ShadowNeuS: Neural SDF Reconstruction by Shadow Ray Supervision

Jingwang Ling\textsuperscript{1} Zhibo Wang\textsuperscript{2} Feng Xu\textsuperscript{1*}

\textsuperscript{1}School of Software and BNRist, Tsinghua University \hspace{1cm} \textsuperscript{2}SenseTime Research

\textbf{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 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 are available at \url{https://github.com/gerwang/ShadowNeuS}.

\section{1. Introduction}

Neural field \cite{hodan2022neural} 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 \cite{sun2013ssd, bao2019fast} and point cloud-based \cite{chen2020dense, li2021pvn} 3D reconstruction. So more and more works are using neural fields as the 3D scene representation for various related tasks.

Among these works, NeRF \cite{mildenhall2020nerf} is a representative method that incorporates a part of physically based light transport \cite{wright1979computer} 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 \textit{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.

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 \textit{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 [2, 20, 23, 40, 54]. 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 [18] 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 [43] 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 [26, 31] to reconstruct an implicit representation of 3D shapes. It is also possible to optimize a neural field from multi-view images by differentiable rendering [2, 28, 51]. NeRF [27] 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, [29, 41, 50] propose to combine NeRF with surface representation to reconstruct high-quality and well-defined surfaces. While all the above works require known camera viewpoints, [12, 24, 42] 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. [3, 4, 54] require co-located camera-light setup to avoid shadows in captured images. [5, 6, 55] assume smooth environment lights and ignore shadows. [11, 33, 37, 47, 49, 57] adapt neural networks conditioned on the light direction to model light-dependent shadows. Among them, [11, 47, 49, 56, 57] 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. [17, 45, 52] 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 [30, 36, 44] adopts a transient sensor to capture time-resolve signals, which enables reconstructing the scene beyond the camera’s view frustum. Photometric stereo [9, 22] reconstructs surface normals from images captured under directional lights. Normals can be integrated to produce a depth map but require non-trivial processing [7, 8].

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 [16, 19], 2D height maps [14, 32, 35, 53] and 3D voxel grids [21, 34, 46]. 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 [53] and intricate thin structures [46]. The most recent work in this area is DeepShadow [18], which reconstructs a neural depth map from shadows. A different setup with fixed lighting
but multiple viewpoints is also adopted by [39], which integrates Shadow Mapping to reconstruct a neural representation. Concurrently and independently, [48] 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 [27] training as supervising camera rays. From there, we discover a ray supervision scheme generalizable to arbitrary rays. The scheme makes it feasible for shadow rays to supervise the optimization of 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 $t_1, \ldots, t_N$, started from the camera center $o$ along the camera ray direction $v$. With the scene density $\sigma_i$ and emitted radiance $c_i$ at each sample point $p(t_i) = o + t_i v$, the estimated radiance $C$ at the camera can be formulated as follows,

$$C(o, v) = \sum_{i=1}^{N} T_i \sigma_i c_i,$$  \hspace{2cm} (1)

where $\alpha_i = 1 - \exp( - \sigma_i (t_{i+1} - t_i) )$ is the discrete opacity and $T_i = \exp(-\sum_{j=1}^{i} \sigma_j \cdot (t_{j+1} - t_i))$ indicates the light transmittance, i.e., the proportion of the emitted light reach the camera from the point $p(t_i)$. 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 can 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

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

where and $L$ is the intensity of the light source, $x$ is the location of a ray sensor and $l$ is the light direction. To obtain 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 $l$. 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 [18] 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) $S = \{ u \in \mathbb{R}^3 | f(u) = 0 \}$, 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 [51] is used to compute the intersection point $x$ at the current SDF. Then we can compute the incoming radiance $C_m(x, l)$ at the intersection by volume rendering. As we are modeling an SDF instead of a density field, we replace the discrete opacity $\alpha_i$ in Eq. (2) by the one derived from the SDF following NeuS [41], as

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

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 $x$ given the SDF, ray marching is the most straightforward choice. However, as it is non-differentiable, it is prone to be misled by surface points with incorrect depths, leading to worse results. To optimize the intersection points using backpropagated gradients, we use implicit differentiation [1, 51], which makes the intersection point differentiable to the SDF network parameters as

$$\hat{x} = x - \frac{v}{n \cdot v} f(x), \quad (4)$$

where $v$ is the camera ray direction and $n = \nabla_x f(x)$ is
the surface normal derived from the SDF network. Then, we use \( C_{\text{in}}(\hat{x}, l) \) as the differentiable radiance at intersection \( \hat{x} \). As \( x \) acts as the start position of a shadow ray, it can be optimized by gradients from Eq. (2). When the computed incoming radiance \( C_{\text{in}}(\hat{x}, l) \) 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 \( \hat{x} \) in Eq. (4) only differentiates along the camera direction \( v \). When supervising \( C_{\text{in}}(\hat{x}, l) \) 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 \( \hat{x} \) moves perpendicular to the camera direction \( v \), 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 [20,23].

Therefore, we first obtain a pixel subset \( \Omega \) corresponding to surface boundaries, and a differentiable ratio \( w \) for each boundary pixel using the surface walk procedure in [54]. Then we locate two intersections \( x_n \) and \( x_1 \) at different depths within the pixel and compute their incoming radiance \( C_{\text{in}}(\hat{x}_n, l) \) and \( C_{\text{in}}(\hat{x}_1, l) \) respectively. When computing the incoming radiance corresponding to pixel \( p \), we average the incoming radiance at boundary pixels as

\[
\hat{C}_{\text{in}} = \begin{cases} 
C_{\text{in}}(\hat{x}, l) & p \notin \Omega \\
wC_{\text{in}}(\hat{x}_n, l) + (1 - w)C_{\text{in}}(\hat{x}_1, l) & p \in \Omega
\end{cases}
\]  

(5)

Then, we can supervise the computed incoming radiance \( \hat{C}_{\text{in}} \) with a pixel \( I_o \) on a binary shadow image as

\[
\mathcal{L}_{\text{shadow}} = \|\hat{C}_{\text{in}} - I_o\|_1.
\]

(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 \( \rho_d \) and a specular component \( \rho_s \). Following [22,47], we use a weighted combination of spherical Gaussian basis to represent the specular component \( \rho_s \) as \( \rho_s = y^T D(h, n) \), where \( h = \frac{l - v}{\|l - v\|} \) is the half-vector between light direction \( l \) and view direction \( -v \), \( D \) is the specular basis and \( y \) is the specular coefficients. We model another MLP network \( g \) to regress material properties \( (\rho_d, y) = g(x) \) at surface location \( x \).

The outgoing radiance at point \( x \) can be formulated as

\[
C(x, -v) = (\rho_d + \rho_s) C_{\text{in}}(x, l)(l \cdot n)
\]

(7)

The outgoing radiance \( \hat{C} \) 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 \( I_o \) on an RGB image as

\[
\mathcal{L}_{\text{rgb}} = \|\hat{C} - I_o\|_1
\]

(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 \( l \) and intensity \( L \) are known and uniform for all shadow rays. For point light, we calculate the light direction and intensity at point \( x \) as

\[
L = \frac{L_p}{\|q - x\|^2}, \quad l = \frac{q - x}{\|q - x\|}
\]

(9)

where \( L_p \) is a scalar point light intensity and \( q \) is the light location.

**Training.** To regularize the network to output valid SDF, we add an Eikonal loss [15] on \( M \) sample points as

\[
\mathcal{L}_{\text{eik}} = \frac{1}{M} \sum_i (\|\nabla f(p_i)\|_2 - 1)^2.
\]

(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 [41] 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 using a surface walk process [54] 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 [18], 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 [27]. 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 [13]. 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. We select lights randomly sampled on the upper hemisphere, similar to the camera position selection in NeRF. 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 [4]. 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.

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 [18], 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, qualitative and quantitative results show that our method achieves better shape reconstruction on both datasets. As shown at the top row of Fig. 4, our method achieves visually comparable results with DeepShadow on reconstructing a terrain-like geometry. For more complex inputs, our method reconstructs more detailed and complete structures than DeepShadow, as shown at the bottom row 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 re-
results show that our method brings significant improvement in reconstructing complex scenes. Please see Tab. 2 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 [9, 22] which also consider shadows. SDPS-Net [9] is a deep-learning method that augments the training dataset with images under shadows, and Li et al. [22] 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 Tab. 3, we achieve the lowest depth and normal reconstruction errors, illustrating the effectiveness 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. 3 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 comparing with our complete method on our synthetic directional RGB inputs.
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 complete 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. 9. 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 test our method on more challenging synthetic data. As shown in Fig. 8, 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 inter-reflections 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 thin structure neural SDF, as indicated by very recent works [10, 25].

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. Experiments show that our technique outperforms the SOTAs in single-view reconstruction, and it has the power to reconstruct scene geometries out of the camera’s line of sight.

Acknowledgements This work was supported by the National Key R&D Program of China (2018YFA0704000), Beijing Natural Science Foundation (M22024), the NSFC (No.62021002), and the Key Research and Development Project of Tibet Autonomous Region (XZ202101ZY0019G). This work was also supported by THUIBCS, Tsinghua University, and BLBCI, Beijing Municipal Education Commission.
References


