Learning Signed Distance Field for Multi-view Surface Reconstruction

Jingyang Zhang Yao Yao Long Quan
The Hong Kong University of Science and Technology
{jzhangbs,yyaoag,quan}@cse.ust.hk

Abstract

Recent works on implicit neural representations have shown promising results for multi-view surface reconstruction. However, most approaches are limited to relatively simple geometries and usually require clean object masks for reconstructing complex and concave objects. In this work, we introduce a novel neural surface reconstruction framework that leverages the knowledge of stereo matching and feature consistency to optimize the implicit surface representation. More specifically, we apply a signed distance field (SDF) and a surface light field to represent the scene geometry and appearance respectively. The SDF is directly supervised by geometry from stereo matching, and is refined by optimizing the multi-view feature consistency and the fidelity of rendered images. Our method is able to improve the robustness of geometry estimation and support reconstruction of complex scene topologies. Extensive experiments have been conducted on DTU, EPFL and Tanks and Temples datasets. Compared to previous state-of-the-art methods, our method achieves better mesh reconstruction in wide open scenes without masks as input.

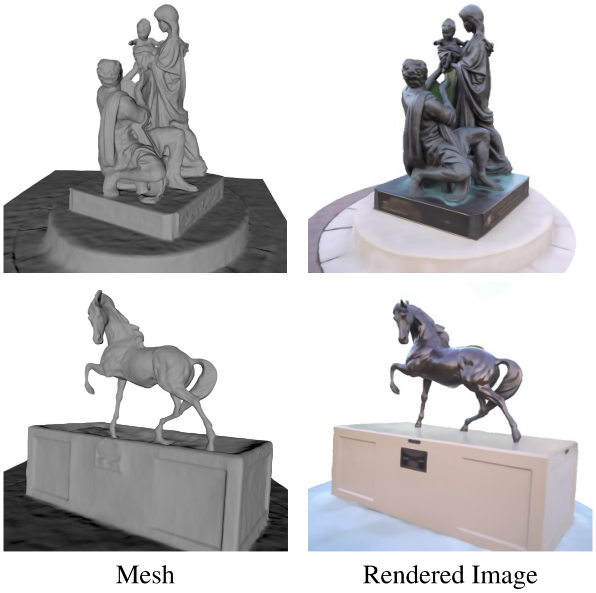


Figure 1. Our reconstructions on *Family* and *Horse* of the *Tanks and Temples* dataset. The proposed MVSDF is able to reconstruct correct mesh topologies with fine details for highly texture-less and reflective surfaces.

1. Introduction

Surface reconstruction from calibrated multi-view images is one of the key problems in 3D computer vision. Traditionally, surface reconstruction can be divided into two substeps: 1) depth maps and point clouds are reconstructed from images via multi-view stereo (MVS) algorithms [2, 8, 41, 37, 44]; 2) a surface, usually represented as a triangular mesh, is extracted from dense points by maximizing the conformity to points [20, 17, 26]. Optionally, a surface refinement step can be applied to recover geometry details through multi-view photo-consistency [34, 42, 5, 4, 7]. While this pipeline has been proven to be effective and robust in various scenarios, the reconstructed geometry may be suboptimal due to the accumulated loss in representation conversions from images to points, then to mesh. For example, errors introduced in point cloud reconstruction would pass to the surface reconstruction, causing wrong mesh topology and difficult to be recovered. Although recent learning-based MVS [15, 13, 45, 32, 46] and mesh extraction [31, 28, 39, 33] methods have been proposed to boost the reconstruction quality of each substep independently, it is still desirable to reconstruct the optimal surface from images in an end-to-end manner.

Alternatively, recent works on neural representations show that mesh surface can be directly constructed from images through implicit representations and differentiable rendering [30, 24, 36, 25, 49, 39, 29]. Surface geometry and color information of the scene are usually represented as implicit functions, which are directly modeled by multi-layer perceptrons (MLPs) in the network and optimized through differentiable rendering. The triangle mesh can be extracted from the implicit field via the Marching Cube algorithm [26, 28]. Compared with classical meshing pipelines, these methods are able to reconstruct the scene geometry in an end-to-end manner and generate synthesized images at the same time. However, as all scene parameters are jointly optimized at the same time, geometry is only a by-product of the entire differential rendering pipeline, and ambiguities exist in geometry and appearance[52]. To mitigate the problem, implicit differentiable renderer (IDR) [49] applies manually labeled object masks as input, but it is not feasible for a large number of images and is sometimes not well defined for real-world image inputs.

In this paper, we present MVSDF, a novel neural surface reconstruction framework that combines implicit neural surface estimation with recent advanced MVS networks. On the one hand, we follow the implicit differentiable renderer[49] to represent the surface as zero level set of a signed distance field (SDF) and the appearance as a surface light field, which are jointly optimized through render loss. On the other hand, we introduce deep image features and depth maps from learning-based MVS [46, 51, 50] to assist the implicit SDF estimation. The SDF is supervised by inferred depth values from the MVS network, and is further refined by maximizing the multi-view feature consistency at the surface points of the SDF. We find that the surface topology can be greatly improved with the guidance from MVS depth maps, and our method can be applied to complex geometries even without input object masks. Also, compared to render loss in IDR, the multi-view feature consistency imposes a photometric constraint at an early stage of the differentiable rendering pipeline, which significantly improves the geometry accuracy and helps to preserve high-fidelity details in final reconstructions.

Our method has been evaluated on *DTU* [14], *EPFL* [40] and *Tanks and Temples* [18] datasets. We compare our method with classical meshing pipelines and recent differentiable rendering based networks on both mesh reconstruction and view synthesis quality. Both quantitative and qualitative results demonstrate that our method is able to recover complex geometries even without object masks as input.

2. Related Works

Multi-view Stereo Multi-view stereo [38, 10, 6] is a well-developed approach to recover the dense representation of the scene from overlapping images. The fundamental principle of MVS is that a surface point should be visually consistent in all visible views. Traditionally, MVS evaluates the matching cost of image patches for all depth hypotheses and finds the one that best describes the input images. Depth hypotheses can be uniformly sampled from predefined camera frustum [3] or propagated from neighboring pixels and adjacent views [21, 8, 1, 9, 37, 44, 19]. Usually depth map outputs are fused into a unified point cloud, and further converted into a mesh surface [20, 17, 26]. However, such conversions may be lossy. For example, depth maps may be over-filtered so that holes would appear in the final reconstruction. Moreover, surface details may be over-smoothed during the conversion from point cloud to mesh.

Recently, deep learning techniques have been applied to multi-view stereo [15, 13, 46, 50, 51]. In the network, hand-crafted image features are converted into deep features and engineered cost regularizations are replaced by learned ones [32, 46, 47, 12, 43]. Although the overall depth quality is improved, it is still difficult to match pixels in texture-less or non-Lambertian regions. In this work, we aim to improve the reconstruction of texture-less regions by the interpolation capability of implicit functions, and non-Lambertian regions by explicit view-dependent appearance modeling.

Implicit Neural Surface Implicit neural representations have also gained popularity in 3D scene reconstructions. Occupancy fields [28, 33] are proposed to model the object surface using the per-point occupancy information, while DeepSDF [31] applies a signed distance field to describe the 3D geometry of the scene. These methods usually take 3D point cloud as input and optimize corresponding implicit fields using ground truth labels, which can be viewed as learned mesh reconstruction from dense point cloud. The marching cube algorithm [26] is usually applied to extract the mesh surface from neural implicit functions. Compared with widely used mesh surfaces and discretized volumes, implicit neural functions are able to model continuous surfaces with fix-sized multi-layer perceptrons, making it more natural and efficient to represent complex geometries with arbitrary topologies.

Surface Reconstruction by Differentiable Rendering Apart from MVS, rendering and view synthesis based methods [16] provide an alternative to estimate scene geometries by minimizing the difference between rendered and input images. The geometry is either represented using soft representations such as density/transparency fields [39, 29, 23, 27, 52], or explicit representations such as occupancy fields [30] and signed distance fields [24, 25, 49].

Our method is most related to IDR [49] which uses SDF and surface light field as the scene representation. These two implicit networks are jointly trained by the render loss, and image masks are applied for constrained SDF optimization. However, the reconstruction quality of IDR highly depends on the accuracy of the input masks, and inaccurate masks may result in either missing or extra mesh surfaces. As auto object segmentation methods [53, 35] cannot always be perfect, IDR applies manually labeled masks to ensure the reconstruction quality. In this work, we introduce multi-view stereo and feature consistency as our geometry constraints to improve the surface quality and relax the requirement of image masks.

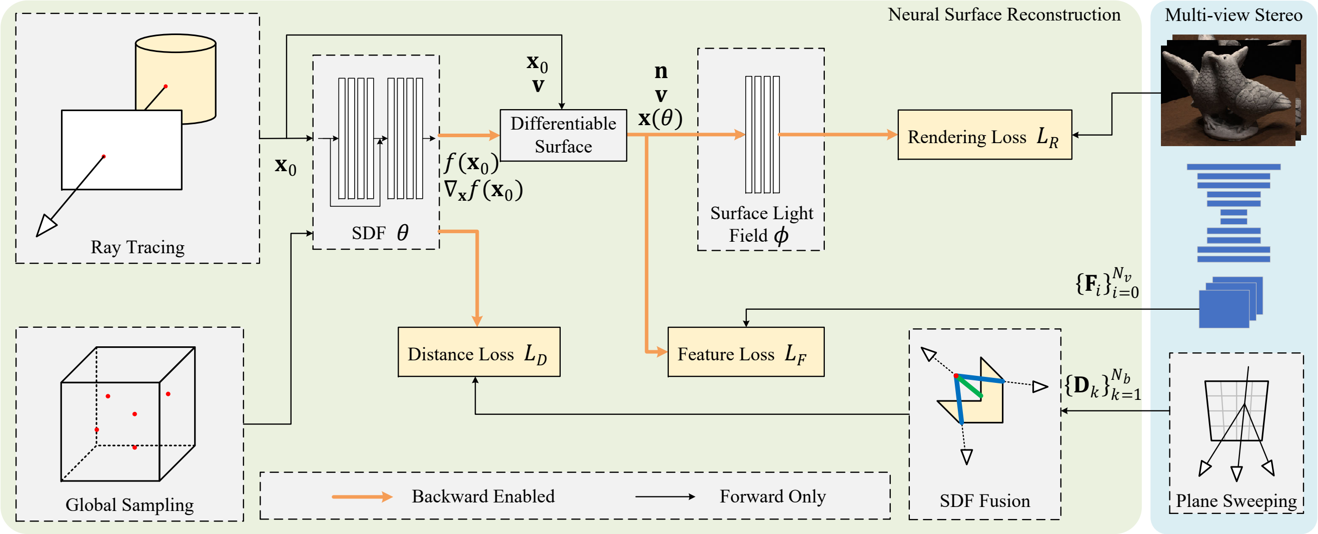


Figure 2. **Illustration of the proposed framework.** In the network, points are sampled from the space and their signed distance values are directly supervised by MVS depth outputs (Sec. 3.2). Then, surface points are calculated by ray tracing and refined by deep feature consistency (Sec. 3.3). Finally, the SDF geometry and surface light field are jointly optimized by the render loss (Sec. 3.3).

3. Method

3.1. Geometry and Appearance Representations

In our network, the surface S_θ is explicitly modeled as the zero level set of a SDF, which is represented by a MLP f in the network. We define θ the learnable parameters of f. The MLP will take a query location $\mathbf{x}$ as input and output a distance from the query to the closest surface point.

$$S_\theta = \{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}; \theta) = 0\} \tag{1}$$

Inspired by IDR [49], our scene appearance is represented by a surface light field using another MLP g with learnable parameters ϕ. The surface light field takes the query surface point $\mathbf{x}$, its normal vector $\mathbf{n}$ and the unit vector of the viewing ray $\mathbf{v}$ as input and outputs the RGB color $\mathbf{c}$ of the query.

$$\mathbf{c} = g(\mathbf{x}, \mathbf{n}, \mathbf{v}; \phi) \tag{2}$$

During the rendering, the intersection point of the viewing ray and the surface is obtained by sphere tracing, and the point normal can be calculated as the analytical gradient of the implicit surface $\mathbf{n} = \nabla_{\mathbf{x}} f(\mathbf{x}; \theta)$.

Differentiable Surface Intersection The sphere tracing is not a differentiable operation in the network. Following previous works [30, 49], we construct the first order approximation of the function from network parameters to the intersection location. For the current network parameters θ_0, viewing ray $\mathbf{v}$ and the intersection point $\mathbf{x}_0$ on this ray, we take implicit differentiation on the equation $f(\mathbf{x}; \theta) \equiv 0$, and the surface intersection can be expressed as a function of θ:

$$\mathbf{x}(\theta) = \mathbf{x}_0 - \frac{f(\mathbf{x}_0; \theta) - f(\mathbf{x}_0; \theta_0)}{\nabla_{\mathbf{x}} f(\mathbf{x}_0; \theta_0) \cdot \mathbf{v}} \mathbf{v} \tag{3}$$

where $f(\mathbf{x}_0; \theta_0)$ and $\nabla_{\mathbf{x}} f(\mathbf{x}_0; \theta_0)$ are constants.

3.2. Geometry Supervision

Multi-view stereo algorithms are able to provide high-quality depth maps as a dense representation of the scene. In this section, we describe how to use MVS depth maps to supervise our SDF optimization.

Multi-view Depth Map Estimation In our network, the MVS module aims to generate deep image features and qualified depth maps for all input images. We apply the open-sourced Vis-MVSNet [51] as our depth generation module. For a reference image $\mathbf{I}_0$ and its N_v neighboring source images $\{\mathbf{I}_i\}_{i=1}^{N_v}$, a standard UNet is first applied to extract deep image feature maps $\{\mathbf{F}_i\}_{i=0}^{N_v}$. Then, all feature maps will be warped into the camera frustum of $\mathbf{I}_0$ and construct a 3D cost volume $\mathbf{C}$. We further regularize the cost volume by 3D CNNs and obtain the probability distribution of the depth samples by *softmax*. Finally, the depth $\mathbf{D}_0$ is regressed from the probability volume by taking the depth *expectation*. In addition, for a pixel $\mathbf{p}$ in the depth map, we evaluate its probability sum $\mathbf{P}(\mathbf{p})$ around the predicted depth value as an indicator of the depth confidence [46]. Pixels with low confidence will be filtered out to generate a cleaned depth map.

Direct SDF Supervision Previous work [30] proposes to train the implicit network by minimizing the difference between the traced depth map and the ground truth one. However, such strategy can only affect network outputs near the current surface estimation. To ensure the SDF to be correctly recovered in the whole space, we instead randomly sample points from the whole space and compute the distance from the sample point to the MVS depth map.

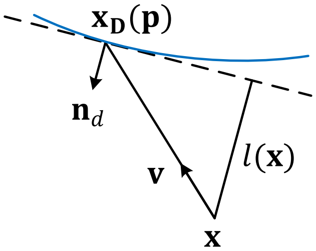


Figure 3. Approximated signed distance.

Specifically, given a sample point $\mathbf{x}$ and a depth map $\mathbf{D}$, we first project $\mathbf{x}$ to the depth map at location $\mathbf{p}$. Then we backproject the MVS depth $\mathbf{D}(\mathbf{p})$ at the same location to the space as $\mathbf{x_D}(\mathbf{p})$. As is shown in Fig. 3, the signed distance from $\mathbf{x}$ to MVS surface is approximated as

$$l(\mathbf{x}) = \text{sgn}[(\mathbf{x_D}(\mathbf{p}) - \mathbf{x}) \cdot \mathbf{v}](-\mathbf{n}_d \cdot \mathbf{v})\|\mathbf{x_D}(\mathbf{p}) - \mathbf{x}\| \quad (4)$$

where $\mathbf{n}_d$ is the normal calculated from depth. Also, if the probability sum $\mathbf{P}(\mathbf{p})$ is smaller than a threshold T_{prob}, we consider this pixel as in the background and will exclude the corresponding point from the distance computation. This approximated signed distance can be used to supervise the SDF training, and we define the distance loss L_D as:

$$L_D(\theta) = \frac{1}{|\mathbb{S}|}\sum_{\mathbf{x}\in\mathbb{S}} |f(\mathbf{x};\theta) - l(\mathbf{x})| \quad (5)$$

where $\mathbb{S}$ is the set of valid sample points.

Signed Distance Fusion in a Mini-batch One problem of Eq. 5 is that the approximated signed distance $l(\mathbf{x})$ calculated from a single depth map is usually not reliable. First, a sample point in the free space may be occluded in a given view. Second, the approximated $l(\mathbf{x})$ may be inaccurate when non-planar surfaces occur. To improve the accuracy of $l(\mathbf{x})$, we group N_b views in a mini-batch during training, and $l(\mathbf{x})$ will be refined by fusing multiple observations from N_b depth maps within the mini-batch.

For a query point $\mathbf{x}$, we first calculate its approximated signed distances $\{l_k(\mathbf{x})\}_{k=1}^{N_b}$ in each depth map. According to the sign of $l_k(\mathbf{x})$, we define that a point is outside the surface if at least T_{out} distances from $\{l_k(\mathbf{x})\}_{k=1}^{N_b}$ are positive. After the query point is determined to be inside or outside, we collect the per-view distance with the same sign and take the minimum depth distance as the absolute value of the fused distance $l(\mathbf{x})$. We find that such simple fusion strategy can effectively filter out erroneous observations from single depth map, and the fused $l(\mathbf{x})$ is accurate enough to be used to guide the SDF optimization.

3.3. Local Geometry Refinement

The geometry supervision in Sec. 3.2 can correctly recover the surface topology. However, as depth maps from MVS networks are usually noisy, it is rather difficult to restore surface details in the final mesh reconstruction. To this end, we propose to optimize the feature consistency and the rendered image consistency during the network training.

Feature Consistency In traditional MVS or mesh reconstruction pipelines, dense point clouds or mesh surfaces are usually refined via multi-view photo-consistency optimization [42, 5, 4, 22, 47]. The photo-consistency of a surface point is defined as the matching cost (e.g., ZNCC) among multiple views. In our work, note that deep image features have already been extracted in Vis-MVSNet. Inspired by [50] , we instead minimize the multi-view deep feature consistency.

Suppose a surface point $\mathbf{x}$ is obtained via ray tracing in view 0, we denote its projections in view 0 and its neighboring views as $\{\mathbf{p}_i\}_{i=0}^{N_v}$. As these projections refer to the same 3D point in space, their deep image features should be consistent. The feature loss is defined as:

$$L_F(\theta) = \frac{1}{N_v N_c}\sum_{i=1}^{N_v} |\mathbf{F}_0(\mathbf{p}_0) - \mathbf{F}_i(\mathbf{p}_i)| \quad (6)$$

where N_c is the number of feature channels, $\mathbf{p}_i = \mathbf{K}_i(\mathbf{R}_i\mathbf{x} + \mathbf{t}_i)$ is the projection of $\mathbf{x}$ in view i and $[\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i]$ the corresponding camera parameters. Deep image feature at pixel $\mathbf{p}_i$, denoted as $\mathbf{F}_i(\mathbf{p_i})$, is obtained by bilinear interpolation. To optimize the SDF via the feature consistency loss, we derived the gradient of $L_F(\theta)$ with respect to the network parameter θ as:

$$\frac{\partial L_F(\theta)}{\partial\theta} = \frac{\partial L_F}{\partial\mathbf{x}} \cdot \frac{\partial\mathbf{x}}{\partial\theta} = (\sum_{i=0}^{N_v} \frac{\partial L_F}{\partial\mathbf{F}_i}\frac{\partial\mathbf{F}_i}{\partial\mathbf{p}_i}\frac{\partial\mathbf{p}_i}{\partial\mathbf{x}}) \cdot \frac{\partial\mathbf{x}}{\partial\theta} \quad (7)$$

where $\partial\mathbf{F}_i/\partial\mathbf{p}_i$ is the gradient of the feature map. The last term $\partial\mathbf{x}/\partial\theta$ can be calculated from the derivative of Eq. 3.

Compared with the render loss to be discussed in the next paragraph, the proposed feature loss introduces a photo-consistency constraint in an early stage of the whole differentiable rendering pipeline, which reduces the geometry and appearance ambiguity during the joint optimization. We show in ablation study that $L_F(\theta)$ can effectively increase the mesh reconstruction quality (see Tab. 3).

Rendered Image Consistency The rendered image consistency is widely used in recent differentiable rendering pipelines [30, 29, 49]. For pixel $\mathbf{p}$ in the image, we can trace its surface intersection $\mathbf{x}$ in the space. The rendered color $\mathbf{c}(\mathbf{p})$ of pixel $\mathbf{p}$ can be directly fetched from the surface light field by feeding $\mathbf{x}(\theta)$, $\nabla_{\mathbf{x}} f(\mathbf{x};\theta)$ and $\mathbf{v}$ into function g. The render loss is then calculated as the L1 distance from the rendered color to the input image color:

$$L_R(\theta,\phi) = \frac{1}{|\mathbb{S}_I|}\sum_{\mathbf{p}\in\mathbb{S}_I} |\mathbf{c}(\mathbf{p}) - \mathbf{I}(\mathbf{p})| \quad (8)$$

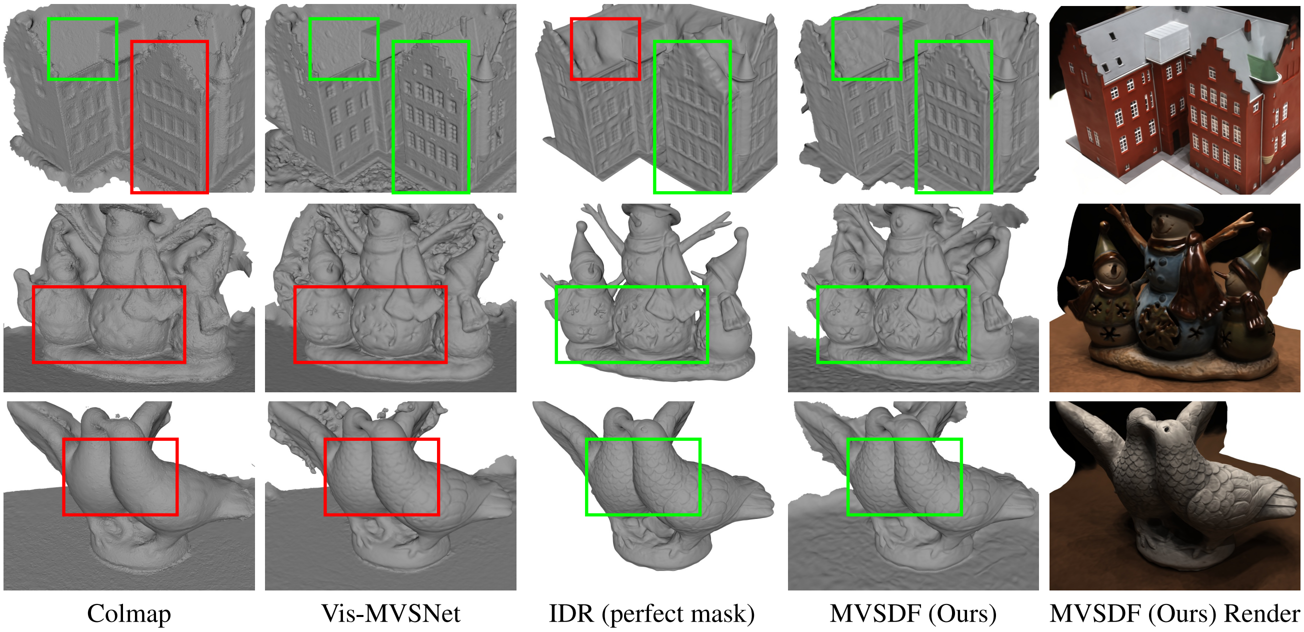


Figure 4. **Qualitative results on DTU dataset.** Our method produces both high quality meshes and rendered images without requiring masks as input.

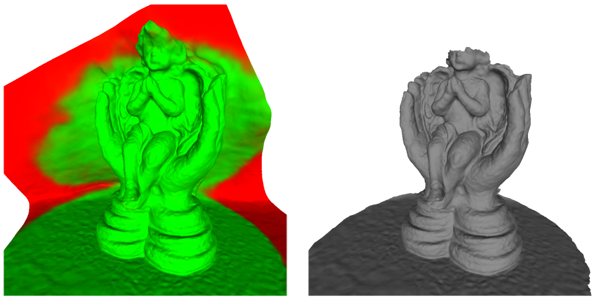


Figure 5. **Illustration of surface indicator and mesh trimming.** Extra surfaces are trimmed with a graph-cut based algorithm (Sec. 4.1) according to learned surface indicators (Sec. 3.4, green indicates accurate surface).

where $\mathbb{S}_I$ denote the set of image pixels whose view ray can intersect the surface in the space.

The render loss can jointly optimize the geometry θ and the appearance ϕ. Compared with L_F, L_R is more sensitive to local color changes, and plays an important role in recovering high-fidelity surface details.

3.4. Valid Surface Indicator

If input images cannot fully cover the object of interest, the surface of unseen areas will be not well defined, and will tend to produce extrapolated surfaces in the background areas. To distinguish such invalid surface, we use another indicator function to mark whether a space point can be traced from some input views. Specifically, function $h(\mathbf{x};\gamma)$ represents an indicator that $\mathbf{x}$ is in the valid surface. During each training iteration, indicators of successfully traced locations $\{\mathbf{x}^+\}$ are set to 1. To prevent $h(\mathbf{x};\gamma)$ reporting 1 everywhere, we also randomly sample points $\{\mathbf{x}^-\}$ in the space and set the background indicator to 0. We then apply the binary cross entropy as our indicator loss.

$$L_P(\gamma) = \sum_{\mathbf{x}^+} -\log h(\mathbf{x}^+;\gamma) + \sum_{\mathbf{x}^-} -\log(1 - h(\mathbf{x}^-;\gamma)) \tag{9}$$

Note that our MVS depth map is filtered using the corresponding probability map and we will not apply ray tracing on those filtered pixels. As a result, filtered regions in MVS depth maps will tend to be assigned with the background indicator of 0. In other words, we could identify the invalid surface area according to filtered MVS depth maps.

3.5. Loss

Apart from the aforementioned losses, we further regularize our SDF by a Eikonal loss [11] that restricts the expectation of the gradient magnitude to be 1.

$$L_E(\theta) = \mathbb{E}_{x\in\mathbb{R}^3}(\|\nabla_{\mathbf{x}} f(\mathbf{x};\theta)\| - 1)^2 \tag{10}$$

The final loss is expressed as a weighted sum of all the aforementioned losses.

$$\begin{aligned} L =& w_R L_R(\theta,\phi) + w_F L_F(\theta) + w_D L_D(\theta) \\ &+ w_E L_E(\theta) + w_P L_P(\gamma) \end{aligned} \tag{11}$$

where the weight w will be changed over the network training. Our training process is divided into three stages: 1) in the first stage w_D is set to be dominant so as to determine the initial topology; 2) in the second stage, the significance of w_F is increased to recover finer structures in the surface; 3) in the final stage, both w_D and w_F is decreased so that the render loss can restore fine-scale details of the surface.

	Chamfer (mm)					PSNR				
	Colmap [37]	Vis-MVSNet [51]	DVR [30]	IDR [49]	MVSDF (Ours)	Colmap [37]	Vis-MVSNet [51]	DVR [30]	IDR [49]	MVSDF (Ours)
24	0.99	0.98	4.10	1.63	**0.83**	18.44	18.35	16.23	23.29	**25.02**
37	2.35	2.10	4.54	1.87	**1.76**	14.37	14.71	13.93	**21.36**	19.47
40	0.73	0.93	4.24	**0.63**	0.88	19.24	18.60	18.15	24.39	**25.96**
55	0.53	0.46	2.61	0.48	**0.44**	18.27	19.07	17.14	22.96	**24.14**
63	1.56	1.89	4.34	**1.04**	1.11	19.92	17.55	17.84	**23.22**	22.16
65	1.01	**0.67**	2.81	0.79	0.90	13.80	17.17	17.23	23.94	**26.89**
69	0.89	**0.67**	2.53	0.77	0.75	21.23	21.81	16.33	20.34	**26.38**
83	1.14	**1.08**	2.93	1.33	1.26	22.67	23.11	18.10	21.87	**25.79**
97	0.91	**0.67**	3.03	1.16	1.02	18.19	18.68	16.61	22.95	**26.22**
105	1.46	0.95	3.24	**0.76**	1.35	20.43	21.68	18.39	22.71	**27.29**
106	0.79	**0.66**	2.51	0.67	0.87	20.73	21.03	17.39	22.81	**27.78**
110	1.08	0.85	4.80	0.90	**0.84**	17.93	18.41	14.43	21.26	**23.82**
114	0.44	**0.30**	3.09	0.42	0.34	19.08	19.42	17.08	25.35	**27.79**
118	0.68	**0.45**	1.63	0.51	0.47	22.05	23.85	19.08	23.54	**28.60**
122	0.73	0.51	1.58	0.53	**0.46**	22.04	24.29	21.03	27.98	**31.49**
Mean	1.02	**0.88**	3.20	0.90	**0.88**	19.23	19.85	17.26	23.20	**25.92**

Table 1. **Quantitative results on DTU dataset.** Our method achieves the best mean Chamfer distance as Vis-MVSNet [51] and the highest PSNR score among all methods.

4. Experiments

4.1. Implementation

Network Architecture The SDF is implemented by an 8-layer MLP with 512 hidden units and a skip connection in the middle. Positional encoding [29] is applied to input position to capture the high frequency information. This MLP simultaneously outputs the distance, the surface indicator probability and a descriptor of the location as an input of the surface light field function. Similarly, the surface light field is implemented by a 4-layer MLP with 512 hidden units. The function takes the point location, its descriptor, normal and viewing ray as input. Only the viewing direction is enhanced by positional encoding as the point descriptor has already included the rich positional information. In MVS module, we use one reference and two source images ($N_v = 2$) as input to Vis-MVSNet, and will output deep feature ($N_c = 32$) maps of all images and the reference depth map. The MVS module is pretrained on *BlendedMVS* [48] dataset and the parameters are fixed during training.

Training For each input scene, the network is end-to-end trained for 10800 steps with a batch size of $N_b = 8$. In each training step, 4096 pixels are uniformly sampled from each of the 8 images in the mini-batch for tracing surface intersections. Additionally, the same number of 3D points are sampled from the space to calculate the distance loss and the Eikonal loss. To recover correct topologies of thin structures, we need to sample more points near the object surface. This can be achieved by jittering surface points obtained from MVS depth maps. In distance fusion, the minimal number for outside decision $T_{out} = 2$. The initial learning rate is 10^{-3} and is scaled down by 10 when reaching $4/6$ and $5/6$ of the whole training process. As mentioned in Sec. 3.5, weights of the losses are set according to the training stages. Please refer to the supplementary material for detailed setting.

The memory consumption is related to batch size, number of samples per images and number of source images. For *DTU* dataset, our training setting takes ∼20 GB VRAM. And the whole training process takes 5.5 hours for one scan with 49 images on an NVidia RTX Titan.

Mesh Extraction and Trimming After network training, a mesh can be extracted from the SDF in a predefined bounding box by the Marching Cube algorithm [26] with volume size of 512^3. For scenes whose camera trajectory does not surround the object (e.g., DTU dataset), extrapolated surfaces would appear in background areas, and we propose to filter these areas according to the surface indicator described in Sec. 3.4. We first evaluate the valid surface indicator for each mesh vertex, and assign each triangle an indicator score as the average score of its three vertices. Next, instead of deleting all triangles with low indicator scores, we propose a graph-cut based method to smoothly filter out those outlier surfaces.

We define a graph $G = (V, E)$ over the mesh from Marching Cube, where each triangle represents a graph node $v \in V$ and each edge between two adjacent triangles represents a graph edge $e \in E$. A source and a sink node $s, t \in V$ are also defined. Triangles are linked to s with edge weights 1 if their indicator scores are greater than $T_{trim} = 0.94$, or to t otherwise. Adjacent triangles are also linked with weights 10 to encourage smoothness. After a min-cut of the constructed graph is obtained, the triangles linked with t are removed. The proposed trimming algorithm can effectively filter out extrapolated background surfaces, as is illustrated in Fig. 5.

4.2. Benchmark on DTU Dataset

We first evaluate our method on the *DTU* MVS dataset. *DTU* dataset contains 128 scans captured in laboratory. For each scan, there are 49 calibrated cameras located on front side of the upper sphere surrounding the captured object. In this paper, we evaluate both the surface mesh and the

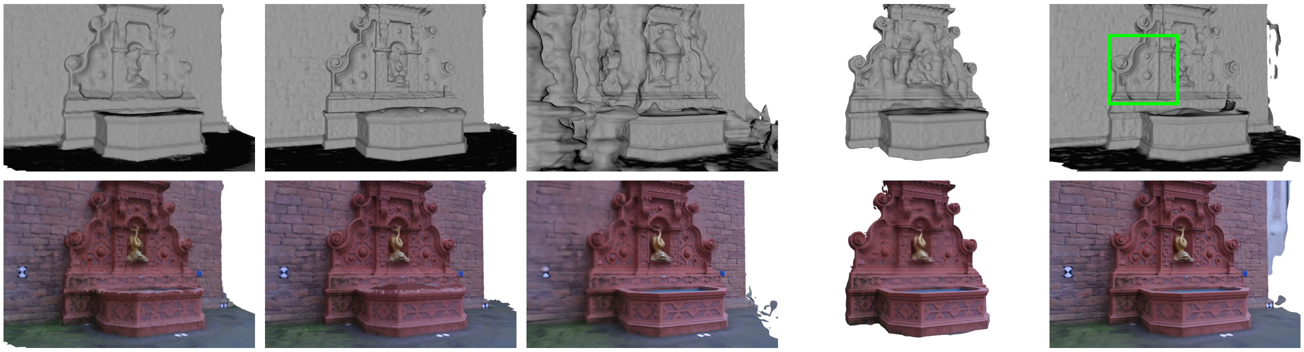


Figure 6. **Qualitative results on EPFL dataset.** Our method is able to generate both high quality mesh and rendering results.

	Chamfer ($\times10^{-2}$)				PSNR			
	Colmap	Vis-MVSNet	IDR	MVSDF (Ours)	Colmap	Vis-MVSNet	IDR	MVSDF (Ours)
Fountain-P11	6.35	**6.12**	18.42 (7.88)	6.84	20.17	24.33	24.58 (23.43)	**25.27**
Herzjesu-P8	8.99	7.47	32.19	**6.38**	16.13	23.45	24.75	**28.75**
Mean	7.67	6.80	25.30	**6.61**	18.15	23.89	24.67	**27.01**

Table 2. **Quantitative results on EPFL dataset.** Our method achieves the lowest mean Chamfer distance and the best PSNR score among all methods. Values in the parenthesis represent the results of IDR when the wall is excluded.

rendered image using the same set of scans as in [49].

The reconstructed mesh model is evaluated by the Chamfer distance to the ground truth point cloud, and the rendered image is evaluated using the PSNR score to the input image. We compare our method to 1) Colmap [37], which represents state-of-the-art traditional MVS algorithms; 2) Vis-MVSNet [51], which represents state-of-the-art learning-based algorithms and 3) DVR [30] and IDR [49], which represent recent rendering based surface reconstruction methods. Depth maps from Colmap and Vis-MVSNet are fused into point clouds and converted into surface meshes by the screened Poisson surface reconstruction (sPSR) [17] with trim parameter 5. As Colmap and Vis-MVSNet do not estimate the surface texture, we follow [49] to assign color from input images when back projecting depth maps to point clouds.

Quantitative results are shown in Tab. 1. The proposed method achieves the best mean Chamfer distance (0.88) and PSNR (25.92) among all methods. Qualitative results are shown in Fig. 4. Both our method and IDR are able to recover high-quality details in mesh surfaces. Compared to IDR, we have less distortions in reconstructed mesh surfaces. It is also noteworthy that our results are reconstructed without any manual masks.

4.3. Benchmark on EPFL Dataset

Our method is also evaluated on the *EPFL* dataset. *EPFL* dataset contains 2 outdoor scenes, *Fountain-P11* and *Herzjesu-P8*, with ground truth meshes. We compare our method with Colmap, Vis-MVSNet and IDR. As *Fountain-P11* and *Herzjesu-P8* mainly consist of planar surfaces, they can not be well handled by the silhouette based methods. To fairly compare our method with IDR, for *Fountain-P11*, we also test the case where the wall is excluded from the masks.

Qualitative results are shown in Fig. 6. Similar to *DTU*, our method is able to produce both high-quality mesh and renderings. The mesh from IDR contains inflated surfaces, which is more serious when the wall is included in input masks. The reason is that there is a large gap between the mask visual hull and the real surface so the solution may stuck at local minimum in this unrestricted area. In contrast, the topology is correctly recovered in our reconstruction. For quantitative results shown in Tab. 2, our method achieves the best mean Chamfer distance (6.61) and PSNR (27.01) among all methods.

4.4. Additional Qualitative Results

We additionally provide qualitative results on *Family* and *Horse* in the *Tanks and Temples* [18] (Fig. 1,7) and two scenes in the *BlendedMVS* [48] dataset (Fig. 8). For *Horse*, as the foundation part is highly texture-less and reflective, the estimated point cloud from Vis-MVSNet is incomplete. Although surfaces can be interpolated in some extent during the mesh reconstruction, the output mesh is bumpy and the rendered image is rather noisy. In contrast, the proposed method is able to produce complete and accurate mesh surface together with realistic view-dependent rendering.

4.5. Ablation Study

In this section, we discuss how different losses in the network would affect the final geometry reconstruction. The following three settings are tested using the *DTU* dataset: 1) *no feature*: the feature loss is removed from the network by setting $w_F = 0$; 2) *no render*: the render loss is removed

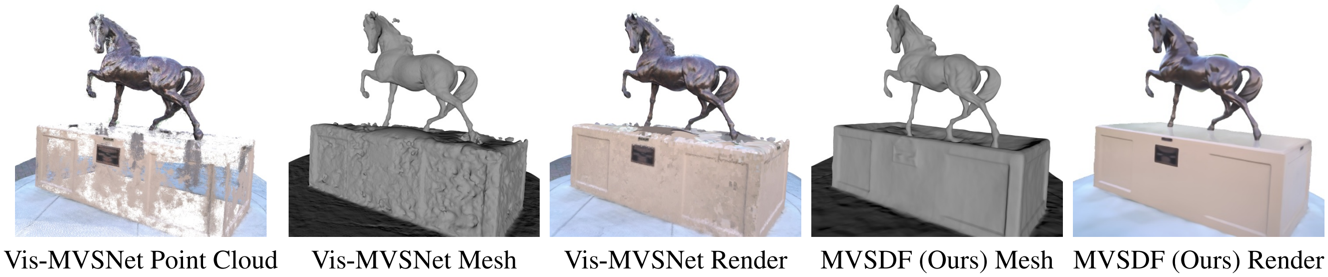

Vis-MVSNet Point Cloud Vis-MVSNet Mesh Vis-MVSNet Render MVSDF (Ours) Mesh MVSDF (Ours) Render

Figure 7. **Qualitative results on Tanks and Temples dataset.** Traditional methods produce holes in texture-less and reflective regions. In contrast, our end-to-end system is able to reconstruct accurate mesh and rendering results in these areas.

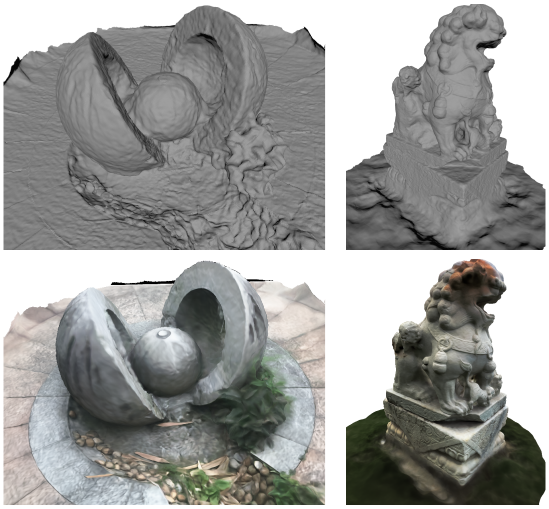


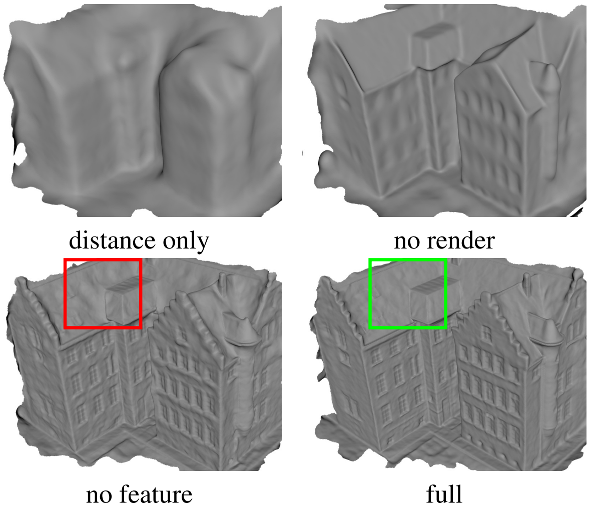
Figure 8. **Qualitative results on BlendedMVS dataset.** Meshes (upper) and rendered images (lower) of two scenes reconstructed by MVSDF are illustrated.

	L_D	L_F	L_R	Mean Chamfer (mm)
distance only	✓			3.56
no render	✓	✓		1.75
no feature	✓		✓	1.06
full	✓	✓	✓	0.88

Table 3. **Quantitative results of ablation studies on DTU dataset.** Both the feature loss and the render loss can improvement the mesh reconstruction quality.

by setting $w_R = 0$ and 3) *distance only*: both the feature loss and the render loss are disabled $w_R = w_F = 0$.

Mesh surface results of different settings are illustrated in Fig. 9. We find that the feature loss (from *distance only* to *no render*) can successfully refine the surface, but is still coarse compared to the *full* setting. The render loss is able to refine the model to its finest detail level (from *distance only* to *no feature*), however, it is not as robust as the feature loss and will causes erroneous surface in the roof area. Quantitative results are shown in Tab. 3. Both the feature loss and the render loss can significantly boost the reconstruction quality, showing the effectiveness of each component of the proposed method.


distance only

no render

no feature

full

Figure 9. **Ablation study on training losses.** Both the feature loss and the render loss are able to refine the surface geometry. Moreover, the reconstruction is more robust if the feature loss is applied.

5. Conclusion

In this work, we introduce a novel neural surface reconstruction framework that combines implicit neural surface estimation with recent advanced MVS networks. In our network, the geometry and appearance are represented as neural implicit functions by MLPs. The geometry is directly supervised by MVS depth maps to recover the surface topology and is locally refined via deep feature consistency and rendered image loss. The proposed method has been extensively evaluated on different datasets. Both qualitative and quantitative results have shown that our method outperforms previous methods in terms of both geometry accuracy and rendering fidelity, demonstrating the effectiveness of the proposed framework.

6. Acknowledgments

This work is supported by Hong Kong RGC GRF 16206819, 16203518, T22-603/15N and Guangzhou Okay Information Technology with the project GZETDZ18EG05.

References

[1] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, 2011. 2

[2] Neill DF 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

[3] Robert T Collins. A space-sweep approach to true multi-image matching. In *Computer Vision and Pattern Recognition (CVPR)*, 1996. 2

[4] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 4

[5] Amaël Delaunoy and Emmanuel Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International Journal of Computer Vision*, 95(2):100–123, 2011. 1, 4

[6] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 2

[7] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. In *European Conference on Computer Vision (ECCV)*, 2006. 1

[8] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2009. 1, 2

[9] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[10] Michael Goesele, Brian Curless, and Steven M Seitz. Multi-view stereo revisited. In *Computer Vision and Pattern Recognition (CVPR)*, 2006. 2

[11] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, 2020. 5

[12] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[13] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[14] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[15] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Neural Information Processing Systems (NIPS)*, 2017. 1, 2

[16] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 2

[17] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 1, 2, 7

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 2, 7

[19] Andreas Kuhn, Christian Sormann, Mattia Rossi, Oliver Erdler, and Friedrich Fraundorfer. Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In *International Conference on 3D Vision (3DV)*, 2020. 2

[20] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *International Conference on Computer Vision (ICCV)*, 2007. 1, 2

[21] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005. 2

[22] Shiwei Li, Sing Yu Siu, Tian Fang, and Long Quan. Efficient multi-view surface refinement with adaptive resolution control. In *European Conference on Computer Vision (ECCV)*, 2016. 4

[23] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Neural Information Processing Systems (NeurIPS)*, 2020. 2

[24] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2

[25] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[26] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 1, 2, 6

[27] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[29] 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, 4, 6

[30] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 4, 6, 7

[31] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[32] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[33] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[34] Jean-Philippe Pons, Renaud Keriven, and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007. 1

[35] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[36] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[37] 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, 2, 6, 7

[38] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition (CVPR)*, 2006. 2

[39] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2

[40] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[41] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012. 1

[42] Hoang-Hiep Vu, Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 34(5):889–901, 2011. 1, 4

[43] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[44] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[45] Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao. Mvscrf: Learning multi-view stereo with conditional random fields. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[46] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3

[47] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[48] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7

[49] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 3, 4, 6, 7

[50] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4

[51] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. In *British Machine Vision Conference (BMVC)*, 2020. 2, 3, 6, 7

[52] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[53] Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2