Looking Through the Glass: Neural Surface Reconstruction Against High Specular Reflections (Supplementary Materials)

Jiaxiong Qiu[1] Peng-Tao Jiang[2] Yifan Zhu[1] Ze-Xin Yin[1] Ming-Ming Cheng[1] Bo Ren[1*]
[1]VCIP, CS, Nankai University [2]Zhejiang University

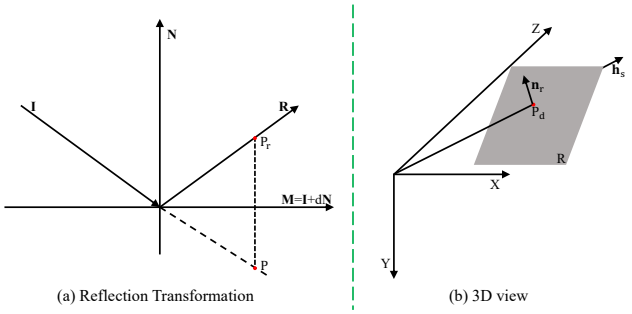


Figure 1. (a): Reflection transformation [2]. (b) 3D view of an auxiliary plane in the camera coordinate system. R is an auxiliary plane of a camera ray $\mathbf{h}_s$. $\mathbf{n}_r$ is the unit normal vector of R.

Unlike recent methods of transparent object reconstruction [3, 7] which aim to reconstruct 3D transparent objects, our goal is to recover the object surface behind transparent objects (*i.e.*, glasses). Here, we provide more details of our method and experiments. Specifically, we provide the details of the reflection transformation [2] and projection algorithm we adopt in our manuscript (Sec. A), the choice of the ratio in the linear summation (Sec. B), additional details of experiments (Sec. C), additional results (Sec. D), additional analysis (Sec. E) and the future work (Sec. F).

A. Projection Algorithm

Fig. 1 (a) shows an illustration of the reflection transformation [2]. Suppose a ray is incident on a glass $\mathbf{M}$, the incident direction is $\mathbf{I}$, the reflected direction is $\mathbf{R}$ and the plane equation of $\mathbf{M}$ is defined as:

$$\mathrm{L} \cdot \mathrm{P} = \mathrm{Ax} + \mathrm{By} + \mathrm{Cz} + \mathrm{D} = 0 \tag{1}$$

where $\mathrm{L} = (\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D})$ and $\mathrm{P} = (\mathrm{x}, \mathrm{y}, \mathrm{z}, 1)$. The unit normal vector $\mathbf{N}$ of $\mathbf{M}$ is $(\mathrm{A}, \mathrm{B}, \mathrm{C}, 0)$.

Given $\mathrm{P_r}$ is a point on the reflected light, and P is its virtual image, then $\mathrm{L} \cdot \mathrm{P_r}$ is the vertical distance of $\mathrm{P_r}$ from $\mathbf{M}$, then we have:

$$\mathrm{P} = \mathrm{P_r} - 2(\mathrm{L} \cdot \mathrm{P_r})\mathbf{N} = \mathrm{M_r P_r} \tag{2}$$

where $\mathrm{M_r}$ is denoted by:

$$\mathrm{M_r} = \begin{bmatrix} 1-2\mathrm{A}^2 & -2\mathrm{AB} & -2\mathrm{AC} & -2\mathrm{AD} \\ -2\mathrm{AB} & 1-2\mathrm{B}^2 & -2\mathrm{BC} & -2\mathrm{BD} \\ -2\mathrm{AC} & -2\mathrm{BC} & 1-2\mathrm{C}^2 & -2\mathrm{CD} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Obviously, Eqn. (2) is a differentiable function. In our work, the camera ray is along the negative $\mathbf{R}$ and we use the reflection transformation to trace the incident ray. Fig. 1 (b) shows the 3D view of a auxiliary plane. The auxiliary plane is built in the camera coordinate system. Given the Cartesian viewing direction unit vector $\mathbf{v} = (\mathrm{x_v}, \mathrm{y_v}, \mathrm{z_v})$ and plane position $\mathrm{d_r}$, then we have:

$$\mathrm{P_d} = \mathrm{d_r}\mathbf{v} = (\mathrm{d_r x_v}, \mathrm{d_r y_v}, \mathrm{d_r z_v}) \tag{4}$$

$\mathrm{P_d}$ is on the auxiliary plane (*i.e.*, Eqn. (1)). Given $\mathbf{n}_r = (\mathrm{A}, \mathrm{B}, \mathrm{C})$, then we have:

$$\begin{aligned} \mathrm{D} &= -(\mathrm{Ad_r x_v} + \mathrm{Bd_r y_v} + \mathrm{Cd_r z_v}) \\ &= -\mathrm{d_r}(\mathrm{Ax_v} + \mathrm{By_v} + \mathrm{Cz_v}) \\ &= -\mathrm{d_r}\mathbf{n}_r \cdot \mathbf{v} \end{aligned} \tag{5}$$

Based on the above description, we present our strategy of acquiring the input points $\mathbf{p}_r$ of the plane path in Algorithm 1.

B. Linear Summation

We fuse the appearances of two paths by a linear summation to be the rendered image, which can be supervised by the captured RGB image. As Tab. 1 shows, different ratios of the target object appearance have different effects on the reconstruction quality of the target object. We select 0.3 as the default ratio in our model according to these results.

*Bo Ren is the corresponding author.

Algorithm 1: Transforming sampled points along a camera ray $\mathbf{h}_s$.

Input: The plane normal $\mathbf{n}_r = (\mathrm{A}, \mathrm{B}, \mathrm{C})$, the plane position $\mathrm{d_r}$, the camera center $\mathbf{o}$, the depth t, the view direction $\mathbf{v}$ and the sampled points $\mathbf{p}$ along $\mathbf{h}_s$.

Output: Spatial points $\mathbf{p}_r$ of the plane path in the camera coordinate system.

$\mathbf{p}' = \mathbf{p}_t \cup \mathbf{p}'_t = \mathbf{p} - \mathbf{o}$;
$\mathrm{P_d} = \mathrm{d_r}\mathbf{v}$;
$\mathrm{D} = -\mathrm{d_r}\mathbf{n}_r \cdot \mathbf{v}$;
$\mathbf{p}_t = \{\mathrm{t}\mathbf{v} | \mathrm{t} \in [0, \mathrm{d_r}]\}$;
$\mathbf{p}'_t = \{\mathrm{t}\mathbf{v} | \mathrm{t} \in [\mathrm{d_r}, 1]\}$;
$\{\mathrm{D}, \mathbf{n}_r\} \rightarrow \mathrm{M_r}$;
$\mathbf{p}_a = \mathrm{M_r^{-1}}\mathbf{p}'_t$;
$\mathbf{p}_r = \mathbf{p}_t \cup \mathbf{p}_a$.

Table 1. Effects of different ratios of the target object appearance on 'scan24'. The standard deviation is 0.68.

Ratio	0.1	0.3	0.5	0.7	0.9
Chamfer distance ↓	3.29	2.07	2.43	3.99	3.17

C. Additional Experimental Details

C.1. Hierarchical Sampling

We follow the hierarchical sampling of NeuS [6] to generate the input spatial points. Specifically, we sample 64 points along a ray uniformly at first, then we perform the importance sampling for 4 times. The total number of sampled points is 128. We sample extra 32 points outside the sphere according to NeRF++ [9].

C.2. Neural Network Architecture

The whole network architecture consists of three parts: SDF predicting, auxiliary plane predicting, and color predicting. For SDF predicting, we follow the neural network architecture of NeuS, which is activated by Softplus where $\beta = 100$. Weight normalization is adopted for stable training. The input is concatenated with the features from the fourth layer by a skip connection. For auxiliary plane predicting, the volume density part consists of three linear layers with ReLU, the position and normal branches both consist of two linear layers. The hidden layers of two branches are activated by ReLU, while the last layers of two branches are activated by Sigmoid and Tanh respectively. For color predicting, the hidden layers are activated by ReLU, and the last layer is activated by Sigmoid.

C.3. Datasets

Tab. 2 reports the metrics of our synthetic and real-world datasets. For the synthetic dataset, we set the kernel size of the Gaussian filter to 11 for generating the reflection effect. We randomly pick a scene ('Scan114') of the DTU dataset [1] as the source of high specular reflections. We select 10 scenes from a total of 15 scenes on the DTU dataset [1] based on visual reality. They are: 'Scan24', 'Scan37', 'Scan40', 'Scan55', 'Scan63', 'Scan65', 'Scan69', 'Scan83', 'Scan97', and 'Scan105'. For the real-world dataset, we capture one scene ('Toys') and collect 5 scenes from the Internet: 'Buddha' [1], 'Figure' [2], 'Plate' [3], 'Porcelain' [4] and 'Bronze' [5]. Examples of real-world scenes in our experiments are shown in Fig. 6.

C.4. Inference Time

For object surface reconstruction, the inference time of NeuS-HSR is 36 seconds under resolution = 64 and threshold = 0.0. For rendering a novel view at the resolution of 800×600, NeuS-HSR takes around 96 seconds without the ground-truth mask on a single NVIDIA Tesla V100 GPU.

C.5. Baselines

Because original neural implicit baselines are trained and tested on the datasets without HSR, we retrain all these models on each scene of our synthetic and real-world datasets.

NeuS [6]. To obtain the results of NeuS, we use their released official codes [6] with the default setting in all scenes.

UNISURF [4]. To compare with UNISURF, we adopt their officially released codes [7] with the default setting in the synthetic scenes.

VolSDF [8]. To compare with VolSDF, we use their officially released codes [8] with the default setting in all scenes.

COLMAP [5]. To obtain the results of COLMAP, we use the official command version of COLMAP [9] and run sequential commands provided in their documents [10] in all scenes.

C.6. Q&A

Q1. *How about the quality of rendered images in novel views?*

A1. The goal of our work is to reconstruct the target object against HSR accurately with multi-view images as supervision. We conduct an evaluation of novel view

[1] https://www.bilibili.com/video/BV1M44y1z7XX
[2] https://www.bilibili.com/video/BV1BP4y1Y7bV
[3] https://www.bilibili.com/video/BV1BP4y1Y7bV
[4] https://www.bilibili.com/video/BV1UP4y1h7tW
[5] https://www.bilibili.com/video/BV1SU4y1E7QR
[6] https://github.com/Totoro97/NeuS
[7] https://github.com/autonomousvision/unisurf
[8] https://github.com/lioryariv/volsdf
[9] https://github.com/colmap/colmap
[10] https://colmap.github.io/

Table 2. Metrics of datasets used in our experiments.

Scene	Synthetic	Buddha	Toys	Figure	Plate	Porcelain	Bronze
Views	49/64	56	23	60	56	60	43
Resolution	1600 × 1200	1920 × 1080	1372 × 1029	1920 × 1080	1920 × 1080	1080 × 1920	1080 × 1920

Table 3. Model parameters of NeuS-HSR and baselines.

Method	UNISURF [4]	VolSDF [8]	NeuS [6]	NeuS-HSR
#Params	0.8M	1.4M	1.4M	1.5M

Table 4. Comparison of novel view synthesis on 'Bronze'.

Method	NeRF++ [9]	NeuS [6]	NeuS-HSR
PSNR↑	15.92	15.51	15.93
SSIM↑	0.480	0.489	0.502

synthesis on 'Bronze'. We select the first 3 images and the last 7 images of the sequential images for testing. The average scores of the test sets on PSNR and SSIM are present in Tab. 4.

Q2. *Why use the same appearance function F_c in two paths?*

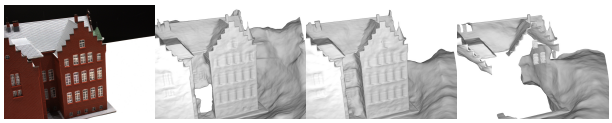
A2. Firstly, we use the same F_c to save model parameters. Secondly, because the two paths of our framework are trained in one stage and the supervision is only the captured image, we adopt the same F_c to separate the two appearances in the same domain. Lastly, we consider F_c as an implicit function that maps 3D locations, normals, view directions, and neural features to color values. We use different 3D locations and normals from two paths as the input to acquire different color values by F_c.

Q3. *Why is the auxiliary plane built in the camera coordinate system?*

A3. Our model is trained by a view at each iteration, we build an auxiliary plane of each view in the camera coordinate system for simplification as Fig. 1 shows. We transform the 3D locations to the camera coordinate system first, then we can apply the reflection transformation by the auxiliary plane directly.

Q4. *How about the performance of NeuS-HSR in non-HSR scenes?*

A4. NeuS-HSR is built on two physical assumptions in HSR scenes. In non-HSR scenes, we can set the ratio of the linear summation to 1.0, then NeuS-HSR degrades to NeuS and can achieve the same performance as NeuS. The qualitative results are shown in Fig. 2



(a) Supervision (b) NeuS (1.0) (c) Ours (0.7) (d) Ours (0.3)

Figure 2. Performance in a non-HSR scene.

D. Additional results on Synthetic Dataset

D.1. Signed Distance Fields

We visualize the signed distance fields in Fig. 3. Our model extracts a more accurate SDF of the scene than NeuS according to the distribution of signed distance fields. Specifically, our signed distance fields present the geometric characteristics of Bunny's tangent plane.

D.2. Components

Fig. 7 shows the components of NeuS-HSR on the synthetic dataset. Our method faithfully enhances the target object appearance and preserves HSR in the auxiliary plane appearance without any priors. Besides, the plane normal and position of each auxiliary plane on a camera ray in a view, indicate that the auxiliary planes tend to be a planar reflector. Hence, Fig. 7 illustrates that our model achieves the physical decomposition of HSR scenes.

D.3. Comparisons

More qualitative comparisons between NeuS-HSR and other state-of-the-art methods on the synthetic dataset are shown in Fig. 5. All neural implicit approaches are trained without ground-truth masks. COLMAP [5] generates too much noise around the target object surface to calculate the metric (*i.e.*, Chamfer distance) of its results in our manuscript.

E. Trainable Standard Deviation on the Real-World Dataset

In NeuS [6], the optimization process reduces the standard deviation automatically then the surface becomes sharper. We conduct a comparison between NeuS-HSR (blue curve) and NeuS (orange curve) on the trainable standard deviation. The standard deviation of our method converges to a smaller value compared to the standard deviation of NeuS, and our method achieves clearer and sharper results on the real-world dataset than NeuS as our manuscript shows.

F. Future Work

In the future, we plan to extend our approach to handle glasses with different thicknesses. In our daily life, the thicker the glass, the more obvious the specular reflections. One possible scheme may be adding thickness to our auxiliary plane module. Besides, our method can also evolve to tackle highly reflective object surfaces (*e.g.*, cars).

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. 2

[2] Ronald Goldman. *Matrices and transformations*. Graphics Gems, 1990. 1

[3] 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. 1

[4] 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. 2, 3

[5] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[6] 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*, 34:27171–27183, 2021. 2, 3

[7] Jiamin Xu, Zihan Zhu, Hujun Bao, and Wewei Xu. A hybrid mesh-neural representation for 3d transparent object reconstruction. *arXiv preprint arXiv:2203.12613*, 2022. 1

[8] 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. 2, 3

[9] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2, 3

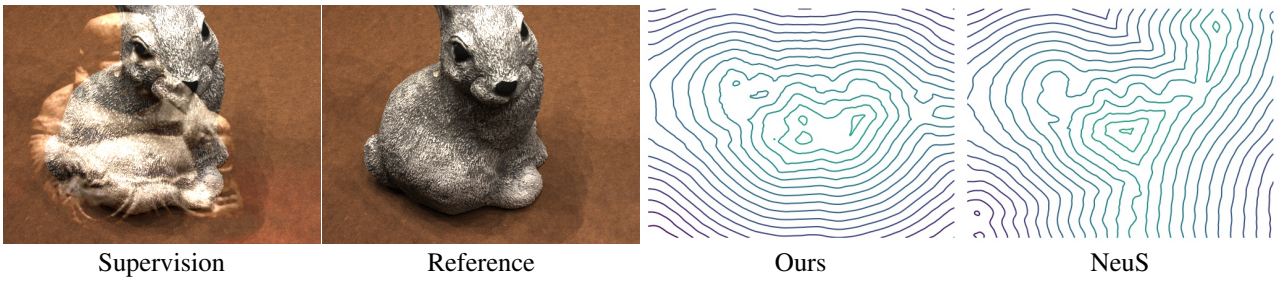


Figure 3. Visualization of signed distance fields.

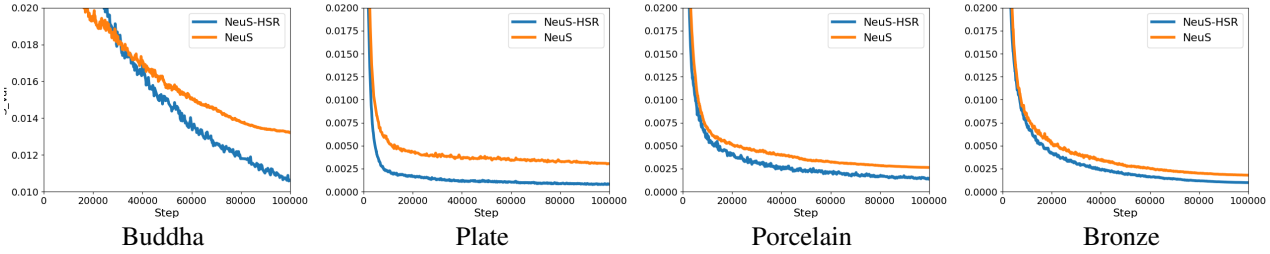


Figure 4. Comparison of trainable standard deviation.

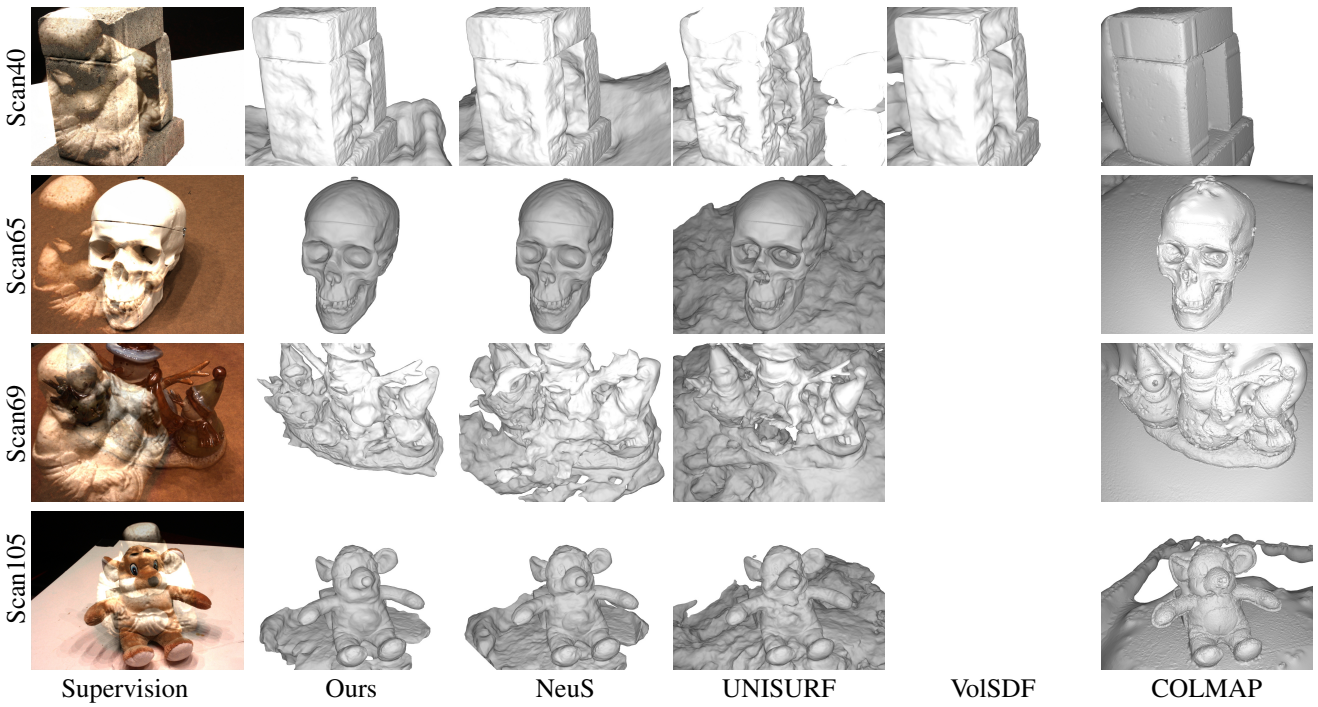


Figure 5. Qualitative comparisons between NeuS-HSR and baselines on the synthetic dataset.

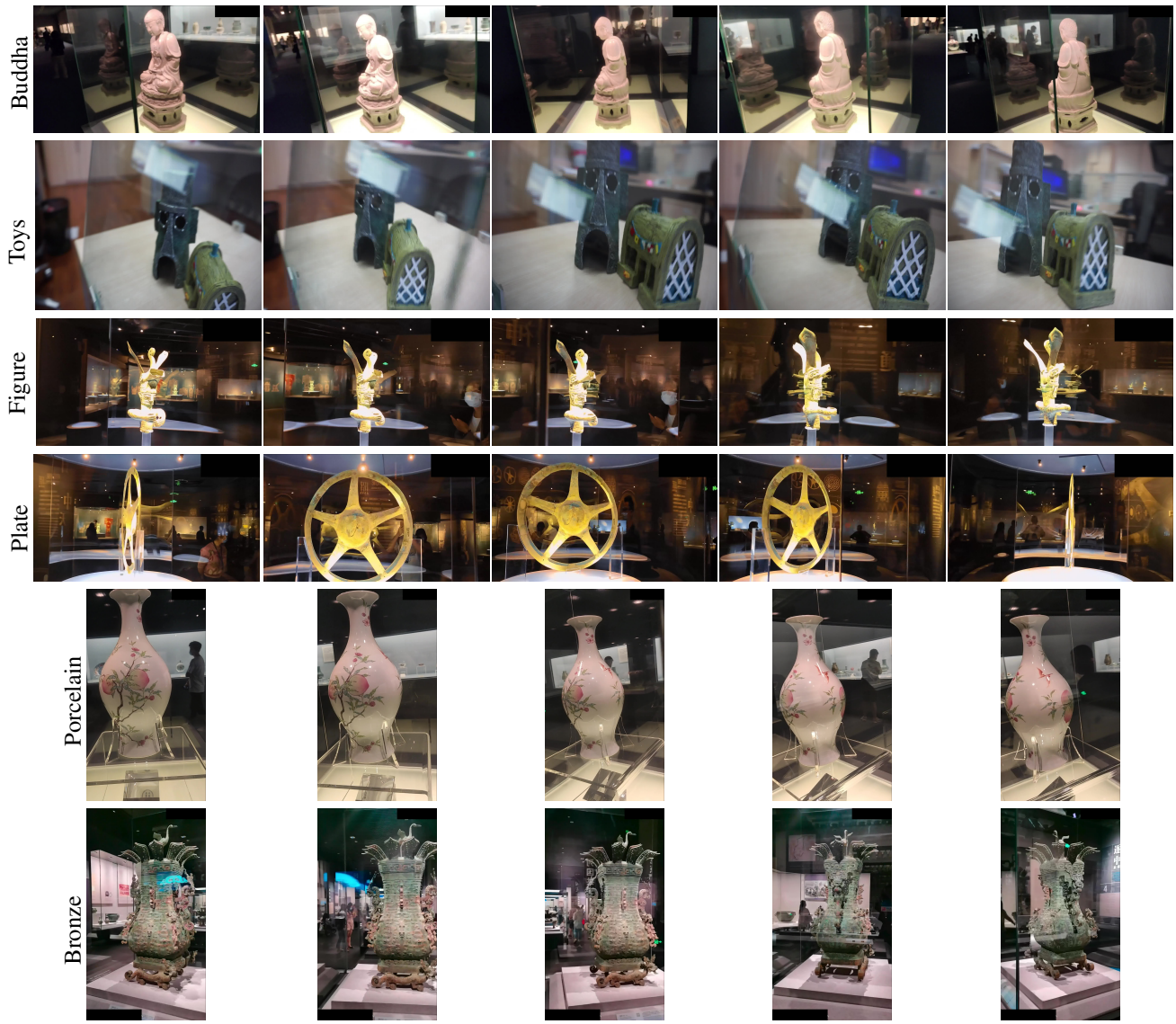


Figure 6. Examples of the real-world dataset.

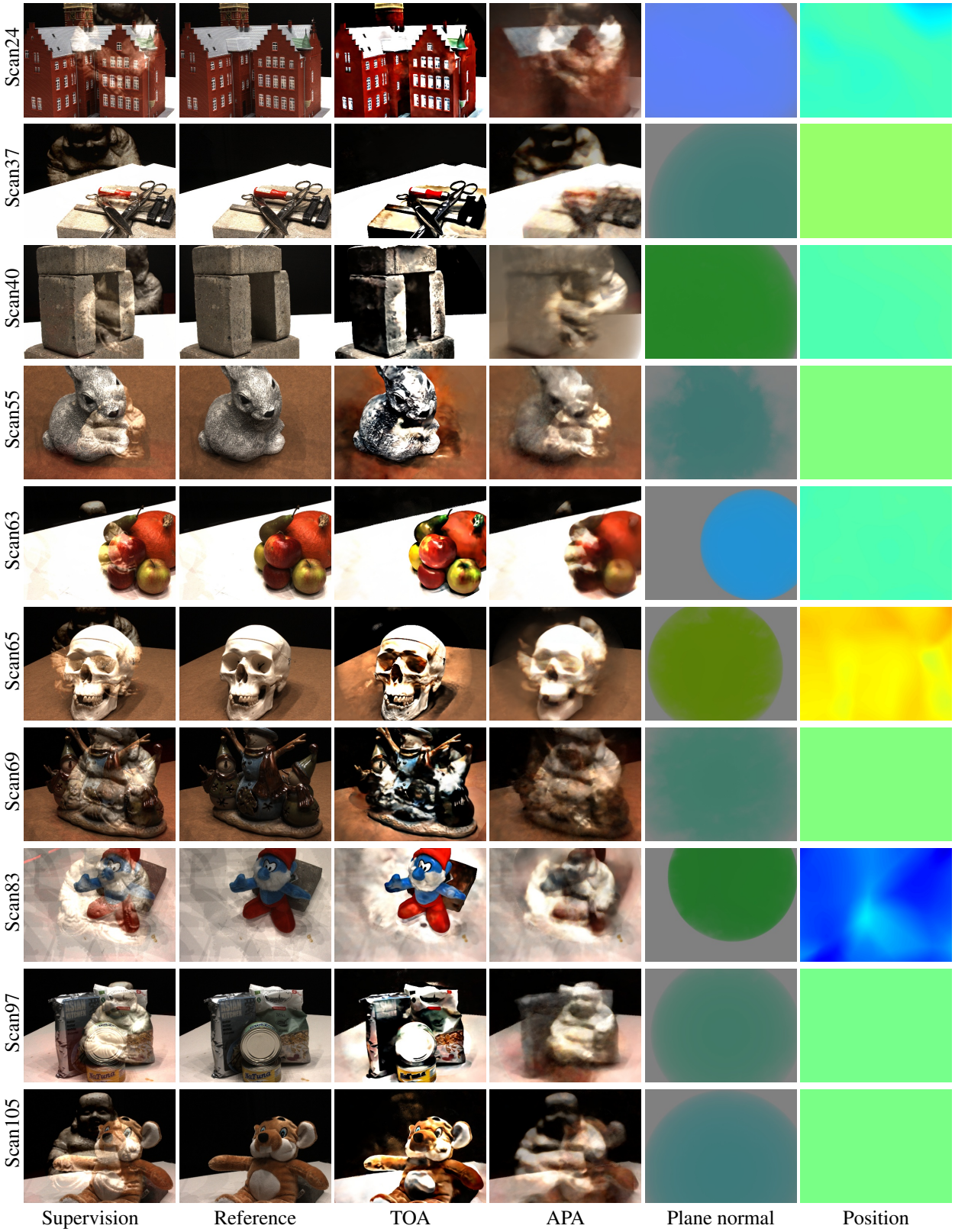


Figure 7. Components of NeuS-HSR on the synthetic dataset. 'TOA': Target object appearance. 'APA': Auxiliary plane appearance.