# Supplementary Materials

Cho-Ying Wu[2], Jialiang Wang[1], Michael Hall[1], Ulrich Neumann[2] and Shuochen Su[1]

[1]Meta Reality Labs, [2]University of Southern California

{jialiangw,michaelhall,shuochsu}@fb.com, {choyingw, uneumann}@usc.edu

## A. Overview

In Sec. B, we provide more details of all datasets and include some samples. We especially focus on our proposed large-scale SimSIN and UniSIN datasets. In Sec. C, we show both numerical and visual analysis on UniSIN's test set. In Sec. D, we describe the closed-form solutions to the least-square problem used in Sec.3.2 of the paper. In Sec. E, we first create a simple oracle that uses the least-square alignment from DPT's outputs to groundtruth's metric depth and evaluate the performance of this oracle method. We then present more studies on our loss combinations. In Sec. F, we explain more on the terms of depth-relevant and depth-irrelevant low-level cues mentioned in paper Sec.3.2. In Sec. G, we present more results on VA and Hypersim. We further present downstream applications using depth maps from our DistDepth, including depth-aware AR effects and 3D photos. Last, we include a video that explains our work and include demonstrations for 3D photos, depth-aware AR effects, and real-time depth sensing.

## B. Datasets

We show a collection of data samples in Fig.S2 for all datasets we include.

**VA and Hypersim:** We render a delicately constructed 3D environment [2] using Unreal Engine 4 (UE4) [1], including 7K left-right paired images. This environment contains challenging indoor scenes, such as cabinet cubes with different lighting conditions, thin structures, and complex decorators. We also include several pre-rendered scenes in the Hypersim dataset [15] for qualitative evaluation.

**NYUv2:** This public dataset [19] includes various indoor scenes captured by Kinect that produces monocular RGB images and depth from the time-of-flight laser system. Although NYUv2 is popular for single-image depth estimation, the images are low-resolution and contain noise due to its older camera model, making it hard to claim practicability in recent AR/VR creation needs. Thus, we also collect our large-scale real dataset for training and evaluation.

**SimSIN:** This dataset includes 515K left-right paired images using Habitat simulator [21] with 3D environments of Replica [20], MP3D [4], and HM3D [13]. These 3D environments are created by processing real room scans. We use our large-scale SimSIN as the training dataset. One can find that the rendered images from Replica have better quality in both appearance and geometry than MP3D and HM3D, but Replica contains only 18 scenes with lower scene variation. In contrast, MP3D and HM3D contain 90 and 900 various scenes. Thus we aggregate these 3D environments to attain higher dataset diversity and also maintain appearance and geometry quality.

**UniSIN:** The collected UniSIN, including real university scenes using recent high-performing ZED stereo cameras, which contains better imaging quality than NYUv2. Its training split contains 500 sequences with about 200K left-right images, and the testing data includes 1K images for numerical evaluation. The scenes include private or public spaces, and we organize the number of sequences for each type of space in Table S1.

Both our SimSIN and UniSIN are large-scale datasets of indoor scenes with stereo pairs to fulfill training with left-right consistency. We also enumerate other existing stereo datasets (that are not used in this work) as follows.

(1) Large-scale datasets that focus on driving scenarios: KITTI [9], vKITTI [8], Cityscapes [7], and Argoverse [5].

(2) Indoor datasets but small-scale: Middlebury 2021 (48 images), Middlebury 2014 (66 images) [16], or some scenes in ETH3D two-view Stereo (32 images) [17].

(3) Datasets scraped from web or in-the-wild: WSVD [22] and Holopix50K [10].

(4) Datasets of 3D movies with non-realistic scales: SceneFlow [12] and Sintel [3].

Although category (2) also targets at indoor scenes, they focus on stereo matching algorithms, where smaller numbers of images may be efficient for training [11]. By contrast, our DistDepth is a monocular depth estimation method whose DepthNet only takes a single image and predicts its depth map. Furthermore, self-supervised learning

Table S1. **Number of sequences for different types of space.**

| Private room | Office | Hallway | Lounge | Meeting room |
|---|---|---|---|---|
| 44 | 94 | 118 | 93 | 13 |
| Large room | Classroom | Library | Kitchen | Playroom |
| 29 | 47 | 24 | 24 | 14 |

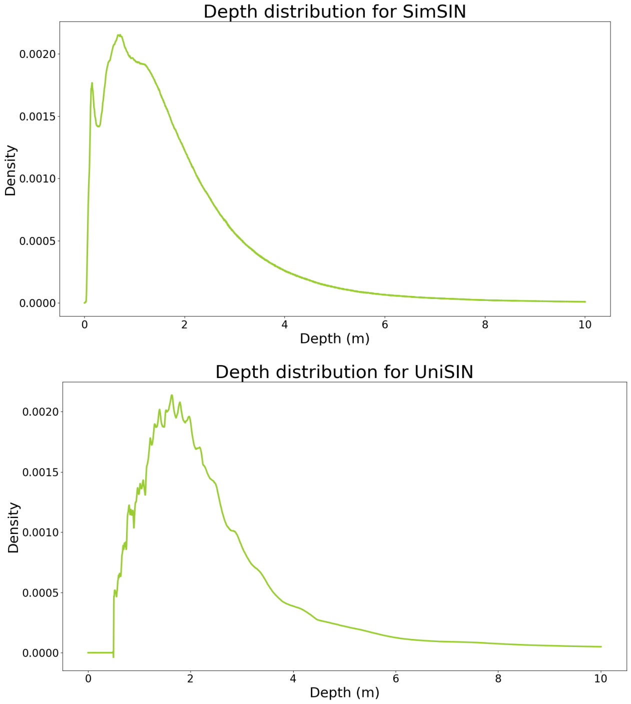


Figure S1. **Depth distributions for SimSIN and UniSIN.**

Table S2. **Numerical analysis on the UniSIN test set.** Our DistDepth trained on simulation data can reach similar performances of that trained on real data. Lower errors can be attained compared with MonoDepth2.

| Error | MonoDepth2 (Sim) | MonoDepth2 (Real) | DistDepth (Sim) | DistDepth (Real) |
|---|---|---|---|---|
| MAE | 0.610 | 0.571 | 0.518 | **0.505** |
| AbsRel | 0.175 | 0.163 | 0.135 | **0.130** |
| RMSE | 0.742 | 0.688 | 0.623 | **0.611** |
| RMSElog | 0.232 | 0.200 | **0.159** | 0.162 |

Table S3. **Performances of converting relative depth into metric depth using linear regression.** We convert DPT's outputs to metric depth by linear relations from optimizing least-square errors with RANSAC. See the text in Sec. E.

| Error | Linear Regression Oracle | MonoDepth2 | DistDepth |
|---|---|---|---|
| MAE | 0.390 | 0.295 | 0.253 |
| AbsRel | 0.359 | 0.203 | 0.175 |
| RMSE | 0.645 | 0.432 | 0.374 |
| RMSElog | 0.283 | 0.251 | 0.213 |

inherently requires larger training datasets since there are no direct image-to-depth mappings accessible at training time. Therefore, we collect large-scale datasets, SimSIN and UniSIN, to achieve our aims of self-supervised monocular depth estimation with left-right consistency and verify our advantages of closing sim-to-real gaps.

We plot the depth distributions for our newly created datasets, SimSIN and UniSIN, in Fig. S1.

## C. Numerical Analysis on UniSIN

We also exhibit numerical analysis on the UniSIN test set that contains 1K images from non-overlapping scenes with the training set in Table.S2.

## D. Least-Square Alignment

In Sec.3.2 of the paper, we adopt a least-square alignment by minimizing the difference of $a_s D_t^* + a_t$ and $D_t$, where $a_s$ is the scale term, and $a_t$ is the shift term. For simplicity, here we drop time-step notations $t$ in depth $D$ and use $D_i^*$ for indexing depth values at the $i$-th pixel. Using the above notations, we can write the least-square problem as

$$\min_{a_s, a_t} \sum_i \|a_s D_i^* + a_t - D_i\|^2, \tag{S1}$$

Then, we change symbols with $\overrightarrow{\mathbf{d}_i^*} = [D_i^*, 1]^\top$ and $\overrightarrow{\mathbf{a}} = [a_s, a_t]^\top$. Then Eq.S1 becomes

$$\min_{\overrightarrow{\mathbf{a}}} \sum_i \|\overrightarrow{\mathbf{d}_i^*}^\top \overrightarrow{\mathbf{a}} - D_i\|^2, \tag{S2}$$

which corresponds to the normal form of a least-square problem. Then, the optimal solution of $\overrightarrow{\mathbf{a}}$ is

$$\overrightarrow{\mathbf{a}} = \sum_i \left( \overrightarrow{\mathbf{d}_i^*} \overrightarrow{\mathbf{d}_i^*}^\top \right)^{-1} \sum_i \left( \overrightarrow{\mathbf{d}_i^*} D_i \right). \tag{S3}$$

## E. More Studies

We first present an oracle that converts outputs of DPT pretraining to metric depth using linear relations with groundtruth depth, i.e., using red lines shown in Fig. 3 of the main paper as the converter. We then calculate the depth errors of all points to the regressed linear relation.

Those linear relations are optimal in terms of minimizing the least-square errors with RANSAC [6], which discovers slope and intercept for the conversion under the linear assumption between relative and metric depth in DPT [14].

We exhibit the performances in Table S3 on the VA dataset. One can observe that this oracle performs much worse than MonoDepth2 and DistDepth. The results show that optimal linear mappings are weak in capturing uncertainty in either depth estimation models or data that cause outliers (in the scatter plots of paper Fig. 3).

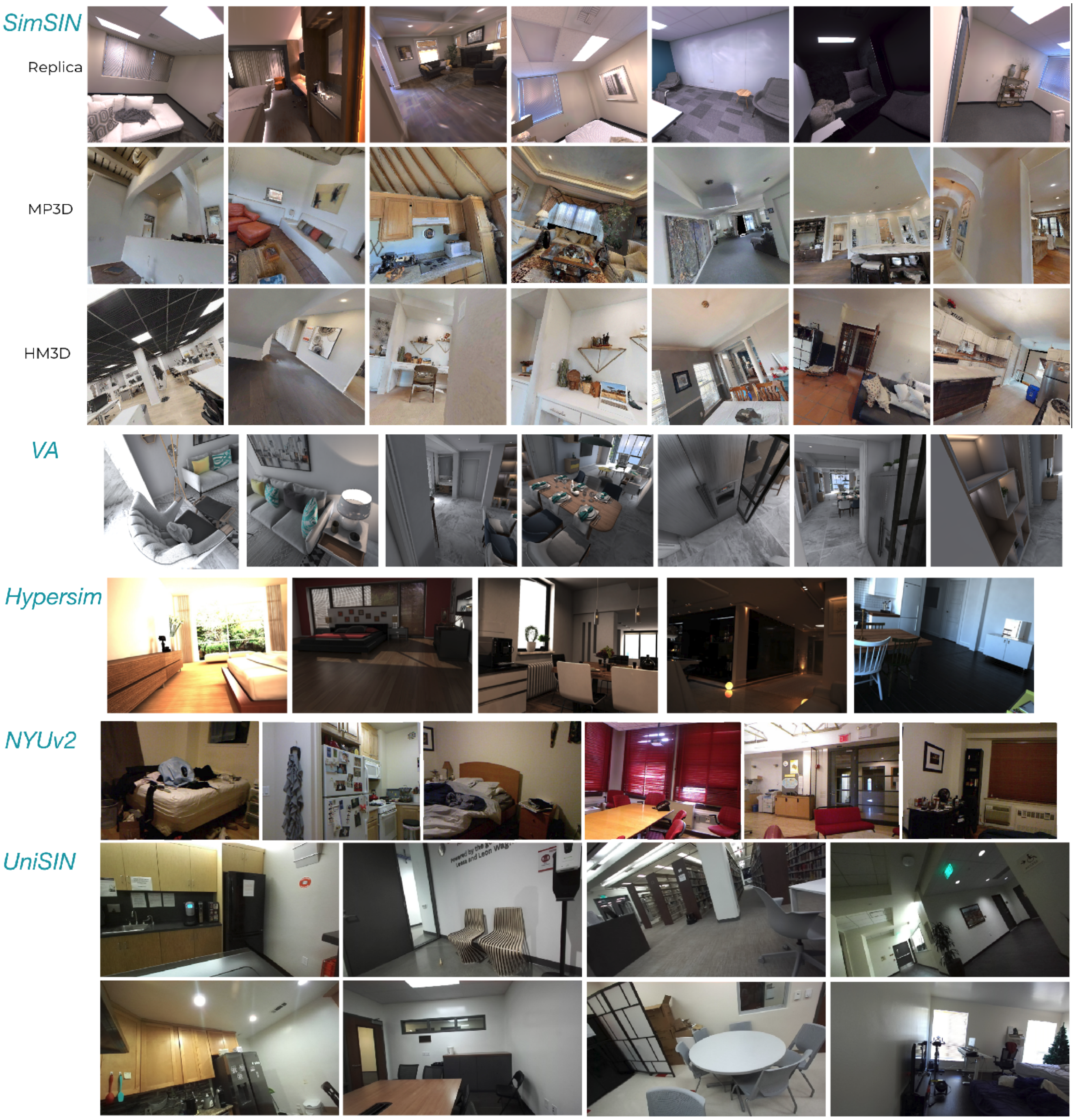


Figure S2. **Sample images of all datasets used in this work.**

We next present numerical ablation studies on the loss term combinations: (1) without distillation: using $\mathcal{L}_{LR}$ and $\mathcal{L}_{temp}$ (2) with distillation of the statistical loss only: using $\mathcal{L}_{LR}$, $\mathcal{L}_{temp}$, and $\mathcal{L}_{stat}$, and (3) full loss terms: using $\mathcal{L}_{LR}$, $\mathcal{L}_{temp}$, $\mathcal{L}_{stat}$, and $\mathcal{L}_{spat}$. We show the comparison on the VA dataset in Fig. S3.

We further study DepthNet's architecture using ResNet in Table S4, which shows decreasing errors with deeper networks.

Next, we also compare the inference speed in Fig. S4 of convolutional neural networks (CNN) and Dense Vision Transformer (D-ViT) in DPT.

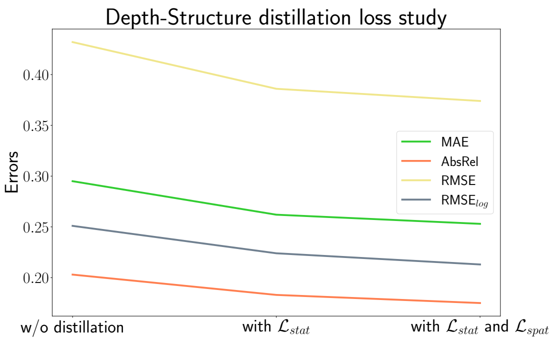


Figure S3. **Numerical studies on the loss combination.** (1) w/o distillation loss (2) with statistical distillation loss only (3) with full distillation loss.

Table S4. **Numerical studies on the DepthNet architecture choices.** We adopt ResNet of different numbers of layers as the DepthNet's architecture and find that deeper ResNet produces lower errors.

| Error | ResNet50 | ResNet101 | ResNet152 |
|---|---|---|---|
| MAE | 0.261 | 0.255 | 0.253 |
| AbsRel | 0.182 | 0.177 | 0.175 |
| RMSE | 0.383 | 0.377 | 0.374 |
| RMSElog | 0.221 | 0.217 | 0.213 |

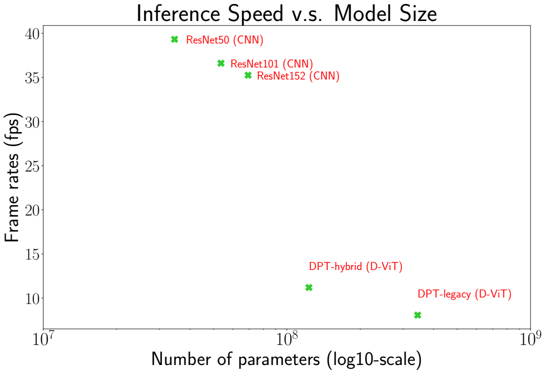


Figure S4. **Comparison on inference speed.** We test on a laptop (with Intel Core i7-10875H CPU and RTX 2080 GPU) and compare architectures for DepthNet using convolutional neural network (CNN) and Dense Vision Transformer (D-ViT) introduced in DPT.

## F. Depth-Relevant and Depth-Irrelevant Low-Level Cues

We illustrate how networks *see* depth in Fig. S5. A low-level feature is reasoned as depth-relevant or depth-irrelevant based on its local regions. A well-trained depth estimator can separate depth-relevant and depth-irrelevant features and produce depth changes only at the depth-relevant positions.

## G. More Results

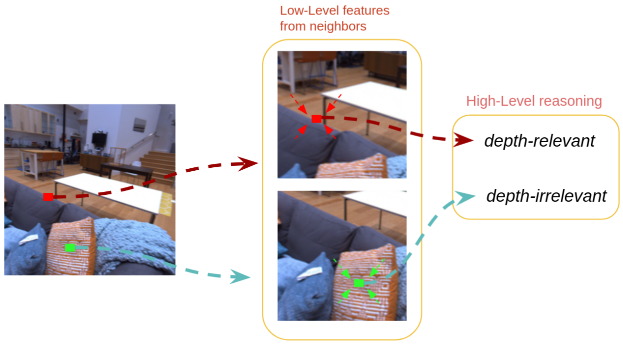


Figure S5. **Explanations on depth-relevant and depth-irrelevant low-level cues.** Low-Level cues at the red point are depth-relevant since it contains object occluding boundaries of its neighborhood. In contrast, low-level cues at the green point are printed patterns around the same depth. A good depth estimator is capable of separating depth-relevant and depth-irrelevant low-level cues.

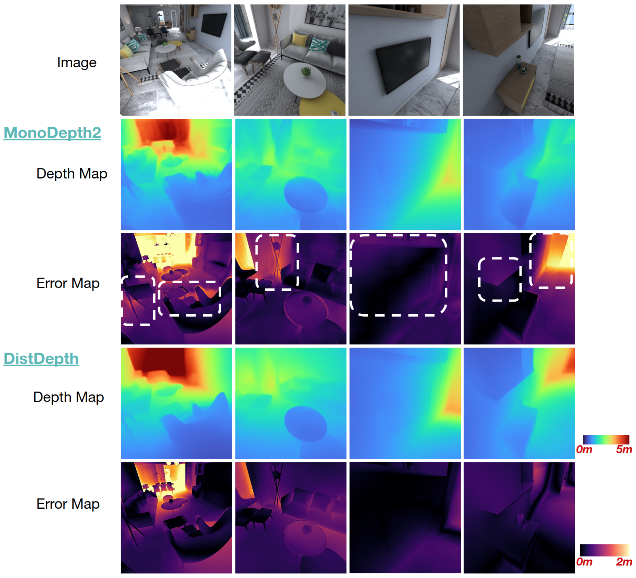


Figure S6. **More Results on the VA dataset that extends Fig. 5 of the main paper.** Depth and error maps generated by our DistDepth and MonoDepth2 are shown.

In Fig.S6-S9, we present more comparison with prior self-supervised monocular depth estimation methods, which are all trained on SimSIN. Our DistDepth predicts better depth maps, lower errors, and also better 3D point cloud than other methods. In addition, we demonstrate downstream applications on 3D photos [18] and depth-aware AR effects in Fig. S10 and S11. We also present several failure cases in Fig. S12.

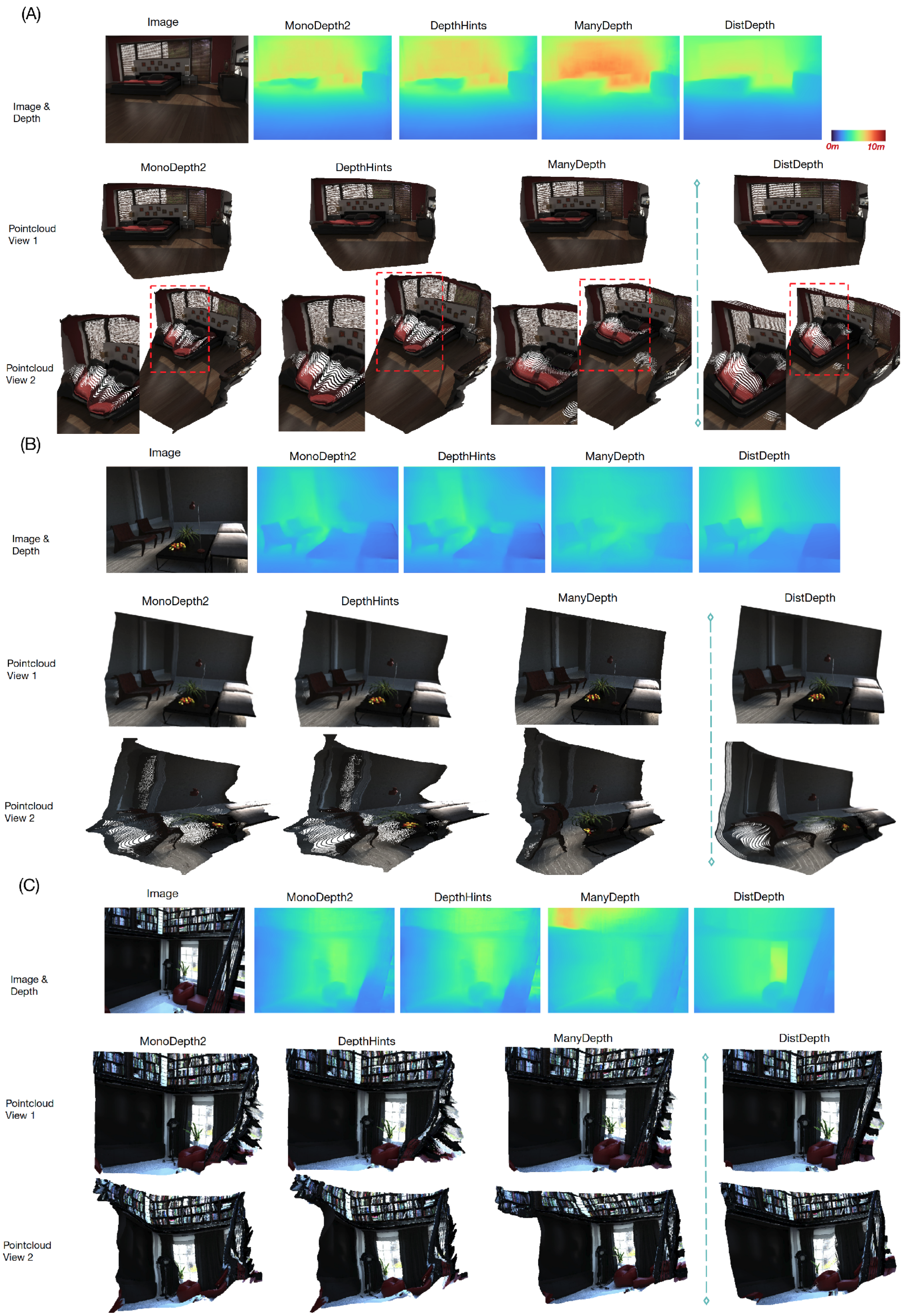


Figure S7. **Results on Hypersim.** Depth map and textured point cloud comparison. DistDepth shows less distortion on the edge of the bed in Example (A) and more structured walls and chairs in Example (B) and (C).

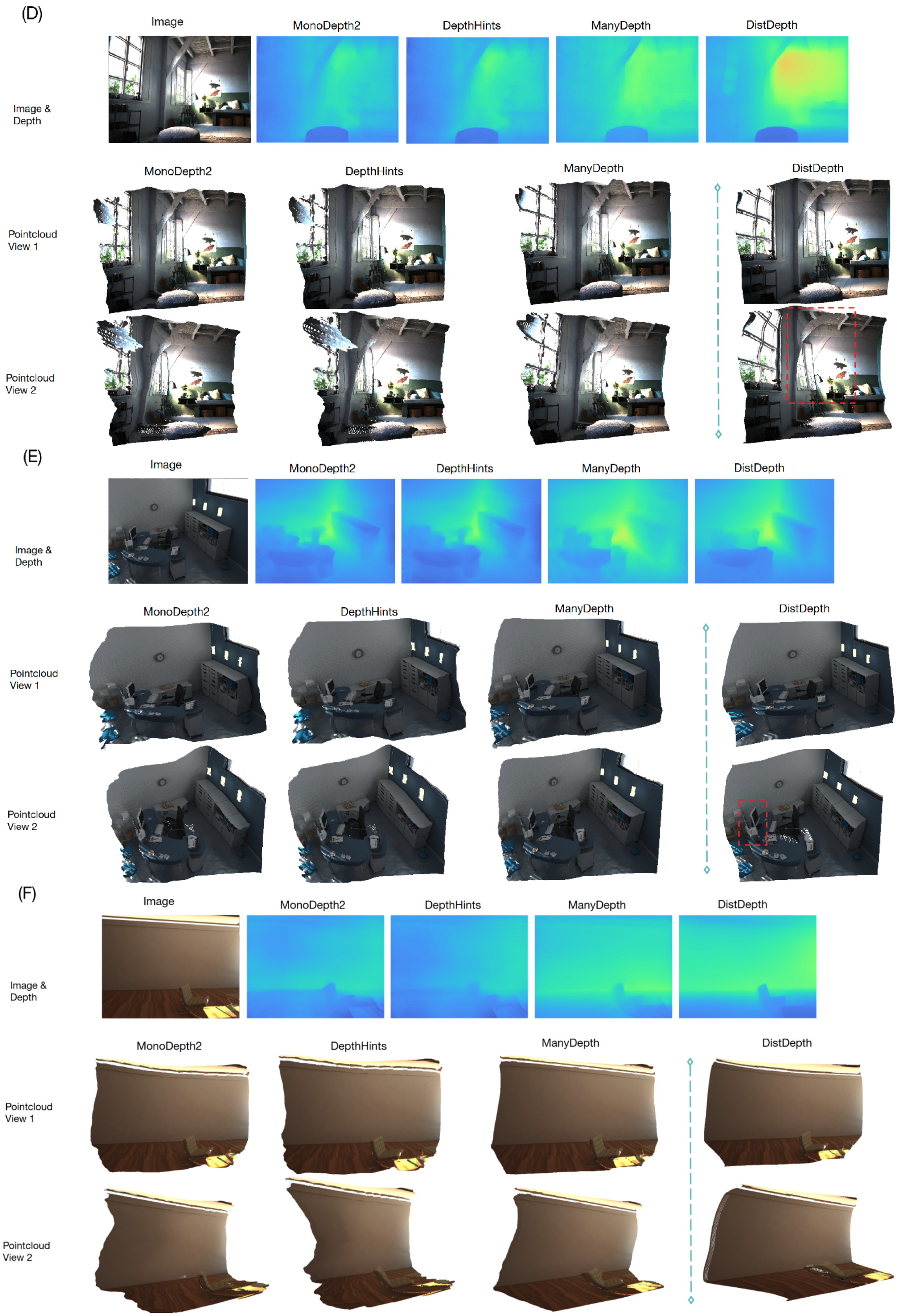


Figure S8. **(Continued) Results on Hypersim.** Our DistDepth has less distortions of highlighted areas in Example (D), (E), and the wall in (F).

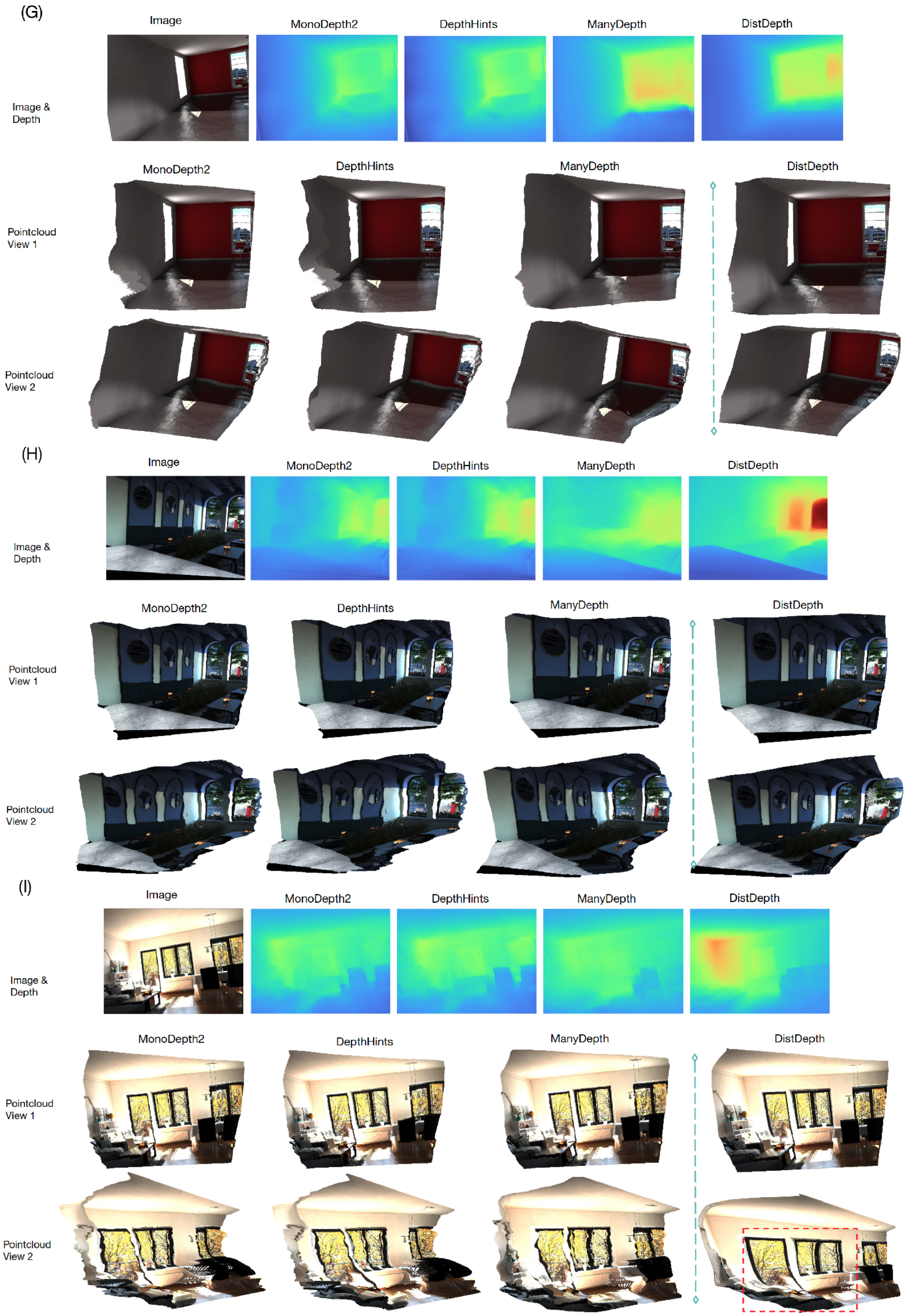
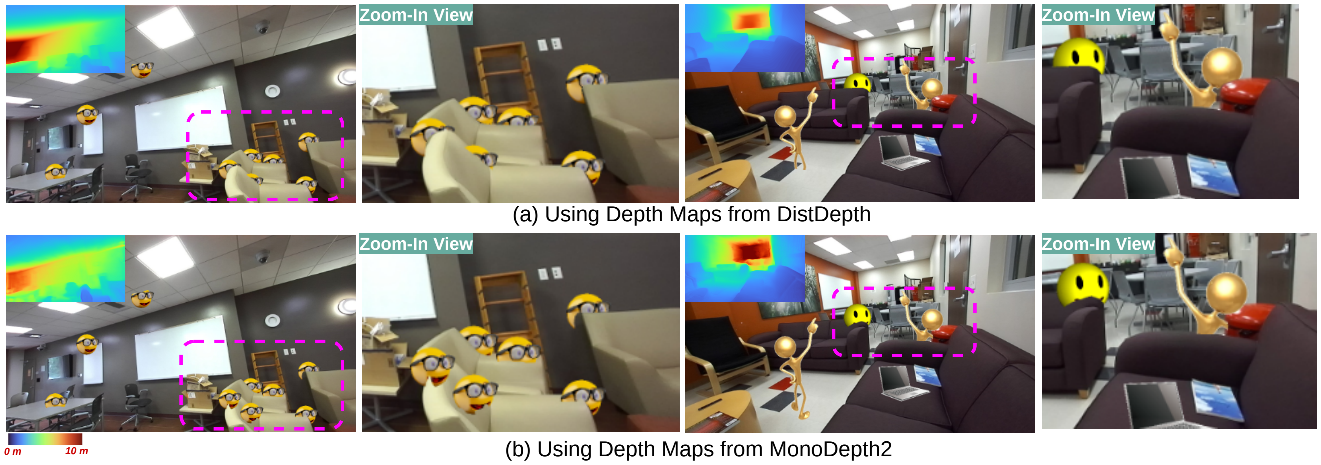


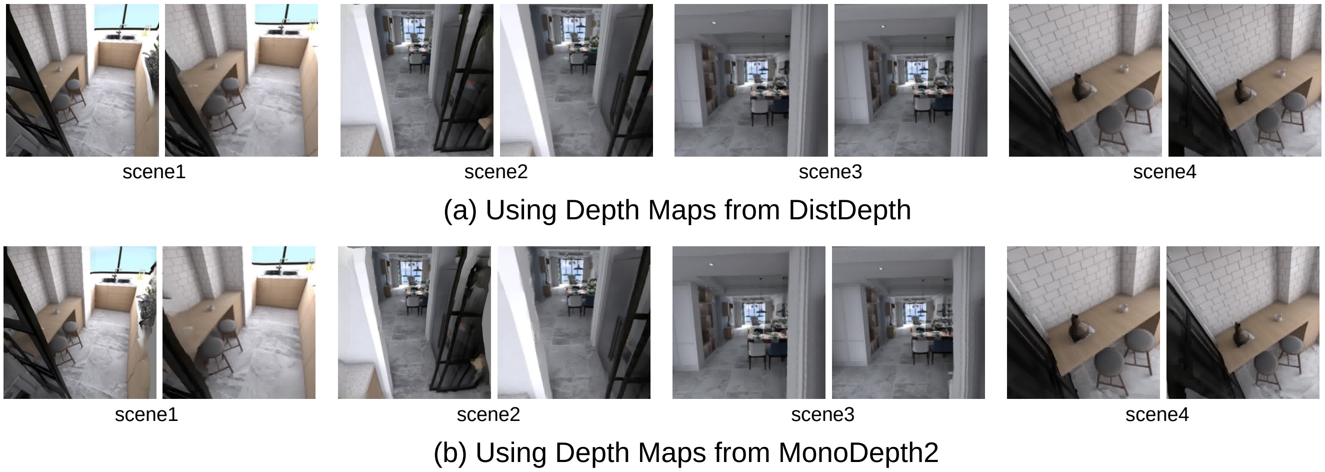
Figure S9. **(Continued) Results on Hypersim.**



(a) Using Depth Maps from DistDepth

(b) Using Depth Maps from MonoDepth2

Figure S10. **Exemplar depth-aware AR occlusion effects and comparison.** We insert several virtual objects into scenes with depth maps to maintain *proper occluding boundaries*. Using depth maps from DistDepth creates more accurate occluding boundaries.



(a) Using Depth Maps from DistDepth

(b) Using Depth Maps from MonoDepth2

Figure S11. **Exemplar 3D photo creation and comparison.** We use images and estimated depth maps to create 3D photos by [18]. Four scenes in two different views are exhibited to show the performances. 3D photos using depth from DistDepth have much less distortions, especially at occluding boundaries. Zoom in for the best view.

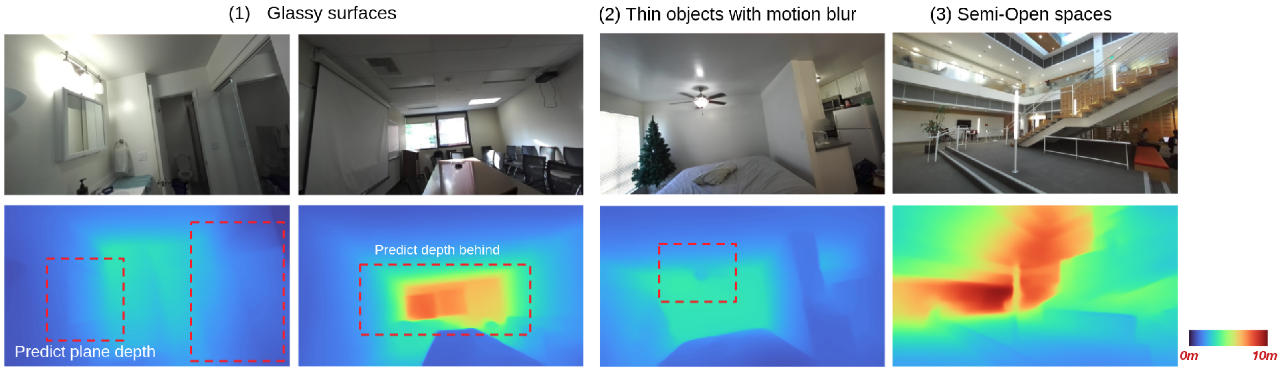


Figure S12. **Failure cases.** (1) Depth values on glossy surfaces may not be perfectly predicted. (2) Structures of thin objects with motion blur may be missing. (3) Training data in either SimSIN or UniSIN are mostly rooms (see Fig. S1), and we predict metric depth that is robust in close ranges. As a result, predictions in semi-open spaces may be less accurate sometimes.

## References

[1] Unreal engine 4. https://www.unrealengine.com/en-US/unreal. 1

[2] Warm harbor environment. https://www.unrealengine.com/marketplace/en-US/product/warmharbor. 1

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 1

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 1

[5] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 1

[6] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of ransac family. *Journal of Computer Vision*, 24(3):271–300, 1997. 2

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[8] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 1

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[10] Yiwen Hua, Puneet Kohli, Pritish Uplavikar, Anand Ravi, Saravana Gunaseelan, Jason Orozco, and Edward Li. Holopix50k: A large-scale in-the-wild stereo image dataset. In *CVPRW*, 2020. 1

[11] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. *3DV*, 2021. 1

[12] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016. 1

[13] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. *NeurIPS Datasets and Benchmarks Track*, 2021. 1

[14] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021. 2

[15] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *ICCV*, 2021. 1

[16] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*. Springer, 2014. 1

[17] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 1

[18] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D photography using context-aware layered depth inpainting. In *CVPR*, 2020. 4, 8

[19] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1

[20] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1

[21] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *NeurIPS*, 2021. 1

[22] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *3DV*, 2019. 1