Neuralangelo: High-Fidelity Neural Surface Reconstruction

Zhaoshuo Li[1,2] Thomas Müller[1] Alex Evans[1] Russell H. Taylor[2] Mathias Unberath[2]
Ming-Yu Liu[1] Chen-Hsuan Lin[1]

[1]NVIDIA Research [2]Johns Hopkins University

https://research.nvidia.com/labs/dir/neuralangelo

Acknowledgements. We thank Alexander Keller, Tsung-Yi Lin, Yen-Chen Lin, Stan Birchfield, Zan Gojcic, Tianchang Shen, and Zian Wang for helpful discussions and paper proofreading. This work was done during Zhaoshuo Li's internship at NVIDIA Research and funded in part by NIDCD K08 Grant DC019708.

A. Additional Hyper-parameters

Following prior work [14–16], we assume the region of interest is inside a unit sphere. The total number of training iterations is 500k. When a given hash resolution is not active, we set the feature vectors to zero. We use a learning rate of 1×10^{-3} with a linear warmup of 5k iterations. We decay the learning rate by a factor of 10 at 300k and 400k. We use AdamW [6] optimizer with a weight decay of 10^{-2}. We set $w_{\text{eik}} = 0.1$. The curvature regularization strength w_{curv} linearly warms up 5×10^{-4} following the schedule of learning rate and decays by the same spacing factor between hash resolutions every time ϵ decreases. The SDF MLP has one layer, while the color MLP has four layers. For the DTU benchmark, we follow prior work [14–16] and use a batch size of 1. For the Tanks and Temples dataset, we use a batch size of 16. We use the marching cubes algorithm [5] to convert predicted SDF to triangular meshes. The marching cubes resolution is set to 512 for the DTU benchmark following prior work [1, 14–16] and 2048 for the Tanks and Temples dataset.

B. Additional In-the-wild Results

We present additional in-the-wild results collected at the NVIDIA HQ Park and Johns Hopkins University in Figure 1. The videos are captured by a consumer drone. The camera intrinsics and poses are recovered using COLMAP [11]. To define the bounding regions, we have developed an open-sourced Blender add-on[1] to allow users interactively select regions of interest using the sparse point cloud from COLMAP. The surfaces are reconstructed using the same setup and hyperparameters as the Tanks and Temples dataset. Neuralangelo successfully reconstructs complex geometries and scene details, such as the buildings, sculptures, trees, umbrellas, walkways, and *etc*. Using the same setup as Tanks and Temples also suggests that Neuralangelo is generalizable with the proposed set of hyper-parameters.

[1]https://github.com/mli0603/BlenderNeuralangelo

	F1 Score ↑		
	NeuS [14]	Geo-Neus [2]	Ours
Barn	0.29	**0.33**	**0.70**
Caterpillar	**0.29**	0.26	**0.36**
Courthouse	**0.17**	0.12	**0.28**
Ignatius	**0.83**	0.72	**0.89**
Meetingroom	**0.24**	0.20	**0.32**
Truck	**0.45**	**0.45**	**0.48**
Mean	**0.38**	0.35	**0.50**

Table 1. **Additional quantitative results on Tanks and Temples dataset [4].** Neuralangelo achieves the best surface reconstruction quality and performs best on average in terms of image synthesis. **Best result**. **Second best result.** Best viewed in color.

C. Additional Tanks and Temples Results

We present additional results on the Tanks and Temples dataset [4] in this section.

Surface reconstruction. Concurrent with our work, Geo-NeuS [2] uses the sparse point clouds from COLMAP [11] to improve the surface quality. However, we find that in large-scale in-the-wild scenes, the COLMAP point clouds are often noisy, even after filtering. Using the noisy point clouds may degrade the results, similarly observed in [18]. As evidence, we benchmark Geo-NeuS [2] on Tanks and Temples (Table 1). We find that Geo-NeuS performs worse than NeuS and Neuralangelo in most scenes.

RGB image synthesis. Due to similarities between adjacent video frames, we report PSNR by sub-sampling 10 times input video temporally and evaluating the sub-sampled video frames. Qualitative comparison of Neuralangelo and

NVIDIA HQ Park

Johns Hopkins University

Figure 1. Reconstruction results of **NVIDIA HQ Park** and **Johns Hopkins University**. Videos are captured by a consumer drone.

prior work NeuS [14] is shown in Fig 2. Neuralangelo produces high-fidelity renderings compared to NeuS [14], with details on the buildings and objects recovered. Neither COLMAP [11] nor NeuralWarp [1] supports view synthesis or accounts for view-dependent effects. Thus, we only report the F1 score of the reconstructed surfaces for these two approaches.

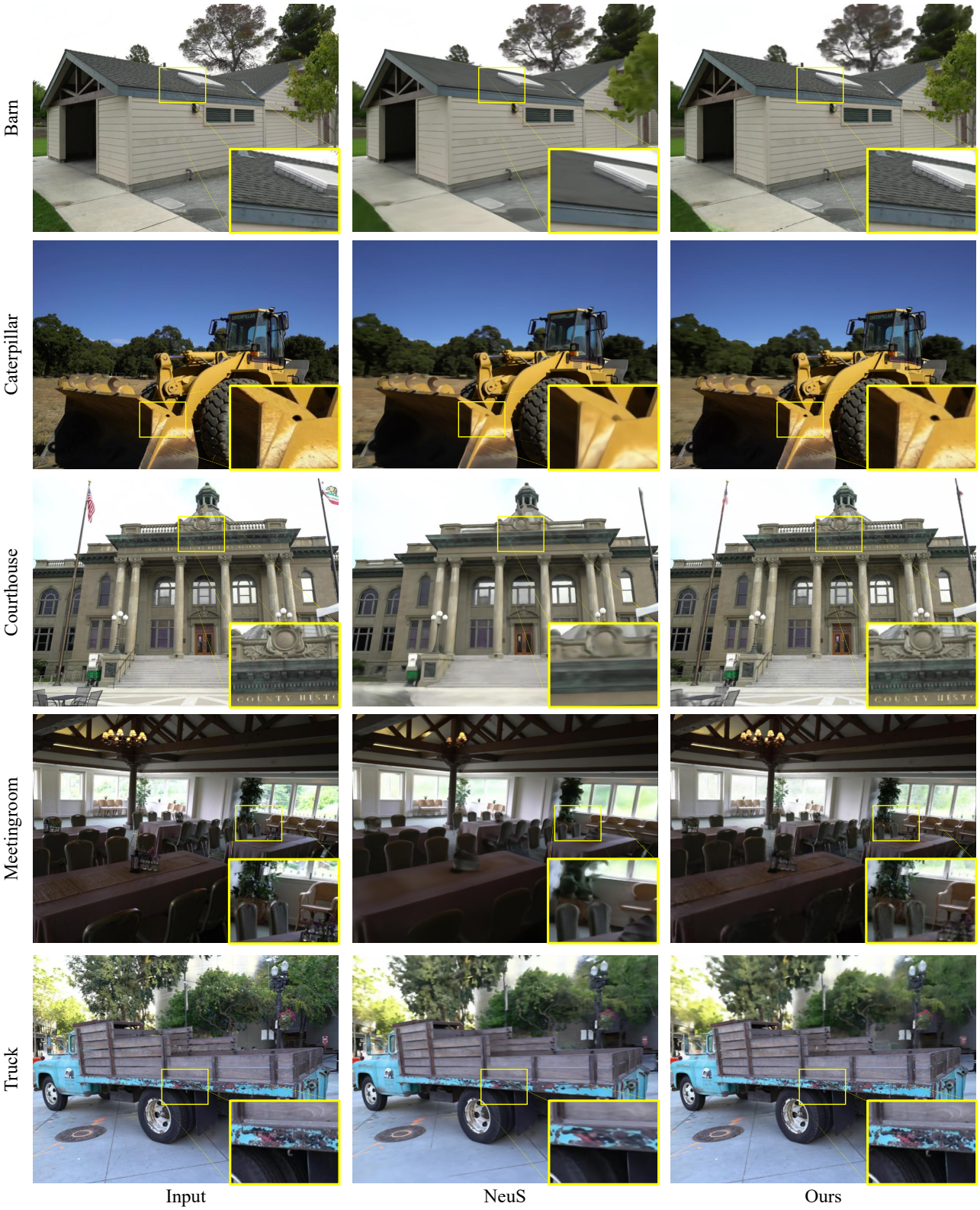


Figure 2. **Qualitative comparison of image rendering on the Tanks and Temples dataset [4].** Compared to NeuS [14], Neuralangelo generates high-quality renderings with texture details on the buildings and objects.

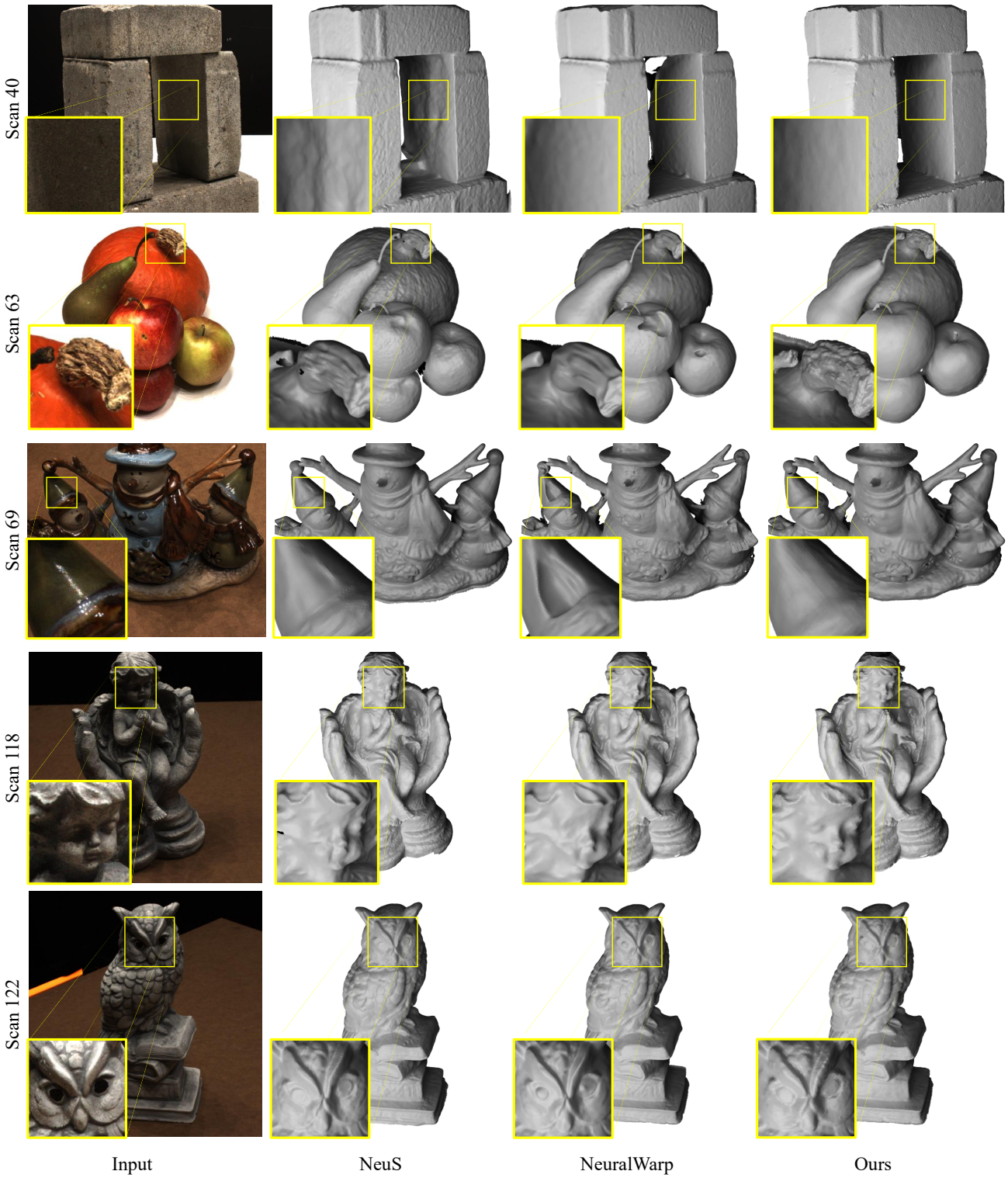


Figure 3. **Qualitative comparison on additional scenes of the DTU benchmark [3].** Neuralangelo can produce both smooth surfaces and detailed structures compared to prior work, despite limited improvement in simply textured and highly reflective objects.

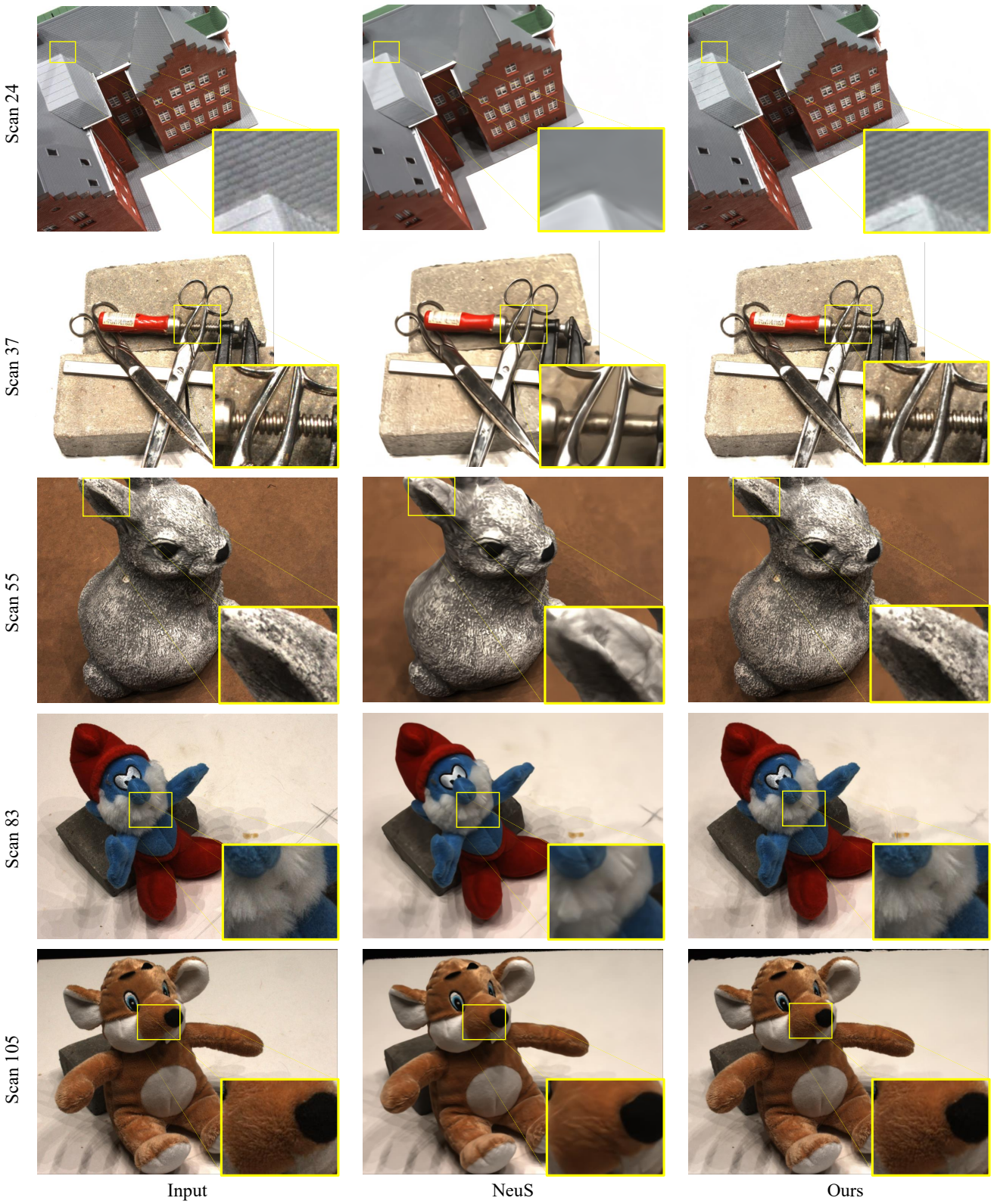


Figure 4. **Qualitative comparison of RGB image synthesis on the DTU benchmark [3].** Compared to NeuS [14], Neuralangelo generates high-fidelity renderings with minute details.

D. Additional DTU Results

We present additional results on the DTU benchmark [3] in this section.

Surface reconstruction. We visualize the reconstructed surfaces of additional scenes of the DTU benchmark. Qualitative comparison with NeuS [14] and NeuralWarp [1] are shown in Fig. 3.

Compared to prior work, Neuralangelo not only can reconstruct smoother surfaces such as in Scan 40, 63, and 69 but also produces sharper details such as in Scan 63 and 118 (*e.g.* the details of the pumpkin vine and the statue face). While Neuralangelo performs better on average across scenes, we note that the qualitative result of Neuralangelo does not improve significantly in Scan 122, where the object of interest has mostly diffuse materials and relatively simple textures. Moreover, we find that Neuralangelo fails to recover details compared to NeuS [14] when the scene is highly reflective, such as Scan 69. Neuralangelo misses the button structures and eyes. Such a finding agrees with the results of Instant NGP [8], where NeRF using Fourier frequency encoding and deep MLP performs favorably against multi-resolution hash encoding for highly reflective surfaces. Future work on improving the robustness of Neuralangelo in reflective scenes, a drawback inherited from hash encoding, can further generalize the application of Neuralangelo.

RGB image synthesis. In the paper, we report the PSNR result of Neuralangelo to quantify the image synthesis quality. Due to the simplicity of the background, we only evaluate the PSNR of the foreground objects given the object masks. We visualize the rendered images in Fig. 4. We only choose NeuS [14] as our baseline as NeuralWarp [1] does not generate rendered images.

Fig. 4 shows that Neuralangelo successfully renders the detailed textures while NeuS produces overly smoothed images. The results suggest that Neuralangelo is able to produce high-fidelity renderings and capture details better.

DTU foreground mask. The foreground object masks are used to remove the background for proper evaluation [1, 9, 14, 16, 19] on the DTU benchmark. We follow the evaluation protocol of NeuralWarp [1] and dilate the object masks by 12 pixels. In all prior work, the foreground object masks used are annotated and provided by the authors of IDR [16].

	Chamfer distance (mm) ↓	
	IDR masks	Our masks
NeuS [14]	1.48	0.99
NeuralWarp [1]	1.20	0.73
Ours	1.29	0.76

Table 2. **Quantitative results on Scan 83 of the DTU dataset [3]** using object masks provided by IDR [16] and annotated by us.

However, we find that the provided masks are imperfect in Scan 83. Fig. 5 shows that part of the object is annotated as background. The masks provided by IDR also only include the foreground objects while the ground truth point clouds include the brick holding the objects. Thus, we manually annotate Scan 83 and report the updated results in Table 2 for additional comparison. We note that fixing the object masks for Scan 83 leads to improved results across all methods.

E. Additional Ablations

We conduct additional ablations and summarize the results in this section.

Color network. For the Tanks and Temples dataset, we add per-image latent embedding to the color network following NeRF-W [7] to model the exposure variation across frames. Qualitative results are shown in Fig. 6. After introducing the per-image embedding, the floating objects used to explain exposure variation have been greatly reduced.

Curvature regularization strength. The curvature regularization adds a smoothness prior to the optimization. As the step size ϵ decreases and finer hash grids are activated, finer details may be smoothed if the curvature regularization is too strong. To avoid loss of details, we scale down the curvature regularization strength by the spacing factor between hash resolutions each time the step size ϵ decreases. Details are better preserved by decaying w_{curv} (Fig. 7).

Numerical *v.s* analytical gradient. We visualize in Fig. 8 the surface normals computed by using both numerical and analytical gradients after the optimization finishes. At the end of the optimization, the step size ϵ has decreased sufficiently small to the grid size of the finest hash resolution. Using numerical gradients is nearly identical to using analytical gradients. Fig. 8 shows that the surface normals computed from both numerical and analytical gradients are indeed qualitatively similar, with negligible errors scattered across the object.

Color network. By default, we follow prior work [14, 16] and predict color conditioned on view direction, surface normal, point location, and features from the SDF MLP. We use spherical harmonics following [17] to encode view direction as it provides meaningful interpolation in the angular domain. When the data is captured with exposure variation in the wild, such as the Tanks and Temples dataset, we further add per-image appearance encoding following NeRF-W [7].

We have also implemented a more explicit color modeling process. The color network is shown in Fig. 9, attempting to better disentangle color-shape ambiguities. However, we do *not* observe improvements in surface qualities using such a decomposition design. The intrinsic decomposed color network contains two branches – albedo and shading branches. The final rendered image $C \in \mathbb{R}^3$ is the sum of the albedo

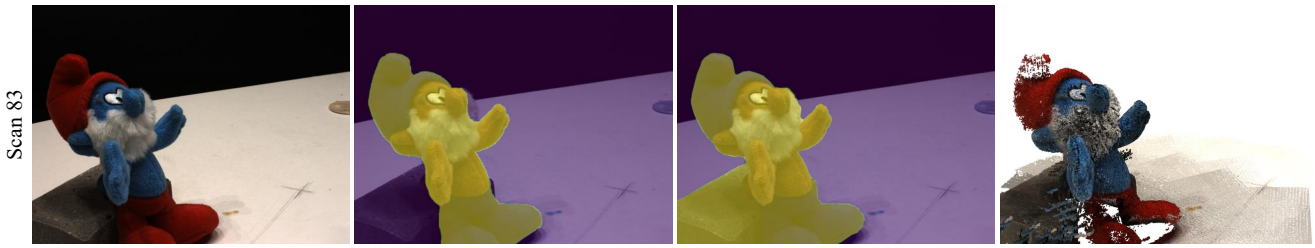


Figure 5. **We manually re-annotate the foreground object masks of the DTU dataset.** We note that the object masks provided by IDR miss the objects partially on Scan 83. The IDR masks also do not include the bricks holding objects, while ground truth point clouds have the brick. Our updated segmentation masks fix the above issues for better evaluation.

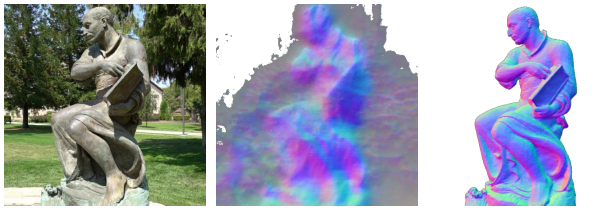


Figure 6. **Qualitative comparison of normal maps without and with per-image embedding.** Floaters are greatly reduced with per-image embedding.

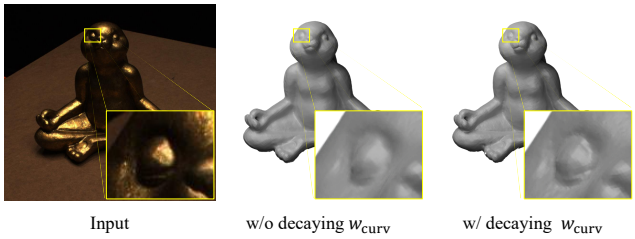


Figure 7. **Qualitative comparison of without and with decaying w_{curv}.** Decaying w_{curv} reduces the regularization strength as ϵ decreases, thus preserving details better.

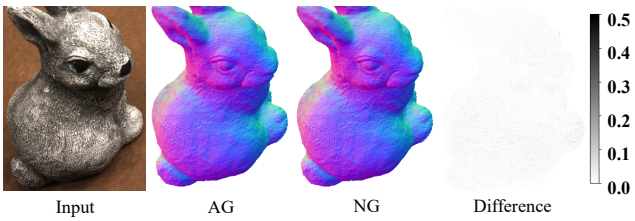


Figure 8. **Qualitative visualizations of surface normals computed from analytical gradient (AG) and numerical gradient (NG).** The results are nearly identical at the end of the optimization due to the small step size ϵ.

image C_a and shading image C_s:

$$C = \Phi(C_a + C_s), \tag{1}$$

where Φ is the Sigmoid function to normalize the predictions into the range of 0 to 1.

The albedo branch predicts RGB values $C_a \in \mathbb{R}^3$ that are view-invariant. It receives point locations and features from the SDF MLP as input. On the other hand, the shading branch predicts gray values $C_s \in \mathbb{R}$ that is view dependent to capture reflection, varying shadow, and exposure changes. We opt for the single channel design for the shading branch as specular highlights, exposure variations, and moving shadows are often intensity changes [10]. The single-channel gray color design also encourages the albedo branch to learn the view-invariant color better as the shading branch is limited in its capacity. Other than the point locations and SDF MLP features, the shading branch is additionally conditioned on reflection direction and view direction following RefNeRF [13] to encourage better shape recovery. We use two hidden layers for the albedo branch and two hidden layers for the diffuse branch to make a fair comparison with the default color network proposed by IDR [16].

We find that with the decomposed color network, the shading branch indeed successfully explains view-dependent effects (Fig. 9). However, flat surfaces tend to be carved away, potentially due to the instability of dot product from reflection computation (Fig. 10). Our future work will explore more principled ways for intrinsic color decomposition.

Computation time. We compare the training and inference time in Table 3 across different setups using our implementation in PyTorch. The experiments are conducted on NVIDIA V100 GPUs. We note that the training time per iteration when using numerical gradients is longer than using analytical gradients due to additional queries of SDF. Using numerical gradients experiences approximately a 1.2 times slowdown compared to using analytical gradients. As NeuS uses 8-layer MLP for SDF MLP and Neuralangelo uses 1-layer MLP, using numerical gradients is still faster than NeuS [14]. We also compare the inference time for surface extraction of 128^3 resolution. As numerical gradients are used only for training, the speed for NG and AG are the same.

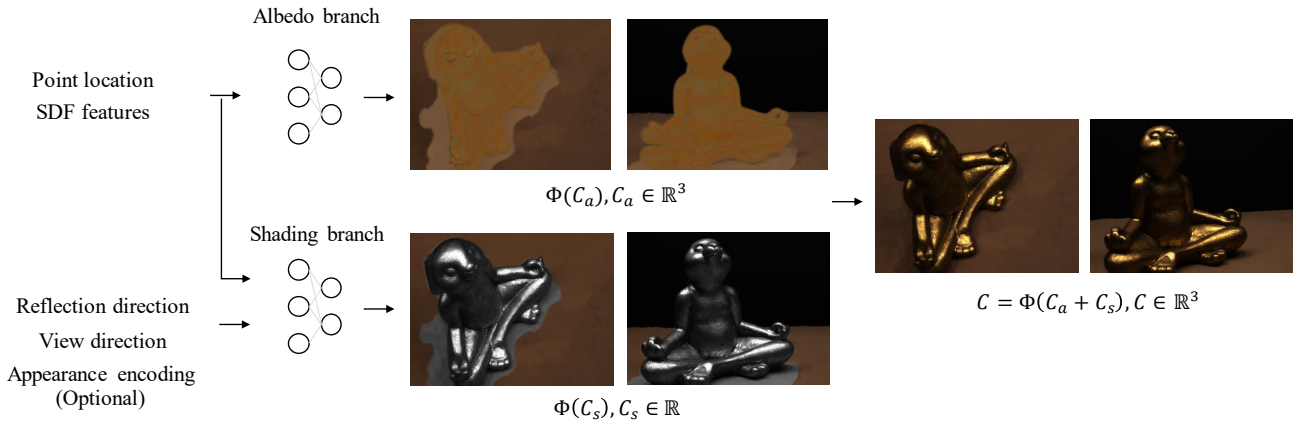
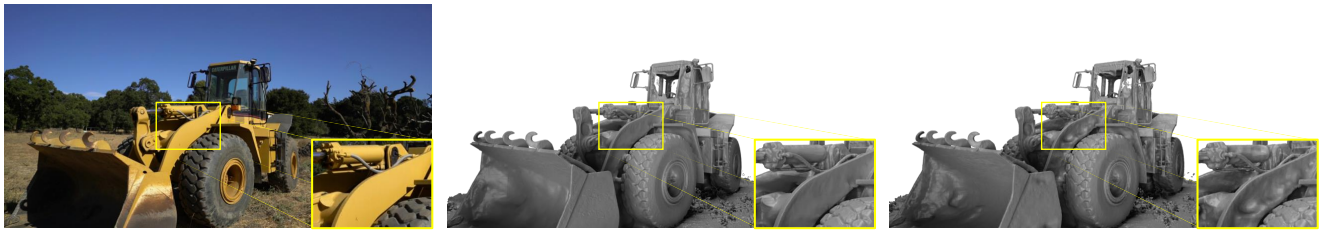


Figure 9. Color network design for intrinsic decomposition. The decomposition scheme includes albedo and shading images.



Input Ours w/ intrinsic decomposition

Figure 10. Qualitative comparison of different color network designs. We find that the intrinsic decomposition we implemented lacks smoothness in regions with homogeneous color, while the color network proposed by IDR [16] produces smooth surfaces.

	Training time (s)	Inference time (s)
NeuS [14]	0.16	0.19
NG (Ours)	0.12	0.08
AG	0.10	0.08

Table 3. **Computational time comparison between NeuS [14], AG and NG using Nvidia V100 GPUs.** Training time reported is per iteration and inference time reported is for surface extraction of 128^3 resolution. There is approximately a 1.2 times slowdown in training time of ours compared to AG. Ours is still faster than NeuS due to the smaller-sized MLP used. For inference time, both ours and AG are more than 2 times faster than NeuS.

NG and AG are more than 2 times faster than NeuS [14] due to the shallow MLP.

F. Derivation of Frequency Encoding

In the paper, we show that using analytical gradients for higher-order derivatives of multi-resolution hash encoding suffers from gradient locality. We show in this section that Fourier frequency encoding [12], which empowers prior work [14–16] on neural surface reconstruction, does not suffer from such locality issue.

Given a 3D position $\mathbf{x}_i$, let the l-th Fourier frequency encoding be

$$\gamma_l(\mathbf{x}_i) = \big(\sin(2^l \pi \mathbf{x}_i), \cos(2^l \pi \mathbf{x}_i)\big). \tag{2}$$

The derivative of $\gamma_l(\mathbf{x}_i)$ w.r.t. position can thus be calculated as

$$\frac{\partial \gamma_l(\mathbf{x}_i)}{\partial \mathbf{x}_i} = \big(2^l \pi \cdot \cos(2^l \pi \mathbf{x}_i), -2^l \pi \cdot \sin(2^l \pi \mathbf{x}_i)\big). \tag{3}$$

We note that $\frac{\partial \gamma_l(\mathbf{x}_i)}{\partial \mathbf{x}_i}$ is continuous across the space, and thus does not suffer from the gradient locality issue as the multi-resolution hash encoding. Moreover, the position $\mathbf{x}_i$ is present in the derivative, thus allowing for second-order derivatives computation w.r.t. position for the curvature regularization.

While Fourier frequencies encoding is continuous, our coarse-to-fine optimization with varying step size in theory still anneals over the different frequencies when computing higher-order derivatives for more robust optimization. We experiment this idea on the DTU benchmark [3] and observed an improved Chamfer distance: from 0.84 to 0.79. The improvement in surface reconstruction confirms the benefits of using a coarse-to-fine optimization framework.

References

[1] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022. 1, 2, 6

[2] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *arXiv preprint arXiv:2205.15848*, 2022. 1

[3] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 4, 5, 6, 8

[4] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 1, 3

[5] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[7] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 6

[8] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 6

[9] 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*, pages 5589–5599, 2021. 6

[10] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *European Conference on Computer Vision*, pages 615–631. Springer, 2022. 7

[11] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[12] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 8

[13] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 7

[14] 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. 1, 2, 3, 5, 6, 7, 8

[15] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1, 8

[16] 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*, 33:2492–2502, 2020. 1, 6, 7, 8

[17] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021. 6

[18] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Critical regularizations for neural surface reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6270–6279, 2022. 1

[19] Jingyang Zhang, Yao Yao, and Long Quan. Learning signed distance field for multi-view surface reconstruction. *International Conference on Computer Vision (ICCV)*, 2021. 6