

A Probability-guided Sampler for Neural Implicit Surface Rendering

Gonalo Dias Pais^{1,2*}, Valter Piedade², Moitrey Chatterjee¹,
Marcus Greiff³, and Pedro Miraldo¹

¹ Mitsubishi Electric Research Laboratories (MERL), Cambridge

² Instituto Superior Tcnico, Universidade de Lisboa

³ Toyota Research Institute, Los Altos

Abstract. Several variants of Neural Radiance Fields (NeRFs) have significantly improved the accuracy of synthesized images and surface reconstruction of 3D scenes/objects. In all of these methods, a key characteristic is that none can train the neural network with every possible input data, specifically, every pixel and potential 3D point along the projection rays due to scalability issues. While vanilla NeRFs uniformly sample both the image pixels and 3D points along the projection rays, some variants focus only on guiding the sampling of the 3D points along the projection rays. In this paper, we leverage the implicit surface representation of the foreground scene and model a probability density function in a 3D image projection space to achieve a more targeted sampling of the rays toward regions of interest, resulting in improved rendering. Additionally, a new surface reconstruction loss is proposed for improved performance. This new loss fully explores the proposed 3D image projection space model and incorporates near-to-surface and empty space components. By integrating our novel sampling strategy and novel loss into current state-of-the-art neural implicit surface renderers, we achieve more accurate and detailed 3D reconstructions and improved image rendering, especially for the regions of interest in any given scene. Project page: <https://merl.com/research/highlights/ps-neus>.

Keywords: Neural implicit surface renderer · Non-uniform sampler · Probability density function · Signed distance functions

1 Introduction

Recovering the 3D structure of the scene and rendering it from new views is valuable for numerous tasks such as Augmented Reality/Virtual Reality asset creation, 3D reconstruction [53, 54], environment mapping [18, 36, 41, 66], etc. In the last few years, Neural Radiance Fields (NeRF) [26] have emerged as a promising solution for this task. These learn a mapping from a 3D point and a viewing direction to its color and volume density. In theory, it may be desirable to train NeRFs on every pixel and every scene image from the training data.

*Work was partly done while interning at MERL.

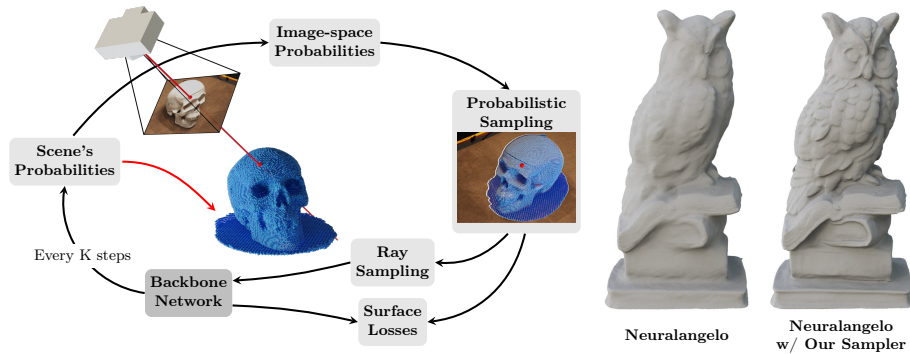


Fig. 1: Schematic of our pipeline and an example: We leverage neural implicit representations to guide the sampling of rays during the training of the neural surface rendering pipeline. In particular, we sample a **point** in the three-dimensional image space (image coordinates + depth) (**Probability-guided Sampling**) to obtain the ray (**Ray Sampling**) and some additional points around the surface, along the ray. These are then passed through the **Backbone Network**, which is regularized using the sampled depth (**Surface Reconstruction Losses**). The two figures on the right show a 3D reconstruction of an example DTU scene (DTU122) using Neuralangelo as a backbone, with (rightmost) and without (second from right) our sampling strategy.

However, given the large amount of data, this is infeasible, *i.e.*, one must sample the image pixels uniformly and the points on the projection ray.

Recent approaches like NeuS [50] and its variants [16, 21, 51, 52] leverage neural implicit representations to achieve finer detail and higher-resolution 3D surface reconstruction, particularly of 3D objects. These methods typically employ Signed Distance Fields (SDF) or occupancy to implicitly represent the foreground surfaces. All NeuS-inspired methodologies adhere to a standardized sampling strategy involving uniform sampling of pixels and corresponding projection rays followed by hierarchical sampling of 3D points along the projection rays.

This paper argues that the full potential of neural implicit representations has not yet been explored. While many existing methods focus solely on guiding the sampling of 3D points along projection rays, disregarding the crucial aspect of pixel sampling [3–5, 26], trivial solutions to pixel sampling, such as uniform sampling, generate worse rendering quality in areas of interest in the scene due to insufficient representation during training. This leads us to the following research question: *Can the implicit surface representation guide training rays and points for accurate 3D reconstruction and rendering?*

There are two trivial solutions to the proposed research question: (i) The more straightforward one would be to cast a ray for every possible pixel in every camera and check whether any volume density accumulated along the ray. This solution is computationally expensive since one must run the model for all cameras and all pixels. Indeed, Sun *et al.* [45] partially follows this idea but reduces the sampling space by evaluating it in patches to make the training feasible. The

pixels are still uniformly sampled inside each patch. (ii) The second approach voxelizes the implicit surface representation and projects every possible 3D point to every camera, which is computationally expensive. Additionally, occlusions – which need to be factored in for view-dependent rendering – cannot be trivially handled using such a technique. Similar ideas were followed in works such as RegSDF [61], where the authors use the Structure from Motion (SfM) points instead. However, this is a simplification of the trivial solution and only handles much simpler sampling scenarios, which do not need to deal with voxelization, view dependency, and training issues raised from dense 3D points.

Our paper introduces a novel solution for image pixel sampling. It leverages a 3D probability density function estimated from the scene’s SDF within the 3D image space facilitated by per-camera grids. This enables efficient interpolation of camera pixels and depth approximation. Our method offers a streamlined update process, requiring only a single model run for the 3D probability density function update, thus ensuring speed and adaptability. Unlike conventional methods relying on additional depth data, our approach constructs the sampling space through 3D coordinate transformations and view constraints without any additional depth supervision. Notably, our technique enhances 3D scene rendering across various backbones. The sampling pipeline and a mesh reconstruction are represented in Fig. 1. Our main contributions are:

1. A probabilistic 3D orthographic image projection sampling for neural implicit surface rendering that is view-dependent and feasible for training;
2. A new loss function that combines near-the-surface and empty space components for better modeling foreground and background regions of the scene;
3. The sampling is agnostic to the implicit model, *i.e.*, the derived extra pipeline steps can be used with different models without changing the backbone;
4. We show that coupling our probabilistic sampling with current state-of-the-art neural implicit representation methods (namely [21, 50]) improves 3D reconstruction and rendering in regions of interest of the scene.

2 Related Work

Neural rendering: Reconstructing 3D structures from multi-view images is a core problem of computer vision, with approaches based on SfM [12, 37, 38, 53] or Simultaneous Localization and Mapping (SLAM) [6, 8, 18, 36, 41, 66]. NeRF [26] introduces a more recent view synthesis strategy that enables dense reconstructions, using volume rendering. During training, projection rays and 3D points are uniformly sampled from the image and any given projection ray, respectively. Image synthesis is achieved by rendering the sampled 3D points by volume, which gives their color and volume density values. Lately, several NeRF variants have been proposed, focusing on improving view synthesis quality [3, 4, 63], improving computational performance [3, 28, 42], scaling up to large-scale environments [9, 46, 48, 64], dynamic scenes [1, 20, 22, 24, 31, 34, 47], *etc.*

Neural implicit volume rendering: A drawback of the volume rendering in NeRF is that it imposes insufficient constraints for representing 3D surfaces.

This prevents it from learning intricate 3D object details, making high-quality reconstructions infeasible. To solve this problem, occupancy approaches [23, 25, 29, 30, 32, 35] and SDF approaches [2, 7, 11, 21, 44, 50, 51, 57, 58, 61, 62, 65] were proposed. In particular, NeuS [50] and VolSDF [57] use SDF as an implicit representation for a surface and are trained from multiple views. Both outperform NeRF-based methods, even handling scenes with occlusions. Further improvement was achieved by reducing the implicit bias, in the depth estimates [65]. Another issue with NeuS is its slow training speed. NeuS2 [51] proposes an efficient parallelization and a new training strategy to address this concern. Neuralangelo [21] introduces multi-resolution hash grids for surface rendering. This approach achieves high-quality reconstruction in highly detailed scenes, with a small cost in training efficiency. While these approaches use the learned implicit representation for sampling on the projection rays, our work utilizes a probability density function for sampling. To further improve surface reconstruction, other approaches use priors such as object masks [29, 59], depth [60, 62], normals [49, 60], or point clouds [10, 61]. These additional inputs guide the surface learning process, improving the reconstruction results and optimization time. We propose a method where sampling is guided by surface estimates, enabling rays to converge toward textured regions without the need for additional inputs.

Pixel Sampler for NeRF: The straightforward approach to sampling the rays while training NeRFs is to uniformly sample both the pixels in the image as well as the 3D points that lie on the corresponding projection rays, passing through the chosen pixels [26]. Most sampling strategies in NeRF propose to improve the 3D sampling on the ray, by exploring anti-aliasing [3, 5, 14, 19] or 3D geometry [23, 50, 57]. In contrast, this paper focuses on a more effective pixel sampling strategy for training similar to [43, 45, 55]. Sun *et al.* [45] proposes a patch sampling based on the depth and color contrast estimates for their pixel sampling, where a pre-trained model trained on the DTU [15] is used to obtain the initial proposals. Neural 3D reconstruction in the Wild [43] attempts to sample only around the surface through voxel-guided and surface-guided sampling. ActRay [55] uses a reinforcement learning agent to reduce the number of rays by focusing on the rays with the highest loss values. In contrast, we design a 3D view-dependent camera probability space, derived from the implicit representation of the surface to sample the pixel and directly gain depth information for the sampled ray. A backbone model, upon which our sampling strategy operates, is trained from scratch without needing additional information.

3 Notations and Background

3.1 Notations

Let a 3D point in world coordinates be given by $\mathbf{x} = [x, y, z] \in \mathcal{X} \subset \mathbb{R}^3$. For a set of cameras $\mathcal{C} = \{1, \dots, C\}$, the same point in the c^{th} camera, is denoted by $\hat{\mathbf{x}}_c = [\hat{x}_c, \hat{y}_c, \hat{z}_c] \in \hat{\mathcal{X}}_c$. $h_c(\cdot)$ transforms the point from the world to the camera c

(see [13]). For c , we define a *3-dimensional image space* such that

$$\mathbf{u}_c \in \mathcal{U}_c = g(\hat{\mathbf{x}}_c) = [\hat{x}_c/\hat{z}_c, \hat{y}_c/\hat{z}_c, \hat{z}_c] = [u_c, v_c, \lambda_c], \quad (1)$$

where u_c , v_c , and \mathcal{U}_c (respectively) are bounded to image size and intrinsic parameters of each camera, and depth $\lambda_c > 0$, which is bijective to the camera reference frame¹. Using the transformation from the world to the camera coordinate system $h_c(\cdot)$ and image projection $g(\cdot)$ (for more detail see [13]), we define the composition $f_c(\cdot)$, such that

$$\mathbf{u}_c = f_c(\mathbf{x}) = g(h_c(\mathbf{x})). \quad (2)$$

To simplify the notations, we sometimes omit the subscript c , for example, $\mathbf{u} = \mathbf{u}_c$, and $\hat{\mathbf{x}} = \hat{\mathbf{x}}_c$. Finally, $|\cdot|$ denotes the determinant of a matrix.

3.2 Neural Implicit Surface Rendering

Consider a set of images of a specific 3D scene, captured from calibrated cameras with known poses. A NeRF [26] creates an implicit 3D representation of the scene from known camera positions to the images. This implicit representation allows for a dense reconstruction of the scene, by simultaneously estimating the volume density and color for every 3D point. A more evolved alternative is proposed in Wang *et al.* [50]. The authors introduce a novel approach to estimate densities from an SDF representation by approximating it using a logistic function:

$$\phi_s(o) = (se^{-so})/(1 + e^{-so}), \quad (3)$$

where s is the logistic scale and o the SDF output. This conversion enables the application of camera-free volume rendering techniques for scene reconstruction. Using the SDF, the scene’s outer surface \mathcal{S} is represented as the zero-level set, defined as $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 : S(\mathbf{x}) = 0\}$, where $S(\cdot)$ is the output of the SDF network. The rendering is then computed using the SDF at a particular 3D point. The volume density at each point along the ray is

$$\alpha_i = \max\left(\frac{\Phi_s(S(\mathbf{x}_i)) - \Phi_s(S(\mathbf{x}_{i+1}))}{\Phi_s(S(\mathbf{x}_i))}, 0\right), \quad (4)$$

where $\Phi_s(\cdot)$ is the sigmoid function². The accumulated volume density is

$$w_i = \alpha_i T_i = \alpha_i \prod_{j=0}^i (1 - \alpha_j), \quad (5)$$

where T_i is the transmittance at the point i along the ray. See [50] for details.

¹Proof and more details given in the supplementary material.

² $\phi_s(\cdot)$ is the derivative of $\Phi_s(\cdot)$, hence $\Phi_s(\cdot)$ is the cumulative density function of the logistic distribution.

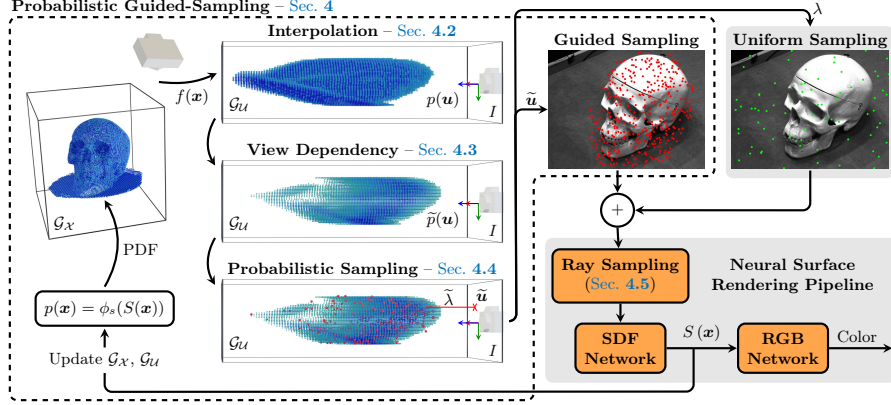


Fig. 2: Proposed guided-sampling: The scene is represented as a 3D grid \mathcal{G}_X , and characterized by a PDF $p(\mathbf{x})$ computed from the SDF network and modeled by a logistic distribution of the SDF values $\phi_s(S(\mathbf{x}))$. We propose to use a 3D image space that includes depth, represented as \mathcal{G}_U , where one can define $p(\mathbf{u})$ based on $p(\mathbf{x})$ as described in **Interpolation** – Sec. 4.2. Then, we consider the camera viewpoint of the scene (such as occlusions), by weighting $p(\mathbf{u})$ as described in **View Dependency** – Sec. 4.3. In the shown grids, color hue maps to the probability value, normalized for each grid. A higher hue is more probable. At every training step, points are sampled from $\tilde{p}(\mathbf{u})$ to create **ray samples** $\tilde{\mathbf{u}}$ (**Probabilistic Sampling** – Sec. 4.4). We sample **rays uniformly** to allow overall image quality and scene exploration.

4 Proposed Approach

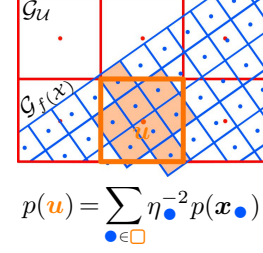
This work introduces a new probability-guided sampler for enhanced scene rendering and 3D reconstruction, seamlessly merging with neural surface pipelines.

4.1 Method Overview

Consider a typical neural surface rendering pipeline, such as the one proposed by NeuS [50]. An intermediate step of such methods consists of obtaining an SDF that models the 3D structure of the foreground of a scene (see **Neural Surface Rendering Pipeline** block in Fig. 2). In this work, we intend to utilize the SDF for more effective sampling while training the neural volume field. In particular, we intend to focus on the important regions of the scene, *i.e.* foreground, when training. Additionally, we leverage the output information from our sampling module to aid the sampling process along the rays and the training with additional surface reconstruction losses. Our method does not require any additional information (*e.g.* SfM points) or models. The proposed training pipeline is depicted in Fig. 2, **Probability-guided Sampler**.

We start by leveraging SDF representation in Eq. 3 to define a Probability Density Function (PDF) over the points in the 3D scene to capture the likelihood

Fig. 3: Bird’s-eye view illustration of the Riemann integral approximation of $p(\mathbf{u})$: This figure shows the 3D image space in **red**, the scene grid transformed to the image space in **blue** and how probability of $\mathbf{u} \in \mathcal{G}_{\mathcal{U}}$ (**orange**) is computed. A **projected scene point** $\bullet \in \mathcal{G}_{f(\mathcal{X})}$ that lies inside the **cell** \square will contribute to the computation of $p(\mathbf{u})$, as shown in the equation.



$$p(\mathbf{u}) = \sum_{\bullet \in \square} \eta_{\bullet}^{-2} p(\mathbf{x}_{\bullet})$$

of it being sampled during training, denoted as $p(\mathbf{x})$:

$$p(\mathbf{x}) = \phi_s(S(\mathbf{x})). \quad (6)$$

Then, we explore a suitable 3D image space from $p(\mathbf{x})$ for effective sampling in the camera’s viewpoint. To compute the probability in a 3D image space, the transformation has to be bijective and consequently invertible to account for the change of variables. From a geometric point of view, the proposed space \mathcal{U} is obtained from \mathcal{X} by transforming the projection rays, which, by definition, are parallel to each other and perpendicular to the image space (*i.e.* orthographic projection space). The new PDF is $p(\mathbf{u})$ and is described in [Sec. 4.2](#).

Next, we deal with the concerns arising from view dependency, such as occlusions. Rather than sampling directly in the image from the scene’s projection and probabilities, where awareness of the viewpoint is limited, we weigh the camera’s PDF $p(\mathbf{u})$ using a volume rendering strategy. This allows for seamless integration of view dependency constraints and provides the foundation for the sampling process. In [Sec. 4.3](#), this PDF is defined as $\tilde{p}(\mathbf{u})$.

The final step of our formulation consists of sampling 3D points on \mathcal{U} using $\tilde{p}(\mathbf{u})$. The proposed method follows a conditional sampling strategy detailed in [Sec. 4.4](#). [Sec. 4.5](#) describes how the sampled \mathbf{u} is used in the neural surface pipeline and details the proposed surface regularization losses designed to guide the training process by considering the sampled depth.

4.2 Interpolation

We aim to transform the density estimate $p(\mathbf{x})$, for a point $\mathbf{x} \in \mathcal{X}$ to the 3-dimensional image space, defined in [Eq. 1](#), which is denoted by $p(\mathbf{v})$, where $\mathbf{v} \in \mathcal{U}$. This transform is given by

$$p(\mathbf{x}) = p(f^{-1}(\mathbf{v})) \left| \frac{\partial f^{-1}(\mathbf{v})}{\partial \mathbf{v}} \right| \implies p(f^{-1}(\mathbf{v})) = \eta^{-2} p(\mathbf{x}), \quad (7)$$

where $\mathbf{v} = f(\mathbf{x})$ and η the depth value of \mathbf{v} .

To simplify and have a more compact representation, we discretize the 3D scene space \mathbf{x} , such that $\mathbf{x} \in \mathcal{G}_{\mathcal{X}} \subset \mathcal{X}$, and the 3D image space of \mathbf{u} such that $\mathbf{u} \in \mathcal{G}_{\mathcal{U}} \subset \mathcal{U}$, with $\mathcal{G}_{\mathcal{X}}$ and $\mathcal{G}_{\mathcal{U}}$ denoting a discretized grid on \mathcal{X} and \mathcal{U} . Note

that \mathbf{v} is not represented in the grid of the 3-dimensional image space $\mathcal{G}_{\mathcal{U}}$. Instead, \mathbf{v} is discretized according to the scene grid $\mathcal{G}_{\mathcal{X}}$ after applying the camera transformation $f(\cdot)$ in Eq. 2, which we define as $\mathcal{G}_{f(\mathcal{X})}$. However, the probability estimates $p(\mathbf{u})$ in $\mathcal{G}_{\mathcal{U}}$ cannot be easily interpolated from Eq. 7 due to the respective space deformation resulting from the discretization and transformation. Therefore, we approximate $p(\mathbf{u})$ as the Riemann integral of all transformed cells of $\mathcal{G}_{\mathcal{X}}$ in \mathbf{u} . We start by making sure $\mathcal{G}_{\mathcal{X}}$ is discretized finely³. For each $\mathbf{u} \in \mathcal{G}_{\mathcal{U}}$, the probability estimate is the sum of the probability densities of all points $\mathcal{G}_{f(\mathcal{X})}$ that lie inside the cell, as defined in Eq. 7. A depiction of the interpolation of $p(\mathbf{u})$ is shown in Fig. 3, where the selected transformed cells have an orange fill. Further details in supplementary material.

4.3 View Dependency

Since $p(\mathbf{u})$ does not account for occlusions created by the camera’s perspective projection, sampling a projection ray based on the object’s geometry alone can result in too many occluded samples and, consequently, loss of training efficiency. To address this issue, we assume that the volume density σ per cell is $p(\mathbf{u})$ as a naive solution discussed in Wang *et al.* [50]⁴. The transmittance T can then be evaluated in the 3-dimensional image space by accumulating the radiance weighted by the volume densities for cells along the ray, corresponding to the image coordinates $[u, v]$. Considering the grid $\mathcal{G}_{\mathcal{U}}$, the transmittance T_i at the depth λ_i corresponding to the i -th cell along $[u, v]$ can be defined as $T_i = e^{-\sum_{k=1}^i p([u, v, \lambda_k]^T)}$, where the k^{th} -cell is sorted by depth. Then, the view-dependent probability $\tilde{p}(\mathbf{u}_i)$ for $\mathbf{u}_i = [u, v, \lambda_i]^T$ is defined as the transmittance weighted by the volume density accumulated along a ray, as shown in Fig. 2,

$$\tilde{p}(\mathbf{u}_i) = \sigma_i T_i = p(\mathbf{u}_i) e^{-\sum_{k=0}^i p([u, v, \lambda_k]^T)}. \quad (8)$$

4.4 Probability-guided Sampling

While in previous neural rendering pipelines, pixels are sampled in the image uniformly, in this work, we combine the two sampling strategies: (i) sampling using the view-dependent space PDF $\tilde{p}(\mathbf{u})$, and (ii) sampling uniformly on the image. The former is better suited for the foreground and the latter for the background, allowing us to regulate the proportion of samples around the image.

Starting with the view-dependent space sampling, we use conditional probabilities, which extend ray importance sampling in [33] to the 3-dimensional space

³To make sure that $\mathcal{G}_{\mathcal{X}}$ is small enough, the scene $\mathcal{G}_{\mathcal{X}}$ is partitioned by a factor F , where each cell is divided at each axis into F equal parts, resulting in a total of F^3 equal cells from the 3 axes. The probability of the newly partitioned cell is the original cell probability divided by F^3 (see supplementary material for more details).

⁴Note that this density representation along the ray causes a bias in the depth estimate [50, 65]. Nonetheless, this formulation remains occlusion-aware.

\mathcal{U} . The first marginal density function is then defined as

$$\tilde{p}(u) = \frac{1}{R_v R_\lambda} \sum_{(v, \lambda)} \tilde{p}([u, v, \lambda]^T), \quad (9)$$

for all (v, λ) cells of u , where R_v and R_λ are resolutions of the grid $\mathcal{G}_\mathcal{U}$ along the axes of v and λ . Then, the first conditional distribution is computed as

$$\tilde{p}(v, \lambda|u) = \frac{\tilde{p}(\mathbf{u})}{\tilde{p}(u)}. \quad (10)$$

The second marginal applied to v can then be expressed as

$$\tilde{p}(v|u) = \frac{1}{R_\lambda} \sum_{\lambda} \tilde{p}(v, \lambda|u), \quad (11)$$

for all λ cells. Finally, the second conditional distribution is then defined as

$$\tilde{p}(\lambda|u, v) = \frac{\tilde{p}(v, \lambda|u)}{\tilde{p}(v|u)}. \quad (12)$$

With the marginals and conditionals defined, we sample $\tilde{\mathbf{u}} = [\tilde{u}, \tilde{v}, \tilde{\lambda}] \in \mathcal{U}$ in the 3-dimensional image space, to obtain the 3D projection ray and image pixels. We start by sampling \tilde{u} from the first marginal, Eq. 9, using inverse transform sampling [27]. Then, we approximate the second marginal $p(v|\tilde{u})$ from the samples \tilde{u} using bilinear interpolation. Following the same inverse sampling strategy, \tilde{v} is sampled according to $p(v|\tilde{u})$. Finally, using trilinear interpolation, we approximate the second conditional $p(\lambda|\tilde{u}, \tilde{v})$, with \tilde{u} and \tilde{v} , and sample $\tilde{\lambda}$.

When sampling uniformly along the rays during training and evaluation, we interpolate the second conditional, Eq. 12, with given values for \tilde{u} and \tilde{v} , and sample $\tilde{\lambda}$ directly. A representation of the sampled output is depicted in Fig. 2.

4.5 Surface Reconstruction Losses

The input to the rendering network (irrespective of what backbone is used) is the 3D points sampled along the rays from the sampled pixel obtained from $\tilde{\mathbf{u}}$. In addition to these obtained by following the sampling strategy of the backbone network, we provide additional 3D points along the ray near the sampled depth $\tilde{\lambda}$ for improved rendering of such regions, keeping the same sampling budget. This is accomplished by drawing samples from a Gaussian distribution $\sim \mathcal{N}(\tilde{\lambda}, \frac{\pi^2}{3s^2})$, where the variance is determined by the normal approximation of the logistic distribution [40], with the mean being the sampled $\tilde{\lambda}$.

In addition to each backbone, we introduce losses for points near the surface (near zero-level set), points within the empty ray space, and points belonging to background rays. Consider M projection rays and N_{fg} foreground points sampled along those rays. The proposed near-surface loss accounts for sampled points within 99.7% of the possible near-surface samples during ray sampling, *i.e.*,

points for the m -th ray, where $m \in \{1, \dots, M\}$, satisfying $\mathcal{N}_{\text{Near}} = \{f(\mathbf{x}) \in \mathcal{U} : f(\mathbf{x}) \in [(u, v, \tilde{\lambda} - 3\frac{\pi}{\sqrt{3}s}), (u, v, \tilde{\lambda} + 3\frac{\pi}{\sqrt{3}s})]\}$, and is given by

$$L_m^{\text{Near}} = \sum_{i \in \mathcal{N}_{\text{Near}}} |S(\mathbf{x}_i)| w_i, \quad (13)$$

where $S(\cdot)$ is the SDF value, and w_i represents the volume density accumulated along a ray of the point i , given by Eq. 5.

For points in the empty ray space, *i.e.*, the complement set of $\mathcal{N}_{\text{Near}}$, denoted as $\mathcal{N}_{\text{Empty}}$, we introduce a loss to encourage small SDF values and exploration:

$$L_m^{\text{Empty}} = \sum_{j \in \mathcal{N}_{\text{Empty}}} [(S(\mathbf{x}_j) - \epsilon) w_j]^2, \quad (14)$$

where ϵ is a small value. We consider view dependency in both losses by incorporating the accumulated volume densities, w_a , where $a \in \{i, j\}$.

Finally, for rays that do not intersect foreground surfaces, *i.e.*, if the sampled depth $\tilde{\lambda}$ is outside of the scene’s boundary, the following background loss ensures that the importance of accurately estimating the scene geometry decreases as one moves farther from the surface:

$$L_m^{\text{Bg}} = \sum_{k=1}^{N_{\text{fg}}} e^{-\beta |S(\mathbf{x}_i)|} w_k. \quad (15)$$

All losses are averaged by over M rays. The total surface loss is computed as $L^{\text{Surf}} = \lambda_1 L^{\text{Near}} + \lambda_2 (L^{\text{Empty}} + L^{\text{Bg}})$. This surface loss is appropriately weighted and added to the existing losses for each backbone.

5 Experiments

5.1 Experimental Setup

We use NeuS [50] and Neuralangelo [21] as backbones to evaluate the effectiveness of our proposed approach. Specifically, we use Neuralangelo’s official implementation⁵ with the default settings and implemented NeuS within Neuralangelo’s framework. Also, we use the default settings for both models when evaluating the proposed probabilistic sampling. With respect to resolutions of the scene space and camera spaces, we use 128 for the 3 dimensions of $\mathcal{G}_{\mathcal{X}}$, and $R_u = R_v = 64$ and $R_\lambda = 128$, for the 3D image space $\mathcal{G}_{\mathcal{U}}$. We set $\lambda_1 = \lambda_2 = 0.5$. We use the scene scale and centers provided with the dataset to define each scene boundary. For each camera, we utilize the image corners to obtain \mathcal{U} boundaries. The depth boundary is computed by shooting a ray from the central image pixel and computing where the ray hits the scene’s boundaries. During training, we update the 3D image space every 2500 and 5000 iterations for Neuralangelo

⁵<https://github.com/NVlabs/neuralangelo>

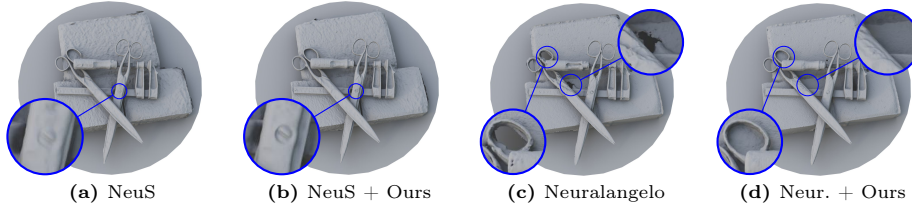


Fig. 4: DTU scan 37: When our sampling and surface reconstruction losses are included in NeuS and Neuralangelo backbones, we get a sharper 3D reconstruction of the foreground objects. Our approach also removes the hole obtained by Neuralangelo.

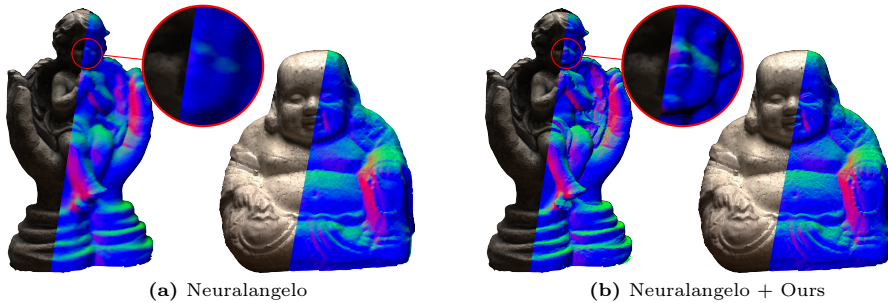


Fig. 5: DTU scans 118 and 114: We can extract more detailed meshes with our sampler and losses. The synthesized normal images are less noisy, with sharper edges in both scans, leading to better reconstruction. Image quality remains similar.

and NeuS, respectively. For a fair comparison, we use the same number of rays and ray points as the baselines. We initialize the camera grids as a sphere [59], facilitating the initial scene exploration. No depth ground-truth is used to supervise or evaluate the model. At the start of training, we set 20% of the rays to be sampled uniformly in the image. As training progresses, the percentage increases to 40%, 60%, and 80%. In the rendering pipeline, we sample 32 points around the sampled depth, as described in Sec. 4.5. We train all backbones with their default losses, with L^{Surf} being assigned a weight of 500. We train and test all models in an A40 GPU using 10 CPU cores.

Datasets: We use DTU [15], as the primary dataset, which has 15 object-focused sequences in a controlled environment, with object masks available to evaluate intricate details. Each DTU scan has 49 or 64 views each. We also evaluate more diverse sequences in BMVS [56] dataset, where we choose 5 object-centric sequences and 4 large-scale sequences from Tanks and Temples (TNT) [17]. The last two datasets have around 200 images in each sequence.

Evaluation Metrics: For image synthesis, we use the Peak Signal-to-Noise Ratio (PSNR). Following the evaluation protocols of prior work, we use NeuralWarp’s evaluation methodology [7] for 3D reconstruction and report the Chamfer distance [15] in DTU and F1-score [17] in TNT. Note that we evaluate the models

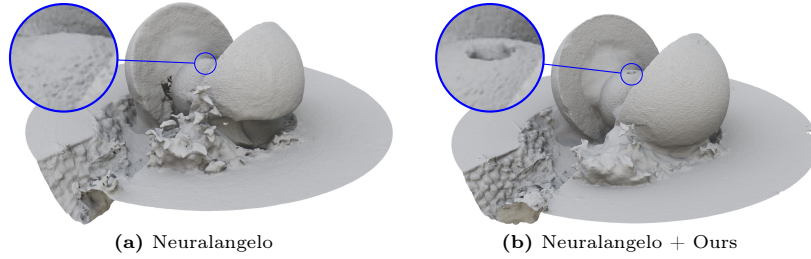


Fig. 6: BMVS sequence "Sphere": We observe that Neuralangelo + Ours get a significantly more complete 3D surface reconstruction. Also, our sampling better captures small and intricate regions, such as the small hole highlighted in the object foreground.



Fig. 7: BMVS sequence "Bandstand": Our sampling and surface reconstruction losses have clear advantages. While NeuS fails to capture the foreground surfaces, the proposed sampling obtains a reasonable 3D structure and high-quality images.

with object masks, when available, to measure the effectiveness of reconstructing the foreground. However, we do not use them during training.

Baselines: Our primary goal is to evaluate the impact of the probabilistic guided ray sampling by comparing it against approaches that do not use it. Towards this end, the main baselines are NeuS [50]⁶ and Neuralangelo [21], where we augment the proposed sampling strategy. For assessing the effectiveness of image synthesis and reconstruction quantitatively, we compare against existing literature: VolSDF [57], RegSDF [61], NeuralWarp [7], and HF-NeuS [52].

5.2 Results

We discuss qualitative and quantitative results on the proposed sampler. We also comment on the ablation study and the computational overhead.

Qualitative results: We show a qualitative comparison of the DTU and BMVS datasets. When observing the 3D reconstruction, the proposed approach preserves finer details, has fewer artifacts, increased completeness, and fewer holes

⁶Our implementation within Neuralangelo framework.

Fig. 8: Reconstruction Error and Variance over training. Our sampling converges faster to a better 3D representation (solid line), while significantly reducing the variance of the logistic output density (dashed).

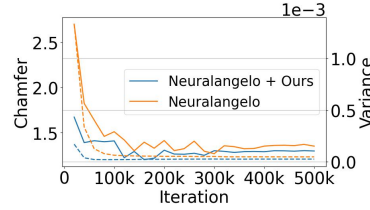


Table 1: Quantitative results on DTU [15]. We highlight the **best** result for each backbone method with and without the proposed sampler. [†] Requires SfM points.

| | | 24 | 37 | 40 | 55 | 63 | 65 | 69 | 83 | 97 | 105 | 106 | 110 | 114 | 118 | 122 | Mean | |
|---------------------------|----------|--------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|
| PSNR \uparrow | Unmasked | NeRF [26] | 26.24 | 25.74 | 26.79 | 27.57 | 31.96 | 31.50 | 29.58 | 32.78 | 28.35 | 32.08 | 33.49 | 31.54 | 31.00 | 35.59 | 35.51 | 30.65 |
| | | VolSDF [57] | 26.28 | 25.61 | 26.55 | 26.76 | 31.57 | 31.50 | 29.38 | 33.23 | 28.03 | 32.13 | 33.16 | 31.49 | 30.33 | 34.90 | 34.75 | 30.38 |
| | | RegSDF † [61] | 24.78 | 23.06 | 23.47 | 22.21 | 28.57 | 25.53 | 21.81 | 28.89 | 26.81 | 27.91 | 24.71 | 25.13 | 26.84 | 21.67 | 28.25 | 25.31 |
| | | NeuS [50] | 23.85 | 27.63 | 27.16 | 29.4 | 32.71 | 33.1 | 30.58 | 34.25 | 29.97 | 33.69 | 35.34 | 32.81 | 31.96 | 36.72 | 37 | 31.74 |
| | | NeuS + Ours | 28.28 | 28.1 | 28.16 | 24.71 | 33.1 | 33.97 | 29.59 | 33.25 | 30.35 | 33.61 | 35.66 | 32.97 | 32.29 | 37.15 | 35.56 | 31.78 |
| | Masked | Neuralangelo [21] | 30.64 | 27.78 | 32.70 | 34.18 | 35.15 | 35.89 | 31.47 | 36.82 | 30.13 | 35.92 | 36.61 | 32.60 | 31.20 | 38.41 | 38.05 | 33.84 |
| | | Neuralangelo + Ours | 33.73 | 30.36 | 33.55 | 34.06 | 35.22 | 34.64 | 32.49 | 33.2 | 31.93 | 34.17 | 37.64 | 35.3 | 34.01 | 38.04 | 37.87 | 34.41 |
| | | NeuS [50] | 28.93 | 28.29 | 27.53 | 30.57 | 36.48 | 36.48 | 31.83 | 40.59 | 31.26 | 37.19 | 36.87 | 33.9 | 32.65 | 39.63 | 40.88 | 34.21 |
| | | NeuS + Ours | 28.98 | 29.22 | 28.66 | 25.22 | 37.24 | 38.73 | 30.77 | 42.47 | 32.34 | 37.5 | 37.49 | 34.34 | 33.11 | 40.84 | 38.45 | 34.36 |
| | | Neuralangelo [21] | 35.21 | 31.76 | 35.12 | 38.16 | 41.17 | 40.46 | 34.39 | 44.22 | 34.09 | 40.8 | 40.8 | 37.24 | 34.92 | 42.36 | 43.56 | 38.28 |
| Neuralangelo + Ours | 35.13 | 32.86 | 35.2 | 38.51 | 41.41 | 41 | 34.51 | 44.93 | 35.64 | 41.13 | 40.95 | 37.78 | 35.26 | 43.3 | 44.59 | 38.81 | | |
| Chamfer (mm) \downarrow | Unmasked | NeRF [26] | 1.90 | 1.60 | 1.85 | 0.58 | 2.28 | 1.27 | 1.47 | 1.67 | 2.05 | 1.07 | 0.88 | 2.53 | 1.06 | 1.15 | 0.96 | 1.49 |
| | | VolSDF [57] | 1.14 | 1.26 | 0.81 | 0.49 | 1.25 | 0.70 | 0.72 | 1.29 | 1.18 | 0.70 | 0.66 | 1.08 | 0.42 | 0.61 | 0.55 | 0.86 |
| | | HF-NeuS [52] | 0.76 | 1.32 | 0.70 | 0.39 | 1.06 | 0.63 | 0.63 | 1.15 | 1.12 | 0.80 | 0.52 | 1.22 | 0.33 | 0.49 | 0.50 | 0.77 |
| | | RegSDF † [61] | 0.60 | 1.41 | 0.64 | 0.43 | 1.34 | 0.62 | 0.60 | 0.90 | 0.92 | 1.02 | 0.60 | 0.59 | 0.30 | 0.41 | 0.39 | 0.72 |
| | | NeuralWarp [7] | 0.49 | 0.71 | 0.38 | 0.38 | 0.79 | 0.81 | 0.82 | 1.20 | 1.06 | 0.68 | 0.66 | 0.74 | 0.41 | 0.63 | 0.51 | 0.68 |
| | Masked | NeuS [50] | 0.77 | 0.78 | 5.82 | 0.50 | 1.39 | 1.76 | 1.06 | 4.01 | 1.47 | 0.77 | 0.64 | 1.29 | 0.34 | 0.56 | 0.53 | 1.30 |
| | | NeuS + Ours | 1.08 | 0.74 | 1.27 | 2.43 | 1.05 | 1.05 | 1.66 | 1.32 | 2.1 | 0.79 | 0.6 | 1.07 | 0.32 | 0.4 | 2.08 | 1.2 |
| | | Neuralangelo [21] | 0.37 | 0.72 | 0.35 | 0.35 | 0.87 | 0.54 | 0.53 | 1.29 | 0.97 | 0.73 | 0.47 | 0.74 | 0.32 | 0.41 | 0.43 | 0.61 |
| | | Neuralangelo + Ours | 0.39 | 0.68 | 0.32 | 0.33 | 0.87 | 0.58 | 0.53 | 1.3 | 0.93 | 0.70 | 0.5 | 0.74 | 0.31 | 0.37 | 0.38 | 0.6 |

as illustrated in Figs. 1 and 4 to 6. Even in complex scenes, the sampler reduces failure cases, such as in Fig. 7. Guiding the pixel sampling towards surface areas enhances 3D consistency and subsequently improves sharpness while displaying faster convergence and lower model variance as shown in Fig. 8. See supplementary material for more qualitative results, including on TNT.

Quantitative results: For image synthesis on the DTU and the TNT datasets, we assess the proposed method through two distinct analyses: evaluating the image quality and examining the quality in masked regions, given our focus on sampling areas with surfaces. Results are shown in Tab. 1. We observe that adding the proposed sampling improves image quality, outperforming NeuS and Neuralangelo by a mean of $0.04dB$ and $0.57dB$ on PSNR, respectively. Moreover, when only regions of interest are evaluated, we observe a higher improvement over NeuS ($0.15dB$) and about the same gain for Neuralangelo. 3D reconstruction performance on the DTU and TNT datasets is shown in Tabs. 1 and 2. We notice that even while relying on SfM priors, RegSDF achieves worse reconstruction results while we improve against the backbones. In particular, we achieve state-of-the-art performance with our sampling strategy coupled with Neuralangelo.

Ablation study: We ablate our sampling strategy using DTU. We follow previous methods and use a subset of DTU (24, 65, 97, and 122) to ablate the

Table 2: Quantative results for TNT: Small improvement in 3D from a decrease in image quality.

| | PSNR \uparrow | F1 \uparrow |
|---------------------|-----------------|---------------|
| Neuralangelo | 25.99 | 0.58 |
| Neuralangelo + Ours | 25.65 | 0.59 |

Table 3: Ablations: L–surface loss weight; VD–view dependency; RU–uniform ray sampling; FS–fixed s probability update.

| Ablation | L | VD | RU | FS | PSNR \uparrow | Chamfer \downarrow |
|-----------|-----|----|----|----|-----------------|----------------------|
| A1 | 0 | ✓ | ✗ | ✗ | 35.44 | 0.82 |
| A2 | 500 | ✓ | ✗ | ✗ | 35.24 | 0.57 |
| A3 | 500 | ✗ | ✗ | ✗ | 34.91 | 1.04 |
| A4 | 500 | ✓ | ✓ | ✗ | 34.82 | 0.58 |
| A5 | 500 | ✓ | ✗ | ✓ | 35.22 | 0.58 |

method. Results are shown in [Tab. 3](#). The ablations **A1vsA2** show that surface losses substantially improve 3D reconstruction while slightly decreasing image quality. Ablations **A2vsA3** assess the impact of view dependency, where focusing the sampling on occluded areas introduces uncertainty, diminishing both image quality and 3D reconstruction accuracy. **A2vsA4** illustrates the impact of additional ray samples near the surface. These samples improve image quality but have minimal effect on 3D reconstruction. Ablations **A2vsA5** examines a constant value for s (logistic scale in [Eq. 3](#)) in probability updates. When using a constant low-value for s , the 3D image space sampling produces high-variance samples, preventing refinement and introducing uncertainty in the model.

Computational overhead: Evaluation overhead is insignificant. The 3D reconstruction does not change since grid points are directly evaluated in the implicit surface network. For image synthesis, only the λ -axis is sampled since u and v are known (all pixels in the image). The overhead during evaluation is negligible (average +0.7%) when compared between Neuralangelo with and without our sampling in DTU. Training overhead comes from interpolating the camera weights and interpolating and sampling each axis for all cameras in the batch, which amounts to an additional 10% of training time with our implementation.

6 Discussion

In this work, we explore the scene’s implicit surface representation to define a novel sampling strategy for training implicit neural surface fields more effectively. We propose a view-dependent and occlusion-aware 3D orthographic projection space, where we conditionally sample the image coordinates and depth for each camera. We utilize this depth to regularize the scene’s surface during training. Our strategy can be coupled with typical image sampling on neural surface pipelines and does not depend on specific backbones. Experiments show that our proposed sampling strategy improves both image synthesis and 3D reconstruction, preserving the foreground details of the scene.

Future work involves extending our approach to incorporate other sources of information, such as semantics or point clouds, to reduce in-ray uncertainties.

Acknowledgments

Pais and Piedade were partially supported by LARSyS, Portuguese “Fundação para a Ciência e a Tecnologia” (FCT) funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIBP/50009/2020, and 10.54499/UIBD/50009/2020). Pais was also partially supported by FCT grant PD/BD/150630/2020.

References

1. Attal, B., Huang, J.B., Richardt, C., Zollhoefer, M., Kopf, J., O’Toole, M., Kim, C.: Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 16610–16620 (2023)
2. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 6290–6301 (2022)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: IEEE/CVF Int’l Conf. Computer Vision (ICCV). pp. 5855–5864 (2021)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 5470–5479 (2022)
5. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (2023)
6. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. IEEE Trans. Robotics (T-RO) **37**(6), 1874–1890 (2021)
7. Darmon, F., Bascle, B., Devaux, J.C., Monasse, P., Aubry, M.: Improving neural implicit surfaces geometry with patch warping. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 6260–6269 (2022)
8. Davison: Real-time simultaneous localisation and mapping with a single camera. In: IEEE Int’l Conf. Computer Vision (ICCV). pp. 1403–1410 (2003)
9. Deng, J., Wu, Q., Chen, X., Xia, S., Sun, Z., Liu, G., Yu, W., Pei, L.: Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In: IEEE/CVF Int’l Conf. Computer Vision (ICCV). pp. 8218–8227 (2023)
10. Fu, Q., Xu, Q., Ong, Y.S., Tao, W.: Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. Advances in Neural Information Processing Systems (NeurIPS) **35**, 3403–3416 (2022)
11. Gaur, A., Pais, G.D., Miraldo, P.: Oriented-grid encoder for 3d implicit representations. In: Int’l Conf. 3D Vision (3DV) (2024)
12. Geppert, M., Larsson, V., Speciale, P., Schönberger, J.L., Pollefeys, M.: Privacy preserving structure-from-motion. In: European Conf. Computer Vision (ECCV). pp. 333–350 (2020)
13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, 2 edn. (2004)
14. Hu, W., Wang, Y., Ma, L., Yang, B., Gao, L., Liu, X., Ma, Y.: Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In: IEEE/CVF Int’l Conf. Computer Vision (ICCV). pp. 19774–19783 (2023)

15. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanaes, H.: Large scale multi-view stereopsis evaluation. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR). pp. 406–413 (2014)
16. Johnson, E., Habermann, M., Shimada, S., Golyanik, V., Theobalt, C.: Unbiased 4d: Monocular 4d reconstruction with a neural deformation model. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 6597–6606 (2023)
17. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (TOG)* **36**(4) (2017)
18. Kong, X., Liu, S., Taher, M., Davison, A.J.: vmap: Vectorised object mapping for neural field slam. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 952–961 (2023)
19. Kurz, A., Neff, T., Lv, Z., Zollhöfer, M., Steinberger, M.: Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. In: European Conf. Computer Vision (ECCV). pp. 254–270 (2022)
20. Li, T., Slavcheva, M., Zollhöfer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., Lv, Z.: Neural 3d video synthesis from multi-view video. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 5521–5531 (2022)
21. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 8456–8465 (2023)
22. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 6498–6508 (2021)
23. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. *Advances in Neural Information Processing Systems (NeurIPS)* **33**, 15651–15663 (2020)
24. Liu, X., Tai, Y.w., Tang, C.K., Miraldo, P., Lohit, S., Chatterjee, M.: Gear-nerf: Free-viewpoint rendering and tracking with motion-aware spatio-temporal sampling. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (2024)
25. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 4460–4470 (2019)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conf. Computer Vision (ECCV) (2020)
27. Mosegaard, K., Tarantola, A.: Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth* **100**(B7), 12431–12447 (1995)
28. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)* **41**(4), 1–15 (2022)
29. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (2020)
30. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: IEEE/CVF Int’l Conf. Computer Vision (ICCV). pp. 5589–5599 (2021)

31. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: IEEE/CVF Int'l Conf. Computer Vision (ICCV). pp. 5865–5874 (2021)
32. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: European Conf. Computer Vision (ECCV). pp. 523–540 (2020)
33. Pharr, M., Jakob, W., Humphreys, G.: Physically based rendering: From theory to implementation. MIT Press (2023)
34. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 10318–10327 (2021)
35. Saito, S., Simon, T., Saragih, J., Joo, H.: Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 84–93 (2020)
36. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: IEEE/CVF Int'l Conf. Computer Vision (ICCV). pp. 18433–18444 (2023)
37. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (2016)
38. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conf. Computer Vision (ECCV). pp. 501–518 (2016)
39. Shen, J., Agudo, A., Moreno-Noguer, F., Ruiz, A.: Conditional-flow nerf: Accurate 3d modelling with reliable uncertainty quantification. In: European Conf. Computer Vision (ECCV). pp. 540–557 (2022)
40. Stefanski, L.A.: A normal scale mixture representation of the logistic distribution. *Statistics & Probability Letters* **11**(1), 69–70 (1991)
41. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: IEEE/CVF Int'l Conf. Computer Vision (ICCV). pp. 6229–6238 (2021)
42. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 5459–5469 (2022)
43. Sun, J., Chen, X., Wang, Q., Li, Z., Averbuch-Elor, H., Zhou, X., Snavely, N.: Neural 3d reconstruction in the wild. In: ACM SIGGRAPH (2022)
44. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 15598–15607 (2021)
45. Sun, S., Liu, M., Fan, Z., Jiao, Q., Liu, Y., Dong, L., Kong, L.: Efficient ray sampling for radiance fields reconstruction. *Computers & Graphics* **118**, 48–59 (2024)
46. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 8248–8258 (2022)
47. Treitsch, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: IEEE/CVF Int'l Conf. Computer Vision (ICCV). pp. 12959–12970 (2021)

48. Turki, H., Ramanan, D., Satyanarayanan, M.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 12922–12931 (2022)
49. Wang, J., Wang, P., Long, X., Theobalt, C., Komura, T., Liu, L., Wang, W.: Neuris: Neural reconstruction of indoor scenes using normal priors. In: European Conf. Computer Vision (ECCV). pp. 139–155 (2022)
50. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
51. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: IEEE/CVF Int’l Conf. Computer Vision (ICCV). pp. 3295–3306 (2023)
52. Wang, Y., Skorokhodov, I., Wonka, P.: Hf-neus: Improved surface reconstruction using high-frequency details. Advances in Neural Information Processing Systems (NeurIPS) **35**, 1966–1978 (2022)
53. Wei, X., Zhang, Y., Li, Z., Fu, Y., Xue, X.: Deepsfm: Structure from motion via deep bundle adjustment. In: European Conf. Computer Vision (ECCV). pp. 230–247 (2020)
54. Wen, B., Tremblay, J., Blukis, V., Tyree, S., Müller, T., Evans, A., Fox, D., Kautz, J., Birchfield, S.: Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 606–617 (2023)
55. Wu, J., Liu, L., Tan, Y., Jia, Q., Zhang, H., Zhang, X.: Actray: Online active ray sampling for radiance fields. In: ACM SIGGRAPH Asia. pp. 1–10 (2023)
56. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 1790–1799 (2020)
57. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems (NeurIPS) **34**, 4805–4815 (2021)
58. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B.: Bakedsf: Meshing neural sdfs for real-time view synthesis. In: ACM SIGGRAPH (2023)
59. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems (NeurIPS) **33**, 2492–2502 (2020)
60. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in Neural Information Processing Systems (NeurIPS) **35**, 25018–25032 (2022)
61. Zhang, J., Yao, Y., Li, S., Fang, T., McKinnon, D., Tsin, Y., Quan, L.: Critical regularizations for neural surface reconstruction in the wild. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 6270–6279 (2022)
62. Zhang, J., Yao, Y., Quan, L.: Learning signed distance field for multi-view surface reconstruction. In: IEEE/CVF Int’l Conf. Computer Vision (ICCV). pp. 6525–6534 (2021)
63. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv:2010.07492 (2020)

- 64. Zhang, X., Bi, S., Sunkavalli, K., Su, H., Xu, Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 5449–5458 (2022)
- 65. Zhang, Y., Hu, Z., Wu, H., Zhao, M., Li, L., Zou, Z., Fan, C.: Towards unbiased volume rendering of neural implicit surfaces with geometry priors. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 4359–4368 (2023)
- 66. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 12786–12796 (2022)