Progressive Refinement Imaging with Depth-Assisted Disparity Correction: *Supplementary Material*

1. Additional Results

1.1. Quantitative Comparison

We evaluate all methods on the synthetic data set *BunnySynth* by comparing the final, refined color and depth images (see Fig. 1 and Fig. 3) to the ground truth, i.e., the initial color and depth frame at four times the resolution (7680×4320 px).

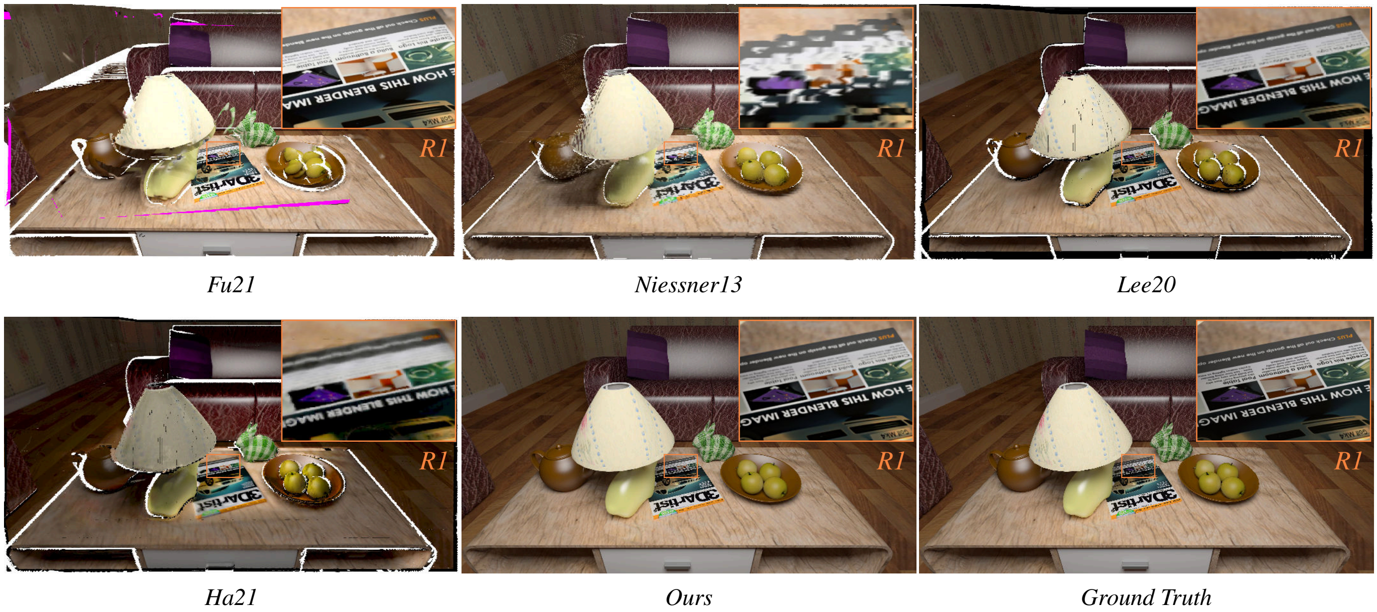
We report PSNR, the structural similarity SSIM [1] and the perceptual quality LPIPS [2] for the refined color in Tab. 1 and RMSE and MAE for the resulting depths in Tab. 2, revealing a significant advantage of our method. Furthermore, we show the error maps (per-pixel absolute error) for the refined color images compared to the ground truth in Fig. 2 and the absolute distance error [mm] for the corresponding depth maps in Fig. 4. Note that invalid (unknown) pixels were excluded in all error calculations for per-pixel metrics.

Table 1: Quantitative evaluation of the refined color for the synthetic data set *BunnySynth*. We report the average PSNR (dB) (higher is better) and SSIM [1] (higher is better) over the full image to evaluate the overall consistency to the ground truth as well as the average error over a selected region *R1* (see Figs. 1 and 2) to evaluate the amount of detail achieved in the refined image. To evaluate perceptual quality, we employ the LPIPS [2] score (lower is better), which uses deep features.

	Full image			Region *R1*		
	PSNR (dB) (↑)	SSIM (↑)	LPIPS (↓)	PSNR (dB) (↑)	SSIM (↑)	LPIPS (↓)
Fu21 [3]	17.30	0.76	0.50	9.82	0.33	0.44
Niessner13 [4]	19.59	0.87	0.41	11.23	0.44	0.64
Lee20 [5]	22.26	0.81	0.42	17.08	0.65	0.31
Ha21 [6]	15.22	0.66	0.58	16.07	0.46	0.51
Ours	29.50	0.96	0.16	18.32	0.73	0.18

Table 2: Quantitative evaluation of the resulting depths for the synthetic data set *BunnySynth*. We report the average RMSE (mm) (lower is better) and MAE (mm) (lower is better) over the full image to evaluate the overall consistency to the ground truth as well as the average error over a selected region *R2* (see Figs. 3 and 4) to evaluate the achieved accuracy at object silhouettes.

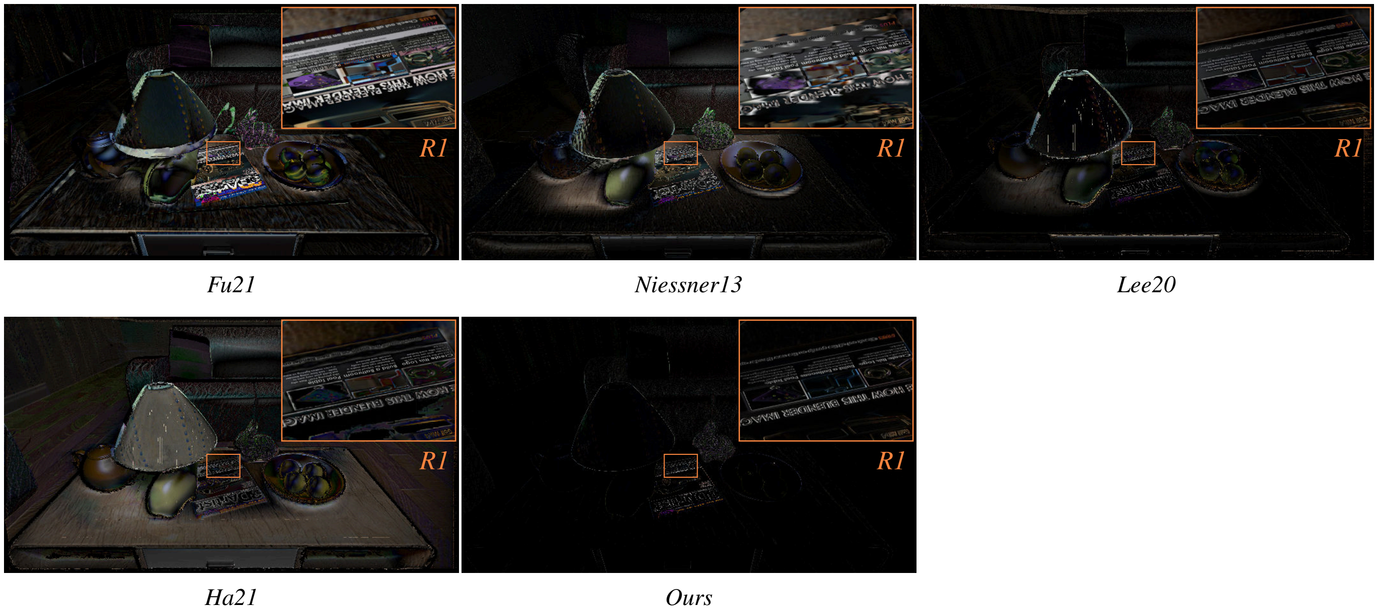
	Full image		Region *R2*	
	RMSE (mm) (↓)	MAE (mm) (↓)	RMSE (mm) (↓)	MAE (mm) (↓)
Fu21 [3]	101.59	10.40	194.93	41.07
Niessner13 [4]	87.92	7.79	181.99	36.12
Lee20 [5]	82.05	7.12	163.76	28.68
Ha21 [6]	80.60	7.03	146.53	23.57
Ours	65.22	3.54	69.56	5.66



Fu21 *Niessner13* *Lee20*

Ha21 *Ours* *Ground Truth*

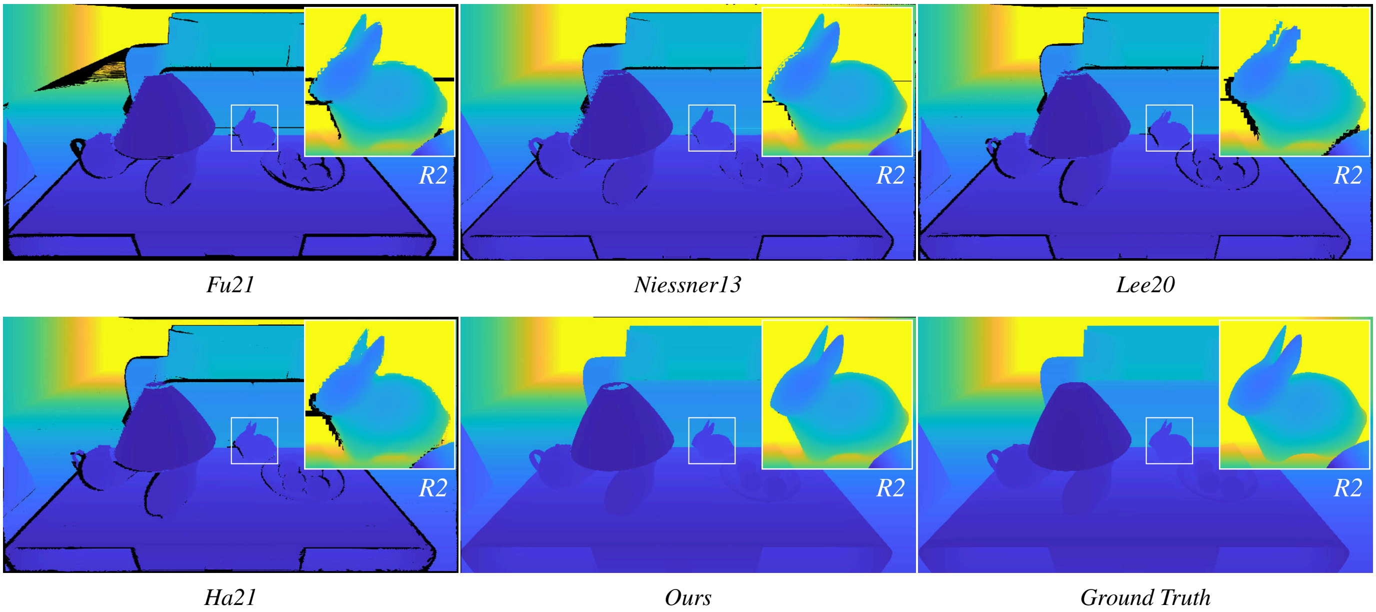
Fig. 1: Results (refined color) of the synthetic data set *BunnySynth*, used for the quantitative comparison in Tab. 1. See also Fig. 10 for a comparison with the initial, unrefined frame.



Fu21 *Niessner13* *Lee20*

Ha21 *Ours*

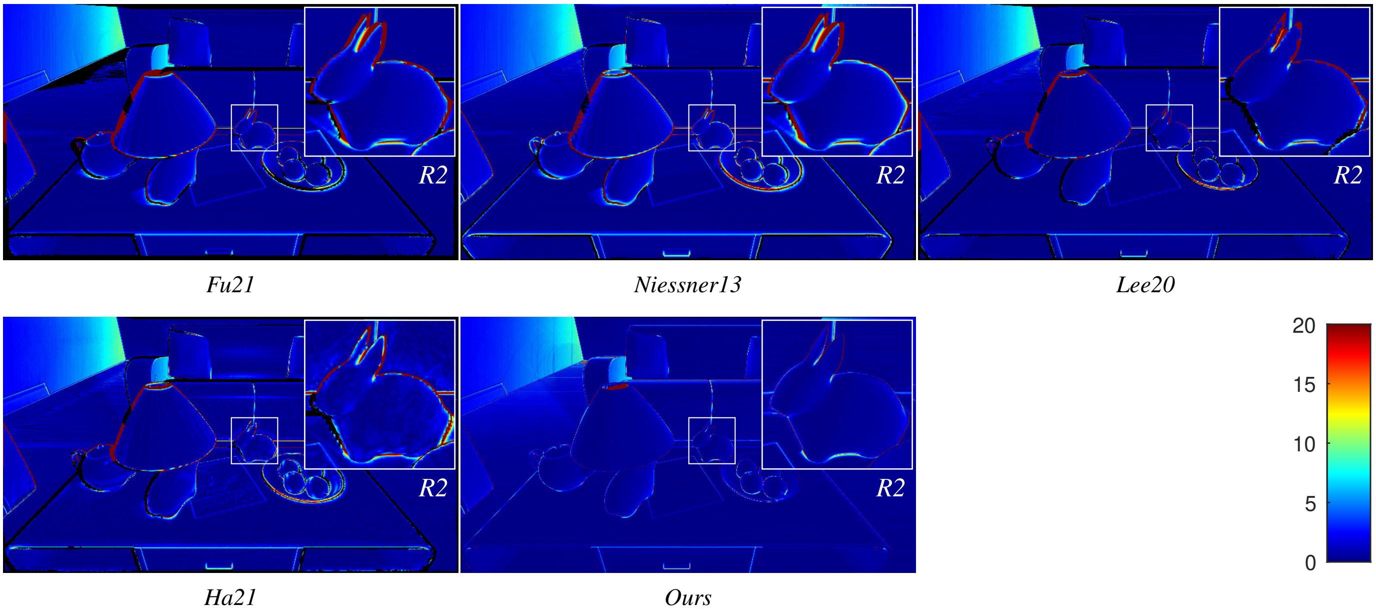
Fig. 2: Error maps (per-pixel absolute error) corresponding to the refined color images shown in Fig. 1.



Fu21 *Niessner13* *Lee20*

Ha21 *Ours* *Ground Truth*

Fig. 3: Results (refined depth) of the synthetic data set *BunnySynth*, used for the quantitative comparison in Tab. 2. Depth maps are shown using a *Parula* colormap ranging from 0.5 m to 4.5 m for the full image and from 0.85 m to 1.0 m for the inset.



Fu21 *Niessner13* *Lee20*

Ha21 *Ours*

Fig. 4: Absolute distance error [mm] corresponding to the resulting depth maps shown in Fig. 3.

1.2. Robustness to Illumination Changes

Fig. 5 and Fig. 6 demonstrate the robustness of our proposed pipeline against illumination changes and differences in white-balance or auto-exposure in the input footage by using a Laplacian pyramid merging based on [7].

Compared to a simple color merging on flat image representations (Fig. 5) or competing methods (Fig. 6), our reconstruction pipeline retains the base color of the initial reference image (Fig. 10) by exploiting a frequency-oriented color fusion and, thus, does not require local or global optimization for color harmonization.

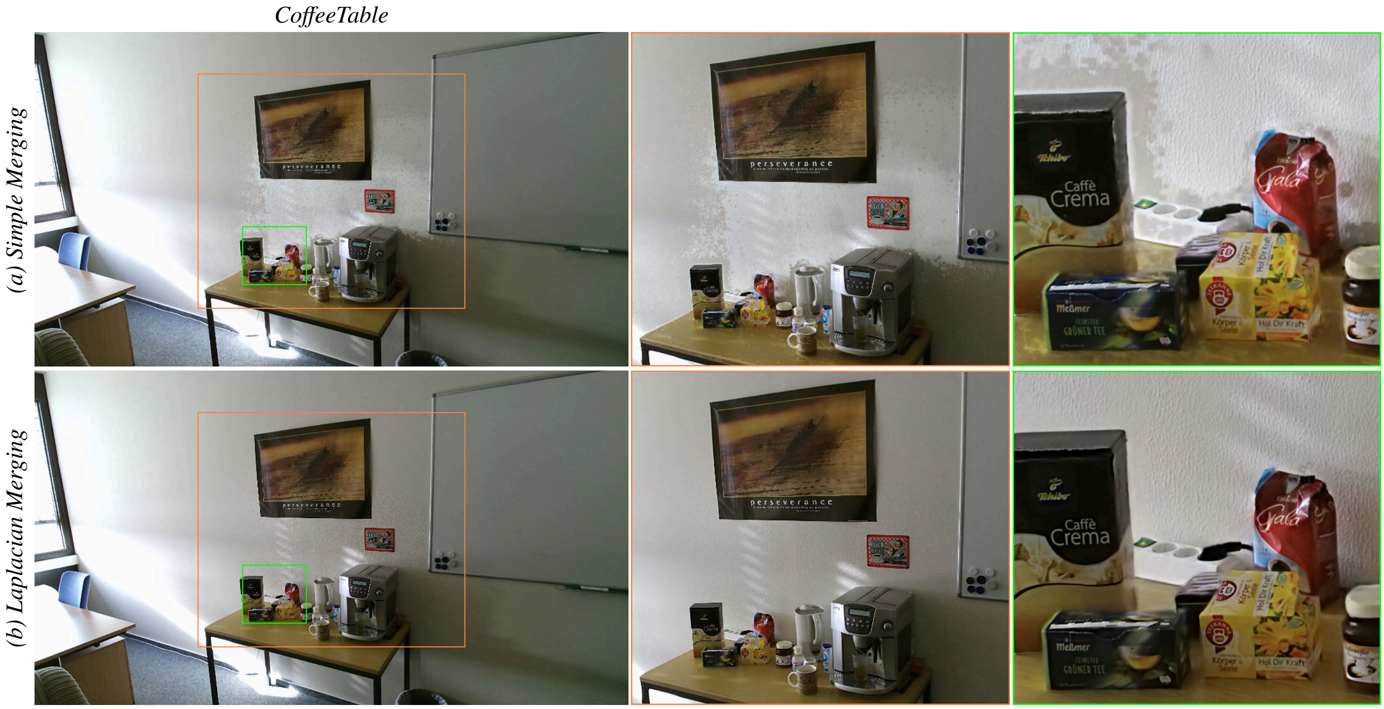


Fig. 5: Robustness to illumination changes. (a) Our approach combined with a simple color merging on flat image representations. (b) Ours with the Laplacian pyramid merging based on [7], as proposed in Sec. 4.7 in our paper.

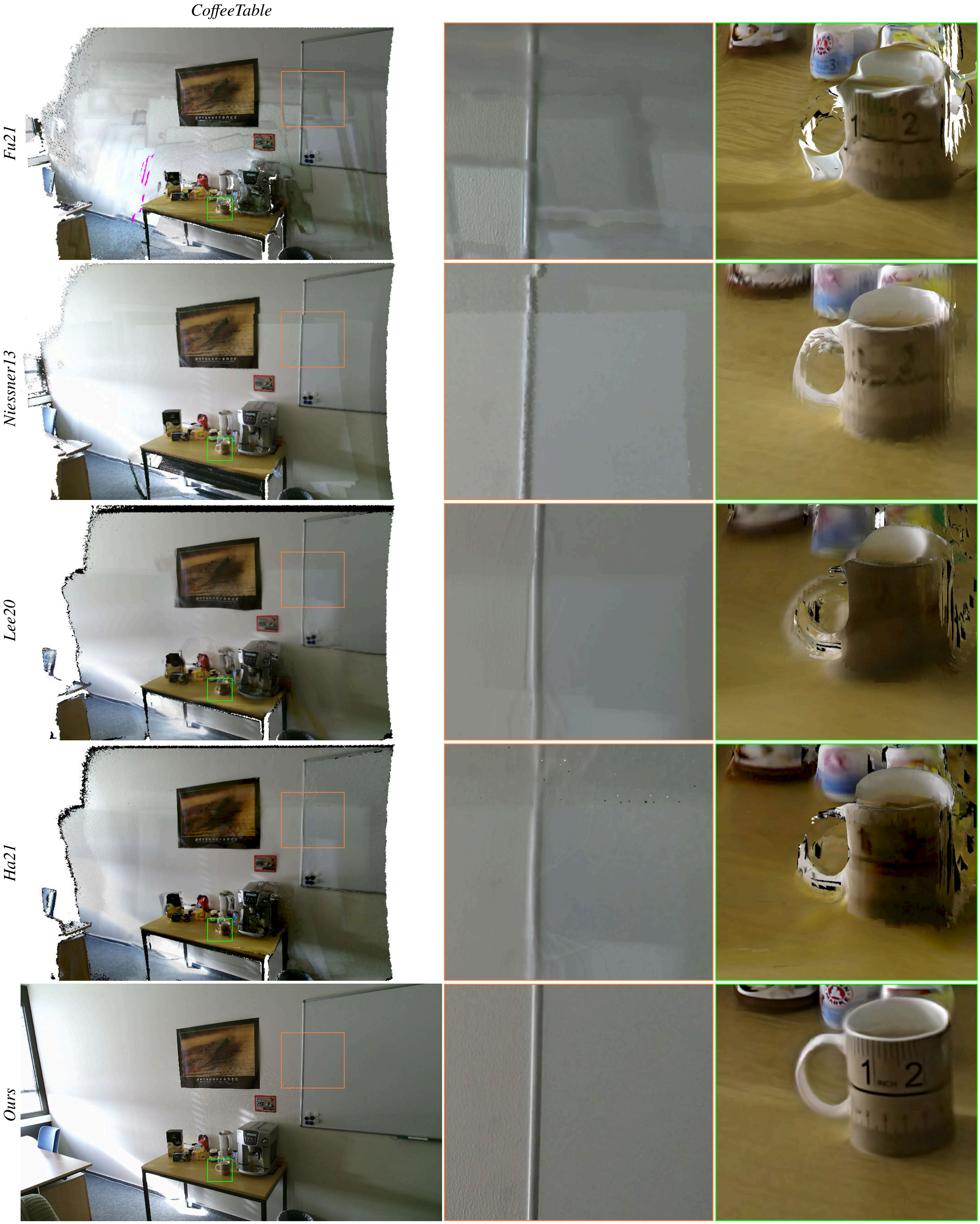


Fig. 6: Robustness to illumination changes. Comparison with the 3D scene reconstruction methods *Fu21* [3], *Niessner13* [4], *Lee20* [5] and *Ha21* [6]. See also Fig. 10 for a comparison with the unrefined reference image.

1.3. Comparison to 2D Image Reconstruction

Fig. 7 compares the results of our approach with the 2D image reconstruction methods *Kluge20* [7] and *Autopano* [8] (based on Brown et al.'s *AutoStitch* [9, 10]), which do not utilize the input depth maps to correct for parallaxes in the scenes. According to [7], *Autopano* is the only method available that is capable of fusing color images with a very high discrepancy in object-space resolution.

However, *Autopano* fails to correctly align the input footage even for data sets with low (*BrickWall*) to moderate (*Memorial*) amounts of disparity, leading to strong distortions and misalignments in the final results. While *Kluge20* works robustly on the *BrickWall* data set, however, for the *Memorial* data set, the limitations of the alignment using a homography lead to ghosting artifacts. Our method is able to reconstruct the silhouettes and captures more details than *Autopano* and *Kluge20*.

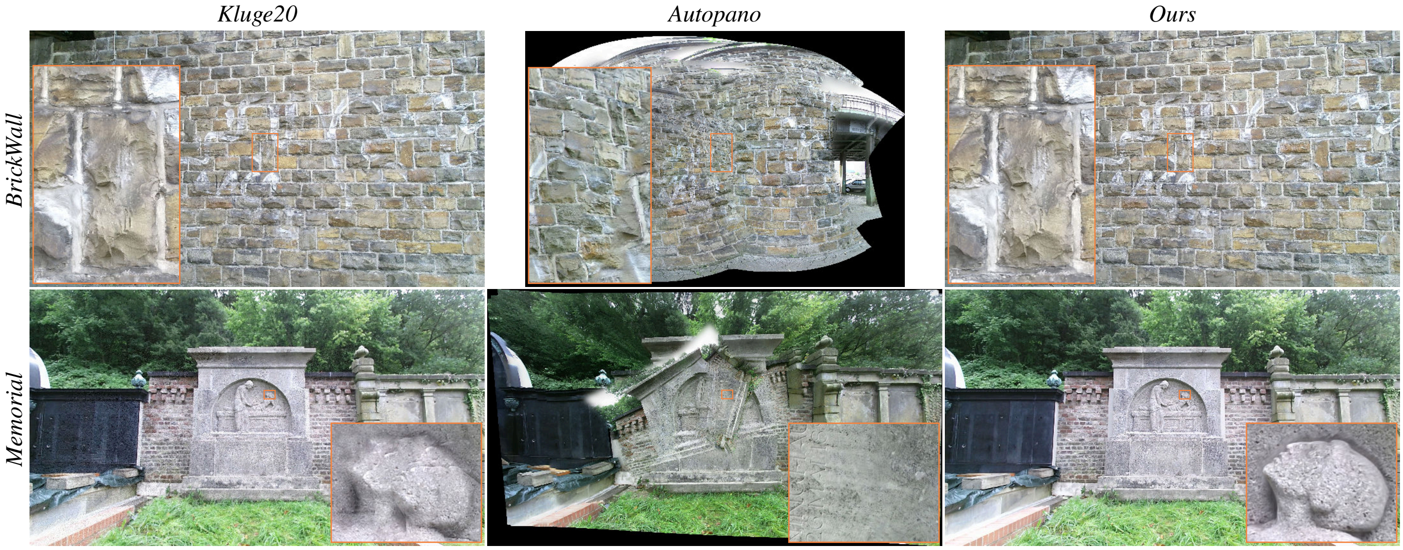


Fig. 7: Comparison with the 2D methods *Kluge20* [7] and *Autopano* [8]. See also Fig. 10 for a comparison with the unrefined reference image.

2. User Guidance

The current per-pixel level-of-refinement map can be visualized to guide the user during reconstruction to areas needing more refinement. Fig. 8 shows the final level-of-refinement map for each data set to visualize the amount of detail incorporated into our method's final reconstruction, as shown in Figs. 11 to 13 (right column).

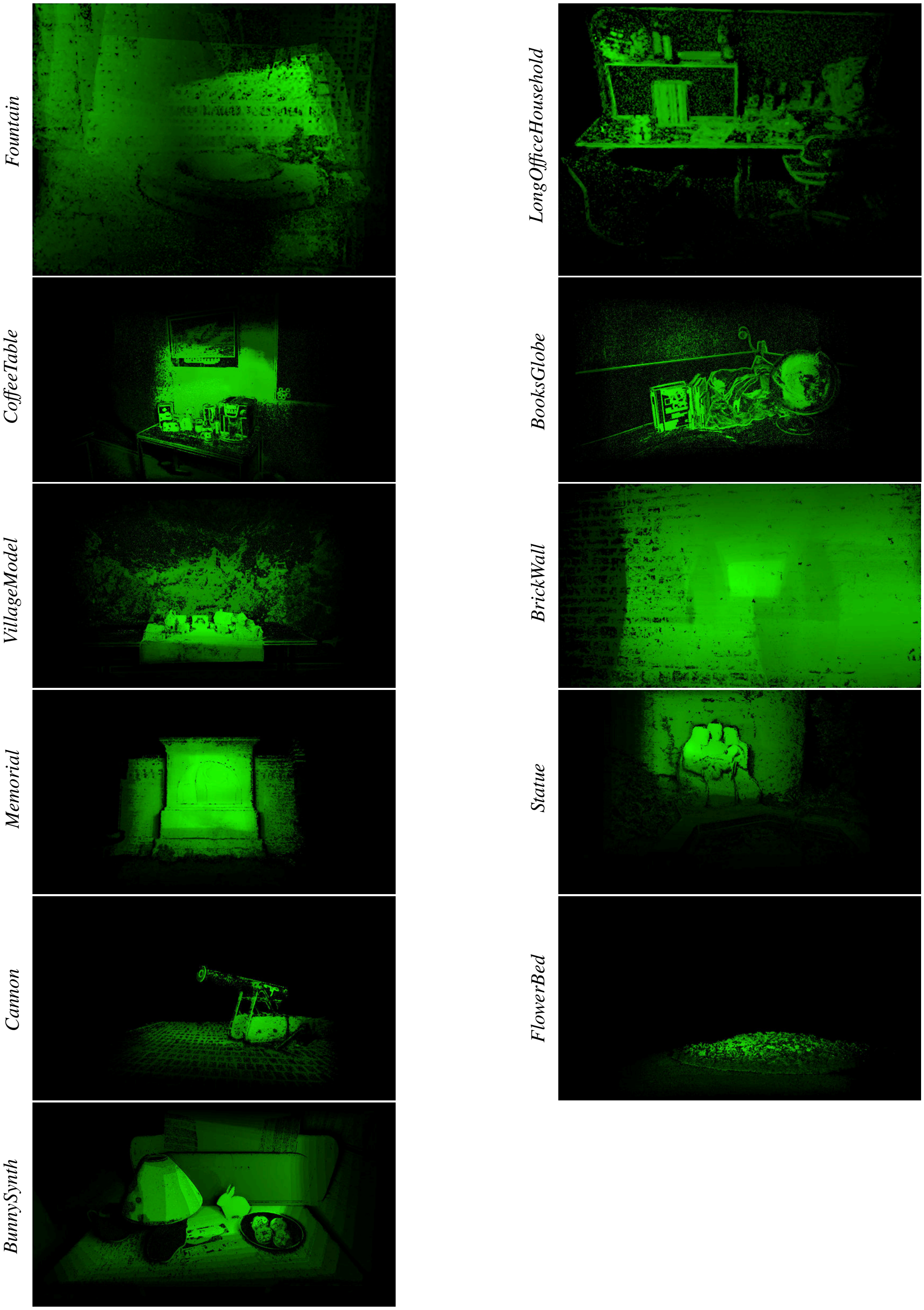


Fig. 8: The final level-of-refinement maps for each data set visualizing the amount of detail incorporated into the final reconstruction (ours). Brighter colors indicate a higher amount of incorporated details.

3. Progressive Voting Scheme for Depth Fusion

Fig. 9 shows an alternative visualization of our proposed voting scheme for depth fusion to demonstrate the impact of the resulting weighting compared to a cumulative average.

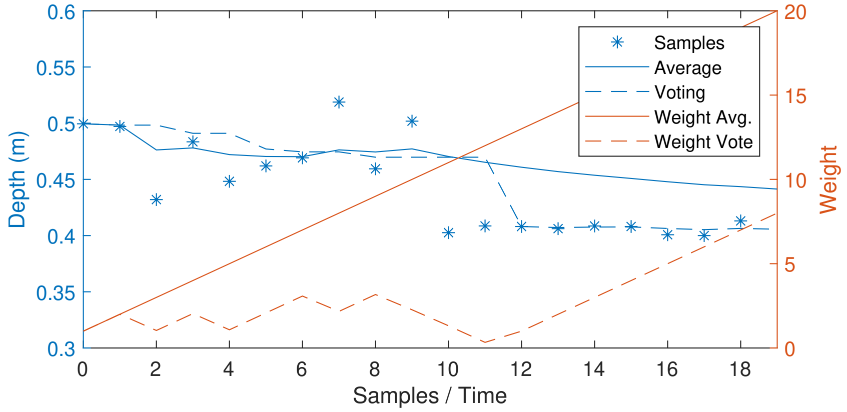


Fig. 9: Our proposed progressive voting scheme for depth fusion applied to a series of unreliable samples, followed by relatively stable samples of the real depth. While a *cumulative average* of samples slowly adapts to the new samples, our *progressive voting* quickly discards less reliable data in favor of a compatible value by adjusting the weighting. If too many new samples fail the compatibility test, i.e., the weight (counter) falls below 0, the depth is set to the new sample and the weight is reset to 1.

4. System Parameters

Tab. 3 shows the system parameters for *Niessner13* (VoxelHashing), *Lee20* (TextureFusion) and *Ha21* (NormalFusion) that had to be changed in order to successfully process the respective data set with 24 GB of GPU memory. Additionally, the internal resolution has been appropriately adjusted in the case of the Kinect v2, i.e., to $s\_adapterWidth = 1920$ and $s\_adapterHeight = 1080$ (pixels). For extensive outdoor scenery, the maximum distance ($s\_sensorDepthMax$ and $s\_SDFMaxIntegrationDistance$) has been increased to 5.0, in the case of *FlowerBed* and the synthetic data set *BunnySynth* to 6.5 (meters). The number of frames to process ($s\_nVideoFrame$) has been set to the total number of frames of the respective data set.

Table 3: Modified system parameters to be able to process the data sets. Parameters that differ from default are printed in italics. $s\_SDFVoxelSize$ is the voxel size (default: 0.004) in meters that was increased until the respective data set could be successfully processed with 24 GB of GPU memory. $s\_texPoolNumPatches$ is the minimum number of required pre-allocated texture tiles for the respective data set, whereas $s\_hashNumSDFBlocks$ is the number of required pre-allocated voxel blocks. The size of a texture tile was kept at the default value $s\_texPoolPatchWidth = 4$ (pixels) for all data sets.

	s_SDFVoxelSize			s_texPoolNumPatches		s_hashNumSDFBlocks		
	Niessner13	Lee20	Ha21	Lee20	Ha21	Niessner13	Lee20	Ha21
Fountain	0.004	0.004	0.004	*2 255 237*	*9 160 400*	*82 500*	*51 265*	*114 600*
CoffeeTable	0.004	0.004	*0.006*	*11 209 510*	*15 738 000*	*102 000*	*96 437*	*103 000*
BooksGlobe	0.004	0.004	0.004	*1 630 512*	*2 210 200*	*12 000*	*13 564*	*33 800*
VillageModel	0.004	0.004	*0.008*	*8 983 136*	*16 508 250*	*140 500*	*167 278*	*71 100*
BrickWall	0.004	*0.008*	*0.025*	*7 550 280*	*15 403 300*	*1 313 000*	*627 372*	*37 700*
Memorial	0.004	*0.005*	*0.009*	*8 644 457*	*16 724 500*	*430 000*	*549 396*	*217 500*
Statue	0.004	0.004	*0.010*	*11 691 649*	*15 288 000*	*293 500*	*295 264*	*60 550*
Cannon	0.004	0.004	*0.011*	*7 074 873*	*14 325 300*	*347 500*	*567 327*	*107 800*
FlowerBed	0.004	*0.006*	*0.014*	*7 018 407*	*14 478 000*	*555 000*	*586 510*	*100 700*
BunnySynth	0.004	0.004	0.004	*2 609 696*	*4 500 000*	*66 000*	*83 560*	*250 350*

5. Reference Images in High Resolution

Fig. 10 shows the unrefined reference images of each data set in high-resolution (full resolution).

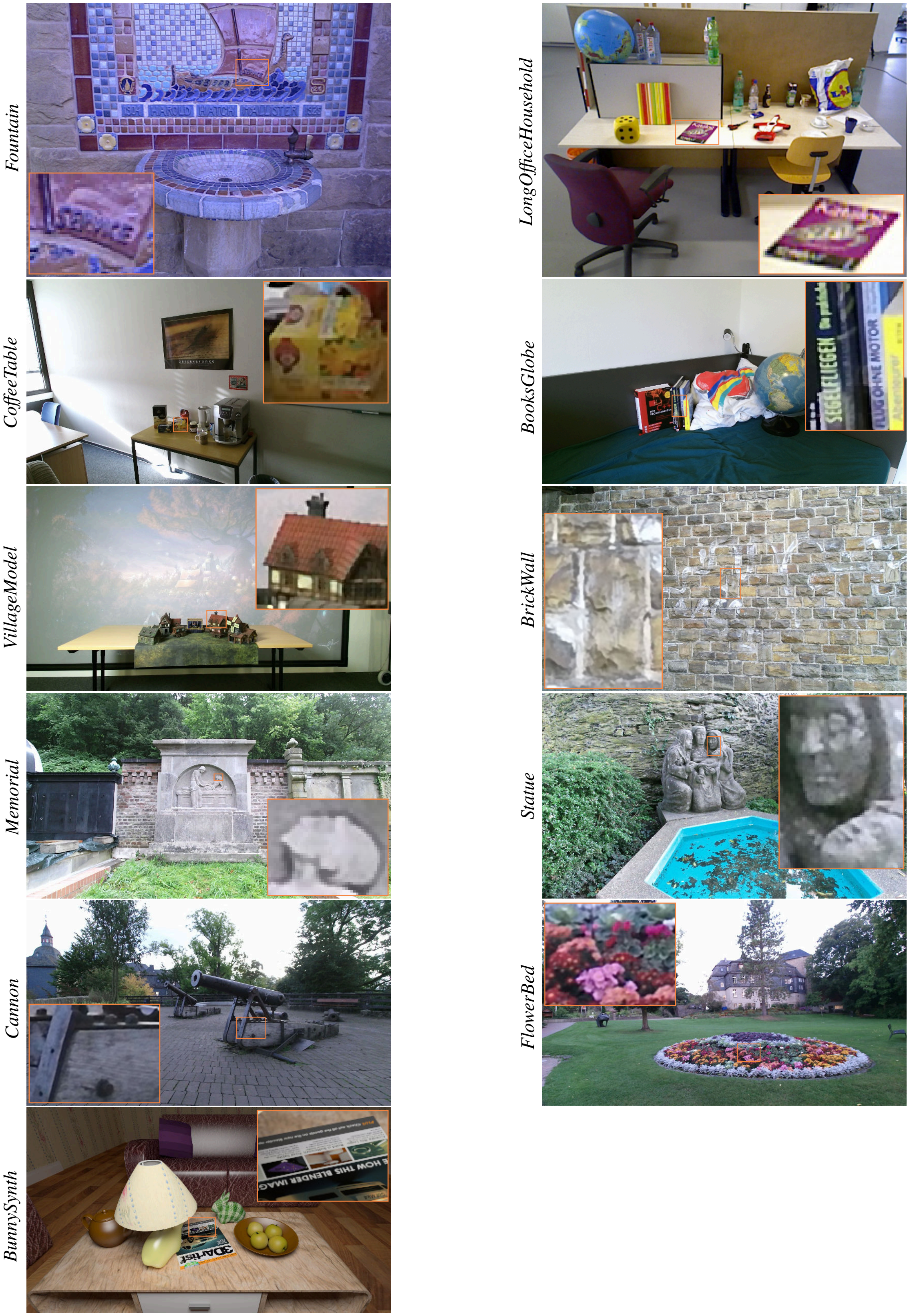


Fig. 10: The unrefined reference images (initial frames) of the data sets.

6. Results in High Resolution

Figs. 11 to 13 show the results for the competing methods and our approach in high-resolution (downscaled to 50%, close-ups in full resolution).

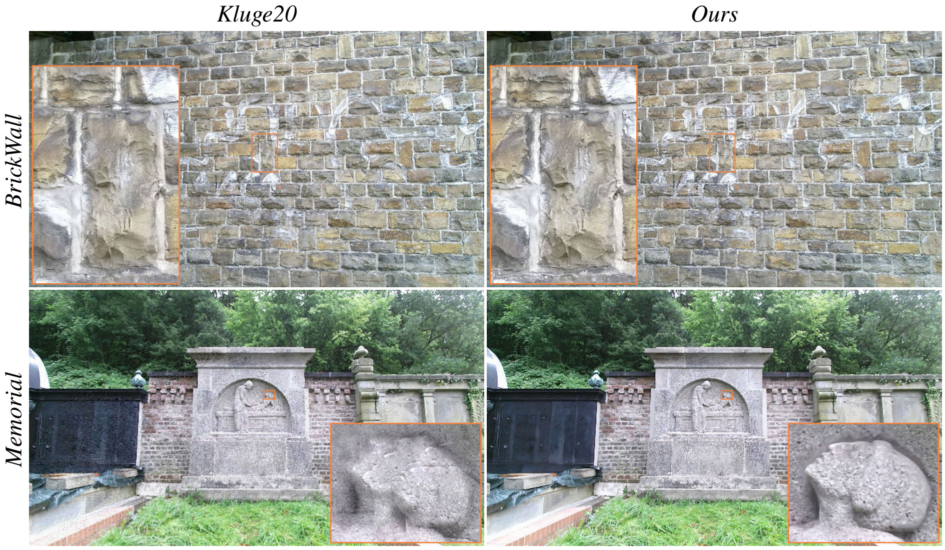


Fig. 11: Comparison with the 2D method *Kluge20* [7]. See also Fig. 10 for a comparison with the unrefined reference image.

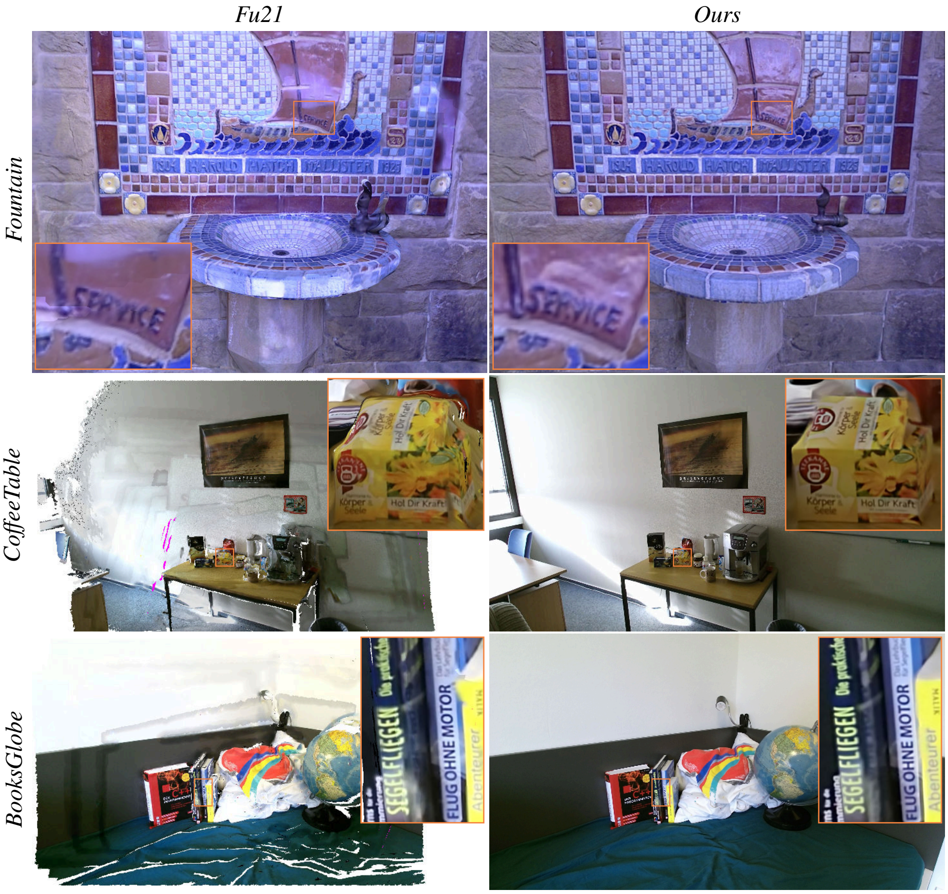
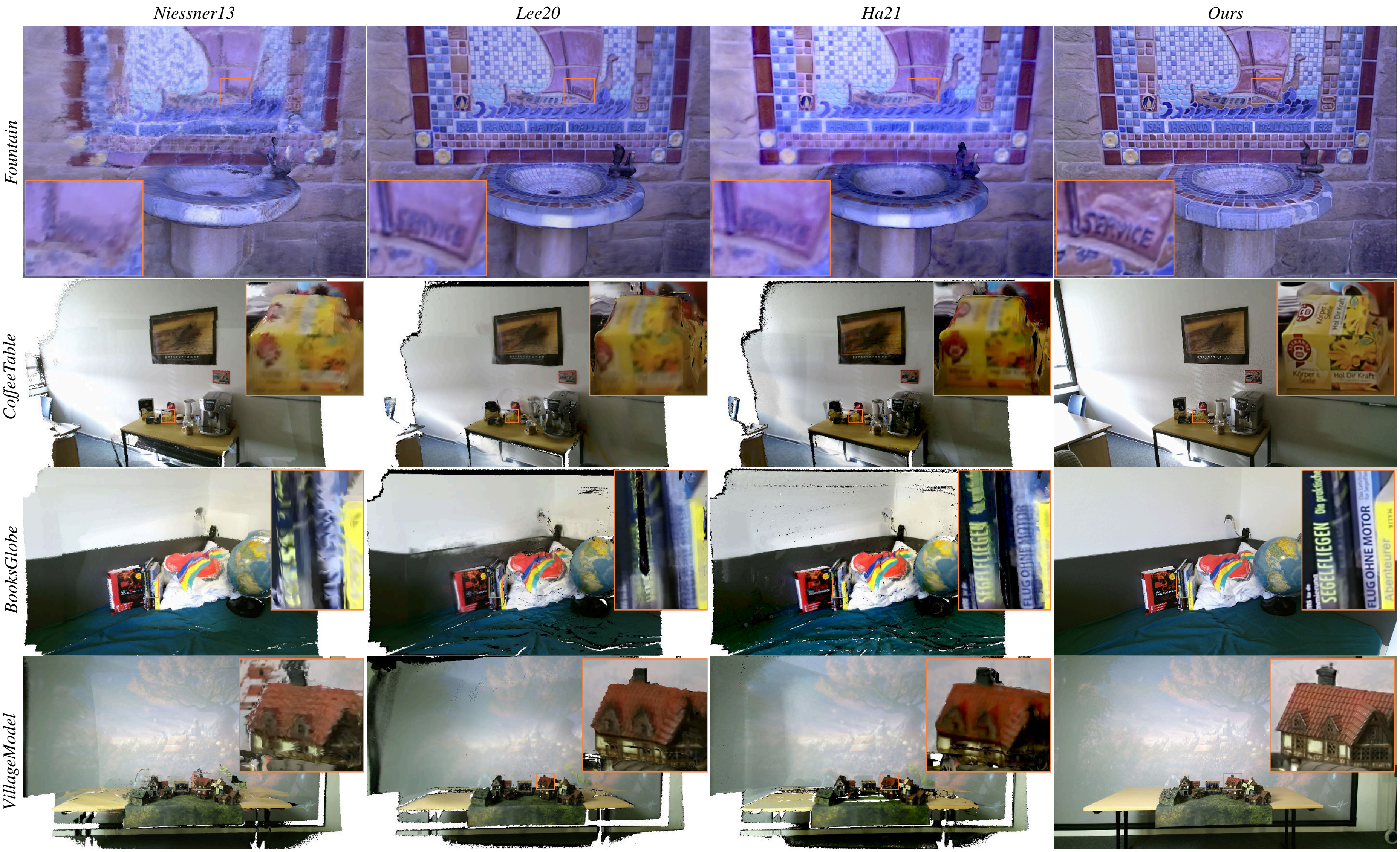


Fig. 12: Comparison with the offline, post-processing approach *Fu21* [3]. See also Fig. 10 for a comparison with the unrefined reference image.


Niessner13
Lee20
Ha21
Ours
Fountain
CoffeeTable
BooksGlobe
VillageModel

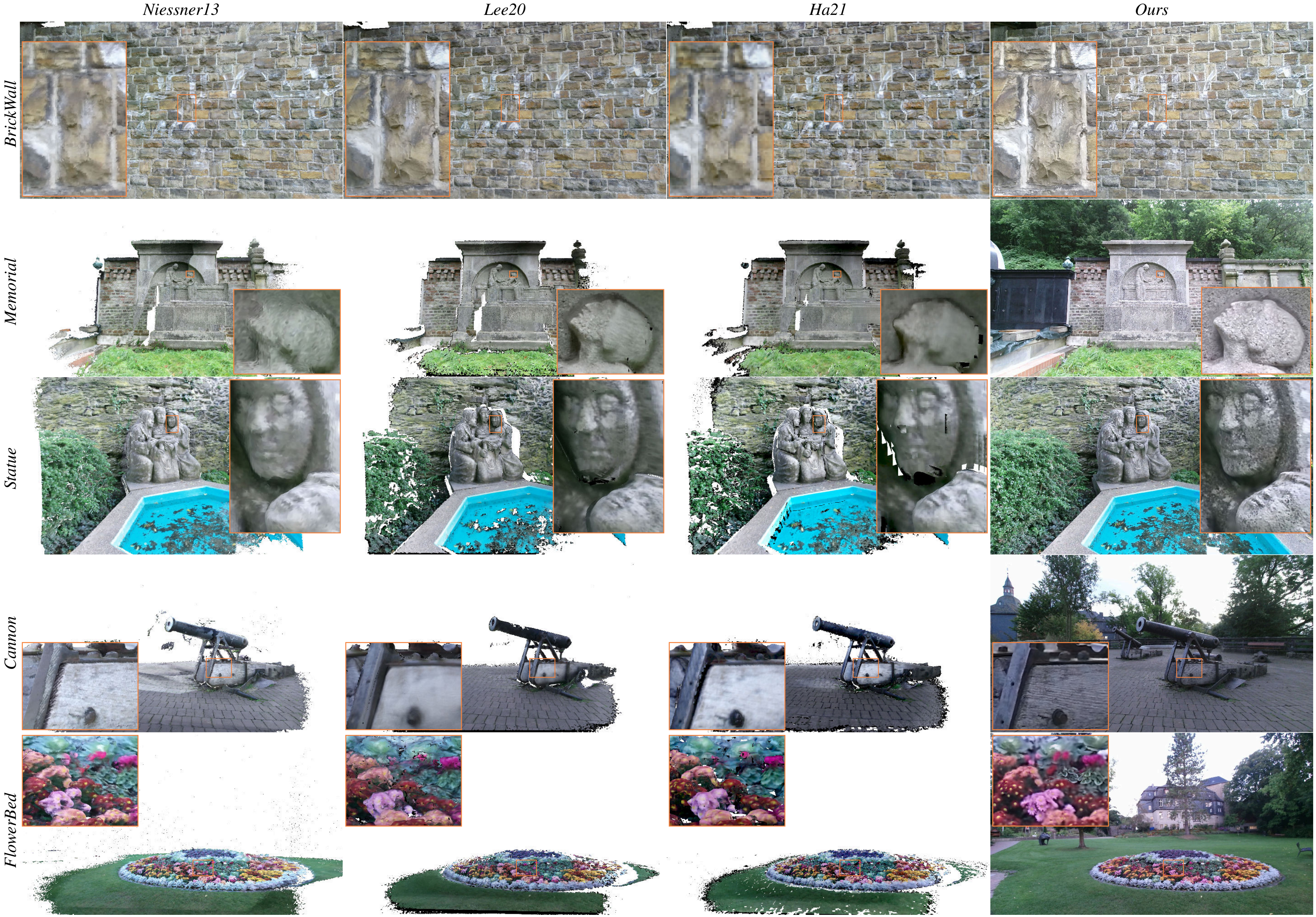


Fig. 13: Comparison with the online scene reconstruction methods *Niessner13* [4], *Lee20* [5] and *Ha21* [6]. See also Fig. 10 for a comparison with the unrefined reference image.

References

[1] Wang, Z, Bovik, AC, Sheikh, HR, Simoncelli, EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Processing (TIP) 2004;13(4):600–612.

[2] Zhang, R, Isola, P, Efros, AA, Shechtman, E, Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 586–595.

[3] Fu, Y, Yan, Q, Liao, J, Zhou, H, Tang, J, Xiao, C. Seamless texture optimization for rgb-d reconstruction. IEEE Transactions on Visualization and Computer Graphics 2021;.

[4] Nießner, M, Zollhöfer, M, Izadi, S, Stamminger, M. Real-time 3D reconstruction at scale using voxel hashing. ACM Trans Graphics 2013;32(6):169.

[5] Lee, JH, Ha, H, Dong, Y, Tong, X, Kim, MH. Texturefusion: High-quality texture acquisition for real-time rgb-d scanning. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). 2020, p. 1272–1280.

[6] Ha, H, Lee, JH, Meuleman, A, Kim, MH. Normalfusion: Real-time acquisition of surface normals for high-resolution rgb-d scanning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2021,.

[7] Kluge, M, Weyrich, T, Kolb, A. Progressive refinement imaging. Computer Graphics Forum 2020;39(1):360–374.

[8] Kolor, . Kolor autopano giga 4.4.2. Available from: `https://download.kolor.com/apg/stable/history`; 2018. [Accessed Oct. 18th 2022].

[9] Brown, M. Autostitch 3.0. Available from: `http://matthewalunbrown.com/autostitch/autostitch.html`; 2018. [Accessed Oct. 18th 2022.].

[10] Brown, M, Lowe, DG. Automatic panoramic image stitching using invariant features. Int Journal of Computer Vision (IJCV) 2007;74(1):59–73.