ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision

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

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

Abstract

By supervising camera rays between a scene and multi-view image planes, NeRF reconstructs a neural scene representation for the task of novel view synthesis. On the other hand, shadow rays between the light source and the scene have yet to be considered. Therefore, we propose a novel shadow ray supervision scheme that optimizes both the samples along the ray and the ray location. By supervising shadow rays, we successfully reconstruct a neural SDF of the scene from single-view pure shadow or RGB images under multiple lighting conditions. Given single-view binary shadows, we train a neural network to reconstruct a complete scene not limited by the camera's line of sight. By further modeling the correlation between the image colors and the shadow rays, our technique can also be effectively extended to RGB inputs. We compare our method with previous works on challenging tasks of shape reconstruction from single-view binary shadow or RGB images and observe significant improvements. The code and data will be released.

1. Introduction

Neural field [40] has been used for 3D scene representation in recent years. It achieves remarkable quality because of the ability to continuously parameterize a scene with a compact neural network. The neural network nature makes it amenable to various optimization tasks in 3D vision, including long-standing problems like image-based [26, 48] and point cloud-based [24, 29] 3D reconstruction. So more and more works are using neural fields as the 3D scene representation for various related tasks.

Among these works, NeRF [25] is a representative method that incorporates a part of physically based light transport [36] into the neural field. The light transport describes light travels from the light source to the scene and then from the scene to the camera. NeRF considers the latter part to model the interaction between the scene and the cameras along the *camera rays* (rays from the camera through the scene). By supervising these camera rays of different viewpoints with the corresponding recorded images, NeRF optimizes a neural field to represent the scene. Then NeRF casts camera rays from novel viewpoints through the optimized neural field to generate novel-view images.

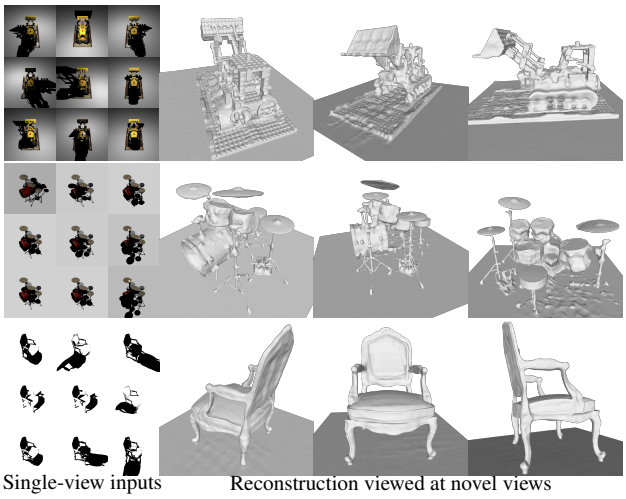


Figure 1. Our method can reconstruct neural scenes from single-view images captured under multiple lightings by effectively leveraging a novel shadow ray supervision scheme.

However, NeRF does not model the rays from the scene to the light source, which motivates us to consider: can we optimize a neural field by supervising these rays? These rays are often called *shadow rays* as the light emitted from the light source can be absorbed by scene particles along the rays, resulting in varying light visibility (a.k.a. shadows) at the scene surface. By recording the incoming radiance at the surface, we should be able to supervise the shadow rays to infer the scene geometry.

Given this observation, we derive a novel problem of supervising the shadow rays to optimize a neural field representing the scene, analogizing to NeRF that models the camera rays. Like multiple viewpoints in NeRF, we illuminate the scene multiple times using different light directions to obtain sufficient observations. For each illumination, we use a fixed camera to record the light visibility at the scene surface as supervision labels for the shadow rays. As rays

connecting the scene and the light source march through the 3D space, we can reconstruct a complete 3D shape not constrained by the camera's line of sight.

We solve several challenges when supervising the shadow rays using camera inputs. In NeRF, each ray's position can be uniquely determined by the known camera center, but shadow rays need to be determined by the scene surface, which is not given and has yet to be reconstructed. We solve this using an iterative updating strategy, where we sample shadow rays starting at the current surface estimation. More importantly, we make the sampled locations differentiable to the geometry representation, thus can optimize the starting positions of shadow rays. However, this technique is insufficient to derive correct gradients at surface boundaries with abrupt depth changes, which coincides with recent findings in differentiable rendering [19,22,51]. Thus, we compute surface boundaries by aggregating shadow rays starting at multiple depth candidates. It remains efficient as boundaries only occupy a small amount of surface, while it significantly improves the surface reconstruction quality. In addition, RGB values recorded by the camera encode the outgoing radiance at the surface instead of the incoming radiance. The outgoing radiance is a coupling effect of light, material, and surface orientation. We propose to model the material and surface orientation to decompose the incoming radiance from RGB inputs to achieve reconstruction without needing shadow segmentation (Row 1 and 2 in Fig. 1). As material modeling is optional, our framework can also take binary shadow images [16] to achieve shape reconstruction (Row 3 in Fig. 1).

We compare our method with previous single-view reconstruction methods (including shadow-only and RGB-based) and observe significant improvements in shape reconstruction. Theoretically, our method handles a dual problem of NeRF. So, comparing the corresponding parts of the two techniques can inspire readers to get a deeper understanding of the essence of neural scene representation to a certain extent, as well as the relationship between them.

Our contributions are:

- A framework that exploits light visibility to reconstruct neural SDF from shadow or RGB images under multiple light conditions.
- A shadow ray supervision scheme that embraces differentiable light visibility by simulating physical interactions along shadow rays, with efficient handling of surface boundaries.
- Comparisons with previous works on either RGB or binary shadow inputs to verify the accuracy and completeness of the reconstructed scene representation.

2. Related Work

Neural fields for 3D reconstruction. A neural field [40] typically parameterizes a 3D scene with a multi-layer perceptron (MLP) network that takes scene coordinates as input. It can be supervised with 3D constraints like point clouds [24, 29] to reconstruct an implicit representation of 3D shapes. It is also possible to optimize a neural field from multi-view images by differentiable rendering [26,48]. NeRF [25] demonstrates remarkable novel-view synthesis quality on scenes with complex geometry. However, the density representation in NeRF is not convenient for regularizing and extracting scene surfaces. Thus, [27, 38, 47] propose to combine NeRF with surface representation to reconstruct high-quality and well-defined surfaces. While all the above works require known camera viewpoints, [10, 23, 39] explore to optimize camera parameters with the neural field jointly.

NeRF does not model the light source and assumes the scene emits the light. This assumption is suitable for view synthesis but not relighting. Several works extend NeRF to relighting, where shadows are an essential factor. [2, 3, 51] require co-located camera-light setup to avoid shadows in captured images. [4, 5, 52] assume smooth environment lights and ignore shadows. [9,31,35,44,46,54] adopt neural networks conditioned on the light direction to model light-dependent shadows. Among them, [9, 44, 46, 53, 54] first reconstruct geometry using multi-view stereo and compute shadows using fixed geometry. None of the works refine the geometry to match the shadows in the captured images. However, we show that it is possible to reconstruct a complete 3D shape from scratch by exploiting information in the shadows.

Single-view reconstruction. [15, 42, 49] explore reconstructing neural fields from a few or a single image, but they require data-driven prior in the pretrained networks thus are in a different scope from ours. Non-line-of-sight imaging [28, 34, 41] adopts a transient sensor to capture time-resolve signals, which enables reconstructing the scene beyond the camera's view frustum. Photometric stereo [8, 21] reconstructs surface normals from images captured under directional lights. Normals can be integrated to produce a depth map but require non-trivial processing [6, 7].

Shape from Shadows. Shadows indicate varying incoming radiance caused by occlusion, providing scene geometry cues. There is a long history of reconstructing shapes from shadows as 1D curves [14, 17], 2D height maps [12,30,33,50] and 3D voxel grids [20,32,43]. These works typically capture under different light directions to get sufficient observations of shadows. Shadows show the potential in these works to reconstruct surface details [50] and intricate thin structures [43]. The most recent work in this area is DeepShadow [16], which reconstructs a neural depth map from shadows. A different setup with fixed lighting

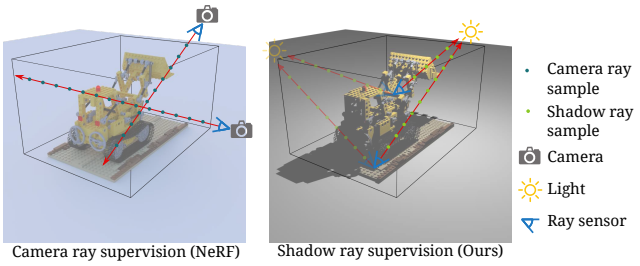


Figure 2. Different kinds of ray supervisions.

but multiple viewpoints is also adopted by [37], which integrates Shadow Mapping to reconstruct a neural representation. Concurrently and independently, [45] proposes to simultaneously use shading and shadows in neural field reconstruction. In particular, they compute shadows at a *non-differentiable* surface point located by root finding, making it rely on a differentiable shading computation. We propose fully differentiable shadow ray supervision that optimizes both the shadow ray samples and the surface point, enabling neural field reconstruction from either pure shadows or RGB images.

3. Ray Supervision in Neural Fields

This section first reveals the essence in NeRF [25] training as supervising *camera rays*. From there, we discover a ray supervision scheme generalizable to arbitrary rays. The scheme makes it feasible to supervise *shadow rays* to optimize a neural scene representation.

3.1. Camera ray supervision in NeRF

NeRF aims to optimize a neural field to fit a scene of interest. To obtain observations of the scene, NeRF requires recording images at multiple camera viewpoints with known camera parameters. Each image pixel records the incoming radiance of a camera ray that passes through the known camera center from a known direction. Since NeRF does not model the external light source and assumes the light is emitted from scene particles to simplify the modeling of a scene with fixed lighting, the incoming radiance is actually attributed to the combined effect of light absorption and emission by the infinitesimal particles along the camera ray.To fit observations, NeRF uses differentiable volume rendering to simulate the same camera ray in the neural field. NeRF uses quadrature to approximate the continuous integral in volume rendering by sampling N distances $t_1, \cdots, t_N$, started from the camera center $\boldsymbol{o}$ along the camera ray direction $\boldsymbol{v}$. With the scene density σ_i and emitted radiance $\boldsymbol{c}_i$ at each sample point $\boldsymbol{p}(t_i) = \boldsymbol{o} + t\boldsymbol{v}$, the estimated radiance C at the camera can be formulated as follows,

$$C(\boldsymbol{o}, \boldsymbol{v}) = \sum_{i=1}^{N} T_i \alpha_i \boldsymbol{c}_i, \tag{1}$$

where $\alpha_i = 1 - \exp\left(-\sigma_i(t_{i+1} - t_i)\right)$ is the discrete opacity and $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \cdot (t_{j+1} - t_j))$ indicates the light transmittance, *i.e.*, the proportion of the emitted light reach the camera from the point $\mathbf{p}(t_i)$. The incoming radiance recorded at the pixel can be used to supervise the simulated radiance C. NeRF trains on a random subset of camera rays in each iteration. As the neural field receives supervision signals from many camera rays marching in different viewpoint directions, it obtains sufficient scene information to optimize the neural field in the space these rays go through.

3.2. Generalized ray supervision

The reason that NeRF can supervise the camera rays to optimize a neural field is that multi-view cameras record the radiance as labels of the rays. Moreover, as each camera is calibrated, each recorded ray's 3D location and orientation are well-defined. We can regard each pixel of the multi-view camera as a "ray sensor" recording the incoming radiance of a particular ray because each pixel is used independently in training. These ray sensors are the key to the NeRF techniques. More generally, if we let the "ray sensors" record other kinds of rays in the scene, it is also possible to achieve scene reconstruction. This motivates us to consider whether we could supervise other rays and design ray sensors to record their radiance.

3.3. Shadow ray supervision

Since camera rays have achieved great success in neural scene reconstruction, as the counterpart in light transport, the ray connecting the scene and the light source, a.k.a. *shadow rays*, should also be able to be used to reconstruct neural scenes. We first consider an ideal setup where many hypothetical ray sensors are placed in the scene at different but known locations, as shown in Fig. 2.To observe the scene along shadow rays, we illuminate the scene with a known directional light. Each ray sensor captures one ray that passes the sensor from the light direction. Different from NeRF, as we model the light source, we assume the scene does not emit light, which is more physically correct and can simplify the following process. Therefore, the incoming radiance at a ray sensor is from the light emitted from the light source and absorbed by infinitesimal particles along the ray. Using similar quadrature as Eq. (1), we can express the incoming radiance simulated in the neural field as

$$C_{\text{in}}(\boldsymbol{x}, \boldsymbol{l}) = L \prod_{i=1}^{N} (1 - \alpha_i), \tag{2}$$

where and L is the intensity of the light source, $\boldsymbol{x}$ is the location of a ray sensor and $\boldsymbol{l}$ is the light direction. To ob-

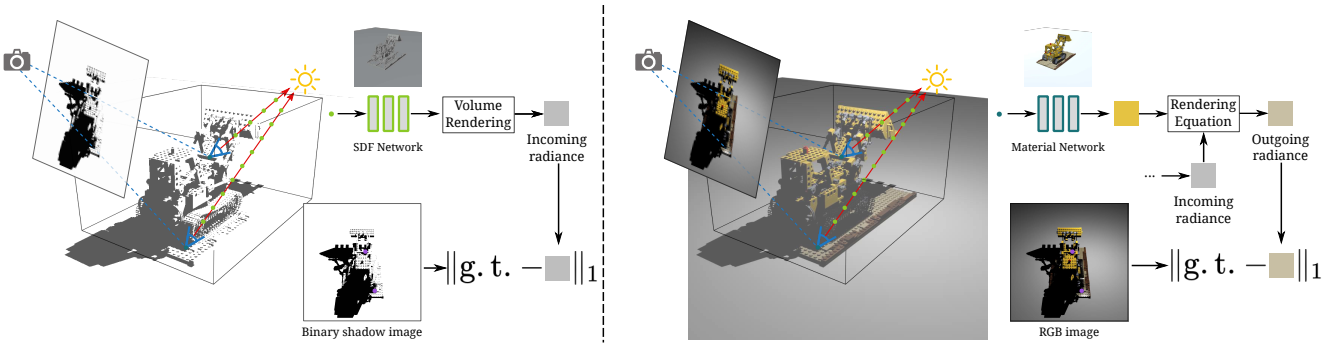


Figure 3. Overview of our method. The proposed shadow ray supervision can be applied to single-view neural scene reconstruction on two input types: binary shadow images (left) and RGB images (right). For binary inputs, we first compute the incoming radiance of a shadow ray using volume rendering. Then, we construct a photometric loss to train the neural SDF to match the shadows. For RGB inputs, we further use a material network and a rendering equation to convert the incoming radiance to the outgoing radiance. The SDF and material networks are trained to match the ground truth colors.

tain sufficient information to constrain the optimization, we require the shadow rays to march the scene in different directions. Therefore, we illuminate the scene with multiple light directions one by one and record the incoming radiance each time. As this ray supervision scheme has been demonstrated successful by NeRF, it is also promising to reconstruct a neural scene here.

4. Shadow ray supervision with a single-view camera

Note that in the above formulation, we adopt hypothetical ray sensors to record the incoming radiance in the light direction and at the known positions in the scene. These ray sensors are ideal because they are placed at desired positions in the scene and always face toward the light. Under these strong assumptions, it is possible to get sufficient supervision for the shadow rays. However, these ray sensors are hard to implement in an actual setup, unlike NeRF, where the ray sensors are just the pixels of multi-view cameras. In this section, we will propose a more practical setting for a real capture setup.

In general, we conduct shadow ray supervision from a single-view camera, which can be a practical alternative to the ray sensors in the previous formulation. We similarly illuminate the scene with a light in direction $\boldsymbol{l}$. The scene is assumed to be opaque, and thus the camera captures exactly the outgoing radiance at visible surfaces. We consider two types of camera inputs: binary shadow images [16] and RGB images, as shown in Fig. 3. Binary shadow images use outgoing radiance to determine whether a point is illuminated, which can be seen as an approximation of binarized incoming radiance. RGB images are a more complex case that records a combined effect of material, surface orientation, and incoming radiance. We will first consider the more straightforward case when we can obtain the incoming radiance at visible surfaces from binary shadow images and then handle the more complex RGB images.

However, another challenge is that, given the recorded pixel values, we still do not know the exact depths of the visible surface points. Thus, we are given scene observations as outgoing radiance in the camera viewing direction at points at unknown depths. This problem is handled by the proposed techniques that determine the depth and relate outgoing radiance to incoming radiance.

We represent the scene as the zero level set of a signed distance function (SDF) $\mathcal{S} = \left\{\boldsymbol{u} \in \mathbb{R}^3 | f(\boldsymbol{u}) = 0\right\}$, where f is a neural network that regresses the signed distance at the input 3D position. The 3D points visible by the camera are the first intersections between the camera rays and the SDF. Note that here the camera rays are only used to determine the surface points but not to construct supervision, which is the job of shadow rays. Specifically, ray marching is used to compute the intersection point $\boldsymbol{x}$ at the current SDF. Then we can compute the incoming radiance $C_{\text{in}}(\boldsymbol{x}, \boldsymbol{l})$ at the intersection by volume rendering. As we are modeling an SDF instead of a density field, we replace the discrete opacity α_i in Eq. (2) by the one derived from the SDF following NeuS [38], as

$$\alpha_i = \max\left(1 - \frac{\Phi_s(f(\boldsymbol{p}(t_{i+1})))}{\Phi_s(f(\boldsymbol{p}(t_i)))}, 0\right), \tag{3}$$

where $\Phi_s(x) = (1 + e^{-sx})^{-1}$ is the sigmoid function and s is a learnable scalar parameter that controls whether Eq. (2) approaches volume rendering or surface rendering.

Differentiable intersection points. To locate the intersection point $\boldsymbol{x}$ given the SDF, ray marching is the most straightforward choice. However, it is non-differentiable and thus cannot be used in neural network training. To optimize the intersection points using backpropagated gradients, we use implicit differentiation [1, 48], which makes

the intersection point differentiable to the SDF network parameters as

$$\widehat{\boldsymbol{x}} = \boldsymbol{x} - \frac{\boldsymbol{v}}{\boldsymbol{n} \cdot \boldsymbol{v}} f(\boldsymbol{x}), \tag{4}$$

where $\boldsymbol{v}$ is the camera ray direction and $\boldsymbol{n} = \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ is the surface normal derived from the SDF network. Then, we use $C_{\text{in}}(\widehat{\boldsymbol{x}}, \boldsymbol{l})$ as the differentiable radiance at intersection $\boldsymbol{x}$. As $\boldsymbol{x}$ acts as the start position of a shadow ray, it can be optimized by gradients from Eq. (2). When the computed incoming radiance $C_{\text{in}}(\widehat{\boldsymbol{x}}, \boldsymbol{l})$ does not agree with the supervision, the SDF network can optimize both the signed distances along the shadow ray and the starting position of the ray to fit the observation.

Multiple shadow rays at boundaries. We observe that $\widehat{\boldsymbol{x}}$ in Eq. (4) only differentiates along the camera direction $\boldsymbol{v}$. When supervising $C_{\text{in}}(\widehat{\boldsymbol{x}}, \boldsymbol{l})$ with the recorded images, it will cause issues at pixels corresponding to surface boundaries. At surface boundaries, a pixel spans disconnected regions at different depths, where each region occupies a part of the pixel's area. When $\widehat{\boldsymbol{x}}$ moves perpendicular to the camera direction $\boldsymbol{v}$, it can significantly change the computed radiance at surface boundaries by changing the area proportional to each region. If we only sample one shadow ray started at one region, it will lead to incorrect gradients similar to the case in differentiable mesh rendering [19, 22].

Therefore, we first obtain a pixel subset Ω corresponding to surface boundaries, and a differentiable area ratio w for each boundary pixel using the surface walk procedure in [51]. Then we locate two intersections $\boldsymbol{x}_{\text{n}}$ and $\boldsymbol{x}_{\text{f}}$ at different depths within the pixel and compute their incoming radiance $C_{\text{in}}(\widehat{\boldsymbol{x}}_{\text{n}}, \boldsymbol{l})$ and $C_{\text{in}}(\widehat{\boldsymbol{x}}_{\text{f}}, \boldsymbol{l})$ respectively. When computing the incoming radiance corresponding to pixel p, we average the incoming radiance at boundary pixels as

$$\widehat{C}_{\text{in}} = \begin{cases} C_{\text{in}}(\widehat{\boldsymbol{x}}, \boldsymbol{l}) & p \notin \Omega \\ wC_{\text{in}}(\widehat{\boldsymbol{x}}_{\text{n}}, \boldsymbol{l}) + (1-w)C_{\text{in}}(\widehat{\boldsymbol{x}}_{\text{f}}, \boldsymbol{l}) & p \in \Omega \end{cases} \tag{5}$$

Then, we can supervise the computed incoming radiance $\widehat{C}_{\text{in}}$ with a pixel I_{s} on a binary shadow image as

$$\mathcal{L}_{\text{shadow}} = \|\widehat{C}_{\text{in}} - I_{\text{s}}\|_1. \tag{6}$$

Decomposing incoming radiance by inverse rendering. To cope with RGB images, we incorporate an inverse rendering equation consisting of material, incoming radiance, and surface orientation. We model the non-Lambertian BRDF as a diffuse component $\boldsymbol{\rho}_d$ and a specular component $\boldsymbol{\rho}_s$. Following [21, 44], we use a weighted combination of spherical Gaussian basis to represent the specular component $\boldsymbol{\rho}_s$ as $\boldsymbol{\rho}_s = \boldsymbol{y}^T D(\boldsymbol{h}, \boldsymbol{n})$, where $\boldsymbol{h} = \frac{\boldsymbol{l}-\boldsymbol{v}}{\|\boldsymbol{l}-\boldsymbol{v}\|}$ is the half-vector between light direction $\boldsymbol{l}$ and view direction $-\boldsymbol{v}$, D is the specular basis and $\boldsymbol{y}$ is the specular coefficients. We model another MLP network g to regress material properties $(\boldsymbol{\rho}_d, \boldsymbol{y}) = g(\boldsymbol{x})$ at surface location $\boldsymbol{x}$.

The outgoing radiance at point $\boldsymbol{x}$ can be formulated as

$$C(\boldsymbol{x}, -\boldsymbol{v}) = (\boldsymbol{\rho}_d + \boldsymbol{\rho}_s) C_{\text{in}}(\boldsymbol{x}, \boldsymbol{l})(\boldsymbol{l} \cdot \boldsymbol{n}) \tag{7}$$

The outgoing radiance $\widehat{C}$ corresponding to a boundary pixel is the weighted combination of multiple samples, similar to Eq. (5). Now we can supervise the computed radiance using a pixel I_{r} on an RGB image as

$$\mathcal{L}_{\text{rgb}} = \|\widehat{C} - I_{\text{r}}\|_1 \tag{8}$$

Light source modeling. Our technique supports directional light or point light as the light source to compute the incoming radiance in Eq. (2). For directional light, the light direction $\boldsymbol{l}$ and intensity L are known and uniform for all shadow rays. For point light, we calculate the light direction and intensity at point $\boldsymbol{x}$ as

$$L = \frac{L_p}{\|\boldsymbol{q} - \boldsymbol{x}\|_2^2}, \; \boldsymbol{l} = \frac{\boldsymbol{q} - \boldsymbol{x}}{\|\boldsymbol{q} - \boldsymbol{x}\|_2} \tag{9}$$

where L_p is a scalar point light intensity and $\boldsymbol{q}$ is the light location.

Training. To regularize the network to output valid SDF, we add an Eikonal loss [13] on M sample points as

$$\mathcal{L}_{\text{eik}} = \frac{1}{M} \sum_{i}^{M} (\|\nabla f(\boldsymbol{p}_i)\|_2 - 1)^2. \tag{10}$$

We train the Eikonal loss with Eq. (6) or Eq. (8) depending on whether binary shadow images or RGB images are used as supervision.

Our technique is mainly evaluated on bounded scenes of an object on the ground. To bound the camera rays, we set camera rays that do not intersect with the SDF to intersect with the ground. To resolve the scale ambiguity from single-view inputs and reconstruct a scene with the accurate scale, we assume the ground plane's position and orientation are known. More discussion on the handling of the ground plane can be found in the supplementary material.

5. Experiments

5.1. Implementation details

We adopt an SDF MLP network similar to NeuS [38] for both the binary shadow inputs and RGB inputs. When handling RGB inputs, the SDF network outputs an extra 256-dimensional feature vector. It will be concatenated with 3D position and surface normal to regress diffuse and specular coefficients by another MLP network. During training, we randomly select four images in each batch, and for each image, 256 pixel positions are sampled as supervision signals. The camera ray intersection points are located by ray marching, and possible surface boundaries are computed

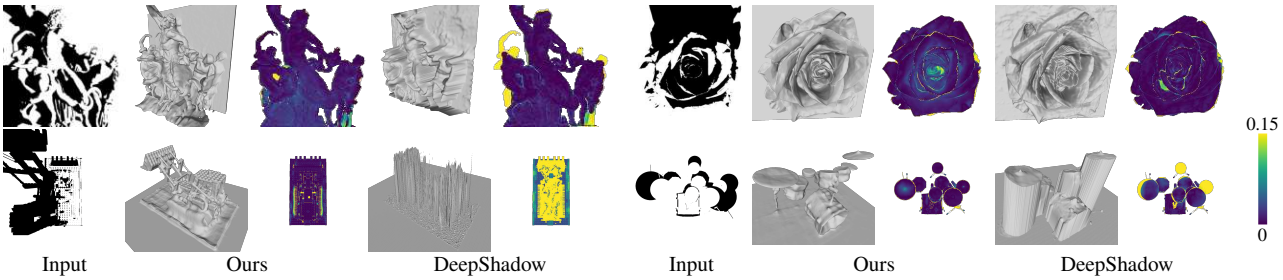


Figure 4. Comparison on binary shadow inputs. Each result's heat map shows error distribution compared to the ground truth depth map.

using a surface walk process [51] started at these intersection points. We train the network for 150k iterations, which takes about 24 hours on a single RTX 2080Ti. More implementation details can be found in the supplementary material.

5.2. Evaluation

To demonstrate the ability to leverage information from shadow rays in scene reconstruction, we evaluate our method on single-view binary shadow images and RGB images captured under multiple known light directions. We first present qualitative and quantitative comparisons with state-of-the-art methods supporting similar inputs. Then, we evaluate the effectiveness of the shadow ray supervision scheme with a comprehensive ablation study. Finally, we show more results and applications of the proposed method.

Dataset. The aforementioned experiments are performed on three datasets. First, we use the dataset released by DeepShadow [16], which contains binary shadow images of six scenes under different point lights. Each scene is terrain-like and captured by a vertical-down camera. For more complex scenes captured by other viewpoints, we find that no publicly available dataset satisfies our needs. Therefore, we construct new synthetic and real datasets for a thorough evaluation. For synthetic data, we render eight scenes using objects from the NeRF synthetic dataset [25]. Each test case is built by adding a horizontal plane to model the ground, placing the object on the plane, and rendering the scene using Blender [11]. We render binary shadow images and RGB images of resolution 800×800. To test different light types, we render each scene with 100 directional lights and 100 point lights. Our synthetic dataset features realistic materials with specular effects. Transparency and inter-reflections are disabled as these effects are beyond our assumption. We also capture a real dataset to investigate our method's applicability to real capture setups. For each scene, we place the object on the ground, illuminate the scene with only a handheld cellphone flashlight and capture it with a fixed camera. We capture around 40 RGB images when the handheld flashlight moves around the scene and obtain the light locations similarly to [3]. We place a checkerboard on the ground and capture one additional image with the same fixed camera to calibrate the ground. Please see Tab. 1 for a summary of used datasets.

	RGB	Binary shadow	Directional light	Point light
DeepShadow [16]		✓		✓
Our Synthetic	✓	✓	✓	✓
Our Real	✓			✓

Table 1. Datasets used in the evaluation.

Metrics. As the compared methods output depth maps or normal maps of the visible regions, we also evaluate the quality of single-view reconstruction by depth errors in L1 (Depth L1) and normal errors in mean angular error (Normal MAE) computed in the visible foreground region. It should be noted that as some compared methods output a depth map without a specific scale, Depth L1 is calculated after aligning the depth map to the ground truth using ICP.

5.2.1 Comparison on binary shadow inputs

On binary shadow images, we compare our method with DeepShadow [16], the only existing method that supports scene reconstruction from similar inputs. We find DeepShadow works better with a vertical-down camera, possibly because it represents the scene geometry as a depth map. Therefore, we conduct this experiment on the DeepShadow dataset and the test samples captured under a similar viewpoint in our synthetic dataset. Although this setup gives advantages to Deepshadow [16], qualitative and quantitative results show that our method achieves better shape reconstruction on both datasets. As shown at the top left of Fig. 4, our method achieves visually comparable results with DeepShadow [16] on reconstructing a terrain-like geometry. For more complex inputs, our method reconstructs more detailed and complete structures than DeepShadow [16], as shown at the bottom left of Fig. 4. Benefiting from the shadow ray supervision of the complex shadow cast by the occluded geometry, our method can reconstruct the invisible regions, as shown by the results at the bottom right. The results show that our method brings significant improvement in reconstructing complex

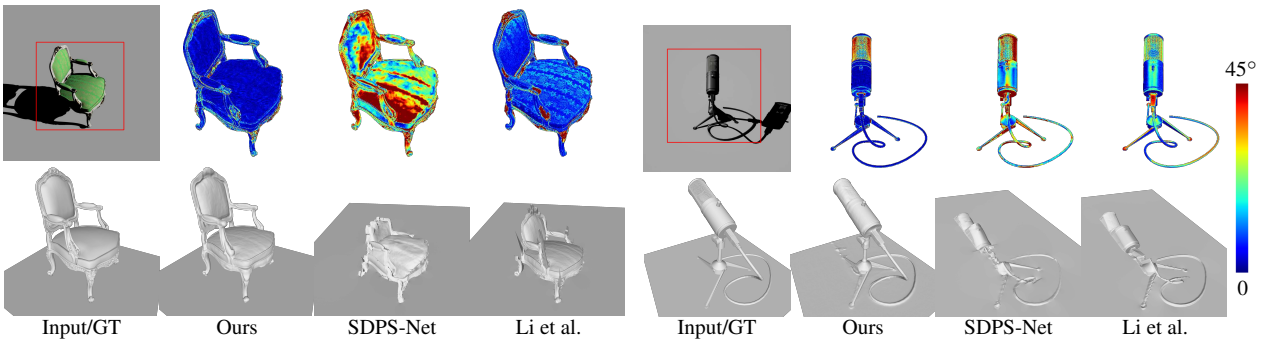


Figure 5. Comparison on RGB inputs. The heat maps in the first row show the error distribution compared to the ground truth normal map.

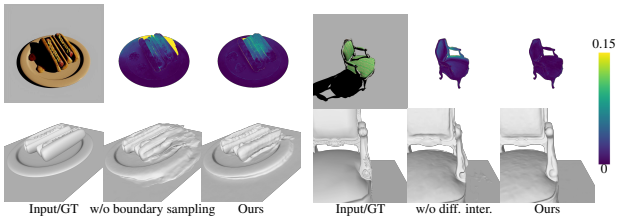


Figure 6. Qualitative comparison with different ablations.

Method	Metric	Avg
W/o diff. inter.	Depth L1↓	0.0812
W/o boundary sampling	Depth L1↓	0.1724
Ours Full	Depth L1↓	**0.0538**
W/o diff. inter.	Normal MAE↓	16.26
W/o boundary sampling	Normal MAE↓	25.75
Ours Full	Normal MAE↓	**12.76**

Table 2. Quantitative comparison of reconstruction quality between different ablations.

scenes. Please see the supplementary for the quantitative results. Note that our method requires the depth of the ground plane. This is also used by DeepShadow to initialize its depth map prediction.

5.2.2 Comparison on RGB inputs

On RGB inputs from our synthetic dataset, we compare our method with two state-of-the-art photometric stereo methods [8, 21] which also consider shadows. [8] is a deep-learning method that augments the training dataset with images under shadows, and [21] is a recent neural field method that considers shadows cast by the reconstructed depth map. Both achieve higher performance in photometric stereo with the leverage of shadows. Compared with these methods, our method can better leverage shape cues in the shadows to reconstruct shapes with more precise global structure as shown in Fig. 5. Thanks to the shadow ray supervision of 3D neural SDF representation, our method can better handle abrupt depth changes at surface boundaries. As shown in the table in the supplementary material, we achieve the lowest depth and normal reconstruction errors. The purpose of this experiment is not to compare these methods on the same inputs but to illustrate the efficiency of the proposed shadow ray supervision scheme in leveraging shadow information.

5.2.3 Ablation Study

To demonstrate the effectiveness of the proposed differentiable intersection points and boundary sampling strategy, we construct two ablations by removing the two techniques and comparing them with our complete method on our synthetic directional RGB inputs. In the first ablation, we only sample one shadow ray at a boundary pixel. As shown in the left half of Fig. 6, without boundary sampling, the reconstructed geometry will extrude along the image plane direction, leading to significant errors around the boundary. In the second ablation, we directly use the non-differentiable intersection points. From the right half of Fig. 6, we can see that the errors around the left arm increase as the network fail to update the depth using inaccurate backpropagated gradients. Quantitative results in Tab. 2 show that our proposed techniques greatly enhance the performance of geometry reconstruction.

5.2.4 More Results

We present more results to demonstrate the ability of the proposed method to reconstruct occluded geometry and synthesize images under novel lighting. We also test our method on handling more various inputs, including real images.

Reconstructing invisible geometry. Our method can reconstruct geometry that is not directly visible from the camera. As shown in Fig. 7, our method reconstructs more com-

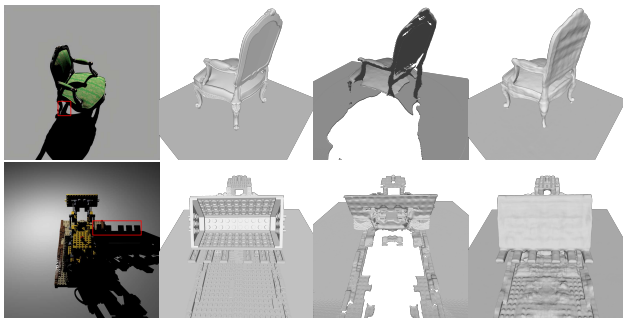


Figure 7. Results of invisible geometry reconstruction. The third column illustrates the region visible by the camera.

plete geometry than the visible region in the third column, *e.g.*, the invisible chair leg and the bulldozer blade. As these invisible shapes cast shadows captured by the camera (labeled by red boxes in Fig. 7), the corresponding shadow rays can supervise the shape to match the shadows.

Novel-light synthesis. After reconstructing the neural scene, we can re-render the scene under a novel light direction, as shown in Fig. 8. Besides shading and specular effects, we can generate accurate shadows on the ground and the object itself, consistent with the object's shape. The results also indicate that it is beneficial to integrate shadow ray supervision into a neural relighting pipeline. Please also see the supplementary video for continuous relighting results.

Results on more various inputs. In order to demonstrate the generalization of our method, we first test our method on more challenging synthetic data. As shown in Fig. 9, our method can reconstruct scenes with multiple objects (Column 2). Our method still successfully reconstructs some leaves and stems for inputs with extremely complex structures for single-view reconstruction (Column 1). We further apply our method to our real data. As shown in Fig. 10, our method reconstructs complete 3D shapes and accurate surface details from the simple setup and can handle the ground with non-trivial materials. Reconstructed results from real inputs can also generate realistic relighting results.

5.3. Limitations

The effectiveness of the proposed shadow ray supervision in reconstructing neural scenes is demonstrated by extensive experiments. However, as an early attempt to model shadow rays, our method is based on several assumptions. We assume the scene does not emit light and ignore interreflections to simplify light modeling. We observe that some thin structures are too complex that they can still be missing in our reconstruction. It is a general limitation and can be improved by the progress in neural SDF.

Figure 8. Results of novel-light synthesis.

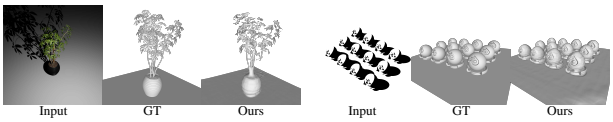


Figure 9. Results on more various inputs.

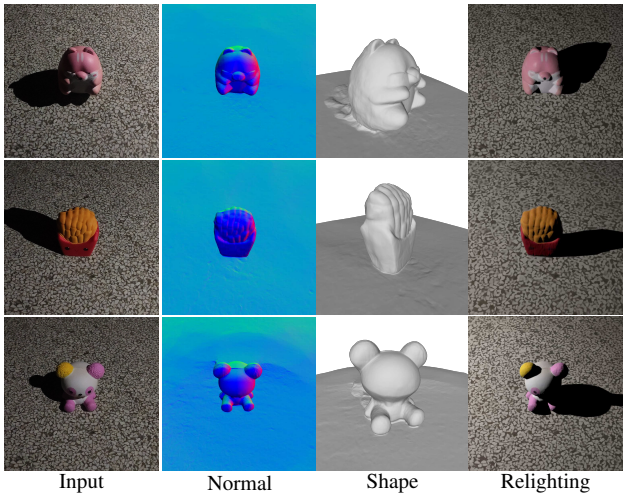


Figure 10. Results on real data.

6. Conclusion

Compared with NeRF supervising camera rays, we achieve fully differentiable supervision of shadow rays in a neural scene representation. This technique enables shape reconstruction from single-view multi-light observations and supports both pure shadow and RGB inputs. Our technique works well for both point and directional lights and can be used for 3D reconstruction and relighting. A multi-ray sampling strategy is proposed to handle challenges posed by surface boundaries in locating shadow rays. Experimental results show that our technique outperforms the SOTAs in both shape-from-shadow and photometric stereo, and it has the power to reconstruct scene geometries out of the line of sight.

References

[1] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2032–2041, 2019. 4

[2] Sai Bi, Zexiang Xu, Pratul P. Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *CoRR*, abs/2008.03824, 2020. 2

[3] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. 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 III*, volume 12348 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2020. 2, 6

[4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12664–12674. IEEE, 2021. 2

[5] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P. A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. 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 10691–10704, 2021. 2

[6] Xu Cao, Hiroaki Santo, Boxin Shi, Fumio Okura, and Yasuyuki Matsushita. Bilateral normal integration. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part I*, volume 13661 of *Lecture Notes in Computer Science*, pages 552–567. Springer, 2022. 2

[7] Xu Cao, Boxin Shi, Fumio Okura, and Yasuyuki Matsushita. Normal integration via inverse plane fitting with minimum point-to-plane distance. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2382–2391. Computer Vision Foundation / IEEE, 2021. 2

[8] 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, 7, 13

[9] Zhaoxi Chen and Ziwei Liu. Relighting4d: Neural relightable human from videos. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV*, volume 13674 of *Lecture Notes in Computer Science*, pages 606–623. Springer, 2022. 2

[10] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. GARF: gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *CoRR*, abs/2204.05735, 2022. 2

[11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 6

[12] Michael Daum and Gregory Dudek. On 3-d surface reconstruction using shape from shadows. In *1998 Conference on Computer Vision and Pattern Recognition (CVPR '98), June 23-25, 1998, Santa Barbara, CA, USA*, pages 461–468. IEEE Computer Society, 1998. 2

[13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3789–3799. PMLR, 2020. 5

[14] Michael Hatzitheodorou and John R. Kender. An optimal algorithm for the derivation of shape from shadows. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1988, 5-9 June, 1988, Ann Arbor, Michigan, USA*, pages 486–491. IEEE, 1988. 2

[15] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5865–5874. IEEE, 2021. 2

[16] Asaf Karnieli, Ohad Fried, and Yacov Hel-Or. Deepshadow: Neural shape from shadow. *CoRR*, abs/2203.15065, 2022. 2, 4, 6, 12

[17] John R. Kender and Earl Smith. Shape from darkness: Deriving surface information from dynamic shadows. In Tom Kehler, editor, *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 1: Science*, pages 664–669. Morgan Kaufmann, 1986. 2

[18] 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. 11

[19] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.*, 39(6):194:1–194:14, 2020. 2, 5

[20] Michael S. Langer, Gregory Dudek, and Steven W. Zucker. Space occupancy using multiple shadowimages. In *Pro-*

ceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1995, August 5 - 9, 1995, Pittsburgh, PA, USA, pages 285–290. IEEE Computer Society, 1995. 2

[21] 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, 7, 13

[22] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph.*, 37(6):222, 2018. 2, 5

[23] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: bundle-adjusting neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5721–5731. IEEE, 2021. 2

[24] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4460–4470. Computer Vision Foundation / IEEE, 2019. 1, 2

[25] 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, 2, 3, 6, 11

[26] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3501–3512. Computer Vision Foundation / IEEE, 2020. 1, 2

[27] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5569–5579. IEEE, 2021. 2

[28] Matthew O'Toole, David B. Lindell, and Gordon Wetzstein. Confocal non-line-of-sight imaging based on the light-cone transform. *Nat.*, 555(7696):338–341, 2018. 2

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 165–174. Computer Vision Foundation / IEEE, 2019. 1, 2

[30] Daniel Raviv, Yoh-Han Pao, and Kenneth A. Loparo. Reconstruction of three-dimensional surfaces from two-dimensional binary images. *IEEE Trans. Robotics Autom.*, 5(5):701–710, 1989. 2

[31] Viktor Rudnev, Mohamed Elgharib, William A. P. Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Neural radiance fields for outdoor scene relighting. *CoRR*, abs/2112.05140, 2021. 2

[32] Silvio Savarese, Marco Andreetto, Holly E. Rushmeier, Fausto Bernardini, and Pietro Perona. 3d reconstruction by shadow carving: Theory and practical evaluation. *Int. J. Comput. Vis.*, 71(3):305–336, 2007. 2

[33] Silvio Savarese, Holly E. Rushmeier, Fausto Bernardini, and Pietro Perona. Shadow carving. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 1*, pages 190–197. IEEE Computer Society, 2001. 2

[34] Siyuan Shen, Zi Wang, Ping Liu, Zhengqing Pan, Ruiqian Li, Tian Gao, Shiying Li, and Jingyi Yu. Non-line-of-sight imaging via neural transient fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(7):2257–2268, 2021. 2

[35] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7495–7504. Computer Vision Foundation / IEEE, 2021. 2

[36] Shlomi Steinberg and Ling-Qi Yan. A generic framework for physical light transport. *ACM Trans. Graph.*, 40(4):139:1–139:20, 2021. 1

[37] Kushagra Tiwary, Tzofi Klinghoffer, and Ramesh Raskar. Towards learning neural representations from shadows. *CoRR*, abs/2203.15946, 2022. 3

[38] 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. 2, 4, 5, 11, 12

[39] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *CoRR*, abs/2102.07064, 2021. 2

[40] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Comput. Graph. Forum*, 41(2):641–676, 2022. 1, 2

[41] Shumian Xin, Sotiris Nousias, Kiriakos N. Kutulakos, Aswin C. Sankaranarayanan, Srinivasa G. Narasimhan, and Ioannis Gkioulekas. A theory of fermat paths for non-line-of-sight shape reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6800–6809. Computer Vision Foundation / IEEE, 2019. 2

[42] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance

fields on complex scenes from a single image. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII*, volume 13682 of *Lecture Notes in Computer Science*, pages 736–753. Springer, 2022. 2

[43] Shuntaro Yamazaki, Srinivasa G. Narasimhan, Simon Baker, and Takeo Kanade. The theory and practice of coplanar shadowgram imaging for acquiring visual hulls of intricate objects. *Int. J. Comput. Vis.*, 81(3):259–280, 2009. 2

[44] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part I*, volume 13661 of *Lecture Notes in Computer Science*, pages 266–284. Springer, 2022. 2, 5

[45] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K. Wong. S^3-nerf: Neural reflectance field from shading and shadow under a single viewpoint. *CoRR*, abs/2210.08936, 2022. 3

[46] Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf: Neural incident light field for physically-based material estimation. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXI*, volume 13691 of *Lecture Notes in Computer Science*, pages 700–716. Springer, 2022. 2

[47] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. 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 4805–4815, 2021. 2

[48] 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. 1, 2, 4

[49] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4578–4587. Computer Vision Foundation / IEEE, 2021. 2

[50] Yizhou Yu and Johnny T. Chang. Shadow graphs and 3d texture reconstruction. *Int. J. Comput. Vis.*, 62(1-2):35–60, 2005. 2

[51] 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. 2, 5, 6, 12

[52] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5453–5462. Computer Vision Foundation / IEEE, 2021. 2

[53] 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. 2

[54] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18622–18631. IEEE, 2022. 2

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 [25]) and our method. We list each method's components in Tab. 3 to better illustrate their correspondences.

B. Additional implementation details

Network architecture. We adopt an 8-layer geometry MLP following [38]. 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 [18], 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 0.1.

	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 3. Corresponding components in camera and shadow ray supervision.

Shadow ray sampling. We place 80 uniform samples along the shadow ray and use the hierarchical sampling strategy in [38] 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 [51] 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 subpixel regions using ray marching and compute the shadow rays started at each region respectively, as shown in Fig. 11. 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 [51] 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.

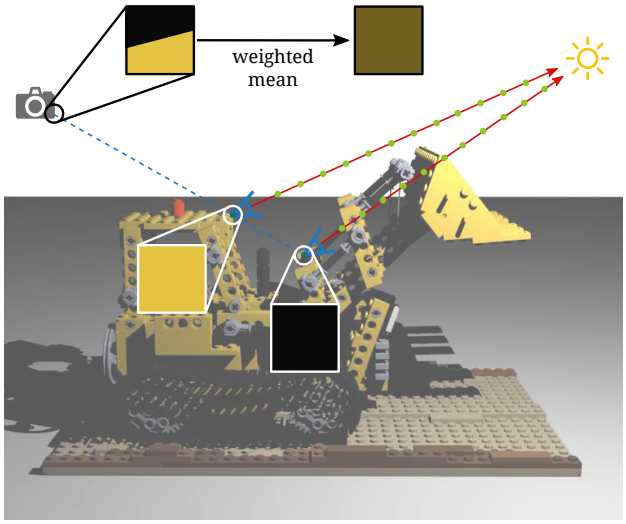


Figure 11. At a boundary pixel, we compute two shadow rays started at different depths and combine their results by weighted mean.

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 [16] and a non-terrain dataset proposed by us. The results on the DeepShadow dataset are shown in Tab. 4, and the results on our dataset are shown in Tab. 5, 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

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 4. 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 5. Quantitative comparison of reconstruction quality on our binary shadow dataset.

depth error in Tab. 6. We report DeepShadow's results from their publicly available code, which are slightly better than their paper results.

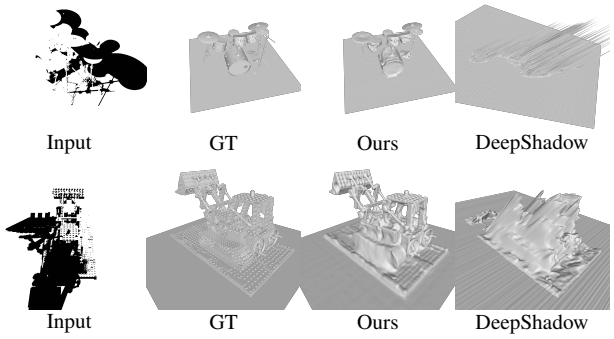


Figure 12. Qualitative comparison on our side-view binary shadow dataset.

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. 12.

C.3. Quantitative comparison on RGB inputs

We show the quantitative results of SDPS-Net [8], Li et al. [21] and our method on our RGB dataset in Tab. 7. 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 0. As shown in Fig. 13, 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. 15, 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.

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. 16 and Tab. 8, our method can reconstruct

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 6. 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 7. Quantitative comparison of reconstruction quality on our RGB dataset.

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 8. Reconstruction quality using different numbers of input images.

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

Table 9. Reconstruction quality when supervising only background or foreground shadows.

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.

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. 17, 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. 9, our method achieves the lowest reconstruction error when supervising foreground and background shadows. The two parts of shadows are indispensable in accurate shape reconstruction.

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. 14, 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 applications, such as relighting using a point light or an environment map and material editing. In Fig. 18, we show that our method generates plausible results in these applications. Please also see the supplementary video for more results.

G. Synthetic dataset examples

In Fig. 19, we show different data types from our synthetic dataset.

H. Real dataset examples

In Fig. 20, we show the objects, capture setup, and example images from our real dataset.

I. 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 recon-

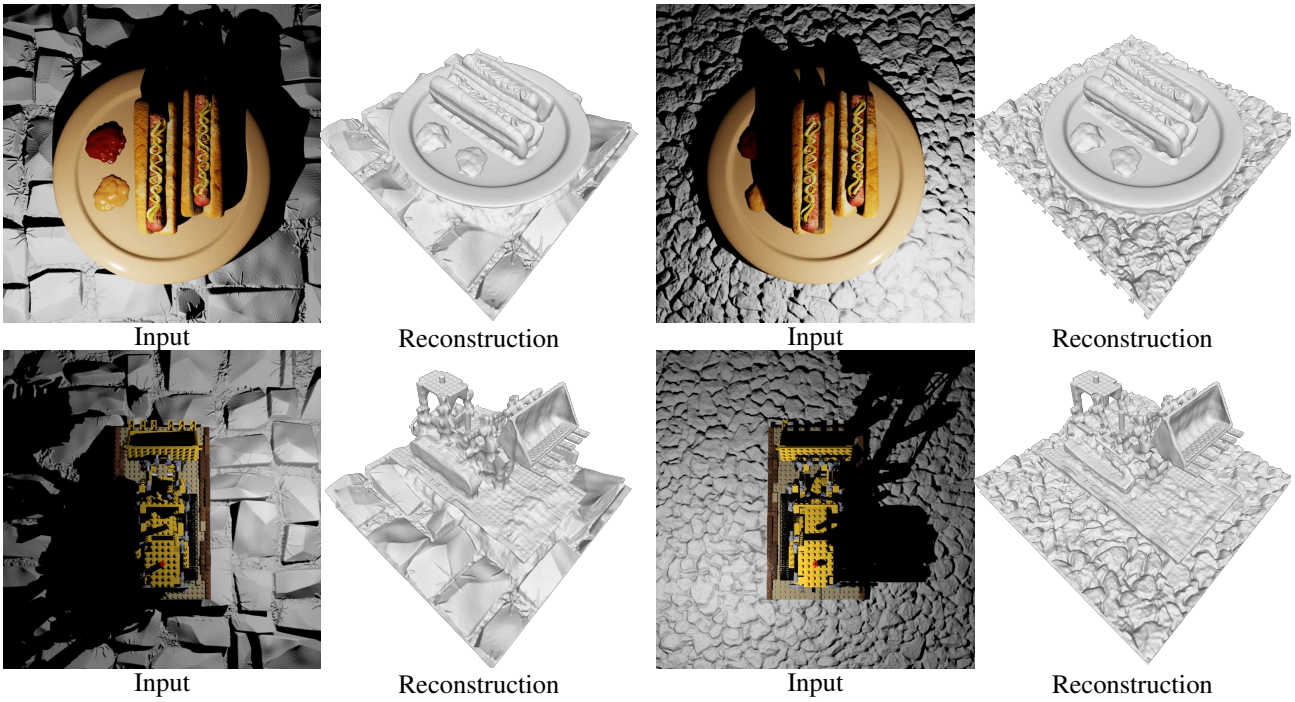


Figure 13. Results in the presence of bumpy grounds.

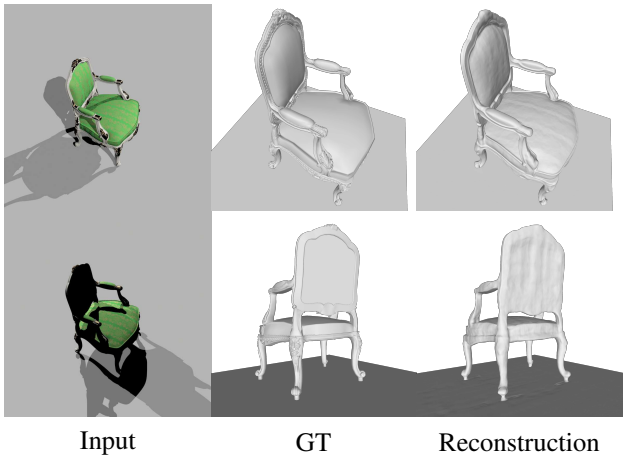


Figure 14. Results on the scene illuminated by two lights.

structed by exploiting shadows on the visible surface, revealing scenes beyond the camera's line of sight.

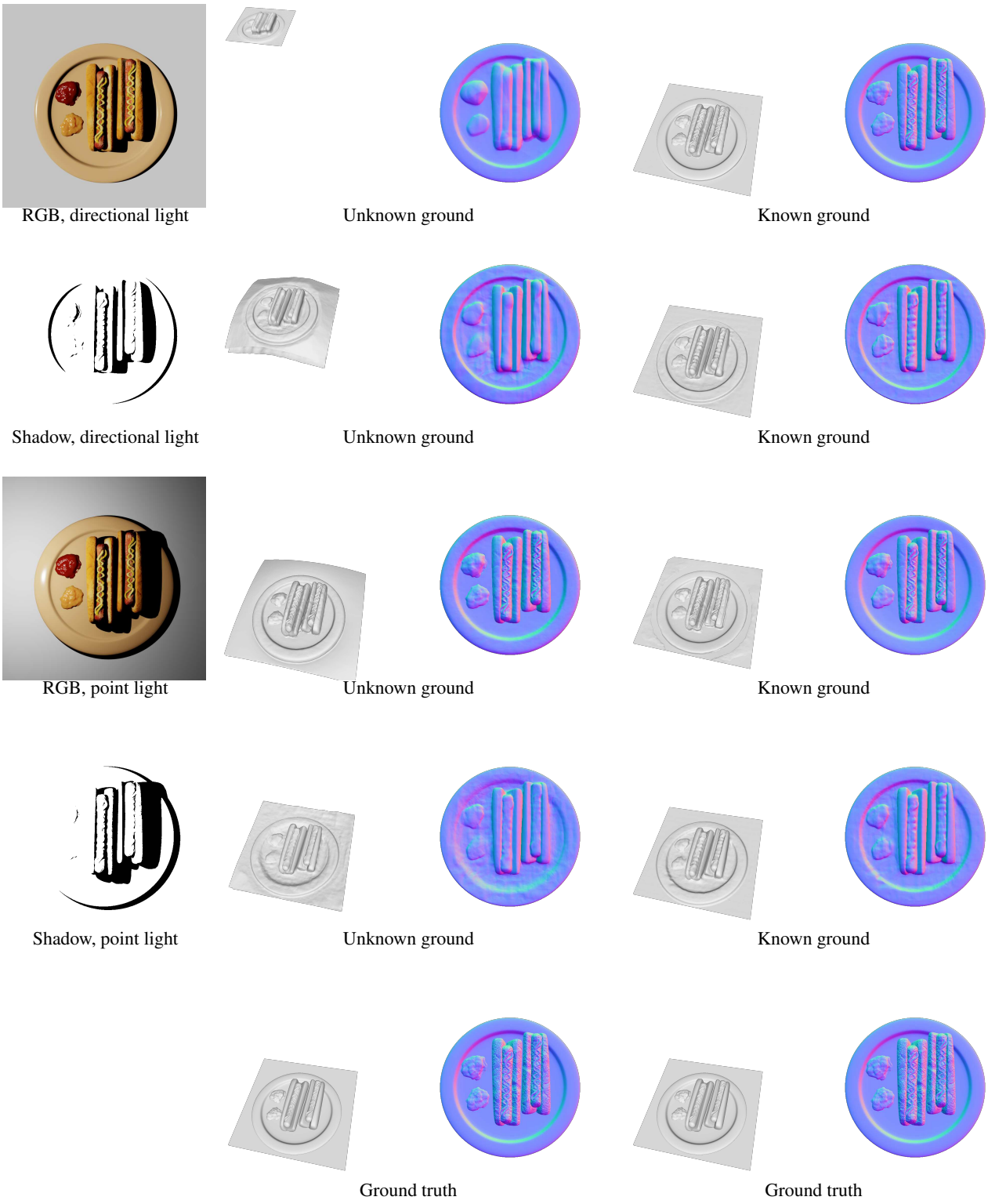


Figure 15. Comparison between known and unknown grounds.

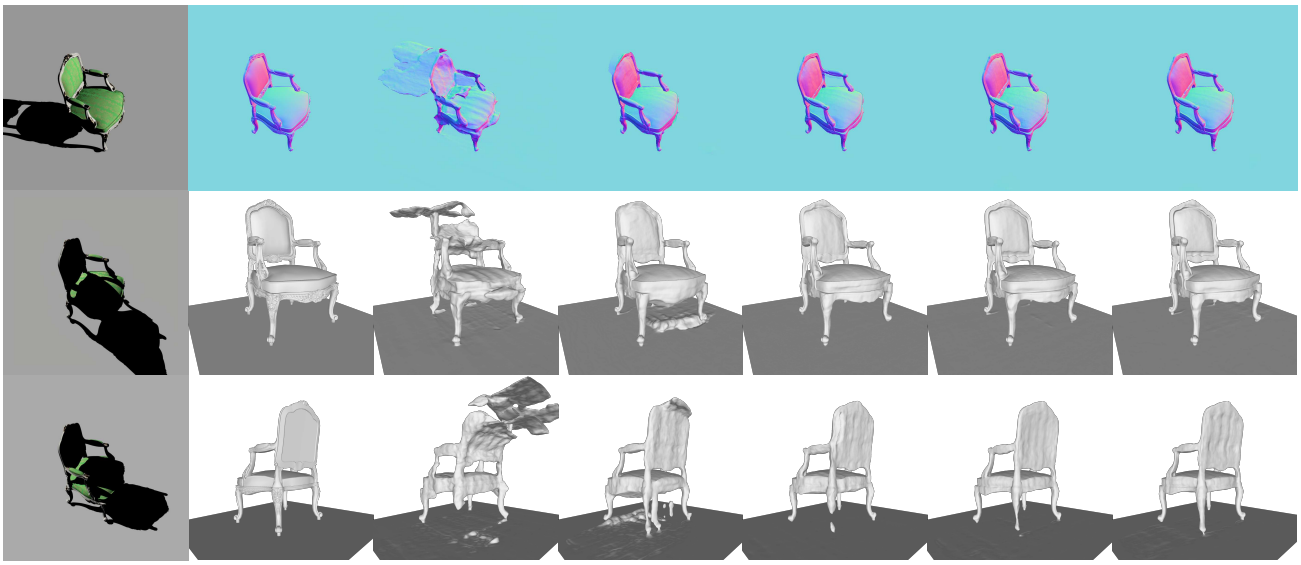


Figure 16. Analysis on different numbers of input images.

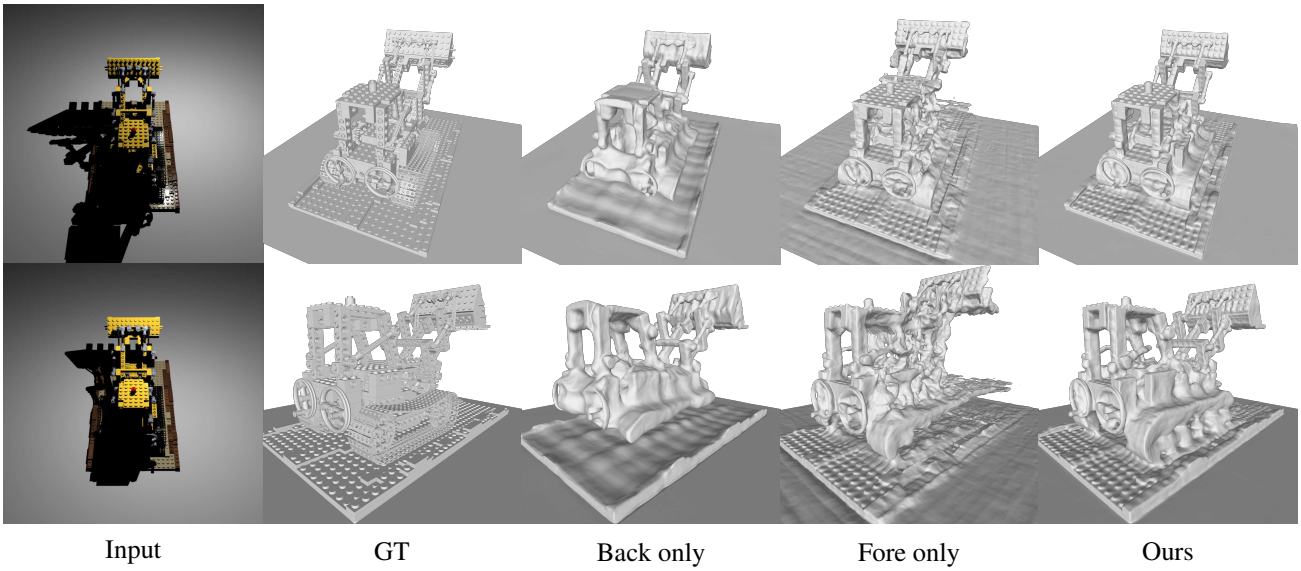


Figure 17. Comparison of shape reconstruction when supervising only background or foreground shadows.

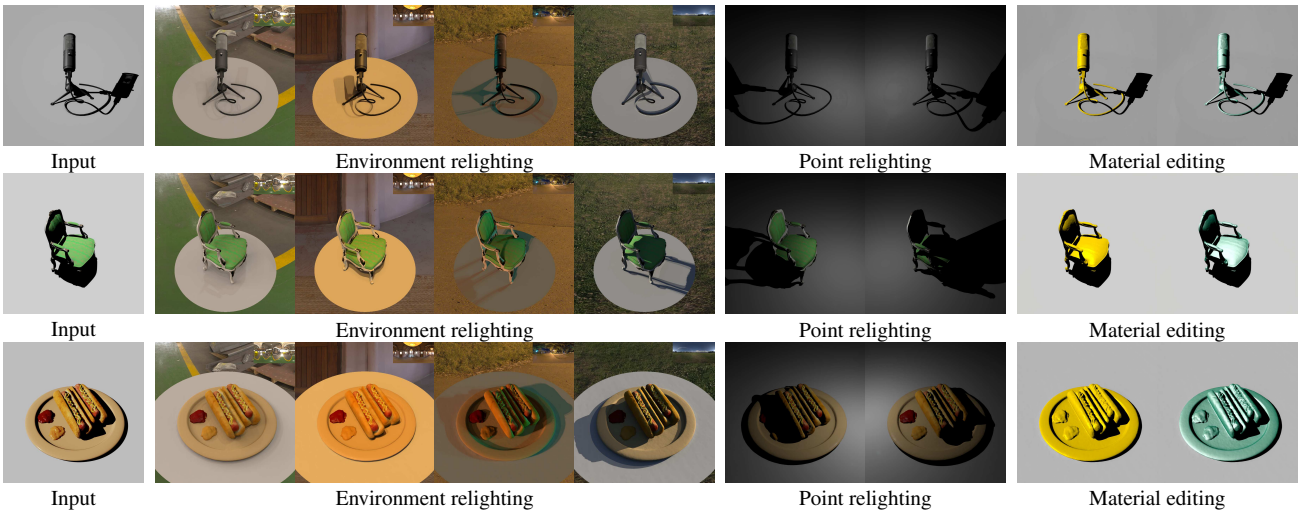


Figure 18. Applications.

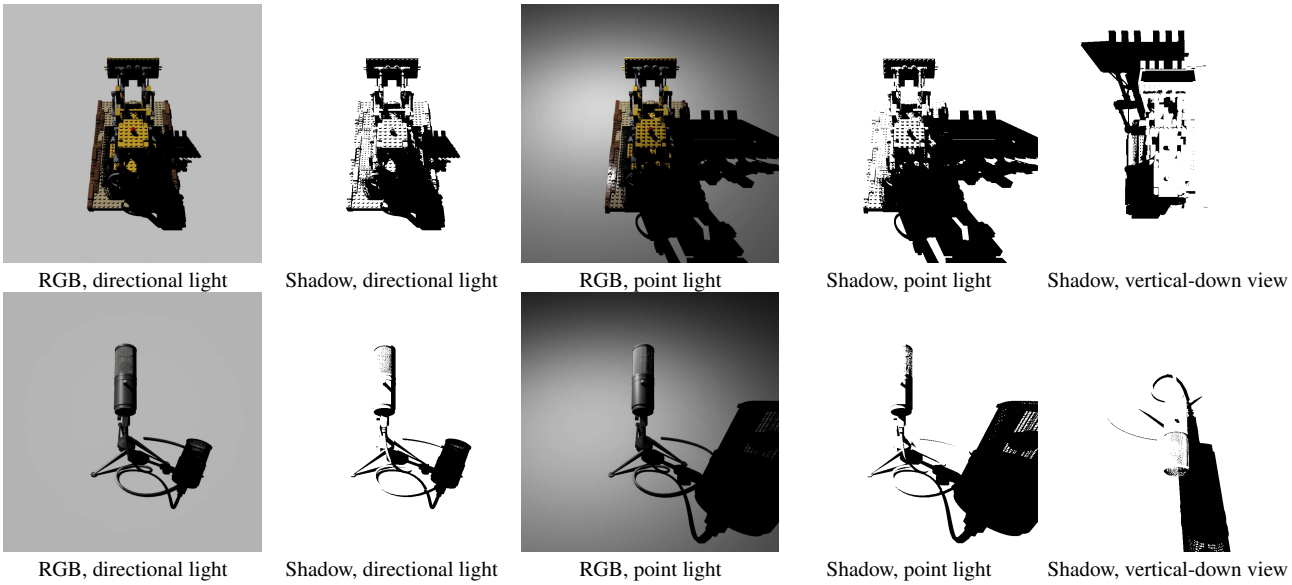
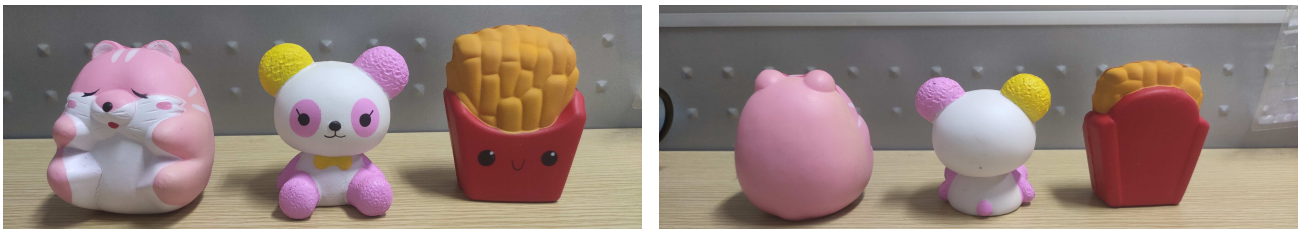


Figure 19. Example data from our synthetic dataset.



Captured objects

Capture setup

Example input images

Figure 20. More details of our real dataset.