VOXURF: VOXEL-BASED EFFICIENT AND ACCURATE NEURAL SURFACE RECONSTRUCTION SUPPLEMENTARY MATERIALS

Tong Wu[1,2], **Jiaqi Wang**[1], **Xingang Pan**[3], **Xudong Xu**[2], **Christian Theobalt**[3], **Ziwei Liu**[4], **Dahua Lin**[1,2,5]
[1]Shanghai AI Laboratory, [2]The Chinese University of Hong Kong, [3]Max Planck Institute for Informatics,
[4]S-Lab, Nanyang Technological University, [5]Centre of Perceptual and Interactive Intelligence
{wt020, xx018, dhlin}@ie.cuhk.edu.hk, wangjiaqi@pjlab.org.cn,
{xpan,theobalt}@mpi-inf.mpg.de, ziwei.liu@ntu.edu.sg

A EXPERIMENTAL DETAILS

A.1 DATASETS.

The **DTU** (Jensen et al., 2014) dataset contains different static scenes with 49 or 64 posed multi-view images for each scene. It covers a variety of objects with different materials, geometry, and texture. We evaluate our approach on DTU with the same 15 scenes following IDR (Yariv et al., 2020) and quantitatively compare it with previous work on Chamfer Distance, given the ground truth point clouds. The **BlendedMVS** (Yao et al., 2020) dataset contains 113 scenes that cover a variety of real-world environments, providing 31 to 143 posed multi-view images for each. We select 7 challenging scenes following NeuS (Wang et al., 2021) and present qualitative comparisons with previous works.

A.2 IMPLEMENTATION DETAILS

We set the expected number of voxels to be 96^3 at the coarse stage and 256^3 at the fine stage, including an up-scaling step. We use a batch size of 8,192 rays with the point sampling step size on a ray to be half of the voxel size. We train our coarse initialization stage for 10k iterations and the fine geometry optimization stage for 20k iterations with an Adam optimizer. The initial learning rate is set as 1^{-3} for all the MLPs and 0.1 for voxels in the coarse stage, while the SDF voxel starts by 5^{-3} in the fine stage.

We use the same hyper-parameters for all scenes. We use a 3-layer MLP for the coarse training stage and two 4-layer MLPs for the dual color network in the fine training stage. We choose level 2.0 for the hierarchical geometry feature and set the dimension of the learnable feature voxel grid as 6.

For the coarse stage, we set $\lambda_{tv} = 10^{-4}$ and $\lambda_s = 2\times10^{-4}$ for the regularization terms and introduce an additional TV term for $V^{(feat)}$ with a weight of 10^{-2}. The Gaussian kernel is 5^3 in size with $\sigma_g = 0.8$. For the fine stage, we set $\lambda_0 = 0.5$ for the reconstruction loss; we set $\lambda_{tv} = 10^{-3}$ and $\lambda_s = 5 \times 10^{-4}$ for the regularization terms. The fine SDF grid starts with a resolution of 160^3, which is then up-scaled by trilinear interpolation to 256^3 after 15000 iterations.

For the s value in Φ_s in Eqn. 2, we design a function based on the iteration, $s = 1/(i/r + 1/s_{start})$, where s_{start} controls the beginning value of s, i denotes the iteration number, and r basically controls the decaying speed of s along with the increasing iterations. We set $s_{start} = 0.2$, $r = 50$ for the coarse stage and $s_{start} = 0.05$, $r = 50$ for the fine stage.

For all the experimental results on the DTU (Jensen et al., 2014) dataset, our method is trained on the training set with around 90% images for each scene, following (Wang et al., 2021) for the splitting scheme, and the 10% images are used for evaluation of the novel view synthesis task. We notice that the CD performance is only slightly influenced compared to training on the full dataset. For experiments on the BlendedMVS (Yao et al., 2020) dataset, we use all the images for training.

Table R1: Quantitative evaluation on the DTU dataset for novel view synthesis. Our method outperforms the baselines on all the three metrics.

Metric	PSNR↑				SSIM↑				LPIPS↓			
Scan	NeRF	DVGO	NeuS	Ours	NeRF	DVGO	NeuS	Ours	NeRF	DVGO	NeuS	Ours
24	26.97	27.77	26.13	**27.89**	0.772	0.830	0.764	**0.857**	0.331	0.277	0.348	**0.239**
37	25.99	25.96	24.08	**26.90**	0.811	0.833	0.798	**0.870**	0.206	0.184	0.222	**0.160**
40	27.68	27.75	26.73	**28.81**	0.786	0.791	0.747	**0.841**	0.304	0.303	0.352	**0.274**
55	29.39	30.42	28.06	**31.02**	0.917	0.939	0.887	**0.950**	0.143	0.116	0.177	**0.108**
63	33.07	34.35	28.69	**34.38**	0.936	0.953	0.937	**0.957**	0.128	0.095	0.129	**0.083**
65	30.87	31.18	31.41	**31.48**	0.954	0.956	0.958	**0.960**	0.114	0.103	0.112	**0.094**
69	27.90	29.52	28.96	**30.13**	0.844	0.921	0.909	**0.928**	0.308	0.190	0.223	**0.181**
83	33.49	36.94	31.56	**37.43**	0.948	**0.969**	0.950	0.968	0.125	**0.084**	0.120	**0.084**
97	27.43	27.67	25.51	**28.35**	0.900	0.914	0.901	**0.923**	0.200	0.168	0.192	**0.155**
105	31.68	32.85	29.18	**32.94**	0.910	0.928	0.896	**0.932**	0.186	0.154	0.218	**0.148**
106	30.73	33.75	32.60	**34.17**	0.879	0.933	0.914	**0.947**	0.244	0.167	0.201	**0.138**
110	29.61	**33.10**	30.83	32.70	0.872	**0.941**	0.917	0.937	0.241	**0.153**	0.200	**0.153**
114	29.37	30.18	29.32	**30.97**	0.901	0.914	0.897	**0.926**	0.193	0.174	0.216	**0.159**
118	33.44	36.11	35.91	**37.24**	0.915	0.957	0.948	**0.964**	0.199	0.123	0.156	**0.110**
122	33.41	36.99	35.49	**37.97**	0.935	0.967	0.957	**0.972**	0.142	0.088	0.114	**0.076**
mean	30.07	31.64	29.63	**32.16**	0.885	0.916	0.892	**0.929**	0.204	0.159	0.199	**0.144**

Table R2: Quantitative evaluation on DTU dataset (without mask).

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
Colmap (Schönberger et al., 2016)	0.81	2.05	0.73	1.22	1.79	1.58	1.02	3.05	1.40	2.05	1.00	1.32	0.49	0.78	1.17	1.36
NeRF (Mildenhall et al., 2020)	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49
UNISURF (Oechsle et al., 2021)	1.32	1.36	1.72	0.44	1.35	0.79	0.80	1.49	1.37	0.89	0.59	1.47	0.46	0.59	0.62	1.02
VolSDF Yariv et al. (2021)	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	**0.70**	0.66	**1.08**	0.42	0.61	0.55	0.86
NeuS (Wang et al., 2021)	1.00	1.37	0.93	0.43	**1.10**	**0.65**	**0.57**	1.48	1.09	0.83	**0.52**	1.20	**0.35**	**0.49**	0.54	0.84
Ours	**0.72**	**0.75**	**0.47**	**0.39**	1.47	0.76	0.81	**1.02**	**1.04**	0.92	**0.52**	1.13	0.40	0.53	**0.53**	**0.76**

A.3 Details for baseline methods

We include the following baseline methods for comparison: **IDR** (Yariv et al., 2020) reconstructs high-quality surfaces with implicit representation based on foreground object masks and the corresponding mask loss. **NeuS** (Wang et al., 2021) is a state-of-the-art approach that develops a volume rendering method for surface reconstruction, where the mask supervision is optional. Reconstruction results for NeuS are implemented with its official code [1] and the pre-trained models, and the novel view rendering results are provided by the authors. **NeRF** (Mildenhall et al., 2020) first proposes to use the neural radiance field for novel view synthesis. Though not specifically designed for surface reconstruction, we can extract a noise geometry from a trained NeRF model with a selected threshold. In this paper, the reconstruction evaluation results for NeRF are directly taken from (Wang et al., 2021) for a fair comparison, while we also implement NeRF with nerf-pytorch [2] for novel view synthesis. **DVGO** (Sun et al., 2022a) accelerates NeRF with a hybrid representation. We use the official code [3] and implement DVGO-v2 (Sun et al., 2022b) for comparison, which is 2-3 times faster than the DVGO-v1. Similarly, we select a threshold to extract the geometry from the density voxel grid, as to be introduced below. Results for these methods with the aid of foreground object masks are presented in Table 1 of the main text.

We also include several baselines that do not rely on foreground object masks: **Colmap** (Schönberger et al., 2016) is a widely-used classical Multi-view stereo method. **UNISURF** (Oechsle et al., 2021) uses the occupancy field to represent the geometry and improves reconstruction quality by shrinking the sample region of volume rendering during training. **VolSDF** (Yariv et al., 2021) defines the volume density function as Laplace's cumulative distribution function (CDF) applied to a SDF representation for surface reconstruction. We also compare with **NeRF** (Mildenhall et al., 2020) and **NeuS** (Wang et al., 2021) under the without-mask setting. All

[1] https://github.com/Totoro97/NeuS

[2] https://github.com/yenchenlin/nerf-pytorch

[3] https://github.com/sunset1995/DirectVoxGO

Table R3: Ablation over the level selection for hierarchical geometry feature. The performance first increases together with the level and then converges after level 2.0.

level	0	0.5	1	1.5	2	2.5	3
CD	0.98	0.75	0.74	0.74	**0.72**	0.73	**0.72**

Table R4: Ablation over the geometry feature design. It indicates that a combination of both *Gradient* and *SDF* produces the best result.

CD (mean)	0.79	0.76	0.74	**0.72**
Gradient		✓		✓
SDF			✓	✓

Table R5: Ablation over different smoothness priors. Our gradient smoothness loss is proved effective by this quantitative evaluation.

CD (mean)	1.18	0.79	0.74	**0.72**
SDF TV		✓		✓
Gradient smoothness loss			✓	✓

the results above are directly adopted from the original papers. Their comparisons to our method under this setting are shown in Table R2.

B Additional experimental results

B.1 Novel view synthesis.

We report the results for novel view synthesis on the DTU dataset in Table R1. Our method outperforms the baselines in all three metrics, including PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) (VGG). Examples of rendered images at testing views are shown in Fig. S8 and Fig. S9 in Sec. I.

B.2 Comparisons for the w/o mask setting.

Our method can also work on cases where the background is not clean. Following NeRF++ (Zhang et al., 2020) and MipNeRF-360 (Barron et al., 2022), we invert the background points outside the unit sphere into the unit sphere by $x' = x/r^2, y' = y/r^2, z' = z/r^2$, where $r = \sqrt{x^2 + y^2 + z^2}$. We then represent the background with another density voxel grid, together with a feature grid and a shallow MLP. We report our results and comparisons to previous approaches in Table R2.

B.3 Additional ablation studies and analysis

Ablation over the hierarchical geometry feature. For hierarchical geometry feature design, we explore different design details, including the level selection and the effect of gradient and SDF value, as shown in Table R3 and Table R4, respectively.

Ablation over smoothness priors. (1) We introduce two regularization terms as smoothness priors during training, *i.e.* the TV on the SDF voxel grid and a gradient smoothness prior. We carry out an ablation study on them during the fine training stage in Table R5, where we reveal the effectiveness of our gradient smoothness loss via quantitative comparisons. (2) Furthermore, we also evaluate the effectiveness of 3D convolution with a Gaussian kernel during coarse training and post-process. Experimental results show that the CD error increases from 0.72 to 0.74 if we remove Gaussian kernel during coarse training. And the it will be slightly reduced from 0.720 to 0.715 if we remove Gaussian kernel for post-process. The post-processing stage improves the visualization quality with a minor sacrifice of the quantitative performance.

Ablation over the voxel grid resolution. The voxel grid resolution denotes the number of voxels contained in $V^{(sdf)}$ of the fine training stage. We study the effect of voxel grid resolution by keeping all the other settings to be the same, as shown in Table R6. Increasing the voxel grid resolution from 128^3 to 192^3 and from 192^3 to 256^3 consistently results in lower Chamfer Distance (CD) with longer training time. However, the case with 320^3 achieves a similar CD with 256^3, and requires a higher training cost. Thus, we take the number of voxels to be 256^3 as the default setting in the other experiments.

Table R6: The effect of voxel grid resolution on reconstruction performance and training time. All the cases below are trained with the same settings except for the voxel grid resolution.

Resolution	128^3	192^3	256^3	320^3
CD (mean)	0.79	0.75	0.72	0.73
Train time	11 mines	12 mins	14 mins	17 mins

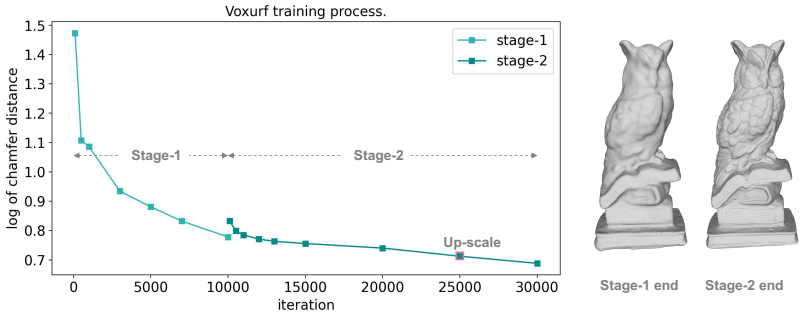

Figure S1: The two-stage training process of Voxurf. The number in the vertical axis is calculated by $\log_{10}(10x)$ for better visualization.

Two-stage training process. Our method adopts a two-stage training pipeline. We show the curve of Chamfer Distance and the visualization result by the end of each stage in Fig. S1. We show that 1) we can obtain a coherent shape by the end of the Stage-1 (coarse training stage), while the performance is limited by the low resolution that the details are hard to be reconstructed; 2) the fine details are recovered by the end of Stage-2 (fine training stage) and the overall structure is consistent with the coarse shape of Stage-1.

Threshold selection. To extract the surface from a trained DVGO (Sun et al., 2022a) model, we can first obtain the alpha value for any point in the 3D space by density interpolation and activation. And then, we show how we select a proper alpha threshold when we extract the surface in Fig. S2. A small threshold like 0.001 and 0.01 usually results in noise areas floating above the surface, while a large one like 0.5 and 0.8 would lead to an incomplete surface with large holes. We thus select 0.1 as the alpha threshold that is adopted in this paper.

C DVGO+NeuS w/ Smoothness

Our method introduces several designs to boost the smoothness of the reconstruction results, including the 3D convolution with a Gaussian kernel $\mathcal{G}(V, k_g, \sigma_g)$ for coarse stage training (not adopted in fine stage) (Sec. 5.1), the gradient smoothness loss $\mathcal{L}_{smooth}(\nabla V^{(sdf)})$ (Sec. 5.3), a 3D convolution with a Gaussian kernel $\mathcal{G}(V, k_g, \sigma_g)$ for post-processing during inference (Sec. 5.3), and the SDF total variation loss $\mathcal{L}_{TV}(V^{(sdf)})$ (Sec. 5.3).

Since we observe the combination of DVGO and NeuS (DVGO + NeuS) produces continuous but noisy surfaces, it is interesting to apply these smoothness designs to the baseline. In this way, we can more clearly verify the effectiveness of other technical contributions in Voxurf. Specifically, the SDF total variation loss (SDF TV) has already been adopted to DVGO + NeuS for better results in Fig. 1 and Table. 1. We further evaluate the Gaussian kernel for training, gradient smoothness loss, and

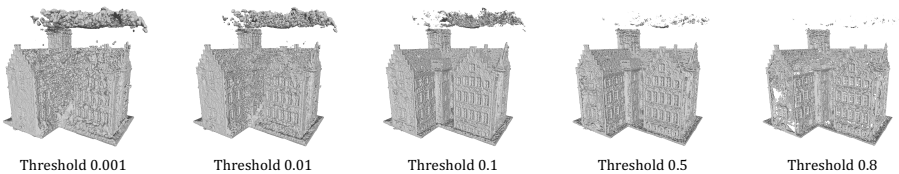

Figure S2: Comparisons of Alpha threshold selection for surface extraction from a trained DVGO (Sun et al., 2022a) model. A large threshold leads to holes and the incomplete surface, while a small one leads to floating noises above the surface.

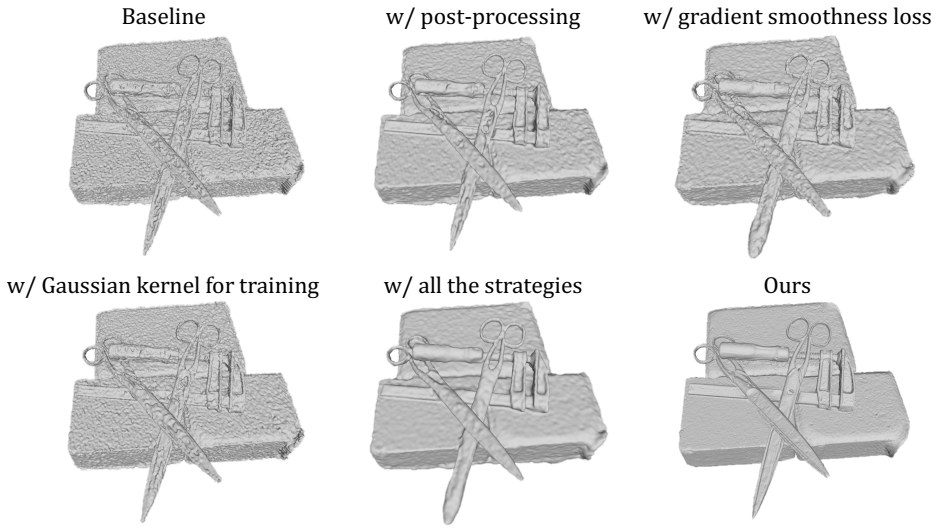

Figure S3: Qualitative comparisons of the DVGO+SDF baseline enhanced with different combinations of smoothness priors. A huge performance gap still exists when the strongest smoothness strategies are applied.

Table R7: The experimental results of applying Gaussian kernel for post-processing, gradient smoothness loss, and Gaussian kernel for training on DVGO + NeuS. Notably, SDF TV has already been adopted as a default setting, thus it is not discussed here.

CD (mean)	1.13	1.22	0.99	1.00	1.00	1.00	0.98
Post-processing		✓		✓		✓	✓
Gradient smoothness loss			✓	✓			✓
Gaussian kernel for training					✓	✓	✓
Voxurf				**0.72**			

Gaussian kernel for post-processing on DVGO + NeuS. The most simple approach to make surfaces smooth is adopting the Gaussian kernel for post-processing. As shown in Fig. S3, the surfaces clearly become smooth, but the chamfer distance (CD) becomes worse as in Table R7. As either the Gradient smooth loss or Gaussian kernel for training is leveraged, both the chamfer distance (CD) and surfaces become better. Notably, the Gaussian kernel for post-processing will not harm the performance here with these smooth priors (*i.e.*, Gradient smoothness loss, Gaussian kernel for training) adopted. Finally, via combining all of these designs, we achieve smooth surfaces and lower CD. However, both the quantitative comparison and visualization show that Voxurf is significantly better than DVGO + NeuS w/ smoothness. Moreover, after adopting all these designs, the training time of DVGO + NeuS w/ smoothness increases from 12min to 15min which is slightly longer than Voxurf (14min), singe Voxurf does not need the Gaussian kernel for training in the second stage thanks to the other designs.

D Discussions on the assumption of color-geometry dependency

The assumption of color-geometry dependency is based on the idea of shape-from-shading, which has been proven effective in surface reconstruction by previous approaches (Yariv et al., 2020; 2021). This technique generally does more good than harm, while side effects do exist in some cases where the surface texture is not correlated with the geometry/normal. We carry out several experiments with an example in Fig. S4, where we can observe obvious relief-like structures on a plane surface caused by the texture.

This is a common problem shared by most of the recent neural surface reconstruction methods (Oechsle et al., 2021; Yariv et al., 2020; 2021; Wang et al., 2021). Nevertheless, this problem can

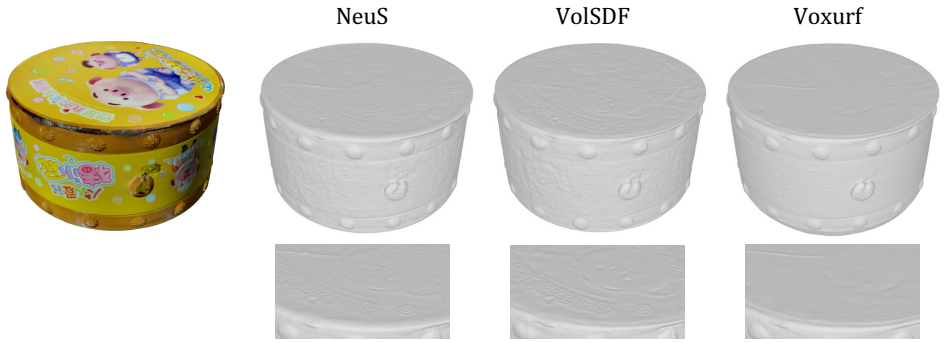

Figure S4: An example where the surface color is not correlated with geometry. The assumption of the color-geometry dependency leads to relief-like structures in VolSDF (Yariv et al., 2021), NeuS (Wang et al., 2021), and Voxurf, where our method is **least** affected by the side effect. It reveals that this problem can be mostly alleviated by multi-view consistency in an accurate reconstruction.

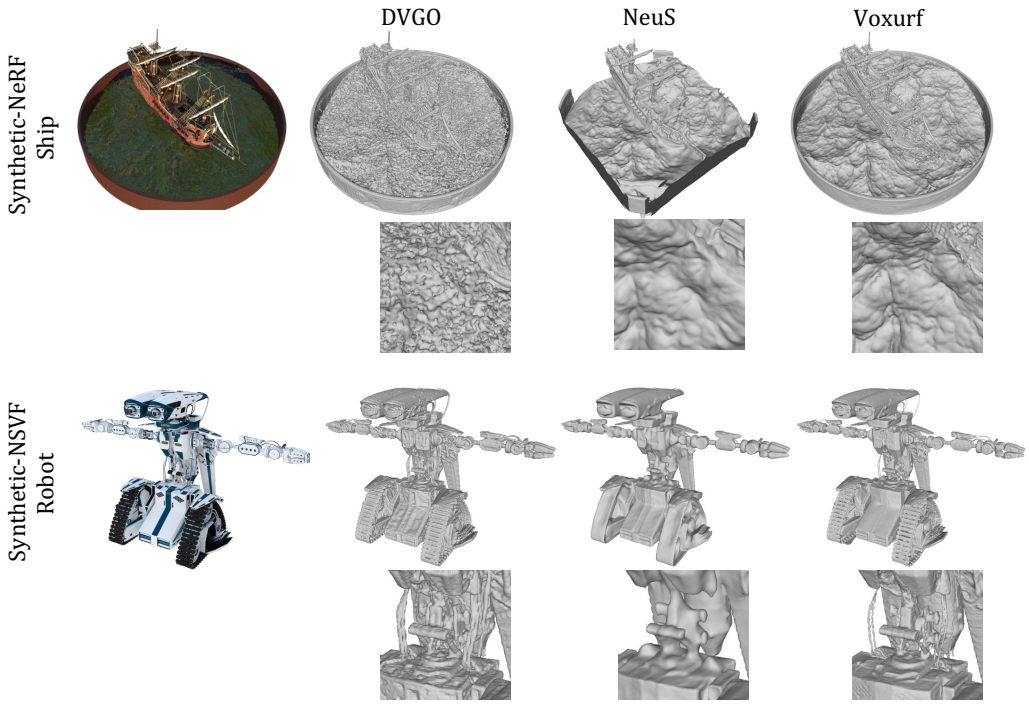

Figure S5: Qualitative comparisons on the *Synthetic-NeRF* and *Synthetic-NSVF* dataset.

be largely alleviated via multi-view consistency when the geometry and color fields are well-trained with enough input views. In Fig. S4, we observe that our method is better at overcoming the problem than previous state-of-the-art methods, being least affected by the textures. Fully addressing this side effect is out of the scope of this work and would be investigated in the future.

E Evaluations on new datasets

We further evaluate our method on other datasets, namely *Synthetic-NeRF* (Mildenhall et al., 2020) and *Synthetic-NSVF* (Liu et al., 2020), and a qualitative comparison can be found in Fig. S7. Compared with the baseline methods, our method shows superior performance, which does not suffer from the heavy noise as DVGO (Sun et al., 2022a) while producing far more accurate thin structures than NeuS (Wang et al., 2021).

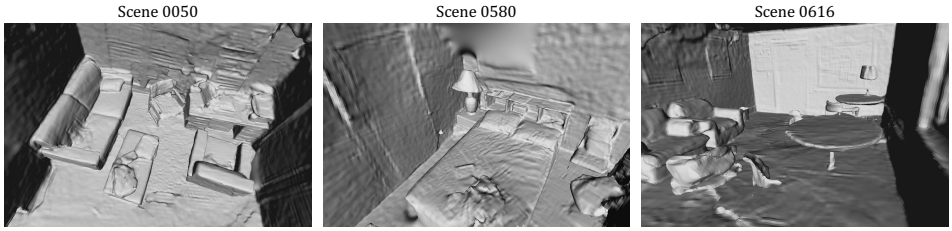

Figure S6: Qualitative results on large scenes, *e.g.*, the ScanNet dataset.

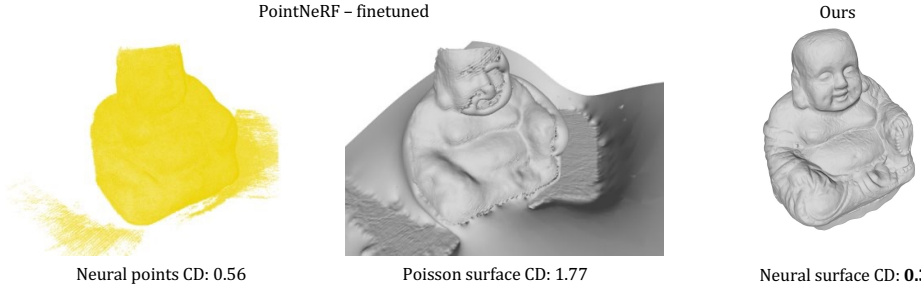

Figure S7: A comparison with Point-NeRF (Xu et al., 2022) in surface reconstruction. We use the Open3D Libarary (Zhou et al., 2018) to perform Poisson surface reconstruction from the point cloud.

These datasets rarely appear in previous benchmarks of surface reconstruction, because they do NOT have publicly available 3D ground truth for evaluation. The average PSNRs of our method on *Synthetic-NeRF* and *Synthetic-NSVF* are 32.38 and 35.18, where DVGO achieves 31.95 and 35.08, respectively. Considering the time limitation, we will provide a full comparison of both geometry and novel view synthesis on multiple methods in the final version. Results for more datasets (*e.g.*, deepvoxels (Sitzmann et al., 2019) and Tanks-and-Temples (Knapitsch et al., 2017)) will also be included in the final version.

F Qualitative Results on Large Scenes

We further evaluate Voxurf on large scenes, *i.e.*, Scannet dataset (Dai et al., 2017). Following MonoSDF (Yu et al., 2022), we adopt depth and normal estimated by MiDaS (Ranftl et al., 2022) as extra supervision. In Fig. S6, we show the qualitative results on scene 0050, 0580, and 0616 of Scannet. The results show that Voxurf can basically model the scene with an accurate layout and some details. However, it still performs unsatisfactory in reconstructing planes, *e.g.*, floor and wall. We conjecture that the under-constrained voxel grids are not suitable to reconstruct from sparse-view observations like ScanNet videos. It is an import topic for future works.

G Comparisons with more methods

We further include Point-NeRF (Xu et al., 2022) for a comparison, which is an impressive work on NeRF acceleration based on an innovative neural point representation. This approach is proposed for the NVS task, and we manage to obtain the neural points after the finetuning stage and use Poisson surface reconstruction (Kazhdan et al., 2006) to extract a surface from the points. The DTU performance on NVS task of Point-NeRF and our method are not directly comparable since they use different scenes and splits for training and testing. Therefore, we currently compare surface reconstruction results on the only one shared scene as shown in Fig. S7. It can be seen that the neural points after fine-tuning can roughly represent the surface of the scene, while the CD error (0.56) is obviously higher than ours (0.37). When we use Poisson surface reconstruction (Kazhdan et al., 2006), a widely used method to mesh a point cloud, to recover a water-tight surface with the neural points, the CD error substantially increases to 1.77. We will try to adopt Point-NeRF in the benchmark and perform a comprehensive evaluation on both surface reconstruction and rendering in the final version.

H LIMITATIONS

Here we discuss the limitations and further research directions of our Voxurf.

(1) Current neural surface reconstruction approaches (Oechsle et al., 2021; Yariv et al., 2020; 2021; Wang et al., 2021), including our Voxurf, are built upon color-geometry dependence. Although the assumption does more good than harm, it sometimes causes side effects, *e.g.*, a plane with printed textures will lead to relief-like structures. As discussed in Sec. D and Fig. S4, Voxurf is relatively less affected by such cases due to the accurate surface reconstruction enhanced by multi-view consistency, but the undesirable artifacts still exist. It is valuable to explore how to tackle this problem in the future.

(2) Voxurf follows the NeRF-based techniques that represent 3D objects with view-conditioned emitted radiance. This design is insufficient to reconstruct an accurate surface with less texture, strong reflection, and transparency due to strong ambiguity. This is another problem we can explore with Voxurf.

I ADDITIONAL QUALITATIVE COMPARISONS

Finally, we show the qualitative comparisons for novel view synthesis in Fig. S8 and Fig. S9, and we show additional surface reconstruction results in Fig S10, Fig S11, and Fig S12.

Figure S8: Qualitative comparisons on DTU for novel view synthesis. (Part 1/2)

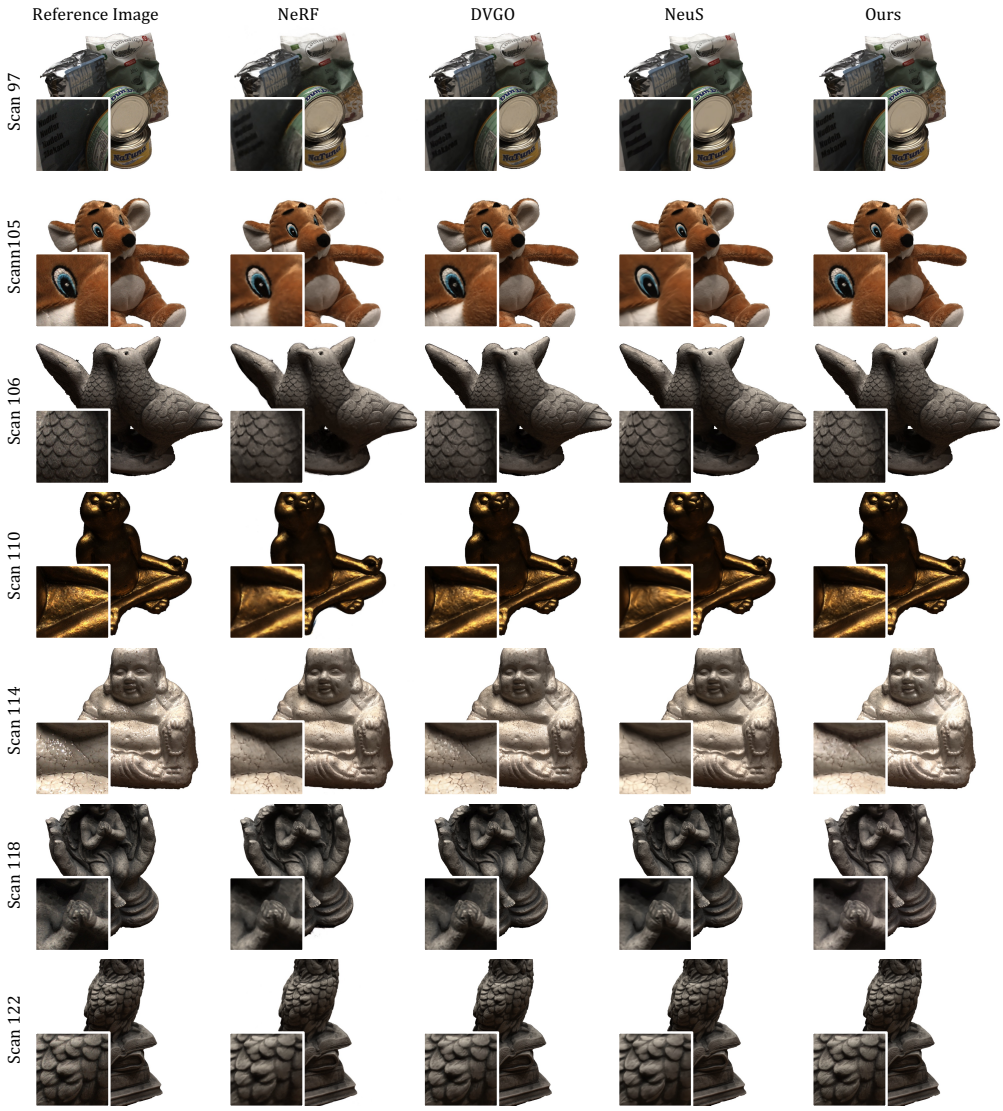

Figure S9: Qualitative comparisons on DTU for novel view synthesis. (Part 2/2)

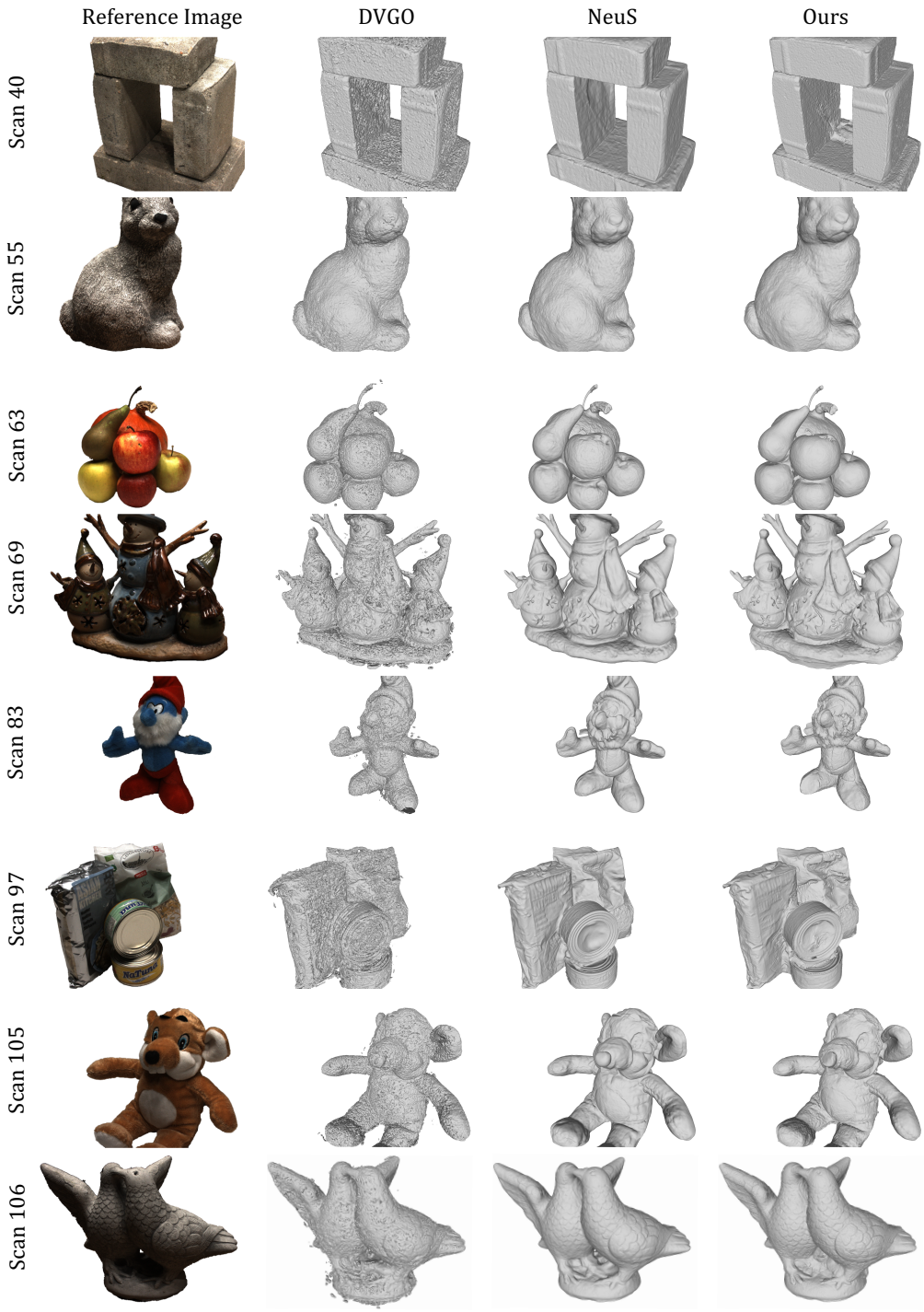

Figure S10: Additional surface reconstruction comparisons on DTU. (Part 1/2)

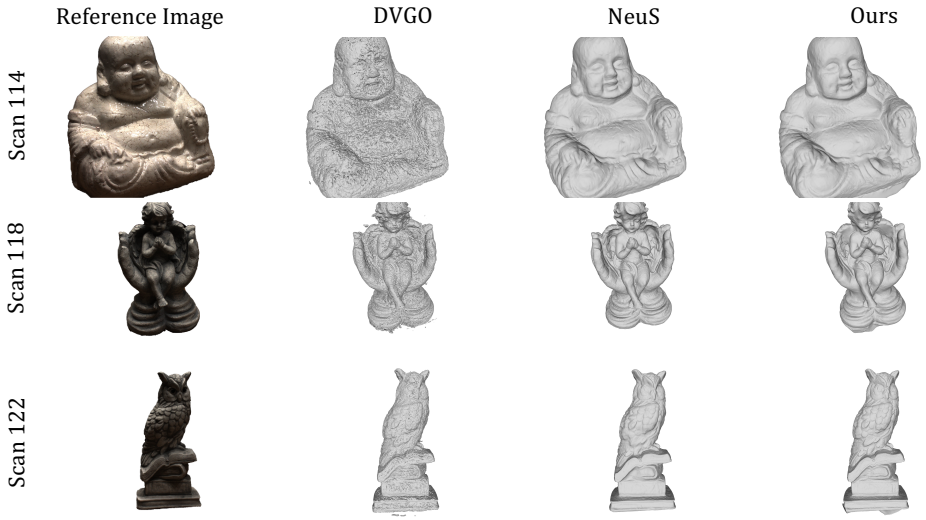

Figure S11: Additional surface reconstruction comparisons on DTU. (Part 2/2)

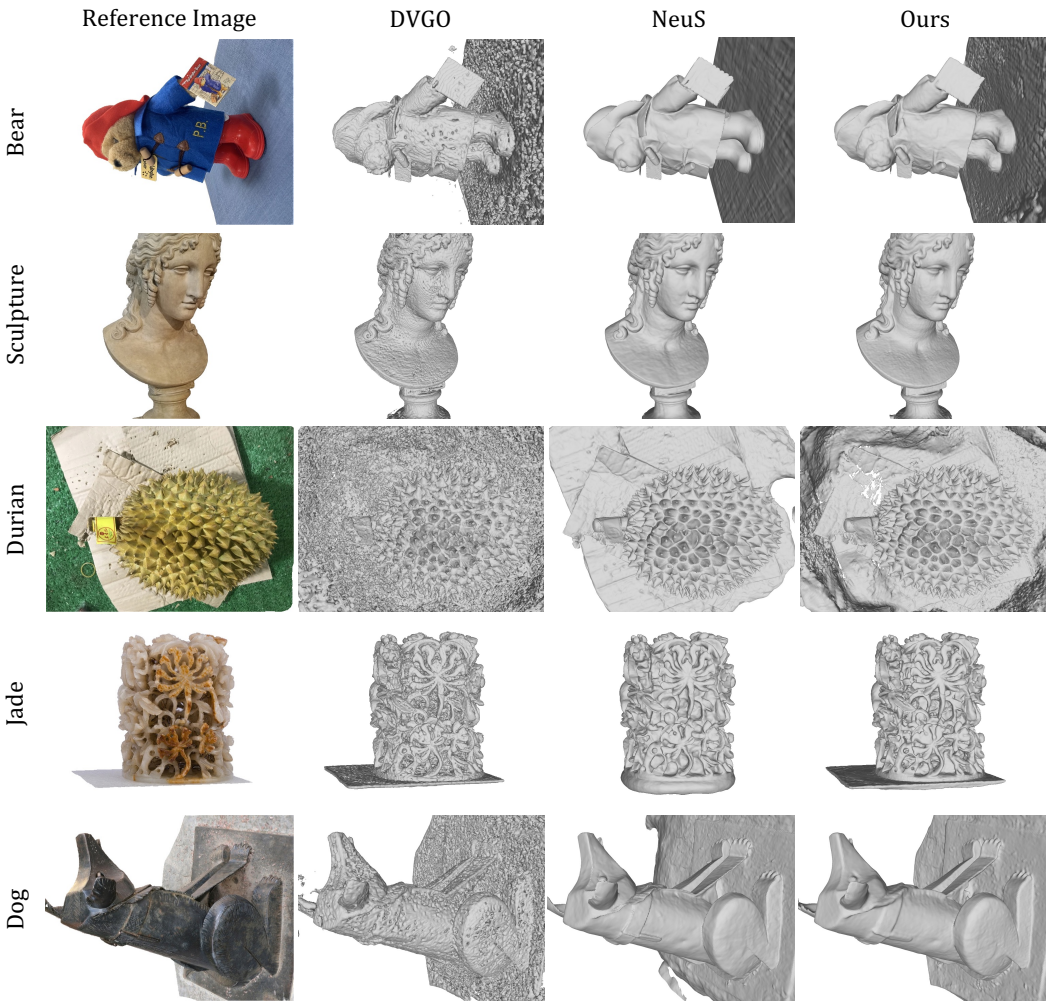

Figure S12: Additional surface reconstruction comparisons on BlendedMVS.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR, 2022.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2017.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 406–413, 2014.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing, volume 7, 2006.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics, 36(4), 2017.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. Advances in Neural Information Processing Systems (NeurIPS), 2020.

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 European conference on computer vision, pp. 405–421. Springer, 2020.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5589–5599, 2021.

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, 44(3), 2022.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In European Conference on Computer Vision, pp. 501–518. Springer, 2016.

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2437–2446, 2019.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2022a.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction, 2022b.

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. Advances in Neural Information Processing Systems (NeurIPS), 2021.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitio, 2022.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1790–1799, 2020.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems (NeurIPS), 33, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In Thirty-Fifth Conference on Neural Information Processing Systems, 2021.

Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in Neural Information Processing Systems (NeurIPS), 2022.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492, 2020.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586–595, 2018.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. arXiv:1801.09847, 2018.