BiFuse: Monocular 360° Depth Estimation via Bi-Projection Fusion

Fu-En Wang*[1,3] fulton84717@gapp.nthu.edu.tw
Yu-Hsuan Yeh*[2] yuhsuan.eic08g@nctu.edu.tw
Min Sun[1,5] sunmin@ee.nthu.edu.tw
Wei-Chen Chiu[2] walon@cs.nctu.edu.tw
Yi-Hsuan Tsai[4] wasidennis@gmail.com

This supplementary material contains 1) detailed illustration of the idea described in Sec. 3.1. (Preliminary) of our main manuscript, 2) more results and analysis of our fusion scheme, 3) more qualitative results in terms of the point cloud, and 4) a video which summarizes our proposed method.

1. Preliminary

In Fig. 1 we provide the detailed illustration of the equirectangular-to-cube and cube-to-equirectangular transformations (denoted as **E2C** and **C2E** respectively) that we mentioned in **Sec. 3.1** of our main manuscript. To be specific, when doing E2C, we need to sample pixels on equirectangular coordinate in order to acquire the corresponding texture for each face of the cube representation. However, if we directly project from equirectangular coordinates onto the cube, we could have some faces with incomplete pixels as these pixels cannot be mapped from the integer equirectangular coordinates. As a result, the technique of **inverse mapping** is usually adopted to solve this problem: for each coordinate on the cube, we compute its corresponding coordinate in the equirectangular system and copy the pixel value. Please note here that, if the corresponding coordinate in the equirectangular system is with floating numbers, the interpolation is applied on its neighboring integer coordinates to retrieve the interpolated pixel value.

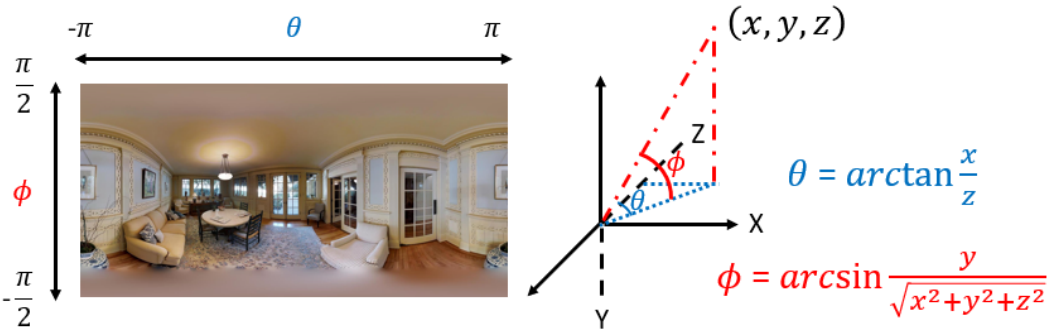

Figure 1. Illustration for the transformation presented in Equation (1) of our main manuscript. As shown on the right-hand side, given a 3-dimensional coordinate (x, y, z) on the face of cubemap, it can be transformed into the corresponding coordinate (θ, ϕ) in terms of equirectangular representation.

2. Fusion Schemes

Illustration of our fusion module. The overview of our fusion block is provided in Figure 2. First, the inputs are the feature maps from equirectangular and cubemap branches, which are fed into their corresponding convolution layers respectively. After concatenating their outputs after convolutions (denoted as $h'_e = H_e(h_e)$ and $h'_c = H_c(C2E(h_c))$ respectively), another convolution layer is used to infer a weighting mask M in order to balance the fusion between two branches. Finally, we obtain the final outputs by having $\bar{h}_e = h_e + M \cdot h'_c$ and $\bar{h}_c = h_c + \text{E2C}((1-M)) \cdot \text{E2C}(h'_e)$. In this way, the infomration betweeen two branches can be well shared.

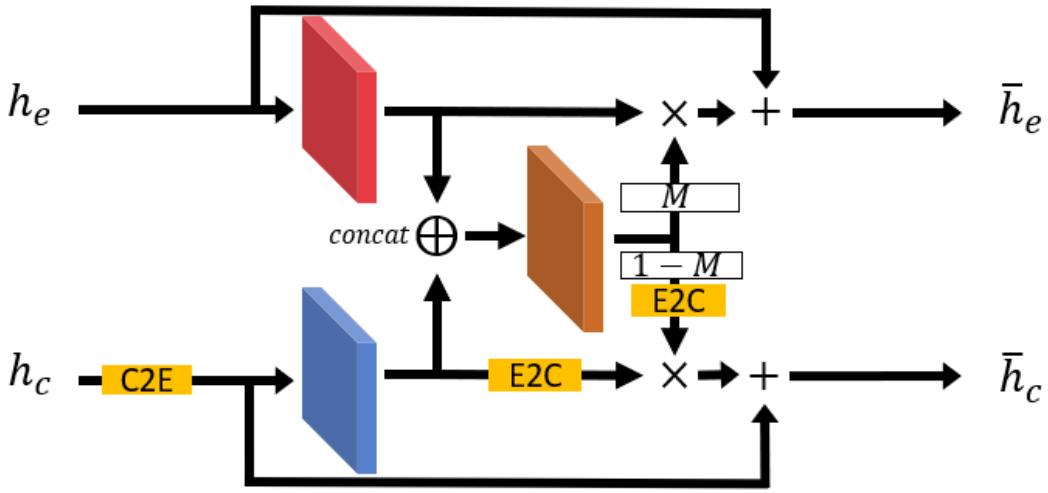

Figure 2. Illustration of our fusion block. The symbol $\times$ denotes element-wise multiplication while the symbol $+$ denotes element-wise summation.

More results and analysis. To demonstrate that our fusion scheme is beneficial to the depth estimation, we build up two model variants with different fusion strategies as the baselines to make comparison: 1) a fusion method proposed in [1] via directly adding up two feature maps, and 2) simply averaging predictions from two branches. In addition to the training/evaluation in the equirectangular coordinate (Table 6 of the main paper), we further provide comparisons in the perspective coordinate. Table 1 shows the quantitative results evaluated on the Matterport3D dataset, and our full model with the bi-directional fusion module is able to outperform other baselines with other fusion strategies, which validates the benefit of our fusion scheme to improve depth estimation in 360° cameras.

Table 1. Quantitative results on the Matterport3D dataset with comparisons to different fusion strategies. Training and evaluation is based on the cubemap coordinate system.

Different fusion	**MRE** $\downarrow$	**MAE** $\downarrow$	**RMSE** $\downarrow$	**RMSE (log)** $\downarrow$	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Yang *et al*. [1]	0.4652	0.2616	0.5490	0.1541	0.8661	0.9453	0.9665
Average	0.4798	0.2642	0.5488	0.1550	0.8630	0.9460	0.9672
Ours	**0.4512**	**0.2473**	**0.5343**	**0.1518**	**0.8792**	**0.9485**	**0.9676**

3. Qualitative Result of Point Clouds

For better visualization on the comparison between our proposed method and the baselines, here we show several qualitative results on four datasets (two cases each) in terms of point clouds, which are based on the depth estimation produced by different approaches: **Matterport3D** in Fig. 3 and Fig. 4; **PanoSUNCG** in Fig. 5 and Fig. 6; **Stanford2D3D** in Fig. 7 and Fig. 8; **360D** in Fig. 9 and Fig. 10, respectively. Each figure has two rows, the first row shows depth predictions, while the second row provides corresponding point clouds in bird eye view. The point cloud results clearly demonstrate that our prediction has sharper boundary than other methods (i.e., FCRN and Omni) and are closer to the ground truth (GT).

References

[1] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single RGB panorama. *CoRR*, abs/1811.11977, 2018. 2

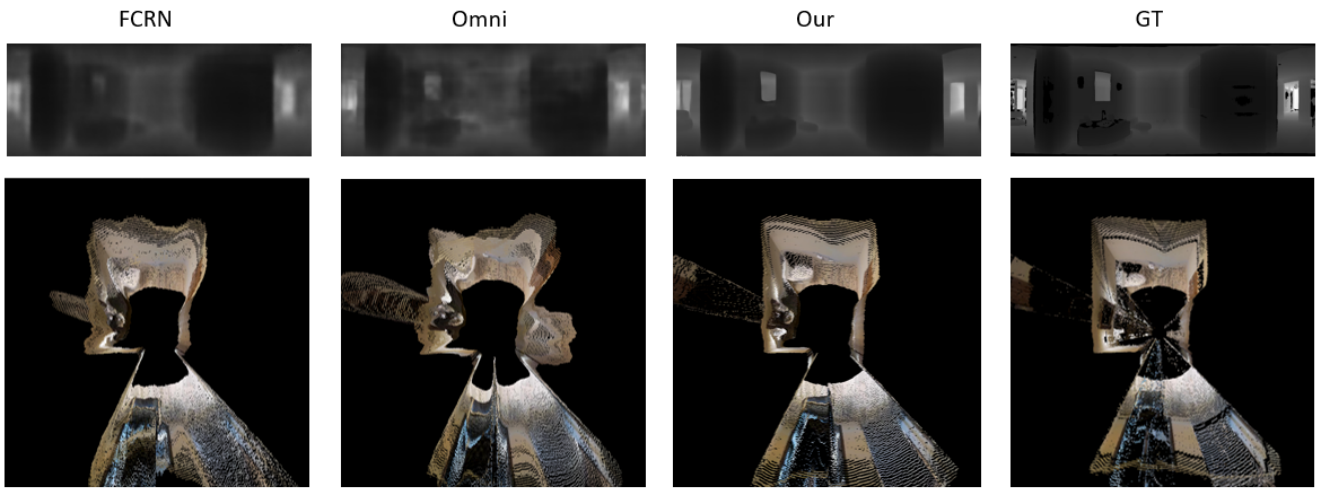

Figure 3. Matterport3D dataset

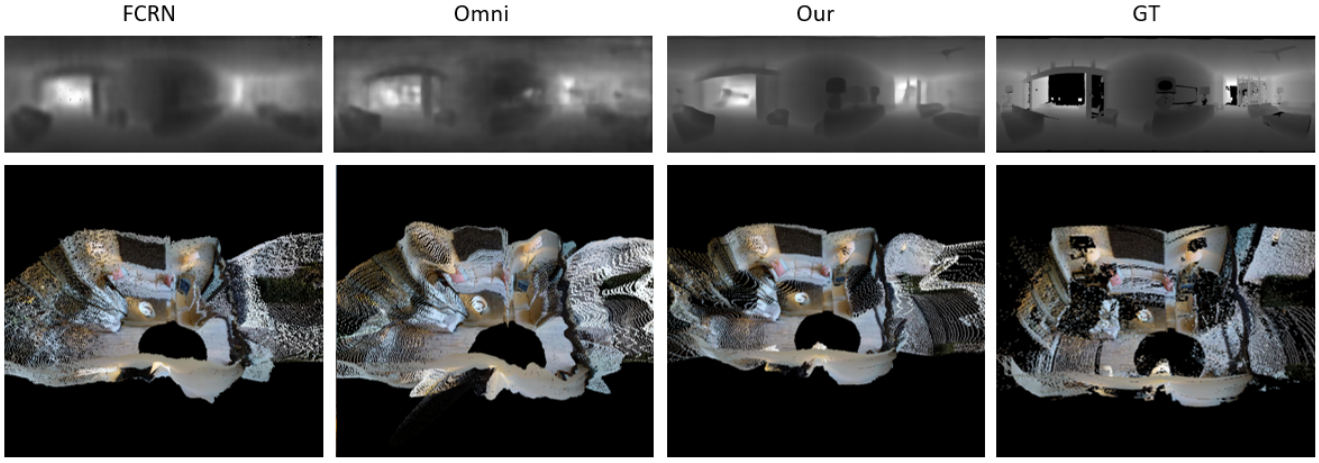

Figure 4. Matterport3D dataset

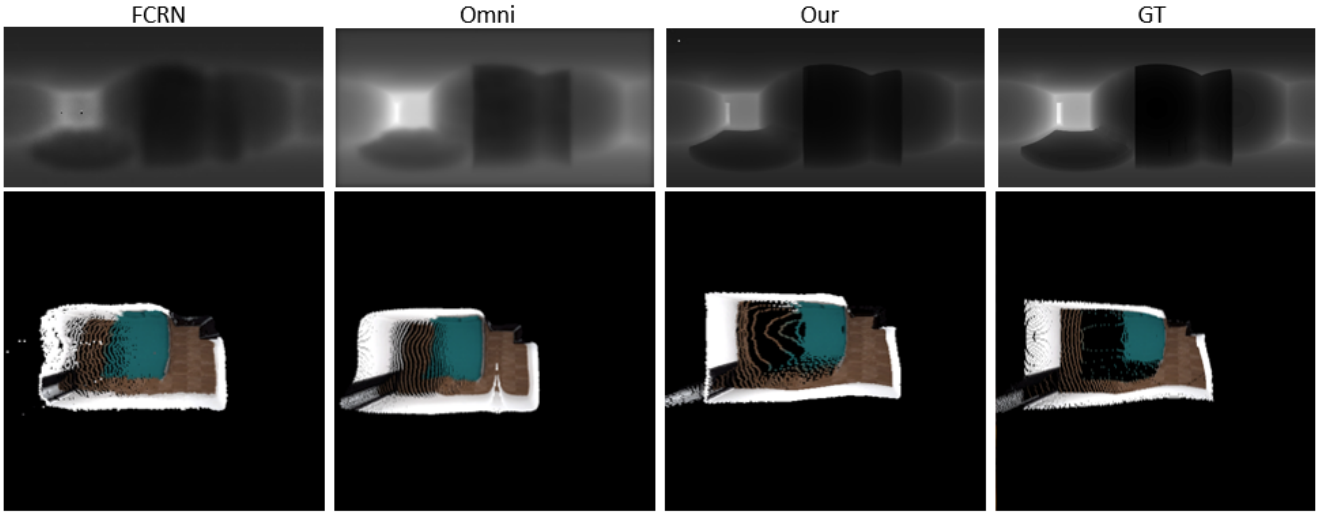

Figure 5. PanoSUNCG dataset

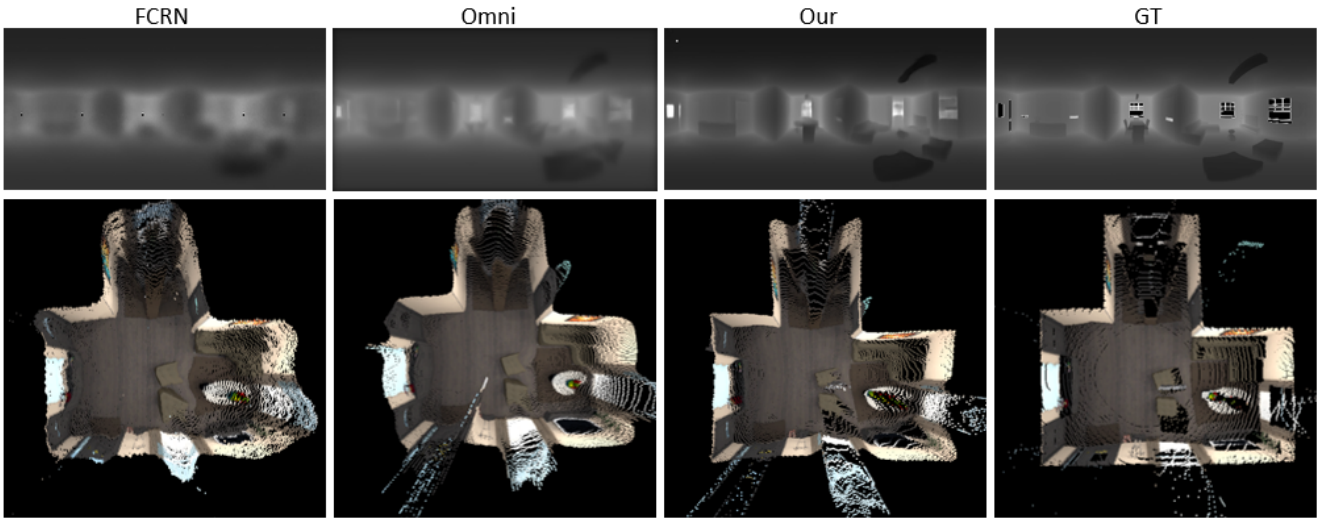

Figure 6. PanoSUNCG dataset

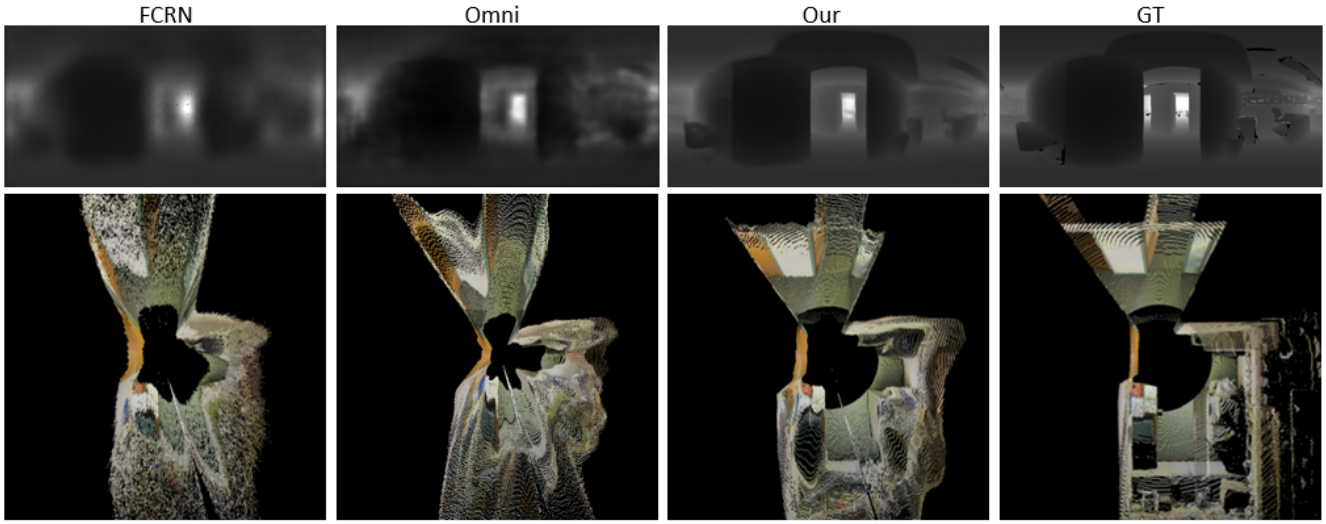

Figure 7. Stanford2D3D dataset

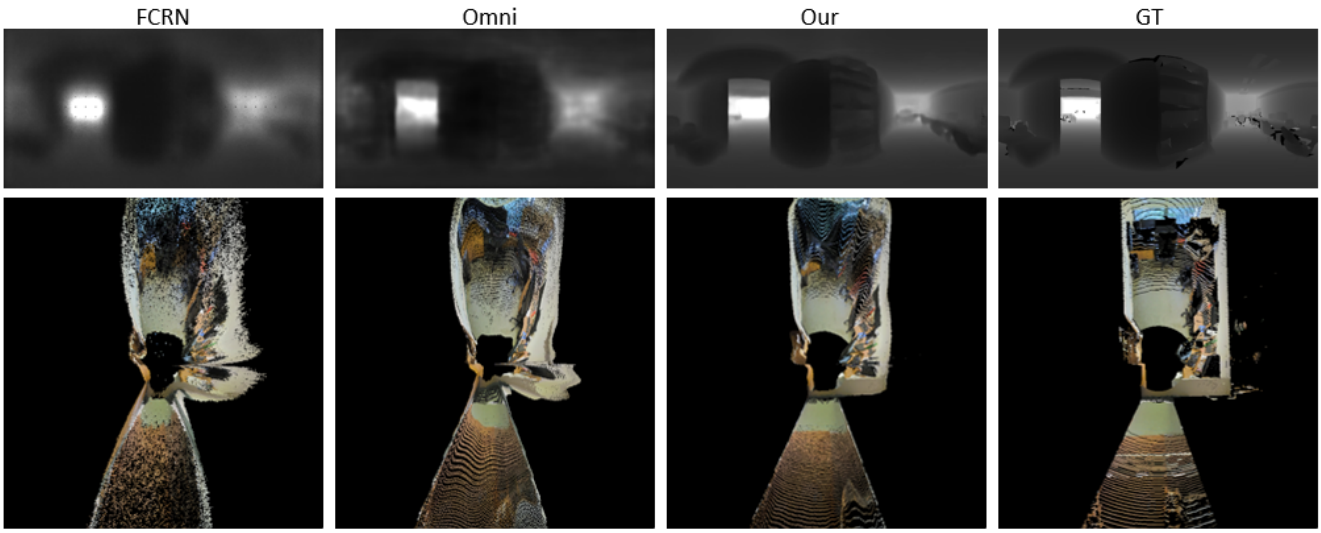

Figure 8. Stanford2D3D dataset

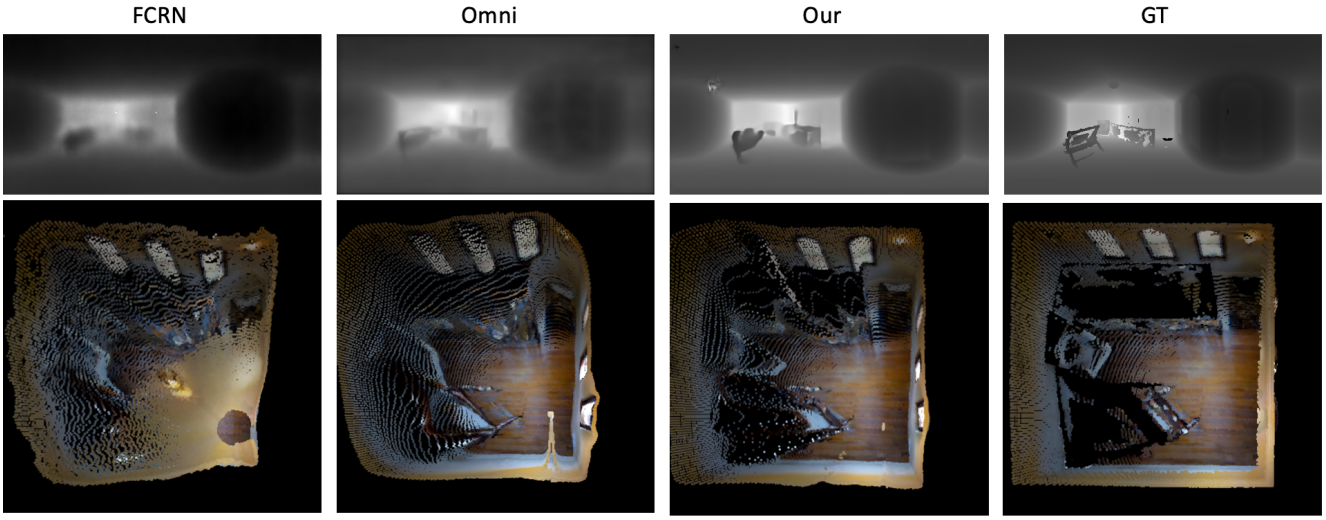

Figure 9. 360D dataset

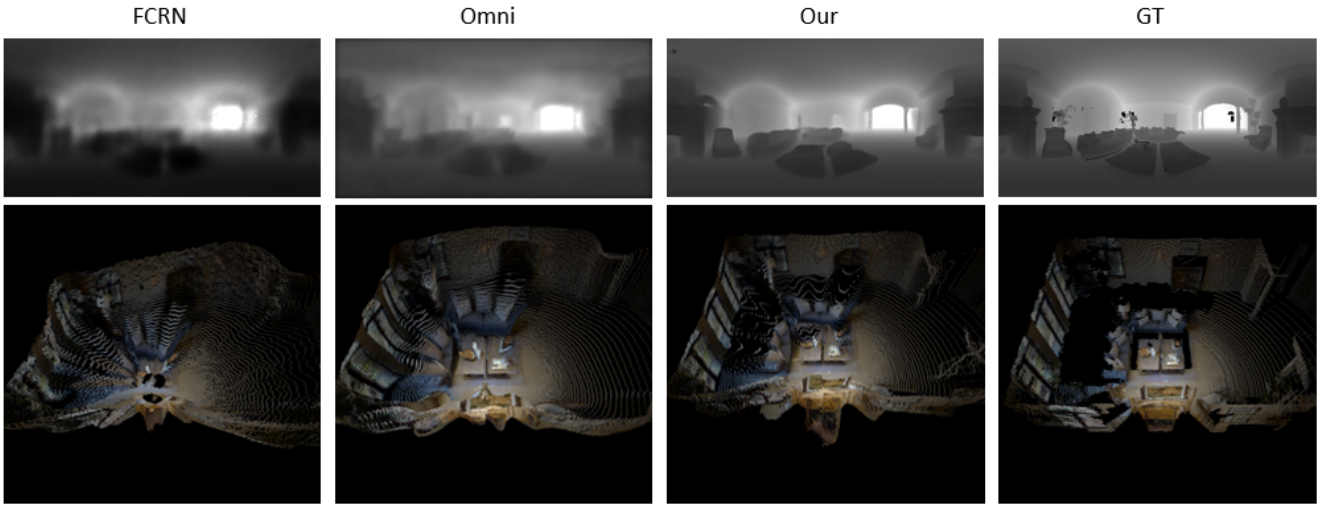

Figure 10. 360D dataset