VolRecon: Volume Rendering of Signed Ray Distance Functions for Generalizable Multi-View Reconstruction

Yufan Ren[1]* Fangjinhua Wang[2]* Tong Zhang[1] Marc Pollefeys[2] Sabine Süsstrunk[1]
[1]IVRL IC EPFL [2]Department of Computer Science, ETH Zurich

Abstract

The success of the Neural Radiance Fields (NeRF) in novel view synthesis has inspired researchers to propose neural implicit scene reconstruction. However, most existing neural implicit reconstruction methods optimize per-scene parameters and therefore lack generalizability to new scenes. We introduce VolRecon, a novel generalizable implicit reconstruction method with Signed Ray Distance Function (SRDF). To reconstruct the scene with fine details and little noise, VolRecon combines projection features aggregated from multi-view features, and volume features interpolated from a coarse global feature volume. Using a ray transformer, we compute SRDF values of sampled points on a ray and then render color and depth. On DTU dataset, VolRecon outperforms SparseNeuS by about 30% in sparse view reconstruction and achieves comparable accuracy as MVSNet in full view reconstruction. Furthermore, our approach exhibits good generalization performance on the large-scale ETH3D benchmark. Code is available at https://github.com/IVRL/VolRecon/.

1. Introduction

The ability to reconstruct 3D geometries from images or videos is crucial in various applications in robotics [16, 43, 52] and augmented/virtual reality [29, 35]. Multi-view stereo (MVS) [13, 15, 39, 47, 54, 55] is a commonly used technique for this task. A typical MVS pipeline involves multiple steps, *i.e.*, multi-view depth estimation, filtering, and fusion [5, 13].

Recently, there has been a growing interest in neural implicit representations for various 3D tasks, such as shape modeling [28, 36], surface reconstruction [49, 56], and novel view synthesis [30]. NeRF [30], a seminal work in this area, employs Multi-Layer Perceptrons (MLP) to model a radiance field, producing volume density and radiance estimates for a given position and viewing direction. While NeRF's scene representation and volume rendering approach has proven effective for tasks such as novel view synthesis, it cannot generate accurate surface reconstruction due to difficulties in finding a universal density threshold for surface extraction [56]. To address this, researchers have proposed neural implicit reconstruction using the Signed Distance Function (SDF) for geometry representation and modeling the volume density function [49, 56]. However, utilizing SDF with only color supervision leads to unsatisfactory reconstruction quality compared to MVS methods [15, 54] due to a lack of geometry supervision and potential radiance-geometry ambiguities [51, 62]. As a result, subsequent works have sought to improve reconstruction quality by incorporating additional priors, such as sparse Struction-from-Motion (SfM) point clouds [11], dense MVS point clouds [60], normals [48, 59], and depth maps [59].

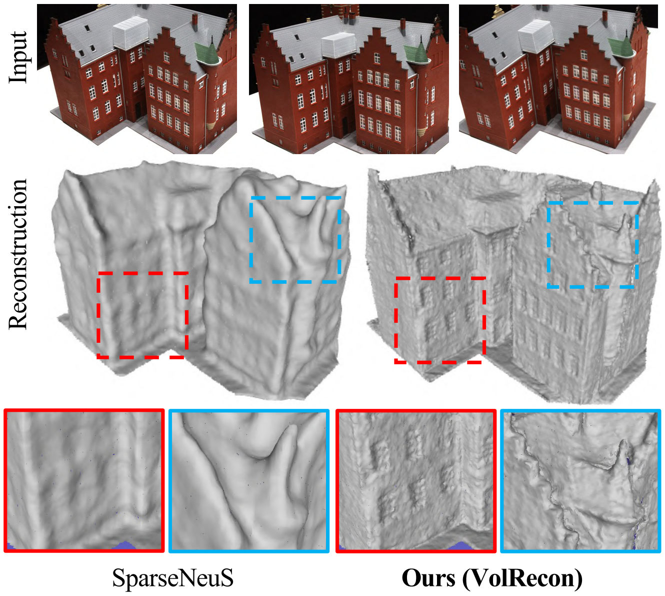

Figure 1. Generalizable implicit reconstructions from three views (top). The state-of-the-art method SparseNeuS [26] produces oversmoothed surfaces (left), while **our (VolRecon)** reconstructs finer details (right). Best viewed on a screen when zoomed in.

Many neural implicit reconstruction methods are restricted to optimizing one model for a particular scene and cannot be applied to new, unseen scenes, *i.e.*, across-scene

*Equal contribution

generalization. However, the ability to generalize learned priors to new scenes is valuable in challenging scenarios such as reconstruction with sparse views [4, 50, 58]. In order to achieve across-scene generalization in neural implicit reconstruction, it is insufficient to simply input the spatial coordinate of a point as NeRF. Instead, we need to incorporate information about the scene, such as the points' projection features on the corresponding images [4, 50, 58]. SparseNeuS [26] recently achieved across-scene generalization in implicit reconstruction with global feature volumes [4]. Despite achieving promising results, SparseNeuS is limited by the resolution of the feature volume due to the memory constraints [22, 32], leading to over-smoothing surfaces even with a higher resolution feature volume, Fig. 1.

In this paper, we propose VolRecon, a novel framework for generalizable neural implicit reconstruction using the Signed Ray Distance Function (SRDF). Unlike SDF, which defines the distance to the nearest surface along any directions, SRDF [63] defines the distance to the nearest surface along a given ray. We utilize a projection-based approach to gather local information about surface location. We first project each point on the ray into the feature map of each source view to interpolate multi-view features. Then, we aggregate the multi-view features to *projection features* using a view transformer. However, when faced with challenging situations such as occlusions and textureless surfaces, determining the surface location along the ray with only local information is difficult. To address this, we construct a coarse global feature volume that encodes global shape priors like SparseNeuS [26, 32]. We use the interpolated features from the global feature volume, *i.e.*, *volume features*, and *projection features* of all the sampled points along the ray to compute their SRDF values, with a ray transformer. Similar to NeuS [49], we model the density function with SRDF and then estimate the image and depth map with volume rendering.

Extensive experiments on DTU [1] and ETH3D [40] verify the effectiveness and generalization ability of our method. On DTU, our method outperforms the state-of-the-art method SparseNeuS [26] by 30% in sparse view reconstruction and 22% in full view reconstruction. Furthermore, our method performs better than the MVS baseline COLMAP [39]. Compared with MVSNet [54], a seminal learning-based MVS method, our method performs better in the depth evaluation and has comparable accuracy in full-view reconstruction. On the ETH3D benchmark [40], we show that our method has a good generalization ability to large-scale scenes.

In summary, our contributions are as follows:

- We propose VolRecon, a new pipeline for generalizable implicit reconstruction that produce detailed surfaces.
- Our novel framework comprises a view transformer to aggregate multi-view features and a ray transformer to compute SRDF values of all the points along a ray.
- We introduce a combination of local projection features and global volume features, which enables the reconstruction of surfaces with fine details and high quality.

2. Related Work

Neural Implicit Reconstruction. Traditional volumetric reconstructions [5, 18, 33] use implicit signed distance fields to produce high-quality reconstructions. Recent works use networks to model shapes as continuous decision boundaries, *i.e.*, occupancy functions [28, 37] or SDF [36]. In NeRF [30], the authors further show that combining neural implicit functions, *e.g.*, Multi-Layer Perceptron (MLP), and volume rendering can achieve photo-realism in novel view synthesis [2, 3, 10, 30, 31]. Since NeRF [30], which originally targets a per-scene optimization problem, several additional methods [4, 50, 58] are proposed to perform generalizable novel view synthesis for unseen scenes. For example, IBRNet [50] projects sampled points along the ray into multiple source views. It aggregates multi-view features into density features and uses a ray transformer, which inputs the density features for all points along the ray to predict the density for each point. For multi-view reconstruction, IDR [57] reconstructs surfaces by representing the geometry as the zero-level set of an MLP, requiring accurate object masks. To avoid using masks, VolSDF [56] and NeuS [49] incorporate SDF in neural volume rendering, using it to modify the density function. Additional geometric priors [11, 48, 59, 60] were proposed to improve the reconstruction quality. Nevertheless, these methods usually require a lengthy optimization for each scene and cannot generalize to unseen scenes.

Recently, SparseNeuS [26] attempts to solve across-scene generalization for surface reconstruction. Similar to [4, 32, 42], SparseNeuS constructs fixed-resolution feature volumes to aggregate image features from multi-view images. An MLP takes the coordinates and corresponding interpolated features from the feature volumes to predict the SDF values as input. SparseNeuS needs high-resolution volumes, *i.e.*, 192^3, but still outputs over-smoothed surfaces. In contrast, we additionally use the *projection feature* that contains local features and use a ray transformer to aggregate features of sampled points along a ray. In this way, our VolRecon model captures both local and global information to achieve finer detail and less noise than SparseNeuS, Fig. 1.

Multi-view Stereo. Based on scene representations, traditional MVS methods fall into three main categories: volumetric [22, 23, 41], point cloud-based [12, 24], and depth map-based [13, 39, 53]. Depth map-based methods are more flexible since they decouple the problem into depth map estimation and fusion [13, 39]. Therefore, most recent learning-

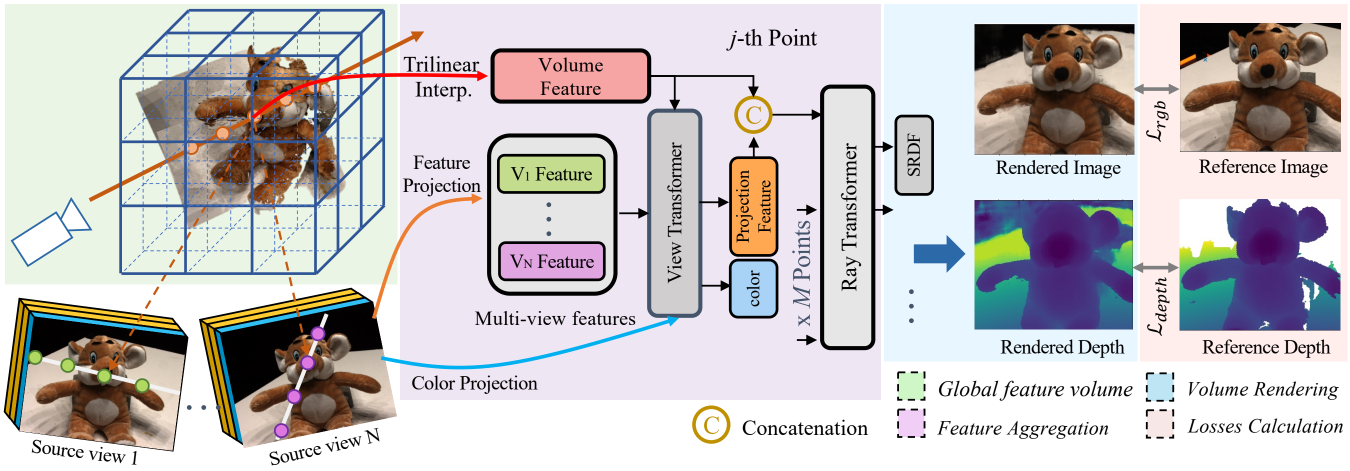

Figure 2. **Structure of VolRecon**. For a set of N source views, we first extract the image features. Then we build a global feature volume to obtain global shape priors. Next, given a ray in the target viewpoint, we project each sampled point on the ray into the source views, aggregate its multi-view features using a view transformer to get the projection feature and blended color of each point. After that, we apply the ray transformer to concatenated features of all the M points along the ray to predict their SRDF values. Finally, we volume render the color and depth. Best viewed on a screen when zoomed in.

based MVS methods [15, 46, 47, 54, 55, 61] perform multi-view depth estimation and then fuse them to a point cloud, which achieves impressive performance on various benchmarks [1, 21, 40]. Note that while much progress has been made in neural implicit reconstruction, the reconstruction performance [26, 49, 56, 57] is still not on par with the state-of-the-art MVS baselines. Yet, our method performs better than COLMAP in few view reconstruction and achieves comparable accuracy as MVSNet [54] in full view reconstruction.

3. Method

In this section, we discuss the structure of VolRecon, illustrated in Fig. 2. The pipeline consists of predicting the Signed Ray Distance Function (SRDF) (Sec. 3.1), volume rendering of the SRDF to predict color and depth (Sec. 3.2), and loss functions (Sec. 3.3).

3.1. SRDF Prediction

Signed Ray Distance Function. Let set $\Omega \in \mathbb{R}^3$ denotes the space and $\mathcal{M} = \partial\Omega$ its boundary surface. The Signed Distance Function $d_\Omega(\mathbf{p})$ defines the shortest distance of a point $\mathbf{p} \in \mathbb{R}^3$ to the surface $\mathcal{M}$. Its sign denotes whether $\mathbf{p}$ is outside (positive) or inside (negative) of the surface,

$$\mathbf{1}_\Omega(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p} \in \Omega \\ 0 & \text{if } \mathbf{p} \notin \Omega \end{cases}, \tag{1}$$

$$d_\Omega(\mathbf{p}) = (-1)^{\mathbf{1}_\Omega(\mathbf{p})} \min_{\mathbf{p}^* \in \mathcal{M}} \|\mathbf{p} - \mathbf{p}^*\|_2, \tag{2}$$

where $||\cdot||_2$ is the L_2-norm and $\mathbf{p}^*$ are points on the surface. Differently, SRDF [5, 63] defines the shortest distance to surface $\mathcal{M}$ along a ray direction $\mathbf{v}$ ($||\mathbf{v}||_2 = 1$),

$$\tilde{d}_\Omega(\mathbf{p}, \mathbf{v}) = (-1)^{\mathbf{1}_\Omega(\mathbf{p})} \min_{\mathbf{p}^* \in \mathcal{M}, \frac{\mathbf{p}^* - \mathbf{p}}{||\mathbf{p}^* - \mathbf{p}||_2} = \mathbf{v}} \|\mathbf{p} - \mathbf{p}^*\|_2. \tag{3}$$

Theoretically, given a point $\mathbf{p}$, its SDF $d_\Omega(\mathbf{p})$ equals to the SRDF $\tilde{d}_\Omega(\mathbf{p}, \mathbf{v})$ with the minimum absolute value in any direction $\mathbf{v}$:

$$d_\Omega(\mathbf{p}) = (-1)^{\mathbf{1}_\Omega(\mathbf{p})} \min_{\mathbf{v}}(|\tilde{d}_\Omega(\mathbf{p}, \mathbf{v})|). \tag{4}$$

Similar to SDF volume rendering [49, 56], we incorporate SRDF in volume rendering to estimate the depth map from the given viewpoints, which can be fused into mesh [5] or dense point clouds [39].

Feature Extraction. Given the source image set $\mathbb{I} = \{\mathbf{I}_1, \cdots, \mathbf{I}_N\}$, where $\mathbf{I} \in \mathbb{D}^{H \times W \times 3}$, $\mathbb{D} \subset [0, 1]$, and H, W are the image height and width, respectively. We use a Feature Pyramid Network [25] to extract feature maps $\{\mathbf{F}_i\}_{i=1}^N \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}$.

Global Feature Volume. We construct a global feature volume $\mathbf{F}_v$ similar to [32, 42] to get global information. Specifically, we first divide the bounding volume of the scene into K^3 voxels. The center point of each voxel is projected onto the feature map of each source view to obtain the features. This is done using bilinear interpolation, where the mean and variance of N features are computed and concatenated as the voxel features. We then use 3D U-Net [38] to regularize and aggregates the information. For each point $\mathbf{p}$, we denote the interpolated feature from $\mathbf{F}_v$ as *volume feature*, $\mathbf{f}_v$. Please refer to supplementary for more details.

View Transformer. Given a pixel in the reference view, we denote the M points on the ray emitted from this pixel as $\{\mathbf{p}(t) = \mathbf{o} + t\mathbf{v}, t \geq 0\}$. By projecting each point $\mathbf{p}$ onto the feature map of each source view, we extract colors $\{\mathbf{c}_i\}_{i=1}^N$ and features $\{\mathbf{f}_i\}_{i=1}^N$ using bilinear interpolation. We apply a *view transformer* to aggregate the multi-view features $\{\mathbf{f}_i\}_{i=1}^N$ into one feature, which we denote as the *projection feature*. Structurally, we use a self-attention transformer [44] with linear attention [19]. Following previous work [7], we add a learnable aggregation token, denoted as $\mathbf{f}_0$, to obtain the projection feature. Since no order of source views is assumed, we do not use positional encoding in the view transformer. The projection feature $\mathbf{f}_p$ and updated multi-view features $\{\mathbf{f'}_i\}_{i=1}^N$ are computed as,

$$\mathbf{f}_p, \{\mathbf{f'}_i\}_{i=1}^N = \text{ViewTrans}(\mathbf{f}_0, \{\mathbf{f}_i\}_{i=1}^N). \tag{5}$$

Visibility is important in multi-view aggregation [39, 47] due to the existence of occlusions. Therefore, mean and variance aggregation [50, 54] may not be robust enough since all views are accounted equally. Using a learnable transformer enables the model to reason about the consistency for aggregation across multiple views.

Ray Transformer. Similar to SDF, SRDF is not locally defined and its value depends on the closest surface along the ray. To provide such non-local information of other points along the ray, we additionally design a *ray transformer* based on linear attention [19]. We first concatenate the projection feature and corresponding volume feature into a combined feature to add global shape prior. After ordering the points in a sequence from near to far, the ray transformer applies positional encoding [50] and self-attention on the combined feature to predict attended features $\{\tilde{\mathbf{f}}_j\}_{j=1}^M$,

$$\{\tilde{\mathbf{f}}_j\}_{j=1}^M = \text{RayTrans}(\{\text{cat}(\mathbf{f}_v, \mathbf{f}_p, \gamma)\}_{j=1}^M), \tag{6}$$

where $\text{cat}(\cdot)$ denotes concatenation and γ positional encoding. Finally, we use an MLP to decode the attended feature to SRDF for each point on the ray.

3.2. Volume Rendering of SRDF

Color Blending. For a point $\mathbf{p}$ at viewing direction $\mathbf{v}$, we blend colors of N source views, $\{\mathbf{c}_i\}_{i=1}^N$, similar to [45, 50]. We compute the blending weight using the updated multi-view features $\{\mathbf{f'}_i\}_{i=1}^N$ from the view transformer. Similar to [45, 50], we concatenate $\{\mathbf{f'}_i\}_{i=1}^N$ with the difference between $\mathbf{v}$ and the viewing direction in the i-th source view, $\mathbf{v}_i$. Then we pass the concatenated features through an MLP and use *Softmax* to get the blending weights $\{\eta_i\}_{i=1}^N$. The final radiance at point $\mathbf{p}$ and viewing direction $\mathbf{v}$ is the weighted sum of $\{\mathbf{c}_i\}_{i=1}^N$,

$$\hat{\mathbf{c}} = \sum_{i=1}^{N} \eta_i \cdot \mathbf{c}_i. \tag{7}$$

Volume rendering. Several works [49, 56] propose to include SDF in volume rendering for implicit reconstruction with the supervision of the pixel reconstruction loss. We adopt the method of NeuS [49] to volume render SRDF, as briefly introduced below. We provide a comparison between rendering SDF and SRDF in supplementary.

Specifically, the color is accumulated along the ray

$$\hat{\mathbf{C}} = \sum_{j=1}^{M} T_j \alpha_j \hat{\mathbf{c}}_j, \tag{8}$$

where $T_j = \prod_{k=1}^{j-1}(1-\alpha_k)$ is the discrete *accumulative transmittance*, and α_j are discrete opacity values defined by

$$\alpha_j = 1 - \exp(-\int_{t_j}^{t_{j+1}} \rho(t)dt), \tag{9}$$

where opaque density $\rho(t)$ is similar to the original definition in NeuS [49]. The difference is that we replace the original SDF with SRDF in $\rho(t)$. For more theoretical details, please refer to [49].

Similar to volume rendering of colors, we can derive the rendered depth as

$$\hat{\mathbf{D}} = \sum_{j=1}^{M} T_j \alpha_j t_j. \tag{10}$$

3.3. Loss Function

We define the loss function as

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \alpha \mathcal{L}_{\text{depth}}. \tag{11}$$

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

$$\mathcal{L}_{\text{color}} = \frac{1}{S} \sum_{s=1}^{S} \left\| \hat{\mathbf{C}}_s - \mathbf{C}_s \right\|_2, \tag{12}$$

where S is the number of pixels and $\mathbf{C}_s$ is the ground truth color.

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

$$\mathcal{L}_{\text{depth}} = \frac{1}{S_1} \sum_{s=1}^{S_1} |\hat{\mathbf{D}}_s - \mathbf{D}_s|, \tag{13}$$

where S_1 is the number of pixels with valid depth and $\mathbf{D}_s$ is the ground truth depth. In our experiments, we choose $\alpha = 1.0$.

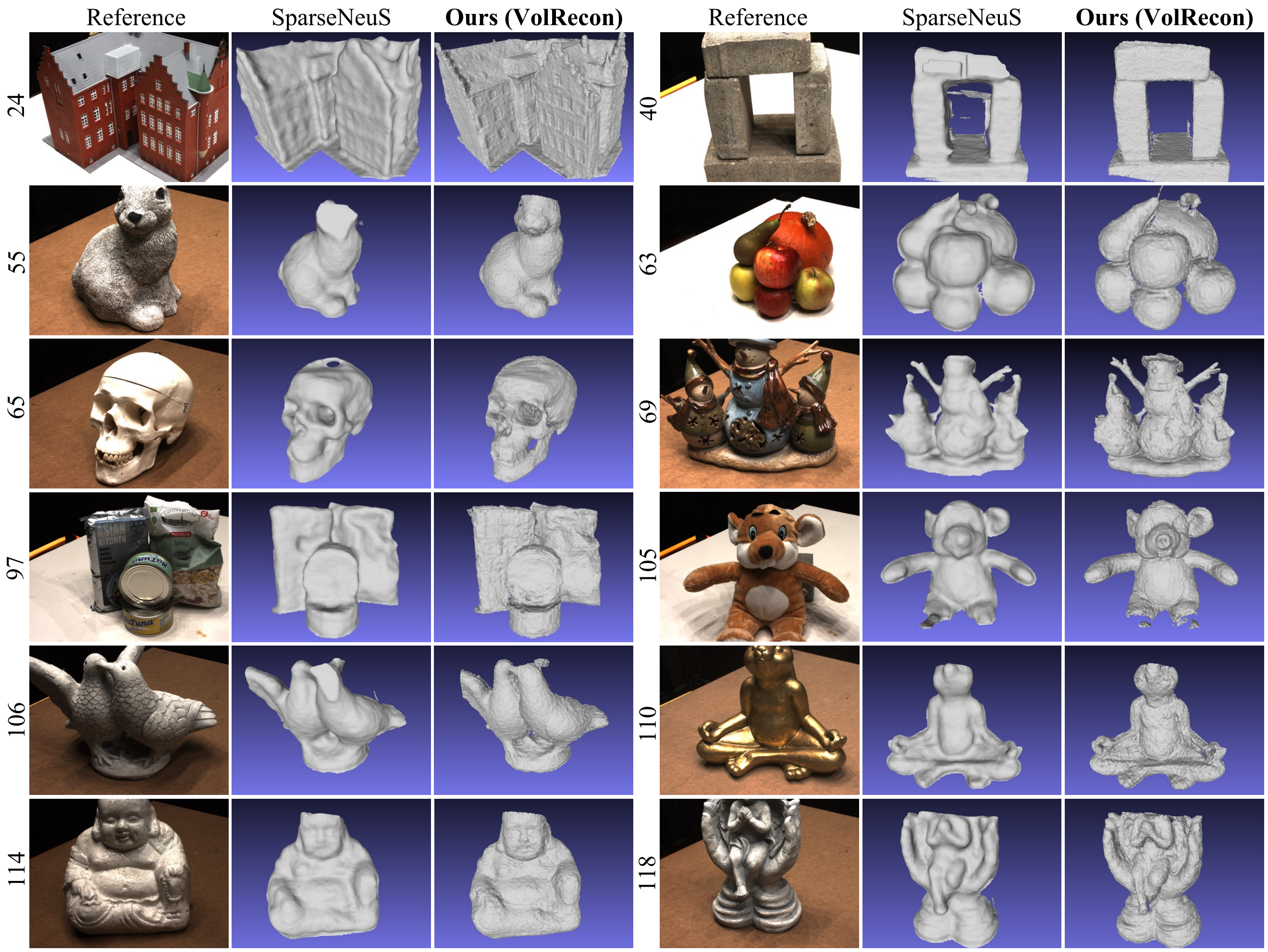

Figure 3. **Sparse view** ($N = 3$) reconstruction on testing scenes in the DTU benchmark [1]. While SparseNeuS [26] produces over-smoothed surfaces, our method (VolRecon) reconstructs with finer details, *e.g.*, scene 24 windows and scene 63 fruit stalks, and sharper boundaries, *e.g.*, scene 97 cans touching part and scene 118 sculpture base, than SparseNeuS [26]. Best viewed on a screen when zoomed in.

4. Experiments

4.1. Experimental Settings

Datasets. Following existing works [11, 26, 49, 56], we use the DTU dataset [1] for training. The DTU dataset [1] is an indoor multi-view stereo dataset with ground truth point clouds of 124 different scenes and 7 different lighting conditions. During experiments, we use the same 15 scenes as SparseNeuS for testing and use the remaining scenes for training. We use the depth maps rendered from the mesh [54] as depth map ground truth. Besides DTU, we also use the ETH3D dataset [40] to test the generalization ability of our method. ETH3D [40] is a challenging MVS benchmark consisting of high-resolution images of real-world large-scale scenes with strong viewpoint variations.

Implementation details. We implement our model in PyTorch [17] and PyTorch Lightning [9]. During training, we use an image resolution of 640×512 and set the number of source images to $N = 4$. We train our model for 16 epochs using Adam [20] on one A100 GPU. The learning rate is set to 10^{-4}. The ray number sampled per batch and the batch size are set to 1024 and 2, respectively. Similar to other volume rendering methods [30, 49], we use a hierarchical sampling strategy in both training and testing. We first uniformly sample N_{coarse} points on the ray and then conduct importance sampling to sample another N_{fine} points on top of the coarse probability estimation. We set $N_{\text{coarse}} = 64$ and $N_{\text{fine}} = 64$ during our experiments. For global feature volume $\mathbf{F}_v$, we set the resolution as $K = 96$. During testing, we set the image resolution to 800×600.

Baselines. We mainly compare our method with: (1) SparseNeuS [26], the state-of-the-art generalizable neural

Scan	Mean↓	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
COLMAP [39]	1.52	**0.90**	2.89	1.63	1.08	2.18	1.94	1.61	1.30	2.34	1.28	1.10	1.42	0.76	1.17	**1.14**
MVSNet [54]	**1.22**	1.05	**2.52**	1.71	**1.04**	1.45	**1.52**	**0.88**	**1.29**	**1.38**	**1.05**	**0.91**	**0.66**	**0.61**	**1.08**	1.16
IDR [57]	3.39	4.01	6.40	3.52	1.91	3.96	2.36	4.85	1.62	6.37	5.97	1.23	4.73	0.91	1.72	1.26
VolSDF [56]	3.41	4.03	4.21	6.12	0.91	8.24	1.73	2.74	1.82	5.14	3.09	2.08	4.81	0.60	3.51	2.18
UNISURF [34]	4.39	5.08	7.18	3.96	5.30	4.61	2.24	3.94	3.14	5.63	3.40	5.09	6.38	2.98	4.05	2.81
NeuS [49]	4.00	4.57	4.49	3.97	4.32	4.63	1.95	4.68	3.83	4.15	2.50	1.52	6.47	1.26	5.57	6.11
PixelNeRF [58]	6.18	5.13	8.07	5.85	4.40	7.11	4.64	5.68	6.76	9.05	6.11	3.95	5.92	6.26	6.89	6.93
IBRNet [50]	2.32	2.29	3.70	2.66	1.83	3.02	2.83	1.77	2.28	2.73	1.96	1.87	2.13	1.58	2.05	2.09
MVSNeRF [4]	2.09	1.96	3.27	2.54	1.93	2.57	2.71	1.82	1.72	2.29	1.75	1.72	1.47	1.29	2.09	2.26
SparseNeuS [26]	1.96	2.17	3.29	2.74	1.67	2.69	2.42	1.58	1.86	1.94	1.35	1.50	1.45	0.98	1.86	1.87
Ours (VolRecon)	1.38	1.20	2.59	**1.56**	1.08	**1.43**	1.92	1.11	1.48	1.42	**1.05**	1.19	1.38	0.74	1.23	1.27

Table 1. Quantitative results of **sparse view** reconstruction on 15 testing scenes of DTU dataset [1]. We report Chamfer distance (lower is better). Methods are separated into four categories (from top to bottom): (1) multi-view stereo (MVS) baselines, (2) per-scene optimization based neural implicit reconstruction methods, (3) generalizable neural rendering methods, and (4) generalizable neural implicit reconstructions. Best scores are in **bold** and second best are underlined.

implicit reconstruction method; note that we report reproduced results using their official repository and the released model checkpoint; (2) generalizable neural rendering methods [4, 50, 58]; (3) per-scene optimization based neural implicit reconstruction methods [34, 49, 56, 57]; (4) MVS methods [39, 54]. We train MVSNet [54] with our training split for 16 epochs. Note that MVS methods are different from neural implicit reconstruction in that they do not implicitly model scene parameters, *e.g.*, SDF, SRDF, and the state-of-the-art MVS methods are unable to render novel views. Similar to [26, 49], we report them as a baseline.

4.2. Evaluation Results

Sparse View Reconstruction on DTU. On DTU [1], we conduct sparse reconstruction with only 3 views. For a fair comparison, we adopt the same image sets and evaluation process as used in SparseNeuS [26]. To calculate SRDF, we define a virtual rendering viewpoint corresponding to each view, which is generated by shifting the original camera coordinate frame for $d = 25mm$ along its x-axis. After rendering the depth maps, we adopt TSDF fusion [5] to fuse the depth maps in a volume with a voxel size of $1.5mm$, and then use Marching Cube [27] to extract the mesh. As shown in Table 1, our method outperforms the state-of-the-art neural implicit reconstruction method SparseNeuS [26] by 30%. As for qualitative visualization shown in Fig. 3, our method generates finer details and sharper boundaries than SparseNeuS. Compared with MVS methods [39, 54], we observe that our method outperforms the traditional MVS method COLMAP [39] by about 10% but is a little worse than MVSNet [54].

Depth map evaluation on DTU. In this experiment, we compare depth estimation with SparseNeuS [26] and MVSNet [54] by evaluating all views in each scan. For each reference view, we use 4 source views with the highest view selection scores according to [54] for depth rendering. For SparseNeuS [26], we set the image resolution to 800 × 600 and render the depth similarly to our method. For MVSNet [54], for a relatively fair comparison, we set the image resolution to 1600 × 1184 since the output depth is downsampled to $1/4$ resolution. As shown in Table 2, our method achieves better performance in all the metrics than MVSNet and SparseNeuS.

Method	$< 1 \uparrow$	$< 2 \uparrow$	$< 4 \uparrow$	Abs. ↓	Rel. ↓
MVSNet [54]	29.95	52.82	72.33	13.62	1.67
SparseNeuS [26]	38.60	56.28	68.63	21.85	2.68
Ours (VolRecon)	**44.22**	**65.62**	**80.19**	**7.87**	**1.00**

Table 2. Depth map evaluation results on DTU [1]. The result of mean absolute error (Abs.) is in millimeters. The results of threshold percentage ($< 1mm$, $< 2mm$, $< 4mm$) and mean absolute relative error (Rel.) are in percentage (%). Best scores are in **bold**.

Full View Reconstruction on DTU. Based on the depth maps of all the views, we further evaluate 3D reconstruction quality. For a fair comparison, we follow the MVS methods to fuse all 49 depth maps of each scan into one point cloud [13, 54]. As shown in Table 3, our method performs better than SparseNeuS and achieves comparable accuracy as MVSNet. As shown in Fig. 4, compared with SparseNeuS, our method shows sharper boundary and fewer holes.

Generalization on ETH3D. To validate the generalization ability of our method, we directly test our model, pretrained using the DTU benchmark [1], on the ETH3D [40] benchmark. We choose 4 scenes for testing: *door*, *statue*, *relief*, and *relief_2*, which have 6, 11, 31, and 31 images, respectively. Compared with DTU, the scale of the scenes increases about

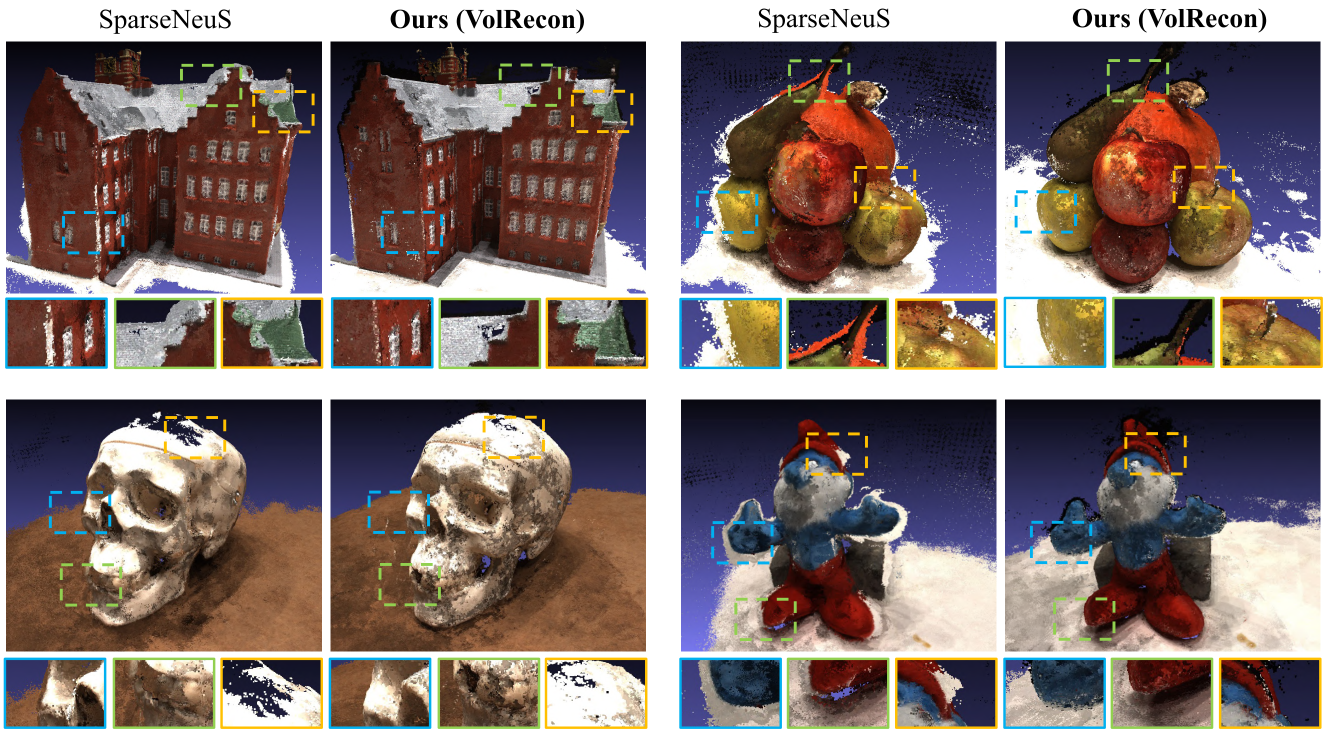

Figure 4. *Point cloud* comparison of **full view** reconstruction on the DTU dataset [1]. Compared with SparseNeuS [26], our method (VolRecon) reconstructs better point clouds, *e.g*. sharper boundary, as the steeple in the top left and pear stalk in the top right, and more complete representation, *e.g*. fewer holes, as skull head top in bottom left, foot in the bottom right. Note that each 3D point is projected with the rendered color and depth from each viewpoint. Best viewed on a screen when zoomed in.

Method	Acc.↓	Comp.↓	Chamfer↓
MVSNet [54]	0.55	**0.59**	**0.57**
SparseNeuS [26]	0.75	0.76	0.76
Ours (VolRecon)	**0.55**	0.66	0.60

Table 3. Point cloud evaluation on DTU [1]. For Accuracy (Acc.), Completeness (Comp.), and Chamfer distance, lower is better. Best scores are in **bold** and second best are underlined.

10×. These large-scale scenes are not suitable to use TSDF fusion [5] due to its limited voxel resolution. We render the depth maps and then fuse them into a point cloud [54] for each scene. As shown in Fig. 5, our method reconstructs large-scale scenes with high quality, which demonstrates that our method has good generalization capability.

4.3. Ablation Study

We conduct ablation studies to analyze the effectiveness of different components in our model. All the experiments are done on the DTU benchmark [1]. We summarize the results of the first three experiments on the sparse view ($N = 3$) reconstruction, depth map evaluation, and full view reconstruction in Table 4.

Ray Transformer. By default, a ray transformer enables each point to attend to the features of other points on the ray. Then we remove the ray transformer and directly use the unattended features to predict SRDF. As shown in Table 4, the performance drops in all the experiments. Without the ray transformer, the SRDF prediction only uses the local information of each point, which is not enough to accurately find the surface location along the ray.

Global Feature Volume. By default, we build a coarse global feature volume to encode global shape priors. We compare with not using global feature volume. The performance becomes worse. We conjecture that the local information from projection features is not enough to accurately locate the surface along a ray. The global feature volume provides global shape priors that are helpful for geometry estimation.

Depth Loss. We remove the depth loss $\mathcal{L}_{\text{depth}}$ during training and observe that the reconstruction quality drops. Though, in sparse view reconstruction, our method still performs comparably to the SparseNeuS [26] and is better than MVSNeRF [4] and IBRNet [50], as shown in Table 1. Many works find that only using pixel color loss $\mathcal{L}_{\text{color}}$ produces bad geometry in novel view synthesis [51], especially in areas with little texture or repetitive patterns. Therefore, many implicit reconstruction methods use careful geometry initialization [49, 56] and geometric priors such as depth maps [59],

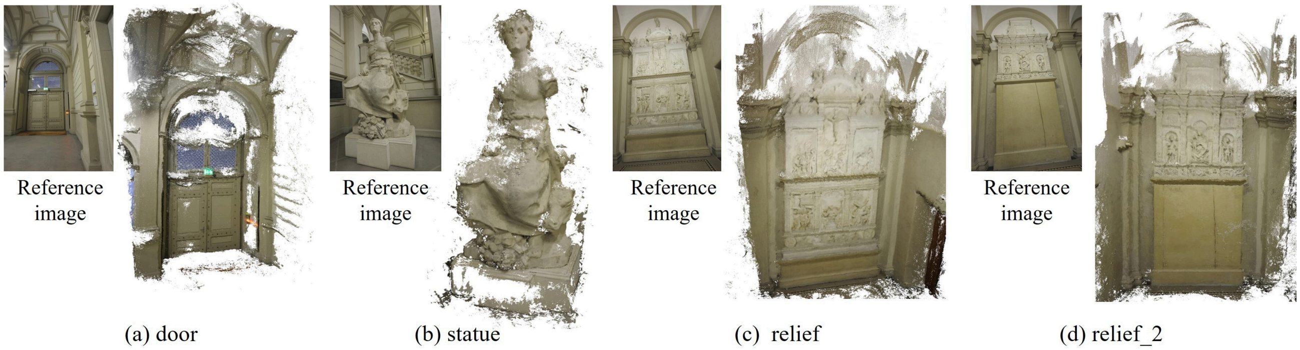

Figure 5. **Generalization ability** of VolRecon. Our model trained on DTU [1] generalizes well the to large-scale strong viewpoint variation benchmark ETH3D [40] without finetuning. Best viewed on a screen when zoomed in.

Method	Sparse View Recon.	Depth Map Eval.					Full View Recon.
	Chamfer↓	$<1\uparrow$	$<2\uparrow$	$<4\uparrow$	Abs.↓	Rel. ↓	Chamfer↓
w/o Ray Trans.	1.79	39.20	60.73	77.38	8.80	1.12	0.66
w/o $\mathbf{F}_v$	1.83	23.29	40.67	59.64	14.90	1.92	0.78
w/o $\mathcal{L}_{\text{depth}}$	2.04	12.84	22.55	34.91	35.00	4.41	1.24
Ours (VolRecon)	**1.38**	**44.22**	**65.62**	**80.19**	**7.87**	**1.00**	**0.60**

Table 4. Ablation study of ray transformer, global feature volume, and depth loss on DTU [1] dataset. Best scores are in **bold**.

normals [48, 59], and sparse point clouds [11] to provide more geometric supervision. Other methods [6, 11] use patch loss, which is common in unsupervised depth estimation methods [14] to provide more robust self-supervision in geometry than pixel color loss.

Number of Views. We vary the number of views N in sparse view reconstruction and summarize the results in Table 5. The reconstruction quality gradually improves with more images. Multi-view information enlarges the observed areas and helps to alleviate problems such as occlusions.

Number of Views	Chamfer ↓
2	1.72
3	1.38
4	1.35
5	**1.33**

Table 5. Ablation study of number of views on DTU benchmark [1] dataset. The Chamfer distance is reported (lower is better). Best score is in **bold**.

5. Limitations & Future Work

There are two limitations of our method. First, the rendering efficiency of our method is limited, which is a common problem in other volume rendering-based methods [4, 30, 50]. It takes about 30s to render an image and depth map with a resolution of 800 × 600. Second, our current model is not suitable for reconstructing very large-scale scenes. The low resolution of our global feature volume results in a decrease in representation performance when the scale of the scene increases. While increasing the resolution of the global feature volume is a potential solution, this will increase memory consumption. Instead, we believe it will be a promising direction to reconstruct progressively in small local volumes like NeuralRecon [42]. To implement this strategy, given a rendering viewpoint, we will select several source views [8, 39, 54] to build a local bounding volume that encloses their view frustums. This will effectively limit the space to a reasonable size and allow us to apply our method within the local region.

6. Conclusion

We introduced VolRecon, a novel generalizable implicit reconstruction method with SRDF. Our method incorporates a view transformer for aggregating multi-view features and a ray transformer for computing SRDF values of all the points along a ray to find the surface location. By utilizing both projection features and volume features, our approach is able to combine local information and global shape prior, and thus produce reconstructions with fine details and of high quality. Our method outperforms the state-of-the-art generalizable neural implicit reconstruction methods on DTU by a large margin. Furthermore, experiments on ETH3D without any fine-tuning demonstrate good generalization ability on large-scale scenes.

Acknowledgements This work was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5-180359.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 120:153–168, 2016. 2, 3, 5, 6, 7, 8

[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. In *ICCV*, pages 5855–5864, 2021. 2

[3] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021. 2

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, pages 14124–14133, 2021. 2, 6, 7, 8

[5] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312. ACM, 1996. 1, 2, 3, 6, 7

[6] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *CVPR*, pages 6260–6269, 2022. 8

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 4

[8] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion. In *CVPR*, pages 15324–15333, 2021. 8

[9] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. 5

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, pages 5501–5510, 2022. 2

[11] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-Neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *CoRR*, abs/2205.15848, 2022. 1, 2, 5, 8

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI*, 32(8):1362–1376, 2009. 2

[13] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, pages 873–881, 2015. 1, 2, 6

[14] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 270–279, 2017. 8

[15] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, pages 2495–2504, 2020. 1, 3

[16] Christian Häne, Torsten Sattler, and Marc Pollefeys. Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. In *International Conference on Intelligent Robots and Systems*, pages 5101–5108. IEEE, 2015. 1

[17] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. In *Programming with TensorFlow*, pages 87–104. Springer, 2021. 5

[18] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Annual ACM Symposium on User Interface Software and Technology*, pages 559–568, 2011. 2

[19] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, volume 119, pages 5156–5165. PMLR, 2020. 4

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[21] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 36(4):1–13, 2017. 3

[22] Ilya Kostrikov, Esther Horbert, and Bastian Leibe. Probabilistic labeling cost for high-accuracy multi-view reconstruction. In *CVPR*, pages 1534–1541, 2014. 2

[23] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, 2000. 2

[24] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE TPAMI*, 27(3):418–433, 2005. 2

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[26] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. SparseNeuS: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, pages 210–227. Springer, 2022. 1, 2, 3, 5, 6, 7

[27] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 21(4):163–169, 1987. 6

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 1, 2

[29] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *ECCV*, pages 268–283. Springer, 2014. 1

[30] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 5, 8

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

[32] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, pages 414–431. Springer, 2020. 2, 3

[33] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM TOG*, 32(6):1–11, 2013. 2

[34] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *CVPR*, pages 5589–5599, 2021. 6

[35] Martin Ralf Oswald, Jan Stühmer, and Daniel Cremers. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *ECCV*, pages 32–46. Springer, 2014. 1

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 1, 2

[37] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, pages 523–540. Springer, 2020. 2

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 3

[39] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518. Springer, 2016. 1, 2, 3, 4, 6, 8

[40] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, pages 3260–3269, 2017. 2, 3, 5, 6, 8

[41] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):151–173, 1999. 2

[42] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, pages 15598–15607, 2021. 2, 3, 8

[43] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *CVPR*, pages 8248–8258, 2022. 1

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30:5998–6008, 2017. 4

[45] Dan Wang, Xinrui Cui, Septimiu Salcudean, and Z Jane Wang. Generalizable neural radiance fields for novel view synthesis with transformer. *CoRR*, abs/2206.05375, 2022. 4

[46] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. IterMVS: iterative probability estimation for efficient multi-view stereo. In *CVPR*, pages 8606–8615, 2022. 3

[47] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. PatchmatchNet: Learned multi-view patchmatch stereo. In *CVPR*, pages 14194–14203, 2021. 1, 3, 4

[48] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. NeuRIS: Neural reconstruction of indoor scenes using normal priors. In *ECCV*, pages 139–155. Springer, 2022. 1, 2, 8

[49] 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. *NeurIPS*, 34:27171–27183, 2021. 1, 2, 3, 4, 5, 6, 7

[50] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. In *CVPR*, pages 4690–4699, 2021. 2, 4, 6, 7, 8

[51] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, pages 5610–5619, 2021. 1, 7

[52] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011. 1

[53] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *CVPR*, pages 5483–5492, 2019. 2

[54] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783. Springer, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[55] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, pages 5525–5534, 2019. 1, 3

[56] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 34:4805–4815, 2021. 1, 2, 3, 4, 5, 6, 7

[57] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 33:2492–2502, 2020. 2, 3, 6

[58] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 2, 6

[59] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 2022. 1, 2, 7, 8

[60] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Critical regularizations for neural surface reconstruction in the wild. In *CVPR*, pages 6270–6279, 2022. 1, 2

[61] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *CoRR*, abs/2008.07928, 2020. 3

[62] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *CoRR*, abs/2010.07492, 2020. 1

[63] Pierre Zins, Yuanlu Xu, Edmond Boyer, Stefanie Wuhrer, and Tony Tung. Multi-view reconstruction using signed ray distance functions (srdf). *CoRR*, abs/2209.00082, 2022. 2, 3