Supplementary Information *for* StarryGazer: Leveraging Monocular Depth Estimation Models for Domain-Agnostic Single Depth Image Completion

Sangmin Hong[1†], Suyoung Lee[2†], Kyoung Mu Lee[1,2*]

[1]IPAI, Seoul National University, Seoul 08826, South Korea.
[2]Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 08826, South Korea.

*Corresponding author(s). E-mail(s): kyoungmu@snu.ac.kr;
Contributing authors: mchiash2@snu.ac.kr; esw0116@snu.ac.kr;
†These authors contributed equally to this work.

S1 Evaluation Metrics

We describe the formula for each evaluation metric used to assess performance. We employ standard metrics commonly used in depth estimation and completion tasks. We note that all metrics except δ_τ are better when they become lower.

- RMSE: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \left| d_v^{gt} - d_v^{pred} \right|^2}$
- MAE: $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \left| d_v^{gt} - d_v^{pred} \right|$
- iRMSE: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \left| 1/d_v^{gt} - 1/d_v^{pred} \right|^2}$
- iMAE: $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \left| 1/d_v^{gt} - 1/d_v^{pred} \right|$
- $$\mathrm{SILog} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left(\log d_v^{\mathrm{pred}} - \log d_v^{\mathrm{gt}}\right)^2 - \left(\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left(\log d_v^{\mathrm{pred}} - \log d_v^{\mathrm{gt}}\right)\right)^2$$
- Rel: $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \left| (d_v^{gt} - d_v^{pred})/d_v^{gt} \right|$
- δ_τ: Percentage of pixels satisfying $\max\left(\frac{d_v^{gt}}{d_v^{pred}}, \frac{d_v^{pred}}{d_v^{gt}}\right) < \tau$

S2 Scale Invariant Metric in Global and Segment Scale

We show the superiority of our method over the simple combination of an affine transformation and an MDE model in Table 1. To further justify synthetic pair generation with random rescaling independently for each segment, we measure SILog metrics in two scales, global and segment, and report the results in Table S1. The result of a typical SILog calculation between the estimated depth map and the ground truth depth map is shown in the 'global' column, which is identical to Table 1. For the segment scale, we first segment the MDE depth map with the segmentation network (f_{seg}), compute the SILog for each segment, and average the calculated SILog metrics. Since we measure scale invariant accuracy within each segment, depth error between segments is not calculated. The segment scale SILog is written in the 'segment' column.

The significant reduction in SILog when computed segment-wise reveals a key insight: while the global scale of monocular depth predictions is unreliable, local regions often exhibit internally

Table S1: Comparison of scale invariant metric (SILog) in two scales (global, segment). The values are multiplied by 100.

Method	Global	Segment
DepthAnything with global affine transformation	184.87	8.51
DepthAnything with segment-wise affine transformation	1.754	1.18
Ours	**0.022**	**0.023**

consistent depth structures. This observation provides strong empirical motivation for our proposed method—refining relative depth predictions by segmenting and rescaling each region individually to synthesize training data. By leveraging this piecewise geometric consistency, **StarryGazer** circumvents the need for ground-truth depth while achieving high accuracy across domains.

S3 Comparison with Additional Baseline Methods

S3.1 Supervised Depth Completion Methods

We mostly compared our method with unsupervised depth completion approaches in the main manuscript. Here, we list the results of several state-of-the-art supervised depth completion methods in Table S2 and S3 for NYU Depth V2 and KITTI DC dataset, respectively. Considering that the supervised methods require ground truth dense depth maps when training the network, our method achieves reasonable performance to the supervised methods without using the ground truth.

S3.2 Monocular Depth Estimation Methods

As shown in Tables S4 and S5, we present comparisons with several state-of-the-art MDE methods on the NYU Depth V2 and KITTI DC datasets. Unlike MDE methods, which rely solely on RGB input, our approach leverages both RGB and sparse absolute depth data. Our method consistently outperforms the MDE methods in both datasets, achieving the lowest RMSE and Rel values, along with high $\delta_{1.25}$ scores, clearly demonstrating the effectiveness of integrating sparse depth information to improve estimation accuracy. On the NYU Depth V2 dataset, our method achieves an RMSE of 0.171 and a Rel of 0.039, and on the KITTI DC dataset, it achieves an RMSE of 1.061 and a Rel of 0.039, outperforming the MDE methods. These results highlight the importance of combining RGB and sparse depth inputs to produce more reliable and accurate depth estimations.

S4 Experiments on Rendered Handpose Dataset

As originally designed for hand pose estimation, the Rendered Handpose dataset [25] consists of RGB images, corresponding depth maps, segmentation masks, and key point locations. Since there is no depth information on the background of humans, we train the model only on the human region of the segmentation mask. It consists of 41,258 and 2,728 training and testing pairs, respectively.

The quantitative results in Table S6 show that our method estimates much more accurate depth values than the unsupervised or MDE-based methods. Figure S1 presents the visual results on the Rendered Handpose dataset. In this dataset, sparse depth points are available only in the human region, with no depth information provided for the background. The absence of background depth data makes it particularly challenging to estimate background depth values accurately. Nevertheless, our method accurately estimates depth values for points on the human figure, achieving a close approximation to the ground truth. In contrast, the competing method struggles to capture the fine details of the human figure. These results underscore our approach's superior out-of-domain generalization, highlighting the model's robustness.

Table S2: Quantitative Comparison with Supervised Depth Completion Methods on NYU Depth V2.

Method	RMSE ↓	Rel ↓
CSPN [1]	0.117	0.016
DeepLiDAR [5]	0.115	0.022
GuideNet [6]	0.101	0.015
NLSPN [4]	0.092	0.012
ACMNet [45]	0.105	0.015
TWISE [9]	0.097	0.013
RigNet [7]	0.090	0.012
DySPN [3]	0.090	0.012
SpAgNet [10]	0.114	0.015
CompletionFormer [8]	0.090	0.012
Ours	0.171	0.039

Table S3: Quantitative Comparison with Supervised Depth Completion Methods on KITTI DC.

Method	MAE ↓	RMSE ↓
CSPN [1]	0.279	1.019
DeepLiDAR [5]	0.226	0.758
GuideNet [6]	0.218	0.736
NLSPN [4]	0.199	0.741
PENet [2]	0.210	0.730
ACMNet [45]	0.206	0.744
TWISE [9]	0.195	0.840
RigNet [7]	0.203	0.712
GuideFormer [11]	0.207	0.721
DySPN [3]	0.192	0.709
CompletionFormer [8]	0.203	0.708
SemAttNet [38]	0.205	0.709
Ours	0.242	1.061

Table S4: Quantitative comparison with MDE methods on NYU Depth V2 dataset.

Method	RMSE ↓	$\delta_{1.25}$ ↑	Rel ↓
ZoeDepth [15]	0.277	0.953	0.077
ZeroDepth [16]	0.269	0.954	0.074
NeWCRFs [28]	0.334	0.922	0.095
IEBins [29]	0.314	0.936	0.087
Metric3D [19]	0.187	0.987	0.045
DepthAnything [18]	0.206	0.984	0.056
DepthAnything V2 [37]	0.206	0.979	0.044
Ours	**0.171**	**0.999**	**0.039**

Table S5: Quantitative comparison with MDE methods on KITTI DC dataset.

Method	RMSE ↓	$\delta_{1.25}$ ↑	Rel ↓
ZoeDepth [15]	2.281	0.971	0.053
ZeroDepth [16]	2.087	0.968	0.057
NeWCRFs [28]	2.129	0.974	0.052
IEBins [29]	2.011	0.978	0.050
Metric3D [19]	1.766	0.989	0.039
DepthAnything [18]	1.896	0.982	0.046
DepthAnything V2 [37]	1.861	0.983	0.045
Ours	**1.061**	**0.995**	**0.039**

Table S6: Quantitative comparison on Rendered Handpose.

Method	MAE (m)↓	RMSE (m)↓
NLSPN [4]	0.485	0.563
CompletionFormer [8]	0.443	0.524
Depth Prompting [47]	0.700	0.797
Metric3D V2 [35] (global)	0.479	0.564
Metric3D V2 [35] (segment)	0.447	0.532
UniDepth V2 [36] (global)	0.407	0.526
UniDepth V2 [36] (segment)	0.398	0.478
Ours	**0.352**	**0.408**

S5 Additional Ablation Studies

S5.1 Change of Performance according to the Number of Sparse Depth Points

In Table S7, we compare our method with KBNet [48] according to the number of points in input sparse depth maps. Our method shows better MAE and RMSE values for all tested configurations (50, 200, 500, and 2000 points). This validates the robustness of our approach in handling different levels of sparsity. Moreover, the tendency for a larger performance gap when the number of points gets smaller shows that our model is capable of producing highly accurate estimation results even with a small amount of given information. We present a qualitative analysis of our method in varying levels of input sparsity

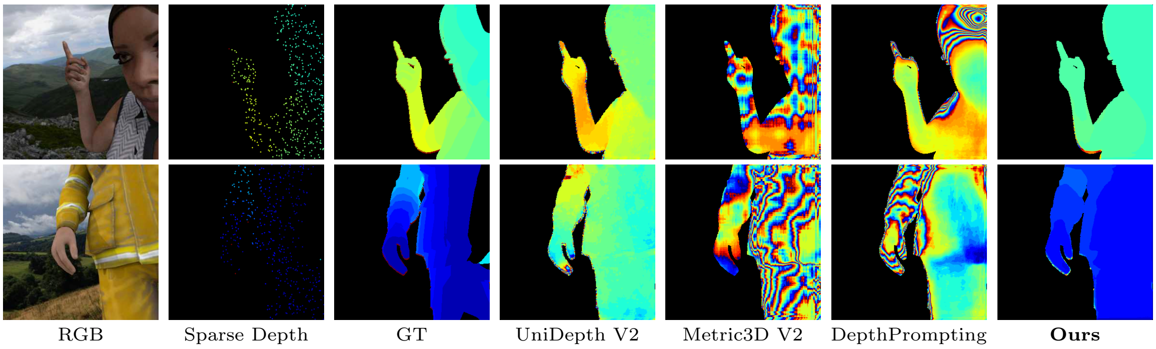

Fig. S1: Qualitative results on the Rendered Handpose dataset (out-of-domain). Sparse depth points exist only on the human figure; the background has no depth.

Table S7: Sparsity analysis on NYU Depth V2: Quantitative comparison with KBNet using MAE and RMSE across different numbers of sparse input points (50, 200, 500, 2000).

# of Points	50		200		500		2000	
Method	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓
KBNet [50]	0.300	0.450	0.180	0.280	0.106	0.198	0.102	0.170
Ours	**0.256**	**0.400**	**0.150**	**0.240**	**0.100**	**0.171**	**0.097**	**0.146**

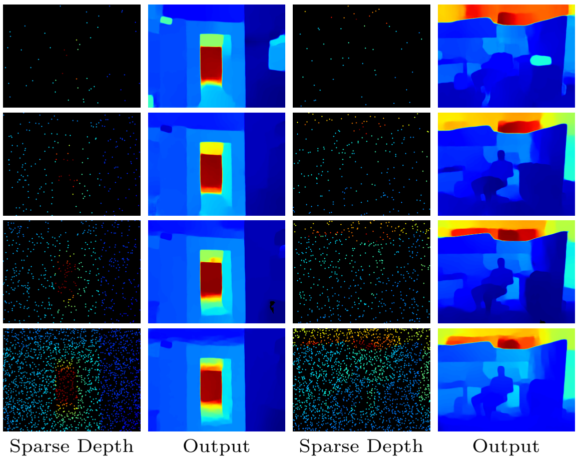

Fig. S2: Qualitative analysis of **StarryGazer** with varying sparse point counts. Rows correspond to n=50, 200, 500, and 2000 points.

in Figure S2. It illustrates how the quality of depth completion improves as the number of input points (n) increases. Specifically, at n = 50, the depth map successfully captures most of the structure of the scene, showing the model's robustness even with a very sparse input despite the slight loss of some finer details. As the number of input points increases, from n = 200 to n = 500, the depth maps begin to more accurately represent the scene. The most accurate estimation is observed when n = 2000, where the depth map is considerably more detailed. This progressive enhancement underscores the importance of input density in achieving precise depth estimation, as higher point counts provide richer spatial information for the model to leverage.

Table S8: Ablations on filling the gaps on NYU Depth V2.

Method	MAE ↓	RMSE ↓
Masking out the gaps	0.141	0.232
Ours	**0.100**	**0.171**

S5.2 Effect on Filling the Gaps after Rescaling

In our training pipeline (Stage 1), gaps with no depth information are produced in synthetic depth maps because of the missing data in the segmentation masks; due to the segment sensitivity of the segmentation network, there are regions that do not belong to any segment, and such regions are transformed into gaps after the rescaling process. While just masking out these regions when calculating the loss may be a valid alternative, it can lose valuable spatial information that could contribute to the overall learning process. Moreover, masking can make the training more complicated by introducing discontinuities inside the depth map. As described in Section 3.2, we conduct a gap-filling process to ensure continuity of the synthetic depth map. The gap-filling process can be described as follows:

1. Identify non-zero elements to determine regions with depth data.
2. Compute the average value within the non-zero neighborhood of each zero-valued pixel by applying a 5 × 5 convolution operation, effectively smoothing over gaps.
3. Update only the zero-valued elements in the depth image, preserving original depth values where they exist.

This process enhances the continuity and quality of the synthetic depth images, which is crucial for effectively training the depth completion model. We conduct experiments to compare the effect of applying an average filter to fill in the gaps versus masking the gaps from the loss for training. The results are presented in Table S8. By utilizing the average filter to fill gaps or holes, the process yields a more consistent and continuous training dataset, which in turn enhances the model's overall performance.

Table S9: Performance comparison with different Monocular Depth Estimation (MDE) models on the NYU Depth V2 dataset.

MDE Model	MAE ↓	RMSE ↓
ZoeDepth [15]	0.092	0.171
Metric3D V2 [35]	0.112	0.186
UniDepth V2 [36]	0.070	0.152
DepthAnything [18]	0.100	0.171
DepthAnything V2 [37]	0.095	0.169

S5.3 Ablations on the Type of backbone MDE models

As shown in Table S9, we evaluate the effect of MDE models on the final performance by replacing the DepthAnything [18] model with other Monocular Depth Estimation (MDE) models. Compared to DepthAnything, UniDepth V2 [36] shows substantially better performance, reducing MAE by about 30% and also improving RMSE. We attribute the result to the better generalizability of the model.

Table S10: Inference time comparison with existing depth completion methods.

Method	Inference time (ms)
SS-S2D [12]	80
DFuseNet [13]	80
DDP [44]	80
VOICED [51]	44
AdaFrame [53]	40
SynthProj [49]	60
ScaffNet [52]	32
KBNet [50]	16
Ours	**89**

S6 Inference Time Comparison

We measure and compare the inference time in Table S10 with a target depth map resolution of 304 by 228 pixels. While our methods require two large models (monocular depth estimation and semantic segmentation) for training, only the MDE model is used in the inference phase since we do not generate synthetic pairs. In the main experiment, we use DepthAnything with a ViT-S backbone that has the smallest number of parameters among the available configurations to mitigate the increase in inference time. Despite showing a longer inference time, we argue that our method is still meaningful, considering the improved performance and the practical applicability of our approach.

Table S11: Ablations on the type of segment maps used for generating synthetic dense depth maps.

M_{seg}	I_{seg}	MAE ↓	RMSE ↓
✗	✓	0.167	0.266
✓	✓	0.124	0.188
✓	✗	**0.100**	**0.171**

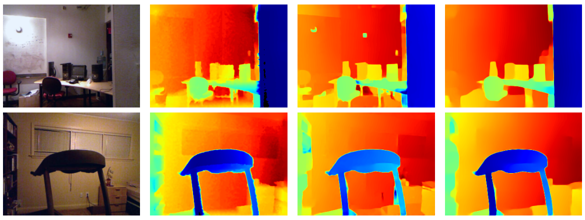

Fig. S3: Qualitative comparison of depth completion using RGB-based segmentation (I_{seg}) and relative depth-based segmentation (M_{seg}) for synthetic data generation.

S7 Alternative Approach without MDE Models

In our approach, synthetic training pairs are typically generated by applying segmentation to relative depth maps using MDE models [36, 37]. However, MDE models may struggle in complex scenes, leading to inaccuracies. To address the challenge, we propose an alternative approach that bypasses the need for MDE models by directly segmenting the RGB image and applying affine transformations based on grayscale values to create synthetic pairs.

We compare three strategies for generating synthetic dense depth: using segmentation from depth maps (M_{seg}), from RGB images (I_{seg}), and a mixed strategy randomly choosing between the two per iteration. As shown in Table S11, M_{seg} consistently yields the most accurate results, with lower MAE and RMSE. I_{seg} performs worse due to over-segmentation in regions with uniform depth, while the mixed strategy gives intermediate results.

Qualitatively shown in Figure S3, M_{seg} produces smoother and more stable depth predictions, while I_{seg} can better preserve fine boundaries. This flexibility allows our method to adapt depending on the availability and quality of MDE outputs. M_{seg} is ideal when MDEs are reliable, while I_{seg} provides a viable alternative in their absence.