ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision Supplementary Material

Jingwang Ling[1] Zhibo Wang[2] Feng Xu[1*]

[1]School of Software and BNRist, Tsinghua University [2]SenseTime Research

A. Relationship between camera and shadow ray supervision

Ray supervision is the core of our method. As the ray supervision is general for arbitrary rays, it leads to a dual relationship between camera ray supervision (*e.g.* NeRF [6]) and our method. We list each method's components in Tab. 1 to better illustrate their correspondences.

B. Additional implementation details

Network architecture. We adopt an 8-layer geometry MLP following [7]. When handling RGB inputs, we model another 4-layer material MLP. We use Softplus for the geometry MLP and ReLU for the material MLP as activation. The hidden layers for both MLPs are 256 dimensional. A 3D position with 6-frequency positional encoding is used as the input for the geometry MLP. The geometry MLP outputs a signed distance and a 256-dimensional feature vector. The feature vector is then concatenated with the 3D position and normal vector as the input for the material MLP. The material MLP outputs a 3-channel diffuse albedo and 27 specular coefficients, with output activation by Softplus ($\beta = 100$). The specular coefficients are used to linearly combine nine spherical Gaussian bases with different shininess to produce a 3-channel specular color. The diffuse and specular colors are represented in the linear color space.

Training. Our networks are trained using Adam [3], with the learning rate first linearly warmed up from 0 to 10^{-3} in the first 5k iterations and then cosine decayed to a minimum learning rate of 5×10^{-5}. The weight of the Eikonal loss is set to 0.01, which we find a lower weight leads to more thin structures reconstructed.

Shadow ray sampling. We place 80 uniform samples along the shadow ray and use the hierarchical sampling strategy in [7] to sample another 64 points near the surface. The far bound is determined by a scene bounding sphere. The near bound is set to 0 so that detailed shadows by sample points near the starting surface can be modeled. We are able to model these near sample points because the SDF-to-density formula (Eq. (3) in the main paper) is dependent on the ray and normal direction. This property is suitable for modeling rays that start at the surface. When the ray goes outward (the dot product between the ray direction and normal direction is greater than 0), we obtain zero densities at near sample points. Thus, the ray will not be incorrectly blocked by its starting surface. When the ray goes inward, it will be appropriately occluded by the starting surface, generating attached shadows.

Camera ray intersection. We use ray marching with 256 steps to locate the intersection between a camera ray and the SDF. We then use a surface walk process in [9] to locate the boundary points. The surface walk process starts at the intersection points with a maximum of 16 steps. In each step, a point moves along the surface with a step size of 2×10^{-3} until it reaches a boundary point whose surface normal direction is perpendicular to the camera ray direction. The boundary point separates a pixel into two regions. We locate the intersection points in the two sub-pixel regions using ray marching and compute the shadow rays started at each region respectively, as shown in Fig. 1. The results of the shadow rays are combined by an area ratio proportional to each region. The area ratio is made differentiable by relating the area to the deformation of the boundary point.

Our setting differs from [9] in that while they use edge sampling to refine an initial geometry, we are optimizing a geometry from scratch. To accelerate convergence, we adopt a coarse-to-fine strategy that optimizes 100×100 low-resolution images in the first 5k iterations and progressively upscales the images to the full 800×800 resolution. This strategy enlarges the pixel footprint, resulting in more boundary points to be considered in the early training iterations.

C. Additional comparison results

C.1. Quantitative comparison on binary shadow inputs

We evaluate two binary shadow datasets: A terrain-like dataset proposed by DeepShadow [2] and a non-terrain

*Corresponding author

	Camera ray supervision (NeRF)	Shadow ray supervision (Ours)
Ray direction	View direction	Light direction
Ray starting point	Camera location	Surface location
Supervision label	Incoming radiance at the camera	Incoming radiance at the surface
Particle-ray interactions	Absorption and emission	Absorption
Capture setup	Multiple views	Multiple lights

Table 1. Corresponding components in camera and shadow ray supervision.

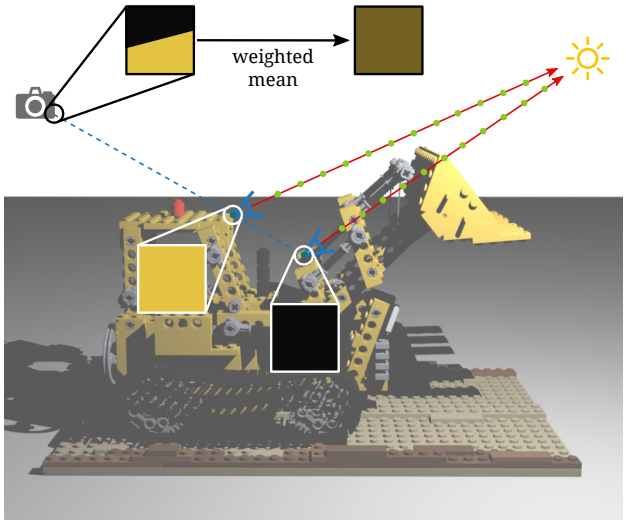


Figure 1. At a boundary pixel, we compute two shadow rays started at different depths and combine their results by weighted mean.

dataset proposed by us. The results on the DeepShadow dataset are shown in Tab. 2, and the results on our dataset are shown in Tab. 3, respectively. Our depth reconstruction outperforms DeepShadow on both terrain-like and non-terrain scenes. Our normal reconstruction is better than DeepShadow on non-terrain scenes and comparable on terrain-like scenes.

The normalized mean depth error (Depth nMZE) used in DeepShadow's paper is only suitable for terrain-like scenes. Therefore, we propose to compute depth error by aligning the depth map to the ground truth using ICP (denoted as Depth L1). For completeness, we also show quantitative results on the DeepShadow dataset using normalized mean depth error in Tab. 4. We report DeepShadow's results from their publicly available code, which are slightly better than their paper results.

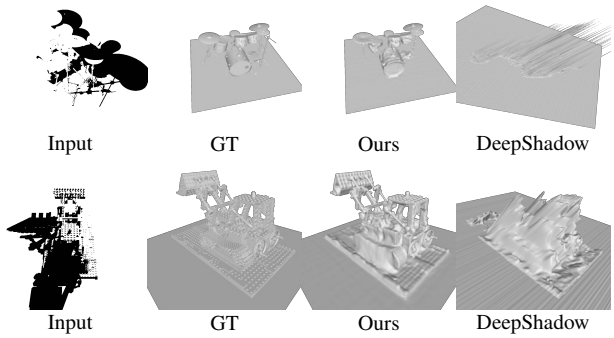


Figure 2. Qualitative comparison on our side-view binary shadow dataset.

C.2. Qualitative comparison on our side-view binary shadow inputs

We mainly conduct comparisons on our binary shadow dataset using a vertical-down viewpoint because previous works that adopt a depth map representation work better at a vertical-down camera. For completeness, we provide qualitative comparison results on our side-view binary shadow dataset in Fig. 2.

C.3. Quantitative comparison on RGB inputs

We show the quantitative results of SDPS-Net [1], Li et al. [4] and our method on our RGB dataset in Tab. 5. We achieve the lowest depth and normal reconstruction error.

D. Discussion on the handling of ground

D.1. Results on non-planar grounds

Given single-view images, the scale of the reconstructed scene is unconstrained. One possible way to resolve scale ambiguities is to calibrate the ground position, which is adopted in the evaluation of our method. We mainly evaluate planar grounds because they are common in real-world indoor setups and can easily calibrate by a checkerboard. However, our method is not inherently limited to planar grounds. When the ground is non-planar, we require that the depth map of the ground is known. We initialize the ground surface by regularizing the SDF at the ground to be

Method	Metric	Cactus	Rose	Bread	Sculptures	Surface	Relief	Avg
DeepShadow	Depth L1↓	0.0091	**0.0132**	0.0634	0.0334	0.0078	0.0067	0.0223
Ours	Depth L1↓	**0.0063**	0.0202	**0.0256**	**0.0199**	**0.0036**	**0.0053**	**0.0135**
DeepShadow	Normal MAE↓	20.79	24.32	**22.44**	26.66	12.15	**19.19**	20.93
Ours	Normal MAE↓	**20.02**	**18.35**	27.37	**23.19**	**7.04**	22.13	**19.68**

Table 2. Quantitative comparison of reconstruction quality on the DeepShadow dataset.

Method	Metric	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg
DeepShadow	Depth L1↓	0.7107	0.1855	1.6975	0.0123	0.4365	0.0134	0.8787	0.0810	0.5020
Ours	Depth L1↓	**0.0945**	**0.0532**	**1.1930**	**0.0054**	**0.0287**	**0.0119**	**0.0689**	**0.0408**	**0.1870**
DeepShadow	Normal MAE↓	51.88	18.98	**25.48**	21.51	38.42	20.81	31.87	28.71	29.71
Ours	Normal MAE↓	**18.08**	**13.27**	36.84	**10.51**	**24.94**	**12.01**	**24.23**	**21.83**	**20.21**

Table 3. Quantitative comparison of reconstruction quality on our binary shadow dataset.

Method	Metric	Cactus	Rose	Bread	Sculptures	Surface	Relief	Avg
DeepShadow	Depth nMZE↓	0.1001	0.0760	0.1166	0.1779	0.0952	**0.1424**	0.1180
Ours	Depth nMZE↓	**0.0392**	**0.0709**	**0.1001**	**0.0678**	**0.0381**	0.1427	**0.0765**

Table 4. Quantitative comparison on the DeepShadow dataset using normalized mean depth error.

Method	Metrics	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Avg
SDPS-Net	Depth L1↓	1.2627	0.8706	1.9185	0.5964	0.7254	0.1700	1.3678	0.4190	0.9163
Li et al.	Depth L1↓	1.2285	0.9467	1.8904	0.1372	0.6376	0.8242	1.2676	**0.1027**	0.8794
Ours	Depth L1↓	**0.0090**	**0.0383**	**0.7959**	**0.0145**	**0.0316**	**0.0057**	**0.0419**	0.1360	**0.1341**
SDPS-Net	Normal MAE↓	31.90	31.59	55.65	42.10	39.00	31.11	34.92	45.21	38.94
Li et al.	Normal MAE↓	14.72	25.93	**34.60**	9.31	21.77	43.49	25.68	13.34	23.61
Ours	Normal MAE↓	**7.65**	**17.09**	37.73	**6.70**	**17.87**	**9.21**	**11.95**	**12.02**	**15.03**

Table 5. Quantitative comparison of reconstruction quality on our RGB dataset.

0. As shown in Fig. 3, our method successfully reconstructs the object shapes in the presence of bumpy grounds.

D.2. Comparison between known and unknown grounds

To investigate the effect of the ground, we compare results with known and unknown grounds under different input types. As shown in Fig. 5, our method still achieves reasonable reconstruction when the ground is unknown, but the reconstruction exhibits a scale drift, especially when using directional light inputs. When the scale of the reconstruction deviates, its quality also decreases, possibly because it only occupies a small portion of the scene bounding sphere. Therefore, we choose to calibrate the ground in the evaluation to obtain scale-accurate reconstruction under arbitrary input types.

Number of images	Depth L1↓	Normal MAE↓
3	0.1427	28.03
5	0.0216	10.52
10	0.0189	8.88
20	0.0127	7.59
50	**0.0074**	**7.01**

Table 6. Reconstruction quality using different numbers of input images.

E. Additional evaluation

E.1. Analysis on the number of input images

To investigate our method's robustness, we evaluate it on the *Chair* scene using different numbers of input images. As shown in Fig. 6 and Tab. 6, our method can reconstruct reasonable geometry under five input images. When the input image number increases, the reconstructed structures become more accurate. In general, our method is robust to the number of input images.

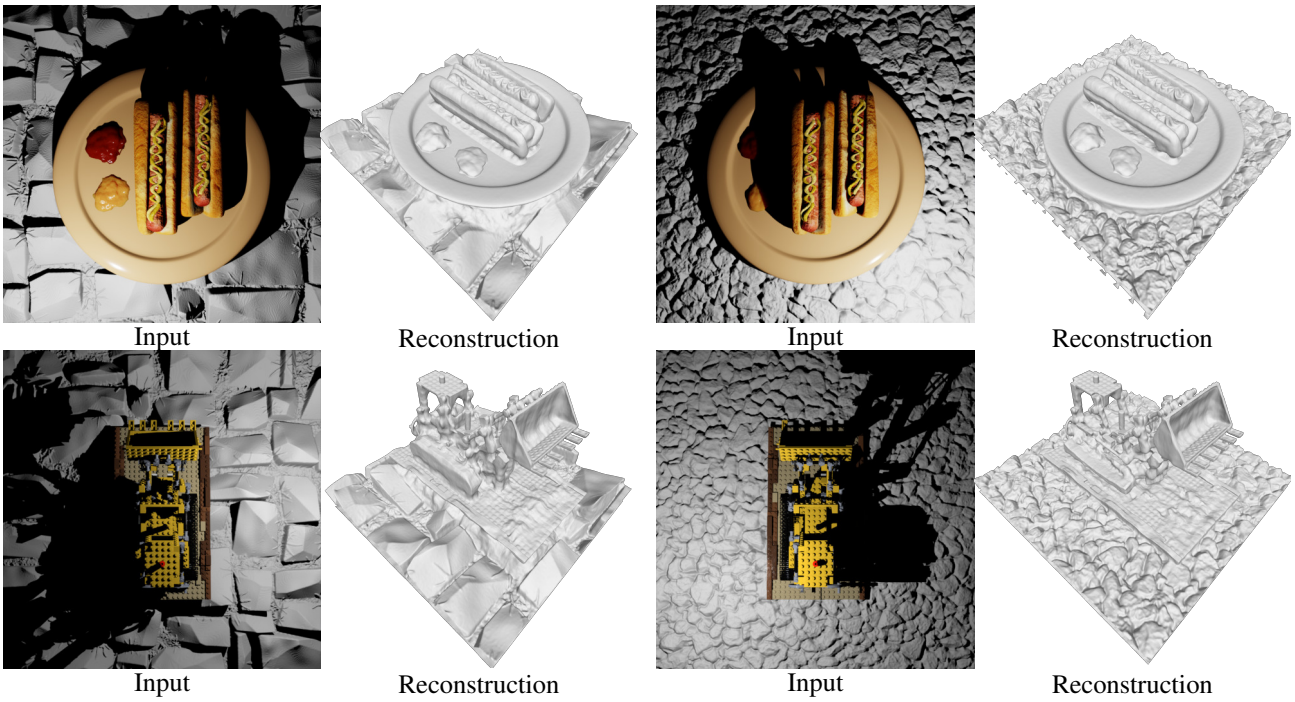


Figure 3. Results in the presence of bumpy grounds.

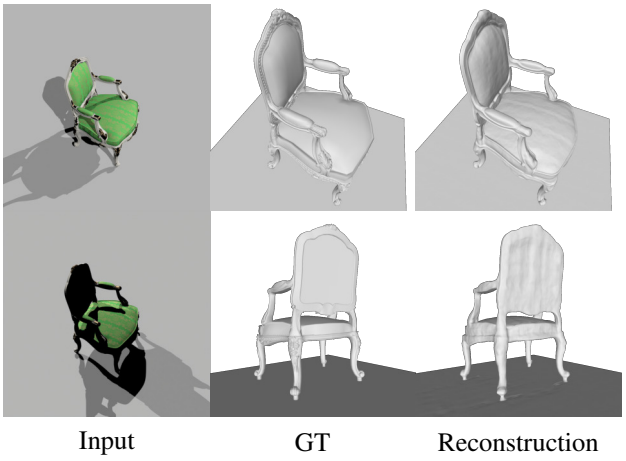


Figure 4. Results on the scene illuminated by two lights.

E.2. Effect of foreground and background shadows in reconstruction

To investigate how the supervision of foreground and background shadows affects shape reconstruction, we compare our method on the *Lego* scene with two variants that only supervise the background or foreground shadows. As shown in Fig. 7, when we only supervise shadows cast on the ground, we cannot reconstruct detailed structures on the top of the bulldozer. The middle part is also missing, as it mainly casts shadows on the object itself. When we only supervise foreground shadows, we can reconstruct the detailed structures, but the reconstructed bulldozer shovel is at an incorrect depth. As shown in Tab. 7, our method achieves the lowest reconstruction error when supervising foreground and background shadows. The two parts of shadows are indispensable in accurate shape reconstruction.

	Depth L1↓	Normal MAE↓
Back only	0.05827	29.93
Fore only	0.13569	23.94
Ours	**0.02955**	**19.59**

Table 7. Reconstruction quality when supervising only background or foreground shadows.

E.3. Results on scene illuminated by two lights

We mainly evaluate our method illuminated by one known light. However, our method can be extended to handle multiple known lights. As shown in Fig. 4, by supervising the sum of the incoming radiance of two lights, our method can still reconstruct a complete 3D shape of the chair.

F. Applications

Our method can reconstruct shapes and materials from single-view RGB images. Therefore, it supports multiple

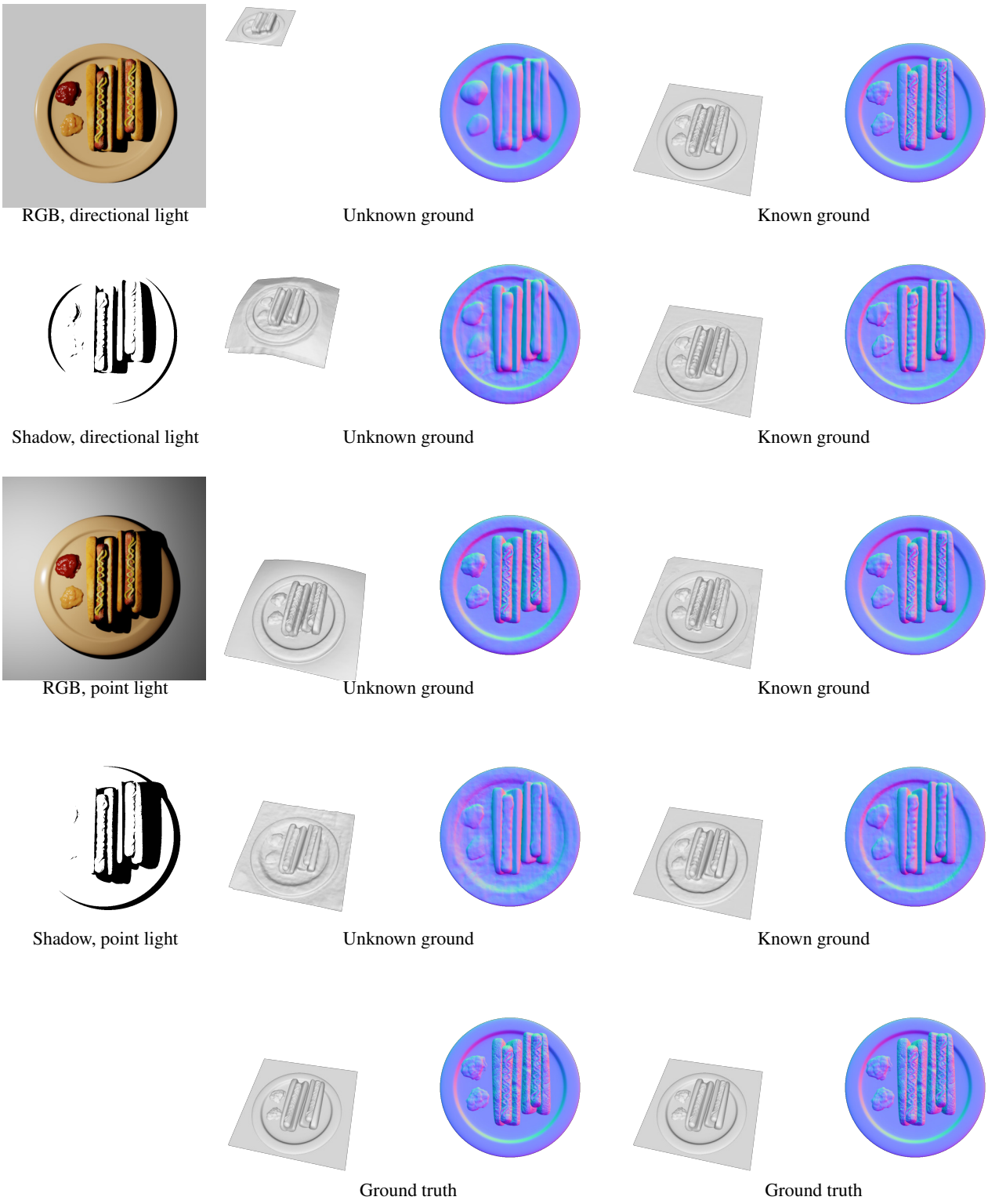


Figure 5. Comparison between known and unknown grounds.

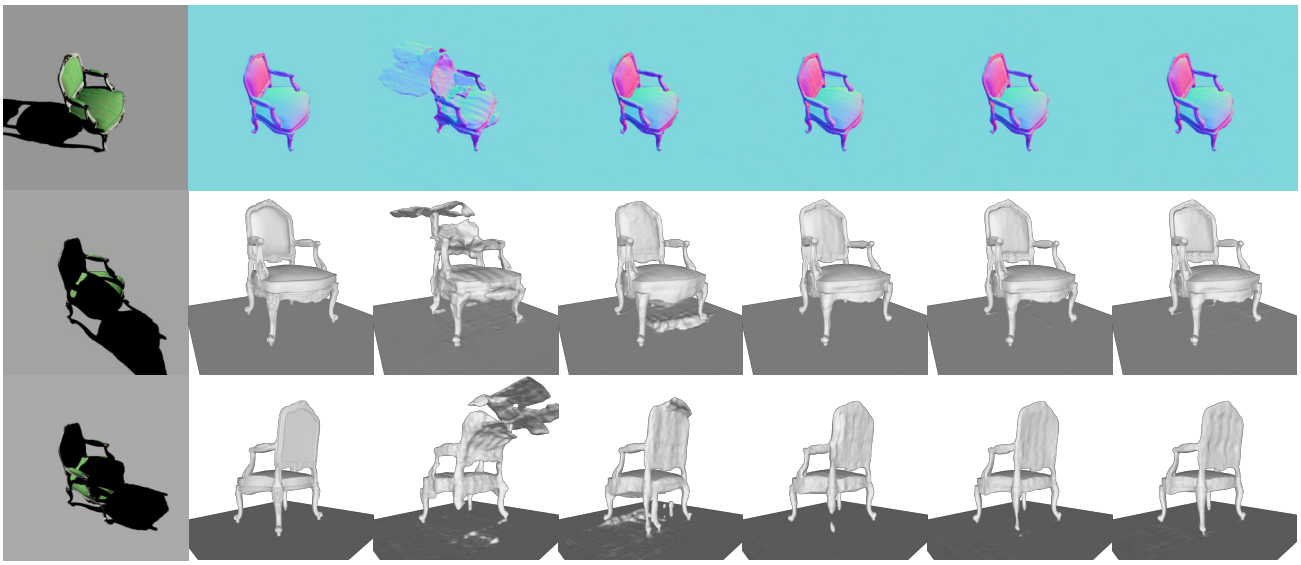


Figure 6. Analysis on different numbers of input images.

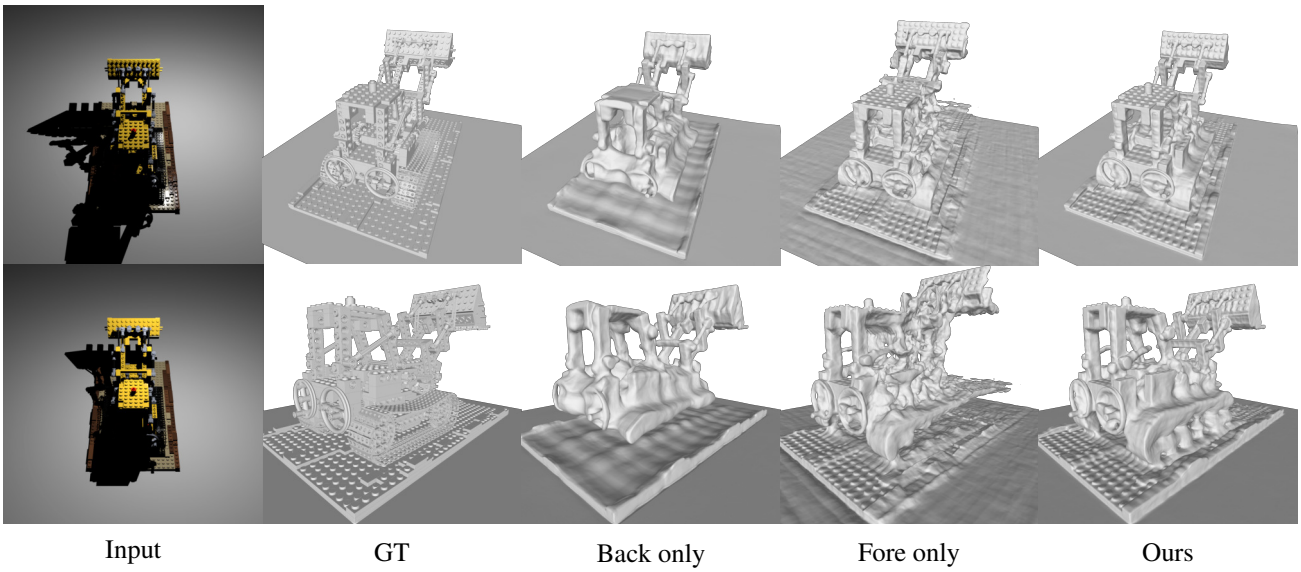


Figure 7. Comparison of shape reconstruction when supervising only background or foreground shadows.

applications, such as relighting using a point light or an environment map and material editing. In Fig. 10, we show that our method generates plausible results in these applications. Please also see the supplementary video for more results.

G. Discussion on surface locating method

We use NeuS-like volume rendering for shadow rays due to its wider basin of convergence [5], which helps discover better reconstructions. However, for camera rays, straightforward NeuS-like volumetric sampling is impractically complex because each sample is costly and the sam-

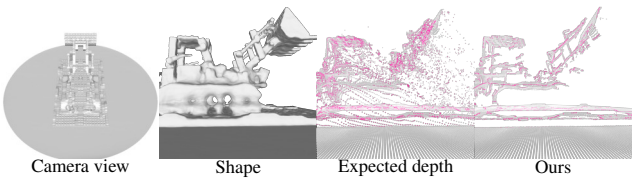


Figure 8. Visualized intersections of the *same* SDF (column 2) using the viewpoint in column 1. Boundaries are shown in magenta.

ple count is too large. An alternative method to our proposed surface intersection is presented in [10], which computes expected terminated depth by weighting depth samples by volume densities. Both "expected depth" [10] and our method are differentiable and reduce the sample count. However, we initially tried "expected depth" in early experiments and found that it computes incorrect "averaged" intersections at surface boundaries (Fig. 8 column 3). This greatly hindered optimization, as shown in the qualitative comparison in Fig. 9. By incorporating implicit differentiation [8] with edge sampling [9], our framework computes fully differentiable, correct intersections with a reasonable sample count (Fig. 8 column 4).

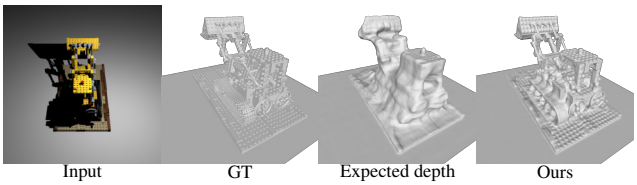


Figure 9. Qualitative comparison of reconstructed shape between "expected depth" and our method.

H. Synthetic dataset examples

In Fig. 11, we show different data types from our synthetic dataset.

I. Real dataset examples

In Fig. 12, we show the objects, capture setup, and example images from our real dataset.

J. Social impact

As our method targets shape reconstruction from single-view inputs, it could be extended to be misused for improper surveillance. In particular, 3D shapes can be reconstructed by exploiting shadows on the visible surface, revealing scenes beyond the camera's line of sight.

References

[1] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K. Wong. Self-calibrating deep photometric stereo networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8739–8747. Computer Vision Foundation / IEEE, 2019. 2

[2] Asaf Karnieli, Ohad Fried, and Yacov Hel-Or. Deepshadow: Neural shape from shadow. *CoRR*, abs/2203.15065, 2022. 1

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[4] Junxuan Li and Hongdong Li. Neural reflectance for shape recovery with shadow handling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16200–16209. IEEE, 2022. 2

[5] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4), jul 2019. 6

[6] 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 Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020. 1

[7] 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. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27171–27183, 2021. 1

[8] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 7

[9] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. IRON: inverse rendering by optimizing neural sdfs and materials from photometric images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5555–5564. IEEE, 2022. 1, 7

[10] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6):237:1–237:18, 2021. 7

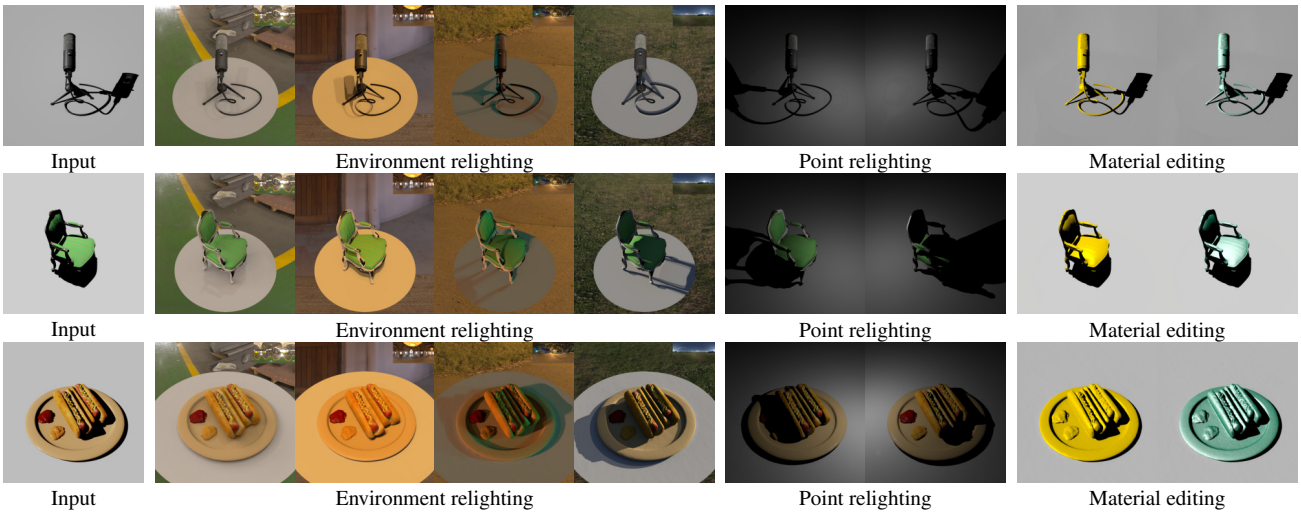


Figure 10. Applications.

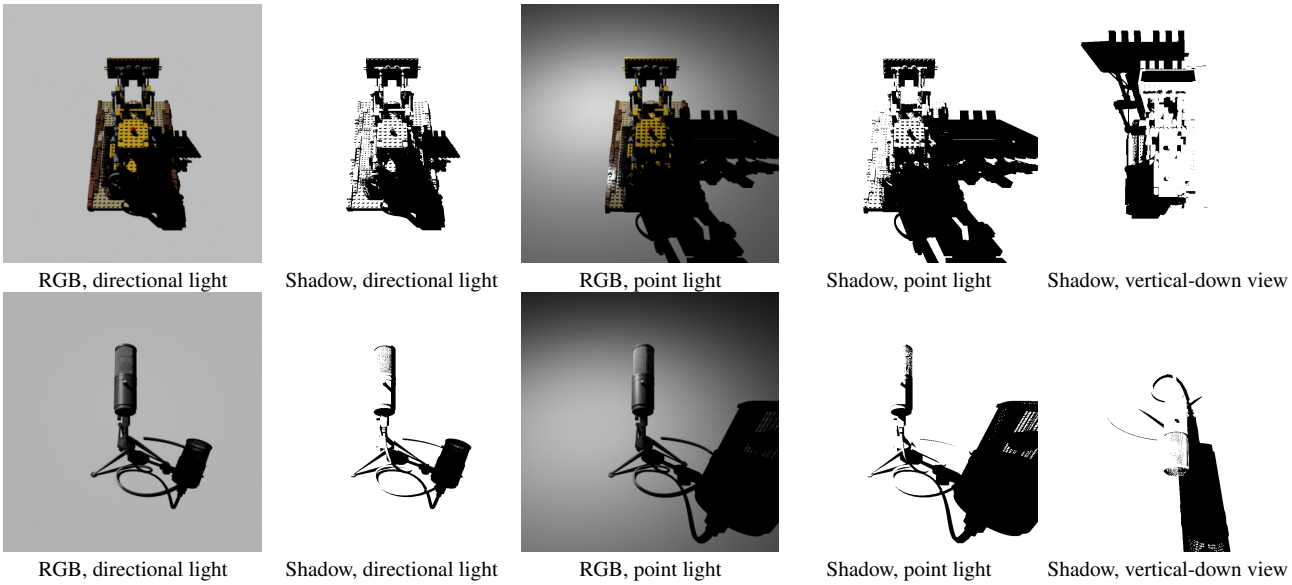
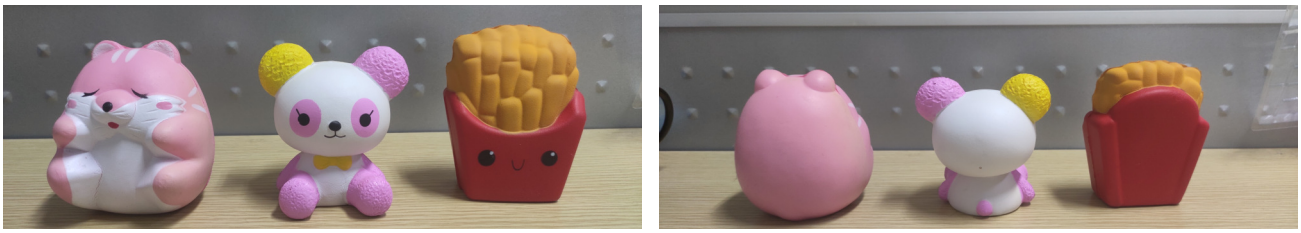


Figure 11. Example data from our synthetic dataset.



Captured objects

Capture setup

Example input images

Figure 12. More details of our real dataset.