Supplementary Material:Neural Reconstruction of Transparent Objects with Self-Occlusion Aware Refraction-Tracing

1. More results

1.1. Reconstruction using surface rendering or volume rendering.

Qualitative and quantitative results using full views with different rendering techniques are presented in Figure 1 and Table 1. As you can see, volume rendering yields better results with rich details.

(a) Surface rendering (b) Volume rendering

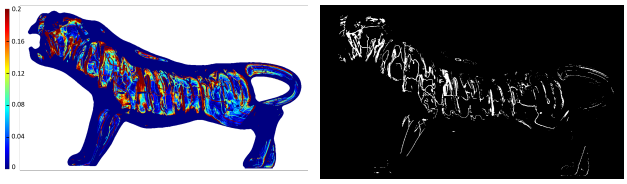
Figure 1: Reconstruction with surface rendering or volume rendering.

1.2. DRT dataset.

Reconstruction with full views(72 views). The reconstructions are presented in Figure 5, and the quantitative comparisons are shown in Table 2.

	Dog		Rabbit		Tiger	
	Acc ↓	Comp ↓	Acc ↓	Comp ↓	Acc ↓	Comp ↓
Surface	10.8523	1.1775	1.2415	1.0435	15.162	1.0975
Volume	**0.7601**	**0.6274**	**0.5839**	**0.4941**	**0.7099**	**0.5705**

Table 1: Reconstruction with volume rendering or surface rendering.



(a) Error map (b) Important pixels

Figure 2: Refraction errors and important pixels (the pixels with top 20% large errors in an image). To speed up the network optimization, we perform refraction tracing on the explicit mesh to obtain the errors of all pixels for each view and sample more camera rays from the pixels with top 20% large errors in an image.

1.3. Self-collected objects.

Besides the Bull, we collected two more real transparent objects, Mouse and Tiger. The qualitative comparisons are provided in Figure 3.

2. Data Acquisition of self-collected objects

Similar to the previous work [5, 3], a monitor is used as background, which displays gray-coded patterns. The transparent object is rotated using a turntable to capture the object from a set of (8 by default) viewpoints which are uniformly sampled from 360° space. For each viewpoint, a series of horizontal and vertical stripe patterns are displayed for silhouette extraction and environment matte [6]. The environment matte allows our method to calculate the location where a refracted camera ray hits on the monitor background. The obtained ray-location correspondences are used for geometry optimization.

	DRT[3]					**Our**				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
Pig	0.6566	0.6863	0.7318	0.6974	0.7142	**0.5669**	**0.4689**	**0.8286**	**0.8671**	**0.8474**
Dog	0.9072	0.8704	0.5588	0.5466	0.5526	**0.7601**	**0.6274**	**0.6712**	**0.7376**	**0.7029**
Mouse	0.8018	0.839	0.5535	0.4951	0.5226	**0.7788**	**0.6811**	**0.5792**	**0.6219**	**0.5998**
Monkey	0.945	0.8923	0.4354	0.4491	0.4422	**0.8415**	**0.7467**	**0.4273**	**0.5547**	**0.4827**
Horse	0.6636	0.6095	0.79084	0.9007	0.8422	**0.6193**	**0.4099**	**0.8262**	**0.9504**	**0.884**
Tiger	0.8191	0.723	0.7435	0.791	0.7665	**0.7099**	**0.5705**	**0.756**	**0.8447**	**0.7979**
Rabbit	0.5971	0.6202	0.7865	0.7516	0.7686	**0.5839**	**0.4941**	**0.8141**	**0.8516**	**0.8324**
Hand	0.4789	0.5855	0.6969	0.496	0.5796	**0.3947**	**0.314**	**0.7481**	**0.8017**	**0.7740**

Table 2: The evaluation of reconstructions with full views using full metrics. Compared with DRT [3] and Li *et al*. [1], our method achieves the best performance in all cases

Method	Definition
Accuracy	$e_{\mathbf{r}\to\mathcal{G}} = \min_{\mathbf{g}\in\mathcal{G}} \lvert\mathbf{r}-\mathbf{g}\rvert$
Completeness	$e_{\mathbf{g}\to\mathcal{R}} = \min_{\mathbf{r}\in\mathcal{R}} \lvert\mathbf{g}-\mathbf{r}\rvert$
Precision	$P(\xi) = \frac{100}{\lvert\mathcal{R}\rvert} \sum_{\mathbf{r}\in\mathcal{R}} [e_{\mathbf{r}\to\mathcal{G}} < \xi]$
Recall	$R(\xi) = \frac{100}{\lvert\mathcal{G}\rvert} \sum_{\mathbf{g}\in\mathcal{G}} [e_{\mathbf{g}\to\mathcal{R}} < \xi]$
F-score	$F(\xi) = \frac{2P(\xi)R(\xi)}{P(\xi)+R(\xi)}$

Table 3: **Metric definitions**. Let $\mathcal{G}$ be the ground truth and $\mathcal{R}$ a reconstructed point set being evaluated.

3. Adaptive sample strategy

We leverage an adaptive ray sampling strategy to speed up the optimization. Specifically, in the optimization, we sample more rays in the regions of an object with large refraction errors, typically the regions with rich details. However, this strategy requires calculating the refraction errors of all the pixels for each image, which will be significantly time-consuming if we adopt volume rendering [4] to calculate the errors. Fortunately, the errors are only used to guide the ray sampling instead of optimizing the geometry, so the errors are not required to be differentiable.

As a result, we utilize a more efficient way to calculate the errors. We use the marching cube technique [2] to obtain the explicit mesh of the optimized geometry and conduct refraction-tracing on the mesh to obtain the errors at a fast speed. An error map of a view is shown in Figure 2. We also visualize the pixels with top 20% large errors, we will sample more points from the important pixels. It takes about 0.27 seconds to get an error map per view.

4. Evaluation metrics

The definitions of 3D reconstruction metrics are shown in Table 3, and the ξ is a pre-defined threshold, $\xi = l/256$, where l is the length of the longest side of the object bounding box.

References

[1] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. Through the looking glass: Neural 3d reconstruction of transparent shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1262–1271, 2020. 2

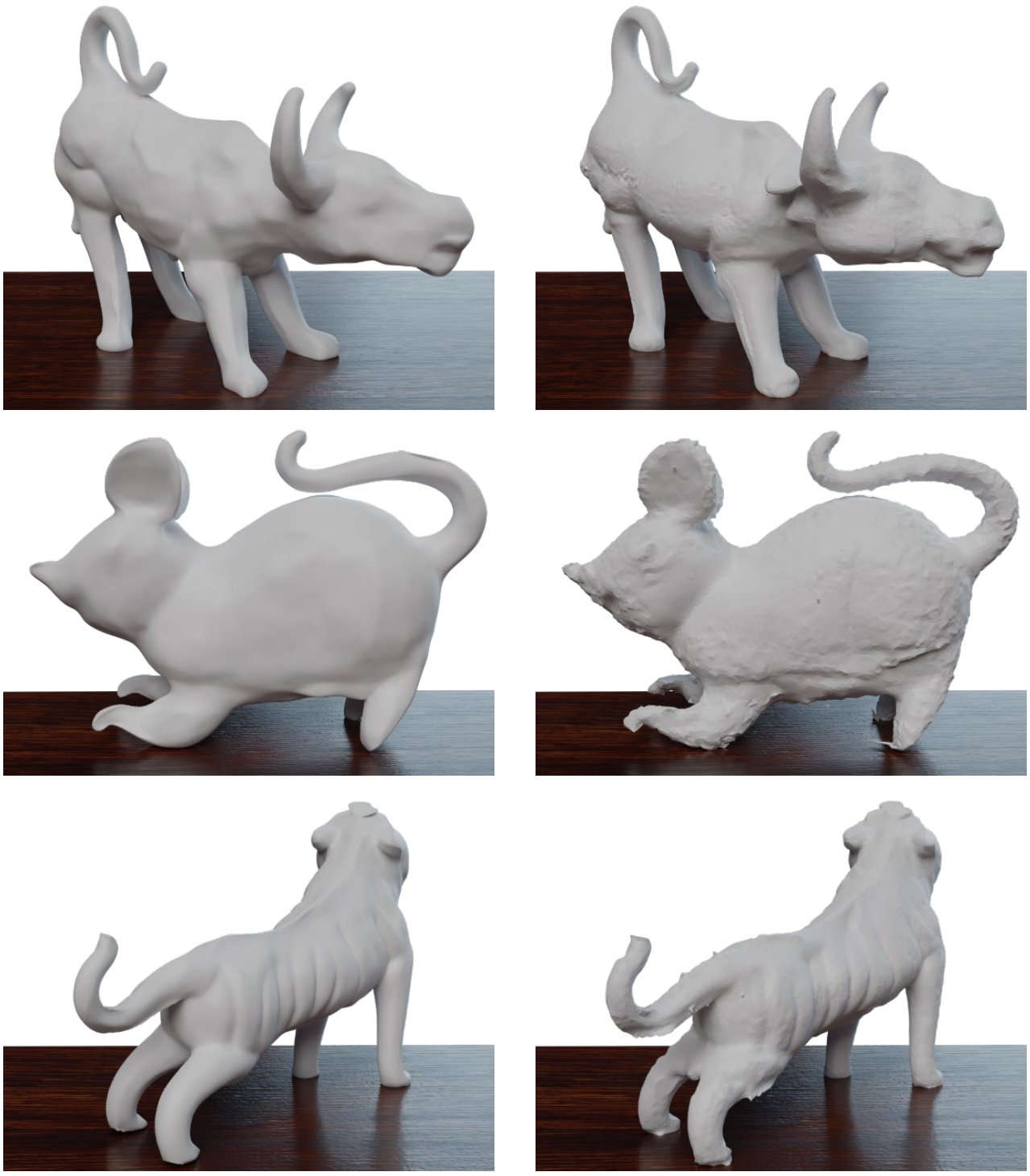
[2] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2

[3] Jiahui Lyu, Bojian Wu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Differentiable refraction-tracing for mesh reconstruction of transparent objects. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020. 1, 2, 3, 4, 5

[4] 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. 2

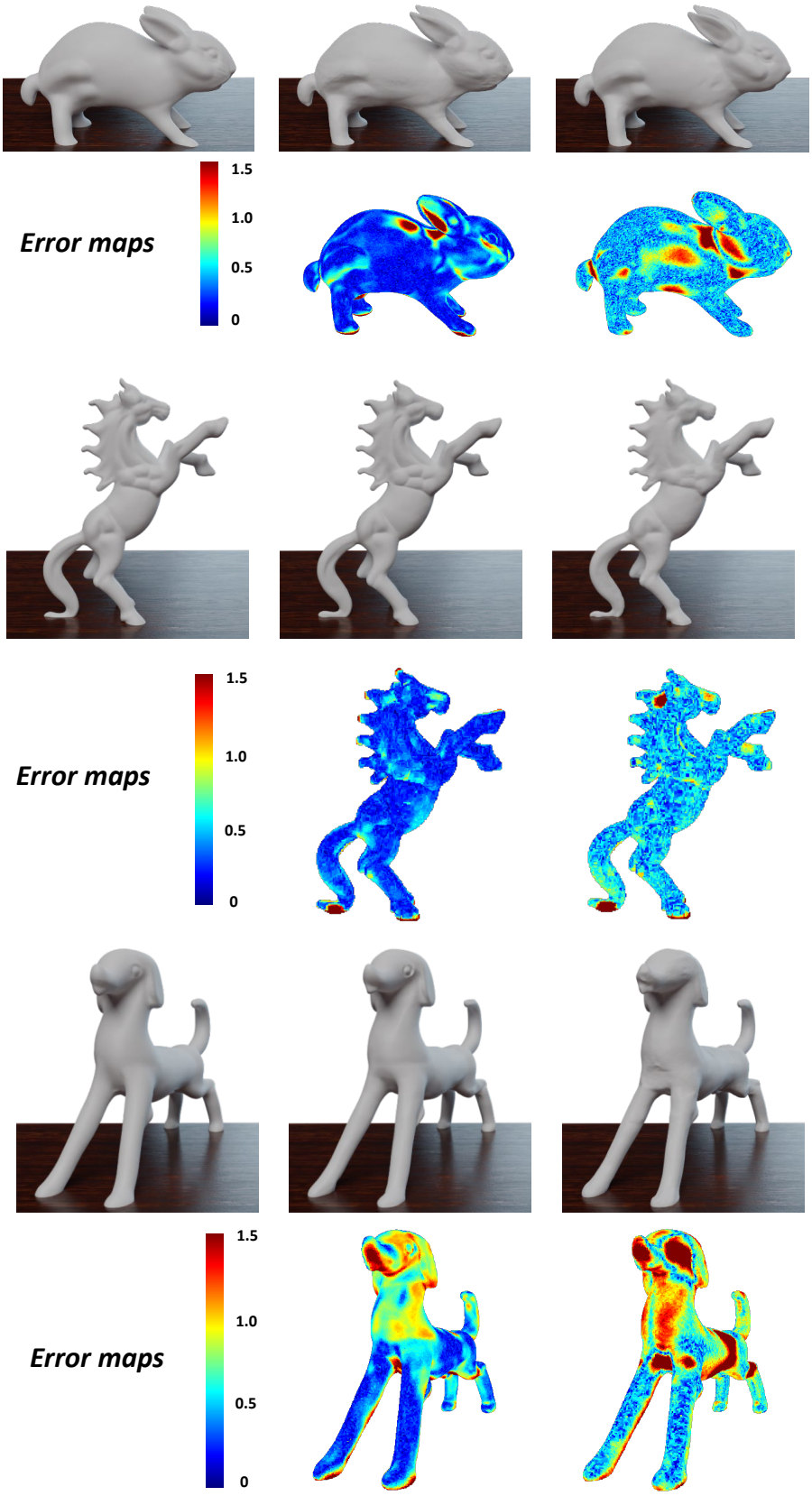
[5] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. *arXiv preprint arXiv:1805.03482*, 2018. 1

[6] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 205–214, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 1



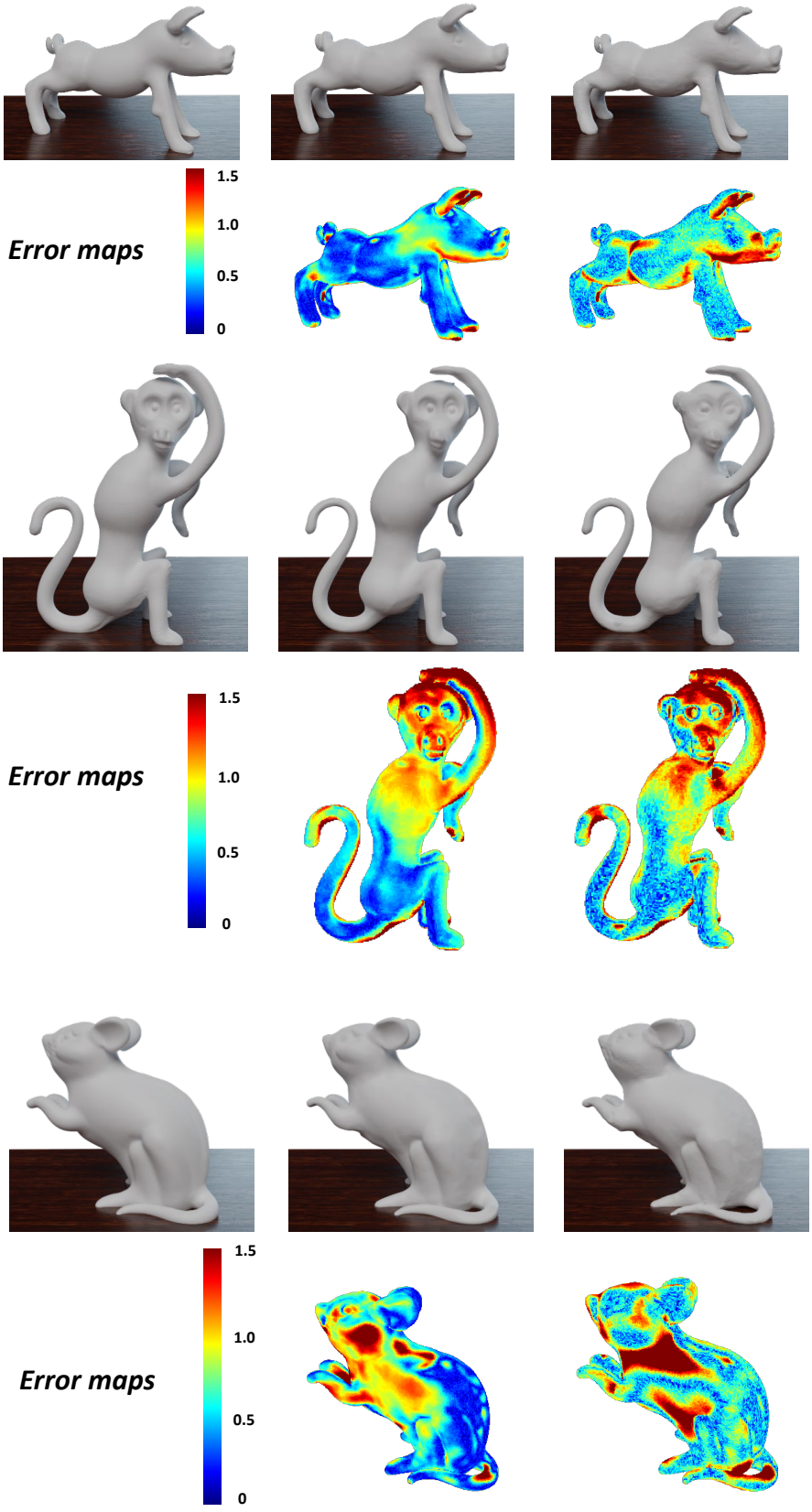
Ours DRT [3]

Figure 3: The reconstructions of real data.



Ground Truth Ours DRT [3]

Figure 4: The reconstructions of dense views(72 views) on DRT's dataset.



Ground Truth Ours DRT [3]

Figure 5: The reconstructions of dense views(72 views) on DRT's dataset.