Technical Appendix

Technical Appendix contains experimental details and additional experimental results. The code of this paper and some additional analysis will be available at https://github.com/yuniw18/EGformer.

A. Experimental environment

A.1. Discussions on dataset

For experiments of the main paper, we use Structured3D (rgb rawlight) [20] and Pano3D (Matterport3D Train and Test /w Filmic High Resolution) [1] datasets for training and evaluation. Other datasets are not used for the following reasons:

PanosunCG [17] Due to license issue, PanoSunCG dataset is no longer publicly available.

3D60 [21] Recently, it has been pointed out in that the 3D60 dataset encounters an issue with its depth representation [14, 1]. The images in this dataset exhibit a correlation between their pixel brightness and depth, as illustrated in Figure A. Consequently, depth estimation networks can predict the depths solely by analyzing the pixel brightnesses of input 3D60 images. Therefore, 3D60 dataset is unsuitable for evaluating the performances of equirectangular depth estimation networks.

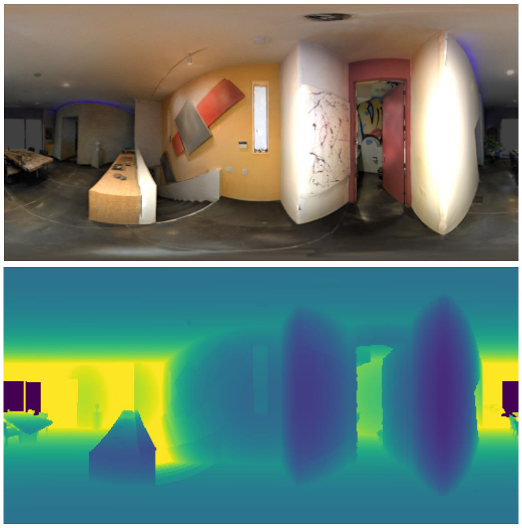


Figure A: Pixel brightness issue of 3D60 dataset

Matterport3D [3] Because Matterport3D is not natively available in equirectangular format, a stitching process is required to be employed as an equirectangular depth dataset. However, the resulting equirectangular format of Matterport3D varies depending on the stitching algorithm used. This circumstance makes it difficult to use Matterport3D as a benchmark for equirectangular depth estimation tasks. Moreover, Pano3D [1] already includes a rendered version of Matterport3D in its dataset. Hence, using both Pano3D and the Matterport3D for evaluation would be redundant.

Stanford2D3D [2] Figure B shows that the top and down parts of equirectangular images in Stanford dataset are not rendered properly. Since geometric structure is often crucial in enhancing performance of equirectangular depth estimation tasks [7], images with imperfect structure may negatively impact the performance and thus impede fair comparison.

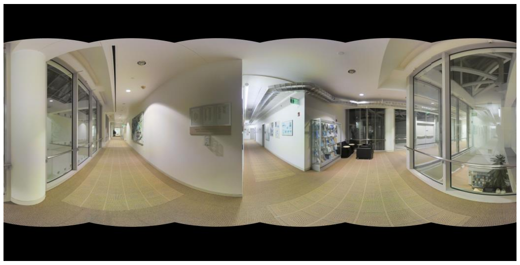


Figure B: Rendering issue of Stanford dataset

A.2. Common training details

For training, official training split of Pano3D and Structured3D dataset is used. Because the depth values in Pano3D and Structured3D datasets vary in scale, we use scale-and-shift-invariant loss [13] as defined by Eq.(3) to properly train the network with multiple depth dataset [6, 4, 16, 13, 12, 19]. Here, D indicates predicted depths of each method, D^g represents ground truth depths and D^A is aligned depths.

$$\begin{aligned} s, t &= \operatorname*{argmin}_{s,t}(s \cdot D + t - D^g) \\ D^A &= s \cdot D + t \end{aligned} \tag{1}$$

$$\begin{aligned} \mathcal{L}_{pix} &= \frac{1}{n} \cdot \sum_{k=1}^{n} |(D^A - D^g)| \\ \mathcal{L}_{grad} &= \frac{1}{n} \cdot \sum_{k=1}^{n} |\nabla_x (D^A - D^g) + \nabla_y (D^A - D^g)| \end{aligned} \tag{2}$$

$$\mathcal{L}_{total} = \mathcal{L}_{pix} + 0.5 \cdot \mathcal{L}_{grad} \tag{3}$$

Because we use scale-and-shift invariant loss function, the output of all methods is post-processed equally to be in the range between $(0, 1)$ via a sigmoid function. Each

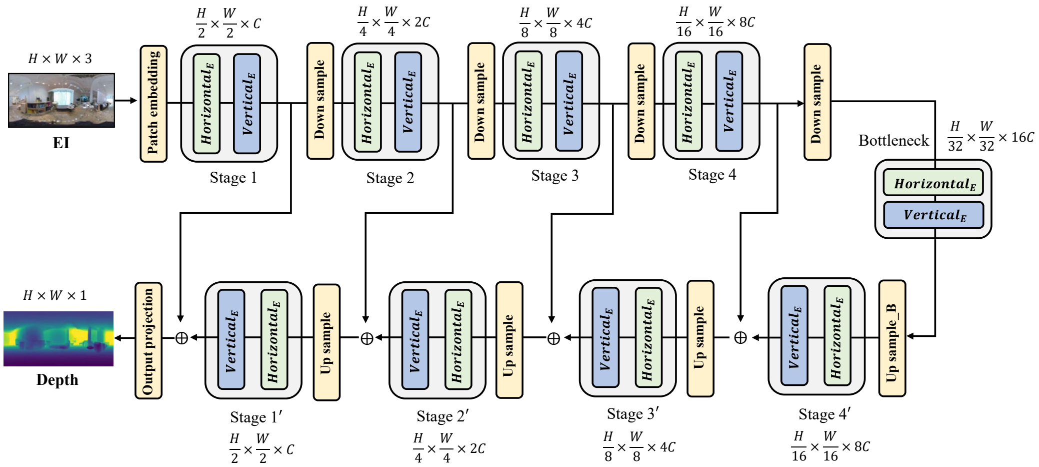

$H \times W \times 3$
EI
Patch embedding
$\frac{H}{2} \times \frac{W}{2} \times C$
$Horizontal_E$
$Vertical_E$
Stage 1
Down sample
$\frac{H}{4} \times \frac{W}{4} \times 2C$
Stage 2
$\frac{H}{8} \times \frac{W}{8} \times 4C$
Stage 3
$\frac{H}{16} \times \frac{W}{16} \times 8C$
Stage 4
Bottleneck
$\frac{H}{32} \times \frac{W}{32} \times 16C$
$H \times W \times 1$
Depth
Output projection
Up sample
Up sample_B
Stage 1′
$\frac{H}{2} \times \frac{W}{2} \times C$
Stage 2′
$\frac{H}{4} \times \frac{W}{4} \times 2C$
Stage 3′
$\frac{H}{8} \times \frac{W}{8} \times 4C$
Stage 4′
$\frac{H}{16} \times \frac{W}{16} \times 8C$

Figure C: Overall Architecture of EGformer variant. $\oplus$ indicates concatenation operation.

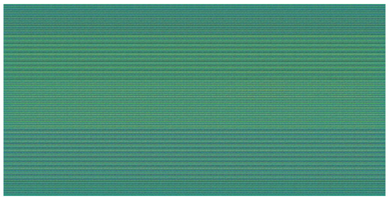


Figure D: Predicted depth of Panoformer which is trained from the scratch. Under our experimental setup, Panoformer fails to learn proper features when it is trained from the scratch.

method is trained with Structured3D dataset for 50 epochs with the learning rate of $5 \cdot 10^{-5}$. Continuously, each method is trained with Pano3D + Structured3D dataset for 20 epochs additionally with the learning rate of $5 \cdot 10^{-5}$ and exponential learning rate decay rate of 0.95. AdamW [9] optimizer is used with the batch size of 1 for all methods.

A.3. Specific training details of each method

When we train each method under our experimental setup, we failed to train some methods (*e.g.* Panoformer, Yun *et al.*) when they are trained from the scratch. Figure D shows a predicted depth of Panoformer which is trained from the scratch. We conjecture that the deconvolution layers [10] in Panoformer cause failures of training if the weight of deconvolution layer is not properly initialized. Therefore, to train all methods as fair as possible, we utilize the pre-trained model in which each author provides in the open-source community (*i.e.* github) as follows:

Panoformer [14] We initialize the weight of convolution and deconvolution layers (except the transformer) in Panoformer network using the official pre-trained model in their github repository. Subsequently, the network is trained under the environment specified in common training details (Section A.2).

EGformer We first pre-trained the model from the scratch using Structured3D dataset for 50 epochs. Then, from the scratch again, we initialize the weight of convolution layers (except the transformer) in EGformer using that pre-trained model. Subsequently, the network is trained under the environment specified in common training details (Section A.2).

Yun *et al.* [19] We initialize the weight of convolution layers in Yun *et al.* (*i.e.* decoder part) using the official pre-trained model (trained using Structured3D) in their github repository. Subsequently, the network is trained under the environment specified in common training details (Section A.2).

Bifuse [18] We initialize the weight of Bifuse network using the official pre-trained model (trained using Stanford2D3D) in their github repository. Subsequently, the network is trained under the environment specified in common training details (Section A.2)

SliceNet [11] We initialize the weight of SliceNet network using the official pre-trained model (trained using Structured3D) in their github repository. Then, the network is trained additionally for 20 epochs using Pano3D + Structured3D dataset [1].

[1] Training with Structured3D for 50 epochs is not executed here because we load the weights of 'all' layers in SliceNet using the official pre-trained model which is trained with Structured3D.

B. Network architecture

The overall network architecture of EGformer is described in Figure C, which is similar to Panoformer [14] network architecture. The channel size is set to $C = 32$ in our experimental setup. The yellow blocks are composed of several convolution, activation and normalization layers. Figure E shows examples of them. Further details on network architecture can be found in **Code Appendix**.

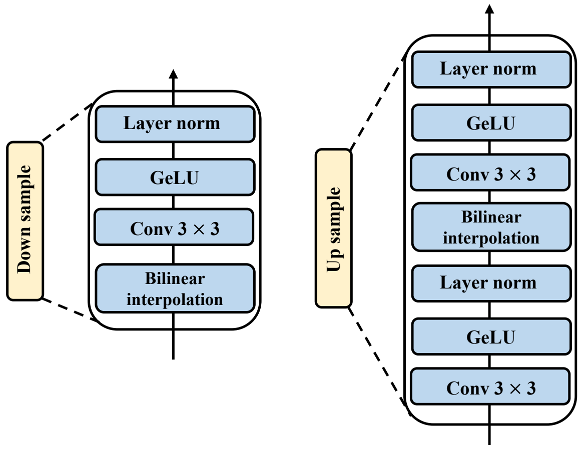


Figure E: Down sample and up sample layer

C. Additional experimental results

C.1. Further analysis on local and global attention

As discussed in various studies [15, 8], vision transformer (ViT) [5], in which Yun *et al*. [19] based on, requires large scale dataset due to lack of inductive bias. To alleviate the data insufficiency of equirectangular depth dataset, therefore, Yun *et al*. [19] performs transfer learning from the model which is trained with abundant depth dataset of typical 2D images in their paper. However, under our experimental setup, only equirectangular depth datasets are utilized. As discussed in the main paper, we speculate that this environment may impair the performances of Yun *et al*. significantly. Therefore, to see the pros and cons of global attention more clearly, we conduct an additional experiment. Table A shows the experimental results on Structured3D dataset. EGformer, Panoformer and Yun *et al*. in Table A is trained with experimental environment described in Section A.3. 'Yun *et al*. + transfer' in Table A represents the model which is trained via the following training environment: We initialize the weight of Yun *et al*. network using the official pre-trained model which is trained with depth datasets of typical 2D images + Structured3D dataset; then, we fine-tune the network via Structured3D dataset for 10 epochs with $5 \cdot 10^{-5}$ learning rate.

Method	Abs.rel	Sq.rel	RMS.lin	δ^1	#Param	FLOPs
Yun *et al.* [19]	0.0505	0.0499	0.3475	0.9700	123.7M	589.4G
Panoformer [14]	0.0394	0.0346	0.2960	0.9781	20.4M	77.7G
EGformer	0.0342	0.0279	0.2756	0.9810	16.3M	73.9G
Yun *et al.* + transfer	0.0342	0.0282	0.2590	0.9842	123.7M	589.4G

Table A: Additional analysis of local and global attention on Structured3D testset. 'Yun *et al.* + transfer' represents the network which executes transfer learning from the model trained with abundant depth datasets of typical 2D images.

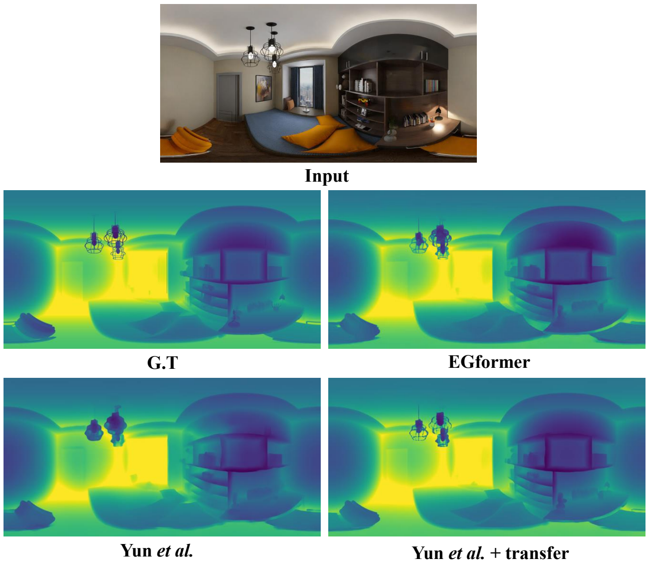
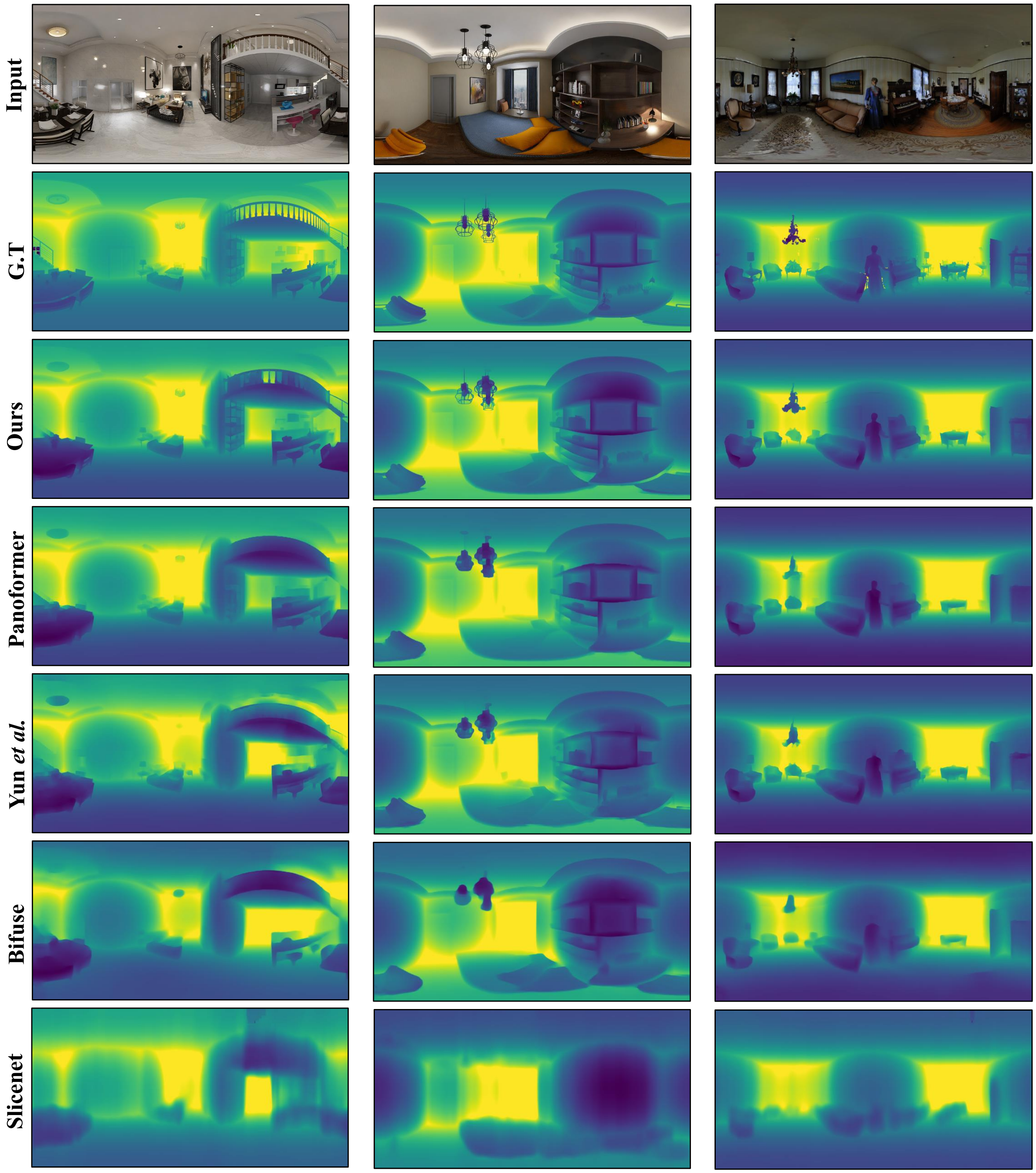


Figure F: Qualitative results of Table A. It is shown that EGformer yields comparable or better results than Yun *et al.* (+ transfer) in terms of details with significantly lower computational cost and fewer parameters.

As shown in Table A, it is observed that 'Yun *et al.* + transfer' yields the best depth estimation results overall which verifies that the number of dataset is critical for the global attention. However, it should be noted that local attention based studies (*i.e.* Panoformer, EGformer) yield comparable depth estimation results with significantly lower computational cost, fewer parameters and less datasets used as shown in Table A. These results clearly demonstrate that the local attention is much more 'efficient' than global attention in equirectangular depth estimation tasks. Figure G shows the qualitative results of the methods in Table A. It is shown that EGformer yields comparable or better results than Yun *et al.* (+ transfer) in terms of details.

C.2. Additional qualitative results

Here, we present the additional qualitative results of each method described in the main paper as shown in Figure G.



(a) Structured3D (b) Structured3D (c) Pano3D

Figure G: Additional qualitative results of each method.

References

[1] Georgios Albanis, Nikolaos Zioulis, Petros Drakoulis, Vasileios Gkitsas, Vladimiros Sterzentsenko, Federico Alvarez, Dimitrios Zarpalas, and Petros Daras. Pano3d: A holistic benchmark and a solid baseline for 360deg depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3727–

3737, 2021. 1

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1

[4] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29:730–738, 2016. 1

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27, 2014. 1

[7] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 889–898, 2020. 1

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 3

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[10] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 2

[11] Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti. Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11536–11545, 2021. 2

[12] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 1

[13] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 1

[14] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: Panorama transformer for indoor 360 depth estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 195–211. Springer, 2022. 1, 2, 3

[15] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[16] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, pages 348–357. IEEE, 2019. 1

[17] Fu-En Wang, Hou-Ning Hu, Hsien-Tzu Cheng, Juan-Ting Lin, Shang-Ta Yang, Meng-Li Shih, Hung-Kuo Chu, and Min Sun. Self-supervised learning of depth and camera motion from 360 videos. In *Asian Conference on Computer Vision*, pages 53–68. Springer, 2018. 1

[18] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020. 2

[19] Ilwi Yun, Hyuk-Jae Lee, and Chae Eun Rhee. Improving 360 monocular depth estimation via non-local dense prediction transformer and joint supervised and self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3224–3233, 2022. 1, 2, 3

[20] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020. 1

[21] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018. 1