Sphere-Guided Training of Neural Implicit Surfaces

Andreea Dogaru[1,2] Andrei Timotei Ardelean[1,2] Savva Ignatyev[1]
Egor Zakharov[1] Evgeny Burnaev[1,3]

[1]Skolkovo Institute of Science and Technology [2]Friedrich-Alexander-Universität Erlangen-Nürnberg
[3]Artificial Intelligence Research Institute

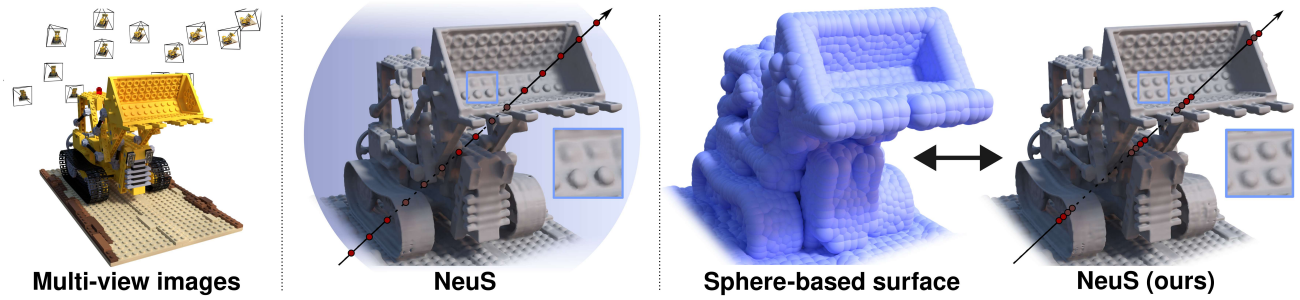


Figure 1. We propose a new hybrid approach for learning neural implicit surfaces from multi-view images. In previous methods, the volumetric ray marching training procedure is applied for the whole bounding sphere of the scene (middle left). Instead, we train a coarse sphere-based surface reconstruction (middle right) alongside the neural surface to guide the ray sampling and ray marching. As a result, our method achieves an increased sampling efficiency by pruning empty scene space and better quality of reconstructions (right).

Abstract

In recent years, neural distance functions trained via volumetric ray marching have been widely adopted for multi-view 3D reconstruction. These methods, however, apply the ray marching procedure for the entire scene volume, leading to reduced sampling efficiency and, as a result, lower reconstruction quality in the areas of high-frequency details. In this work, we address this problem via joint training of the implicit function and our new coarse sphere-based surface reconstruction. We use the coarse representation to efficiently exclude the empty volume of the scene from the volumetric ray marching procedure without additional forward passes of the neural surface network, which leads to an increased fidelity of the reconstructions compared to the base systems. We evaluate our approach by incorporating it into the training procedures of several implicit surface modeling methods and observe uniform improvements across both synthetic and real-world datasets. Our codebase can be accessed via the project page[†].

[†] https://andreeadogaru.github.io/SphereGuided

1. Introduction

The task of multi-view 3D reconstruction remains the focus of modern computer vision and graphics research. It has major practical significance in AR/VR metaverses, synthetic media, medical imaging, and the special effects industry. This task is classically addressed via the multi-view stereo (MVS) reconstruction systems [3, 4, 6, 9, 10, 26, 38], which estimate the underlying scene geometry in the form of a point cloud using a photometric consistency between the different views. However, in recent years they have been largely phased out by the methods that represent the scene as neural implicit fields [5, 14, 16–19, 22, 29, 31–33, 35–37]. These approaches have multiple advantages compared to the classical MVS. For example, they can easily accommodate non-Lambertian and texture-less surfaces [8], are good at interpolating unseen parts of the geometry by leveraging regularization [12], and at the same time can achieve an impressive quality of renders [2].

This work focuses on improving the subset of such methods specialized in opaque surface reconstruction [5, 22, 31, 35]. Most of these approaches employ neural signed distance fields [23] (SDFs) trained using volumetric ray marching [5, 31, 35]. The training step of this procedure contains

two stochastic elements: sampling a *ray* corresponding to a training pixel and sampling a set of *points* along the ray to approximate the color integral. The sampling efficiency at these steps largely determines the resulting quality of the reconstructions. While in the abovementioned methods the training rays are selected uniformly within the scene volume, their point sampling procedure typically employs a multi-stage importance [31] or uncertainty [5, 35] sampling to improve the accuracy of the reconstructions.

At the same time, it was shown [32] that neural signed distance fields benefit from the surface-based sampling of rays for surface rendering methods, such as IDR [36], which the modern multi-view reconstruction systems do not incorporate. Additionally, some of the novel-view synthesis works [16, 18, 37] successfully combined a simple two-stage coarse-to-fine sampling with explicitly defined surface guides to achieve a better rendering quality, as opposed to using the sophisticated multi-stage sampling procedures of the surface reconstruction methods. To guide the ray marching, they use explicit coarse surface approximations in the form of a set of volumetric primitives [18] or sparse octrees [16, 37]. However, these methods require a complete scene reconstruction to fit such an approximation [18, 37] or employ a heuristic optimization procedure [16] which we show performs poorly for the surface reconstruction task.

Inspired by these approaches, we improve the existing surface reconstruction methods' ray sampling and marching procedures using explicitly defined coarse representations. We propose training a coarse reconstruction as a sphere cloud which guides both sampling steps during volume rendering. We also propose a new optimization approach for coarse reconstruction based on gradient descent, which allows us to train it alongside the implicit surface field. Additionally, we introduce a point resampling scheme, which prevents the spheres from getting stuck in the local minima, and a repulsion mechanism that ensures high degrees of exploration of the reconstructed surface. Finally, we provide empirical evidence of the proposed method's applicability to different approaches for implicit surface modeling. Specifically, we pair our method with several modern systems [5, 22, 31, 35] for surface reconstruction and observe uniform improvements in the resulting quality across multiple 3D reconstruction benchmarks.

2. Related works

Implicit volumetric representations. Neural implicit representations have gained much attention in recent years for the problem of multi-view 3D surface reconstruction. Their widespread adoption started after several works have introduced training approaches based on the differentiable rendering of implicit functions. They initially relied on the surface rendering [21, 28, 36] procedure, where the pixel's color is approximated using the radiance of a single point in the volume. However, they were recently phased out by the training procedures based on the volume rendering with multiple samples via ray marching. Introduced in a seminal work on novel view synthesis, Neural Radiance Fields (NeRFs) [19], the volumetric ray marching has been later adapted [22, 31, 35] to the problem of surface modeling since it significantly improved the reconstruction quality. The ray marching procedure estimates the color along the ray using the volume rendering integral, approximated as a sum of the weighted radiances at multiple points throughout the volume. The aforementioned works employ methods based on importance [31], uncertainty [35] or surface intersection-based [22] sampling to obtain this set of points, increasing the approximation accuracy compared to more simple strategies, such as uniform sampling.

We propose a new hybrid surface representation that improves ray marching by limiting the sampling space to a volume coarsely bounding the scene. This is used in conjunction with the ray marching mechanism of the base neural reconstruction method which further optimizes the selection of samples around the reconstructed surface. We also use this hybrid surface representation to guide the sampling of the training rays, improving the quality of reconstructions given the same training time.

Hybrid representations. To improve the training efficiency and rendering frame rate, multiple hybrid representations [1, 7, 17, 20, 24, 25, 37] have been proposed, which jointly optimize the implicit and explicit representations. These methods employ point clouds [1, 24, 25], hash tables [20], sparse voxel grids [7, 17, 37], and volumetric primitives [14] to improve both the training and rendering procedures in terms of either their speed or the resulting quality. Below we discuss the methods that are most closely related to our approach.

Iso-Points [32] introduced joint optimization of the signed distance functions with a point-based surface representation. In our approach, we use a *sphere-based* representation, which allows us to sample both the rays *and* points along these rays to lie near the surface, thus modifying both the ray-sampling and the ray-marching procedures. Closely related to our work are Neural Sparse Voxel Fields [16] and Neural 3D Reconstruction in the Wild [30] systems. They both employ sparse voxel grids to guide the ray marching. However, the method in [16] uses a greedy optimization strategy to train these representations, which, as we show, results in an inferior reconstruction quality compared to our gradient-based training. Compared to [30], our method does not employ the initialization using a sparse point-cloud, and it trains the guiding reconstruction from scratch.

3. Method

Our approach addresses a multi-view 3D reconstruction problem. The goal is to estimate the surface of a scene, de-

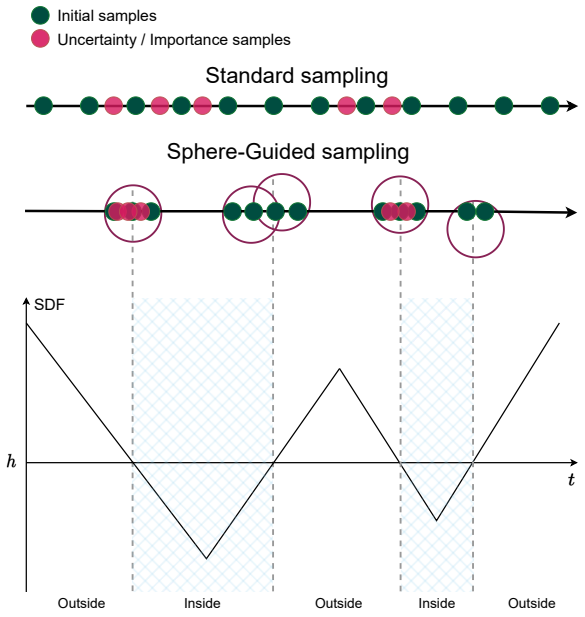


Figure 2. Our method works by filtering the samples along the ray that lie outside of the surface region, approximated by a trainable sphere cloud. Such filtering improves the sample efficiency in the optimization process and allows the implicit function to converge to a better optimum.

noted as $\mathcal{S}$, given a collection of images with the associated camera parameters. In our case, this surface is extracted as a level set of the learned implicit representations: either a signed distance function (SDF) or an occupancy field. In this section, we begin by describing the volume rendering approach utilized by most state-of-the-art methods. Then, we show how a learnable sphere cloud $\mathbf{S}$ could be used to improve the volume rendering-based training process and finally describe the optimization pipeline for the sphere cloud itself.

3.1. Volume rendering

We assume the underlying implicit model f to represent the geometry of the surface, and that there is a transformation that maps it to a surface density function $\sigma : \mathbb{R}^3 \to \mathbb{R}^+$, defined at each point $\mathbf{x}$ in the volume.

In order to render the surface defined by σ via volumetric rendering, we first need to consider a ray $\mathbf{p}(t) = \mathbf{o} + t\mathbf{v}$, $t \geq 0$, emanated from the camera origin $\mathbf{o} \in \mathbb{R}^3$ in the direction $\mathbf{v} \in \mathbb{S}^2$, and the corresponding color $C(\mathbf{o}, \mathbf{v})$ of the pixel on the image plane of that camera. We also need to define a radiance field $c : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3$, which produces a view-dependent color at each point in the volume. The observed color $C(\mathbf{o}, \mathbf{v})$ can then be expressed as the following integral along the ray:

$$C(\mathbf{o}, \mathbf{v}) = \int_0^{+\infty} w(t)c(\mathbf{p}(t), \mathbf{v})dt, \tag{1}$$

where $w(t)$ is the probability of a ray terminating at $\mathbf{p}(t)$, which can be derived from the density σ:

$$w(t) = T(t)\sigma(t), \quad T(t) = \exp\left(-\int_0^t \sigma(s)ds\right). \tag{2}$$

In practice, the color integral is approximated by evaluating the density and radiance at a set of n sampled points $\mathcal{P} = \{\mathbf{p}_i = \mathbf{o} + t_i\mathbf{v}\}_{i=1}^n$ using the discretized version [19] of the equations above:

$$\hat{C}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{N} T_i \alpha_i c_i, \quad T_i = \prod_{j=1}^{i-1}(1 - \alpha_j). \tag{3}$$

Here T_i denotes the accumulated transmittance and α_i — the opacity value at point $\mathbf{p}_i$, which can also be estimated from the density function via the following formula:

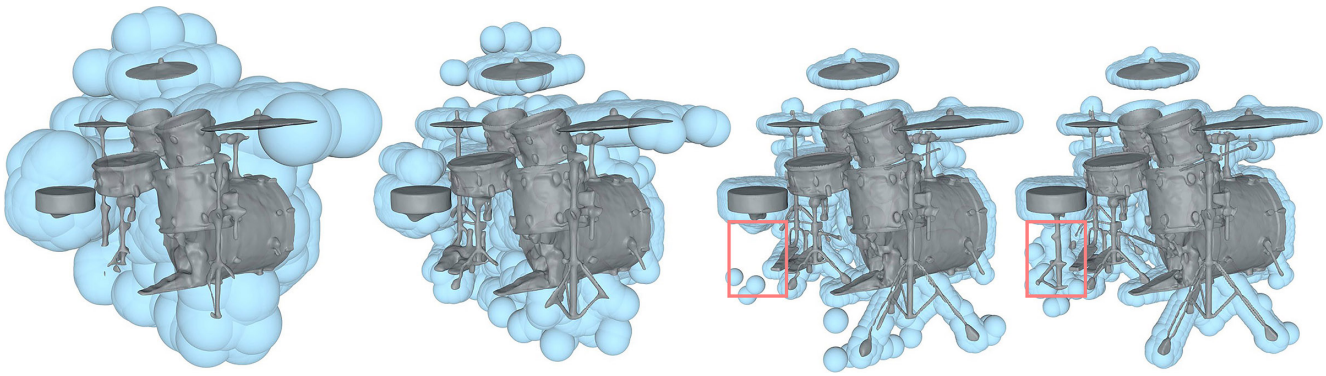
$$\alpha_i = 1 - \exp\left(-\int_{t_i}^{t_{i+1}} \sigma(t)dt\right). \tag{4}$$

3.2. Sphere-guided volume rendering

The sampling strategy for the points $\mathcal{P}$ has a major impact on the resulting reconstructions since it directly affects the approximation quality of eq. 1. To improve it, some methods [22] employ the root-finding procedure to obtain the first intersection with the surface along the ray and sample more points near it. Other methods [5, 31, 35] are first estimating a dense set of proposals $\mathcal{T}$ via importance or uncertainty sampling. Then, $\mathcal{P}$ is obtained either via inverse transform sampling by evaluating the density σ at the proposals $\mathcal{T}$ and normalizing it along the ray [5,35], or in some cases by using an entire set of proposals [31].

Algorithm 1: Sphere-guided sampling.

Input: ray $\mathbf{p}(t)$, spheres $\mathbf{S}$, #samples n
Initialize a set of intervals $\mathcal{I} = \varnothing$
for $\mathbf{S}_i \in \mathbf{S}$ **do**
  Add sphere-ray intersection to $\mathcal{I}$:
  $\mathcal{I} := \mathcal{I} \cup \mathbf{S}_i \cap \mathbf{p}(t)$
end
Find a minimal set of intervals
  $\{[s_k, t_k]\} : \bigcup_k [\mathbf{p}(s_k), \mathbf{p}(t_k)] = \mathcal{I}$
Initialize a set of points $\mathcal{T}_0 = \varnothing$
for $k = 1 \dots K$ **do**
  Set $n_k := \lfloor n/(t_k - s_k) \rfloor$
  $\mathcal{T}_0 := \mathcal{T}_0 \cup \text{linspace}(s_k, t_k, n_k)$
end
Obtain $\mathcal{T}$ using $\mathcal{T}_0$ and the sampling method of choice
return $\mathcal{T}$



(a) $5 \cdot 10^4$ iterations (b) $7.5 \cdot 10^4$ iterations (c) $1.5 \cdot 10^5$ iterations (d) $3 \cdot 10^5$ iterations

Figure 3. Visualization of the training process. Initially, we assign a large radius to all spheres in the cloud (a) and gradually reduce it during the optimization down to a minimum value (c). Our proposed repulsion loss prevents the clumping of the spheres and encourages exploration, which results in an improved reconstruction of the thin surfaces (d).

To further improve the efficiency of both proposal sampling and root-finding procedures, we utilize a set of guiding spheres $\mathbf{S}$ which cover the object's surface. They allow us to ensure that the training samples $\mathcal{P}$ are mainly generated from the areas of interest, making the implicit surface function converge to a better optimum, especially for the scenes with high-frequency details. We achieve that by applying both the root-finding and proposal sampling procedures only within the volume, defined by the sphere cloud $\mathbf{S}$, as illustrated in Figure 2. For the sampling of proposals $\mathcal{T}$, our method is described in the Algorithm 12, while the details of the modified root-finding approach can be found in the supplementary materials. In short, the algorithm first intersects a given ray with the sphere cloud, yielding a set of intersections $\mathcal{I}$. Then it finds the minimum coverage of the intersections. That is, $\{[s_k, t_k]\}$ is the minimal set of segments with union $\mathcal{I}$. Then each of these segments is linearly sampled to obtain the initial set of proposals $\mathcal{T}_0$. This set is finally upsampled using the base method.

3.3. Sphere cloud optimization

At the beginning of training, we initialize the sphere cloud $\mathbf{S}$ of size M with centers $\{\mathbf{c}_i\}_{i=1}^{M}$, uniformly distributed across the volume of the scene, and set the radii of the spheres to an initial value r_{max}. The training then proceeds by alternating the updates of the sphere cloud and the implicit function. Importantly, we only update the sphere centers $\mathbf{c}_i$ via an optimization-based process and rely on scheduling their radii to decrease from the initial value r_{max} to the minimum r_{min} via a fixed schedule. Also, in our approach, all spheres in the cloud are assigned the same radius value. Figure 3 illustrates the sphere cloud optimization during the training process.

The main learning signal for the sphere centers comes from moving them towards the estimated surface $\hat{\mathcal{S}}$, which is defined as an h-level set of the implicit function $f : \hat{\mathcal{S}} = \{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = h\}$, where h depends on the type of the function (e.g., for SDF $h = 0$). This can be formulated as a following loss:

$$\mathcal{L}_{\text{surf}} = \sum_{i=1}^{M} \big\| f(\mathbf{c}_i) - h \big\|_2. \tag{5}$$

This objective ensures that the sphere centers lie in the proximity of the reconstructed surface, i.e., maximizes the precision. However, it does not guarantee that the point cloud covers an entire object's surface. To address that, we design a repulsion term that prevents the neighboring spheres from clumping together and encourages exploration of the entire surface region:

$$\mathcal{L}_{\text{rep}} = \sum_{i=1}^{M} \sum_{j \in K(i)} r_n \frac{\mathbb{I}(||\mathbf{c}_j - \mathbf{c}_i||_2 < d)}{||\mathbf{c}_j - \mathbf{c}_i||_2}, \tag{6}$$

where $K(i)$ denotes the indices of the k-nearest spheres to $\mathbf{S}_i$, r_n is the current radius of the spheres, and d is a hyperparameter, which sets the maximum distance for the repulsion. Since the magnitude of this loss depends on the current radius of the spheres, the repulsion has more effect in the beginning of the training, encouraging better exploration of the scene volume.

Our final objective for optimization of the centers of the spheres is the following:

$$\mathcal{L} = \mathcal{L}_{\text{surf}} + \lambda \mathcal{L}_{\text{rep}}. \tag{7}$$

The radius scheduling in our method defines the exploration-exploitation trade-off and, in principle, could be picked separately for each scene. However, we found out that the following exponential schedule works well in most cases:

$$r_n = \max(r_{\text{max}}\, e^{-n\beta},\ r_{\text{min}}). \tag{8}$$

Here, n denotes the training iteration, and β is a hyperparameter controlling the decay rate. We use the same β value across datasets and set it so that the radius reaches the minimum value of r_{min} in less than half of the training iterations.

To avoid problems with the sphere cloud convergence, we apply a resampling procedure for the empty spheres which get stuck without reaching the surface. This process is typically applied up to 8 times during training, depending on the total number of iterations. Similarly to [16], we sample K points inside each sphere at which we evaluate the implicit function and find the spheres which have no surface inside them. We then resample these spheres near the ones which contain a surface region. Lastly, to avoid choosing training rays that do not intersect the surface of the object, we sample their endpoints uniformly from the volume bounded by the spheres. For more details on the sphere resampling and sphere-guided ray sampling procedures, please refer to the supplementary materials.

4. Experiments

We conduct our main experiments using three popular 3D reconstruction benchmarks: DTU MVS [11], BlendedMVS [34] and Realistic Synthetic 360 [19] and evaluate our approach by combining it with four different methods for implicit surface reconstruction.

4.1. Base methods

Our approach acts as an addition to the 3D reconstruction systems that learn neural implicit surfaces through volume rendering. Therefore, we apply it to four representative systems to showcase its effectiveness and applicability. UNISURF [22] represents the geometry of the scene via an occupancy field that is learned through a combination of surface and volume rendering approaches. VolSDF [35] and NeuS [31] propose to train an SDF via volume rendering by transforming it into occupancy defined along the ray. NeuralWarp [5] builds upon VolSDF by using an additional loss term that directly enforces the photo consistency of the learned geometry by warping patches across different views. Our approach can be seamlessly incorporated into all these systems, and we provide additional implementation details in the supplementary materials.

4.2. Training process

For each of the base models, we have used the official codebase, except for VolSDF, for which we use the code provided as part of the NeuralWarp method. Therefore, for the implementation aspects of the base methods, including the architectures and training details, we refer to the respective publications. We employ the same optimizer, scheduling, hyperparameters, number of iterations, and other technicalities as in the reference methods to train the implicit functions.

The sphere cloud optimization process is adapted for each system with regard to the implicit geometry type and the total number of iterations used for training. The sphere radius in the SDF-based methods decays exponentially from $r_{\text{max}} = 0.4$ to $r_{\text{min}} = 0.04$, while for UNISURF it ranges from $r_{\text{max}} = 2.0$ to $r_{\text{min}} = 0.1$. The repulsion penalty considers the $k = 10$ nearest spheres that intersect with each sphere in the cloud, i.e. $d = 2 * r_n$, and its weight is $\lambda = 0.1$ in UNISURF and $\lambda = 0.0001$ in the other methods. We found a number of 15,000 spheres to be sufficient for representing most scenes, which are scaled to fit in the bounding sphere of radius one. We employ the Adam [13] optimizer with a learning rate of 10^{-4} for the optimization of the centers of the spheres in all experiments.

4.3. Realistic Synthetic 360 evaluation

The Realistic Synthetic 360 dataset was introduced in [19] as a benchmark for the novel view synthesis task. Each of its eight scenes features an object realistically rendered from 100 training viewpoints and paired with a ground truth mesh. Though this dataset was not originally intended for the 3D reconstruction task, it contains objects with complex geometries and non-Lambertian materials, representing a challenge for classical 3D reconstruction systems. As some of the ground truth meshes contain internal surfaces that are not visible in any of the training views, we filter them by removing the non-visible parts. We perform a similar filtering step for the reconstructed meshes and compute the Chamfer distance between the cleaned meshes by sampling one million points on each surface. We report the distance computed at the original scale of the meshes multiplied by 10^2 in Table 1. We also report the qualitative results in Figure 4.

We can see that our method achieves improvements across most of the scenes for all of the compared methods, which is especially noticeable in scenes such as ficus and materials. We hypothesize that this is the case because of the complex structure of the reconstructed surface. This complexity does not allow the standard methods to effectively estimate the location of the high-density regions, which hinders the optimization process, leading to both reduced qualities of reconstructions and renders.

4.4. DTU and BlendedMVS evaluation

The DTU MVS dataset [11] contains 49 or 64 images with fixed camera positions and ground truth point clouds of 80 scenes acquired using a structured light scanner. We follow recent works [5, 22, 31, 35, 36] and perform the evaluation on the subset of 15 diverse scenes selected by [36]. The authors also provide the corresponding segmentation masks for each of the chosen scenes, which are used in the "w/ masks" experiments. Additionally, since most of the objects in the DTU dataset have a relatively simple geometry, we

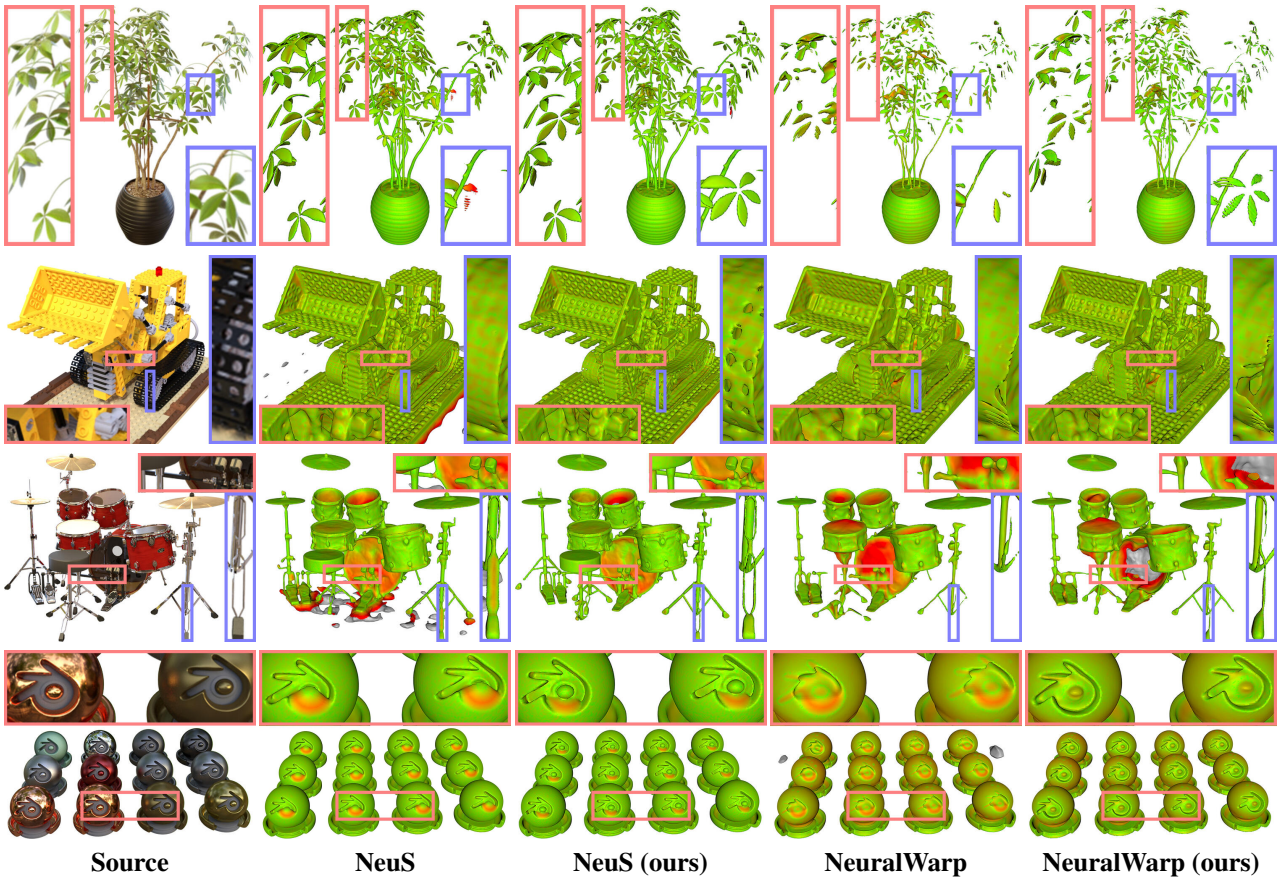


Figure 4. Qualitative results on the **Realistic Synthetic 360** dataset [19]. Our method improves upon the base **NeuS** [31] and **NeuralWarp** [5] surface modeling approaches, especially in the areas with thin details. The reconstructions are color-coded using the one-way Chamfer distance between the prediction and the ground truth, green color denotes a lower error.

	Scene name								
Method	Chair	Drums	Ficus	Hotdog	Lego	Mats	Mic	Ship	Mean
COLMAP [27]	0.77	1.26	0.96	1.95	1.36	2.19	1.33	1.00	1.42
NeuS [31]	**0.38**	1.88	0.51	**0.52**	0.68	0.40	**0.60**	0.60	0.70
NeuS (ours)	0.39	**1.20**	**0.40**	0.57	**0.61**	**0.31**	0.67	**0.54**	**0.59**
NeuS w/ masks	**0.40**	**0.90**	0.41	0.58	0.67	0.28	0.59	0.73	0.57
NeuS w/ m. (ours)	0.45	0.94	**0.32**	**0.54**	0.67	**0.27**	**0.57**	**0.71**	**0.56**
NeuralWarp [5]	0.43	3.00	0.94	1.65	0.81	1.02	**0.75**	1.27	1.23
NeuralWarp (ours)	**0.41**	**2.67**	**0.61**	**1.44**	**0.76**	**0.92**	0.80	**1.07**	**1.09**

Table 1. Quantitative results on the Realistic Synthetic 360 dataset [19]. We evaluate the Chamfer distance to compare the performance of different methods. We can see that base methods with our proposed modification achieve better performance than the original versions across most of the scenes.

conducted an additional qualitative evaluation for a subset of objects from the BlendedMVS dataset [34].

For a quantitative evaluation on the DTU dataset, we convert the trained implicit functions to triangle meshes using the Marching Cubes algorithm [15]. We then measure the Chamfer distances between the ground-truth point clouds and the extracted meshes using the standard evaluation procedure and report the scores in Table 2. These results are obtained without mask supervision during training, except for the "NeuS w/ masks" experiment. However, we follow other works in applying ground-truth masks for post-processing the meshes before calculating the metrics. Specifically, we filter the mesh using the visual hull obtained from the multi-view segmentations dilated with a radius of 12. The visual hull is calculated as the intersection of the silhouette cones emitted from each of the training cameras. Notably, we use

	Scene ID															
Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
UNISURF [22]	1.16	1.01	1.16	**0.36**	1.27	0.72	**0.73**	1.33	1.58	0.72	0.53	1.21	0.41	0.69	0.51	0.89
UNISURF (ours)	**1.10**	**0.98**	**1.14**	0.37	**1.08**	**0.66**	0.89	1.33	**1.19**	**0.69**	**0.52**	**0.98**	**0.38**	**0.46**	**0.50**	**0.82**
VolSDF* [35]	0.94	1.73	1.04	0.47	0.87	0.74	0.84	1.24	1.31	0.71	0.78	1.61	0.61	0.71	0.54	0.94
VolSDF* (ours)	**0.91**	**1.05**	**0.65**	**0.42**	**0.86**	**0.69**	**0.72**	**1.20**	**1.14**	**0.64**	**0.66**	**1.16**	**0.43**	**0.54**	**0.51**	**0.77**
NeuS [31]	0.93	1.06	0.81	0.38	1.02	0.60	**0.58**	1.43	1.15	**0.78**	0.57	1.15	0.35	0.45	0.46	0.78
NeuS (ours)	**0.83**	**0.91**	**0.69**	**0.36**	**0.95**	**0.54**	0.65	**1.37**	1.15	0.82	**0.55**	**0.86**	**0.33**	**0.42**	**0.42**	**0.72**
NeuS w/ masks	**0.83**	0.98	0.56	0.37	1.13	0.59	0.60	1.45	0.95	0.78	**0.52**	1.43	0.36	**0.45**	0.45	0.76
NeuS w/ m. (ours)	0.85	**0.92**	**0.46**	0.37	**1.05**	0.59	**0.58**	**1.27**	**0.88**	0.78	0.53	**0.93**	**0.33**	0.46	0.45	**0.70**
NeuralWarp [5]	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
NeuralWarp (ours)	0.49	0.77	**0.37**	0.40	0.81	0.87	**0.72**	**1.19**	1.07	**0.66**	**0.64**	**0.70**	**0.37**	**0.58**	**0.48**	0.68

* denotes unofficial implementation

Table 2. We present a quantitative evaluation of our method on the DTU [11] dataset. We have combined our approach with four popular implicit surface reconstruction systems: UNISURF [22], NeuS [31], NeuralWarp [5], and VolSDF [35]. In this comparison, we follow the previous works by measuring the Chamfer distance (lower the better) between the ground truth point cloud and the reconstructions, pre-processed by removing the regions outside the object via ground-truth segmentation masks. The models in this comparison were trained *without* segmentation masks unless stated otherwise. We highlight the better score between the original approach and our modification.

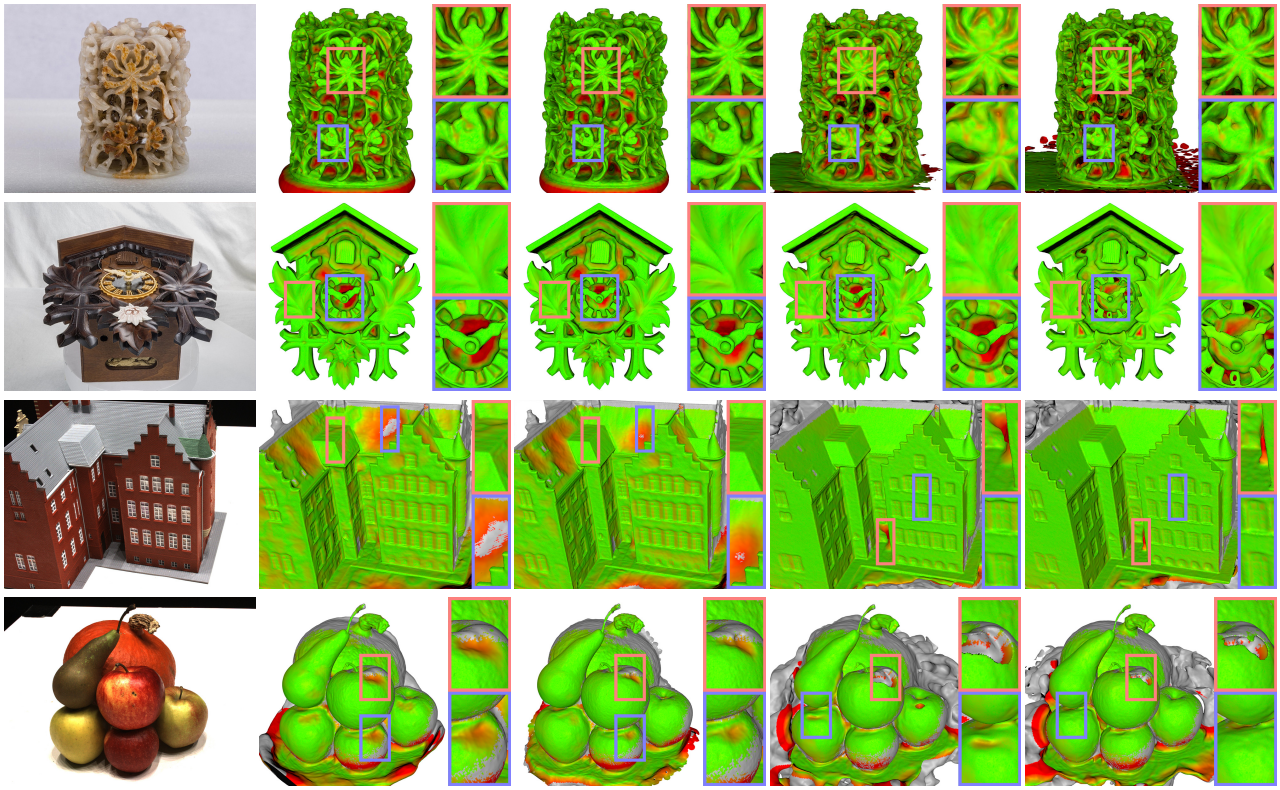


Figure 5. Qualitative results on the real-world **BlendedMVS** [34] (rows 1-2) and **DTU** [11] (rows 3-4) datasets. Our method achieves more accurate reconstructions compared to the base approaches. Green color denotes a lower one-way Chamfer error. For the clock scene (row 2), we rendered the geometry from a frontal angle, not present in the training views, to highlight the differences in the reconstructions.

the same dilation radius for all of the compared methods, hence the reported metrics may differ from the ones presented in the original publications. As we can see from the results, our method improves three out of four base methods uniformly across most of the scenes.

4.5. Ablation study

We have conducted an ablation study to evaluate the components of our method. We present the main results in Table 3, and include additional experiments in the supplementary material.

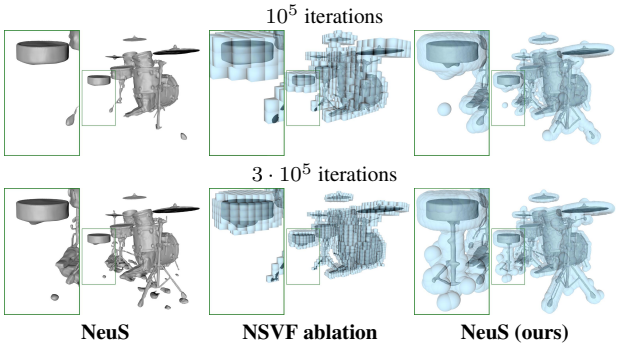


Figure 6. Qualitative comparison of the geometry progression during optimization between our spheres and a NSVF-style explicit geometry. The models are trained without mask supervision to highlight the benefits of the proposed sphere guidance.

	Scene name							
Method	Chair	Drums	Ficus	Hotdog	Lego	Mats	Mic	Ship
NeuS	0.38	1.88	0.51	**0.52**	0.68	0.40	**0.60**	0.60
NSVF ablation	**0.37**	3.12	0.49	**0.52**	0.64	0.34	3.38	0.73
w/o $\mathcal{L}_{\text{rep}}$	0.76	2.37	0.79	2.71	1.03	0.95	1.87	4.30
NeuS (ours)	0.39	**1.20**	**0.40**	0.57	**0.61**	**0.31**	0.67	**0.54**

Table 3. Ablation study on the importance of our repulsion loss and the benefit of joint optimization of the guiding primitive.

For the first experiment, we evaluate our gradient-based sphere cloud optimization scheme. To do that, we replace our sphere cloud with a sparse voxel octree data structure and use its optimization method proposed in Neural Sparse Voxel Fields [16] paper. We initialize our sparse voxel grid with 512 voxels and use the same scheduling for pruning and subdivision, as in the original NSFV approach, multiplied by a factor of two (since we use two times more iterations for training). For the pruning of voxels, we utilize the same resampling criteria as in our method but relaxed it to allow the voxels within an $\epsilon = 0.01$ distance to the surface not to be pruned. We observe that NSVF underperforms due to the greedy voxel pruning optimization strategy. The errors in the coarse reconstruction become permanent after the empty voxels are pruned. In comparison, our gradient-based approach allows the exploration and inclusion of the neighboring areas initially missing from the coarse reconstruction.

In the second experiment, we ablate the repulsion loss $\mathcal{L}_{\text{rep}}$ and observe that it leads to spheres clumping together in the learned sphere cloud, not covering the entire object, which eventually produces poor reconstructions. For additional ablation experiments, as well as rendered results, please refer to the supplementary material.

4.6. Limitations

While our approach improves the performance of base implicit surface reconstruction methods, it still inherits some of the limitations of the chosen framework. And, in some cases, even amplifies them by taking samples closer to the current approximation of the surface. In Figure 7, we show that the method fails to reconstruct the whole Ficus and creates artifacts on the Materials scene. In the first case, similarly to the baseline, the surface is not formed for several stems. Our spheres do not recover these regions since we skip rays that do not intersect the sphere cloud, leading to limited exploration capabilities after the initial convergence phase. On the materials scene, when using NeuralWarp, appearance variations on the base of the globes get incorrectly baked into the objects' surface. This artifact also appears and may be magnified in the case of our method.

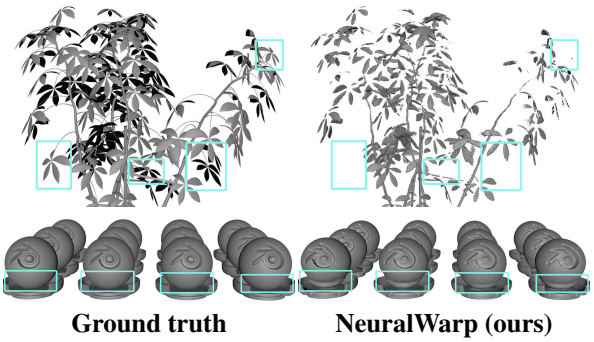


Figure 7. The main limitation of our method is the reliance upon coarse geometry estimated relatively early in the training. Our method, by design, samples more from the regions with a large number of spheres, which may cause artifacts related to exploration. Also, systematic artifacts are not corrected and may be amplified by our algorithm.

5. Conclusion

We have presented a method for improving volume rendering of implicit surface representations by restricting the modeled volume to the region of interest, defined by the set of trainable spheres. We have shown that focusing the optimization process on the estimated surface region leads to an increased quality of reconstructions obtained by the implicit functions. At the same time, our proposed sphere cloud optimization approach ensures that the guiding representation closely follows the estimated surface of the object and accurately represents it at each step of the training. We have conducted an extensive evaluation of our method, which includes combining it with four base systems for implicit function training, and found it to improve their performance across multiple benchmark datasets. Our method also shows clear gains for modeling the real-world scenes when applied without *any* additional supervision and post-filtering of the obtained reconstructions, which shows its suitability for in-the-wild applications.

Acknowledgements. The work was supported by the Analytical center under the RF Government (subsidy agreement 000000D730321P5Q0002, Grant No. 70-2021-00145 02.11.2021).

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision (ECCV)*, pages 696–712. Springer, 2020. 2

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[3] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, 2011. 1

[4] Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2008. 1

[5] Francois Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6260–6269, 2022. 1, 2, 3, 5, 6, 7

[6] Carlos Hernández Esteban and Francis J. M. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 46–53, 2003. 1

[7] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2022. 2

[8] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Found. Trends Comput. Graph. Vis.*, 9:1–148, 2015. 1

[9] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1362–1376, 2010. 1

[10] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2821–2830, 2018. 1

[11] Rasmus Ramsbol Jensen, A. Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014. 5, 7

[12] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, 2022. 1

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[14] Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1440–1449, 2021. 1, 2

[15] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003. 6

[16] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1, 2, 5, 8

[17] Stephen Lombardi, Tomas Simon, Jason M. Saragih, Gabriel Schwartz, Andreas M. Lehrmann, and Yaser Sheikh. Neural volumes. *ACM Transactions on Graphics (TOG)*, 38:1 – 14, 2019. 1, 2

[18] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason M. Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40:1 – 13, 2021. 1, 2

[19] 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 (ECCV)*, 2020. 1, 2, 3, 5, 6

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[21] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3504–3515, 2020. 2

[22] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5569–5579, 2021. 1, 2, 3, 5, 7

[23] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 1

[24] Ruslan Rakhimov, Andrei-Timotei Ardelean, Victor Lempitsky, and Evgeny Burnaev. Npbg++: Accelerating neural point-based graphics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15969–15979, June 2022. 2

[25] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022. 2

[26] 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 (ECCV)*, 2016. 1

[27] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 6

[28] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[29] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5459–5469, 2022. 1

[30] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3D reconstruction in the wild. In *SIGGRAPH Conference Proceedings*, 2022. 2

[31] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 1, 2, 3, 5, 6, 7

[32] Yifan Wang, Shihao Wu, A. Cengiz Öztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–383, 2021. 1, 2

[33] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5438–5448, 2022. 1

[34] 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. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 6, 7

[35] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1, 2, 3, 5, 7

[36] 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*, 33:2492–2502, 2020. 1, 2, 5

[37] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5752–5761, 2021. 1, 2

[38] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *British Machine Vision Conference (BMVC)*, 2020. 1