NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction

Peng Wang[†], **Lingjie Liu**[‡], **Yuan Liu**[†], **Christian Theobalt**[‡], **Taku Komura**[†], **Wenping Wang**[◇]
[†]The University of Hong Kong [‡]Max Planck Institute for Informatics
[◇]Texas A&M University
[†]{pwang3,yliu,taku}@cs.hku.hk [‡]{lliu,theobalt}@mpi-inf.mpg.de
[◇]wenping@tamu.edu

Abstract

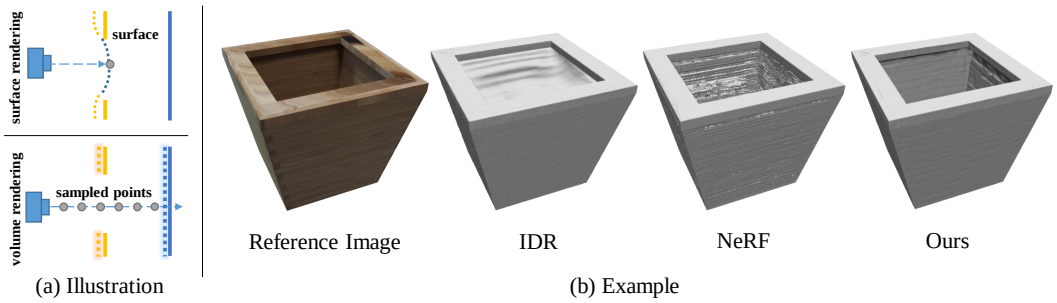
We present a novel neural surface reconstruction method, called *NeuS*, for reconstructing objects and scenes with high fidelity from 2D image inputs. Existing neural surface reconstruction approaches, such as DVR [Niemeyer et al, 2020] and IDR [Yariv et al., 2020], require foreground mask as supervision, easily get trapped in local minima, and therefore struggle with the reconstruction of objects with severe self-occlusion or thin structures. Meanwhile, recent neural methods for novel view synthesis, such as NeRF [Mildenhall et al., 2020] and its variants, use volume rendering to produce a neural scene representation with robustness of optimization, even for highly complex objects. However, extracting high-quality surfaces from this learned implicit representation is difficult because there are not sufficient surface constraints in the representation. In NeuS, we propose to represent a surface as the zero-level set of a *signed distance function* (SDF) and develop a new volume rendering method to train a neural SDF representation. We observe that the conventional volume rendering method causes inherent geometric errors (i.e. bias) for surface reconstruction, and therefore propose a new formulation that is free of bias in the first order of approximation, thus leading to more accurate surface reconstruction even without the mask supervision. Experiments on the DTU dataset and the BlendedMVS dataset show that NeuS outperforms the state-of-the-arts in high-quality surface reconstruction, especially for objects and scenes with complex structures and self-occlusion.

1 Introduction

Reconstructing surfaces from multi-view images is a fundamental problem in computer vision and computer graphics. 3D reconstruction with neural implicit representations has recently become a highly promising alternative to classical reconstruction approaches [31, 6, 2] due to its high reconstruction quality and its potential to reconstruct complex objects that are difficult for classical approaches, such as non-Lambertian surfaces and thin structures. Recent works represent surfaces as signed distance functions (SDF) [39, 42, 14, 18] or occupancy [24, 25]. To train their neural models, these methods use a differentiable surface rendering method to render a 3D object into images and use the rendered images to compare against input images for supervision. For example, IDR [39] produces impressive reconstruction results, but it fails to reconstruct objects with complex structures that causes abrupt depth changes. The cause of this limitation is that the surface rendering method used in IDR only considers a single surface intersection point for each ray. Consequently, the gradient only exists at this single point, which is too local for effective back propagation and would get optimization stuck in a poor local minimum when there are abrupt changes of depth on images. Furthermore, object masks are needed as supervision for converging to a valid surface. As illustrated in Fig. 1 (a) top, with the radical depth change caused by the hole, the neural network

would incorrectly predict the points near the front surface to be blue, failing to find the far-back blue surface. The actual test example in Fig. 1 (b) shows that IDR fails to correctly reconstruct the surfaces near the edges with abrupt depth changes.

Recently, NeRF [23] and its variants have explored to use a volume rendering method to learn a volumetric radiance field for novel view synthesis. This volume rendering approach samples multiple points along each ray and perform α-composition of the colors of the sampled points to produce the output pixel colors for training purposes. The advantage of the volume rendering approach is that it can handle abrupt depth changes, because it considers multiple points along the ray and so all the sample points, either near the surface or on the far surface, produce gradient signals for back propagation. For example, referring Fig. 1 (a) bottom, when the near surface (yellow) is found to have inconsistent colors with the input image, the volume rendering approach is capable of training the network to find the far-back surface to produce the correct scene representation. However, since it is intended for novel view synthesis rather than surface reconstruction, NeRF uses volume rendering to learn only a volume density field from which it is difficult to extract a high-quality surface. Fig. 1 (b) shows a surface extracted as a level-set surface of the density field computed by NeRF. Although the surface correctly accounts for abrupt depth changes, it contains conspicuous noise in some planar regions.



(a) Illustration (b) Example

Figure 1: (a) Illustration of the surface rendering and volume rendering. (b) A toy example of bamboo planter, where there are occlusions on the top of the planter. Compared to the state-of-the-art methods, our approach can handle the occlusions and achieve better reconstruction quality.

In this work, we present a new neural rendering scheme, called *NeuS*, for multi-view surface reconstruction. NeuS uses the *signed distance function* (SDF) for surface representation and uses a novel volume rendering scheme to learn a neural SDF representation. Specifically, by introducing a density distribution induced by SDF, we make it possible to apply the volume rendering approach to learning an implicit SDF representation and thus have the best of both worlds, i.e. an accurate surface representation using a neural SDF model and robust network training in the presence of abrupt depth changes as enabled by volume rendering. Note that simply applying a standard volume rendering method to the density associated with SDF would lead to discernible bias (i.e. inherent geometric errors) in the reconstructed surfaces. This is a new and important observation that we will elaborate later. Therefore we propose a novel volume rendering scheme to ensure unbiased surface reconstruction in the first-order approximation of SDF. Experiments on both DTU dataset and BlendedMVS dataset demonstrated that NeuS is capable of reconstructing complex 3D objects and scenes with severe occlusions and delicate structures, even without foreground masks as supervision. It outperforms the state-of-the-art neural scene representation methods, namely IDR [39] and NeRF [23], in terms of reconstruction quality.

2 Related Works

Classical Multi-view Surface and Volumetric Reconstruction. Traditional multi-view 3D reconstruction methods can be roughly classified into two categories: point- and surface-based reconstruction [2, 6, 8, 31] and volumetric reconstruction [5, 3, 32]. Point- and surface-based reconstruction methods estimate the depth map of each pixel by exploiting inter-image photometric consistency [7] and then fuse the depth maps into a global dense point cloud [20, 41]. The surface reconstruction is usually done as a post processing with methods like screened Poisson surface reconstruction [13]. The reconstruction quality heavily relies on the quality of correspondence matching, and the difficulties in

matching correspondence for objects without rich textures often lead to severe artifacts and missing parts in the reconstruction results. Alternatively, volumetric reconstruction methods circumvent the difficulty of explicit correspondence matching by estimating occupancy and color in a voxel grid from multi-view images and evaluating the color consistency of each voxel. Due to limited achievable voxel resolution, these methods cannot achieve high accuracy.

Neural Implicit Representation. Neural implicit representation has recently become a promising alternative to conventional scene representations, e.g. point cloud, voxel grids, meshes, due to its continuous nature, which is free of the limitation of finite resolution. This representation has been applied successfully to shape representation [21, 22, 26, 4, 1, 9, 40, 27], novel view synthesis [33, 19, 12, 23, 17, 28, 29, 36] and multi-view 3D reconstruction [39, 24, 14, 11, 18].

Our work mainly focuses on learning implicit neural representation encoding both geometry and appearance in 3D space from 2D images via classical rendering techniques. Limited in this scope, the related works can be roughly categorized based on the rendering techniques used, i.e. surface rendering based methods and volume rendering based methods. Surface rendering based methods [24, 14, 39, 18] assume that the color of ray only relies on the color of an intersection of the ray with the scene geometry which makes the gradient only be backpropagated to a local region near the intersection. Therefore, such methods struggle with reconstructing complex objects with severe self-occlusions, sudden depth changes and thin parts. Furthermore, it usually requires object masks as supervision. On the contrary, our method performs well for such challenging cases without the need of masks.

Volume rendering based methods, such as NeRF[23], render an image by α-compositing colors of the sampled points along each ray. Since during training, the gradient can be back-propagated to every sample points, it can handle sudden depth changes and synthesize high-quality images. However, extracting high-fidelity surface from the learned implicit field is difficult because the density-based scene representation lacks sufficient constraints its level sets. In contrast, our method combines the advantages of surface rendering-based and volume rendering-based methods by constraining the scene space as a density field induced by a signed distance function and applying volume rendering to train this density-based representation with robustness. UNISURF [25], a concurrent unpublished work, also learns an implicit surface via volume rendering. It improves the reconstruction quality by shrinking the sample region of volume rendering during the optimization. Our method differs from UNISURF in that UNISURF represents the surface by occupancy values and gradually reduces the sample regions at some predefined steps to make the occupancy value converge to the surface while our method represents the scene by a signed distance function (SDF) and thus can naturally extract the surface as the zero-level set of the SDF, yielding better reconstruction accuracy than UNISURF, as will be seen later in the experiment section.

3 Method

Given a set of posed images $\{\mathcal{I}_k\}$ of a 3D object, our goal is to reconstruct the surface $\mathcal{S}$ of the object. Note that in the paper we only focus on solid objects and scenes. The surface is represented by the zero-level set of an implicit signed distance function (SDF) encoded by a fully connected neural network (MLP). In order to learn the weights of this network, we developed a novel volume rendering method to render images from the implicit SDF and minimize the difference between the rendered images and the input images. This volume rendering approach ensures robust optimization in NeuS for reconstructing objects of complex structures.

3.1 Rendering Procedure

Scene representation. With NeuS, the 3D scene of an object to be reconstructed is represented by two functions: $f : \mathbb{R}^3 \to \mathbb{R}$ that maps a point $\mathbf{x} \in \mathbb{R}^3$ to its signed distance to the object, and $c : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3$ that encodes the color associated with a point $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{v} \in \mathbb{S}^2$. Both functions are encoded by neural networks of Multi-layer Perceptron (MLP). The surface $\mathcal{S}$ of the object is represented by the zero-set of its SDF, that is,

$$\mathcal{S} = \left\{ \mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0 \right\}. \tag{1}$$

In order to apply a volume rendering method to training the SDF network, we first introduce a probability density function $\phi_s(f(\mathbf{x}))$, called *S-density*, where $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^3$, is the signed distance

function and $\phi_s(x) = se^{-sx}/(1+e^{-sx})^2$, commonly known as the *logistic density distribution*, is the derivative of the Sigmoid function $\Phi_s(x) = (1+e^{-sx})^{-1}$, i.e. $\phi_s(x) = \Phi_s'(x)$. In principle $\phi_s(x)$ can be any unimodal (i.e. bell-shaped) density distribution centered at 0; here we choose the logistic density distribution for its computational convenience. Note that the standard deviation of $\phi_s(x)$ is given by $1/s$, which is also a trainable parameter, that is, $1/s$ approaches to zero as the network training converges. Note that in our method the opacity value α used for volume rendering depends on the S-density in a new way that differs from the conventional formulation of volume rendering, which directly uses the given density function as opacity.

Intuitively, the main idea of NeuS is that, with the aid of the S-density field $\phi_s(f(\mathbf{x}))$, volume rendering is used to train the SDF network with only 2D input images as supervision. Upon successful minimization of a loss function based on this supervision, the zero-level set of the network-encoded SDF is expected to represent an accurately reconstructed surface $\mathcal{S}$, with its induced S-density $\phi_s(f(\mathbf{x}))$ assuming prominently high values near the surface.

Rendering. To learn the parameters of the MLPs of the SDF and the color field, we devise a volume rendering scheme to render images from the proposed SDF representation and compare the rendered images with the input images for network supervision. Given a pixel, we denote the ray emitted from this pixel as $\{\mathbf{p}(t) = \mathbf{o} + t\mathbf{v} | t \geq 0\}$, where $\mathbf{o}$ is the center of the camera and $\mathbf{v}$ is the unit direction vector of the ray. We accumulate the colors along the ray by

$$C(\mathbf{o}, \mathbf{v}) = \int_0^{+\infty} w(t) c(\mathbf{p}(t), \mathbf{v}) \mathrm{d}t, \tag{2}$$

where $C(\mathbf{o}, \mathbf{v})$ is the output color for this pixel, $w(t)$ a weight for the point $\mathbf{p}(t)$, and $c(\mathbf{p}(t), \mathbf{v})$ the color at the point $\mathbf{p}$ along the viewing direction $\mathbf{v}$. As a weight function, $w(t)$ is required to satisfy that $w(t) \geq 0$ and $\int_0^{+\infty} w(t)\mathrm{d}t = 1$.

Requirements on weight function. The key to learning an accurate SDF representation from 2D images is to build an appropriate connection between output colors and SDF, i.e., to derive an appropriate weight function $w(t)$ on the ray based on the SDF f of the scene. In the following, we list the requirements on the weight function $w(t)$.

1. **Unbiased**. Given a camera ray $\mathbf{p}(t)$, $w(t)$ attains a locally maximal value at a surface intersection point $\mathbf{p}(t^*)$, i.e. with $f(\mathbf{p}(t^*)) = 0$, that is, the point $\mathbf{p}(t^*)$ is on the zero-level set of the SDF $(\mathbf{x})$.
2. **Occlusion-aware**. Given any two depth values t_0 and t_1 satisfying $f(t_0) = f(t_1)$, $w(t_0) > 0$, $w(t_1) > 0$, and $t_0 < t_1$, there is $w(t_0) > w(t_1)$. That is, when two points have the same SDF value (thus the same SDF-induced S-density value), the point nearer to the view point should have a larger contribution to the final output color than does the other point.

An unbiased weight function $w(t)$ guarantees that the intersection of the camera ray with the zero-level set of SDF contributes most to the pixel color. The occlusion-aware property ensures that when a ray sequentially passes multiple surfaces, the rendering procedure will correctly use the color of the surface nearest to the camera to compute the output color.

Next, we will first introduce a naive way of defining the weight function $w(t)$ and explain why it is not appropriate for reconstruction before introducing our novel construction of $w(t)$. In fact, we will show that directly using the standard pipeline of volume rendering would produce an undesirable bias in surface reconstruction.

Naive solution. To make the weight function occlusion-aware, a natural solution is based on the standard volume rendering formulation [23] which defines the weight function by

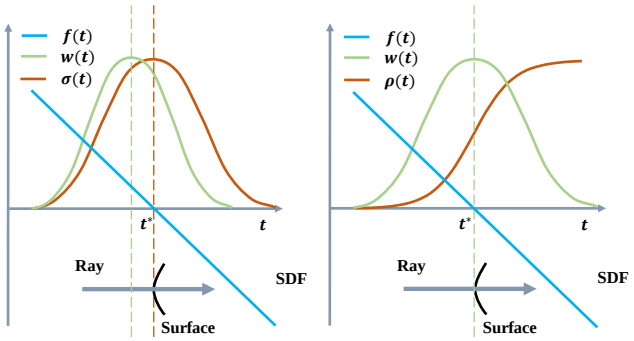
$$w(t) = T(t)\sigma(t), \tag{3}$$

where $\sigma(t)$ is the so-called the *volume density* in classical volume rendering and $T(t) = \exp(-\int_0^t \sigma(u)du)$ here denotes the *accumulated transmittance* along the ray under consideration. To adopt the standard volume density formulation [23], here $\sigma(t)$ is set to be equal to the S-density value, i.e. $\sigma(t) = \phi_s(f(\mathbf{p}(t)))$ and the weight function $w(t)$ is computed by Eqn. 3. Although the resulting weight function is occlusion-aware, it is biased as it introduces inherent errors in the reconstructed surfaces. As illustrated in Fig. 2 (a), the weight function $w(t)$ attains a local maximum at a point before the ray reaches the surface point $\mathbf{p}(t^*)$, satisfying $f(\mathbf{p}(t^*)) = 0$. This fact will be proved in the Appendix.

Our solution. To introduce our solution, we first introduce a straightforward way to construct an unbiased weight function, which directly uses the normalized S-density as weights

$$w(t) = \frac{\phi_s(f(\mathbf{p}(t)))}{\int_0^{+\infty} \phi_s(f(\mathbf{p}(u)))\mathrm{d}u}. \quad (4)$$

This construction of weight function is obviously unbiased, but not occlusion-aware. For example, if the ray penetrates two surfaces, the SDF function f will have two zero points on the ray, which leads to two peaks on the weight function $w(t)$ and the resulting weight function will mix and average the colors of two surfaces without considering occlusions due to their order of depth.



(a) Naive Solution (b) Our Solution

Figure 2: Illustration of (a) weight bias of naive solution, and (b) the weight function defined in our solution, which is unbiased in the first-order approximation of SDF.

To this end, now we shall design the weight function $w(t)$ that is both occlusion-aware and unbiased in the first order approximation of SDF, based on the aforementioned straightforward construction. To ensure an occlusion-aware property of the weight function $w(t)$, we will still follow the basic framework of volume rendering as Eqn. 3. However, different from the conventional treatment of setting $\sigma(t) = \phi_s(f(\mathbf{p}(t)))$ as in naive solution above, we define our function $w(t)$ from the S-density in a new manner. We first define an opaque density function $\rho(t)$, which is the counterpart of the volume density σ in standard volume rendering. Then we compute the new weight function $w(t)$ by

$$w(t) = T(t)\rho(t), \quad \text{where } T(t) = \exp(-\int_0^t \rho(u)\mathrm{d}u). \quad (5)$$

How we derive opaque density ρ. We will first consider a simple case where there is only one surface intersection, and the surface is simply a plane. Since Eqn. 4 is indeed correct under this assumption, we derive the underlying opaque density ρ corresponding to the weight definition of Eqn. 4 using the framework of volume rendering. Then we will generalize this opaque density to the general case of multiple surface intersections by the volume rendering technique.

Specifically, in the simple case of a single plane intersection, it is easy to see that the signed distance function $f(\mathbf{p}(t))$ is $-|\cos(\theta)| \cdot (t - t^*)$, where $f(\mathbf{p}(t^*)) = 0$, and θ is the angle between the view direction $\mathbf{v}$ and the outward surface normal vector $\mathbf{n}$. Because the surface is assumed locally, $|\cos(\theta)|$ is a constant. It follows from Eqn. 4 that

$$\begin{aligned} w(t) =& \frac{\phi_s(f(\mathbf{p}(t)))}{\int_{-\infty}^{+\infty} \phi_s(f(\mathbf{p}(u)))\mathrm{d}u} \\ =& \frac{\phi_s(f(\mathbf{p}(t)))}{\int_{-\infty}^{+\infty} \phi_s(-|\cos(\theta)| \cdot (u - t^*))\mathrm{d}u} \\ =& \frac{\phi_s(f(\mathbf{p}(t)))}{|\cos(\theta)|^{-1} \cdot \int_{-\infty}^{+\infty} \phi_s(u - t^*)\mathrm{d}u} \\ =& |\cos(\theta)|\phi_s(f(\mathbf{p}(t))). \end{aligned} \quad (6)$$

Recall that the weight function within the framework of volume rendering is given by $w(t) = T(t)\rho(t)$, where $T(t) = \exp(-\int_0^t \rho(u)\mathrm{d}u)$ denotes the *accumulated transmittance*. Therefore, to derive $\rho(t)$, we have

$$T(t)\rho(t) = |\cos(\theta)|\phi_s(f(\mathbf{p}(t))). \quad (7)$$

Since $T(t) = \exp(-\int_0^t \rho(u)\mathrm{d}u)$, it is easy to verify that $T(t)\rho(t) = -\frac{\mathrm{d}T}{\mathrm{d}t}(t)$. Further, note that $|\cos(\theta)|\phi_s(f(\mathbf{p}(t))) = -\frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))$. It follows that $\frac{\mathrm{d}T}{\mathrm{d}t}(t) = \frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))$. Integrating both sides of this equation yields

$$T(t) = \Phi_s(f(\mathbf{p}(t))). \quad (8)$$

Taking the logarithm and then differentiating both sides, we have

$$\begin{aligned} \int_{-\infty}^{t} \rho(u)\mathrm{d}u &= -\ln(\Phi_s(f(\mathbf{p}(t)))) \\ \Rightarrow \rho(t) &= \frac{-\frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}. \end{aligned} \tag{9}$$

This is the formula of the opacity density $\rho(t)$ in case of single plane intersection. The weight function $w(t)$ induced by $\rho(t)$ is shown in Figure 2(b). Now we generalize the opaque density to the general case where there are multiple surface intersections along the ray $\mathbf{p}(t)$. In this case, $-\frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))$ becomes negative on the segment of the ray with increasing SDF values. Thus we clip it against zero to ensure that the value of ρ is always non-negative. This gives the following opaque density function $\rho(t)$ in the general case of multiple surface intersections.

$$\rho(t) = \max\left(\frac{-\frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}, 0\right). \tag{10}$$

Based on this equation, the weight function $w(t)$ can be computed with standard volume rendering as in Eqn. 5. The illustration in the case of multiple surface intersection is shown in Figure 3.

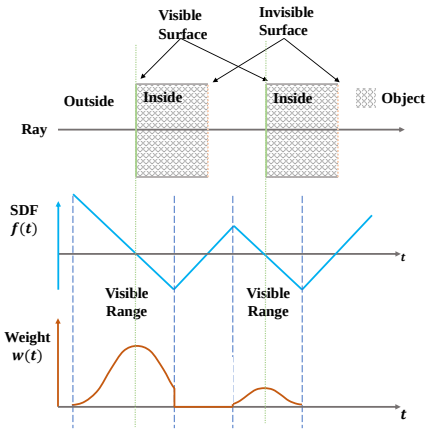


Figure 3: Illustration of weight distribution in case of multiple surface intersection.

The following theorem states that in general cases (i.e., including both single surface intersection and multiple surface intersections) the weight function defined by Eqn. 10 and Eqn. 5 is unbiased in the first-order approximation of SDF. The proof is given in the Appendix.

Theorem 1 *Suppose that a smooth surface $\mathbb{S}$ is defined by the zero-level set of the signed distance function $f(\mathbf{x}) = 0$, and a ray $\mathbf{p}(t) = \mathbf{o} + t\mathbf{v}$ enters the surface $\mathbb{S}$ from outside to inside, with the intersection point at $\mathbf{p}(t^*)$, that is, $f(\mathbf{p}(t^*)) = 0$ and there exists an interval $[t_l, t_r]$ such that $t^* \in [t_l, t_r]$ and $f(\mathbf{p}(t))$ is monotonically decreasing in $[t_l, t_r]$. Suppose that in this local interval $[t_l, t_r]$, the surface can be tangentially approximated by a sufficiently small planar patch, i.e., $\nabla\mathbf{f}$ is regarded as fixed. Then, the weight function $w(t)$ computed by Eqn. 10 and Eqn. 5 in $[t_l, t_r]$ attains its maximum at t^*.*

Discretization. To obtain discrete counterparts of the opacity and weight function, we adopt the same approximation scheme as used in NeRF [23], which is similar to the composite trapezoid quadrature. This scheme samples n points $\{\mathbf{p}_i = \mathbf{o} + t_i\mathbf{v} | i = 1, ..., n, t_i < t_{i+1}\}$ along the ray to compute the approximate pixel color of the ray as

$$\hat{C} = \sum_{i=1}^{n} T_i \alpha_i c_i, \tag{11}$$

where T_i is the discrete *accumulated transmittances* defined by $T_i = \prod_{j=1}^{i-1}(1 - \alpha_j)$, and α_i is discrete opacity values defined by

$$\alpha_i = 1 - \exp(-\int_{t_i}^{t_{i+1}} \rho(t)\mathrm{d}t), \tag{12}$$

which can further be shown to be

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{p}(t_i)) - \Phi_s(f(\mathbf{p}(t_{i+1})))}{\Phi_s(f(\mathbf{p}(t_i)))}, 0\right). \tag{13}$$

This detailed derivation of this formula for α_i is given in the Appendix.

3.2 Training

To learn the NeuS representation, we minimize the difference between the rendered pixel colors and the ground truth pixel colors, without any 3D supervision. Besides colors, if the ground truth masks are provided, we can also utilize the mask for supervision.

Specifically, we optimize our neural networks and inverse standard deviation s by randomly sampling a batch of pixels and their corresponding rays in world space $P = \{C_k, M_k, \mathbf{o}_k, \mathbf{v}_k\}$, where C_k is its pixel color and $M_k \in \{0, 1\}$ is its optional mask value, from an image in every iteration. We assume the point sampling size is n and the batch size is m. The loss function is defined as

$$\mathcal{L} = \mathcal{L}_{color} + \lambda \mathcal{L}_{reg} + \beta \mathcal{L}_{mask}. \tag{14}$$

The color loss $\mathcal{L}_{color}$ is defined as

$$\mathcal{L}_{color} = \frac{1}{m} \sum_{k} \mathcal{R}(\hat{C}_k, C_k). \tag{15}$$

Same as IDR[39], we empirically choose $\mathcal{R}$ as L1 loss, which in our observation is robust to outliers and stable in training.

We add an Eikonal term [9] on the sampled points to regularize the SDF of f_θ by

$$\mathcal{L}_{reg} = \frac{1}{nm} \sum_{k,i} (|\nabla f(\hat{\mathbf{p}}_{k,i})| - 1)^2. \tag{16}$$

The optional mask loss $\mathcal{L}_{mask}$ is defined as

$$\mathcal{L}_{mask} = \text{BCE}(M_k, \hat{O}_k), \tag{17}$$

where $\hat{O}_k = \sum_{i=1}^{n} T_{k,i} \alpha_{k,i}$ is the sum of weights along the camera ray, and BCE is the binary cross entropy loss.

Hierarchical sampling. Like other volume rendering techniques, the strategy of sampling will significantly influence the final results. In this work, we follow a similar hierarchical sampling strategy as NeRF [23]. We first uniformly sample the points on the ray and then conduct importance sampling on top of the coarse probability estimation. The difference is that, unlike NeRF which simultaneously optimizes a coarse network and a fine network, we only maintain one network, where the probability in coarse sampling is computed based on the S-density $\phi_s(f(\mathbf{x}))$ with a large fixed standard deviation while the probability of fine sampling is computed based on $\phi_s(f(\mathbf{x}))$ with the learned standard deviation.

4 Experiments

4.1 Experimental settings.

Datasets. To evaluate our approach and baseline methods, we use 15 scenes from the DTU dataset [10], same as those used in IDR [39], with a wide variety of materials, appearance and geometry, including challenging cases for reconstruction algorithms, such as non-Lambertian surfaces and thin structures. Each scene contains 49 or 64 images with the image resolution of 1600×1200. Each scene was tested with and without foreground masks provided by IDR [39]. We further tested on 7 challenging scenes from the low-res set of the BlendedMVS dataset [38](CC-4 License). Each scene has $31 - 143$ images at 768×576 pixels and masks are provided by the BlendedMVS dataset. We further captured two thin objects with 32 input images to test our approach on thin structure reconstruction.

Baselines. (1) The state-of-the-art surface rendering approach – IDR [39]: IDR can reconstruct surface with high quality but requires foreground masks as supervision for training; Since IDR has demonstrated superior quality compared to another surface rendering based method – DVR [24], we did not conduct a comparison with DVR. (2) The state-of-the-art volume rendering approach – NeRF [23]: NeRF achieves impressive results in novel view synthesis, however, extracting high-quality surface is not trivial. We use a density threshold of 25 to extract mesh from the learned

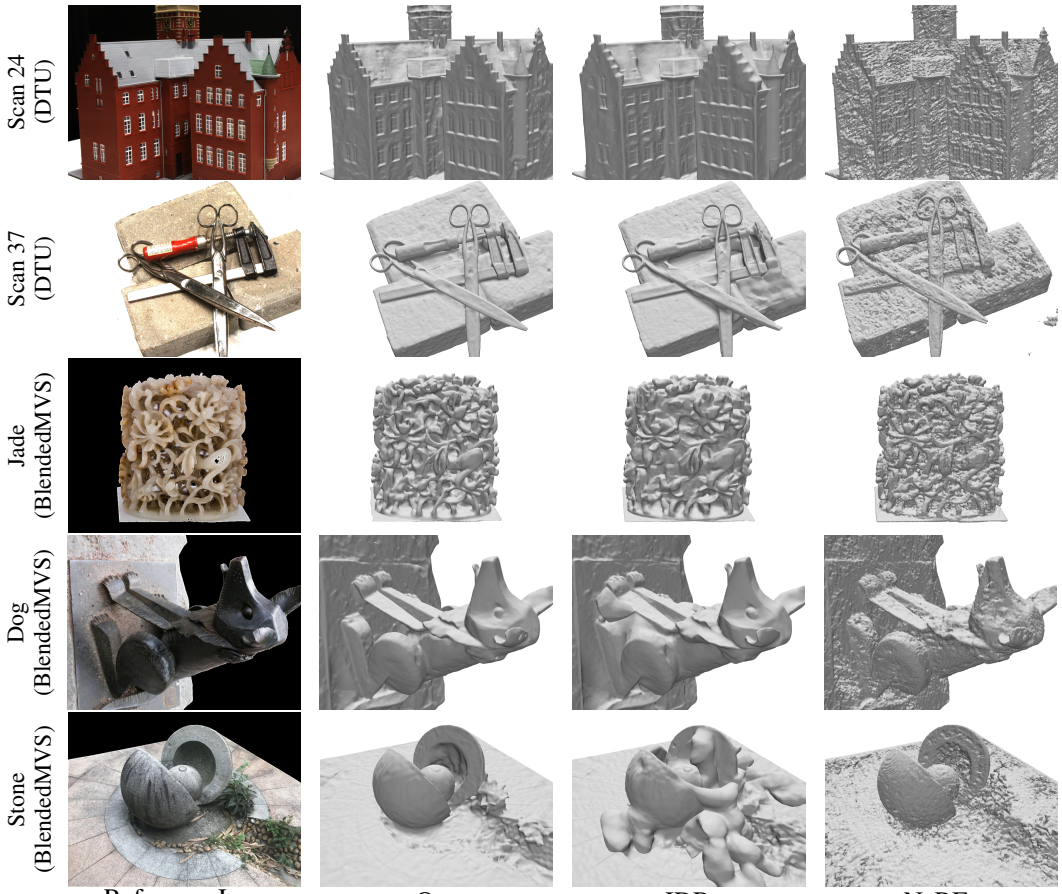


Figure 4: Comparions on surface reconstruction with mask supervision.

implicit field. We validate this choice in the Appendix. (3) A widely-used classical MVS method – COLMAP [31]: We reconstruct a mesh from the output point cloud of COLMAP with Screened Poisson Surface Reconstruction [13]. (4) The concurrent work which unifies surface rendering and volume rendering with an occupancy field as scene representation – UNISURF [25]. More details of the baseline methods are included in the Appendix.

Implementation details. Similar to the network architecture of IDR [39], the signed distance function f is modeled by a MLP that consists of 8 hidden layers with hidden size of 256. The function c for color prediction is modeled by a MLP with 4 hidden layers with size of 256, which is conditioned on the spatial location $\mathbf{p}$, normal $\mathbf{n}$, and the feature vector from f. Positional encoding [23] is applied to spatial location $\mathbf{p}$ with 6 frequencies and to view direction $\mathbf{v}$ with 4 frequencies. We assume the region of interest is inside a unit sphere. The number of coarse and fine sampling is 64 and 64 respectively. For the 'w/o mask' setting, we sample additional 32 points outside the sphere, the outside scene is presented using NeRF++ [43]. Geometric initialization is used to produce an approximate SDF as proposed in [1]. We sample 512 rays per batch and train our model for 300k iterations for 14 hours (for the 'w/ mask' setting) and 16 hours (for the 'w/o mask' setting) on a single NVIDIA RTX2080Ti GPU.

	w/ mask			**w/o** mask			
ScanID	IDR	NeRF	Ours	COLMAP	NeRF	UNISURF	Ours
scan24	1.63	1.83	**1.15**	**0.81**	1.90	1.32	1.37
scan37	1.87	2.39	**0.95**	2.05	1.60	1.36	**1.21**
scan40	**0.63**	1.79	0.80	**0.73**	1.85	1.72	**0.73**
scan55	0.48	0.66	**0.39**	1.22	0.58	0.44	**0.40**
scan63	**1.04**	1.79	1.26	1.79	2.28	1.35	**1.20**
scan65	0.79	1.44	**0.72**	1.58	1.27	0.79	**0.70**
scan69	0.77	1.50	**0.69**	1.02	1.47	0.80	**0.72**
scan83	1.33	1.20	**0.94**	3.05	1.67	1.49	**1.01**
scan97	1.16	1.96	**1.14**	1.40	2.05	1.37	**1.16**
scan105	**0.76**	1.27	0.77	2.05	1.07	0.89	**0.82**
scan106	0.67	1.44	**0.66**	1.00	0.88	**0.59**	0.66
scan110	**0.90**	2.61	1.35	**1.32**	2.53	1.47	1.69
scan114	0.42	1.04	**0.39**	0.49	1.06	0.46	**0.39**
scan118	**0.51**	1.13	**0.51**	0.78	1.15	0.59	**0.49**
scan122	0.53	0.99	**0.52**	1.17	0.96	0.62	**0.51**
mean	0.90	1.54	**0.82**	1.36	1.49	1.02	**0.87**

Table 1: Quantitative evaluation on DTU dataset. COLMAP results are achieved by trim=0.

4.2 Comparisons

We conducted the comparisons in two settings, with mask supervision (w/ mask) and without mask supervision (w/o mask). We measure the reconstruction quality with the Chamfer distances in the same way as UNISURF [25] and IDR [39] and report the scores in Table 1. The results show that our approach outperforms the baseline methods on the DTU dataset in both settings – w/ and w/o mask in terms of the Chamfer distance. Note that the reported scores of IDR in the setting of w/ mask and NeRF and UNISURF in the w/o mask setting are from IDR [39] and UNISURF [25].

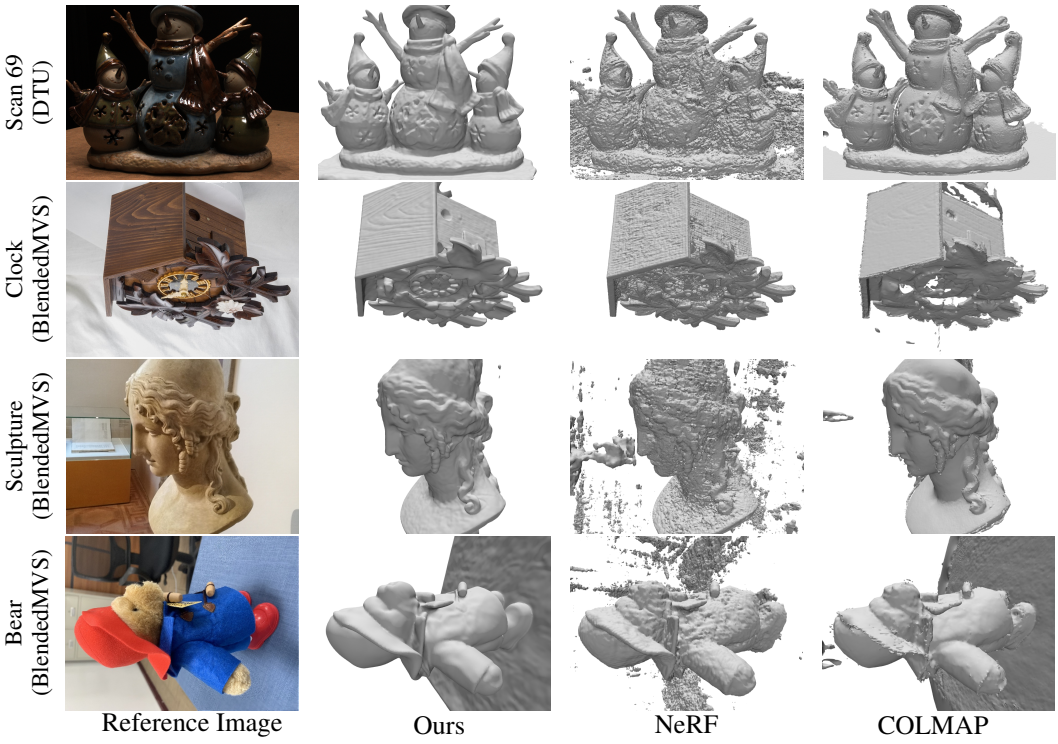


Figure 6: Comparions on surface reconstruction without mask supervision.

We conduct the qualitative comparisons on the DTU dataset and the BlendedMVS dataset in both settings, w/ mask and w/o mask, in Figure 4 and Figure 6, respectively. As shown in Figure 4 for the setting of w/ mask, IDR shows limited performance for reconstructing thin metals parts in Scan 37 (DTU) and Jade (BlendedMVS), and fails to handle sudden depth changes in Stone (BlendedMVS) due to the local optimization process in surface rendering. The extracted meshes of NeRF's results are noisy since the volume density field has not sufficient constraint on level sets of 3D geometry. Regarding the w/o mask setting, we visually compare our method with NeRF and COLMAP in the setting of w/o mask in Figure 6, which shows our reconstructed surfaces are with more fidelity than baselines. We further show a comparison with UNISURF [25] on two examples in the w/o mask setting. Note that we use the qualitative results of UNISURF reported their paper for comparison. Our method works better for the objects with abrupt depth changes. More qualitative images are included in the Appendix.

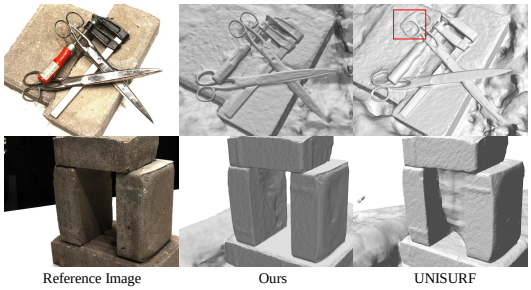


Figure 5: Visual comparisons with UNISURF.

4.3 Analysis

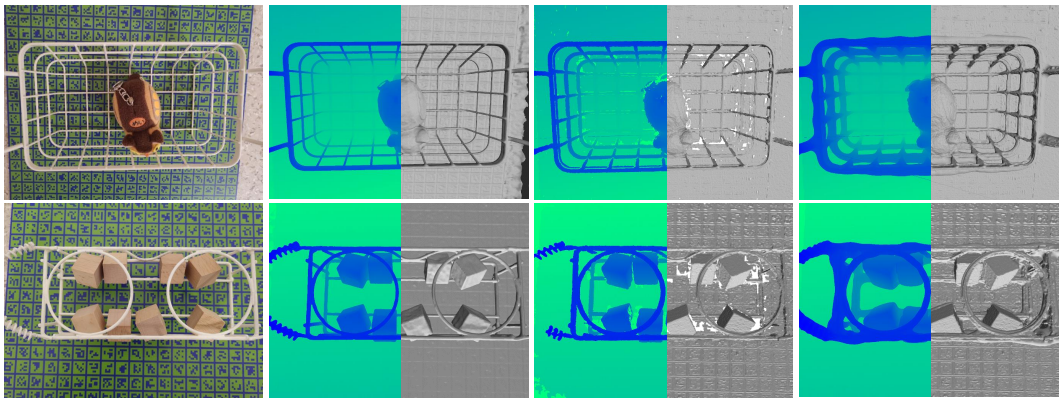
Ablation study. To evaluate the effect of the weight calculation, we test three different kinds of weight constructions described in Sec. 3.1: (a) Naive Solution. (b) Straightforward Construction as shown in Eqn. 4. (d) Full Model. As shown in Figure 7, although reconstruction geometry of naive solution looks plausible, the quantitative result is worse than our weight choice (d) in terms of the Chamfer distance. This is because it introduces a bias to the surface reconstruction. If direct construction is used, there are severe artifacts.

We also studied the effect of geometric initialization [1]. When the random initialization is used, artifacts appear at nose and eyes of the skull. More analysis can be found in the Appendix.

Thin structures. We additionally show results on two challenging thin objects with 32 input images. Note that the plane with rich texture under the object is used for camera calibration. As shown in Fig. 8, our method is able to accurately reconstruct these thin structures, especially on the edges with abrupt depth changes. Furthermore, different from the methods [34, 15, 37, 16] which only target at high-quality thin structure reconstruction, our method can handle the scenes which have a mixture of thin structures and general objects.

(a) Naive Solution	(b) Direct Construction	(c) **w/o** Geo-Init.	(d) Full Model	Reference Image
1.68	3.22	0.87	0.72	Chamfer Distance

Figure 7: Ablation study. The bottom line shows the Chamfer distance between the reconstruction results and ground-truth model.



Reference Image (a) Ours (b) COLMAP$_{trim\,=\,10}$ (c) COLMAP$_{trim\,=\,7}$

Figure 8: Comparison on scenes with thin structure objects. Left half is the depth map while right half is the reconstructed surface.

5 Conclusion

We have proposed *NeuS*, a new approach to multiview surface reconstruction that represents 3D surfaces as neural SDF and developed a new volume rendering method for training the implicit SDF representation. NeuS produces high-quality reconstruction and successfully reconstructs objects with severe occlusions and complex structures. It outperforms the state-of-the-arts both qualitatively and

quantitatively. One limitation of our method is that although our method does not heavily rely on correspondence matching of texture features, the performance would still degrade for textureless objects (we show the failure cases in the Appendix). Moreover, NeuS has only a single scale parameter s that is used to model the standard deviation of the probability distribution for all the spatial location. Hence, an interesting future research topic is to model the probability with different variances for different spatial locations together with the optimization of scene representation, depending on different local geometric characteristics. Negative societal impact: like many other learning-based works, our method requires a large amount of computational resources for network training, which can be a concern for global climate change.

Acknowlegements

We thank Michael Oechsle for providing the results of UNISURF. Christian Theobalt was supported by ERC Consolidator Grant 770784. Lingjie Liu was supported by Lise Meitner Postdoctoral Fellowship.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[3] Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 388–393. IEEE, 2001.

[4] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5932–5941, 2019.

[5] Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 418–425, 1999.

[6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[7] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[8] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 25(361-369):2, 2016.

[9] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

[10] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014.

[11] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] Srinivas Kaza et al. *Differentiable volume rendering using signed distance functions*. PhD thesis, Massachusetts Institute of Technology, 2019.

[13] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013.

[14] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. *arXiv preprint arXiv:2103.11571*, 2021.

[15] Lingjie Liu, Duygu Ceylan, Cheng Lin, Wenping Wang, and Niloy J. Mitra. Image-based reconstruction of wire art. 36(4):63:1–63:11, 2017.

[16] Lingjie Liu, Nenglun Chen, Duygu Ceylan, Christian Theobalt, Wenping Wang, and Niloy J. Mitra. Curvefusion: Reconstructing thin structures from rgbd sequences. 37(6), 2018.

[17] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33, 2020.

[18] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020.

[19] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):65, 2019.

[20] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. pages 1–8, 01 2007.

[21] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[22] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[23] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[24] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[25] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *arXiv preprint arXiv:2104.10078*, 2021.

[26] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[27] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *ArXiv*, abs/2003.04618, 2020.

[28] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *ICCV*, 2019.

[29] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020.

[30] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv:1602.07868*, 2016.

[31] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[32] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[33] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1119–1130, 2019.

[34] A. Tabb. Shape from silhouette probability maps: Reconstruction of thin objects in the presence of silhouette extraction and calibration error. pages 161–168, June 2013.

[35] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

[36] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d scene representation and rendering. *arXiv preprint arXiv:2010.04595*, 2020.

[37] Peng Wang, Lingjie Liu, Nenglun Chen, Hung-Kuo Chu, Christian Theobalt, and Wenping Wang. Vid2curve: Simultaneous camera motion estimation and thin structure reconstruction from an rgb video. *ACM Trans. Graph.*, 39(4), July 2020.

[38] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799, 2020.

[39] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020.

[40] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. *arXiv preprint arXiv:2012.06434*, 2020.

[41] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[42] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. *arXiv preprint arXiv:2104.00674*, 2021.

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

- Appendix -

A Derivation for Computing Opacity α_i

In this section we will derive the formula in Eqn. 13 for computing the discrete opacity α_i. Recall that the opaque density function $\rho(t)$ is defined by Eqn. 10 as

$$\begin{aligned}\rho(t) =& \max\left(\frac{-\frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}, 0\right)\\ =& \max\left(\frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}, 0\right),\end{aligned}$$

where $\phi_s(x)$ and $\Phi_s(x)$ are the probability density function (PDF) and cumulative distribution function (CDF) of logistic distribution, respectively. First consider the case where the sample point interval $[t_i, t_{i+1}]$ lies in a range $[t_\ell, t_r]$ over which the camera ray is entering the surface from outside to inside, i.e. the signed distance function is decreasing on the camera ray $\mathbf{p}(t)$ over $[t_\ell, t_r]$. Then it is easy to see that $-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v}) > 0$ in $[t_i, t_{i+1}]$. It follows from Eqn. 12 that,

$$\begin{aligned}\alpha_i =& 1-\exp\left(-\int_{t_i}^{t_{i+1}}\rho(t)\mathrm{d}t\right)\\ =& 1-\exp\left(-\int_{t_i}^{t_{i+1}}\frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}\mathrm{d}t\right).\end{aligned} \tag{18}$$

Note that the integral term is computed by

$$\int\frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}\mathrm{d}t = -\ln(\Phi_s(f(\mathbf{p}(t)))) + C, \tag{19}$$

where C is a constant. Thus the discrete opacity can be computed by

$$\begin{aligned}\alpha_i =& 1-\exp\left[-\left(-\ln(\Phi_s(f(\mathbf{p}(t_{i+1}))))+\ln(\Phi_s(f(\mathbf{p}(t_i))))\right)\right]\\ =& 1-\frac{\Phi_s(f(\mathbf{p}(t_{i+1})))}{\Phi_s(f(\mathbf{p}(t_i)))}\\ =& \frac{\Phi_s(f(\mathbf{p}(t_i)))-\Phi_s(f(\mathbf{p}(t_{i+1})))}{\Phi_s(f(\mathbf{p}(t_i)))}.\end{aligned} \tag{20}$$

Next consider the case where $[t_i, t_{i+1}]$ lies in a range $[t_\ell, t_r]$ over which the camera ray is exiting the surface, i.e. the signed distance function is increasing on $\mathbf{p}(t)$ over $[t_\ell, t_r]$. Then we have $-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v}) < 0$ in $[t_i, t_{i+1}]$. Then, according to Eqn. 10, we have $\rho(t) = 0$. Therefore, by Eqn. 12, we have

$$\alpha_i = 1-\exp\left(-\int_{t_i}^{t_{i+1}}\rho(t)\mathrm{d}t\right) = 1-\exp\left(-\int_{t_i}^{t_{i+1}}0\mathrm{d}t\right) = 0.$$

Hence, the alpha value α_i in this case is given by

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{p}(t_i)))-\Phi_s(f(\mathbf{p}(t_{i+1})))}{\Phi_s(f(\mathbf{p}(t_i)))}, 0\right). \tag{21}$$

This completes the derivation of Eqn. 13.

B First-order Bias Analysis

B.1 Proof of Unbiased Property of Our Solution

PROOF OF THEOREM 1: Suppose that the ray is going from outside to inside of the surface. Hence, we have $-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v}) > 0$, because by convention the signed distance function $f(\mathbf{x})$ is positive outside and negative inside of the surface.

Recall that our S-density field $\phi_s(f(\mathbf{x}))$ is defined using the logistic density function $\phi_s(x) = se^{-sx}/(1+e^{-sx})^2$, which is the derivative of the Sigmoid function $\Phi_s(x) = (1+e^{-sx})^{-1}$, i.e. $\phi_s(x) = \Phi_s'(x)$.

According to Eqn. 5, the weight function $w(t)$ is given by

$$w(t) = T(t)\rho(t),$$

where

$$\rho(t) = \max\left(\frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}, 0\right).$$

By assumption, $-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v}) > 0$ for $t \in [t_l, t_r]$. Since ϕ_s is a probability density function, we have $\phi_s(f(\mathbf{p}(t))) > 0$. Clearly, $\Phi_s(f(\mathbf{p}(t))) > 0$. It follows that

$$\rho(t) = \frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))},$$

which is positive. Hence,

$$\begin{aligned}
w(t) =& T(t)\rho(t)\\
=& \exp\left(-\int_0^t \rho(t')\mathrm{d}t'\right)\rho(t)\\
=& \exp\left(-\int_0^{t_l} \rho(t')\mathrm{d}t'\right)\exp\left(-\int_{t_l}^t \rho(t')\mathrm{d}t'\right)\rho(t)\\
=& T(t_l)\exp\left(-\int_{t_l}^t \rho(t')\mathrm{d}t'\right)\rho(t)\\
=& T(t_l)\exp\left[-(-\ln(\Phi_s(f(\mathbf{p}(t))))+\ln(\Phi_s(f(\mathbf{p}(t_l)))))\right]\rho(t)\\
=& T(t_l)\frac{\Phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t_l)))}\frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}\\
=& \frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})T(t_l)}{\Phi_s(f(\mathbf{p}(t_l)))}\phi_s(f(\mathbf{p}(t))).
\end{aligned} \tag{22}$$

As a first-order approximation of signed distance function f, suppose that locally the surface is tangentially approximated by a sufficiently small planar patch with its outward unit normal vector denoted as $\mathbf{n}$. Because $f(\mathbf{x})$ is a signed distance function, locally it has a unit gradient vector $\nabla f = \mathbf{n}$. Then we have

$$\begin{aligned}
w(t) =& \frac{-(\nabla f(\mathbf{p}(t))\cdot\mathbf{v})T(t_l)}{\Phi_s(f(\mathbf{p}(t_l)))}\phi_s(f(\mathbf{p}(t)))\\
=& \frac{|\cos(\theta)|T(t_l)}{\Phi_s(f(\mathbf{p}(t_l)))}\phi_s(f(\mathbf{p}(t))),
\end{aligned} \tag{23}$$

where θ is the angle between the view direction $\mathbf{v}$ and the unit normal vector $\mathbf{n}$, that is, $\cos(\theta) = \mathbf{v}\cdot\mathbf{n}$. Here $|\cos(\theta)|T(t_l)\cdot\Phi_s(f(\mathbf{p}(t_l)))^{-1}$ can be regarded as a constant. Hence, $w(t)$ attains a local maximum when $f(\mathbf{p}(t)) = 0$ because $\phi_s(x)$ is a unimodal density function attaining the maximal value at $x = 0$.

We remark that in this proof we do not make any assumption on the existence of surfaces between the camera and the sample point $\mathbf{p}(t_l)$. Therefore the conclusion holds true for the case of multiple surface intersections on the camera ray. This completes the proof. □

B.2 Bias in Naive Solution

In this section we show that the weight function derived in naive solution is biased. According to Eqn. 3, $w(t) = T(t)\sigma(t)$, with the opacity $\sigma(t) = \phi_s(f(\mathbf{p}(t)))$. Then we have

$$\begin{aligned}\frac{\mathrm{d}w}{\mathrm{d}t} =& \frac{\mathrm{d}(T(t)\sigma(t))}{\mathrm{d}t} \\ =& \frac{\mathrm{d}T(t)}{\mathrm{d}t}\sigma(t) + T(t)\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} \\ =& \left[\exp\left(-\int_0^t \sigma(t)\mathrm{d}t\right)(-\sigma(t))\right]\sigma(t) + T(t)\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} \\ =& T(t)(-\sigma(t))\sigma(t) + T(t)\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} \\ =& T(t)\left(\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} - \sigma(t)^2\right).\end{aligned} \tag{24}$$

Now we perform the same first-order approximation of signed distance function f near the surface intersection as in Section B.1. In this condition, the above equation can be rewritten as

$$\begin{aligned}\frac{\mathrm{d}w}{\mathrm{d}t} =& T(t)\left((\nabla f(\mathbf{p}(t))\cdot\mathbf{v})\phi_s'(f(\mathbf{p}(t))) - \phi_s(f(\mathbf{p}(t)))^2\right) \\ =& T(t)\left(\cos(\theta)\phi_s'(f(\mathbf{p}(t))) - \phi_s(f(\mathbf{p}(t)))^2\right).\end{aligned} \tag{25}$$

Here $\cos(\theta)$ can be regarded as a constant. Now suppose $\mathbf{p}(t^*)$ is a point on the surface $\mathbb{S}$, that is, $f(\mathbf{p}(t^*)) = 0$. Next we will examine the value of $\frac{\mathrm{d}w}{\mathrm{d}t}(t)$ at $t = t^*$. First, clearly, $T(t^*) > 0$ and $\phi_s(f(\mathbf{p}(t^*)))^2 > 0$. Then, since $\phi_s'(0) = 0$, we have

$$\frac{\mathrm{d}w}{\mathrm{d}t}(t^*) = T(t^*)(\cos(\theta)\phi_s'(0) - \sigma(t^*)^2) = -T(t^*)\phi_s(0)^2 < 0.$$

Hence $w(t)$ in naive solution does not attain a local maximum at $t = t^*$, which corresponds to a point on the surface $\mathbb{S}$. This completes the proof. □

C Second-order Bias Analysis

In this section we briefly introduce our local analysis in the interval $[t_l, t_r]$ near the surface intersection, in second-order approximation. In this condition, we follow the similar assumption as Section B that the signed distance function $f(\mathbf{p}(t))$ monotonically decreases along the ray in the interval $[t_l, t_r]$.

According to Eqn. 24, the derivative of $w(t)$ is given by:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = T(t)\left(\frac{\mathrm{d}\sigma(t)}{\mathrm{d}t} - \sigma(t)^2\right).$$

Clearly, we have $T(t) > 0$. Hence, when $w(t)$ attains local maximum at $\bar{t}$, there is $\left(\frac{\mathrm{d}\sigma(\bar{t})}{\mathrm{d}t} - \sigma(\bar{t})^2\right) = 0$.

The case of our solution. In our solution, the volume density is given by $\sigma(t) = \rho(t)$ following Eqn. 10. After organizing, we have

$$\frac{\mathrm{d}^2 f}{\mathrm{d}t}(\mathbf{p}(\bar{t}))\cdot\phi_s(f(\mathbf{p}(\bar{t}))) + \left(\frac{\mathrm{d}f}{\mathrm{d}t}(\mathbf{p}(\bar{t}))\right)^2\phi_s'(f(\mathbf{p}(\bar{t}))) = 0.$$

Here we perform a local analysis at $\bar{t}$ near the surface intersection t^*, where $f(\mathbf{p}(t^*)) = 0$, $\bar{t} = t^* + \Delta_t$. And we let $\frac{\mathrm{d}f}{\mathrm{d}t}(\mathbf{p}(t^*)) = \mu$, and $\frac{\mathrm{d}^2 f}{\mathrm{d}t^2}(\mathbf{p}(t^*)) = \tau$. As a second-order analysis, we assume that in this local interval $t \in [t_l, t_r]$, $\frac{\mathrm{d}^2 f}{\mathrm{d}t^2}(\mathbf{p}(t))$ is fixed. After substitution and organization, the induced equation for local maximum point $\bar{t}$ is

$$\tau\cdot\left(1 + e^{-s(\mu\Delta_t + \frac{1}{2}\tau\Delta_t^2)}\right) = (\mu + \tau\Delta_t)^2\cdot\left(s\left(1 - e^{-s(\mu\Delta_t + \frac{1}{2}\tau\Delta_t^2)}\right)\right), \tag{26}$$

which we will analyze later.

The case of the naive solution. Here we conduct a similar local analysis as in case of our solution. Regarding naive solution, when $w(t)$ attains local maximum at $\bar{t}$, there is:

$$(\mu + \tau\Delta_t) \cdot \left(-\left(1 - e^{-2s(\mu\Delta_t + \frac{1}{2}\tau\Delta_t^2)}\right)\right) = e^{-s(\mu\Delta_t + \frac{1}{2}\tau\Delta_t^2)}. \tag{27}$$

Comparison. Based on Eqn. 26 and Eqn. 27, we can numerically solve the equations on Δ_t for any given values of μ, τ, and s. Below we plot the curves of Δ_t versus increasing s for different (fixed) values of μ, τ in Fig. 9.

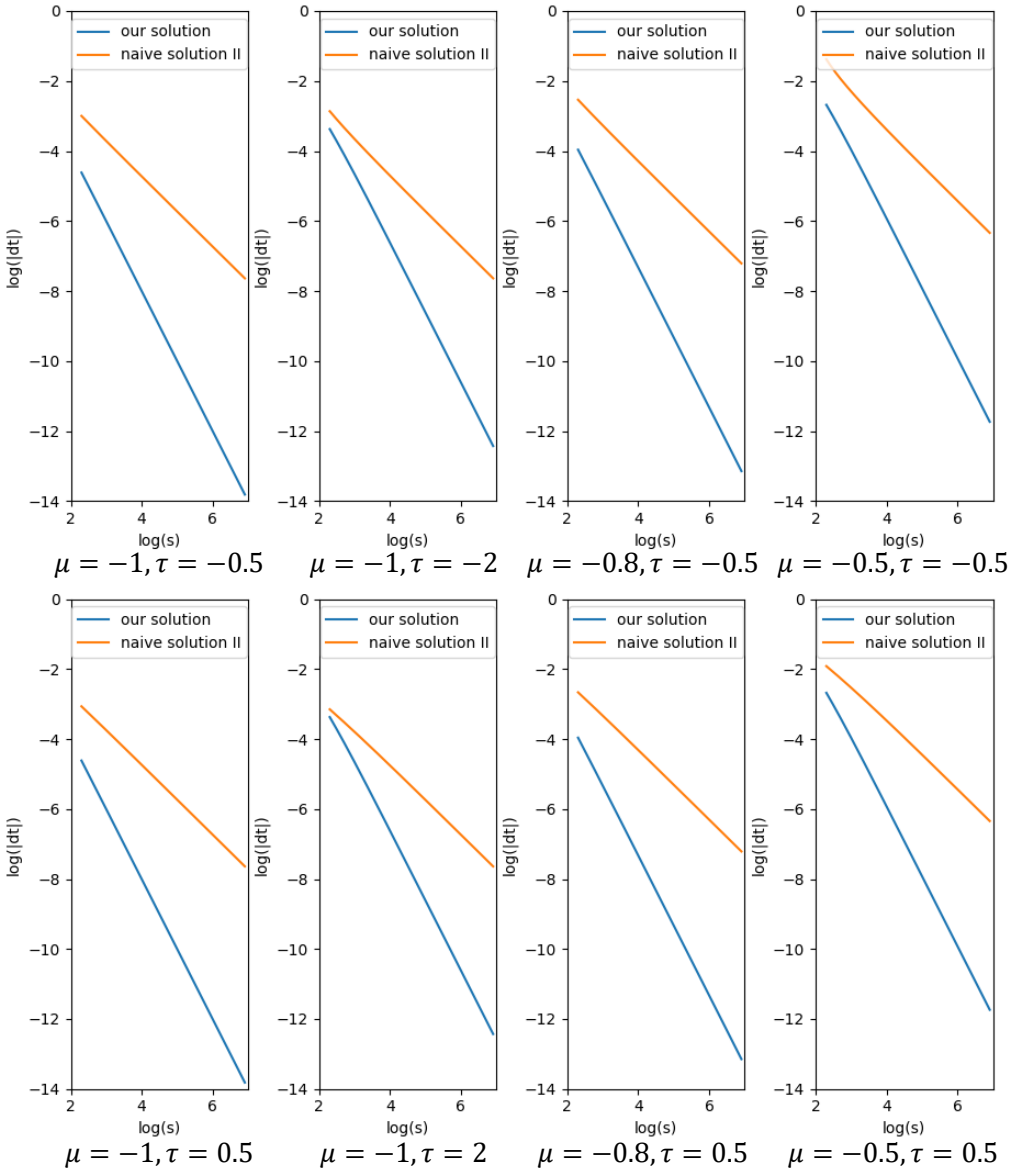


Figure 9: The curve of Δ_t versus s, given fixed μ, τ. Note that the axes are illustrated in $\ln(|\Delta_t|)$ and $\ln(s)$.

As shown in Fig. 9, the error of local maximum position $\Delta_t = O(s^{-2})$ for our solution and the error $\Delta_t = O(s^{-1})$ for the naive solution. That is to say, our error converges to zero faster than the error of the naive solution does as the standard deviation $1/s$ of the S-density approaches to 0, which is quadratic convergence versus linear convergence.

D Additional Experimental Details

D.1 Additional Implemenation Details

Network architecture. We use a similar network architecture as IDR[39], which consists of two MLPs to encode SDF and color respectively. For the SDF MLP, we replace original ReLU with Softplus with $\beta = 100$ as activation functions for all hidden layers. A skip connection [26] is used to connect the input with the output of the fourth layer. The color MLP takes not only the spatial location $\mathbf{p}$ as inputs but also the view direction $\mathbf{v}$, the normal vector of SDF $\mathbf{n} = \nabla f(\mathbf{p})$, and a 256-dimensional feature vector from the SDF MLP. Same as IDR, we use weight normalization [30] to stabilize the training process.

Alpha and color computation. In the implementation, we actually have two types of sampling points - the sampled section points $\mathbf{q}_i = \mathbf{o} + t_i\mathbf{v}$ and the sampled mid-points $\mathbf{p}_i = \mathbf{o} + \frac{t_i+t_{i+1}}{2}\mathbf{v}$, with section length $\delta_i = t_{i+1} - t_i$, as illustrated in Figure 10. To compute the alpha value α_i, we use the section points, which is $\max(\frac{(\Phi_s(f(\mathbf{q}_i)) - \Phi_s(f(\mathbf{q}_{i+1})))}{\Phi_s(f(\mathbf{q}_i))}, 0)$. To compute the color c_i, we use the color of the mid-point $\mathbf{p}_i$.

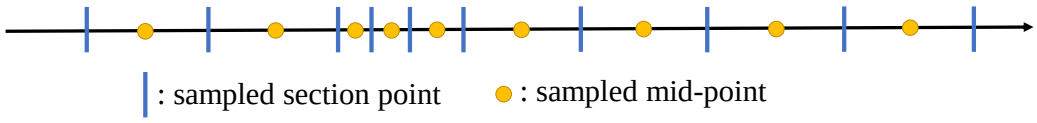


Figure 10: The section points and mid-points defined on a ray.

D.2 Baselines

IDR[39]. To implement IDR, we use their officially released codes[1] and pretrained models on the DTU dataset.

NeRF[23]. To implement NeRF, we use the code from nerf-pytorch[2]. To extract surfaces from NeRF, we use the density level-set of 25, which is validated by experiments to be the best level-set with smallest reconstruction errors, as shown in Table 2 and Figure 11.

COLMAP[31]. We use the officially provided CLI(command line interface) version of COLMAP. Dense point clouds are produced by sequentially running following commands: (1) *feature_extractor*, (2) *exhaustive_matcher*, (3) *patch_match_stereo*, and (4) *stereo_fusion*. Given dense point clouds, meshes are produced by (5) *poisson_mesher*.

UNISURF[25]. Since the code of the concurrent work UNISURF has not been released yet, the quantitative and qualitative results in the paper are provided by the authors of UNISURF.

Scan ID	Threshold 0	Threshold 25	Threshold 50	Threshold 100	Threshold 500
Scan 40	2.36	**1.79**	1.86	2.07	4.26
Scan 83	1.65	**1.20**	1.37	2.24	29.10
Scan 114	1.62	**1.04**	1.10	1.43	8.66

Table 2: The Chamfer distances between the ground-truth and the level-set surfaces extracted from the NeRF results using different threshold values on three scenes from the DTU dataset.

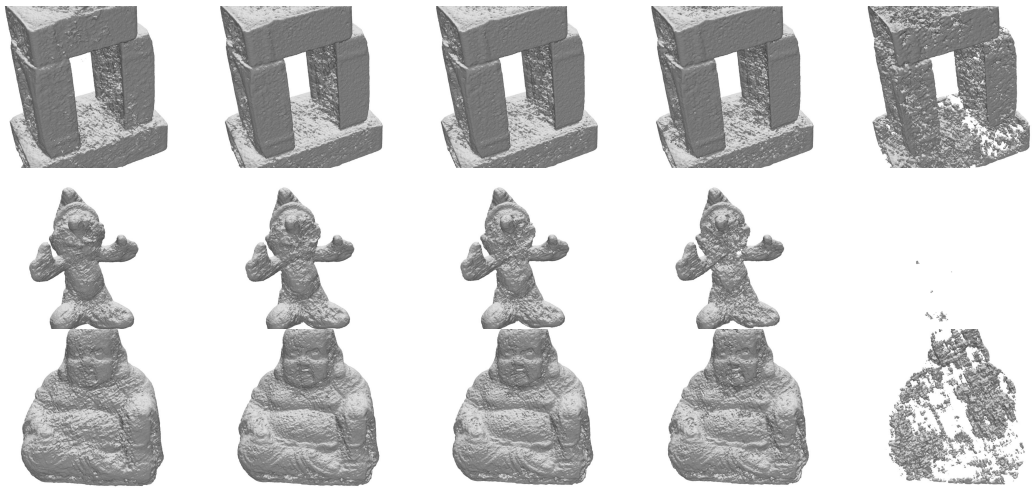


Figure 11: The visualization of the level-set surfaces extracted from the NeRF results using different threshold values.

E More Experimental Results

E.1 Effect of Geometric Initialization

Although our method can produce plausible results with random initialization, our method with geometric initialization achieves better quality. As shown in Figure 12, using random initialization produces axis-aligned artifacts due to the spectral-bias of positional encoding [35] while the geometric initialization [1] does not have this kind of artifacts.

E.2 Training Progression

We show the reconstructed surfaces at different training stages of the Durian in the BlendedMVS dataset. As illustrated in Figure 13, the surface gets sharper along the training process. Meanwhile,

[1] https://github.com/lioryariv/idr
[2] https://github.com/yenchenlin/nerf-pytorch

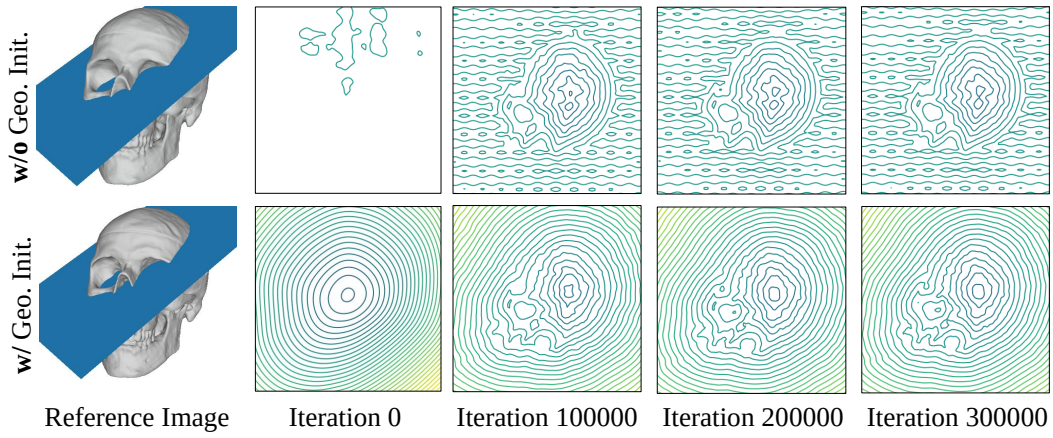


Figure 12: Visualization of signed distance fields on the cutting plane (blue plane of the left image) in different training iterations.

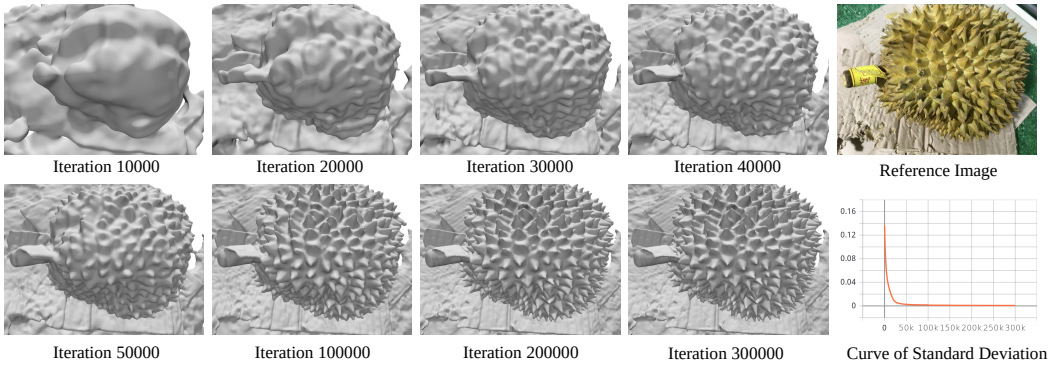


Figure 13: Training progression of the Durian in the BlendedMVS dataset. The bottom right figure shows the curve of the trainable standard deviation in the training progress.

we also provide a curve in the figure to show how the trainable standard deviation in ϕ_s changes in the training process. As we can see, the optimization process will automatically reduce the standard deviation so that the surface becomes more clear and sharper with more training steps.

E.3 Limitation

Figure 14 shows a failure case where our method fails to correctly reconstruct the texutreless region of the inner surface on the right brick. The reason is that such textureless regions are ambiguous for reconstruction in neural rendering.

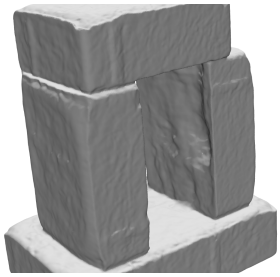


Figure 14: A failure case containing textureless regions.

E.4 Additional Results

In this section, we show additional qualitative results on the DTU dataset and BlendedMVS dataset. Figure 16 shows the comparisons with baseline methods in both **w/** mask setting and **w/o** mask setting. Figure 17 shows addtional results in **w/o** mask setting. Meanwhile, besides the reconstructed surfaces, our method also produces high-quality rendered images as shown in Figure 15.

Figure 15: Rendered images by our method on the DTU dataset.

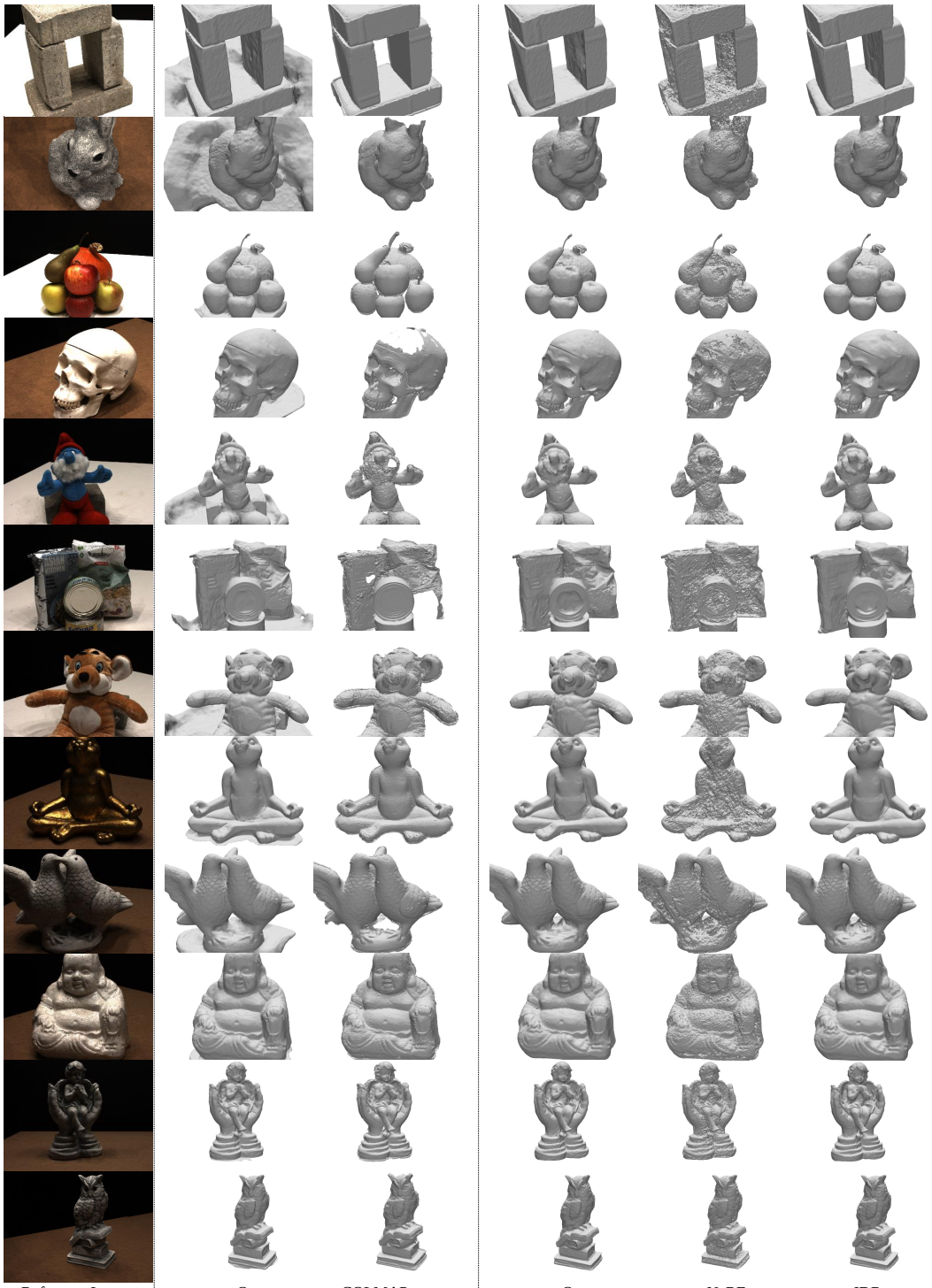


Figure 16: Additional reconstruction results on the DTU dataset.

Figure 17: Additional reconstruction results on BlendedMVS dataset without mask supervision.