Supplementary Material for UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction

Abstract

*In this **supplementary document**, we first formally show that volume rendering converges to surface rendering in the limit of small intervals and a large number of query points (Section 1). Next, we provide more information about the experimental setups for all datasets (Section 2) and provide more details about the baselines (Section 3). Next, we discuss the limiting factors of our method and possible solutions (Section 4). Finally, we present additional results on the DTU dataset, indoor scenes from SceneNet and the BlendenMVS dataset (Section 5).*

1. Convergence Proof

We consider the volume and surface rendering equations

$$\hat{C}_v(\mathbf{r}) = \sum_{i=1}^{N} o_\theta(\mathbf{x}_i) \prod_{j<i} \big(1 - o_\theta(\mathbf{x}_j)\big) c_\theta(\mathbf{x}_i, \mathbf{n}_i, \mathbf{h}_i, \mathbf{d}) \tag{1}$$

$$\hat{C}_s(\mathbf{r}) = c_\theta(\mathbf{x}_s, \mathbf{n}_s, \mathbf{h}_s, \mathbf{d}) \tag{2}$$

with occupancy field o_θ and color field c_θ. The samples $\mathbf{x}_i = \mathbf{o} + t_i \mathbf{d}$ are obtained by drawing N depth values t_i randomly from the interval $[t_s - \Delta, t_s + \Delta]$ centered at t_s. For notation clarity, we will drop the model parameters θ in the following. Without loss of generality, we will consider c as function of $\mathbf{x}$ only as $\mathbf{n}$ and $\mathbf{h}$ are functions of the sample point $\mathbf{x}$ and a single color channel, i.e., $c(\mathbf{x}) \in \mathbb{R}$:

$$\hat{C}_v(\mathbf{r}) = \sum_{i=1}^{N} o(\mathbf{x}_i) \prod_{j<i} \big(1 - o(\mathbf{x}_j)\big) c(\mathbf{x}_i) \tag{3}$$

$$\hat{C}_s(\mathbf{r}) = c(\mathbf{x}_s) \tag{4}$$

We will now show that $\hat{C}_v(\mathbf{r})$ approaches $\hat{C}_s(\mathbf{r})$ for $\Delta \to 0$ and $N \to \infty$.

Theorem 1. *Assuming Multi-layer Perceptrons for the occupancy and color fields with Softplus and ReLU activation functions, respectively, as well as Fourier features at k octaves, we have*

$$\lim_{\substack{\Delta \to 0 \\ N \to \infty}} \hat{C}_v(\mathbf{r}) = \hat{C}_s(\mathbf{r}) \tag{5}$$

Thus, volume and surface rendering become equivalent when reducing the interval and increasing the number of samples.

Proof. Linear layers, Softplus, Sigmoid and ReLU activation functions as well as Fourier features are Lipschitz continuous. Compositions of Lipschitz continuous functions are Lipschitz continuous. Thus, $o(\mathbf{x})$ and $c(\mathbf{x})$ are Lipschitz continuous wrt. $\mathbf{x}$. Let k_o and k_c denote the respective Lipschitz constants. As $o(\mathbf{x}_s) = 0.5$, we have

$$|o(\mathbf{x}_i) - 0.5| \leq k_o \|\mathbf{x}_i - \mathbf{x}_s\|_2 = \xi$$
$$|c(\mathbf{x}_i) - c(\mathbf{x}_s)| \leq k_s \|\mathbf{x}_i - \mathbf{x}_s\|_2 = \gamma$$

with $\xi \to 0$ and $\gamma \to 0$ for $\Delta \to 0$. Further, $o(\mathbf{x}) \in (0, 1)$, so that $\xi < 0.5$. Furthermore:

$$\lim_{\substack{\xi \to 0 \\ N \to \infty}} \sum_{i=1}^{N} (0.5 + \xi)^i = 1 \tag{6}$$

as the limits are interchangeable:

$$\begin{aligned}
\lim_{\xi \to 0} \lim_{N \to \infty} \sum_{i=1}^{N} (0.5 + \xi)^i &= 1 \\
\lim_{\xi \to 0} \sum_{i=1}^{\infty} (0.5 + \xi)^i &= 1 \\
\lim_{\xi \to 0} \left(1 + \frac{2\xi}{0.5 + \xi}\right) &= 1
\end{aligned}$$

as well as

$$\begin{aligned}
\lim_{N \to \infty} \lim_{\xi \to 0} \sum_{i=1}^{N} (0.5 + \xi)^i &= 1 \\
\lim_{N \to \infty} \sum_{i=1}^{N} 0.5^i &= 1
\end{aligned}$$

Therefore, for any $\epsilon > 0$, there exists $\delta > 0$ and $N > 0$ such that:

$$\begin{aligned}
|\xi| < \delta \Rightarrow & \left| \sum_{i=1}^{N} (0.5 + \xi)^i - 1 \right| < \epsilon \\
\Rightarrow & \left| \sum_{i=1}^{N} (0.5 + \xi)^i c(\mathbf{x}_s) - c(\mathbf{x}_s) \right| < \epsilon c(\mathbf{x}_s) \\
\Rightarrow & \left| \sum_{i=1}^{N} (0.5 + \xi)^i \left(c(\mathbf{x}_s) + \gamma\right) - c(\mathbf{x}_s) \right| < \epsilon c(\mathbf{x}_s) + \left| \gamma \sum_{i=1}^{N} (0.5 + \xi)^i \right| \\
\Rightarrow & \left| \sum_{i=1}^{N} o(\mathbf{x}_i) \prod_{j<i} \big(1 - o(\mathbf{x}_j)\big)\, c(\mathbf{x}_i) - c(\mathbf{x}_s) \right| < \epsilon c(\mathbf{x}_s) + \left| \gamma \sum_{i=1}^{N} (0.5 + \xi)^i \right|
\end{aligned}$$

Since $\gamma \to 0$ for $\Delta \to 0$, the right-hand term can become arbitrarily small, completing the proof.

□

Remark: While $N \to \infty$ might appear as a strong assumption, we remark that even for moderate values of N as those used in our experiments $\sum_{i=1}^{N} 0.5^i$ becomes very close to 1 very quickly.

2. Experimental Setup

In this section, we provide more details about the experimental setup of all experiments. We focus here specifically on the spatial definition of the scenes.

2.1. DTU MVS dataset

Previous works [2, 4] consider a unit cube or unit sphere as the region of interest for 3D reconstruction of DTU objects. These methods reconstruct objects using object masks, where the respective visual hull lies inside the unit cube/sphere. As we do not consider masks in our work, we need to model the entire scene within the field of view e.g. including the table. Hence, we define a larger region of interest. We also consider a sphere as the region of interest but with a four-times larger radius. Our ray evaluations take place only inside the area of interest. Note that, our model is not sensitive to this choice, but an overly large area of interest requires adaptation of the sampling accuracy for the root-finding, and a too-small region can not represent the whole scene. As we assume to cover the entire scene inside this region of interest, we consider the background as black.

While we do not consider masks during optimization, we evaluate all methods only inside the mask areas (inside the visual hull). The reason of doing so is that the visual hull is the only region that is represented by all methods including IDR. Hence, we evaluate all methods inside this area. With this procedure, we can guarantee a fair comparison among all baselines and our method. More qualitative results for the *DTU MVS* dataset are shown in Fig. 3.

2.2. Indoor Scene from SceneNet

For the indoor scene, we use COLMAP [3] to obtain the camera extrinsics and intrinsics. We then define the area of interest such that all camera locations are inside a sphere with the sphere's center approximately at the scene's center. As before, we assume a black background. In addition to the main paper, we show the result of one more scene in Fig. 5.

2.3. BlendedMVS

Besides the datasets mentioned previously where we either have a black background (DTU) or a closed scene definition (indoor scene), we also consider the BlendedMVS dataset that has scenes with more complex backgrounds. Here, the multi-view images contain objects as well as complex backgrounds that can be located further away or appear blurred. Since the BlendedMVS dataset consists of a large variety of scene layouts, we must model not only the foreground but also the background of the scenes.

To this end, we found that the setup of NeRF++ [6] is useful for extending our model to complex backgrounds. The main idea is to model foreground and background using two separate models by spatially separating the representations. Similarly, we define the area of interest for reconstruction as a sphere that covers all cameras centered at an approximate scene center. Within this sphere, we use our model for representing the scene. Everything that is located outside, we represent with a NeRF model that has an *inverted sphere parameterization*. We refer the reader to [6] for more details. The inverted sphere parameterization allows for both, representing far-away background as well as background elements that are closer to the area of interest.

For rendering during the optimization process, we apply our rendering procedure for each ray with 64 samples inside the interval and 32 samples in the free space between the camera and interval bound. Furthermore, we uniformly sample 32 points outside of the sphere to roughly capture the background. We use both sets of sampled points for volume rendering. As we aim for 3D reconstruction inside the area of interest, we use significantly fewer samples for capturing the background as used in [6]. This results in less computational effort, while the model is still able to separate foreground and background properly. Additional results are shown in Fig. 5.

2.4. LLFF Dataset

We also test our method on samples of the LLFF dataset that contains forward-facing scenes. First, we use COLMAP to obtain camera extrinsics and intrinsics and define the area of interest of the scenes. In Fig. 6, we provide the reconstruction results. Even though the scenes contain complicated shapes and viewpoints with only small variations, our methods can capture accurate surfaces.

3. Baselines

This section provides additional details about the baselines we compare against.

threshold σ	1	5	10	**50**	100	500
scan65	2.29	1.53	1.26	1.27	1.80	3.15
scan105	3.46	2.27	1.85	1.07	1.32	5.99
scan114	2.88	1.74	1.37	1.06	1.21	2.86

Table 1: **Volume Density Thresholds of NeRF.** We show the Chamfer distance for meshes extracted from a trained NeRF model considering different density thresholds σ. We applied this analysis for three models of the DTU dataset and found that a threshold $\sigma = 50$ leads to the best overall performance.

	NeRF	NeRF no view
scan24	**1.90**	2.37
scan37	**1.60**	2.56
scan40	1.85	**1.84**
scan55	**0.58**	0.86
scan63	**2.28**	3.49
scan65	1.27	**1.24**
scan69	**1.47**	-
scan83	**1.67**	2.07
scan97	**2.05**	2.30
scan105	**1.07**	1.80
scan106	**0.88**	1.05
scan110	**2.53**	5.09
scan114	**1.06**	1.22
scan118	**1.15**	1.22
scan122	**0.96**	1.44
mean	**1.49**	2.04

Table 2: **NeRF Ablation.** We compare the original NeRF against a NeRF model without viewing dependence. The viewing dependence significantly improves the reconstruction abilites.

IDR: For the DTU evaluations, we use the official code[1] and the provided pre-trained models. To test IDR without mask supervision, we set the weight of the mask loss to zero and consider the RGB loss for all rays intersecting surfaces.

COLMAP: We use the same COLMAP procedure as reported in previous works [2,4]. Therefore, the original COLMAP [3] implementation is used to output meshes with the following steps: a) `exhaustive_matcher`, b) `point_triangulator`, c) `patch_match_stereo`, d) `stereo_fusion` and e) `poisson_mesher`. We follow the reference example from the official implementation[2]. We choose the trim-parameter according to the study in [2] which shows that trim 7 leads to the best performance, however results in non-watertight meshes.

NeRF: For the NeRF baseline, we adapt the Pytorch reimplementation [5] to our framework. We apply the NeRF model to the same scene setups as it is used for our method. For extracting meshes we need to choose a threshold for the volume density. Therefore we evaluate NeRF for three DTU objects and five different threshold parameters σ.

In Table 1, we provide a quantitative comparison that shows a superior performance at $\sigma = 50$. In Fig. 1, we depict qualitative results for different threshold parameters. We see that small thresholds, e.g. 1 and 10, lead to bloated reconstructions, while the surface for $\sigma > 50$ shows significantly more missing regions. The qualitative comparison verifies our findings from our quantitative evaluation, and hence, we choose 50 as the threshold for all evaluations in the paper.

Suggested by a reviewer, we analyze the NeRF baseline wrt. to the viewing dependency of the radiance field. We find that the viewing dependency improves the reconstruction capabilities, see Table 2.

[1] https://github.com/lioryariv/idr

[2] https://colmap.github.io/cli.html#example

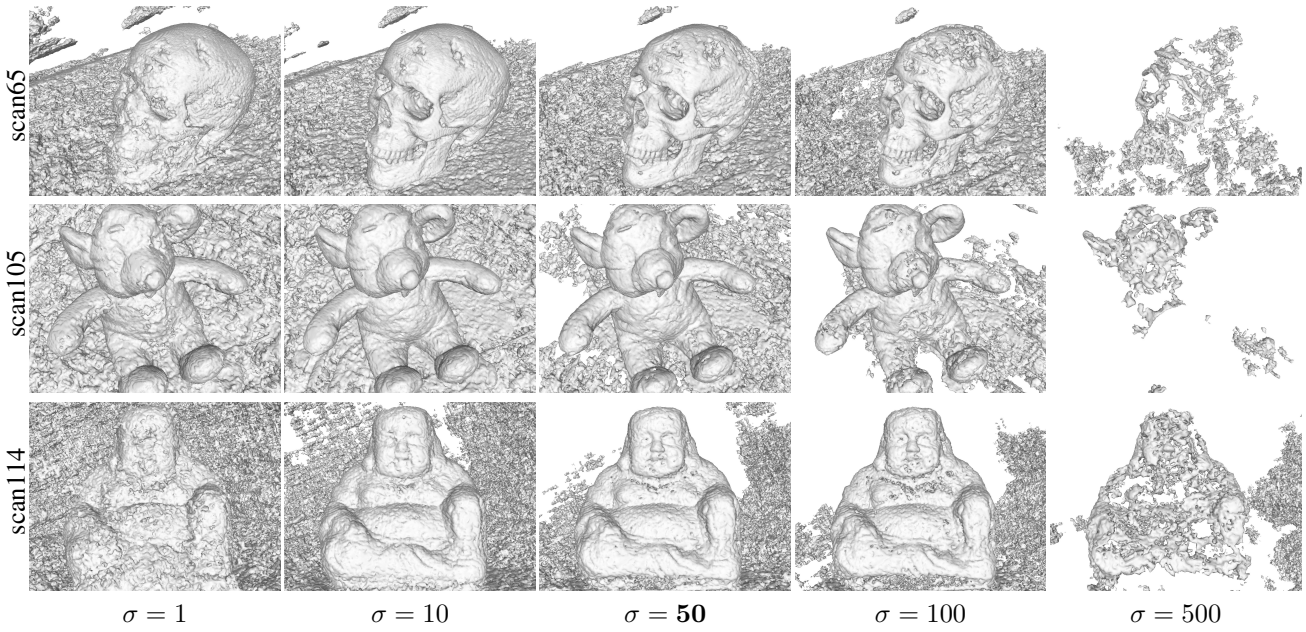


Figure 1: **Volume Density Thresholds of NeRF.** We show a qualitative comparison of meshes extracted from NeRF for different thresholds of the volume density. We see that the extracted meshes significantly differ for various threshold parameter and identify a threshold of 50 as superior, qualitatively and quantitatively

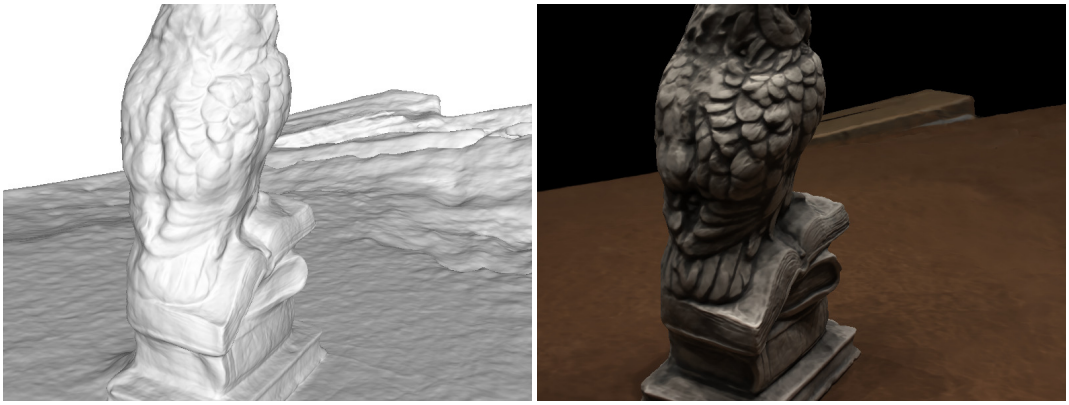


Figure 2: **rarely visible regions.** In this figure, we show an extreme view from the input images. The table area in the back is reconstructed poorly due to missing observations of this surface.

4. Limitations

Overexposed regions: Challenging for nearly all multi-view reconstruction methods are overexposed and textureless image regions. In the DTU dataset, the table appears completely white without any texture in some of the scenes, e.g., second row in Fig. 3. While this problematic region is masked during training of the IDR baseline, our method and the other baselines do not utilize pixel-accurate masks and hence struggle predicting accurate surfaces in these areas. Surfaces extracted by NeRF are typically non-smooth with numerous geometric artifacts. Our method also struggles to output smooth surfaces even though we use a surface regularizer during optimization. We attribute this to the inherent ambiguity of the texture-less overexposed regions that can not be entirely resolved by such simple prior assumptions, encouraging the development of more advanced priors and implicit models that are able to expose their uncertainty in the future.

Rarely visible regions: Similar difficulties arise from rarely visible regions in the input views. In Fig. 2, we show an extreme

view from the dataset, where we see a rarely visible region in the back. In this area, the reconstructed surface of the table appears to be less accurate compared to the rest of the table which is a consequence of missing observations.

Shape-Appearance Ambiguity: For reconstructing neural surfaces from multi-view images, it is necessary to optimize neural implicit surfaces and the appearance representations at the same time. The network architecture and the optimization procedure yield an inductive bias whether predicted views are explained by adapting the shape or the view-dependent appearance. In the bottom row in Fig. 3, we show an example where our method is not able to correctly model the geometry and our model explains the inner part with the view-dependent appearance instead, hence not reconstructing the inner part correctly. NeRF does not show this behavior as its capacity for modeling view-dependent effects is much smaller compared to our model. Hence, for objects with weakly view-dependent appearance, it NeRF less prone to this behavior. However, this also limits NeRF's capabilities in modeling plausible geometry and strong view-dependent effects. IDR can resolve this particular example due to its strong mask supervision.

Possible Solution: To circumvent these limiting cases, we hypothesize that a learning-based prior should be applied to resolve the underlying ambiguities. In future work, we, therefore, consider learning a probabilistic neural surface model which captures regularities and uncertainty across objects. We believe that such priors will help to resolve the ambiguities in texture-less areas, rarely visible regions as well as the aforementioned shape-appearance ambiguity.

5. Additional Results

We show additional qualitative results for the *DTU MVS* dataset in Fig. 3. Fig. 4 shows the geometry and appearance from different views of the objects. In Fig. 5, we show reconstructions on more scenes from *SceneNet* and *BlendedMVS*.

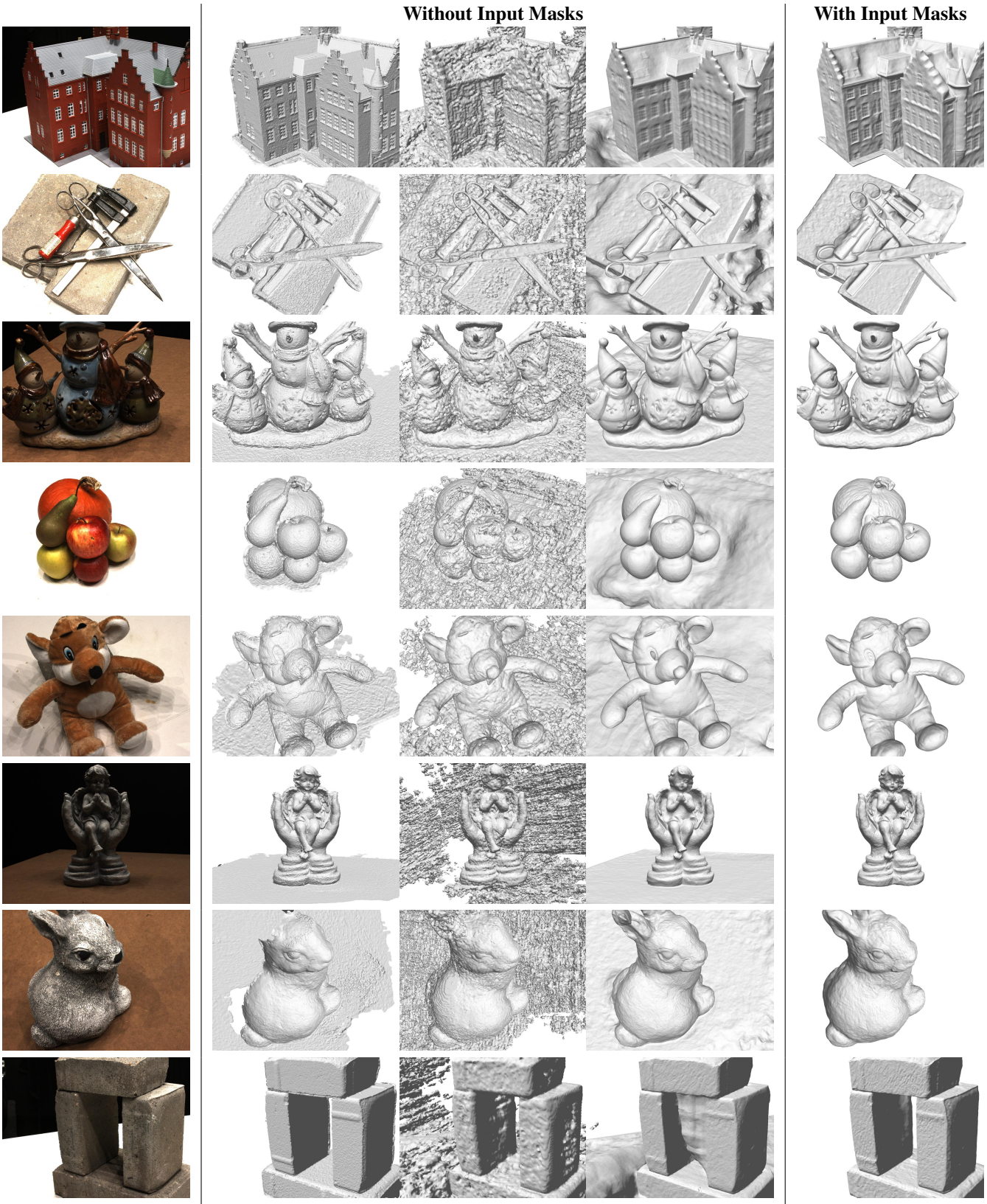
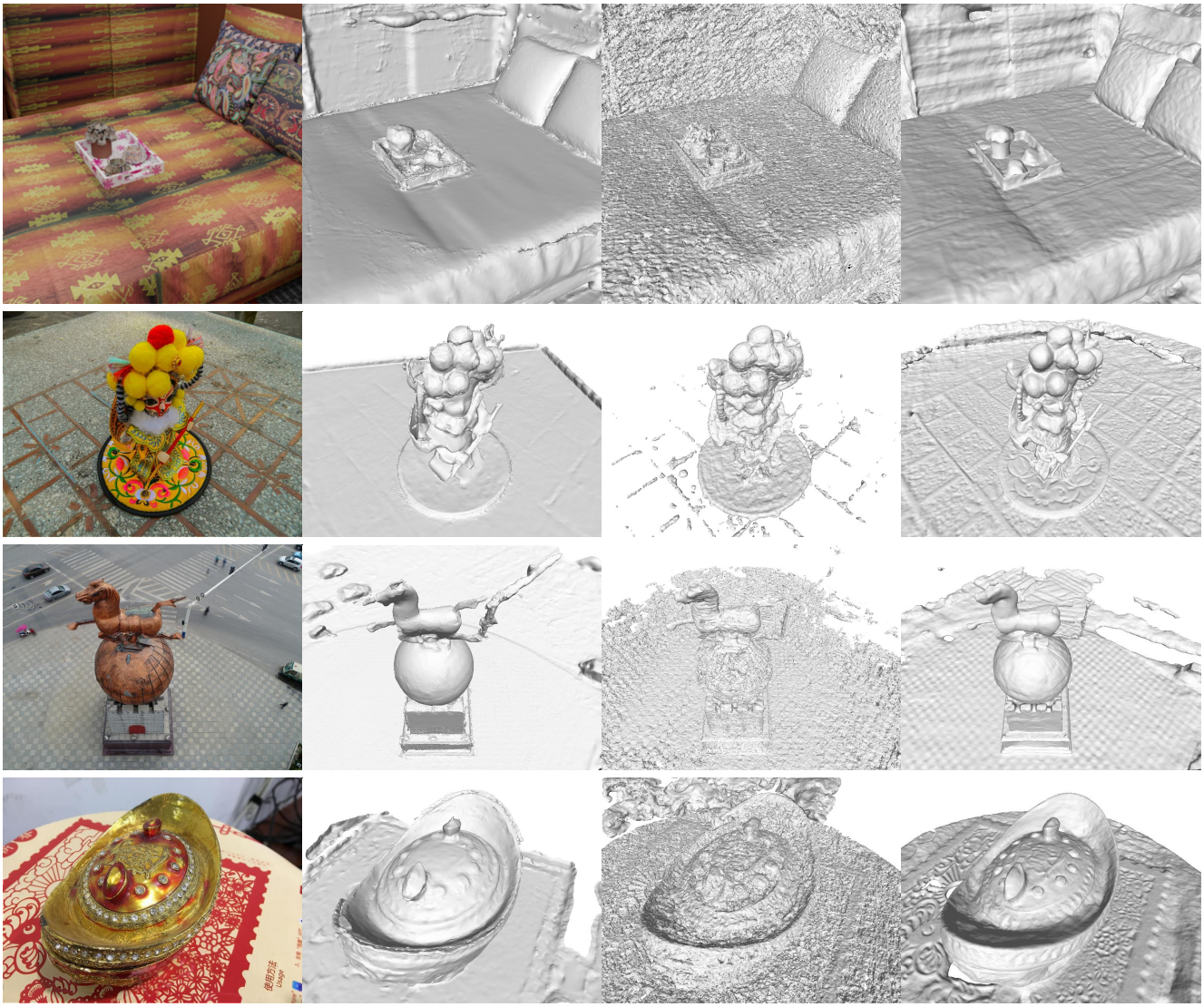


Figure 3: **DTU MVS dataset.** Additional results for objects from the *DTU MVS* dataset.

Figure 4: **Rotations.** In this figure, we show different views from DTU objects and the respective surfaces.



GT view COLMAP [3] NeRF [1] Ours

Figure 5: **Indoor Scene and BlendedMVS Scenes.** We show additional results for one indoor scene (first arrow) and objects from the *Blended MVS* dataset.

GT view **Ours** GT view **Ours**

Figure 6: **Results on LLFF.** Zoom in for details.

References

[1] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 7, 9

[2] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 4

[3] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 3, 4, 7, 9

[4] Lior Yariv, Matan Atzmon, and Yaron Lipman. Universal differentiable renderer for implicit neural representations. *arXiv.org*, 2003.09852, 2020. 3, 4, 7

[5] Lin Yen-Chen. PyTorchNeRF: a PyTorch implementation of NeRF, 2020. 4

[6] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv.org*, 2020. 3