '3D Conv' and 'Linear' denote pseudo-3D convolutions [10] and multilayer perceptrons.

Table 6: Comparison of the inference time and the GPU memory usage (left). The run-time of the proposed components (right).

Method	GPU Mem. (MiB)↓	Infer. Time (ms)↓
SimpleRecon [11]	5719	66
CostDCNet [4]	2273	30
NLSPN [9]	3348	70
ComplFormer [16]	3900	93
Ours	4438	77

Component	Time (ms)↓
Cost volume creation	28.0
Cost volume fusion	36.7
Depth regression	6.1
Depth refinement	6.2
Ours total	77.0

C GPU Memory Usage and Timings

We report the total GPU memory usage and inference time of our framework and its components, as well as those of some representative methods for comparison.(Table 6).

D RGB-D Feature Volume Creation

We elaborate on constructing occupancy $\mathbf{V}_o$, residual $\mathbf{V}_r$, and RGB feature $\mathbf{V}_i$ volumes, where $\mathbf{V}_o, \mathbf{V}_r, \mathbf{V}_i \in \mathbb{R}^{D\times C\times W\times H}$, used for RGB-D feature creation (Section 4.1, Figure 3).

Occupancy and residual volumes. We form $\mathbf{V}_o$ and $\mathbf{V}_r$ using a sparse depth image $\mathbf{S}_t$. Specifically, to make the binary-valued occupancy volume $\mathbf{V}_o$, we initially set all voxels in $\mathbf{V}_o$ to zero. Then, for only valid depth pixel positions (h, w) having depth value $v = \mathbf{S}(h, w)$, we find an index d of the depth plane closest to v and then set $\mathbf{V}_o(d, h, w) = 1$. The residual volume $\mathbf{V}_r$ contains normalized distances between depth planes and sparse depth samples. Specifically, for the voxels having $\mathbf{V}_o(d, h, w) = 1$, we calculate the normalized distance $(v - v_d)/m$, where $v = \mathbf{S}(h, w)$, v_d is the depth value of the d-th depth hypothesis plane, and m is the interval between depth planes.

Image feature volume. Image features extracted from an input RGB image make a volume $\mathbf{V}_i$. We build a spatial pyramid and concatenate multi-scale features [3] to leverage the image contexts at various scales. We denote a feature map as $f \in \mathbb{R}^{C \times W \times H}$. If $\mathbf{S}(h, w)$ does not have a valid depth, we copy $f(:, h, w)$ to all $d \in D$ voxels, denoted as $\mathbf{V}_i(d, :, h, w)$. Otherwise, we find the index d of the depth plane closest to the valid depth sample and set $\mathbf{V}_i(d, :, h, w) = f(:, h, w)$ [4].

E Cost Volume Alignment

We align two cost volumes $\mathbf{V}'_{t-1}$ and $\mathbf{V}_t$ for the temporal volume fusion (Section 4.2). Since the coordinate system of the two volume is not the same under the moving cameras, we warp $\mathbf{V}'_{t-1}$ to align with $\mathbf{V}_t$ of the current frame.

We employ *inverse mapping* for a reliable implementation. Specifically, we transform the center position $\mathbf{v} = [u, v, d]^\top$ of each voxel of $\mathbf{V}_t$ using camera intrinsics $\mathbf{K}$ and camera poses ($\mathbf{T}_{t-1}$ and $\mathbf{T}_t$) as follows:

$$[\tilde{d}\tilde{u}, \tilde{d}\tilde{v}, \tilde{d}, 1]^\top = \mathbf{K}\mathbf{T}_{t-1}^{-1}\mathbf{T}_t\mathbf{K}^{-1}[du, dv, d, 1]^\top.$$

Then, we resample $\mathbf{V}'_{t-1}$ at the transformed voxel coordinates $[\tilde{u}, \tilde{v}, \tilde{d}]^\top$ using trilinear interpolation. If the transformed voxel coordinates fall outside the boundary of $\mathbf{V}'_{t-1}$, we assign zero-valued vectors. By repeating this procedure for all voxels, we obtain a waped cost volume $\mathbf{V}'_{(t-1)\to t}$.

F Depth Refinement Module

Our approach uses non-local spatial propagation networks (NLSPN) [9] to refine regressed depths. NLSPN uses the input affinities between the current pixel and neighbors, a depth to be refined, and a confidence map of the depth to generate enhanced depths.

In prior work, the spatial propagation process is repeatedly applied (18 times in NLSPN) to refine the depth. In addition, state-of-the-art depth completion methods [7, 9, 16] employing spatial propagation networks commonly rely on heavy neural networks, such as Vision Transformer, to output high-quality regressed depth, affinity, and confidence maps simultaneously.

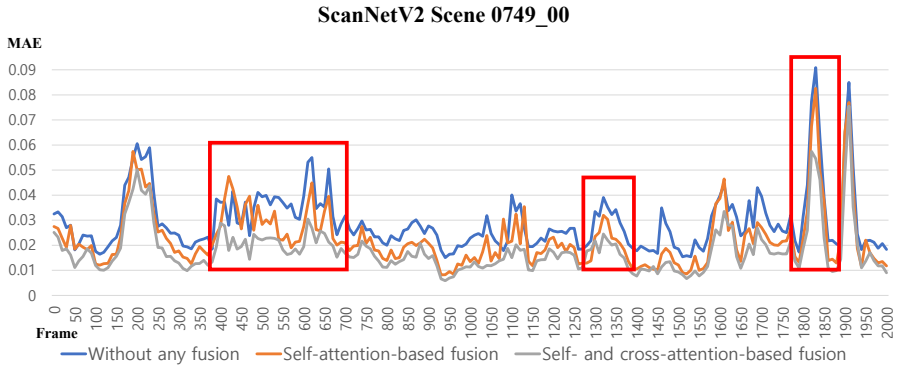

Fig. 9: Performance trend over time with various cost volume fusion schemes.

On the other hand, we use a shallower 2D convolutional network to estimate an affinity map since our regressed depth is accurate enough (Figure 3). Following the same rationale, we iterate the propagation process only six times. Note that without additional networks, a value of a confidence map $\mathbf{C}$ at a pixel position (h, w) is directly computed as $\mathbf{C}(h, w) = \mathbf{P}(d, h, w)$, where $\mathbf{P}$ is a predicted probability volume for regressing depth $\mathbf{D}(h, w)$, and d is the index of a hypothesis depth plane closest to a regressed depth $\mathbf{D}(h, w)$.

G Evaluation Metric

We employ the standard depth image error metrics [9], including mean absolute error (MAE), root mean square error (RMSE), mean absolute error of the inverse depth (iMAE), and root mean squared error of the inverse depth (iRMSE). For 3D reconstruction evaluation on the ScanNetV2 dataset, we measure 3D error metrics, including geometric accuracy (Acc.), geometric completeness (Compl.), Chamfer distance (Chamfer), precision (Prec.), recall, and F-score. We utilize the point sampling and thresholding method by Bozic et al. [1] for the evaluation. To obtain 3D meshes reconstructed from the inferred depth frames, we use a truncated signed distance function (TSDF) with a voxel size of 4cm, which is implemented in the Open3D library [17].

H Performance Trend over Time

We assess the performance trend of our fusion scheme over time on a selected RGB-D stream from the ScanNetV2 dataset (Figure 9). Note that our fusion module consists of self-attention (for the refinement of a single cost volume) and cross-attention (for the fusion of two cost volumes). We computed per-frame MAE values for three options: not using both the self- and cross-attention-based

fusion module, using only the self-attention part of the module, and using the whole fusion module.

As expected, a framework not using the fusion module consistently exhibits the poorest performance over time. Using only the self-attention part of our fusion module (single-view setting) shows the second-best performance. The framework exhibits the most superior performance when leveraging self- and cross-attention-based fusion. Especially our temporal fusion mitigates performance degradation, as indicated by the red boxes in Figure 9.

I Qualitative Results

We present additional visual results comparing our approach with other state-of-the-art depth completion methods on ScanNetV2 dataset [2], VOID Depth Completion benchmark [15], and the KITTI Depth Completion benchmark [14].

I.1 ScanNetV2 test set

We compare our model with CompletionFormer [16], NLSPN [9], and CostDC-Net [4], where all models are trained on the ScanNetV2 training set. Our model shows more accurate depths compared to the other methods, as shown in Figure 10 and our supplementary video. Furthermore, we observe that depth sequences completed by our method exhibit temporally coherent results.

I.2 VOID test set

We train our and other models (CompletionFormer [16], NLSPN [9], and Mondi [8]) on the VOID training set with 0.5% depth density. Subsequently, we assess their performance on VOID test sets with 0.5%, 0.15%, and 0.05% densities to analyze the sparsity-agnostic ability.

As shown in Figure 11, our method exhibits superior accuracy compared to other models on the VOID test set with 0.5% density. This superiority persists in test sets consisting of even sparser depth samples (0.15%, 0.05%), as demonstrated in Figures 12 and 13. The robustness of sparser depth samples is attributed to our ray-based fusion scheme that effectively boosts confident probability distributions (Figure 14).

I.3 KITTI validation set

We qualitatively compare depth maps inferred by our model, NLSPN, and CompletionFormer, which are all trained on the KITTI training set. The completed depth maps of NLSPN and CompletionFormer are visually pleasing, aligning well with structural features such as edges within an RGB image. Our completed depth maps show better accuracy in large areas, such as roads, that occupy a significant portion of the image, as shown in error maps of Figure 15.

I.4 Cross-dataset generalization

We conduct an additional experiment to evaluate the cross-dataset generalization ability of our model. We train our model, NLSPN, CompletionFormer, and CostDCNet on the ScanNetV2 training set, then test them on the VOID test set with 0.5% depth density.

In summary, our method predicts more accurate depth maps than other models (Figure 16). As shown in Figures 11 and 16, models trained on the ScanNetV2 training set generate more visually appealing completion results, exhibiting sharper edges, compared to those trained on the VOID training set, which comprises relatively low-quality ground truth depth maps.

References

1. Bozic, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: Transformerfusion: Monocular rgb scene reconstruction using transformers. Advances in Neural Information Processing Systems **34**, 1403–1414 (2021)
2. 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)
3. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
4. 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)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012), https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
6. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
7. 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)
8. 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)
9. 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)
10. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: proceedings of the IEEE International Conference on Computer Vision. pp. 5533–5541 (2017)

11. 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)
12. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: NeuralRecon: Real-time coherent 3D reconstruction from monocular video. CVPR (2021)
13. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2820–2828 (2019)
14. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: International Conference on 3D Vision (3DV) (2017)
15. 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)
16. 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)
17. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)

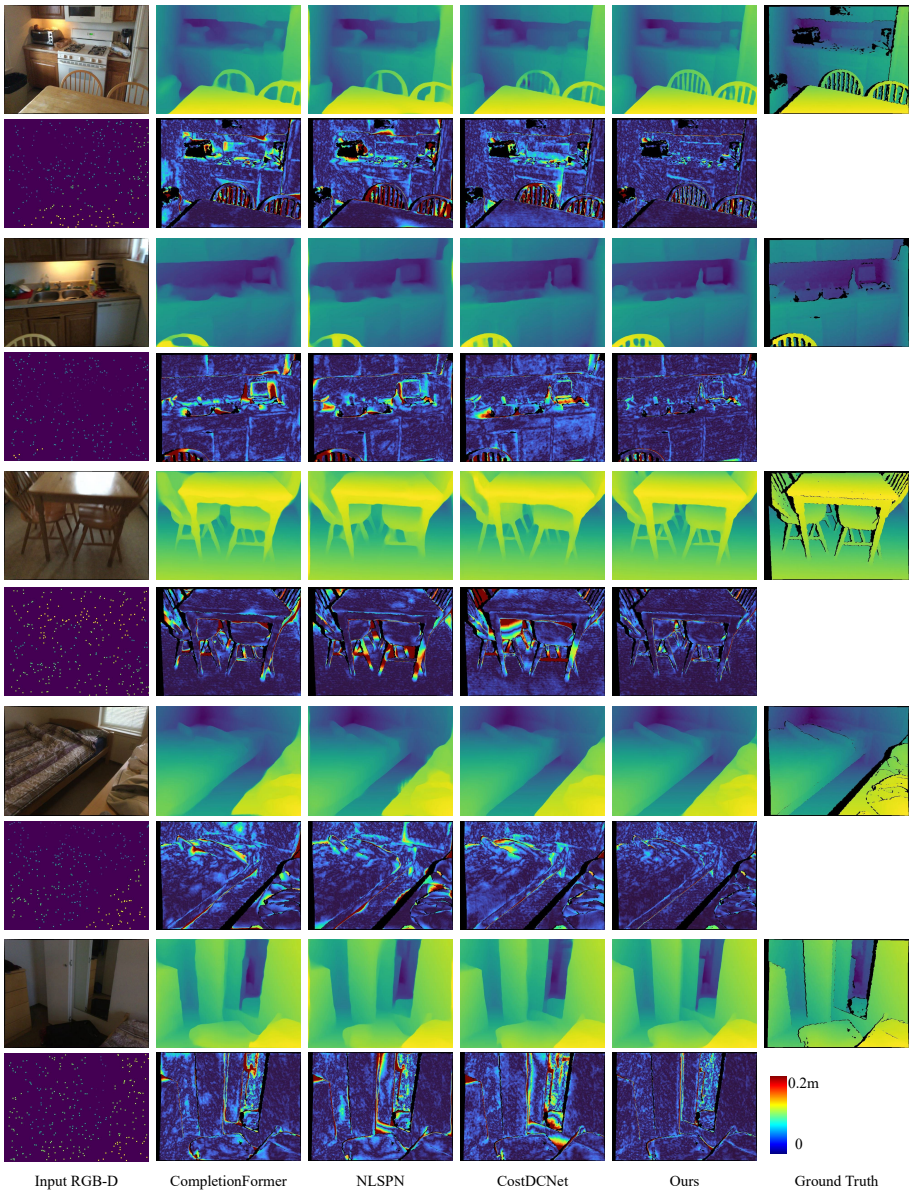

Fig. 10: Visual comparison of our framework with CompletionFormer [16], NLSPN [9], and CostDCNet [4] on the **ScanNetV2** test set with **0.1%** depth density. Completed depth images (upper rows) are presented alongside their error maps (lower rows) for each scene.

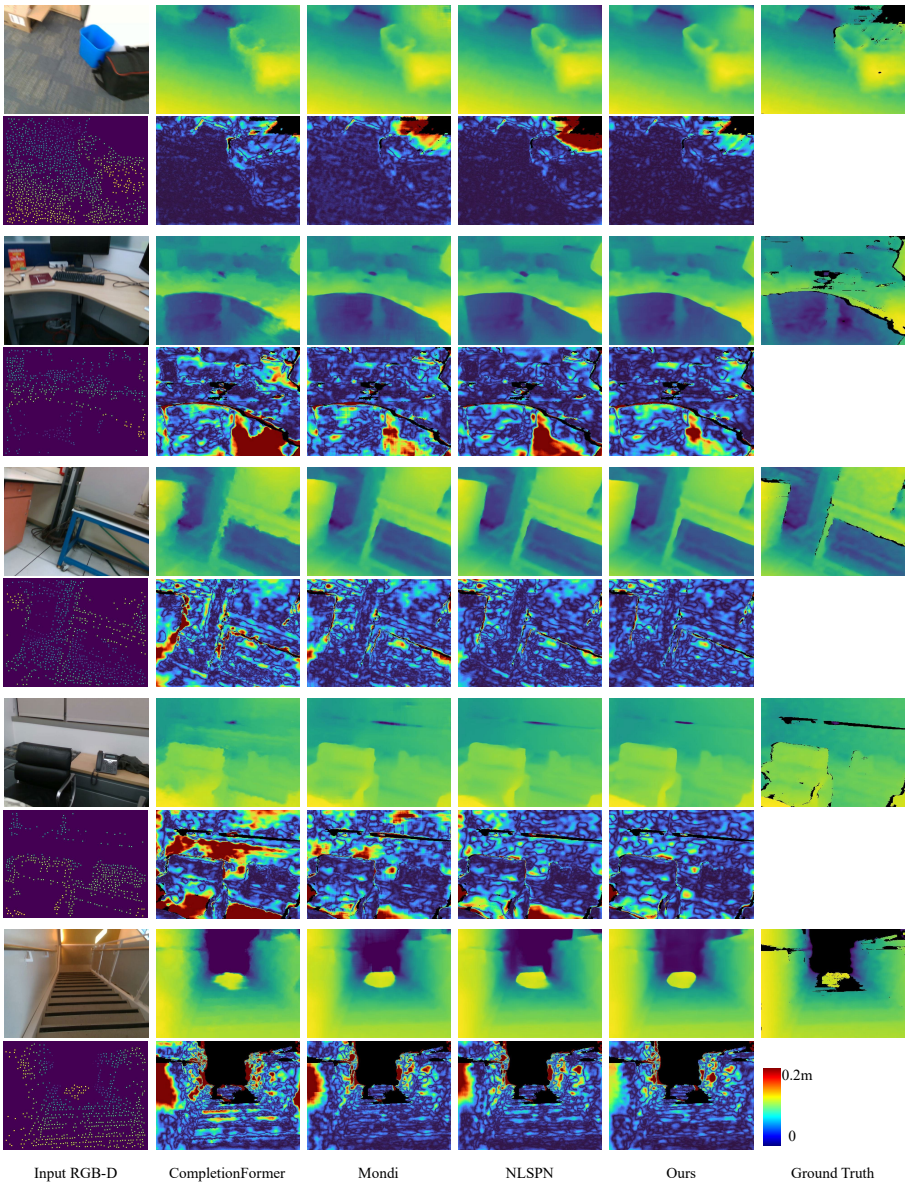

Fig. 11: Visual comparison of our framework with CompletionFormer [16], Mondi [8], and NLSPN [9] on the **VOID** test set with **0.5%** depth density. Completed depth images (upper rows) are presented alongside their error maps (lower rows) for each scene.

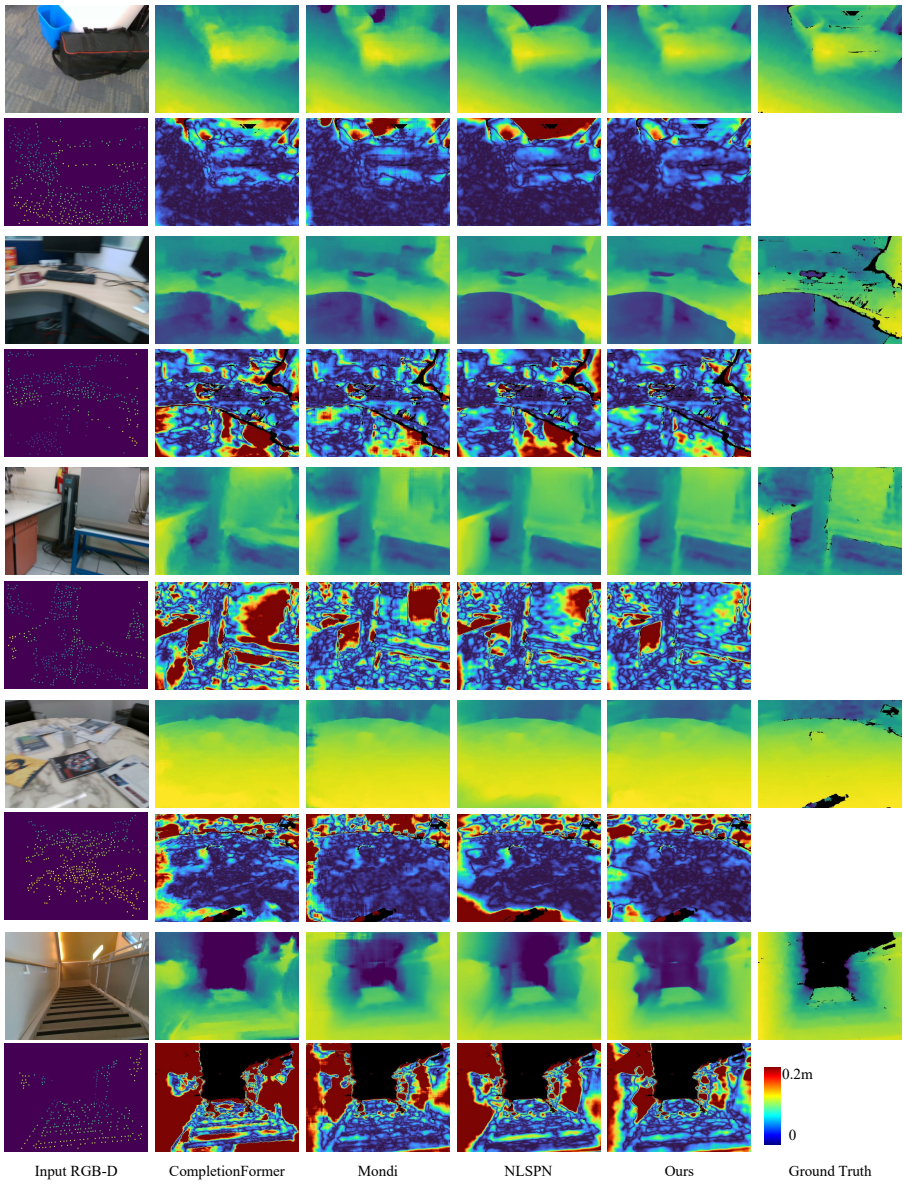

Fig. 12: Visual comparison of our framework with CompletionFormer [16], Mondi [8], and NLSPN [9] on the **VOID** test set with **0.15%** depth density. Completed depth images (upper rows) are presented alongside their error maps (lower rows) for each scene.

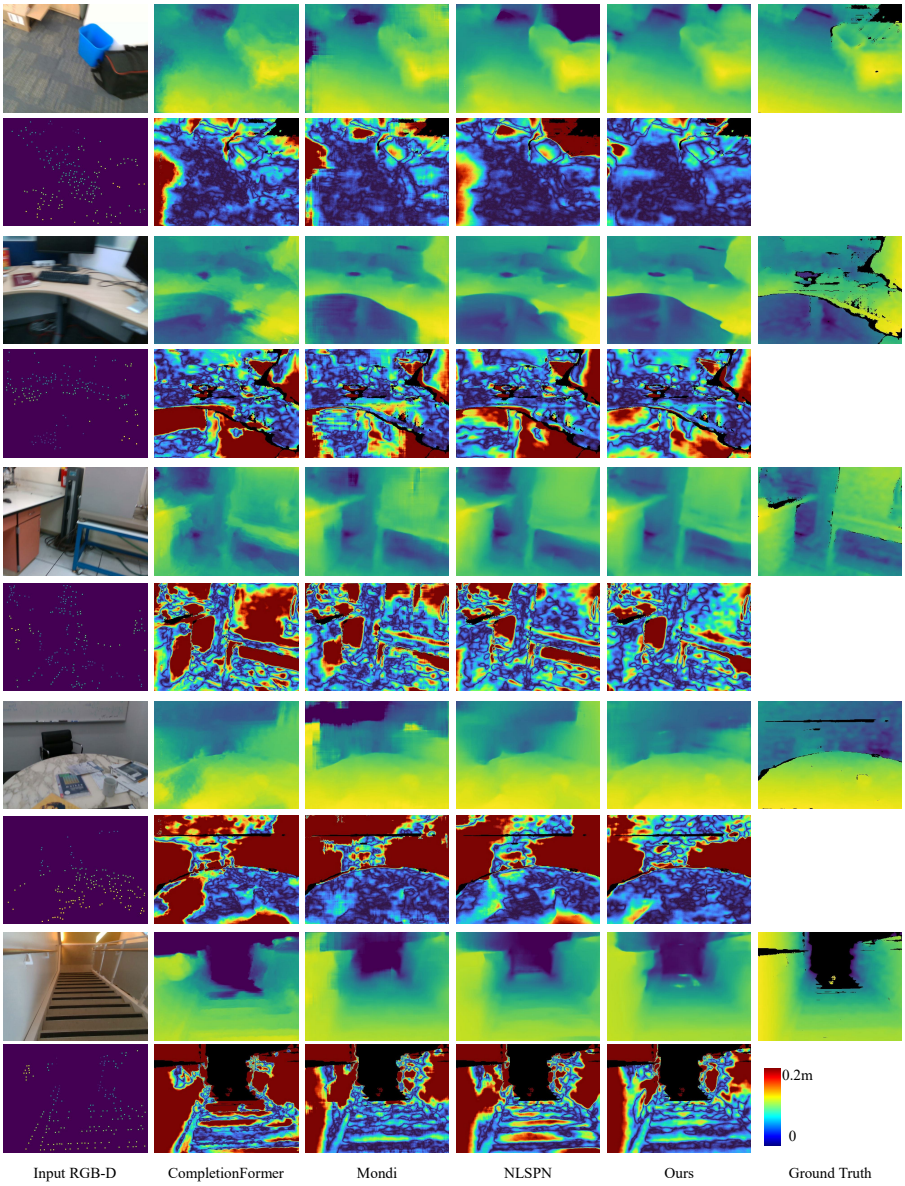

Fig. 13: Visual comparison of our framework with CompletionFormer [16], Mondi [8], and NLSPN [9] on the **VOID** test set with **0.05%** depth density. Completed depth images (upper rows) are presented alongside their error maps (lower rows) for each scene.

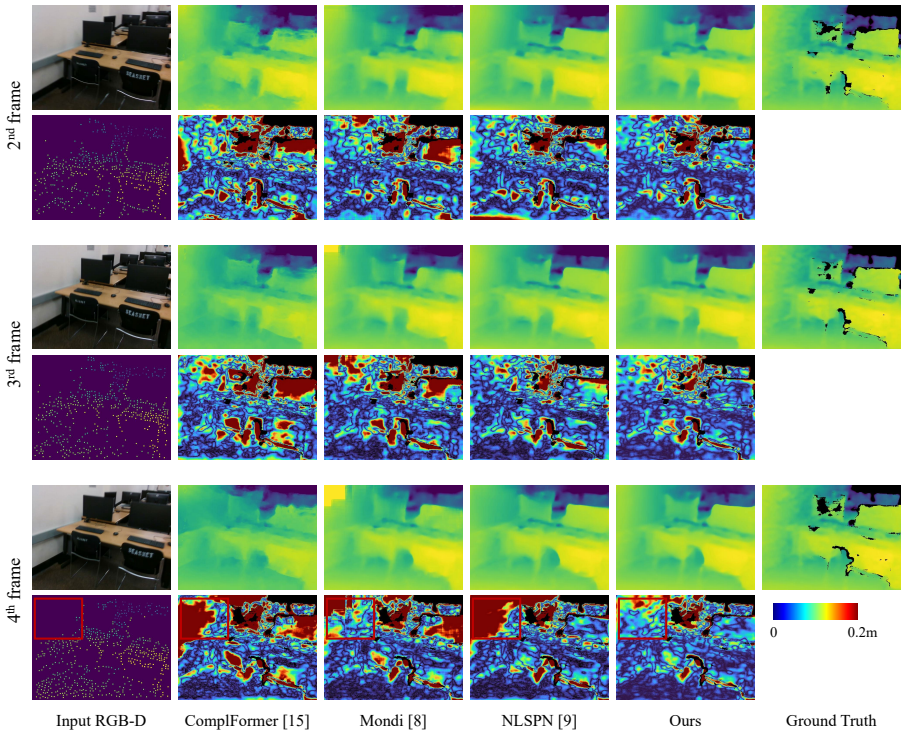

Fig. 14: Completion results for consecutive frames on the **VOID** test set with **0.5%** depth density. Thanks to our ray-based fusion module that leverages previous predictions, we can recover accurate depths in large missing regions, as the red box indicates.

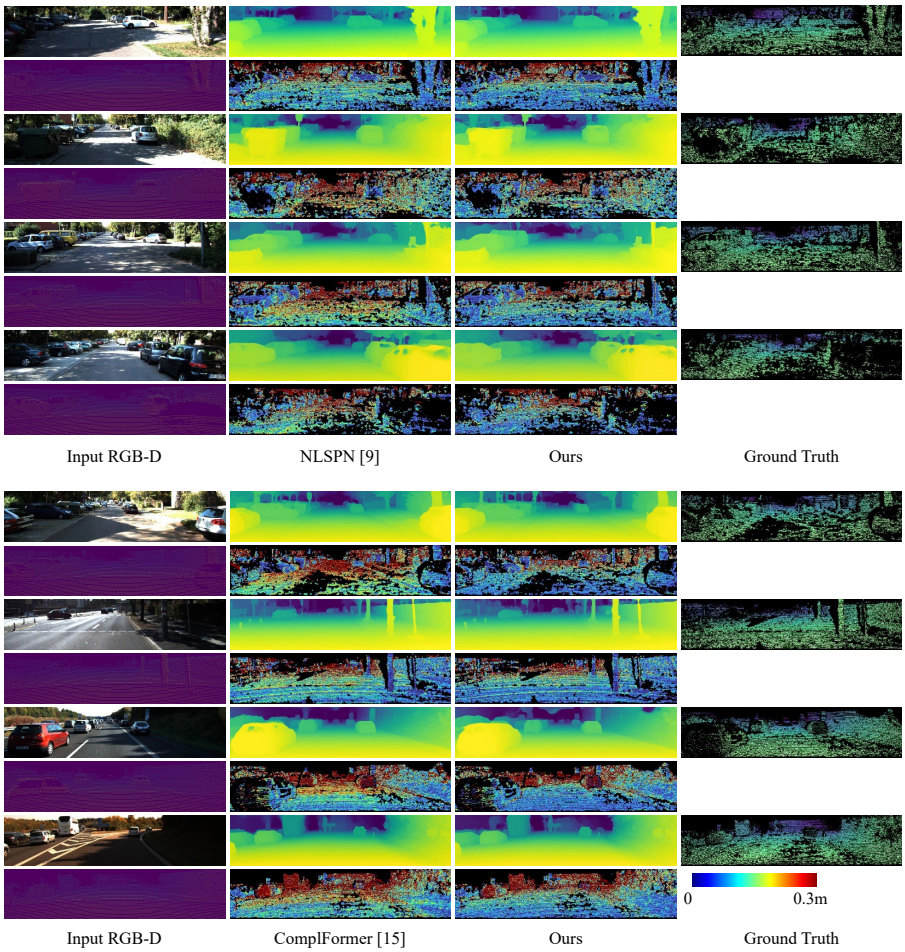

Fig. 15: Visual comparison of our framework with NLSPN [9] and CompletionFormer [16] on the **KITTI** validation set. Completed depth images (upper rows) are presented alongside their error maps (lower rows) for each scene.

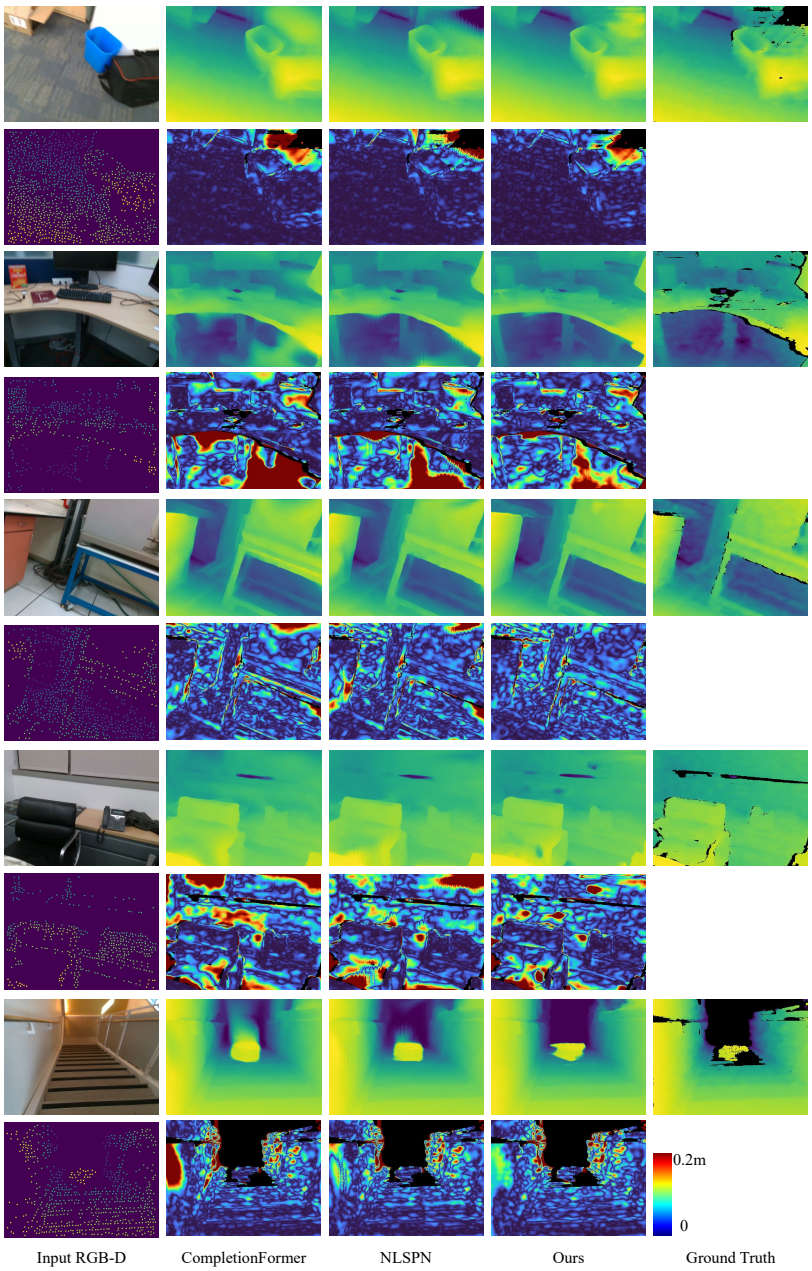

Fig. 16: Visual comparison of our framework with CompletionFormer [16] and NLSPN [9] on the VOID test set with 0.5% depth density in the **cross-dataset setting** (trained with the ScanNetV2 training set). Completed depth images (upper rows) are presented alongside their error maps (lower rows) for each scene.