

# NLOS-NeuS: Non-line-of-sight Neural Implicit Surface

## — Supplementary Material —

Yuki Fujimura<sup>1</sup>

Takahiro Kushida<sup>2</sup>

Takuya Funatomi<sup>1</sup>

Yasuhiro Mukaigawa<sup>1</sup>

<sup>1</sup>Nara Institute of Science and Technology

<sup>2</sup>Ritsumeikan University

{fujimura.yuki, funatomi, mukaigawa}@is.naist.jp, tkushida@fc.ritsumei.ac.jp

### Abstract

*This supplementary material describes the proof of Proposition 1. (Sec. A), the details of our space carving algorithm (Sec. B), training details (Sec. C), examples of background rendering (Sec. D), and examples of directional albedos from different views (Sec. E)*

### A. Proof of Proposition 1

This section provides the proof of the Proposition 1. We rewrite Eq. (4) in the main paper as

$$f(x) = \frac{1}{\alpha} \frac{1}{1 + e^{\frac{x}{\alpha}}}. \quad (1)$$

*Proof.* To analyze the behavior of the mask loss (Eq. (9) in the main paper), we consider a sampling ray near the object boundary. In the empty space, all densities at sampled points on the ray should be zero. Now let  $\epsilon > 0$  be the minimum signed distance value on this ray, which also leads to the highest density on the ray and should be close to zero in the empty space, i.e.,  $f(\epsilon) \rightarrow 0$ .

In the empty space, the signed distance at each position is positive. When  $x > 0$ , the first and second derivatives of  $f(x)$  satisfy  $f'(x) < 0$  and  $f''(x) > 0$ . Thus,  $f(\epsilon)$  can be lower bounded by the first-order derivative approximation around  $x = 0$ :

$$f(\epsilon) > f(0) + f'(0)\epsilon = \frac{1}{2\alpha} - \frac{1}{4\alpha^2}\epsilon. \quad (2)$$

Thus,  $f(\epsilon) \rightarrow 0$  leads to  $\alpha < \epsilon/2$ . When minimizing the mask loss,  $\alpha$  can be upper bounded by  $\epsilon/2$ . If points are sampled densely enough, at the object boundary where  $\epsilon \rightarrow 0$ ,  $\alpha$  converges to 0.  $\square$

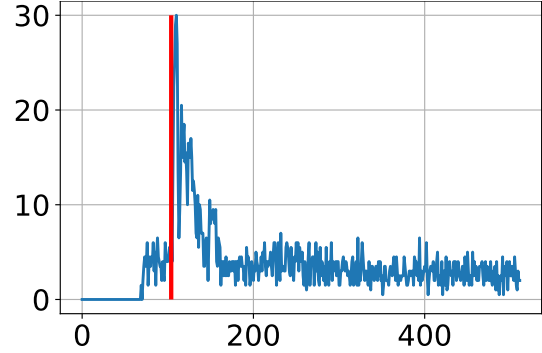


Figure 1. Example of detecting temporal bin corresponding to first-returning photon in real data. Blue plot is measured transient and red line indicates detected temporal bin.

### B. Space carving with the first-returning photons

For the free space loss  $\mathcal{L}_f$ , we apply the space carving algorithm on the basis of the geometry of first-returning photons [3]. In this section, we explain the detection of the temporal bin corresponding to the first-returning photon and robust space carving algorithm.

#### B.1. Detection of the first-returning photons

For the detecting the first-returning photons, we simply compute the difference between the values of nearby temporal bins in a transient. Formally, let  $\tau_m \in \mathbb{R}^B$  be a measured transient. At each temporal bin  $t$ , we compute  $\tau_m(t) - \tau_m(t+1)$ . The bin of the first-returning photon is then computed as  $\arg \min_{t \in S} [\tau_m(t) - \tau_m(t+1)]$ ,  $S = \{t | \tau_m(t) - \tau_m(t+1) > \eta\}$ , where  $\eta$  is a threshold value. However, real data captured using a SPAD contains high-frequency noise and background effects; thus, we apply the Gaussian filter and thresholding to suppress such undesirable signals as preprocessing. Figure 1 shows an example of the detected bin of the first-returning photon in the real data.

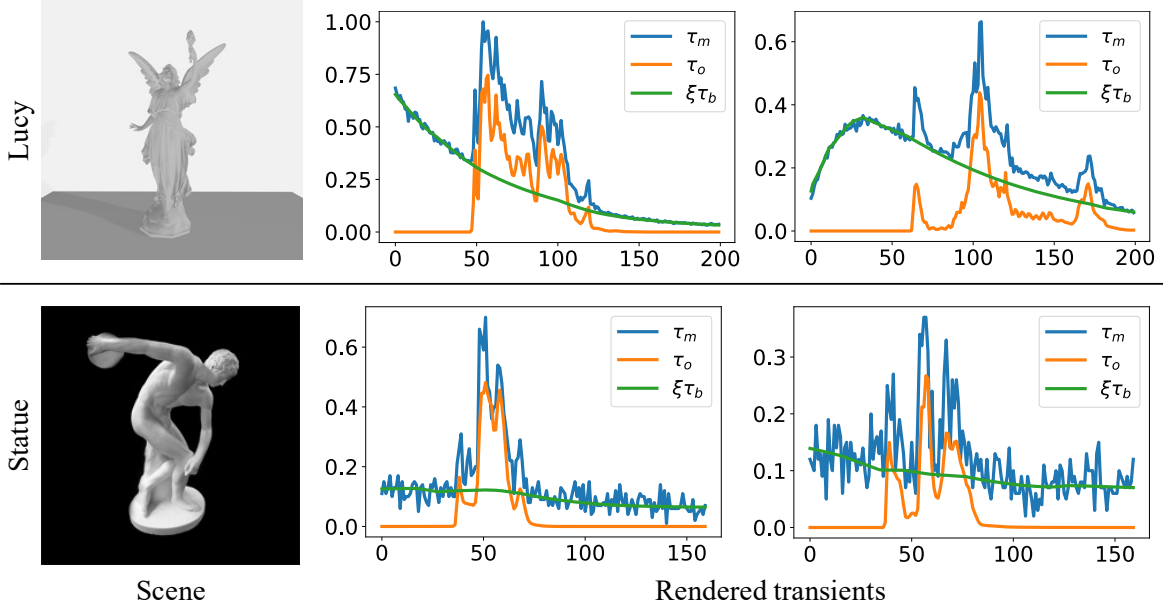


Figure 2. Examples of background rendering. Blue, orange, and green plots indicate measured transients and components from object and background, respectively.

## B.2. Robust space carving

After detecting the temporal bins of the first-returning photons, we can apply the space carving algorithm [3]. Let a detected temporal bin be  $t$  and the corresponding radius be  $r_t$  at a relay wall position  $\mathbf{p}'$ . It is guaranteed that an object does not exist in the sphere centered at  $\mathbf{p}'$  with  $r_t$ . We propose a robust space carving method since the detected temporal bins have some errors due to undesirable signals such as noise. First, the target space is divided into  $128 \times 128 \times 128$  grid voxels  $\mathcal{V}$ . We vote for a voxel  $v \in \mathcal{V}$  if  $v$  is inside each carving sphere. Let  $c(v)$  be the total number of votes of a voxel  $v$  after voting with all spheres. Instead of simply using the carved space, we compute an object space  $\Omega_{obj}$  and free space  $\Omega_{free}$  as  $\Omega_{obj} = \{v | c(v) > 0.99 \times \max_{v' \in \mathcal{V}} c(v')\}$  and  $\Omega_{free} = \mathcal{V} - \Omega_{obj}$  for robust space carving.

## C. Training details

We set the weights of the training loss as  $[\lambda_\tau, \lambda_{ei}, \lambda_z, \lambda_{en}, \lambda_f] = [1., 0.1, 0.01, 0.001, 0.01]$  except for the scene “bike,” where we set  $\lambda_{en} = 0.002$  because we found that the scene consisting of thin structures requires large  $\lambda_{en}$  for reducing  $\alpha$  during training.

When rendering a transient, we sampled 64 angles for both  $\theta$  and  $\phi$  on each scan sphere. For  $\mathcal{L}_{ei}$ , we randomly sampled 4096 points in a target NLOS space at each iteration. For  $\mathcal{L}_f$ , we randomly sampled 4096 voxels in a free space at each iteration.

The optimizer was Adam [1] with the hyperparameters

$$lr = 1.0 \times 10^{-4}, \beta_1 = 0.9, \text{ and } \beta_2 = 0.999.$$

## D. Examples of background rendering

To model background effects in measured transients, a shallow MLP for rendering the background effects is jointly trained with an SDF as explained in Sec. 3.6 in the main paper. Figure 2 shows examples of rendering background effects. In the synthetic Lucy scene, the object is located on a floor, and the measured transient contains reflection from the floor. The background network correctly divides the transient into object and floor components. In the real statue scene, the measured transient has the background effects with high-frequency noise. Although the characteristics of these background effects are different, the background network correctly fit both effects.

## E. Rendering from different views

Our neural implicit representation can render directional albedos from different views with volume rendering after training, similar to novel-view synthesis with the NeRF [2]. Figure 3 shows examples of rendered directional albedos from three different views.

## References

- [1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2
- [2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

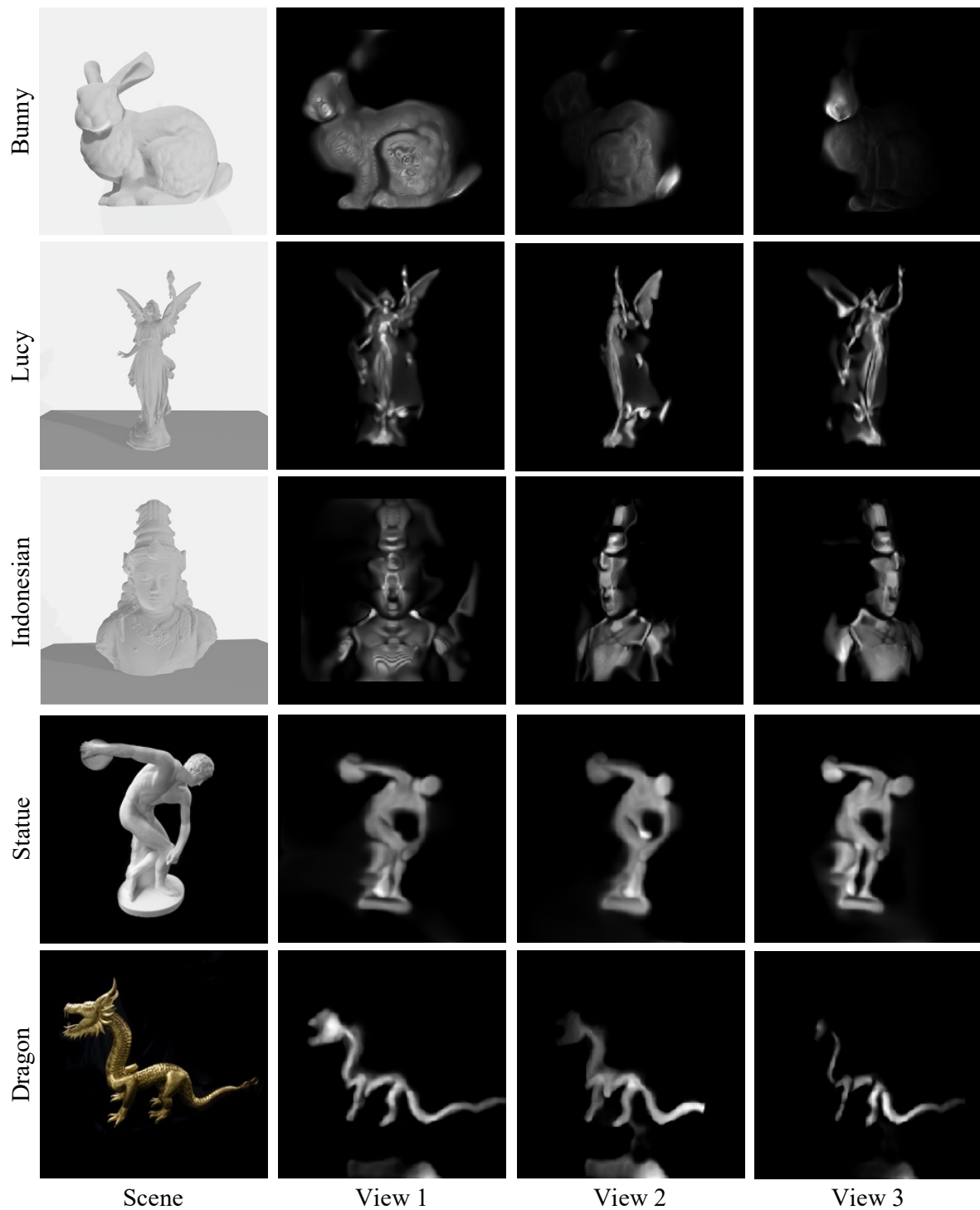


Figure 3. Rendered directional albedos from 3 different views

Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. [2](#)

try of first-returning photons for non-line-of-sight imaging. In *CVPR*, 2017. [1](#), [2](#)

- [3] Chia-Yin Tsai, Kiriakos N. Kutulakos, Srinivasa G. Narasimhan, and Aswin C. Sankaranarayanan. The geome-