FaDIV-Syn: Fast Depth-Independent View Synthesis using Soft Masks and Implicit Blending (Supplementary Material)

Andre Rochow
University of Bonn
rochow@ais.uni-bonn.de

Max Schwarz
University of Bonn
schwarz@ais.uni-bonn.de

Michael Weinmann
Delft University of Technology

Sven Behnke
University of Bonn

In the scope of this supplementary material, we provide further details on the involved network architectures, further results regarding data efficiency as well as as further qualitative results and discussions of limitations and failure cases.

A. Network Architecture Details

Here, we provide more details regarding the architectures of our Soft-Masking network (see Table I), our fusion network (see Table II) as well as the NoSM network architecture (see Table III).

Input	k	c	Output
gray(PSV)	3	$\min(2 \cdot N, 256)$	$down_1$
Pool($down_1$)	3	$\min(2^3 \cdot N, 256)$	$down_2$
Pool($down_2$)	3	$\min(2^4 \cdot N, 256)$	$down_3$
Pool($down_3$)	3	$\min(2^5 \cdot N, 256)$	$down_4$
Pool($down_4$)	3	$\min(2^4 \cdot N, 256)$	up_1
Up(up_1), $down_3$	3	$\min(2^3 \cdot N, 256)$	up_2
Up(up_2), $down_2$	3	$\min(2^2 \cdot N, 256)$	up_3
Up(up_3), $down_1$	3	$\min(2^1 \cdot N, 256)$	up_4
Up(up_4), P	3	$2 \cdot N$	$final$
softmax($masking$)	-	N	$pred$

TABLE I: Soft-Masking network architecture for N grayscale depth planes in the PSV. Each row denotes a convolutional layer, where k is the kernel size and c is the number of output features. **Pool** is 2×2 average pooling, and **Up** denotes bilinear upsampling with a factor of 2.

Input	k_1	c_1	k_2	c_2	Output
PSV	3	$6 \cdot N + N$	3	$12 \cdot N$	$groupconv$
G($groupconv$)	3	$6 \cdot N$	3	$3 \cdot N$	$bottleneck$
$bottleneck$	3	$6 \cdot N$	3	$6 \cdot N$	$down_1$
$down_1$	3	128	3	128	$down_2$
$down_2$	3	256	3	256	$down_3$
$down_3$	3	256	3	256	$down_4$
$down_4$	3	256	3	256	$dilated$
Up($dilated$), $down_3$	3	256	3	128	up_1
Up(up_1), $down_2$	3	128	3	$6 \cdot N$	up_2
Up(up_2), $down_1$	3	$6 \cdot N$	3	$3 \cdot N$	up_3
Up(up_3), $bottleneck$	3	32	3	32	up_4
up_4	1	3	–	–	$pred$

TABLE II: Fusion network with N depth planes in the PSV. Each row shows 2 convolutional layers, where k is the kernel size and c is the number of output features. **G** denotes the gating operation and **Up** denotes upsampling.

Input	k_1	c_1	k_2	c_2	Output
PSV	3	$6 \cdot 6$	3	$12 \cdot 6$	$groupconv$
G($groupconv$)	3	$6 \cdot N$	3	$6 \cdot N$	$bottleneck$
G($bottleneck$)	3	$6 \cdot N$	3	$12 \cdot N$	$down_1$
G($down_1$)	3	128	3	256	$down_2$
G($down_2$)	3	256	3	512	$down_3$
G($down_3$)	3	256	3	1024	$down_4$
G($down_4$)	3	512	3	512	$dilated$
Up($dilated$), $G(down_3)$	3	256	3	256	up_1
Up(up_1), $G(down_2)$	3	$12 \cdot N$	3	$12 \cdot N$	up_2
Up(up_2), $G(down_1)$	3	$6 \cdot N$	3	$6 \cdot N$	up_3
Up(up_3), $G(bottleneck)$	3	32	3	32	up_4
up_4	1	3	–	–	$pred$

TABLE III: NoSM architecture with N depth planes in the PSV. Each row shows 2 convolutional layers, where k is the kernel size and c is the number of output features. **G** denotes the gating operation which reduces the number of feature maps by factor 2 and **Up** denotes upsampling.

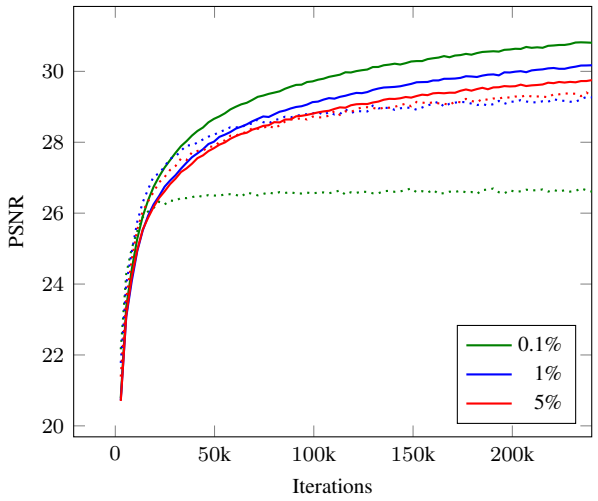

Fig. 1: PSNR on reduced training (solid) and validation (dotted) splits of the RealEstate10k dataset during 17-NoSM network training.

B. Extended Data Efficiency Results

Due to lack of space in the main paper, we shifted details on the data efficiency experiments that belong to Section 4.2 into the supplemental. We train our 17-NoSM ablation on smaller fractions of the full RealEstate10k training dataset, and evaluate on the full test set. The dataset size is reduced by

Train	Model	$\Delta t = 2$			$\Delta t = 5$			$\Delta t = 10$		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
0.1%	17-NoSM	$31.91_{-11.7\%}$	$.9361_{-3.19\%}$	$.0381_{+140\%}$	$28.03_{-13.5\%}$	$.8825_{-6.32\%}$	$.0712_{+122\%}$	$24.70_{-13.4\%}$	$.8129_{-9.07\%}$	$.1257_{+103\%}$
1%	17-NoSM	$34.93_{-3.31\%}$	$.9607_{-0.64\%}$	$.0191_{+20.1\%}$	$31.38_{-3.20\%}$	$.9320_{-1.06\%}$	$.0360_{+12.2\%}$	$27.60_{-3.24\%}$	$.8789_{-1.69\%}$	$.0695_{+12.1\%}$
5%	17-NoSM	$35.19_{-2.58\%}$	$.9616_{-0.55\%}$	$.0173_{+8.81\%}$	$31.92_{-1.54\%}$	$.9366_{-0.57\%}$	$.0327_{+1.87\%}$	$28.20_{-1.16\%}$	$.8874_{-0.74\%}$	$.0631_{+1.77\%}$
35%	17-NoSM	$\mathbf{36.15}_{+0.08\%}$	$.9658_{-0.11\%}$	$\mathbf{.0158}_{-0.63\%}$	$\mathbf{32.52}_{+0.32\%}$	$.9417_{-0.03\%}$	$\mathbf{.0307}_{-4.36\%}$	$\mathbf{28.59}_{+0.24\%}$	$\mathbf{.8944}_{+0.05\%}$	$\mathbf{.0605}_{-2.42\%}$
100%	17-NoSM	$36.12_{+0.00\%}$	$\mathbf{.9669}_{+0.00\%}$	$.0159_{+0.00\%}$	$32.42_{+0.00\%}$	$\mathbf{.9420}_{+0.00\%}$	$.0321_{+0.00\%}$	$28.53_{+0.00\%}$	$.8940_{+0.00\%}$	$.0620_{+0.00\%}$

TABLE IV: Data efficiency experiment. The *Train* column shows the training dataset size relative to the full RealEstate10k train split.

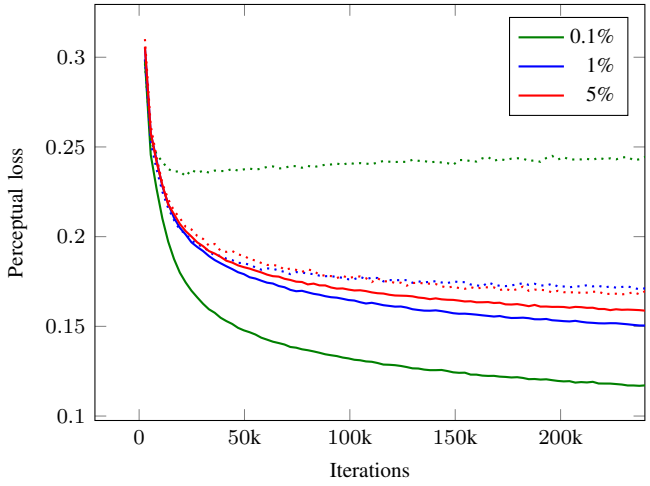

Fig. 2: Perceptual loss on reduced training (solid) and validation (dotted) splits of the RealEstate10k dataset during 17-NoSM network training.

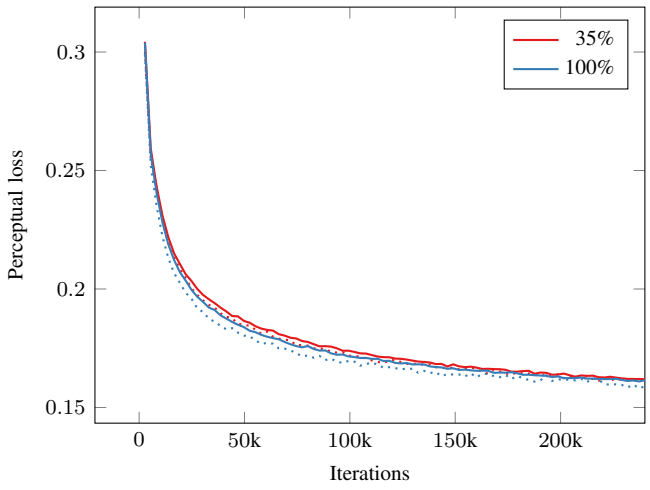

Fig. 3: Perceptual loss on reduced training (solid) and validation (dotted) splits of the RealEstate10k dataset during 17-NoSM network training.

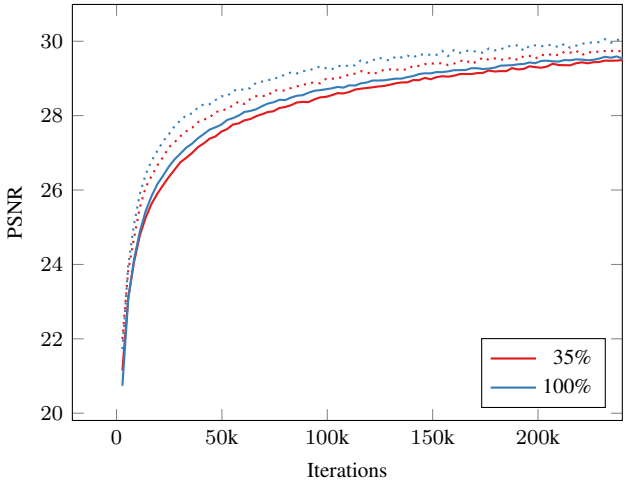

Fig. 4: PSNR on reduced training (solid) and validation (dotted) splits of the RealEstate10k dataset during 17-NoSM network training.

randomly choosing scenes until the specified size is met (35%, 5%, 1%, and 0.1%). As shown in Table IV, all sizes from 35% to 1% give sufficiently good results, where 35% even performs similarly or slightly better than our model trained on the full dataset. It is possible that further training may yield advantages, since we set an upper bound on the training iterations as described in Section 3.3. However, we conclude that 35% of RealEstate10k still contains enough scene and pose variance to prevent the network from overfitting (see Fig. 4). This is to be expected, since the triplet sampling during training greatly augments the number of training samples. The 1% network maintains good performance for SSIM and PSNR but starts losing significantly in LPIPS. Finally, the 0.1% network loses significant performance in all metrics and seems to be outside of the boundary for satisfactory results. As Figs. 1 and 2 show, we observed significant drops in validation performance for the 1% and 0.1% training split. Starting with 35%, we observe that the validation score is actually better than the training score (see Figs. 3 and 4), which is caused by batch normalization: The average parameters used during evaluation seem to work more robustly than the on-line statistics computed for each batch during training. Overall, we

Method	Iterations	SSIM↑	PSNR↑	LPIPS↓
Ours-19	330k	.8985	29.04	.0583
Ours-19	660k	.8989	**29.27**	.0558
Ours-19	990k	**.9008**	29.19	**.0552**

TABLE V: Long-time training behavior. The table shows extrapolation results on the RealEstate10k [3] test set. All variants are trained with 288p, but evaluated on 576p.

conclude that above 35% there are no indications of overfitting at all.

C. Limitations and Failure Cases

We present some examples for failure cases in Figs. 6 to 8.

a) Inpainting: Even though our method is designed in such a way that inpainting and PSV fusion can be performed simultaneously it often struggles to inpaint large missing regions. We visualize two examples in Fig. 6. We expect that this may be a result of (1) the limited receptive field and (2) a number of learned parameters which is insufficient for filling in large regions with reasonable and realistic content. We also show 3D-Photo [1] results in Fig. 6, which uses a separate inpainting network. However, we believe that both our method and 3D-Photo perform similar in inpainting regions, so this does not seem to be an architectural advantage.

b) Camera pose errors: The RealEstate10k dataset has been annotated with ORB-SLAM2 [2] and bundle adjustment. This sometimes leads to inaccurate camera poses. While our method can generally handle small misalignments, larger errors can cause blurred regions as demonstrated in Fig. 7. We expect that it could be advantageous to allow small camera pose corrections instead of assuming that they are fixed. However, predicting camera offsets must be embedded into the pipeline in a learned fashion, unless per scene optimization is desired.

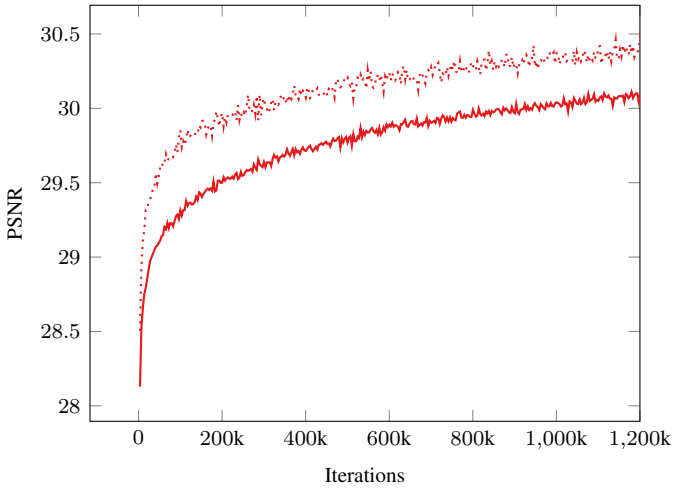

Fig. 5: Long-time training behavior. We show training (solid) and validation (dotted) PSNR during training for more iterations.

c) Biased training data: Our method is trained in such a way that target poses are always on the camera trajectory of the RealEstate10k dataset, where ground truth is available. However, this induces a bias, which may result in less performance for target views outside of the smooth camera trajectories. It could be possible to improve generalization to such poses using semi-supervised techniques [4].

d) Slow convergence: We note that the experiments in the main paper were achieved with a limited number of training iterations, i.e., we did not train until there was no improvement anymore. Table V compares the accuracy of three equivalent networks which were trained for an ascending number of iterations on the test set provided by [1]. Table V and Fig. 5 illustrate that more training still improves the testing/validation accuracy and the optimal performance is not yet achieved. It would be desirable to reach the optimal performance in significantly less training iterations to achieve better results without changing the method itself.

D. Additional Qualitative Results

In addition to the exemplary results already shown, we present more qualitative examples here. Figures 9 and 10 show extrapolation examples for our full network variants (Ours-32, Ours-19) that use Soft-Masking, in comparison to ground truth, 3D Photo [1], and Stereo-Mag [3]. In Fig. 11 we illustrate that our method ablation Ours-17-NoSM without Soft-Masking achieves also reasonable results. To further demonstrate the generalization capability of our method to higher resolutions, we show interpolation sequences in Figs. 12 and 13 with models that are trained in 288p but inferred in 576p (twice the training resolution).

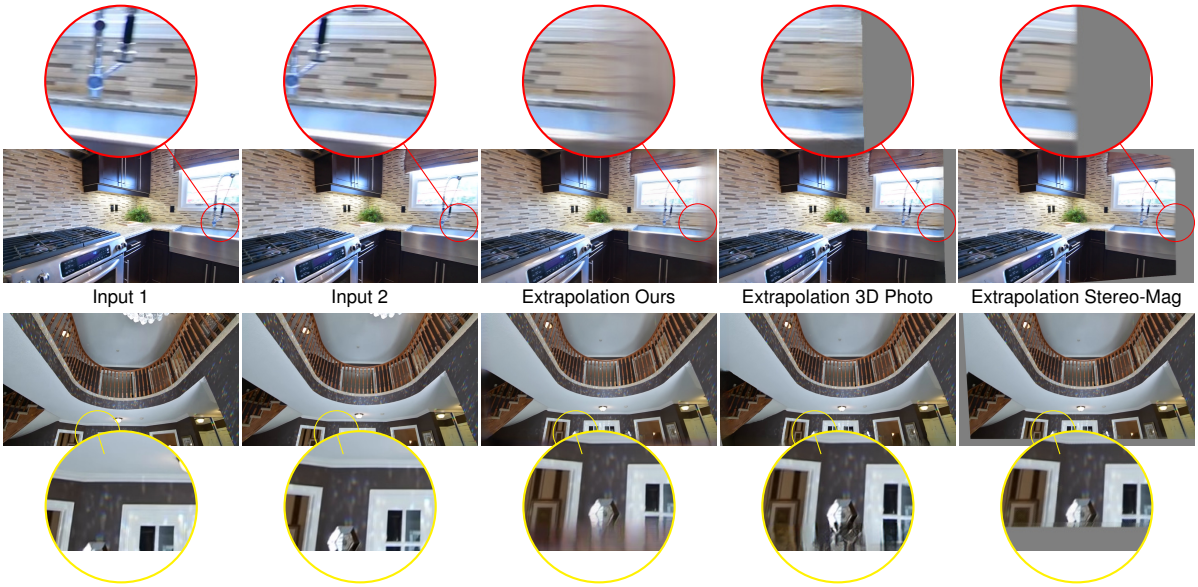

Fig. 6: Limitations of FaDIV-Syn: Inpainting at borders.

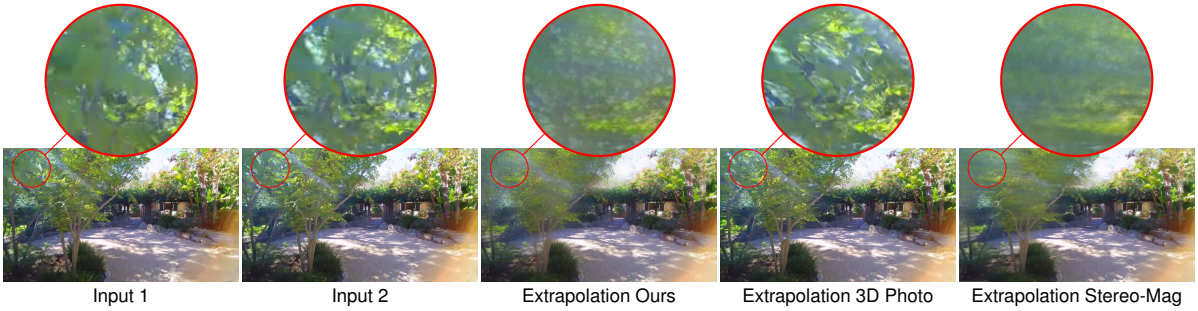

Fig. 7: Limitations of FaDIV-Syn: Insufficient pose alignment.

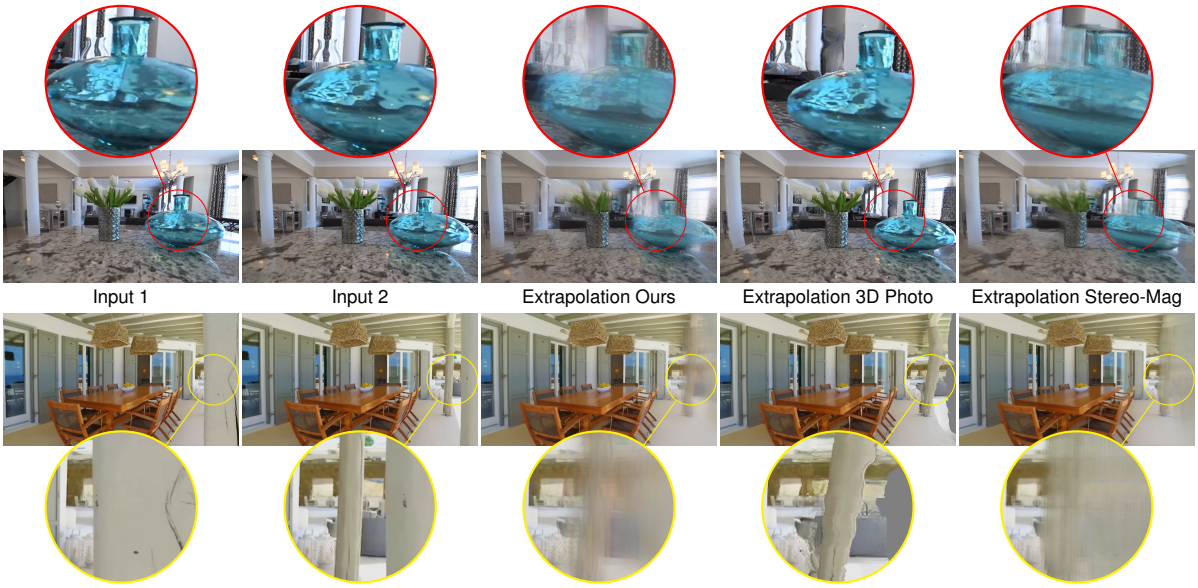

Fig. 8: Limitations of FaDIV-Syn: General failure cases.

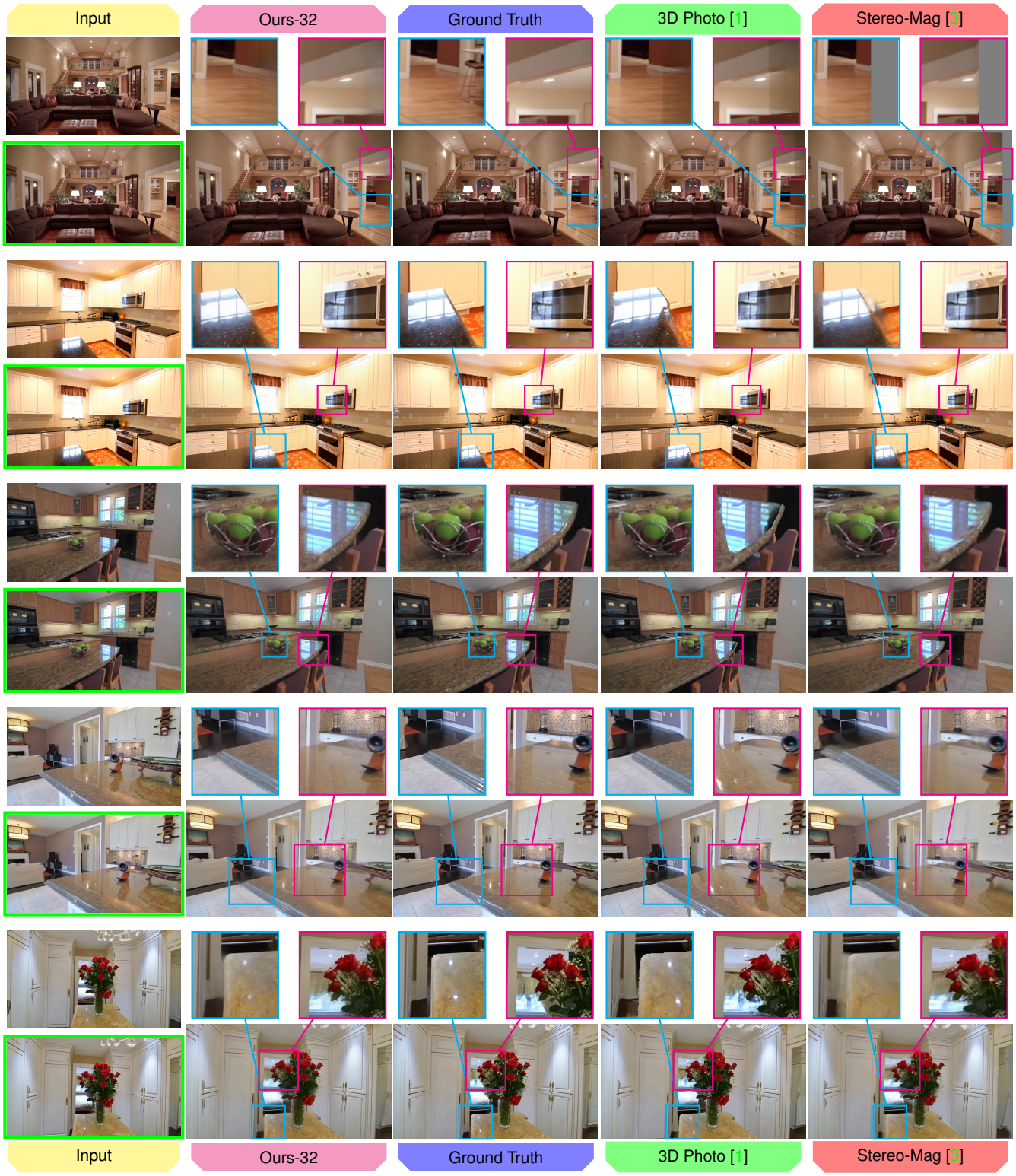

Fig. 9: Extrapolation comparison of our method with 32 planes (Ours-32) against ground truth, 3D Photo [1], and Stereo-Mag [3]. The input frame closer to the target frame is marked in green for easier comparison.

Fig. 10: Extrapolation comparison of our method with 19 planes (Ours-19) against ground truth, 3D Photo [1], and Stereo-Mag [3]. The input frame closer to the target frame is marked in green for easier comparison.

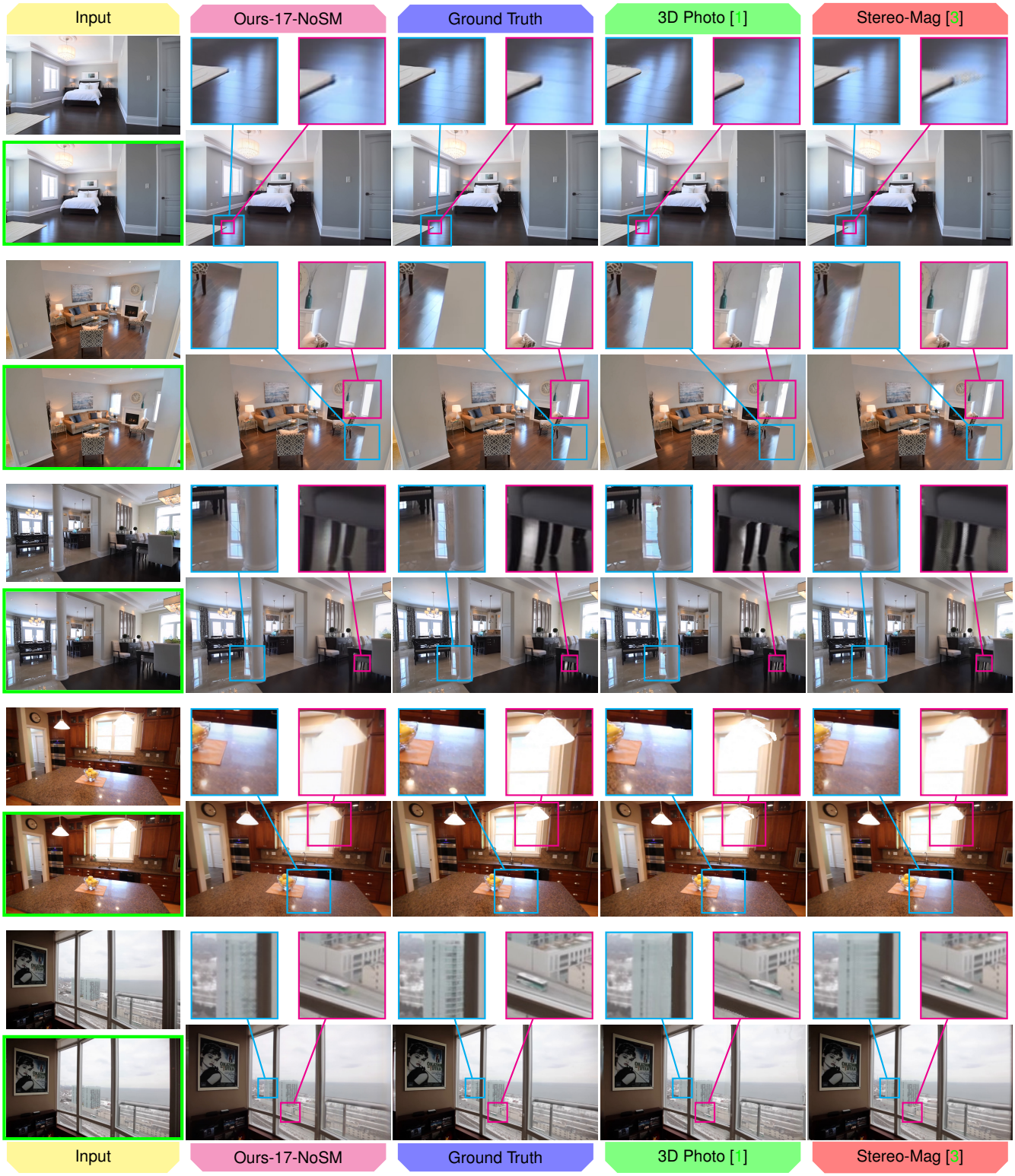

Fig. 11: Extrapolation comparison of our method ablation Ours-17-NoSM without soft masks against ground truth, 3D Photo [1], and Stereo-Mag [3]. The input frame closer to the target frame is marked in green for easier comparison.

Fig. 12: Interpolation using the fast Ours-19 network. The network runs inference in 576p while being trained in 288p. In every block, the top row (green) shows the ground truth trajectory from RealEstate10k, while the bottom row (blue) presents the interpolated result corresponding to the ground truth camera poses.

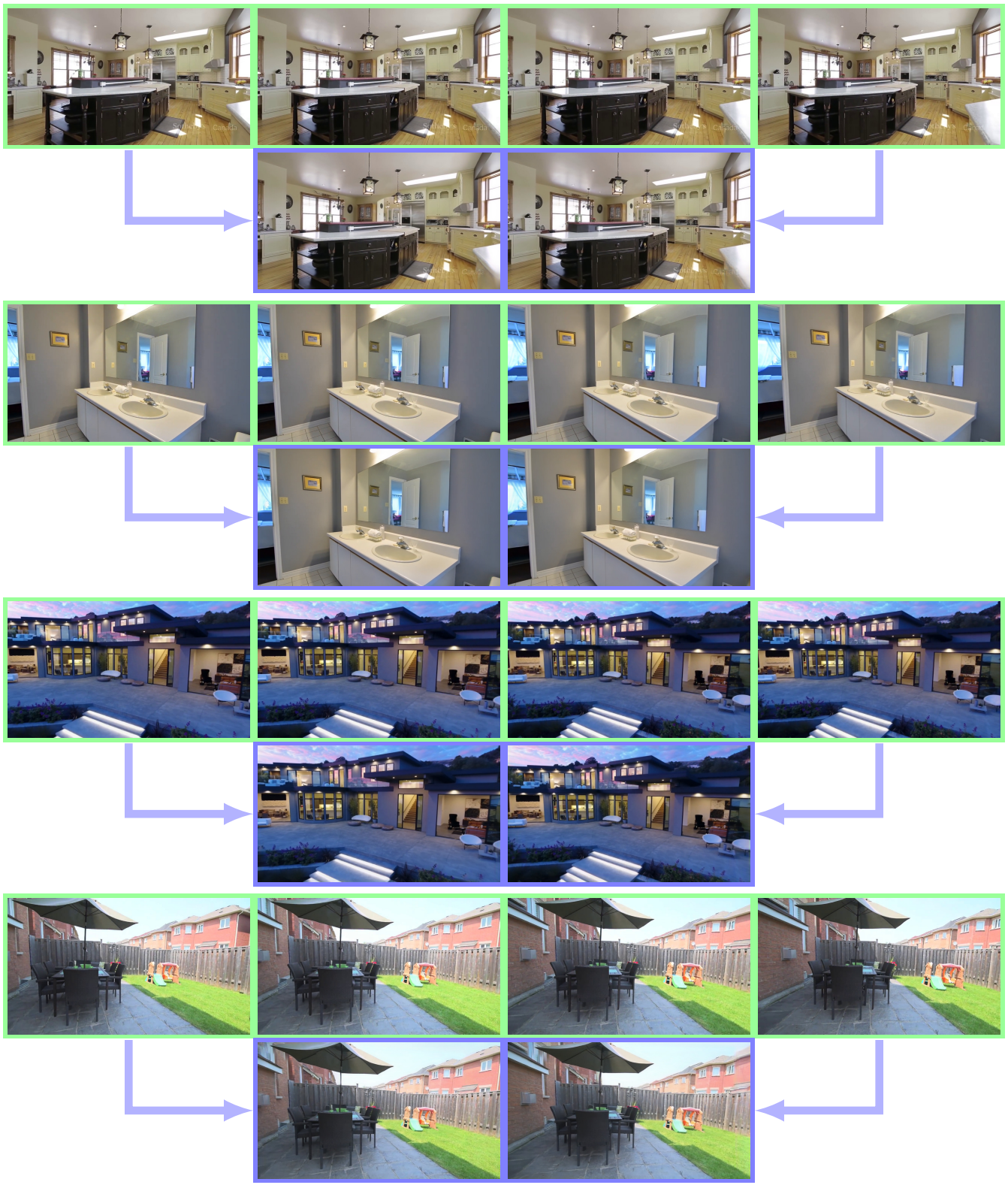

Fig. 13: Interpolation using the Ours-32 network. The network runs inference in 576p while being trained in 288p. In every block, the top row (green) shows the ground truth trajectory from RealEstate10k, while the bottom row (blue) presents the interpolated result corresponding to the ground truth camera poses.

REFERENCES

[1] 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 (CVPR)*, pages 8028–8038, 2020. 3, 5, 6, 7

[2] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics (T-RO)*, 33(5), 2017. 3

[3] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (TOG)*, 2018. 3, 5, 6, 7

[4] Nicolai Hani, Selim Engin, Jun-Jee Chao, and Volkan Isler. Continuous object representation networks: Novel view synthesis without target view supervision. *International Conference on Neural Information Processing Systems (NeurIPS)*, 33, 2020. 3