Geo-Neus: Geometry-Consistent Neural Implicit Surfaces Learning for Multi-view Reconstruction –Supplementary Material–

In this **supplementary document**, we first provide the details of the depth integral used in Geometric bias of volumetric integration in Section A. In Section B, we show the comparison results of other baseline methods (such as Points2Surf [1] and DVR [2]) on DTU. Next, we provide more analysis of our method in Section C and provide the rendering quality of our method in Section D. Additional results on DTU and BlendedMVS dataset can be found in Section E. We discuss the performance of our method with sparse input views in Section F. Finally, we try to evaluate the bias with our proposed losses in Section G.

A Details of the depth integral used in Geometric bias of volumetric integration

In our main paper, we use the depth integral to demonstrate the geometric bias of volumetric integration. Specifically, in volume rendering, the expected color is calculated as: $\hat{C} = \sum_{i=1}^{n} w\left(t_i\right) \hat{c}\left(t_i\right)$, where $w(t_i)$ is the weight converted from the SDF value of point t_i and $\hat{c}\left(t_i\right)$ is the predicted color value at point t_i. Similarly, we calculate the expected depth $\hat{d}$ as: $\hat{d} = \sum_{i=1}^{n} w\left(t_i\right) d\left(t_i\right)$, where $d\left(t_i\right)$ is the distance between the point t_i and the image plane.

B Comparison with other baselines

In this section, we show the comparison results of other baseline methods on DTU. Note that, since our method uses sparse 3D points as supervision signal, it is very interesting to explore if existing implicit surface reconstruction methods from point clouds can be used to reconstruct reasonable surfaces in this special case. To this aim, we select Points2Surf [1], a method can generalize well to new shapes, to conduct experiments on DTU sparse 3D points. We show experiments in Fig 1. As can be seen, the sparse 3D points from SFM methods are distributed very irregularly, which poses great challenges to existing reconstruction methods. Therefore, the performance of Points2Surf degrades in this very challenging case. In addition, we also show the reconstruction results of DVR in Table 1. It shows that our method surpasses DVR a lot.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
DVR	4.10	4.54	4.24	2.61	4.34	2.81	2.53	2.93	3.03	3.24	2.51	4.80	3.09	1.63	1.58	3.20
Geo-Neus	**0.375**	**0.537**	**0.336**	**0.357**	**0.800**	**0.454**	**0.408**	**1.032**	**0.843**	**0.548**	**0.460**	**0.473**	**0.294**	**0.355**	**0.345**	**0.508**

Table 1: Reconstruction results on DTU.

C More analysis

C.1 GPU memory consumption

We show the GPU memory consumption of our proposed strategies in Table 2. We observe that, compared with the Baseline, the SDF loss function barely introduces extra GPU memory consumption

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

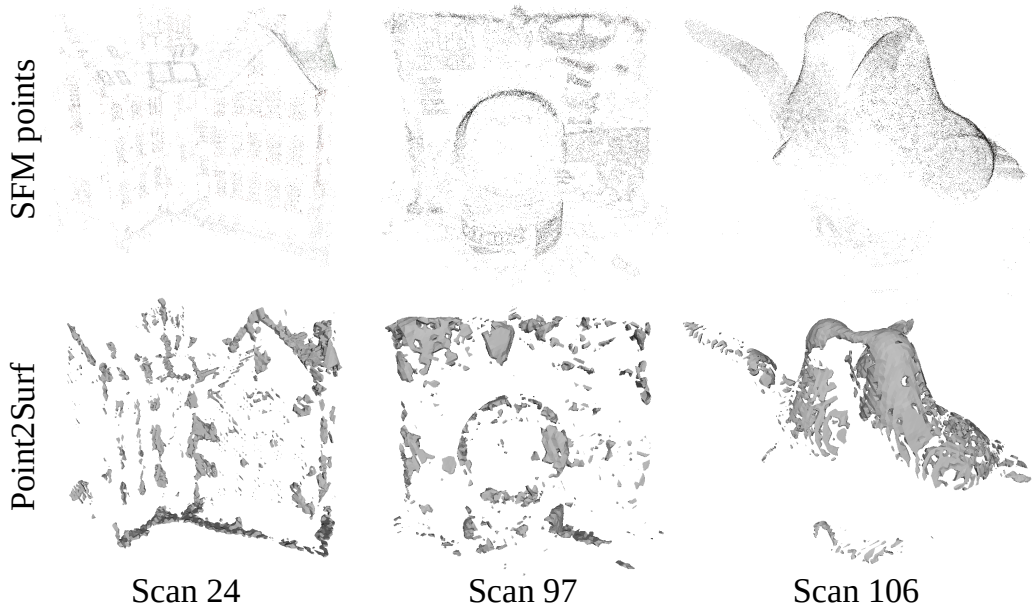

Figure 1: Surface reconstruction from SFM points.

while the photometric loss function only causes an 8% increase in GPU memory consumption. This demonstrates that both our introduced loss functions are GPU memory-friendly to neural surfaces learning methods.

Method	$\mathcal{L}_{color}$	$\mathcal{L}_{SDF}$	$\mathcal{L}_{photo}$	GPU memory [M]
Baseline	✓			7033
Model-A	✓	✓		7045
Model-B	✓		✓	7615
Geo-Neus	✓	✓	✓	7619

Table 2: GPU memory consumption.

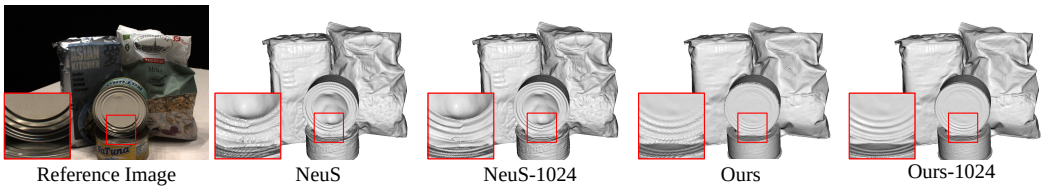

Figure 2: Qualitative comparisons of surfaces reconstructed with higher volume resolution.

C.2 Reconstruction with higher volume resolution

After network training, the surface is usually extracted from the SDF values in a predefined bounding box by the Marching Cube algorithm with the volume size of 512^3. In practice, we find that the resolution of 512 limits the reconstruction quality of our method. Fig. 2 shows the reconstruction results with the resolution of 1024 on DTU Scan 97. With Marching Cubes of higher resolution, more details can be preserved by our method. This is because our proposed method can locate sub-pixel zero-level set of SDF values by our interpolation operation on the SDF values. Our quantitative results are reported in Table 3. There is no prominent improvement in terms of the evaluation metric due to the limitation of the ground truth resolution, but it can be observed in Fig. 2 that the surface quality of our method is significantly enhanced.

Scan	NeuS	NeuS-1024	Ours	Ours-1024
97	1.06	1.01	0.843	0.840

Table 3: Reconstruction results with higher volume resolution.

C.3 The robustness to noisy SFM sparse points

In our main paper, we use a strict radius filter to remove the noise from sparse 3D points from SFM methods. Here we show the performance of our method with the raw sparse 3D points from SFM methods. With these sparse 3D points, we train our model from scratch on DTU scan 24, 37 and 40. The results in Tabel 4 show that our method could also perform well with noisy sparse points directly from SFM methods. This shows the robustness of our method.

settings	scan24	scan37	scan40
unfiltered	0.42	0.63	**0.34**
filtered	**0.38**	**0.54**	**0.34**

Table 4: Reconstruction results with unfiltered and filtered sparse 3D points.

C.4 The effect of linear interpolation in surface points extraction

We perform an ablation study on the surface points extraction methods used in the multi-view photometric consistency constraints. We replace our linear interpolation by the hierarchical sampling used in NeuS to extract the surface points. We show the evaluation results in Table 5. As can be seen, the performance of hierarchical sampling is worse than that of linear interpolation. This further validates the gap between the volume rendering and SDF modeling, supporting our assumption and theoretical analysis.

settings	scan24	scan37
hierarchical sampling	0.537	0.677
linear interpolation	**0.375**	**0.537**

Table 5: Comparison of hierarchical sampling and linear interpolation in surface points extraction.

C.5 View-aware SDF loss

We perform an ablation study on the construction of SDF loss. To handle occlusions, we use view-aware SDF loss as supervision. Given sparse points of SFM, we also try to randomly sample a subset of the sparse points and use it as the SDF supervision. The results in Table 6 show that the view-aware sampling strategy performs better than the random sampling strategy. This verifies the effectiveness of view-aware sampling.

settings	scan24	scan37
random sampling	0.43	0.58
view-aware sampling	**0.38**	**0.54**

Table 6: Comparison of random sampling and view-aware sampling in SDF loss construction.

C.6 Grey-scale images vs. RGB images for photometric consistency loss

In our main paper, we use grey-scale images to construct photometric consistency loss for less time and memory consumption. We also try to use RGB images to construct photometric consistency loss and train our model on DTU scan24. The results in Table 7 show that the performance of our method with RGB images degrades a little bit. We think this is because grey-scale images may reflect more geometric information compared with RGB images.

settings	Chamfer distance [mm]	Running time [h]	GPU memory [G]
RGB images	0.44	24	8.1
grey-scale images	0.38	16	7.6

Table 7: Comparison of grey-scale images and RGB images in photometric consistency computation on DTU scan24.

C.7 Other metrics for photometric consistency loss

In our main paper, we use NCC to compute the photometric consistency loss. Here we investigate other metrics to measure the photometric consistency loss. We replace the NCC by SSIM [3] to retrain our model from scratch on DTU scan24. The performance of using SSIM becomes 0.408, which a little worse than that of using NCC, 0.375.

D Rendering quality

We report PSNR of train image reconstructions on DTU for reference. Quantitative and qualitative results are shown in Table 8 and Fig. 3, respectively. These results show that our method achieves similar rendering performance with the Baseline, NeuS. This is because we focus on reconstructing the geometry of the scenes and do not introduce special mechanisms for rendering optimization. Therefore, our proposed method can facilitate geometry reconstruction without degrading rendering quality.

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
NeRF	26.24	25.74	26.79	27.57	31.96	31.50	29.58	32.78	28.35	32.08	33.49	31.54	31.00	35.59	35.51	30.65
IDR	23.29	21.36	24.39	22.96	23.22	23.94	20.34	21.87	22.95	22.71	22.81	21.26	25.35	23.54	27.98	23.20
VolSDF	26.28	25.61	26.55	26.76	31.57	31.50	29.38	33.23	28.03	32.13	33.16	31.49	30.33	34.90	34.75	30.38
NeuS	28.20	**27.10**	**28.13**	**28.80**	32.05	33.75	**30.96**	**34.47**	29.57	**32.98**	**35.07**	**32.74**	**31.69**	**36.97**	**37.07**	**31.97**
Geo-Neus	**28.49**	27.09	27.78	28.51	**33.37**	**33.78**	30.76	34.12	**29.94**	33.35	34.53	32.50	31.12	36.42	36.85	31.91

Table 8: Rendering results on DTU.

E Additional experimental results

E.1 Detailed ablation results with our proposed strategies

We present the detailed results of the ablation study in Table 9. As can be seen, our proposed strategies almost greatly boost the reconstruction of each scan from DTU dataset, thus leading to significant overall performance improvement.

Method	$\mathcal{L}_{color}$	$\mathcal{L}_{SDF}$	$\mathcal{L}_{photo}$	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
Baseline	✓			1.37	1.21	0.73	0.40	1.20	0.70	0.72	**1.01**	1.16	0.82	0.66	1.69	0.39	0.49	0.51	0.87
Model-A	✓	✓		0.56	0.68	0.51	0.37	0.82	0.48	0.51	1.21	1.13	0.65	0.50	0.77	0.31	0.44	0.41	0.62
Model-B	✓		✓	0.46	0.54	0.34	0.39	0.87	0.46	0.42	1.13	0.88	0.58	0.49	0.47	0.30	0.37	0.38	0.54
Geo-Neus	✓	✓	✓	**0.38**	**0.54**	**0.34**	**0.36**	**0.80**	**0.45**	**0.41**	1.03	**0.84**	**0.55**	**0.46**	**0.47**	**0.29**	**0.36**	**0.34**	**0.51**

Table 9: Detailed results of ablation study on DTU.

E.2 Additional qualitative results

We show additional qualitative results for the DTU dataset in Fig. 4 and Fig. 5. In Fig. 6, we show reconstructions on more scenes from BlendedMVS dataset.

F Reconstruction with sparse input views

In this section, we show our proposed method can reconstruct satisfactory surfaces with sparse input views. Specifically, we take 3 views as input and test our method and NeuS in this challenging case. Note that, in this situation, our method only uses these 3 input views to extract SFM sparse points and compute the SDF loss. Qualitative and quantitative results are shown in Table 10 and Fig. 7. In

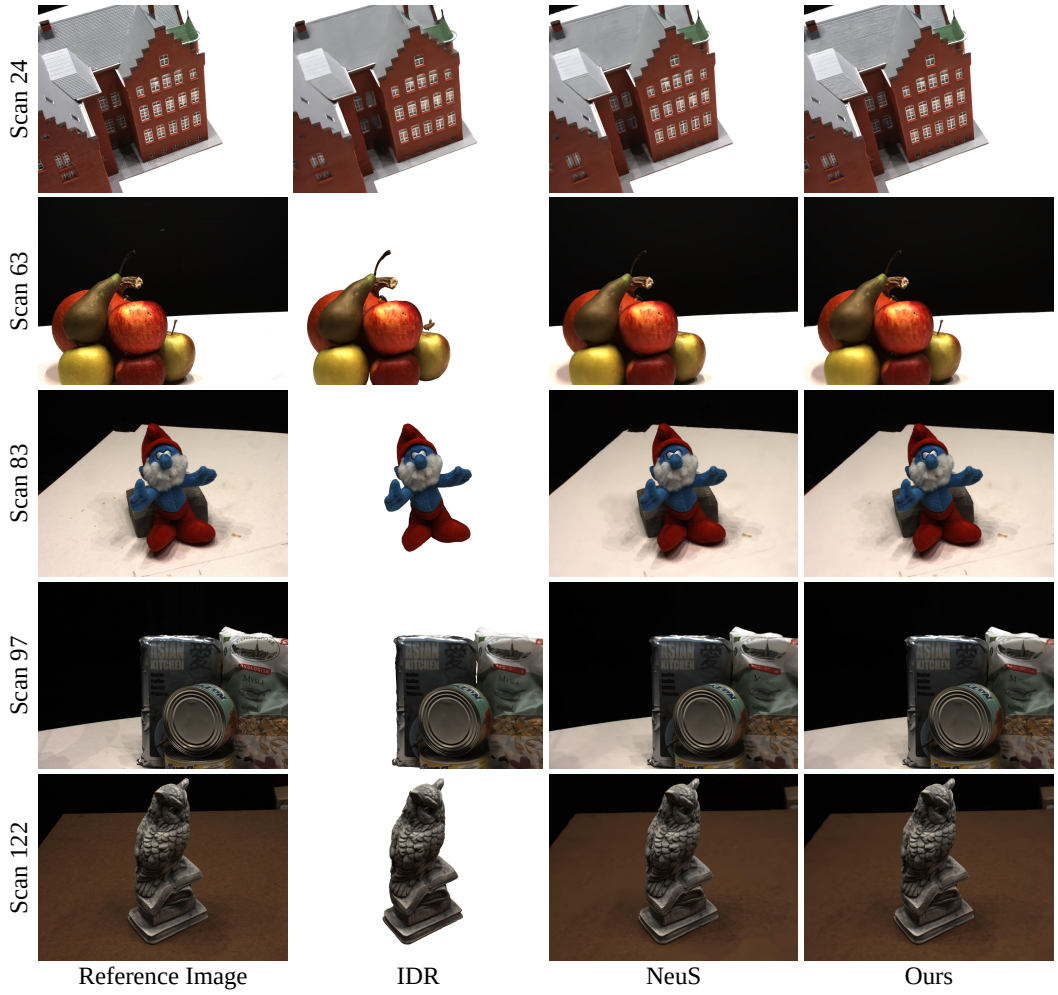

Figure 3: Rendered images on DTU.

addition, We use all input images to test their rendering quality in terms of PSNR. We surprisingly find that our method can still achieve satisfactory results while the performance of NeuS degrades a lot in this challenging case. We think that this is because our proposed explicit geometry constraints can still help regularize the SDF network in this situation. Therefore, our method can still converge with sparser training cameras.

metric	method	scan97	scan106	scan118
Chamfer distance	NeuS	1.75	1.85	3.59
	Ours	**1.045**	**0.782**	**0.855**
PSNR	NeuS	10.70	17.27	16.82
	Ours	**15.26**	**22.48**	**25.78**

Table 10: Reconstruction with 3 views.

G Evaluation on bias with the proposed losses

In this section, we try to evaluate the bias with proposed loss below. For the rendering color of network, we have the formula:

$$C = w\left(t_j\right)\hat{c}\left(\hat{t^*}\right) + \varepsilon_{sample} + \varepsilon_{weight}. \tag{1}$$

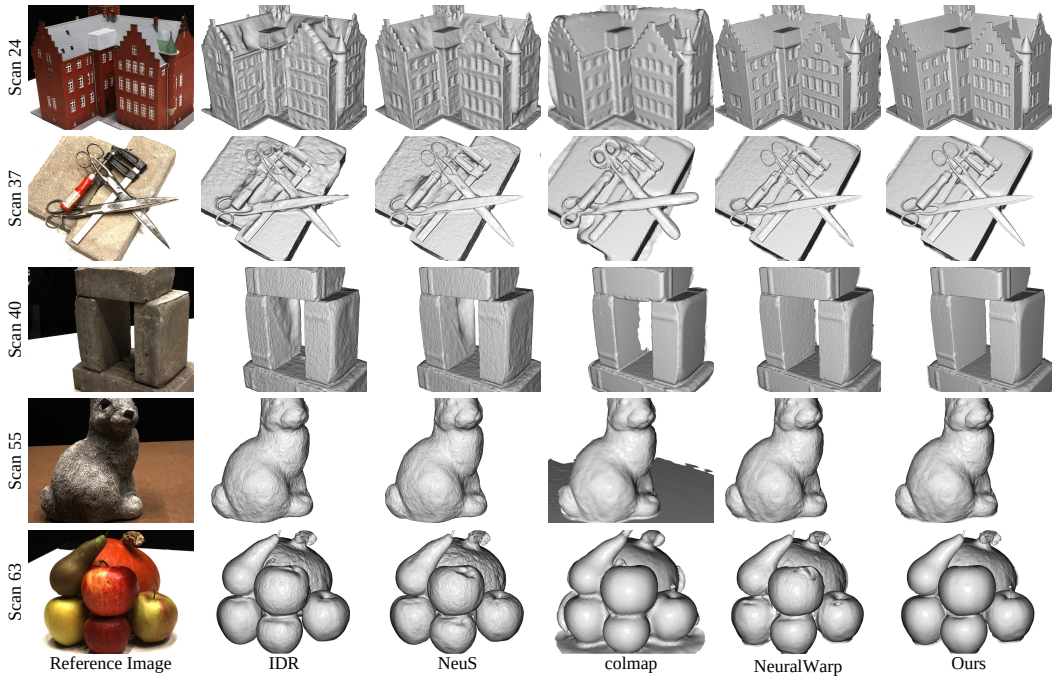

Figure 4: Surfaces reconstructed on DTU (1/2).

To consider the simple case in which the ray intersects once with the surface. We assume that the weight function can be written as $\phi(\hat{sdf}(t))$ and reaches its max at $\hat{sdf}(t) = 0$. The assumption is reasonable and used by related methods such as NeuS, VolSDF, etc. In formula (1), t_j is the nearest sample point from predict intersection point $\hat{t^*}$. With the first order of approximation, the weight $w(t_j)$ can be written as:

$$w\left(t_j\right) = w\left(\hat{t^*} + \delta\right) = w(\hat{t^*}) + \frac{d\phi}{d\hat{sdf}}(\hat{sdf}(\hat{t^*}))\frac{d\hat{sdf}}{dt}(\hat{t^*})\delta = w(\hat{t^*}), \tag{2}$$

where $\frac{d\phi}{d\hat{sdf}}(\hat{sdf}(\hat{t^*})) = \frac{d\phi}{dx}(0) = 0$. In this way, the formula (1) can be written as:

$$C = w\left(\hat{t^*}\right)\hat{c}\left(\hat{t^*}\right) + \varepsilon_{sample} + \varepsilon_{weight}. \tag{3}$$

For a sparse point p_i, we assume there is a ray V_i passing through p_i. Following our assumption that p_i is on the surface, we get the intersection point t^*:

$$\hat{sdf}(t^*) = sdf(t^*) = sdf(p_i) = 0. \tag{4}$$

That is, the intersection predicted by SDF network is the same with the real intersection: $\hat{t^*} = t^*$. Then the formula (3) is:

$$C = w\left(t^*\right)\hat{c}\left(t^*\right) + \varepsilon_{sample} + \varepsilon_{weight}. \tag{5}$$

With the help of proposed losses, the network could better simulate the real color field. But the sample bias and the weight bias still exist because of volume rendering.

References

[1] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 108–124. Springer, 2020.

[2] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

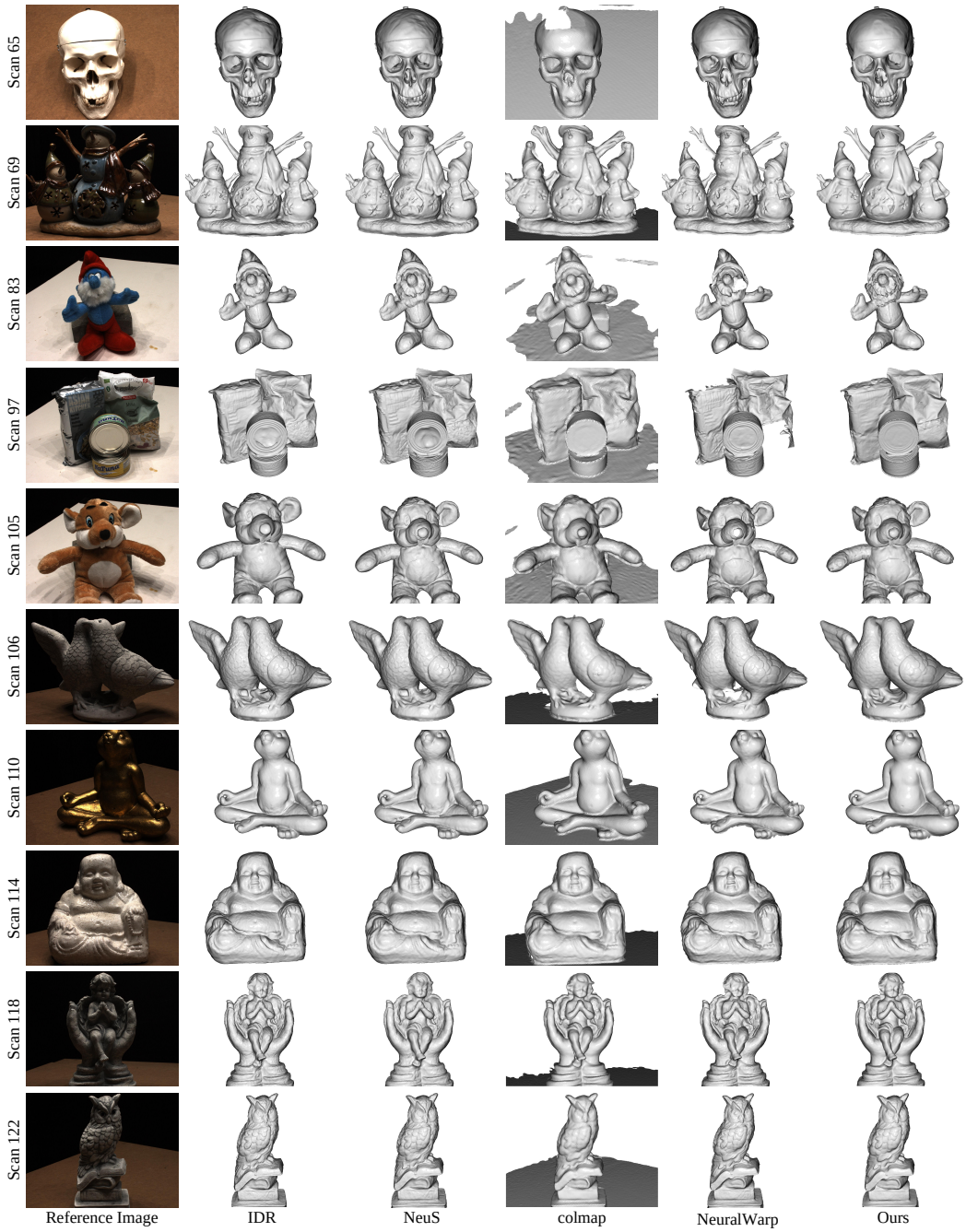

Figure 5: Surfaces reconstructed on DTU (2/2).

[3] 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.

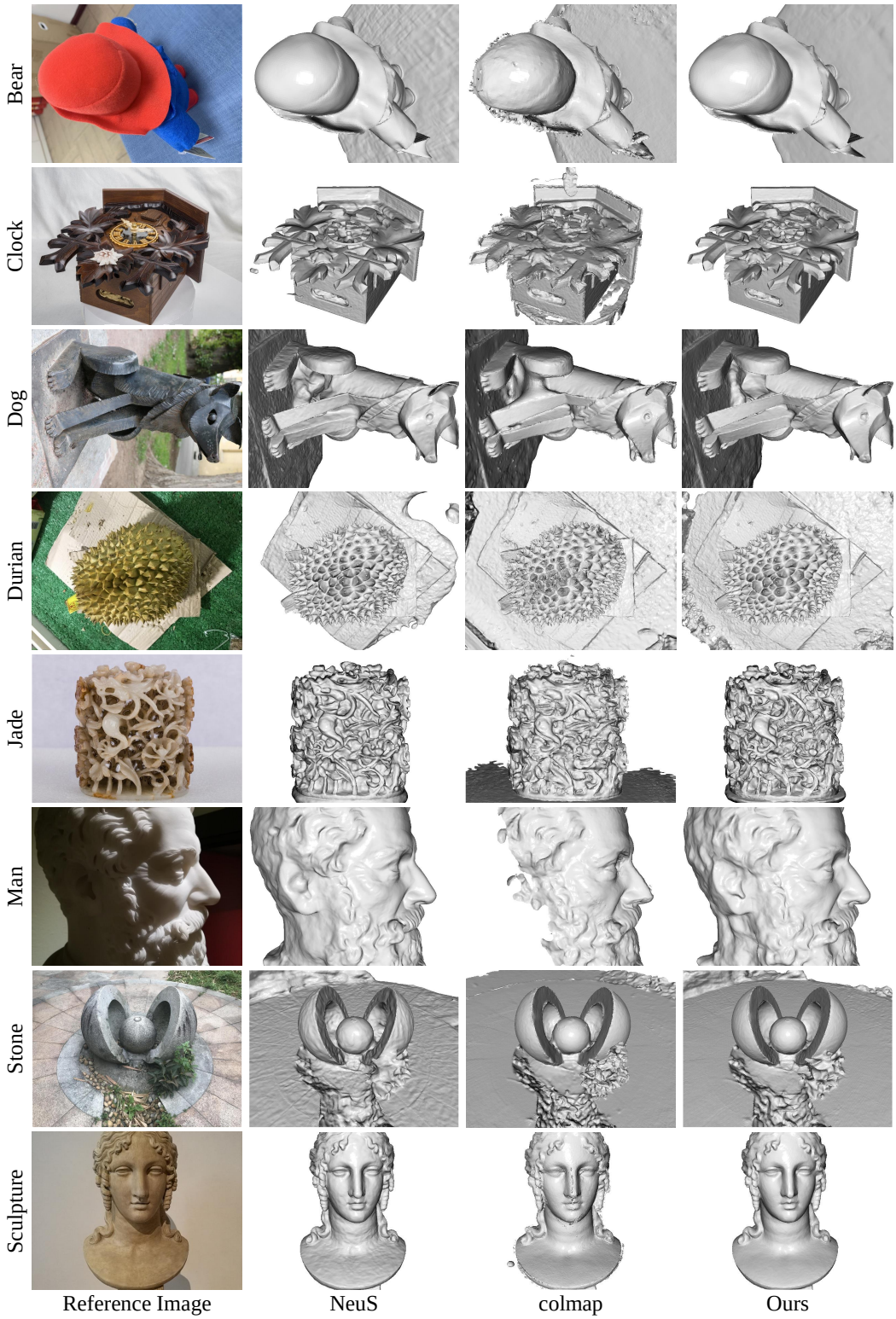

Figure 6: Surfaces reconstructed on BlendedMVS.

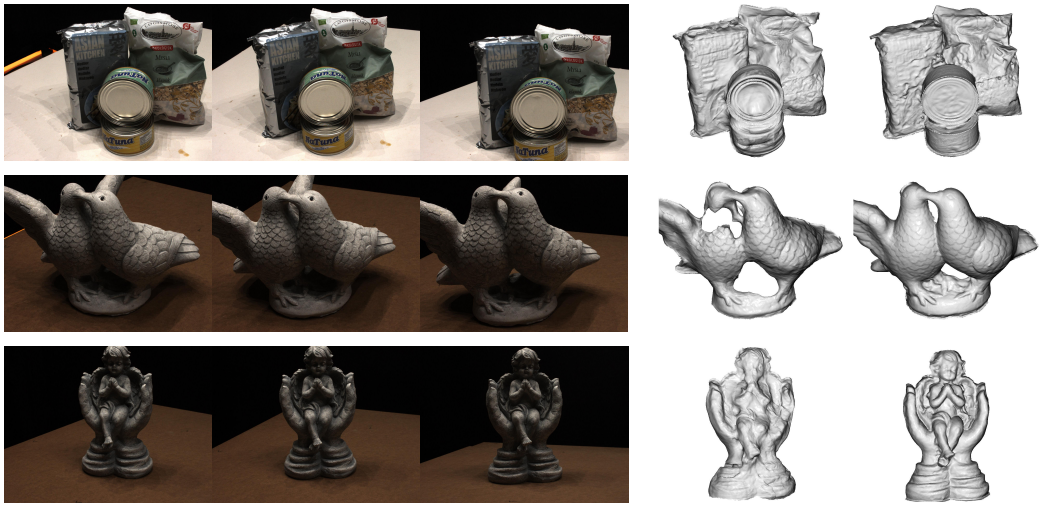

Figure 7: Surface reconstructed with 3 views.