Depth Field Networks for Generalizable Multi-view Scene Representation – Supplementary Material –

Vitor Guizilini[1]*, Igor Vasiljevic[1]*, Jiading Fang[2]*, Rares Ambrus[1], Greg Shakhnarovich[2], Matthew R. Walter[2], and Adrien Gaidon[1]

[1] Toyota Research Institute, Los Altos, CA
[2] Toyota Technological Institute at Chicago, Chicago, IL

1 Implementation Details

1.1 Training parameters

We implemented our models using PyTorch, with distributed training across eight A100 GPUs. We used grid search to choose training parameters, that include: view synthesis weight $\lambda_s = 5.0$, virtual camera loss weight $\lambda_v = 0.5$, virtual camera projection noise $\sigma_v = 0.25$, canonical jittering noise $\sigma_t = \sigma_r = 0.1$, and batch size $b = 32$ (4 per GPU). We use the AdamW optimizer [5], with standard parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, a weight decay of $w = 10^{-4}$, and an initial learning rate of $lr = 2 \cdot 10^{-4}$. For our stereo experiments, we train for 200 epochs, halving the learning rate every 80 epochs. For our video experiments, we train for 100 epochs, halving the learning rate every 40 epochs. Higher-resolution fine-tuning is performed for 50 epochs for stereo experiments, and 10 epochs for video experiments, with $lr = 2 \cdot 10^{-5}$.

1.2 Architecture Details

Following recent work [9], we use $K_o = 20$ and $K_r = 10$ as the number of Fourier frequencies for camera embeddings, with maximum resolution $\mu_o = \mu_r = 60$. Our encoder embeddings have dimensionality $C_e = 960 + 186 = 1146$, due to the use of both image and camera information. Our decoder embeddings have dimensionality $C_d = 186$, since only camera information is required to produce estimates. Our latent representation $\mathcal{R}$ is of dimensionality 2048×512. Input images are resized to 128×192, and following standard protocol [8] output depth maps are compared to ground-truth resized to 480×640. We use the following hyperparameters for our Perceiver IO implementation: 1 block, 1 input cross-attention, 8 self-attention layers (with 8 heads) and 1 output cross-attention. Cross attention layers have only 1 head. We found that larger Perceiver IO models (i.e., with more blocks, number of heads, and self-/cross-attention layers) did not improve results and significantly increased training time. The latest developments in the Perceiver architecture [1] could be used to further improve performance and inference speed, and will be considered in future work.

* Denotes equal contribution.

Train \ Test	0.0m	0.1m	0.2m	0.5m
0.0m	0.101	0.202	0.226	0.291
0.1m	0.097	0.160	0.184	0.242

Table 1: **Effects of canonical jittering at train and test time**. The model trained with canonical jittering ($\sigma_t = \sigma_r = 0.1$m) not only performs better when evaluated at the target location ($\sigma_t = \sigma_r = 0.0$m), but is also more robust to different levels of canonical jittering at test time. The results shown are average Abs. Rel. of the two predicted stereo depths maps, without ground-truth scaling.

2 Canonical Jittering Test-time Ablation

In Section 4.3 of the main text, we ablate the effects of using our proposed data augmentation techniques, designed to improve multi-view consistency in the learned latent representation. In Figure 4b we provide an additional experiment in which we vary the amount of virtual camera noise σ_v at train and test time, and show that training at higher noise levels not only improves depth estimation performance at the target location (up to a certain value, of $\sigma_v = 0.25$m), but also when decoding estimates from novel viewpoints.

Here we perform a similar experiment targeting another proposed data augmentation technique: canonical jittering. Two different models were trained, with and without canonical jittering, and both were evaluated under different noise levels at test time. Note that, while this augmentation does not change scene geometry, it changes the camera embeddings used for encoding and decoding information. Results are presented in Table 1. As we can see, the model trained with canonical jittering not only performs better when evaluating at the target location, but is also more robust to increasing levels of noise at test time.

3 Higher Resolution Fine-Tuning

One of the main challenges of training Transformer-based architectures has been the $O(N^2)$ self-attention memory scaling with input size. This means that the resolution of recent models has been fairly limited (e.g. the view synthesis model of Sajjadi et al. [7] primarily trains on low-resolution images, with a highest resolution of 128×176), hindering their application to real-world scenes. Perceiver IO decouples input resolution from the the learned latent representation, which enables training and real-time inference at higher resolutions [9]. In our experiments, we found it advantageous to train using a resolution curriculum, first at a lower resolution (128×192), and then fine-tune at a higher resolution (240×320). Note that, because the camera parameters are also scaled to the proper resolution, the scene geometry does not change, only (a) the number of embeddings generated per camera, and (b) the image embeddings, since resolution changes image features. Thus, training at lower resolutions enables

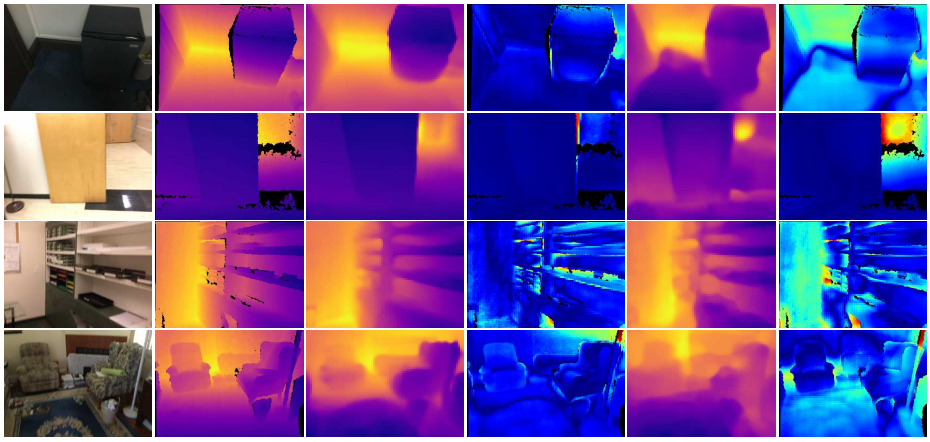

Fig. 1: **Qualitative comparison of DeFiNe** relative to the IIB [9] baseline. Our architecture improves depth estimation quality in (i) smooth and textureless areas, (ii) far away regions, and (iii) image boundaries and depth discontinuities.

the faster learning of our desired multi-view latent scene representation, which can then be fine-tuned at higher resolutions for further improvements. As an alternative, we also experimented with the strategy of *sampling* rays at higher resolution (similar to NeRF [6] and SRT [7]). However, we found that this approach led to unstable training and longer convergence times. As future work, we plan to investigate how training and and inference can be scaled up to even higher resolutions.

4 Comparison to IIB

IIB [9] is a recently proposed stereo depth estimation method that also uses a Perceiver IO-based architecture. Their major contribution is a geometrically-motivated epipolar inductive bias to encourage multi-view consistency. In Table 1 and Figure 4 of the main text, we show that our DeFiNe architecture significantly improves over the IIB baseline on the ScanNet-Stereo benchmark (0.116 vs. 0.089 Abs. Rel.). Given that code and pre-trained models to replicate the IIB results are not available, we trained a model following the instructions in [9], achieving similar performance as reported in the paper.

Some qualitative examples from this model are depicted in Figure 1, as well as examples from our DeFiNe architecture. As we can see, our proposed 3D augmentations and joint view synthesis learning also lead to significant qualitative improvements over IIB results. In particular, we consistently perform better in smooth and textureless areas, as well as far away regions and depth discontinuities. We attribute this behavior to an increase in scene diversity due to our contributions, that enables the learning of a more consistent multi-view latent scene representation.

Timestep	−4	−3	−2	−1	0	+1	+2	+3	+4
% valid pixels	77.7	68.5	62.6	59.5	58.2	58.7	61.2	66.6	75.9
Monodepth2 [2]	0.325	0.336	0.346	0.354	0.361	0.359	0.354	0.347	0.338
PackNet [3]	0.305	0.318	0.330	0.341	0.344	0.344	0.336	0.327	0.338
BTS [4]	0.296	0.306	0.320	0.329	0.334	0.326	0.327	0.319	0.303
DeFiNe (projection)	0.226	0.239	0.251	0.259	0.268	0.261	0.254	0.244	0.231
DeFiNe (query)	0.222	0.230	0.237	0.240	0.242	0.246	0.245	0.240	0.231
DeFiNe (query, all)	0.361	0.381	0.398	0.408	0.412	0.413	0.406	0.390	0.369

(a) Depth interpolation results. Frames at $\{t-5, t+5\}$ are encoded, and depth maps corresponding to camera locations at $\{t-4, \ldots, t+4\}$ are decoded.

Timestep	0	1	2	3	4	5	6	7	8
% valid pixels	91.0	76.1	64.5	55.9	49.6	45.2	42.0	39.5	36.0
Monodepth2 [2]	0.351	0.386	0.398	0.405	0.412	0.420	0.431	0.441	0.453
PackNet [3]	0.327	0.358	0.378	0.391	0.400	0.406	0.420	0.428	0.436
BTS [4]	0.315	0.331	0.357	0.377	0.392	0.401	0.413	0.424	0.429
DeFiNe (projection)	0.258	0.276	0.288	0.298	0.311	0.323	0.331	0.340	0.348
DeFiNe (query)	0.237	0.260	0.271	0.280	0.289	0.298	0.307	0.317	0.326
DeFiNe (query, all)	0.326	0.370	0.405	0.438	0.468	0.495	0.520	0.543	0.563

(b) Depth extrapolation results. Frames at $\{t-5, \ldots, t-1\}$ are encoded, and depth maps corresponding to camera locations at $\{t, \ldots, t+8\}$ are decoded.

Table 2: **Depth interpolation and extrapolation results**, on ScanNet (complementary to Figures 7a and 7b of the main text). On valid projected pixels, DeFiNe (query) outperforms the explicit projection of all considered single-frame baselines, and it also outperforms the explicit projection of its own estimates, obtained from encoded views (projection). Furthermore, it also enables the estimation of dense depth maps from novel viewpoints, which can be compared to the corresponding ground-truth from that location (query, all).

5 Depth from Novel Viewpoints

In Table 2 we provide numerical values to complement our depth interpolation and extrapolation experiments (Figures 7a and 7b from the main text). These experiments show that querying from our learned latent representation improves over the explicit projection of information from encoded views, while also enabling the estimation of dense depth maps from novel viewpoints. Similarly, in Figure 2 we provide additional qualitative examples of depth extrapolation to future timesteps, showing how DeFiNe can reconstruct unseen portions of the environment in a geometrically-consistent way.

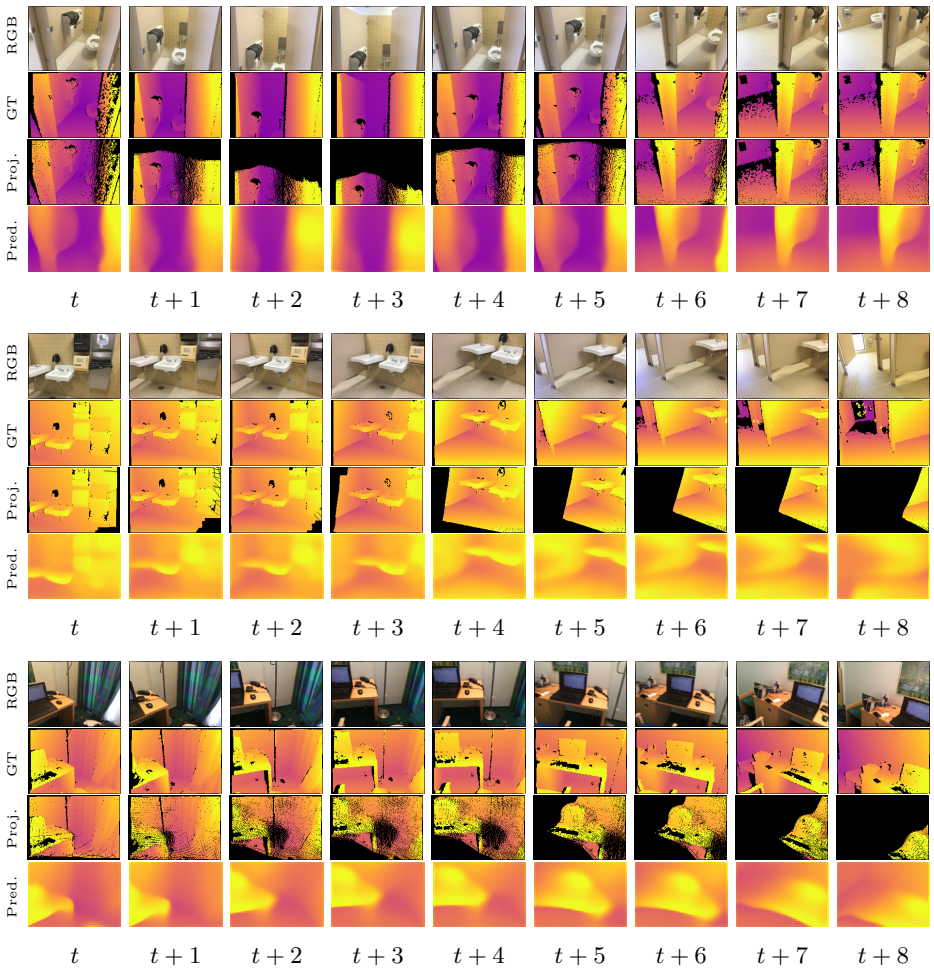

Fig. 2: **ScanNet depth extrapolation examples**, using DeFiNe. In each example, image and camera information from frames at $[t-5, \ldots, t-1]$ is encoded, and depth maps corresponding to camera locations at $[t, \ldots, t+8]$ are decoded, using only camera information. For each timestep, we show sparse projected ground-truth depth maps (third row), and dense predicted depth maps (fourth row). Our DeFiNe architecture is able to extrapolate from encoded information to fill in missing parts of the scene.

References

1. Carreira, J., Koppula, S., Zoran, D., Recasens, A., Ionescu, C., Henaff, O., Shelhamer, E., Arandjelovic, R., Botvinick, M., Vinyals, O., Simonyan, K., Zisserman, A., Jaegle, A.: Hierarchical perceiver. arXiv preprint arXiv:2202.10890 (2022)
2. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction. In: Proceedings of the International Conference on Computer Vision (ICCV) (2019)
3. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3D packing for self-supervised monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
4. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv:1907.10326 (2019)
5. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
6. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 405–421 (2020)
7. Sajjadi, M.S., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lucic, M., Duckworth, D., Dosovitskiy, A., Uszkoreit, J., Funkhouser, T., Tagliasacchi, A.: Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. arXiv preprint arXiv:2111.13152 (2021)
8. Teed, Z., Deng, J.: DeepV2D: Video to depth with differentiable structure from motion. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
9. Yifan, W., Doersch, C., Arandjelović, R., Carreira, J., Zisserman, A.: Input-level inductive biases for 3D reconstruction. arXiv preprint arXiv:2112.03243 (2021)