

Neural RGB-D Surface Reconstruction

– Supplemental Document –

Dejan Azinović¹ Ricardo Martin-Brualla² Dan B Goldman² Matthias Nießner¹ Justus Thies^{1,3}

¹Technical University of Munich ²Google Research ³Max Planck Institute for Intelligent Systems

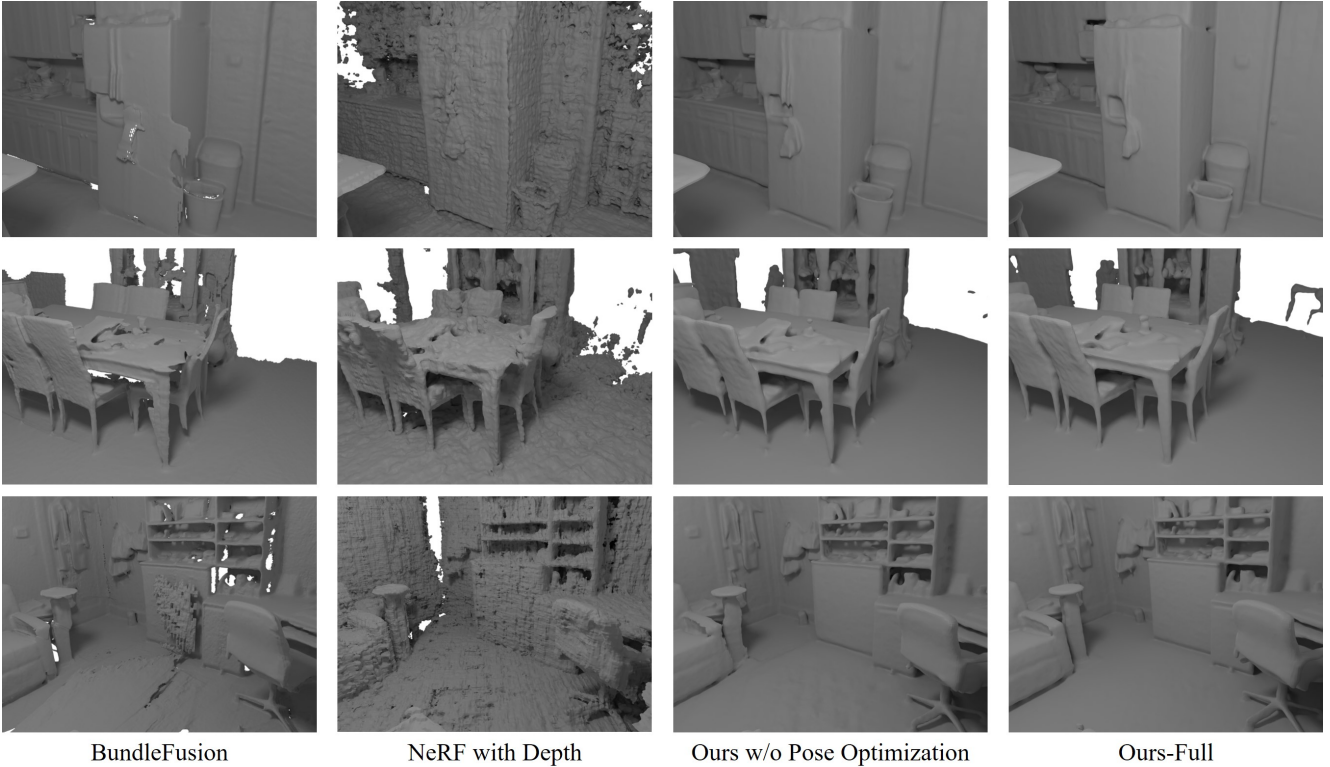


Figure 1. Our method obtains a high-quality 3D reconstruction from an RGB-D input sequence by training a multi-layer perceptron. In comparison to state-of-the-art methods like BundleFusion [3] or the theoretical NeRF [9] with additional depth constraints, our approach results in cleaner and more complete reconstructions. As can be seen, the pose optimization of our approach is key to resolving misalignment artifacts.

1. Implementation Details

We implement our method in TensorFlow v2.4.1 using the ADAM [7] optimizer with a learning rate of 5×10^{-4} and an exponential learning rate decay of 10^{-1} over 2.5×10^5 iterations. In each iteration, we compute a gradient w.r.t. $|P_b| = 1024$ randomly chosen rays. We set the number of S'_f samples to 16. S'_c is chosen so that there is on average one sample for every 1.5 cm of the ray length. Tab. 1 gives an overview of ray length and number of samples for

each of the experiments. Internally, we translate and scale each scene so that it lies within a $[-1, 1]^3$ cube. Depending on scene size, our method takes between 9 and 13 hours to converge on a single NVIDIA RTX 3090 (see Sec. 6). We set the loss weights to $\lambda_1 = 0.1$, $\lambda_2 = 10$ and $\lambda_3 = 6 \times 10^3$. We use 8 bands for the positional encoding of the point coordinates and 4 bands to encode the view direction vector.

To account for distortions or inaccuracies of the intrinsic parameters, a 2D deformation field of the camera pixel space in form of a 6-layer MLP, with a width of 128, is used.

Scene	S'_c	ray length (m)	#frames
Scene 0	512	8	1394
Scene 2	256	4	1299
Scene 5	256	4	1159
Scene 12	320	5	1335
Scene 24	512	8	849
Scene 50	256	4	1163
Scene 54	256	4	1250
Breakfast room	320	5	1167
Green room	512	8	1442
Grey-white room	512	8	1493
ICL living room	320	5	1510
Kitchen 1	512	8	1517
Kitchen 2	640	10	1221
Morning apartment	256	4	920
Staircase	512	8	1149
Thin geometry	256	4	395
White room	512	8	1676

Table 1. We list the number of samples S'_c and the ray length in meters that were used to reconstruct each of the ScanNet scenes and the synthetic scenes. Note that these settings are dependent on the scene size.

2. Per-scene Quantitative Evaluations

In Tab. 3 and Tab. 4 we present a per-scene breakdown of the quantitative analysis from the main paper (see Sec. 4, Tab. 1 and Tab. 2 in the main paper). The corresponding qualitative results are shown in Fig. 9 and Fig. 10.

Reconstruction Evaluation. The goal of our method is to reconstruct a scene from color and depth data, i.e., we do not aim for scene completion. To evaluate the reconstruction quality, we evaluate the quality of reconstructions w.r.t. Chamfer distance ($C-\ell_1$), intersection-over-union (IoU), normal consistency (NC) based on cosine similarity, and F-score. These metrics are computed on surfaces which were visible in the color and depth streams (geometry within the viewing frusta of the input images). Specifically, we subdivide all meshes to have a maximum edge length of below 1.5 cm and use the ground truth trajectory to detect vertices which are visible in at least one camera. Triangles which have no visible vertices, either due to not being in any of the viewing frusta or due to being occluded by other geometry, are culled. This is necessary to avoid computing the error in regions such as occluded geometry in the synthetic ground truth mesh or in regions where the network output is unpredictable because the region was never seen at training time. The culled geometry is sampled with a density of 1 point per cm^2 and the error metrics are evaluated on the

Scene	URL	License
ScanNet	http://www.scan-net.org/	MIT
Breakfast room	https://blendswap.com/blend/13363	CC-BY
Green room	https://blendswap.com/blend/8381	CC-BY
Grey-white room	https://blendswap.com/blend/13552	CC-BY
ICL living room	https://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnuim.html	CC-BY
Kitchen 1	https://blendswap.com/blend/5156	CC-BY
Kitchen 2	https://blendswap.com/blend/11801	CC-0
Morning apart.	https://blendswap.com/blend/10350	CC-0
Staircase	https://blendswap.com/blend/14449	CC-BY
Thin geometry	https://blendswap.com/blend/8381	CC-BY
White room	https://blendswap.com/blend/5014	CC-BY

Table 2. Source and license information of the used data.

sampled point clouds. To evaluate the IoU, we voxelize the reconstruction using voxels with an edge length of 5 cm. The F-score is also computed using a 5 cm threshold.

Synthetic Dataset. Our synthetic dataset which we use for numeric evaluation purposes consists of 10 scenes published under either the CC-BY or CC-0 license (see Tab. 2). We define a trajectory by a Catmull-Rom spline interpolation [12] on several manually chosen control points. We use BlenderProc [4] to render color and depth images for each camera pose in the interpolated trajectory. Noise is applied to the depth maps to simulate sensor noise of a real depth sensor [1, 2, 5, 6]. For the ICL scene [6], we use the color and noisy depth provided by the authors and do not render our own images. The scenes in the dataset have various sizes, complexity and materials like highly specular surfaces or mirrors. BundleFusion [3] is used to get an initial estimate of the camera trajectory. This estimated trajectory is used by all methods other than COLMAP to allow a fair comparison.

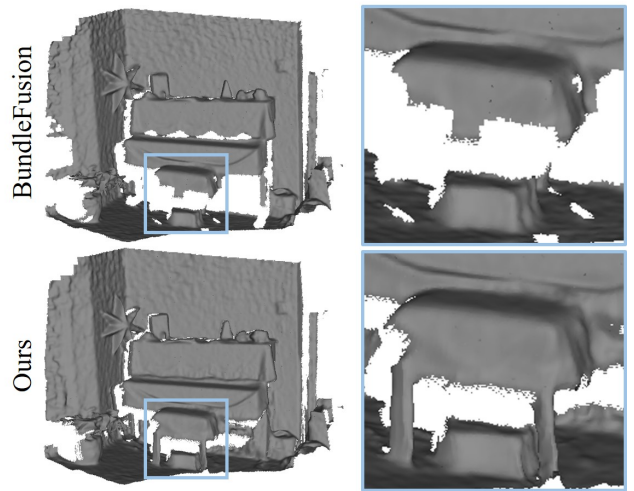


Figure 2. The photometric energy term encourages correct depth prediction in areas where the depth sensor did not capture any depth measurements.






Scene	Method	C- ℓ_1 ↓	IoU ↑	NC ↑	F-score ↑	Pos. error ↓	Rot. error ↓
Breakfast room 	BundleFusion	0.033	0.698	0.944	0.890	0.037	0.697
	RoutedFusion	0.033	0.714	0.918	0.901	-	-
	COLMAP + Poisson	0.033	0.668	0.935	0.893	0.009	0.210
	Conv. Occ. Nets	0.047	0.474	0.879	0.780	-	-
	SIREN	0.060	0.566	0.922	0.822	-	-
	NeRF + Depth	0.041	0.619	0.811	0.854	-	-
	Ours (w/o pose)	0.031	0.720	0.930	0.914	-	-
	Ours	0.030	0.793	0.934	0.920	0.007	0.135
Green room 	BundleFusion	0.024	0.694	0.923	0.926	0.027	0.546
	RoutedFusion	0.018	0.755	0.904	0.969	-	-
	COLMAP + Poisson	0.018	0.849	0.925	0.967	0.014	0.227
	Conv. Occ. Nets	0.053	0.554	0.855	0.737	-	-
	SIREN	0.023	0.746	0.913	0.940	-	-
	NeRF + Depth	0.030	0.668	0.748	0.871	-	-
	Ours (w/o pose)	0.014	0.766	0.931	0.982	-	-
	Ours	0.013	0.921	0.932	0.990	0.012	0.104
Grey-white room 	BundleFusion	0.038	0.567	0.860	0.751	0.056	1.891
	RoutedFusion	0.033	0.606	0.850	0.790	-	-
	COLMAP + Poisson	0.029	0.727	0.899	0.899	0.029	0.296
	Conv. Occ. Nets	0.048	0.480	0.841	0.601	-	-
	SIREN	0.033	0.635	0.868	0.812	-	-
	NeRF + Depth	0.040	0.563	0.764	0.697	-	-
	Ours (w/o pose)	0.032	0.640	0.864	0.806	-	-
	Ours	0.015	0.886	0.924	0.987	0.014	0.146
ICL living room 	BundleFusion	0.018	0.743	0.956	0.958	0.022	0.382
	RoutedFusion	0.019	0.698	0.939	0.976	-	-
	COLMAP + Poisson	0.023	0.727	0.947	0.966	0.029	0.836
	Conv. Occ. Nets	0.112	0.352	0.841	0.507	-	-
	SIREN	0.020	0.768	0.950	0.967	-	-
	NeRF + Depth	0.021	0.689	0.900	0.956	-	-
	Ours (w/o pose)	0.014	0.790	0.964	0.992	-	-
	Ours	0.011	0.905	0.969	0.994	0.007	0.109
Kitchen 1 	BundleFusion	0.234	0.368	0.860	0.620	0.038	0.327
	RoutedFusion	0.265	0.401	0.805	0.680	-	-
	COLMAP + Poisson	0.252	0.459	0.888	0.748	0.103	0.941
	Conv. Occ. Nets	0.262	0.352	0.839	0.483	-	-
	SIREN	0.265	0.357	0.850	0.575	-	-
	NeRF + Depth	0.271	0.336	0.710	0.600	-	-
	Ours (w/o pose)	0.255	0.420	0.887	0.700	-	-
	Ours	0.252	0.447	0.886	0.718	0.030	0.114

Table 3. We compare the quality of our reconstruction on several synthetic scenes for which ground truth data is available. The Chamfer ℓ_1 distance, normal consistency and the F-score [8] are computed between point clouds sampled with a density of 1 point per cm^2 . We use a threshold of 5 cm for the F-score. We further voxelize each mesh to compute the intersection-over-union (IoU) between the predictions and ground truth.


Scene	Method	C- ℓ_1 ↓	IoU ↑	NC ↑	F-score ↑	Pos. error ↓	Rot. error ↓
Kitchen 2 	BundleFusion	0.089	0.441	0.856	0.687	0.050	0.566
	RoutedFusion	0.059	0.572	0.842	0.787	-	-
	COLMAP + Poisson	0.037	0.675	0.919	0.818	0.043	1.154
	Conv. Occ. Nets	0.052	0.484	0.861	0.653	-	-
	SIREN	0.055	0.453	0.898	0.735	-	-
	NeRF + Depth	0.051	0.435	0.708	0.630	-	-
	Ours (w/o pose)	0.034	0.488	0.908	0.796	-	-
	Ours	0.032	0.637	0.903	0.890	0.083	0.450
Morning apartment 	BundleFusion	0.012	0.767	0.885	0.968	0.008	0.165
	RoutedFusion	0.013	0.815	0.870	0.976	-	-
	COLMAP + Poisson	0.017	0.668	0.877	0.959	0.017	0.380
	Conv. Occ. Nets	0.045	0.450	0.802	0.784	-	-
	SIREN	0.013	0.727	0.873	0.966	-	-
	NeRF + Depth	0.022	0.587	0.838	0.975	-	-
	Ours (w/o pose)	0.011	0.787	0.887	0.983	-	-
	Ours	0.011	0.716	0.888	0.982	0.005	0.093
Staircase 	BundleFusion	0.091	0.373	0.860	0.623	0.039	0.643
	RoutedFusion	0.069	0.340	0.864	0.622	-	-
	COLMAP + Poisson	0.074	0.322	0.895	0.628	0.043	0.305
	Conv. Occ. Nets	0.069	0.315	0.838	0.508	-	-
	SIREN	0.067	0.432	0.885	0.676	-	-
	NeRF + Depth	0.087	0.396	0.644	0.624	-	-
	Ours (w/o pose)	0.057	0.457	0.899	0.704	-	-
	Ours	0.045	0.565	0.920	0.853	0.016	0.123
Thin geometry 	BundleFusion	0.019	0.764	0.909	0.922	0.009	0.126
	RoutedFusion	0.023	0.708	0.829	0.881	-	-
	COLMAP + Poisson	0.047	0.440	0.820	0.721	0.079	2.400
	Conv. Occ. Nets	0.022	0.723	0.882	0.910	-	-
	SIREN	0.021	0.733	0.887	0.913	-	-
	NeRF + Depth	0.014	0.825	0.847	0.989	-	-
	Ours (w/o pose)	0.009	0.857	0.911	0.995	-	-
	Ours	0.009	0.865	0.910	0.995	0.010	0.037
White room 	BundleFusion	0.062	0.528	0.869	0.701	0.045	0.375
	RoutedFusion	0.038	0.545	0.817	0.799	-	-
	COLMAP + Poisson	0.036	0.652	0.904	0.796	0.018	0.167
	Conv. Occ. Nets	0.061	0.424	0.853	0.470	-	-
	SIREN	0.046	0.617	0.888	0.752	-	-
	NeRF + Depth	0.073	0.385	0.716	0.619	-	-
	Ours (w/o pose)	0.034	0.631	0.902	0.813	-	-
	Ours	0.028	0.738	0.911	0.915	0.028	0.133

Table 4. We compare the quality of our reconstruction on several synthetic scenes for which ground truth data is available. The Chamfer ℓ_1 distance, normal consistency and the F-score [8] are computed between point clouds sampled with a density of 1 point per cm^2 . We use a threshold of 5 cm for the F-score. We further voxelize each mesh to compute the intersection-over-union (IoU) between the predictions and ground truth.

Method	$C\text{-}\ell_1 \downarrow$	IoU \uparrow	NC \uparrow	F-score \uparrow
Ours (depth-only)	0.017	0.791	0.910	0.944
Ours (full)	0.009	0.865	0.910	0.995

Table 5. Detailed reconstruction results for Fig. 4 from the main paper. Our method reconstructs geometry visible only in color images, leading to significantly better reconstruction results in scenes with geometry which is not captured by the depth sensor.

3. Ablation Studies

In this section, we present additional details for the ablation studies described in the main paper, and show further studies to test the robustness and the limitations of our method. In Fig. 1, the additional results on real data demonstrate the advantages of the signed distance field and our camera refinement.

3.1. Effect of the Photometric Energy Term

In Tab. 5, we list the quantitative evaluation of the experiment on the effectiveness of the photometric energy term from Fig. 4 in the main paper. Fig. 2 shows the effect of the term on a real scene from the ScanNet dataset. The legs of the piano stool were not visible in any of the depth maps. Nevertheless, our method is able to reconstruct them by making use of the corresponding color data.

3.2. Number of Input Frames

The reconstruction quality of any reconstruction method is dependent on the number of input frames. We evaluate our method on the ‘whiteroom’ synthetic scene through multiple experiments in which we remove different numbers of frames in the dataset used for optimization. Reconstruction results are presented in Fig. 3. Note that for these experiments we use the camera poses initialized with BundleFusion which uses all 1676 depth frames.

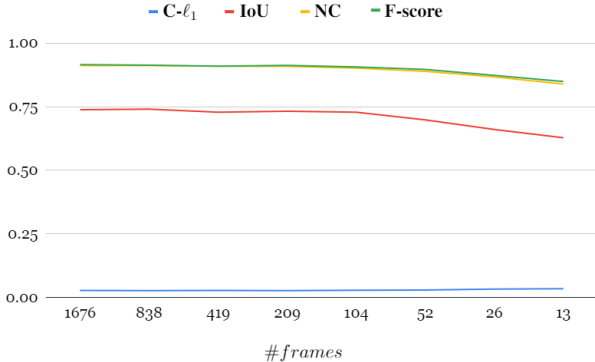


Figure 3. We test the robustness of our method by removing frames from the dataset used for optimization. Our method achieves good reconstruction results using as few as 13 frames.

3.3. Robustness to Noisy Pose Initialization

To analyze the robustness of our method w.r.t. presence of inaccuracies in camera alignment, we apply Gaussian noise to every camera’s position and direction in the ‘white-room’ scene. In Fig. 4 we present reconstruction results for poses of increasing inaccuracy. We separately show the pose errors of the refined cameras in Fig. 5. On the reconstruction metrics, our method is robust to camera position and orientation errors of up to 5 cm and 5° respectively. The pose refinement is robust up to a noise level of 3 cm and 3° . At noise levels with a standard deviation of 10 cm and higher, some cameras are initially positioned inside geometry, preventing our method from refining their position and leading to large errors in geometry reconstruction.

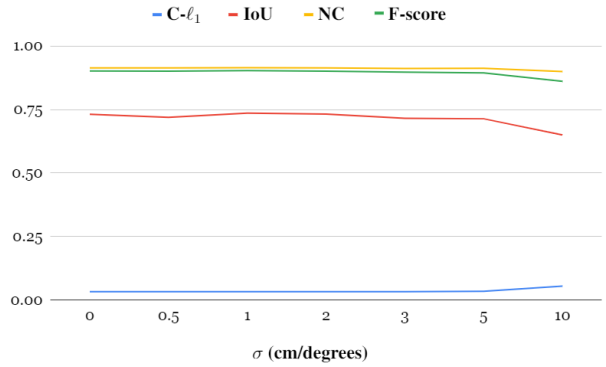


Figure 4. We test the robustness of our reconstructions to noise in the initial camera position and direction. Our method achieves good results even in the presence of significant noise. At $\sigma = 10$ cm, some of the cameras intersect geometry, degrading the reconstruction quality.

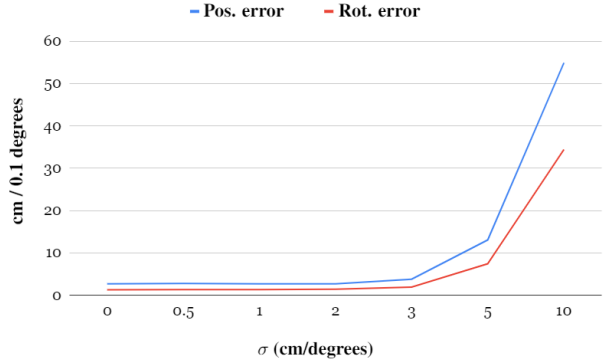


Figure 5. We test the robustness of our pose refinement to noise in the initial camera position and direction. The rotation error has been scaled by a factor of 10 for better visibility. Our method is able to correct poses even in the presence of significant noise. At $\sigma = 10$ cm, some of the cameras start intersecting geometry, making refinement impossible.

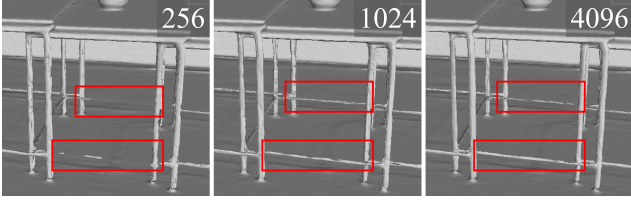


Figure 6. Reconstruction quality with varying batch size.

3.4. Batch Size

Optimization with a lower batch size leads to more noise and might miss areas without depth supervision due to a lower number of multi-view constraints within the batch. A batch size that is too large will slow down the optimization and consume more GPU memory, while not offering improvements in reconstruction quality (see Fig. 6).

3.5. Truncation Size

The reconstruction quality is dependent on the width of the truncation region, as shown in Tab. 6. The truncation region needs to account for the noise in the input (i.e., needs to be greater than the noise of the depth camera). In our experiments a truncation radius of $tr = 5$ cm gives the best results (evaluated based on the mean across multiple scenes).

4. Comparison to RGB-based methods

NeuS [10] and VolSDF [11] are concurrent works that propose learning a signed distance field of an object from a set of RGB images. In contrast to these methods, our focus lies on reconstructing indoor scenes which often have large textureless regions (e.g., a white wall). Methods which use only color input will not have enough multi-view constraints to properly reconstruct these regions. In Fig. 7, we show a case where methods that rely only on color input struggle to reconstruct high-quality geometry.

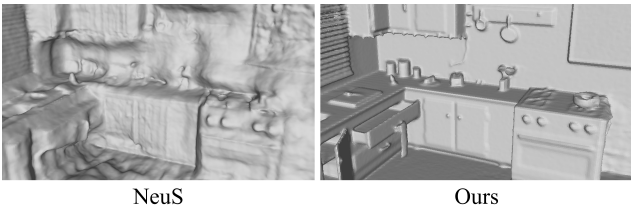


Figure 7. Comparison between NeuS and our method on the ‘morning apartment’ scene.

Truncation (cm)	$C-\ell_1 \downarrow$	IoU \uparrow	NC \uparrow	F-score \uparrow
2	0.053	0.671	0.855	0.862
3	0.023	0.766	0.901	0.930
5	0.021	0.786	0.912	0.933
10	0.024	0.742	0.908	0.912

Table 6. Impact of the truncation width on reconstruction quality.

5. Color Reproduction of Classic and NeRF-style Methods

While our focus lies on geometry reconstruction and not accurate view synthesis, we conducted a brief analysis of the advantages and drawbacks of classic reconstruction methods [3, 13] and MLP-based radiance fields [9] when synthesizing unseen views. Classic reconstruction methods usually do not try to decouple intrinsic material parameters [3, 13] and instead optimize a texture that represents the average observation of all the input views. The resulting texture is usually high-resolution (bounded by the resolution of the input images), but does not allow for correct synthesis of view-dependent effects. Furthermore, inaccuracies in camera calibration may lead to visible seams in the optimized texture. Methods like NeRF that focus purely on high-quality novel view synthesis do not explicitly reconstruct geometry and may thus produce images riddled with artifacts for views that are too far from the input views. We believe that it is possible to combine both of these approaches to improve novel view synthesis on views far away from the ones used during the optimization and would like to encourage research in this direction. Fig. 8 shows an example view synthesis result on the ScanNet dataset, for an out-of-trajectory camera position and orientation.

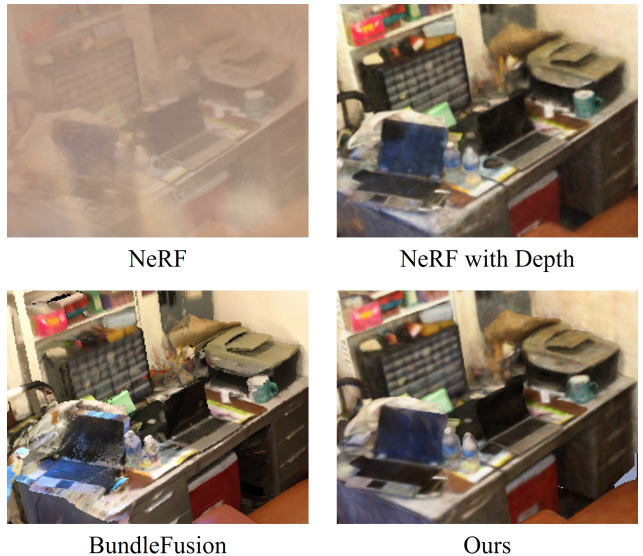


Figure 8. We compare the color synthesis of BundleFusion and NeRF-style methods. NeRF without any depth constraints shows severe fogging when rendering an image from a novel view. This gets resolved after adding depth constraints to the optimization. BundleFusion produces the sharpest results, but suffers from incorrect view-dependent effects and misalignment artifacts. Our method produces results similar to NeRF with a depth constraint. A combination of classic and NeRF-style methods may yield both high-quality geometry and high-quality view synthesis and we encourage further research in this direction.

6. Runtime and Memory Requirements

Our method. The runtime and memory requirements of our method are dependent on the scene size. For smaller scenes where it is enough to have $S'_c = 256$ samples, our method completes 2×10^5 iterations in 9 hours on an NVIDIA RTX 3090 and requires 8.5 GB of GPU memory. When S'_c is set to 512, the runtime increases to 13 hours and the memory requirement to 10.5 GB. The memory consumption can be reduced by using smaller batches.

BundleFusion. We run BundleFusion at a voxel resolution of 1 cm for all scenes. On an NVIDIA GTX TITAN Black, depending on the size of the scene and number of frames in the camera trajectory, it takes 10 to 40 minutes to integrate the depth frames into a truncated signed distance field and extract a mesh using Marching Cubes. The memory usage is around 5.8 GB.

RoutedFusion. To train and test RoutedFusion, we used an NVIDIA RTX 3090. The routing network was trained for 24 hours on images with a resolution of 320×240 pixels. As per suggestion of the authors, we train the fusion network for 20 epochs which takes about 1.5 hours. We reconstruct all scenes at a voxel resolution of 1 cm for a fair comparison to other methods. The runtime ranges from 40 minutes to 6 hours depending on scene size and number of frames. The memory usage also heavily depends on scene size and ranges from 5.5 GB to 23 GB.

COLMAP + Poisson. In the COLMAP + Poisson baseline, the bottleneck is the global bundle adjustment process performed by COLMAP. The total runtime depends on the number of frames in the trajectory. Using all 8 cores of an Intel i7-7700K CPU, it took us about 4 hours to align all 1167 cameras in the ‘breakfast room’. The couple of minutes needed to backproject all depth maps at full resolution and run the screened Poisson surface reconstruction are negligible in comparison.

Convolutional Occupancy Networks. We reconstruct each scene using the pre-trained model provided by the authors. This takes about 2 minutes per scene and requires about 10 GB of memory.

SIREN. We train SIREN for 10^4 epochs on each scene. SIREN is trained over the complete point cloud in each epoch, so the runtime depends on the number of points in the point cloud. In our experiments on an NVIDIA RTX 3090, this ranged from 6 to 12 hours with 12 GB of memory being in use.

NeRF + Depth. We optimize NeRF using 64 samples for the coarse network and 128 samples for the fine network. On an NVIDIA RTX 3090 it takes 6 hours for 2×10^5 iterations to run. The memory usage is 4.7 GB.

References

- [1] Jonathan T. Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. *CVPR*, 2013. 2
- [2] Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. Robot arm pose estimation through pixel-wise part classification. *ICRA*, 2014. 2
- [3] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017. 1, 2, 6
- [4] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blender-proc. *arXiv preprint arXiv:1911.01911*, 2019. 2
- [5] Ankur Handa. Simulating kinect noise for the icl-nuim dataset. <https://github.com/ankurhand/simkinect>. Accessed: 2021-11-15. 2
- [6] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. *ICRA*, 2014. 2
- [7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 1
- [8] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4), July 2017. 3, 4
- [9] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 6
- [10] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 6
- [11] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34, 2021. 6
- [12] Cem Yuksel. A class of c2 interpolating splines. *ACM Trans. Graph.*, 39(5), aug 2020. 2
- [13] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. 33(4), July 2014. 6



Figure 9. We show a qualitative comparison of synthetic scene reconstructions obtained using our method and several baseline methods. The BundleFusion reconstruction is incomplete in some regions, screened Poisson and SIREN attempt to fit noise in the depth data, while the NeRF reconstruction suffers from noise in the density field. Our method manages to fill in gaps in geometry, while maintaining the smoothness of classic fusion approaches.



Figure 10. We show a qualitative comparison of synthetic scene reconstructions obtained using our method and several baseline methods. The BundleFusion reconstruction is incomplete in some regions, screened Poisson and SIREN attempt to fit noise in the depth data, while the NeRF reconstruction suffers from noise in the density field. Our method manages to fill in gaps in geometry, while maintaining the smoothness of classic fusion approaches.