Sensing Surface Patches in Volume Rendering for Inferring Signed Distance Functions

Sijia Jiang, Tong Wu, Jing Hua*, Zhizhong Han

Department of Computer Science, Wayne State University, Detroit, MI, USA
sijiajiang@wayne.edu, tongwu@wayne.edu, jinghua@wayne.edu, h312h@wayne.edu

Abstract

It is vital to recover 3D geometry from multi-view RGB images in many 3D computer vision tasks. The latest methods infer the geometry represented as a signed distance field by minimizing the rendering error on the field through volume rendering. However, it is still challenging to explicitly impose constraints on surfaces for inferring more geometry details due to the limited ability of sensing surfaces in volume rendering. To resolve this problem, we introduce a method to infer signed distance functions (SDFs) with a better sense of surfaces through volume rendering. Using the gradients and signed distances, we establish a small surface patch centered at the estimated intersection along a ray by pulling points randomly sampled nearby. Hence, we are able to explicitly impose surface constraints on the sensed surface patch, such as multi-view photo consistency and supervision from depth or normal priors, through volume rendering. We evaluate our method by numerical and visual comparisons on scene benchmarks. Our superiority over the latest methods justifies our effectiveness.

Introduction

3D reconstruction from multi-view images is an important task in 3D computer vision. Classic methods like structure from motion and multi-view stereo (Schönberger and Frahm 2016; Schönberger et al. 2016) estimate 3D point clouds per the multi-view photo consistency. With deep learning models (Yao et al. 2018; Yu et al. 2022), we are able to learn depth priors from a large scale dataset, which can be further generalized to infer depth maps from unseen images. Although these methods can estimate a coarse geometry from depth predictions, it is still a challenge to recover continuous and complete surfaces with details from multi-view images.

The latest methods (Fu et al. 2022; Yu et al. 2022; Wang et al. 2022; Guo et al. 2022) infer signed distance functions (SDFs) from multi-view images through volume rendering, and then run the marching cubes (Lorensen and Cline 1987) to reconstruct the surface. To supervise signed distances, they transform the predicted signed distances into radiance to render a pixel color by integrating colors along a ray, which can be optimized by minimizing the difference to the ground truth pixel color. Although these methods obtained smooth and complete reconstructions, severe artifacts may appear in the empty space, the reconstructed surfaces may drift away from the GT surface, and few details can be revealed on the surface, due to the unawareness of the surface. Hence, how to sense the surface and further impose more effective surface constraints along with the volume rendering is the key to improve the learning of SDF.

To resolve this problem, we propose to infer an SDF from multi-view images through volume rendering with a better sense of surface. Our novelty lies in the way of establishing a surface patch around an estimated ray-surface intersection, which enables to explicitly impose more effective constraints on surface patches, along with the volume rendering. Specifically, using the predicted signed distances and gradients, we randomly sample queries near an estimated intersection, and pull them onto the zero level set, which produces a surface patch. With the sensed surface patch, we are able to explicitly impose surface constraints such as multi-view photo consistency and supervision from depth or normal priors, to improve the SDF inference through volume rendering. We justify the effectiveness of our modules, and report superiority performance over the latest methods in terms of numerical and visual comparisons on widely used benchmarks. Our contributions are listed below.

- We introduce a method for SDF inference that can get constrained not only through volume rendering but also by constraints that can be explicitly imposed on surfaces. It significantly improves the accuracy of inferred SDFs.
- We propose to use predicted signed distances and gradients to sense a surface patch near the estimated intersection of a ray and the zero level set in volume rendering.
- We justify the feasibility of our idea and report the state-of-the-art performance on the widely used benchmarks.

Related Work

Mult-view 3D Reconstruction. 3D shape reconstruction from multiple images has been extensively studied (Schönberger and Frahm 2016; Schönberger et al. 2016; Mildenhall et al. 2020; Vicini, Speierer, and Jakob 2022). Given multiple RGB images, classic multi-view stereo (MVS) (Schönberger and Frahm 2016; Schönberger et al. 2016) employ multi-view photo consistency to estimate

*Corresponding author: Jing Hua

depth maps. However, these methods are limited by large viewpoint variations and complex illumination. Alternatively, with multiple silhouette images, (Laurentini 1994) proposed to reconstruct 3D shapes as voxel grids using space carving. These methods lack the ability to reveal concave structures and work with high-resolution voxel grids.

Recent methods (Yao et al. 2018) employ neural networks to learn prior knowledge to predict depth maps. During training, they learn priors using depth supervision or multi-view consistency in an unsupervised way, and then generalize the learned priors to predict depth images for unseen cases through a forward pass. Recent works (Huang et al. 2024; Chen, Li, and Lee 2023; Yu, Sattler, and Geiger 2024; Guédon and Lepetit 2023; Zhang, Liu, and Han 2024) leverage 3D Gaussian Splatting for surface reconstruction. However, they are inferior to neural implicit methods in quality due to the explicit and disconnected 3D Gaussians.

These methods reconstructed 3D shapes as point clouds or voxel grids, both of which are discrete 3D representations. While neural implicit representations represent continuous surfaces as the zero level set for 3D reconstruction.

Neural Implicit Representations. Neural implicit representations have become a popular 3D representation, using coordinate-based neural networks to map coordinates to signed distances or occupancy labels. These can be inferred from 3D supervision (Takikawa et al. 2021; Liu et al. 2021; Tang et al. 2021), point clouds (Zhou et al. 2022; Chen, Liu, and Han 2022, 2023b; Ma et al. 2023; Chen, Liu, and Han 2023a, 2024), or multi-view images (Mildenhall et al. 2020; Guo et al. 2022; Zhang et al. 2024; Hu and Han 2023; Jiang, Hua, and Han 2023). Methods using 3D supervision or point clouds typically skip positional encodings, while multi-view approaches use them to capture high-frequency details.

Differentiable rendering enables tuning implicit representations by minimizing errors between the rendered and ground truth images. Surface rendering methods like DVR (Niemeyer et al. 2020) and IDR (Yariv et al. 2020) predict radiance on surfaces and use view direction for high-frequency detail but require background filtering. Volume rendering methods like NeRF (Mildenhall et al. 2020) and its variations (Müller et al. 2022; Azinović et al. 2022) model geometry and color without masks. UNISURF (Oechsle, Peng, and Geiger 2021) and NeuS (Wang et al. 2021) refine occupancy and signed distance fields using revised rendering equations, with improvements via depth (Yu et al. 2022), normals (Guo et al. 2022), and multi-view consistency (Fu et al. 2022).

Our approach differs previous methods by sensing surface patches along rays and imposing explicit surface constraints to recover finer geometric details.

Method

Overview. We aim to recover the geometry of a scene by learning an SDF f from K posed images C_k^{GT}. We can use additional supervision such as depth D_k^{GT} and normal N_k^{GT} maps that are either captured by real sensors or estimated by monocular networks, where $k \in [1, K]$. For randomly sampled 3D queries q, f predicts its signed distance $f(q)$ at q. We parameterize the SDF f using a coordinate-based MLP with parameters θ.

As illustrated in Fig. 1, we aim to infer f_θ along with a color function l_ϕ which is parameterized by another MLP with parameters ϕ through volume rendering. Our optimization is to minimize a loss L according to $\{C_k^{GT}, D_k^{GT}, N_k^{GT}\}$,

$$\min_{\theta,\phi} L(f_\theta, l_\phi, \{C_k^{GT}\}, \{D_k^{GT}\}, \{N_k^{GT}\}). \tag{1}$$

Geometry prediction. We use an MLP to approximate the geometry function f_θ. For a query q, f_θ use the coordinate and the position encoding (Mildenhall et al. 2020) $e(q)$ to capture the geometry with high frequency. The SDF f_θ predicts the signed distance $d = f_\theta(q, e(q))$ at q.

Color Prediction. We use another MLP to approximate the color function l_ϕ. To model the view-dependent color c at q, we also leverage the view direction v, the gradient g in the signed distance field, and the feature z of geometry around q. Hence, we predict color by $c = l_\phi(q, e(q), v, g, z)$. We obtain the gradient g from the geometry function f_θ as $g = \nabla f_\theta$, which can be produced by automatic differentiation from the geometry network. And we use the output of one FC layer from the geometry network as the feature z.

Volume Rendering. We use volume rendering to render the radiance field represented by the SDF f_θ and the color function l_ϕ into images. We can learn the parameters θ and ϕ by minimizing the rendering error to the ground truth images.

We start from emitting a ray r from the camera center o through a randomly sampled pixel on an image. To render a color along the ray r pointing to a view direction v, we sample points $\{q_i | i \in [1, I]\}$ along r by $q_i = o + t_i * v$. We use the geometry network $d_i = f_\theta(q_i, e(q_i))$ and color network $c_i = l_\phi(q_i, e(q_i), v, g_i, z_i)$ to predict the signed distance and color at each sampled point q_i. We follow VolSDF (Yariv et al. 2021) to transform the signed distance d_i to density values σ_i for volume rendering.

Following NeRF (Mildenhall et al. 2020), the color C_r for the ray r is integrated by,

$$\alpha_i = 1 - exp(-\delta_i \sigma_i),\ T_i = \prod_{j=1}^{i-1}(1-\alpha_j),\ C_r = \sum_{i=1}^{I} T_i \alpha_i c_i, \tag{2}$$

where α_i is the alpha value at point q_i, δ_i is the interval between neighboring points, and T_i is the transmittance through q_i. Similarly, we can render depth D_r and normal map N_r using the following equations,

$$D_r = \sum_{i=1}^{M} T_i \alpha_i t_i,\ N_r = \sum_{i=1}^{M} T_i \alpha_i n_i. \tag{3}$$

Surface Sense. We use the geometry function f_θ to sense the surface. NeRF or its variations (Yariv et al. 2021; Wang, Skorokhodov, and Wonka 2022; Wang et al. 2022) apply the secant method (Mescheder et al. 2019) to estimate the intersection of a ray and the surface per the signed distances or occupancy labels predicted at points sampled along the ray, as shown in Fig. 2 (a). Although it is a way of sensing

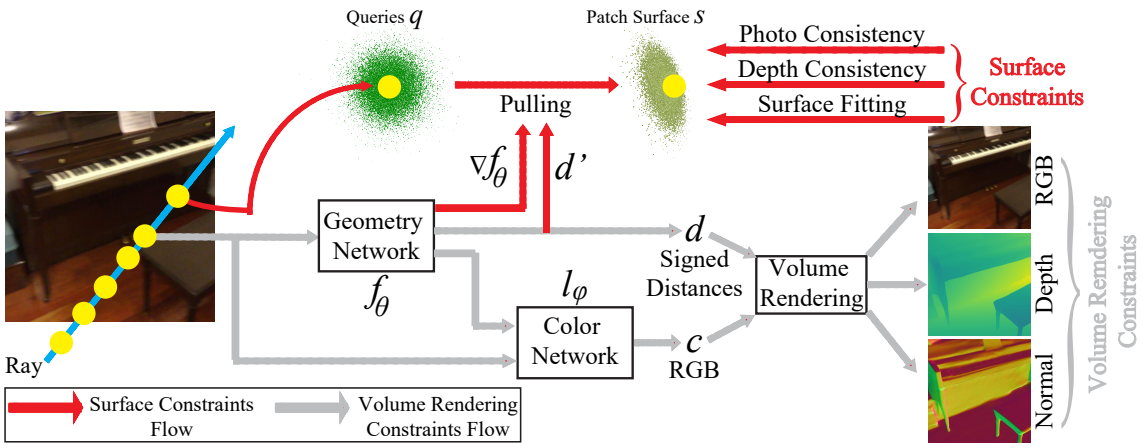


Figure 1: Overview of our method. We infer SDF f_θ from multi-view images including RGB images, depth and normal maps that were either captured by sensors or estimated by monocular networks. Using the predicted signed distances and gradients ∇f_θ, we are enabled to sense a surface patch s by pulling randomly sampled queries q onto the zero level set as shown in Fig. 2 (d). With s, we can infer f_θ using both supervision through volume rendering and constraints that can be explicitly imposed on the sensed surface s.

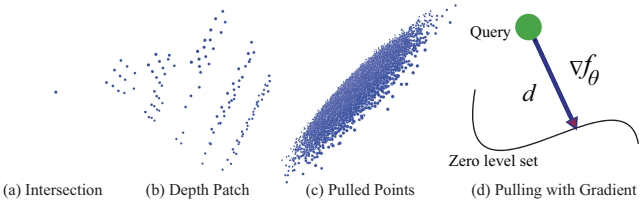


Figure 2: Patch Difference. Current methods mainly impose constraints on single points on surface in (a), rather than a patch, since it is very hard to obtain a 3D surface patch during volume rendering. Different from obtaining a patch from depth in (b), our method projects randomly sampled points on the zero level set to obtain the surface patch which is more representative in (c).

a surface, on the zero level set, only one single point gets supervised during the differentiable rendering procedure.

We introduce a novel way of sensing a surface patch by pulling randomly sampled queries. The sensed surface is represented by pulled queries as shown in Fig. 2 (c), which is a denser and more continuous surface representation than a single point intersection in Fig. 2 (a). It is much more geometry-aware than the single point intersection and also points back-projected from a depth patch in Fig. 2 (b).

Using the character of SDF, we can leverage the gradient and the signed distance to move any points onto the zero level set (Ma et al. 2021; Chou, Chugunov, and Heide 2022; Ma et al. 2022a,b).

More specifically, for a 3D point $q = [x, y, z]$, it is located on the d-level set in a signed distance field represented by f_θ, where $d = f_\theta(q, e(q))$. The gradient at q in the field $\nabla f_\theta(q, e(q)) = [\partial f_\theta/\partial x, \partial f_\theta/\partial y, \partial f_\theta/\partial z]$ points to the level sets with larger signed distances than d. As demonstrated in Fig. 2 (d), we can pull q onto its zero level set by moving it along its gradient g with a stride of $|d|$,

$$q' = q - d \times \nabla f_\theta(q, e(q))/||\nabla f_\theta(q, e(q))||_2, \quad (4)$$

where q' is the projection of point q on the zero level set.

Our idea of sensing a surface patch is to use the projections of randomly sampled points to form a point surface. Specifically, we sample a set of points $\{p_j|j \in [1, J]\}$ around a 3D anchor q using a Gaussian distribution. The Gaussian distribution has a variance τ^2 to cover the area centered at q between neighboring rays. τ^2 determines the size of the sensed patch, we report ablation study on τ^2 in experiments. The anchor q could be either the intersection estimated along a ray or a point back-projected from posed depth maps. Using Eq. 4, we pull each p_j to its projection p'_j on the zero level set which represents the surface. We denote a surface patch s that we sense as,

$$s = \{p'_j|j \in [1, J]\}. \quad (5)$$

Optimization

Constraints through Volume Rendering

We use the following losses to provide constraints through the volume rendering. We follow MonoSDF (Yu et al. 2022) to use losses to supervise rendered RGB, depth and normal maps. The GT depth maps and normal maps are either captured by sensors or estimated by monocular networks.

RGB Rendering Loss. We use the RGB images C_k^{GT} to supervise the images rendered from the field using Eq. 6,

$$L_{RGB} = \sum\nolimits_{r \in B} ||C_k(r) - C_k^{GT}(r)||_1, \quad (6)$$

where $C_K^{GT}(r)$ is the ground truth pixel color on the RGB image and B denotes a set of sampled rays in a mini-batch.

Depth Rendering Loss. With GT depth maps D_k^{GT}, we can supervise the depth maps rendered from Eq. 3 below,

$$L_{DR} = \sum\nolimits_{r \in B} ||D_k(r) - D_k^{GT}(r)||_1, \tag{7}$$

where $D_k^{GT}(r)$ is the depth cue at the ray r in mini-batch B.

Normal Rendering Loss. Given GT normal maps N_k^{GT}, we can also supervise the normal maps rendered from Eq. 3,

$$\begin{aligned} L_{Nor} = \sum\nolimits_{r \in B} &||N_k(r) - N_k^{GT}(r)||_1 + \\ &||1 - N_k(r)^T N_k^{GT}(r)||_1, \end{aligned} \tag{8}$$

where $N_k^{GT}(r)$ is the normal cue at the ray r in mini-batch.

Eikonal Loss. To learn f_θ as a SDF, we constrain the gradients ∇f_θ in the field using the Eikonal term,

$$L_{Eik} = \sum\nolimits_{q \in B} (||\nabla f_\theta(q, e(q))||_2 - 1)^2, \tag{9}$$

where $q \in B$ denotes all points sampled on rays in the mini-batch B.

Surface Constraints

With a surface patch s that we sense using Eq. 5, we are able to explicitly impose surface constraints on s.

Depth Consistency. With GT depth maps, we can regress depth of points p'_j on the surface patch s by projecting them to the current view plane. We compute the consistency of the calculated depth and the regressed depth,

$$L_{DC} = \sum\nolimits_{p'_j \in s} w_j \times (D_k^{GT}(p'_j) - Z_k(p'_j))^2, \tag{10}$$

where $D_k^{GT}(p'_j)$ is the depth interpolated at the projection of p'_j on the depth map D_k^{GT} using bilinear interpolation, and $Z_k(p'_j)$ is the projected depth from p'_j to the plane of D_k^{GT} using the pose and the intrinsic matrix of the camera. We use a mask w_j to rule out p'_j whose projections are out of the view range or $|D_k^{GT}(p'_j) - Z_k(p'_j)|$ is larger than $15mm$ which indicates potential invisibility from the current view.

L_{DC} is different from L_{DR} in Eq. 7. L_{DC} only constrains zero level set in the field, while L_{DR} constrains all locations sampled along a ray r, due to the integration in rendering.

Photometric Consistency. With the surface patch s, we constrain the photo consistency on s across different views. We use the normalization cross correlation (NCC) of patches in one reference gray image $U'_{k1} = gray(C_{k1}^{GT})$ and another source gray image $U'_{k2} = gray(C_{k2}^{GT})$,

$$NCC(U'_{k1}(s), U'_{k2}(s)) = \frac{Cov(U'_{k1}(s), U'_{k2}(s))}{\sqrt{Var(U'_{k1}(s)) Var(U'_{k2}(s))}}, \tag{11}$$

where Cov and Var are the covariance and the variance over the gray level color interpolated at projections of $\{p'_j\}$, denoted as $U'(s)$. We regard the view to be rendered as the reference image U'_{k1} and regard the neighboring eight images as the source images $\{U'_{k2}\}$. We consider occlusion, and only use the top three largest NCC scores to compute the following photometric consistency loss below,

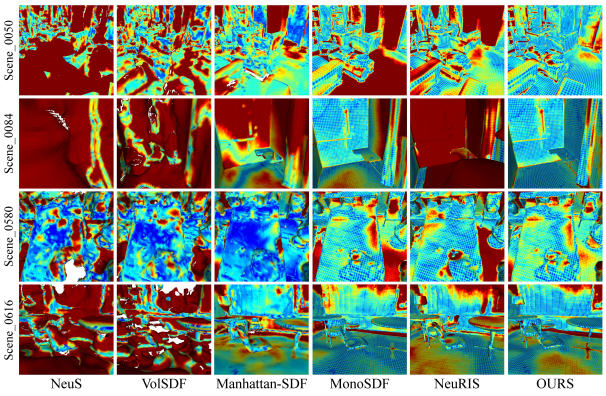


Figure 3: Error map comparison on ScanNet (bigger error: red, smaller error: blue) highlights our superiority.

$$L_{NCC} = \frac{\sum_{k2=1}^{3} 1 - NCC(U'_{k1}(s), U'_{k2}(s))}{3}. \tag{12}$$

Surface Fitting. We improve the smoothness of the surface patch s using a surface fitting loss. We hope all p'_j on s can locate on the same plane determined by the depth supervision $D_k^{GT}(r)$ and the normal supervision $N_k^{GT}(r)$ of ray r. We use $\alpha x + \beta y + \gamma z + \mu = 0$ to represent the plane, and solve $[\alpha, \beta, \gamma, \mu]$ using $D_k^{GT}(r)$ and $N_k^{GT}(r)$. We measure the fitting error using the equation below,

$$L_{Fit} = \sum\nolimits_{p'_j \in s} w_j \times \eta_j \times ||\alpha x_j + \beta y_j + \gamma z_j + \mu||^2, \tag{13}$$

where w_j is the mask introduced in Eq. 10 and η_j is the confidence determined by the gradient $\nabla f_\theta(p'_j, e(p'_j))$. We model the confidence as the consistency between the gradient $\nabla f_\theta(p'_j, e(p'_j))$ and the normal $N_k^{GT}(r)$ of r using the cosine distance $\eta_j = cos(\nabla f_\theta(p'_j, e(p'_j)), N_k^{GT}(r))$, which pushes p'_j more onto the plane if its gradient is well aligned with the normal $N_k^{GT}(r)$ of r. Note that $\nabla f_\theta(p'_j, e(p'_j))$ involves second order derivative due to the p'_j in Eq. 4 which helps the network to find better solutions (Ben-Shabat, Koneputugodage, and Gould 2021).

Loss Function

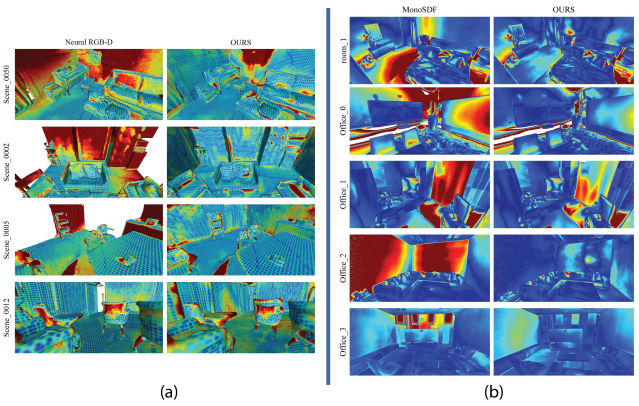
We use all these constraints to infer the geometry and color in the field below,

$$\begin{aligned} L = &L_{RGB} + \lambda_1 L_{DR} + \lambda_2 L_{Nor} + \lambda_3 L_{Eik} \\ &+ \lambda_4 L_{DC} + \lambda_5 L_{NCC} + \lambda_6 L_{Fit}, \end{aligned} \tag{14}$$

where λ_1 to λ_6 are balance weights which make each term contribute to the performance equally.

Experiments and Analysis

We report numerical and visual comparisons with the latest methods on real-world indoor scene to highlight our superiority and justify the effectiveness of module in our method.



(a) (b)

Figure 4: (a) Visual comparison on ScanNet released by NeuralRGBD. We use error maps (bigger error: red, smaller error: blue) to highlight our superiority over NeuralRGBD. (b) Visual comparison on Replica. We use error maps (bigger error: red, smaller error: blue) to highlight our superiority over MonoSDF.

	Acc↓	Comp↓	CD-L1↓	Prec↑	Recall↑	F-score↑
COLMAP	0.047	0.235	0.141	0.711	0.441	0.537
UNISURF	0.554	0.164	0.359	0.212	0.362	0.267
NeuS	0.179	0.208	0.194	0.313	0.275	0.291
VolSDF	0.414	0.120	0.267	0.321	0.394	0.346
Manhattan	0.072	0.068	0.070	0.621	0.586	0.602
NeuRIS	0.050	0.049	0.050	0.717	0.669	0.692
Neuralangelo	0.245	0.272	0.258	0.274	0.311	0.292
MonoSDF	**0.035**	0.048	0.042	0.799	0.681	0.733
Ours	0.036	**0.039**	**0.037**	**0.820**	**0.777**	**0.797**

Table 1: Comparisons on ScanNet released by MonoSDF.

Datasets. We evaluate our method by comparisons with the latest methods for scene reconstruction from multi-view images under both synthetic scenes and real scans. The synthetic indoor scenes are Replica (Straub et al. 2019), released by MonoSDF (Yu et al. 2022), the real scans indoor scenes are ScanNet (Dai et al. 2017), released by either MonoSDF or NeualRGBD (Azinović et al. 2022), and real-world large-scale indoor scenes, Tanks and Temples (Knapitsch et al. 2017), released by MonoSDF.

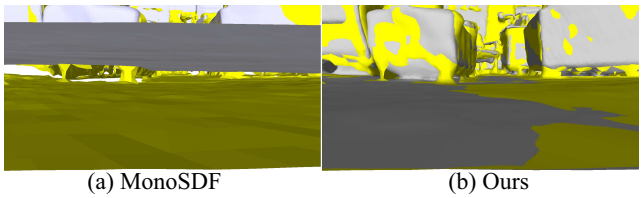
Baselines. We compare our method with SOTA neural implicit-based reconstruction methods, including COLMAP (Schönberger and Frahm 2016), UNISURF (Oechsle, Peng, and Geiger 2021), NeuS (Wang et al. 2021), VolSDF (Yariv et al. 2021), ManhattanSDF (Guo et al. 2022), NeuRIS (Wang et al. 2022), Neuralangelo (Li et al. 2023), and MonoSDF on ScanNet, using MonoSDF's experimental setup with monocular depth and normal cues as supervision. For NeuralRGBD comparisons, we adopt its setting, using sensor-captured depth for fair evaluation.

Evaluation Metrics. As for evaluation metrics, we follow previous methods (Yu et al. 2022; Azinović et al. 2022), and report accuracy, completeness, Chamfer Distance (CD), the F-score with a threshold of 5cm, Precision and Recall, as well as normal consistency (NC).

Details. For each posed view, we sample 1024 rays per training batch. Using VolSDF's error-bounded sampling strategy and architecture, we sample points along rays. To create surface patches, we backproject geometry cues (either predicted monocular or dataset-provided sensor cues) to obtain 3D anchor points q. Around each anchor, we define an isotropic Gaussian distribution $N(q, \tau^2)$ and sample $J = 9$ points, with τ^2 controlling patch size. Ablation studies in Tab. 5 analyze the effect of different τ^2 values. Loss weights are set as $\lambda_1 = 0.1$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$, $\lambda_4 = 0.5$, and $\lambda_6 = 0.5$ for balanced contributions. To improve coarse shape reconstruction, we apply inverse weight annealing for L_{NCC}, setting $\lambda_5 = 0$ for the first 100 epochs and gradually increasing it to 0.1.

		Scene_0050	Scene_0002	Scene_0005	Scene_0012	Mean
NeuralRGBD	Acc[cm]↓	2.84	3.93	**4.13**	2.78	**3.42**
	Comp[cm]↓	10.41	32.72	33.35	3.03	19.88
	Chamfer-L1↓	6.63	18.33	18.74	2.90	11.65
	Prec↑	90.24	73.74	72.68	91.33	82.00
	Recall↑	71.19	34.05	41.30	86.33	58.22
	F-score↑	79.59	46.59	52.67	88.76	66.90
OURS	Acc[cm]↓	**2.78**	**3.70**	5.51	**2.71**	3.68
	Comp[cm]↓	**3.16**	**7.10**	**8.98**	**2.83**	**5.52**
	Chamfer-L1↓	**2.97**	**5.4**	**7.25**	**2.77**	**4.60**
	Prec↑	**92.54**	**86.90**	**80.66**	**91.48**	**87.89**
	Recall↑	**85.27**	**73.12**	**73.83**	**88.33**	**80.14**
	F-score↑	**88.76**	**79.41**	**77.09**	**89.88**	**83.79**

Table 2: Numerical comparison with NeuralRGBD on ScanNet subsets used by NeuralRGBD. GT meshes were generated with TSDF fusion using the same amount of images and depth maps as did NeuralRGBD on each scene. Under the same setting as NeuralRGBD, GT depth is sensor captured depth maps.



(a) MonoSDF (b) Ours

Figure 5: Compactness comparison with MonoSDF (GT mesh:Gray).

Experimental Results

ScanNet from MonoSDF. We present numerical comparisons with the latest methods in Tab. 1. Following MonoSDF (Yu et al. 2022), we use monocular cues estimated by the pretrained Omnidata model (Eftekhar et al. 2021). To align the Omnidata depth (range $[0, 1]$) with the rendered depth D_r in Eq. 7, we solve scale w and shift q parameters via least-squares optimization per mini-batch. These parameters are used to backproject depth and calculate L_{DC} in Eq. 10.

As shown in Tab. 1, our method produces more accurate and smoother surfaces. Neuralangelo struggles with real-world indoor scenes like ScanNet, even with parameter tuning and monocular cues, as evident in our visual comparisons (Fig. 3) and error maps (supplementary). Additionally, our method enhances the quality of volume-rendered images, with visual and PSNR comparisons available in the

	Test split			Train split		
	Normal C.↑	CD-L1↓	F-score↑	Normal C.↑	CD-L1↓	F-score↑
MonoSDF	**92.11**	2.94	86.18	93.86	2.63	92.12
Ours	91.68	**2.81**	**89.73**	**94.29**	**2.37**	**94.09**

Table 3: Comparisons with MonoSDF on Replica.

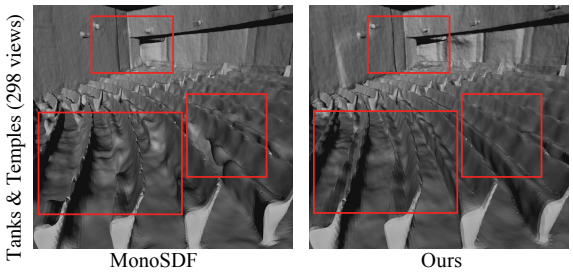


Figure 6: Visual comparison with MonoSDF on Tanks and Temples.

supplementary material.

ScanNet from NeuralRGBD. NeuralRGBD (Azinović et al. 2022) evaluated on a different subset of ScanNet using sensor-captured depth maps but lacked normal maps. In this case, we use only the ground truth depth maps for L_{DR} in Eq. 7 and L_{DC} in Eq. 10. Tab. 2 demonstrates that our method outperforms the latest approaches, while Fig. 4 (a) highlights our more accurate reconstruction, especially on planar structures.

Replica. We evaluate our method on synthetic Replica scenes, following the MonoSDF setup and using monocular cues predicted by the pretrained Omnidata model. The scale and shift are solved as in Tab. 1. Tab. 3 shows our superiority over MonoSDF, both with and without Replica pretraining ("Train split" and "Test split"). Additionally, we compare against SOTA dense monocular SLAM methods (DROID-SLAM (Teed and Deng 2021), NICER-SLAM (Zhu et al. 2023)) and the RGB-D SLAM system NICE-SLAM (Zhu et al. 2022), retrained under our settings. Tab. 4 demonstrates that our method significantly outperforms these approaches across all metrics. Visual comparisons with NeuralRGBD in Fig. 4 (b) further highlight our more accurate reconstructions.

Tanks and Temples. We follow the same experimental setting in MonoSDF (Yu et al. 2022), and apply monocular cues during optimization. Since the GT meshes are not publicly available, we only compare visual results with MonoSDF. The visual comparison in Fig. 6 show that our method can recover more accurate geometry details.

Analysis. For methods learning implicit representations from multi-view images, the most challenging problem is to infer the accurate zero level set of the implicit function. Our significant improvements over the latest methods mainly come from the fact that our surface constraints address this challenge quite well. We visualize a cross section of the reconstructed scenes and GT meshes in Fig. 5. We can see that our surface (in yellow) is much closer to the GT mesh (in grey) than MonoSDF, although both MonoSDF and ours reconstruct complete and smooth surfaces. Since MonoSDF can not recover the zero level set accurate, its reconstructed surface is inflated, not only near this cross section but also other places in the scene, which leads to the floating surface over the GT mesh with large errors.

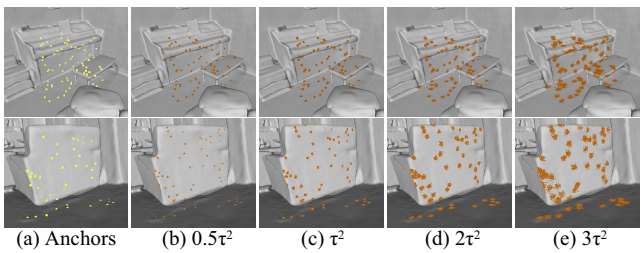


Figure 7: Comparisons on sizes of patch surfaces.

Ablation Studies

Ablation studies on ScanNet released by MonoSDF justify the effectiveness of modules in our method.

Loss. We first justify the effectiveness of losses in Tab. 6. We mainly focus on the losses for surface constraints, since the effectiveness of losses for volume rendering have been widely justified in previous studies. Compared to the baseline, each loss for a surface constraint can improve the performance. Fig. 8 shows that each loss may improve different aspects, for example, the depth consistency makes the surface smoother, the photometric consistency makes the surface more compact, and the surface fitting loss also contributes to the smoothness of the surface.

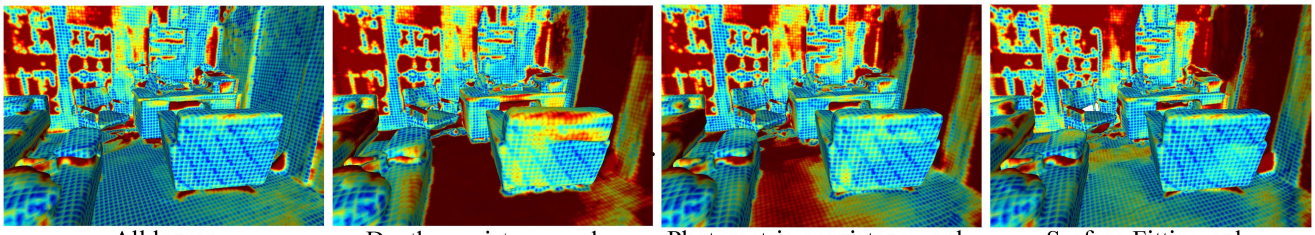
We also try larger weights or smaller weights on all the three losses for surface constraints. The results of "10×" and "0.1×" show that weighting more or less on surface constraints do not balance well with the constraints through volume rendering and improve the reconstruction accuracy.

Point Number J in a Surface Patch s. We explore the effect of point number J on the performance. We try different densities $J = \{1, 9, 16, 25\}$ on surface patches by sampling queries using the same Gaussian distribution. The numerical comparison in Tab. 5 shows that a too small number of queries may not be able to represent a surface, such as $J = 1$, while a too large number of points may not improve the performance further but increase the time complexity. We also compare our patch-level photo consistency with the pixel-level photo consistency. The results of "Pixels" show that patch-level photo consistency achieves better performance.

Guassian Distribution for Sampling. We also conduct an experiment to explore the effect of the variance τ^2 of the Guassian distribution on the performance. We initially set τ^2 to be the distance between two other pixels in the 3D world coordinate, and use it as a baseline. We try different variance candidates including $\{0.5\tau^2, \tau^2, 2\tau^2, 3\tau^2\}$ to sample $J = 9$ points around each anchor. The comparison in Tab. 5 shows that a small variance may not cover a large enough area which degenerates the performance of inference while a large variance covers a too large area where points may be ruled out by our masks. This may decrease the efficiency and also degenerate the performance. We visualize sizes of these

		room_0	room_1	room_2	office_0	office_1	office_2	office_3	office_4	Mean
RGB-D input										
NICE-SLAM	Acc[cm]↓	3.53	3.60	3.03	5.56	3.35	4.71	3.84	3.35	3.87
	Comp[cm]↓	3.40	3.62	3.27	4.55	4.03	3.94	3.99	4.15	3.87
	Recall ↑	86.05	80.75	87.23	79.34	82.13	80.35	80.55	82.88	82.41
	Normal C.↑	91.92	91.36	90.79	89.30	88.79	88.97	87.18	91.17	89.93
RGB monocular input										
DROID-SLAM	Acc[cm]↓	12.18	8.35	3.26	3.01	2.39	5.66	4.49	4.65	5.50
	Comp[cm]↓	8.96	6.07	16.01	16.19	16.20	15.56	9.73	9.63	12.29
	Recall↑	60.07	76.20	61.62	64.19	60.63	56.78	61.95	67.51	63.62
	Normal C.↑	72.81	74.71	79.21	77.53	78.57	75.79	77.69	76.38	76.59
NICER-SLAM	Acc[cm]↓	**2.53**	3.93	3.40	5.49	3.45	4.02	3.34	3.03	3.65
	Comp[cm]↓	3.04	4.10	3.42	6.09	4.42	4.29	4.03	3.87	4.16
	Recall ↑	88.75	76.61	86.10	65.19	77.84	74.51	82.01	83.98	79.37
	Normal C.↑	93.00	91.52	92.38	87.11	86.79	90.19	90.10	90.96	90.27
OURS	Acc[cm]↓	2.54	**1.75**	**2.41**	**2.24**	**1.70**	**2.78**	**2.90**	**2.31**	**2.33**
	Comp[cm]↓	**2.08**	**2.54**	**2.58**	**2.89**	**3.11**	**2.94**	**3.11**	**2.77**	**2.75**
	Recall↑	**96.00**	**91.99**	**93.00**	**88.59**	**90.34**	**86.99**	**88.99**	**91.59**	**90.93**
	Normal C.↑	**95.60**	**94.20**	**94.36**	**90.47**	**92.79**	**92.40**	**92.19**	**94.49**	**93.31**

Table 4: Numerical comparison in each scene on Replica.



All losses | Depth consistency only | Photometric consistency only | Surface Fitting only

Figure 8: Effect of losses. We use MonoSDF as a baseline and apply one loss each time.

J,τ^2	Acc↓	Comp↓	CD-L1↓	Prec↑	Recall↑	F-score↑
1	0.037	0.040	0.039	0.781	0.738	0.760
9	**0.036**	**0.039**	**0.037**	**0.820**	0.777	**0.797**
16	0.037	0.039	0.038	0.802	0.754	0.777
25	0.037	0.041	0.039	0.789	0.732	0.759
Pixels	0.037	**0.039**	0.038	0.801	**0.787**	0.794
$0.5\tau^2$	0.037	0.040	0.039	0.783	0.743	0.763
τ^2	**0.036**	**0.039**	**0.037**	**0.820**	**0.777**	**0.797**
$2\tau^2$	**0.036**	**0.039**	**0.037**	0.805	0.757	0.781
$3\tau^2$	0.037	0.040	0.038	0.793	0.738	0.766

Table 5: Effect of point number J and variance τ^2.

	Acc↓	Comp↓	CD-L1↓	Prec↑	Recall↑	F-score↑
Baseline	**0.035**	0.048	0.042	0.799	0.681	0.733
Only L_{DC}	0.039	0.041	0.040	0.778	0.753	0.766
Only L_{NCC}	0.036	**0.039**	**0.037**	0.801	0.744	0.772
Only L_{Fit}	0.037	**0.039**	0.038	0.794	**0.777**	0.785
$10\times$	0.041	0.043	0.042	0.764	0.753	0.759
$0.1\times$	0.039	0.045	0.042	0.775	0.712	0.743
No w_j	0.038	**0.039**	0.038	0.790	0.760	0.770
No η_j	0.036	0.040	0.038	0.801	0.768	0.784
No pulling	0.037	0.041	0.039	0.772	0.734	0.753
Ours-Full	0.036	**0.039**	**0.037**	**0.820**	**0.777**	**0.797**

Table 6: Effect of Losses and pulling.

patch surfaces in Fig. 7, where all patch surfaces tightly locate on the reconstructed surfaces.

Mask w_j and Weight η_j. We highlight the effect of the mask w_j and the weight η_j by removing one of them each time. We report the comparison in Tab. 6. The decreased results "No w_j" and "No η_j" show that w_j can rule out outliers during pulling while η_j can make network focus more on the most important p'_j on the patch surface.

The effect of pulling. We conduct an experiment to demonstrate that the pulling mechanism can boost the performance. We conduct this experiment without pulling by imposing depth, photometric consistency and surface fitting directly on 3D anchor q rather than the sensed surface patch. As shown in Tab. 6, without pulling, the overall performance decreases.

Conclusion

We propose a method to infer SDFs from multi-view images via volume rendering with surface patch sensing. By using SDF to define a local patch around the estimated ray-surface intersection, we directly apply surface constraints. Leveraging gradients and signed distances, we pull sampled points onto the zero level set, enabling explicit surface constraints that enhance accuracy and capture finer geometry details. Numerical and visual comparisons demonstrate our superiority over recent methods, validated across widely used benchmarks.

References

Azinović, D.; Martin-Brualla, R.; Goldman, D. B.; Nießner, M.; and Thies, J. 2022. Neural RGB-D Surface Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 6290–6301.

Ben-Shabat, Y.; Koneputugodage, C. H.; and Gould, S. 2021. DiGS : Divergence guided shape implicit neural representation for unoriented point clouds. *CoRR*, abs/2106.10811.

Chen, C.; Liu, Y.-S.; and Han, Z. 2022. Latent Partition Implicit with Surface Codes for 3D Representation. In *European Conference on Computer Vision*.

Chen, C.; Liu, Y.-S.; and Han, Z. 2023a. GridPull: Towards Scalability in Learning Implicit Representations from 3D Point Clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Chen, C.; Liu, Y.-S.; and Han, Z. 2023b. Unsupervised Inference of Signed Distance Functions from Single Sparse Point Clouds without Learning Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vsion and Pattern Recognition*.

Chen, C.; Liu, Y.-S.; and Han, Z. 2024. Inferring Neural Signed Distance Functions by Overfitting on Single Noisy Point Clouds through Finetuning Data-Driven based Priors. In *Advances in Neural Information Processing Systems*.

Chen, H.; Li, C.; and Lee, G. H. 2023. NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance. arXiv:2312.00846.

Chou, G.; Chugunov, I.; and Heide, F. 2022. GenSDF: Two-Stage Learning of Generalizable Signed Distance Functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T. A.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. *CoRR*, abs/1702.04405.

Eftekhar, A.; Sax, A.; Malik, J.; and Zamir, A. 2021. Omnidata: A Scalable Pipeline for Making Multi-Task Mid-Level Vision Datasets From 3D Scans. In *International Conference on Computer Vision*, 10786–10796.

Fu, Q.; Xu, Q.; Ong, Y.-S.; and Tao, W. 2022. Geo-Neus: Geometry-Consistent Neural Implicit Surfaces Learning for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems*.

Guo, H.; Peng, S.; Lin, H.; Wang, Q.; Zhang, G.; Bao, H.; and Zhou, X. 2022. Neural 3D Scene Reconstruction with the Manhattan-world Assumption. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Guédon, A.; and Lepetit, V. 2023. SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering. arXiv:2311.12775.

Hu, P.; and Han, Z. 2023. Learning Neural Implicit through Volume Rendering with Attentive Depth Fusion Priors. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, SIGGRAPH '24. ACM.

Jiang, S.; Hua, J.; and Han, Z. 2023. Coordinate Quantized Neural Implicit Representations for Multi-view 3D Reconstruction. In *IEEE International Conference on Computer Vision*.

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics*, 36(4).

Laurentini, A. 1994. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2): 150–162.

Li, Z.; Müller, T.; Evans, A.; Taylor, R. H.; Unberath, M.; Liu, M.-Y.; and Lin, C.-H. 2023. Neuralangelo: High-Fidelity Neural Surface Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, S.-L.; Guo, H.-X.; Pan, H.; Wang, P.; Tong, X.; and Liu, Y. 2021. Deep Implicit Moving Least-Squares Functions for 3D Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Lorensen, W. E.; and Cline, H. E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4): 163–169.

Ma, B.; Han, Z.; Liu, Y.-S.; and Zwicker, M. 2021. Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces. In *International Conference on Machine Learning*.

Ma, B.; Liu, Y.-S.; Zwicker, M.; and Han, Z. 2022a. Reconstructing Surfaces for Sparse Point Clouds with On-Surface Priors. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ma, B.; Liu, Y.-S.; Zwicker, M.; and Han, Z. 2022b. Surface Reconstruction from Point Clouds by Learning Predictive Context Priors. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ma, B.; Zhou, J.; Liu, Y.-S.; and Han, Z. 2023. Towards Better Gradient Consistency for Neural Signed Distance Functions via Level Set Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vsion and Pattern Recognition*.

Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *arXiv:2201.05989*.

Niemeyer, M.; Mescheder, L.; Oechsle, M.; and Geiger, A. 2020. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Oechsle, M.; Peng, S.; and Geiger, A. 2021. UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. In *International Conference on Computer Vision*.

Schönberger, J. L.; and Frahm, J.-M. 2016. Structure-from-Motion Revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schönberger, J. L.; Zheng, E.; Pollefeys, M.; and Frahm, J.-M. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision*.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; Clarkson, A.; Yan, M.; Budge, B.; Yan, Y.; Pan, X.; Yon, J.; Zou, Y.; Leon, K.; Carter, N.; Briales, J.; Gillingham, T.; Mueggler, E.; Pesqueira, L.; Savva, M.; Batra, D.; Strasdat, H. M.; Nardi, R. D.; Goesele, M.; Lovegrove, S.; and Newcombe, R. A. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *CoRR*, abs/1906.05797.

Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; and Fidler, S. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tang, J.; Lei, J.; Xu, D.; Ma, F.; Jia, K.; and Zhang, L. 2021. SA-ConvONet: Sign-Agnostic Optimization of Convolutional Occupancy Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Teed, Z.; and Deng, J. 2021. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*.

Vicini, D.; Speierer, S.; and Jakob, W. 2022. Differentiable Signed Distance Function Rendering. *ACM Transactions on Graphics*, 41(4): 125:1–125:18.

Wang, J.; Wang, P.; Long, X.; Theobalt, C.; Komura, T.; Liu, L.; and Wang, W. 2022. NeuRIS: Neural Reconstruction of Indoor Scenes Using Normal Priors. In *European Conference on Computer Vision*.

Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; and Wang, W. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems*, 27171–27183.

Wang, Y.; Skorokhodov, I.; and Wonka, P. 2022. HF-NeuS: Improved Surface Reconstruction Using High-Frequency Details.

Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *European Conference on Computer Vision*.

Yariv, L.; Gu, J.; Kasten, Y.; and Lipman, Y. 2021. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*.

Yariv, L.; Kasten, Y.; Moran, D.; Galun, M.; Atzmon, M.; Ronen, B.; and Lipman, Y. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems*, 33.

Yu, Z.; Peng, S.; Niemeyer, M.; Sattler, T.; and Geiger, A. 2022. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction. *ArXiv*, abs/2022.00665.

Yu, Z.; Sattler, T.; and Geiger, A. 2024. Gaussian Opacity Fields: Efficient and Compact Surface Reconstruction in Unbounded Scenes. arXiv:2404.10772.

Zhang, W.; Liu, Y.-S.; and Han, Z. 2024. Neural Signed Distance Function Inference through Splatting 3D Gaussians Pulled on Zero-Level Set. In *Advances in Neural Information Processing Systems*.

Zhang, W.; Shi, K.; Liu, Y.-S.; and Han, Z. 2024. Learning Unsigned Distance Functions from Multi-view Images with Volume Rendering Priors. In *European Conference on Computer Vision*.

Zhou, J.; Ma, B.; Liu, Y.-S.; Fang, Y.; and Han, Z. 2022. Learning Consistency-Aware Unsigned Distance Functions Progressively from Raw Point Clouds. In *Advances in Neural Information Processing Systems*.

Zhu, Z.; Peng, S.; Larsson, V.; Cui, Z.; Oswald, M. R.; Geiger, A.; and Pollefeys, M. 2023. NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM. *CoRR*, abs/2302.03594.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition*.