BIPS: Bi-modal Indoor Panorama Synthesis via Residual Depth-aided Adversarial Learning –Supplementary Material–

Changgyoon Oh[1]*, Wonjune Cho[2]*, Yujeong Chae[1]*, Daehee Park[1]*, Lin Wang[3], and Kuk-Jin Yoon[1]

[1] Visual Intelligence Lab., KAIST
{changgyoon, yujeong, bag2824, kjyoon}@kaist.ac.kr
[2] NAVER LABS
wonjune.cho@naverlabs.com
[3] AI Thrust, HKUST Guangzhou and Dept. of CSE, HKUST
linwang@ust.hk

Abstract. Due to the lack of space in the main paper, we provide more details of the proposed method and experimental results in the supplementary material. Specifically, in Sec.1, the detailed architectures and loss functions proposed in our BIPS framework are described. Sec.2 provides implementation details. Sec.3 provides more details about GT layout and residual depth generation. Sec.4 shows verification of FAED score. Sec.5 provides more experimental results with 3D indoor models.

1 Detailed Architecture and Losses

1.1 Generator and Discriminator

As mentioned in Sec. 3.2 of the main paper, we provide the enlarged versions of our proposed network architectures for generator G in Fig. 1 and discriminator D in Fig. 3.

1.2 Losses

For training G, we define pixel-wise loss L_{pixel} along with L_{adv}:

$$L_{\text{pixel}} = \mathbb{E}\left[||I_{\text{out}}^{\text{total}} - I_{\text{in}}^{\text{total}}||_1\right]. \tag{1}$$

The D is trained once before every iteration of training G, by minimizing L_{D}:

$$L_{\text{D}} = \frac{1}{2}\,\mathbb{E}\left[(D(I_{\text{in}}^{\text{total}}) - 1)^2\right] + \frac{1}{2}\,\mathbb{E}\left[(D(I_{\text{out}}^{\text{total}}))^2\right]. \tag{2}$$

1.3 Auto-Encoder Network

As mentioned in Sec. 3.3 of the main paper, we provide the enlarged versions of our proposed network architectures for auto-encoder A in Fig. 2.

* Equal contribution.

2 Implementation Details

When using Matterport3D and 2D-3D-S datasets, we excluded data that has more than 1% of invalid pixels among 5121024 pixels for both RGB and depth. In the experiments, we first clip the incorrectly stored depth value. We scale the depth values such that the maximum depth value to be $D_{max} = 10$, as we find this balances the magnitudes of RGB and D well (note that most depth values lie near 0 than D_{max}). All the RGB-D panorama, either ground truths, generated samples, or the ones being evaluated, have a 512×1024 resolution.

We train the generator G using the Adam optimizer with parameters $\alpha = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Here, α is fixed until half of the maximum epochs, and then linearly decayed to zero. We set λ as 20 and maximum epochs as 100. The batch size is set to 2 for all the cases. We trained our model for 100 epochs with a single A6000 GPU which took 7 days. The inference time of G is 15ms. We train the auto-encoder A using the Adam optimizer [2] with parameters $\alpha = 0.0001$ with 0.99 decay, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The training is done for 60 epochs with batch size of 8. The effectiveness of FAED is validated on the whole dataset.

3 GT layout and residual depth generation

Residual Depth-aided Adversarial Learning (RDAL) uses layout depth data $I_{\text{gt}}^{\text{d,lay}}$, and residual depth data $I_{\text{gt}}^{\text{d,res}}$ simultaneously. Therefore, we generate GT layout and residual depth data. Using corner coordinate, plane equation and camera coordinate, we propose the method to estimate layout depth mathematically. The detailed procedure of the method is as follows: (1) Calculate line equation of ceiling layout. (2) Draw a ray in a 360 degree direction around the camera center. (3) Find the line where the rays first meet and calculate the distance from camera center to line. (4) Repeat on the floor layout and fill the rest part. Finally, we generate layout depth and estimate residual depth by simple subtraction.

4 Verification of FAED

4.1 Perceptual quality

In this section, we provide implementation details of experiments for the verification of FAED described in Sec. 4.1 of the main paper. When X denotes the data to be corrupted (either RGB image or depth map), we generate the corrupted data as follows:

Gaussian Blur We apply the convolution with Gaussian kernel with standard deviation $\alpha \in \{0, 1, 2, 4\}$ to X.

Gaussian Noise For RGB images, we normalize the Gaussian noise N to $[-1, 1]$, and for depth maps, we normalize N to $[-0.1, 0.1]$. Then, we linearly combine N and X with a factor α, as $(1 - \alpha)X + \alpha N$ for $\alpha \in \{0, 0.25, 0.5, 0.75\}$. Note

that, as mentioned in the paper, we normalize the RGB values to lie in $[-1, 1]$ and depth values to lie in $[0, D_{\max}]$ ($D_{\max}$=10).

Uniform Patches We replace five random regions of X with uniform-valued rectangular patches. The size of each patch is randomly set, such that the sum of the areas of replaced patches covers α portion of X, with $\alpha \in \{0, 0.25, 0.5, 0.75\}$. For RGB images, the uniform value of a patch is set to be -1, and for depth maps, the value is set to be 0.

Swirl We apply swirl transformation to X. That is, for a center location (x_0, y_0) of X, we compute the angle,θ, from the center and radius, r, of a pixel at (x, y) as:

$$\theta = \tan^{-1}\left(\frac{y - y_0}{x - x_0}\right), \quad r = \sqrt{(x - x_0)^2 + (y - y_0)^2}.$$

Then, we normalize r and transform θ as:

$$r' = \frac{6r}{\sqrt{x_0^2 + y_0^2}}, \quad \theta' = \theta + \alpha e^{-5r'/(\ln 2\rho)}.$$

Finally, the pixel at (x', y') in the transformed X' has the value of X at $(x_0 + r'\cos(\theta'),\ y_0 + r'\sin(\theta'))$. We set $\rho = 25$, and $\alpha \in \{0, 1, 2, 4\}$.

Salt and Pepper We change the pixel values to either 1 or -1 for RGB images, and 1 or 0 for depth maps, where the value is randomly chosen for each pixel. We control the number of changed pixels as $\alpha \times (512 \times 1024)$ out of the whole 512×1024 pixels, with $\alpha \in \{0, 0.1, 0.2, 0.3\}$.

Discrete Cosine Transform (DCT) We apply discrete cosine transformation and remove high-frequency components of X. Specifically, a ratio α of high-frequency DCT coefficients on each spatial dimension (width and height) are set to zero, resulting in only using $(1-\alpha)^2$ of the whole coefficients. Since most of the high-frequency components do not contribute to creating noticeable artifacts, we set α to be rather high: $\alpha \in \{0, 0.80, 0.85, 0.90\}$. This effectively mimics GAN-like, tiled artifacts by only preserving a few low-frequency components of the data.

In Fig. 5, we show the experimental results for the verification of FAED score. We also include the enlarged version of the Fig. 5 of the main paper.

4.2 Semantic Alignment

In Fig. 4, the visual results of the semantic inconsistency are provided. When the RGB panorama and the depth panorama are not aligned, the 3D indoor model has inconsistent semantic information, e.g., misaligned corner of indoor room and distorted furniture, which makes the 3D indoor model unrealistic and higher FAED score. Consequently, it indicates that the higher FAED score denotes poorer semantic alignment between RGB and depth panorama.

5 RGB-D Panorama Synthesis

5.1 Evaluation on RGB Panorama Synthesis

In Fig. 6-7, we show more qualitative results of RGB panorama evaluation as mentioned in Sec. 4.2 of the main paper. It can be seen that our method outperforms the image inpainting, outpainting, and panorama synthesis methods.

5.2 Evaluation on Depth Panorama Synthesis

In Fig. 8, we show more qualitative results of depth panorama evaluation as mentioned in Sec. 4.2 of the main paper. It can be seen that our method outperforms the image-guided depth synthesis methods.

Details of 2D layout IoU and the results. Similar to [1], we used the top view of the predicted room layout to compute 2D IoU. The difference is that we got the top view of the room layout from a fully generated depth map by projecting 3D points on the XY plane while [1] directly used predicted room boundaries and corners for evaluation. The visual comparison of our method and 'ours without RDAL' with 2D layout IoU is shown in Fig. 9.

5.3 Evaluation on RGB-D Panorama Synthesis

In Fig. 10-11, we show more results of synthesized RGB-D panorama and their 3D indoor models using our proposed method. It can be seen that our method can handle various sensor configurations and generate realistic outputs via mutual gain between RGB and depth information.

In Fig. 12, we show the input RGB-D data and our RGB-D panorama synthesis results intuitively. We visualize the input RGB-D data by converting the partial depth map into partial 3D indoor model and color the partial 3D indoor model if the corresponding RGB value is given. Our method successfully synthesizes highly perceptual 3D indoor model with precise indoor layout and realistic interiors.

5.4 Evaluation on Real Dataset

In Fig. 13, we show more qualitative results of our synthesized RGB-D panorama on real indoor scenes in Matterport3D and 2D-3D-S dataset, as mentioned in Sec. 4.2 of the main paper. Our method synthesizes high-quality RGB-D panorama on real indoor scenes, which are unseen during training.

5.5 Analysis on Robustness of BIPS.

Results with Different Input Data for the Same Scene. We analyzed the effect of different sensor configurations in the same scene. As shown in Fig. 14, various layouts and interiors can be generated conditioned on different sensor configurations. It can be seen that as the input sensor configuration changes,

the generated result also changes correspondingly. Given a complex and visually heterogeneous input, the output panorama results in corresponding complexity. On the other hand, if a homogeneous input (containing a single planar component) is given, simple layout and residuals are generated accordingly. Note that in both cases, our method produces realistic panorama, regardless of the amount of input information.

Generalization for Unseen Input Configurations. To show the robustness against unseen configurations of sensors, we tested our model with cases not covered during training. To this end, we sample unused sensor parameters: $\{\delta_H, \delta_V\} \sim \mathcal{U}[30°, 60°]$ & $\mathcal{U}[90°, 120°]$, for RGB and perspective depth sensors, and $\{\delta_L, \delta_U\} \sim \mathcal{U}[4\eta, 5\eta, 6\eta]$ & $\mathcal{U}[0.25\eta, 0.5\eta, 0.75\eta]$ for LiDARs. Also, we consider various intervals between inputs. Examples of unseen configuration are shown in Fig. 15. The FAED under unseen input configurations is 157.4, while the FAED under the preset input configuration used in training is 198.0. *This means that the preset input configuration is practical and sufficient to generalize the model.* The reason that under unseen input configurations case shows the better FAED is that larger FoV inputs provide more amount of information.

Robustness against Noisy Input Data. To show the effectiveness of our model in real-world data, which has sensor noise or invalid pixels, we further constructed test data with noisy inputs. Three sensor noises are randomly applied to the dataset: (1) Gaussian noise of 10% of pixel value, (2) Gaussian blur with standard deviation 1, (3) Pepper noise to 10% of entire pixels in depth data to mimic invalid pixel in depth measurement. The FAED score of the output data is 202.9, which is comparable with 198.0 of results without noisy inputs. It shows that our model is robust against noisy RGB-D input and can be easily applied to real-world data.

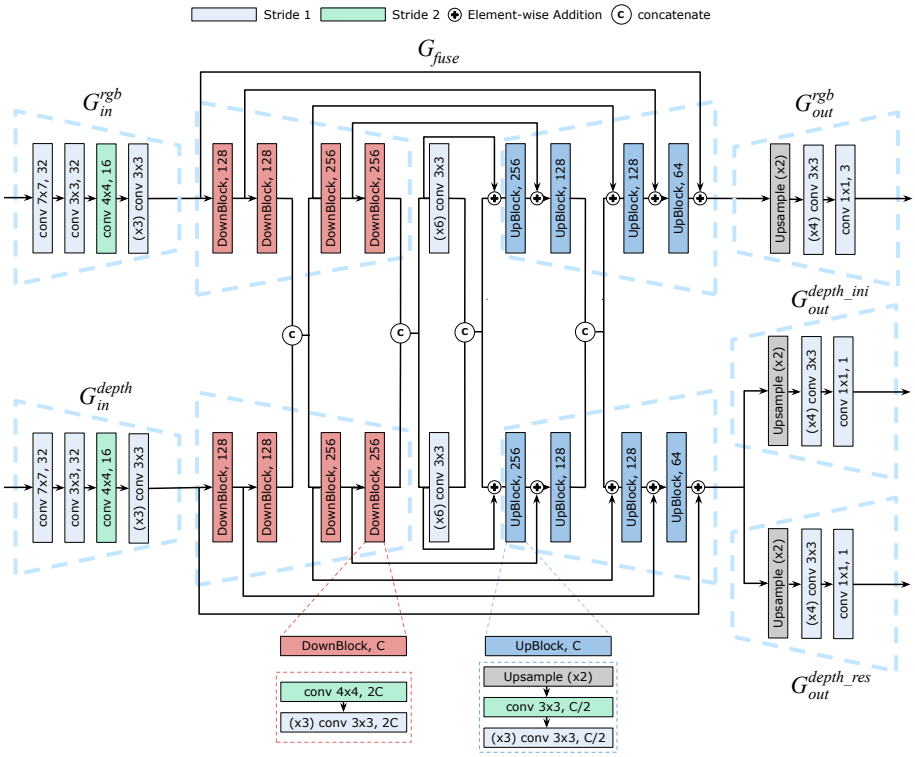

Fig. 1. The proposed generator network architecture.

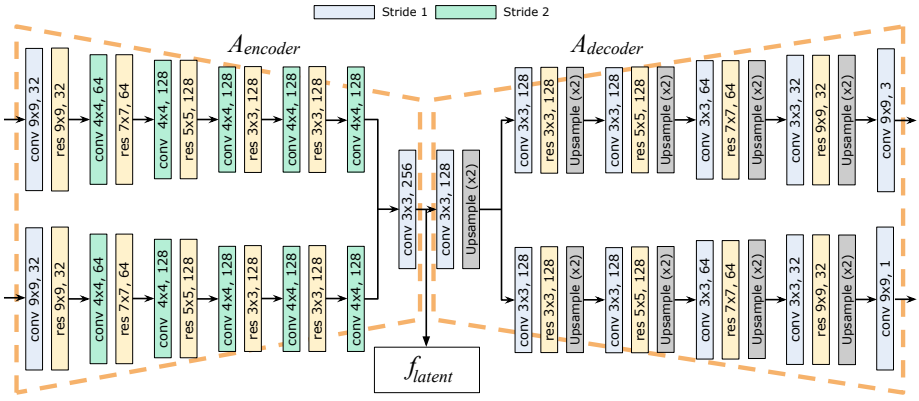

Fig. 2. The proposed auto-encoder network architecture.

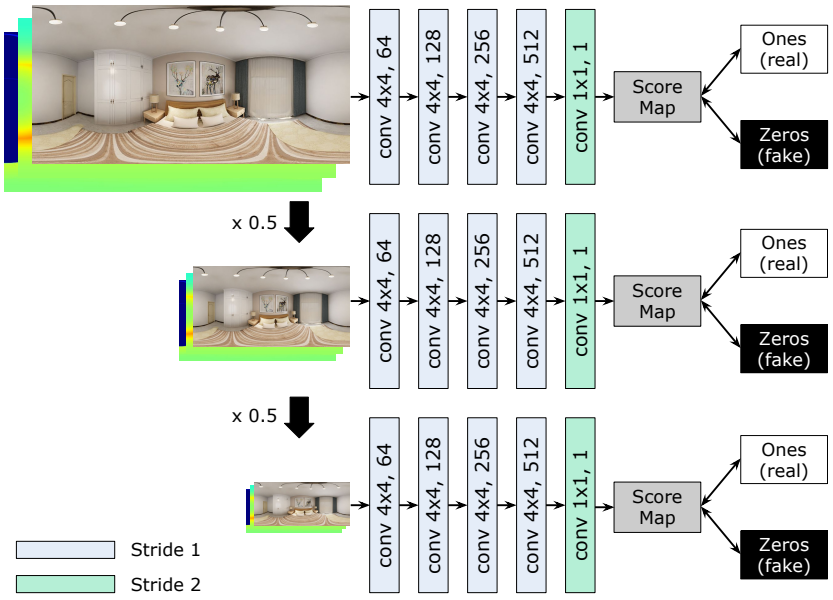

Fig. 3. The proposed discriminator network architecture.

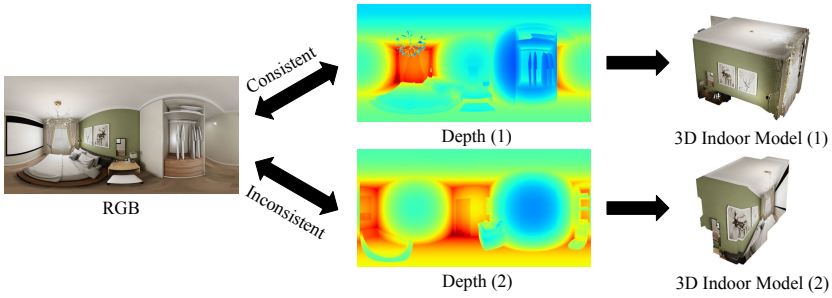

Fig. 4. Visual results of the semantic inconsistency. Depth (1) is well-aligned with RGB. Depth (2) is misaligned with RGB.

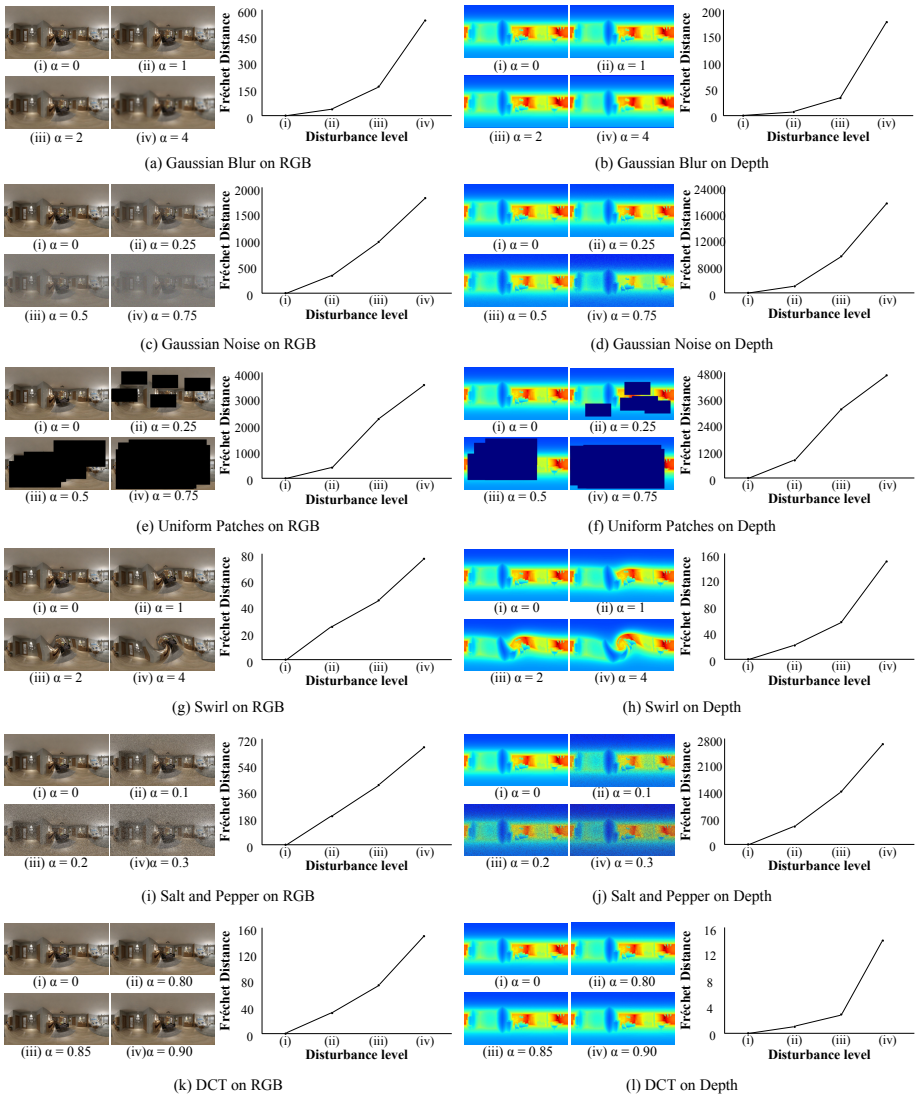

Fig. 5. Verification of FAED on Structured3D dataset.

Fig. 6. More qualitative results for RGB panorama synthesis on Structured3D dataset.

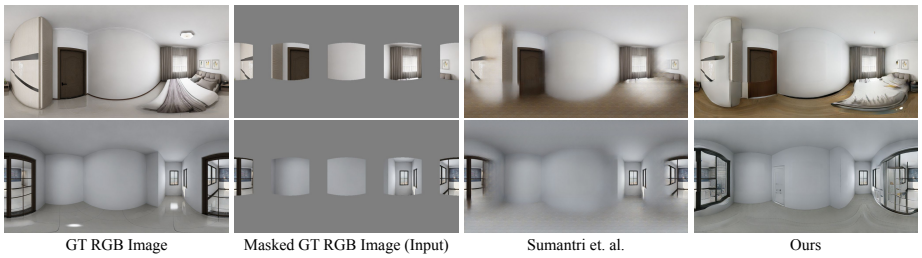

Fig. 7. More qualitative comparison to [3]. Since [3] uses 4 identical perspective RGB masks on horizontal central line, we report our qualitative results in same setting.

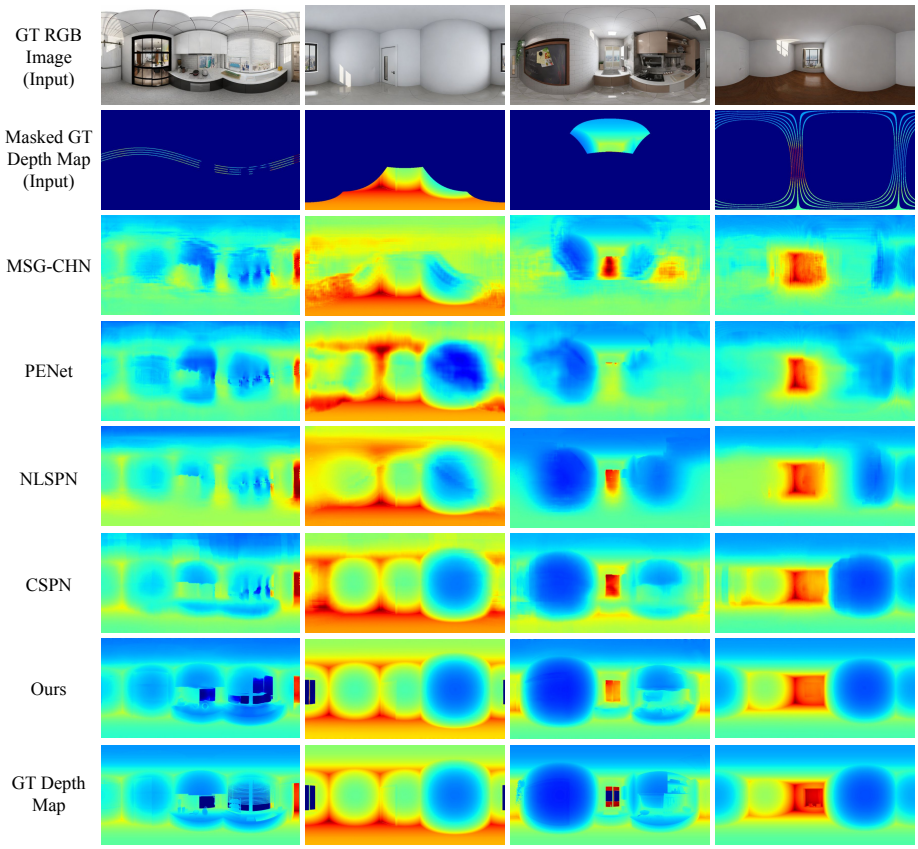

Fig. 8. More qualitative results for image-guided depth panorama synthesis on Structured3D dataset.

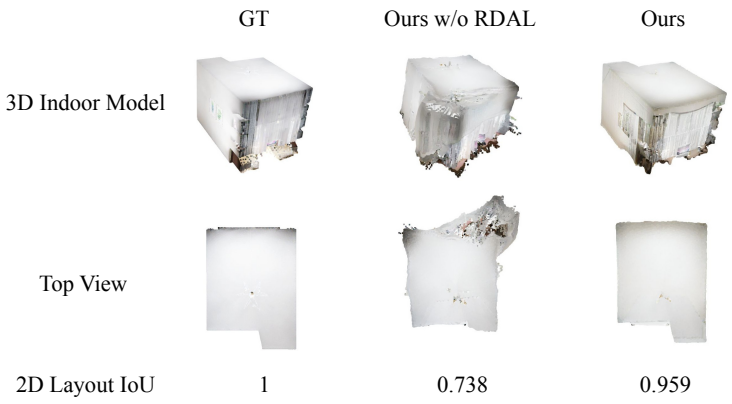

Fig. 9. Visual comparison with 2D layout IoU.

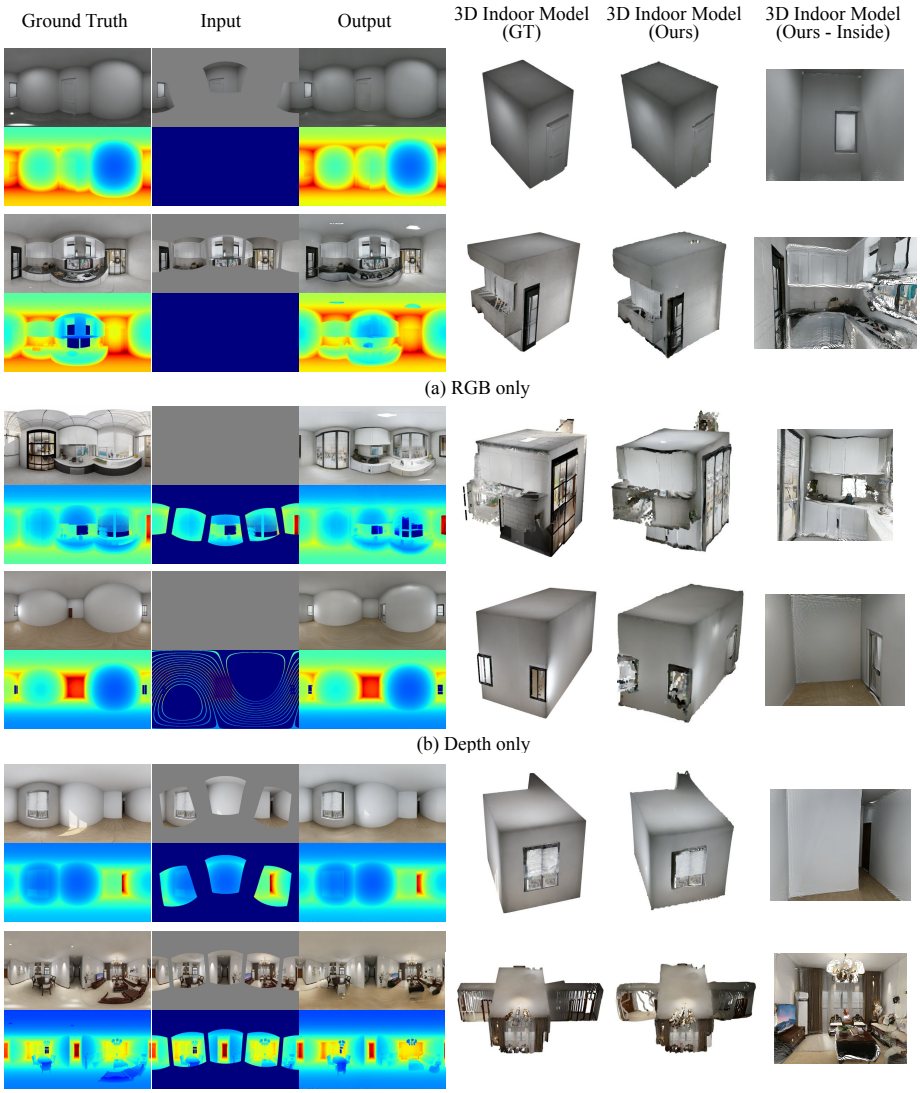

Fig. 10. More qualitative results of synthesized RGB-D panorama and its 3D indoor model using our proposed method on Structured3D dataset. (a) Top 2 rows use only masked RGB image as input. (b) Middle 2 rows use only masked depth map as input. (c) Last 2 rows use NFoV RGB images and NFoV depth maps.

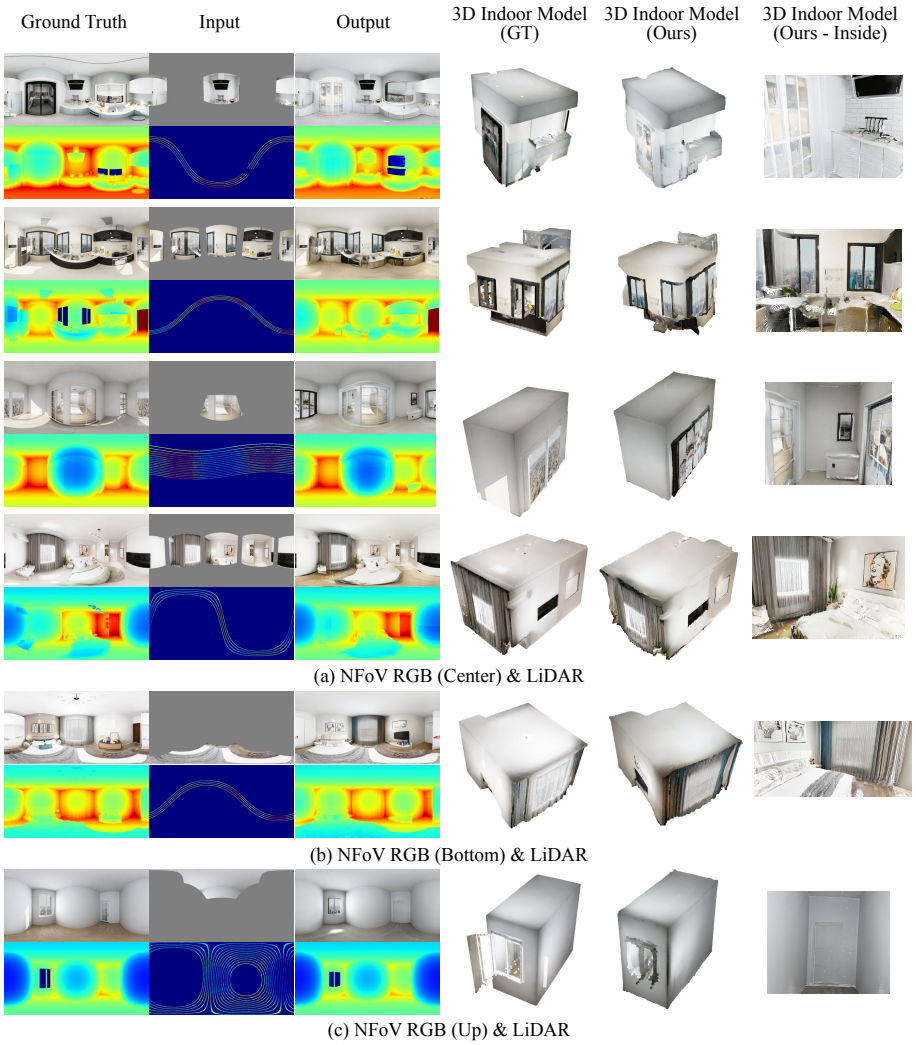

Fig. 11. More qualitative results of synthesized RGB-D panorama and its 3D indoor model using our proposed method on Structured3D dataset. (a) Top 4 rows use NFoV RGB image (center) and LiDAR as input. (b) Fifth row uses NFoV RGB image (bottom) and LiDAR as input. (c) Sixth row uses NFoV RGB image (up) and LiDAR as input.

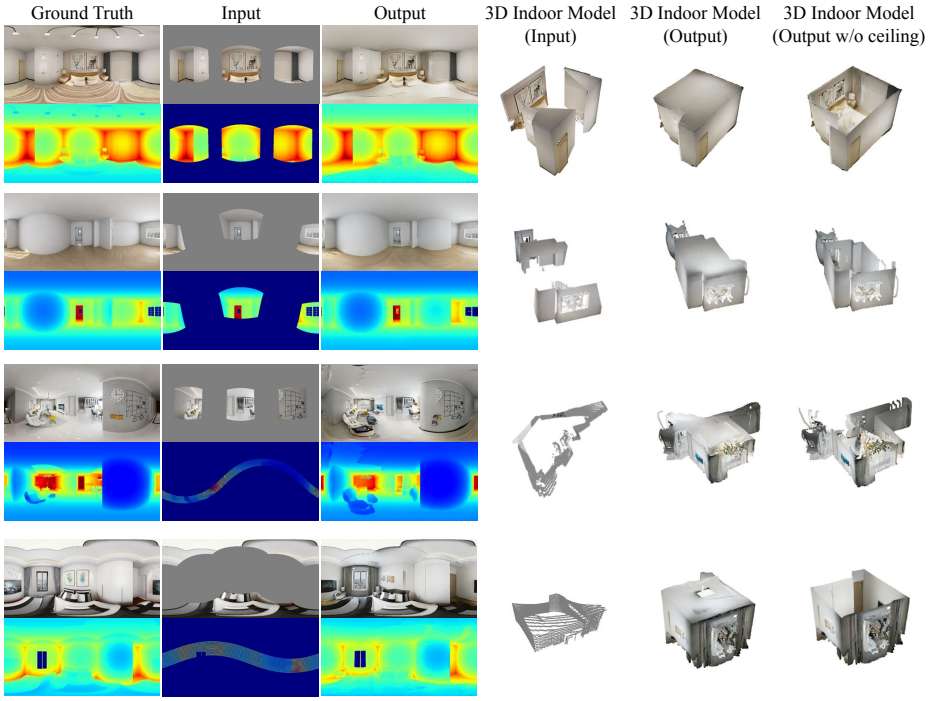

Fig. 12. More qualitative results of synthesized RGB-D panorama and input/output/output without ceiling 3D indoor model using our proposed method. Best viewed in color.

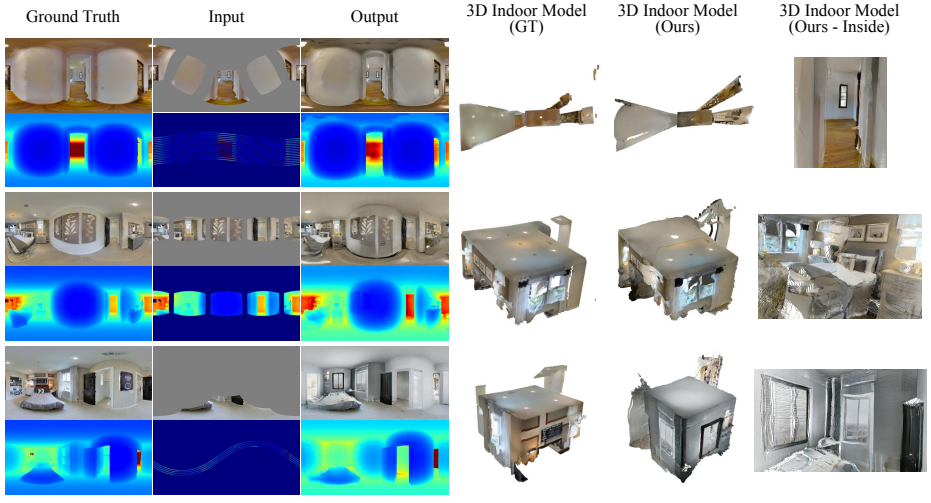

Fig. 13. More qualitative results of synthesized RGB-D panorama and its 3D indoor model using our method on Real dataset.

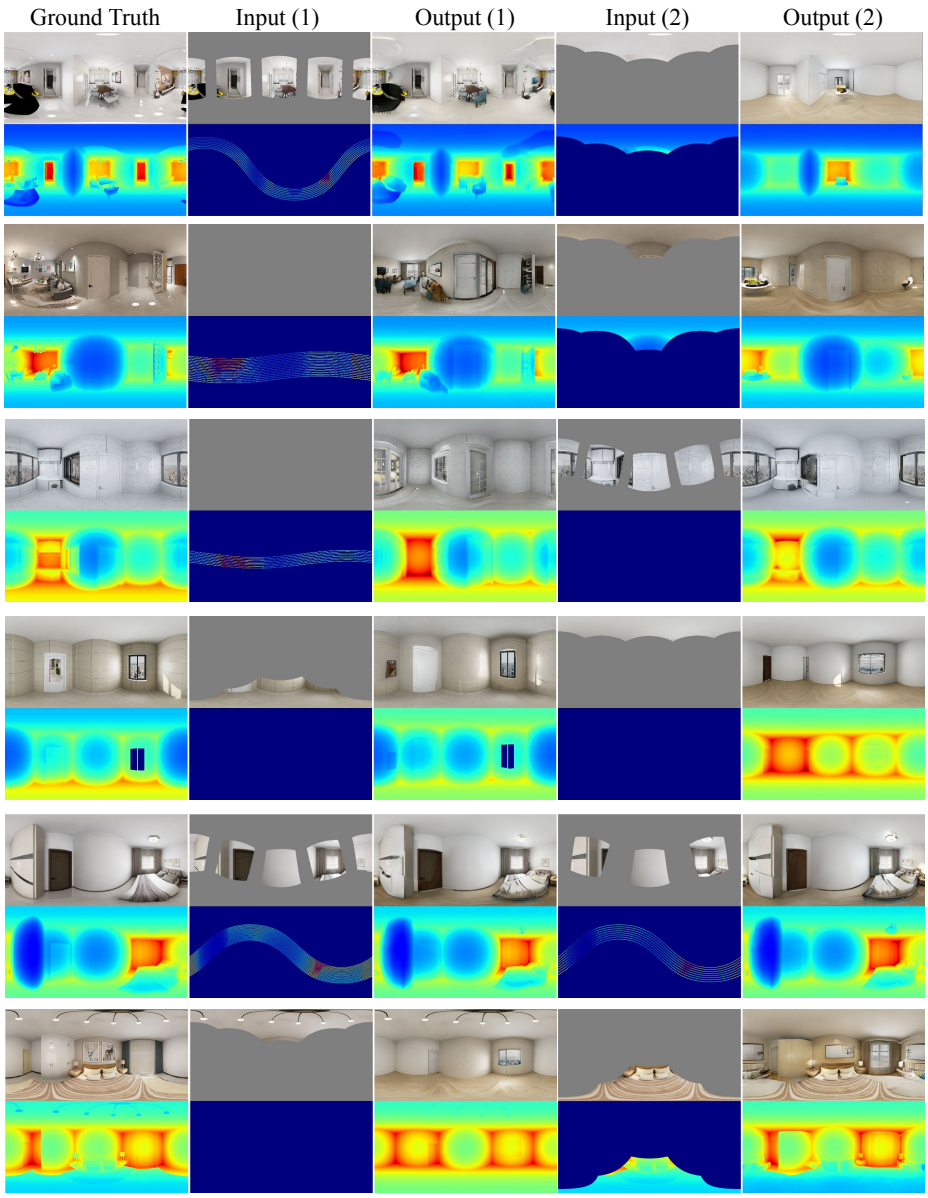

Fig. 14. Results showing the effect of different sensor configurations for the same scene. For each row, from left to right: ground truth, first input, first output, second input, and second output. Best viewed in color.

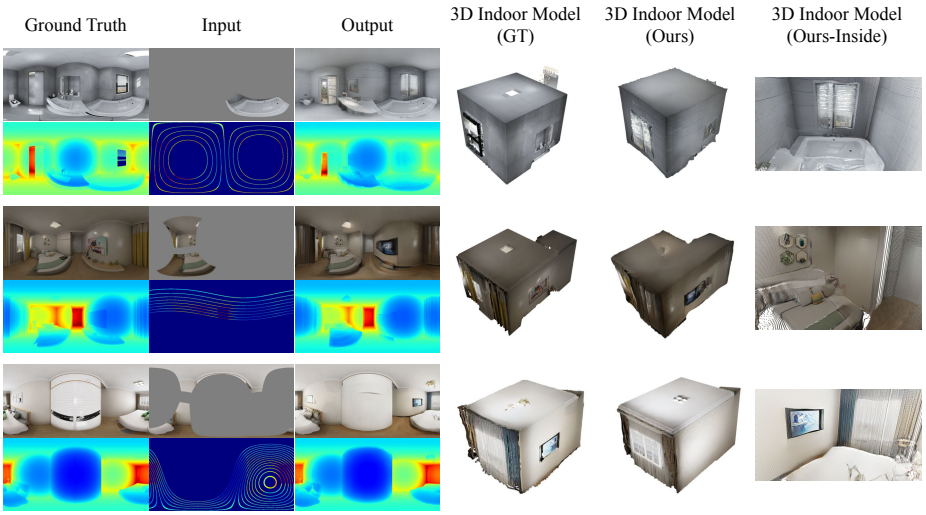

Fig. 15. More qualitative results of synthesized RGB-D panorama and its 3D indoor model under unseen input configuration.

References

1. Choi, D.: 3d room layout estimation beyond the manhattan world assumption. arXiv preprint arXiv:2009.02857 (2020)
2. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
3. Sumantri, J.S., Park, I.K.: 360 panorama synthesis from a sparse set of images on a low-power device. IEEE Transactions on Computational Imaging **6**, 1179–1193 (2020). https://doi.org/10.1109/TCI.2020.3011854