Layered Depth Refinement with Mask Guidance
– Supplementary Material –

1. Potential Negative Societal Impact

As our proposed method refines depth maps predicted by SIDE models, we do not expect it to have any direct negative societal impact. However, potentially, it can be used to generate more accurate 3D reconstructions of people, and if used in a malicious way, they could be reconstructed accurately in an unwanted way.

2. Image Copyrights

Comparison images in Fig. 6 in the main paper and Fig. 6 are results on the Hypersim dataset (CC-BY SA 3.0 License) [7]. Images with human subjects (identifiable and non-identifiable) in Fig. 1, Fig. 2 and Fig. 8 in the main paper and Fig. 1 and Fig. 7 are from `unsplash` [10] or `pixabay` [4], which are websites with freely licensed images that can be used for commercial and non-commercial purposes. The top image in Fig. 7 in the main paper was officially licensed by Adobe Stock [1] (from eranda - stock.adobe.com). Other generic images are from internal RGB-D datasets.

3. Details of Training Data Generation

Perturbations During training, we apply random dilation and erosion operations on the composite depth map. First, a random number of iterations is selected from $U(1, 5)$ each for dilation (k_d) and erosion (k_e). Then, dilation or erosion with a 3×3 kernel is applied k_d or k_e times with the following order: (i) dilation, erosion, erosion and dilation for 50% of the time, and (ii) erosion, dilation, dilation and erosion the rest of the time. This makes sure that most thin structures and isolated regions are lost in the perturbed depth map. For the Gaussian blur, 50% of the time, we use $\sigma \sim U(0, 1)$ for small amounts of blur, and the rest of the time, we use $\sigma \sim U(1, 5)$ for larger amounts of blur. For human hole perturbations, holes in the mask are detected using the hierarchy computed by `cv2.findContours()`, and for each hole, a random value between the mean depth value inside the original hole and the mean depth value in the outer neighborhood of 10 pixels is assigned.

Effect of Human Hole Perturbation We compare the refined depth results generated by a model trained *without* human hole perturbations and our final model trained *with* human hole perturbations (models in the last two rows in Table 3 of the main paper). As shown in Fig. 1, the initial depth predicts wrong values for holes (isolated background regions) in humans. Without human hole perturbations, the model is able to refine smaller holes (between arm and body) but is incapable of correcting a larger hole (between the legs) as it has not seen such challenging cases during training. The hole perturbation scheme aims to mimic those cases by assigning a random value. This simple strategy enables the refinement model to correct larger holes, as shown in Fig. 1.

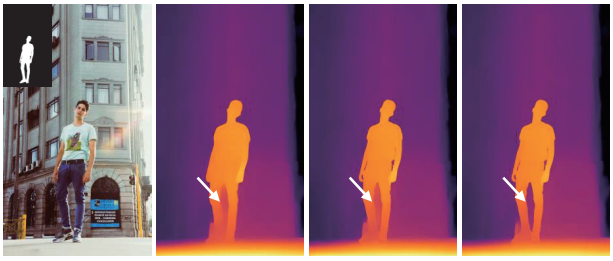
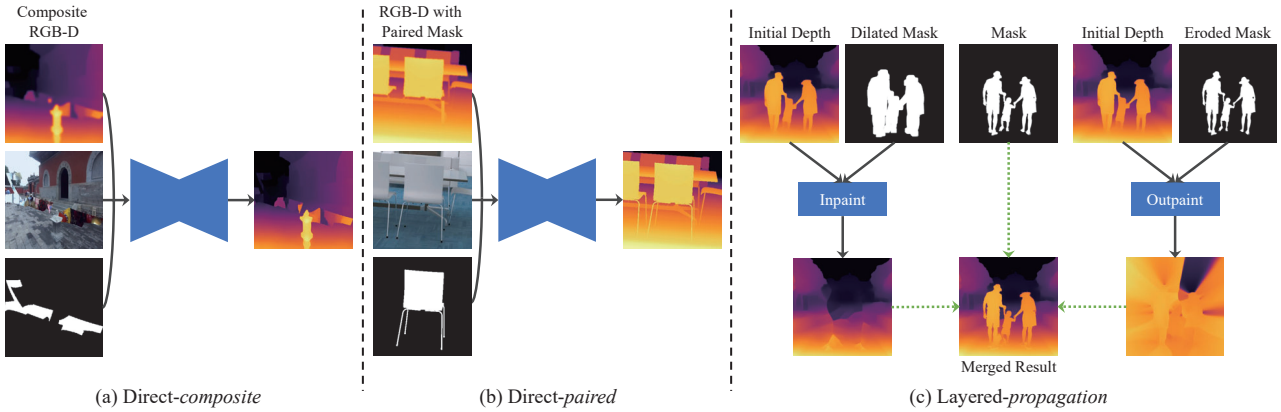


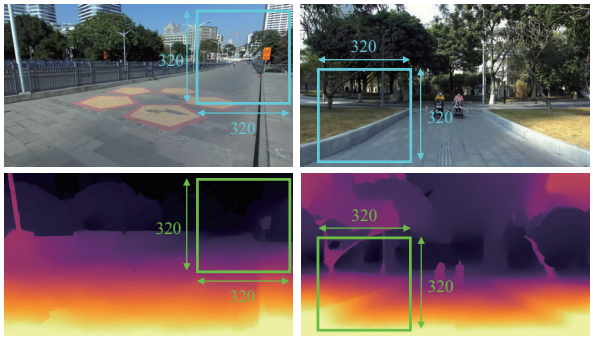
Figure 1. Effect of human hole perturbation. By adding random human hole perturbations when generating the perturbed depth maps during training, our model can correct initially wrong values in large isolated background regions (*holes*) in humans.

Cropping When cropping the mask for training, we filter out small objects by randomly picking objects that are comprised of at least 1% of total pixels in an instance segmentation map. Furthermore, we adaptively crop *around* the object depending on the object size to ensure that the masked region is sufficiently large as shown in Fig. 3. If the object size is smaller than the training patch size (320×320), we randomly crop by the patch size at locations where the entire object is inside the patch. If the object size ($H \times W$) is bigger than the patch size, we crop by $p \times p$, where $p \sim U(s, 2s)$ and s is $\max(H, W)$, at random locations where the entire object is inside the patch. Then, the cropped patch is resized to the training patch size so that it can be used for randomly compositing the RGB and depth map patches. Without this cropping scheme, the mask region often only contains parts

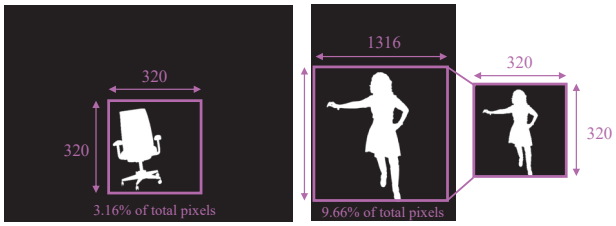


(a) Direct-*composite* (b) Direct-*paired* (c) Layered-*propagation*

Figure 2. Illustrations of baseline models used in our experiments.



(a) Randomly selected RGB and depth are random cropped



(b) Object masks are randomly cropped *around* the object considering the object size (*adaptive cropping*)

Figure 3. RGB-D and mask cropping during training.

of objects or no objects at all (if simply cropped at random locations). For stuff classes (e.g., sky), we crop with $p \sim U(H/2, H)$ at a random location.

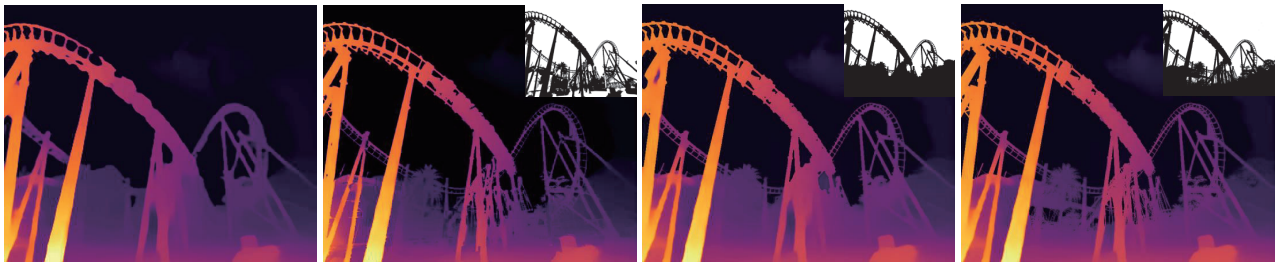
4. Details of Baseline Models

In the main paper, we compared to four baseline models that perform mask-guided depth refinement: Direct-*composite*, Direct-*paired*, Layered-*propagation* and Layered-*ours*. An illustration of the baselines is shown in Fig. 2. In Fig. 2 (a), Direct-*composite* predicts the refined output without layering by training on composite RGB-D inputs. Direct-*paired* also refines without layering but is trained on a paired mask and RGB-D dataset [7] as shown in Fig. 2 (b). We employ the same model architecture as the network shown in Fig. 5 of the main paper for Direct-*composite* and Direct-*paired*.

For Layered-*propagation*, we run the propagation-based image completion algorithm [9] twice to obtain layered outputs, once with the dilated mask for inpainting and the second time with the eroded mask for outpainting as shown in Fig. 2 (c). The two outputs are then merged based on the mask similar to our proposed 2-layer approach. For Layered-*ours*, the same procedure as Fig. 2 (c) is applied but we use our model after stage I training instead of [9] for inpainting/outpainting. For the layered baselines, dilation and erosion are necessary to correct the initially wrong values and their kernel sizes should be set *heuristically* for each input depth to get the best results, unlike our proposed method that is able to automatically figure out the regions to inpaint/outpaint while refining inaccurate areas *without* any heuristics.

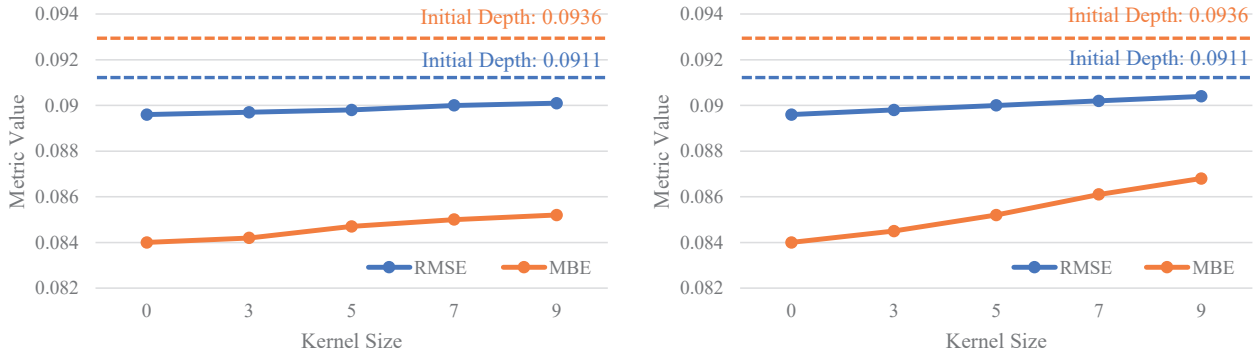
5. Analysis on Mask Quality

As our method refines the initial depth map based on the input mask, its refinement performance is inevitably dependent on the mask quality. To analyze the effect of using different types of masks, in Fig. 4, we show the refined outputs using three different masks generated using commercial masking tools: (i) automatically generated mask from `removebg`, (ii) automatically generated mask using `Photoshop`, and (iii) manually edited mask using `Photoshop`. As shown in Fig. 4, using automatically generated masks already produces significantly enhanced results. With additional manual editing (Fig. 4 (d)), the depth map can be refined even further. In practical application scenarios, users can edit masks instead in order to edit depth maps, which would be easier and more intuitive.



(a) Initial Depth (b) Automatic (remove.bg) (c) Automatic (Photoshop) (d) Manual (Photoshop)

Figure 4. Ablations on automatic and manual mask inputs.



(a) Masks with morphological opening (b) Masks with morphological closing

Figure 5. Quantitative results with degraded masks.

For a numerical analysis on mask quality, we apply morphological opening and closing operations with kernel sizes $k \in \{3, 5, 7, 9\}$ on the ground truth instance segmentation maps from Hypersim [7] and measure the MBE and RMSE after refining the depth maps generated with DPT-Large [5]. The results are plotted in Fig. 5, where $k = 0$ denotes the case using the original ground truth segmentation maps and the dotted lines signify the average metric values of the initial depth maps. As shown in Fig. 5, the error values increase with more severe degradation as expected. However, they are still better than the initial depth.

6. Inference Time

For inference, it takes 16 ms for the initial depth prediction [5, 6] and an additional 78 ms for our refinement method with an NVIDIA TITAN RTX GPU. Note that input images are resized to the spatial resolution used during training prior to entering the network for all methods, 384×384 for [5, 6] and 320×320 for ours.

7. More Visual Results

More results on paired datasets In Fig. 6, we provide more examples on Hypersim [7] along with the relative improvement maps visualizing where the refinement method improved and worsened the initial depth estimation in terms of absolute error. Miangoleh *et al.*'s method [2] often worsens homogeneous regions whereas our method mostly refines along the mask boundaries (edges and holes) and leaves other regions intact.

More results using point clouds In Fig. 7, we visualize the frontal, side and top views of the scene using point cloud representations. With our refined depth, objects are more clearly and accurately cut around the edges and hole regions, resulting in significantly less flying pixels. This can potentially benefit applications such as 3D photography [3, 8].

More results in the wild We provide additional results on real images as image files as part of the supplementary material. We further provide an html file, `comparisons.html`, for easier visual comparisons among the initial depth [5], Miangoleh *et al.*'s refinement method [2] and Ours.

References

[1] AdobeStock. Website with a collection of licensed images. https://stock.adobe.com/. [Online; accessed 16-November-2021]. 1

[2] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting Monocular Depth

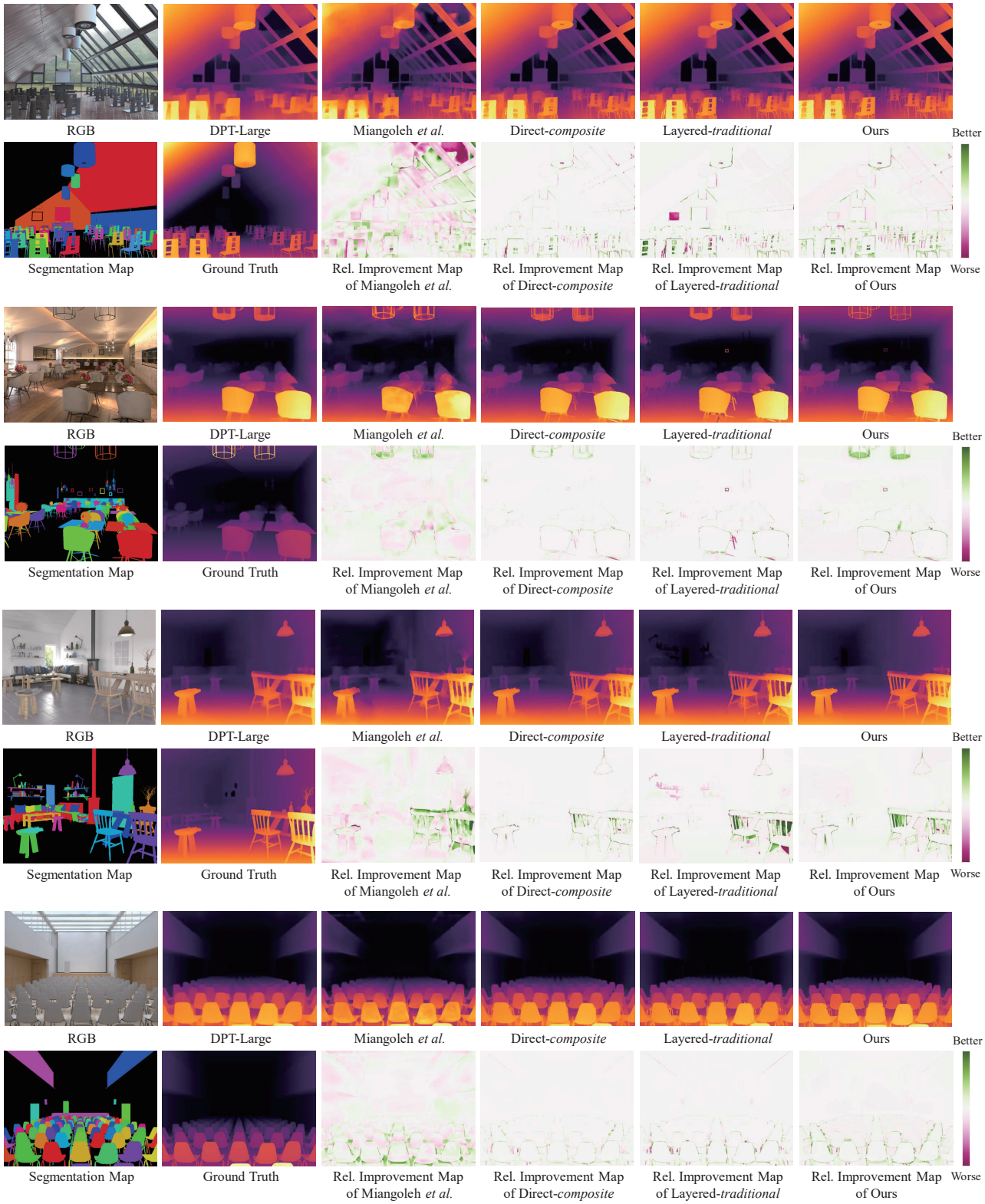


Figure 6. Qualitative results on Hypersim [7]. The relative improvement maps visualize where the refinement method improved and worsened the initial depth estimation by DPT [5]. Our method focuses on the edges and hole regions, accurately refining fine structures.

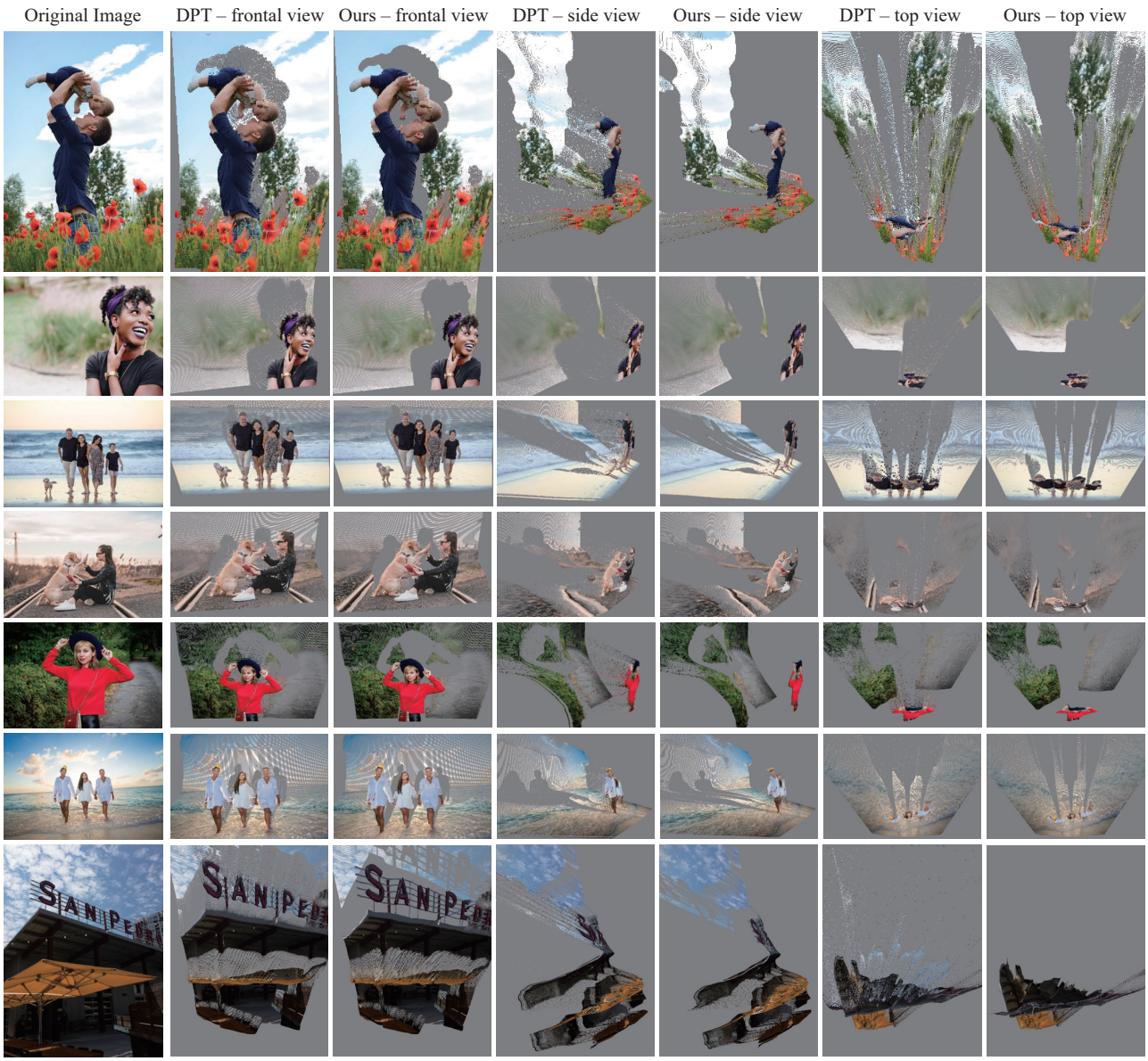


Figure 7. Point cloud visualizations using the initial depth by DPT [5] and refined depth by Ours. With the refined depth, there are less flying pixels and objects are more clearly cut in the frontal, side and top views of the scene.

Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3

[3] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D Ken Burns Effect from a Single Image. *ACM Transactions on Graphics*, 38(6):184:1–184:15, 2019. 3

[4] pixabay. Website with free images that can be used for commercial and non-commercial purposes. https://pixabay.com/. [Online; accessed 16-November-2021]. 1

[5] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *IEEE International Conference on Computer Vision*, 2021. 3, 4, 5

[6] 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*, 2020. 3

[7] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *IEEE International Conference on Computer Vision*, 2021. 1, 2, 3,

4

[8] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D Photography Using Context-Aware Layered Depth Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[9] Alexandru Telea. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools*, 9(1):23–34, 2004. 2

[10] unsplash. Website with free images that can be used for commercial and non-commercial purposes. https://unsplash.com/. [Online; accessed 16-November-2021]. 1