

# BundleRecon: Ray Bundle-Based 3D Neural Reconstruction

Weikun Zhang, Jianke Zhu\*

Zhejiang University

{zhangwk, jkzhu}@zju.edu.cn

## Abstract

*With the growing popularity of neural rendering, there has been an increasing number of neural implicit multi-view reconstruction methods. While many models have been enhanced in terms of positional encoding, sampling, rendering, and other aspects to improve the reconstruction quality, current methods do not fully leverage the information among neighboring pixels during the reconstruction process. To address this issue, we propose an enhanced model called BundleRecon. In the existing approaches, sampling is performed by a single ray that corresponds to a single pixel. In contrast, our model samples a patch of pixels using a bundle of rays, which incorporates information from neighboring pixels. Furthermore, we design bundle-based constraints to further improve the reconstruction quality. Experimental results demonstrate that BundleRecon is compatible with the existing neural implicit multi-view reconstruction methods and can improve their reconstruction quality.*

## 1. Introduction

Multi-view reconstruction is a fundamental problem in the fields of computer vision. The reconstruction quality of traditional methods is limited by the accuracy of feature matching or the voxel resolution of those explicit representation methods. With the introduction of neural rendering [2, 10, 19, 29], many researchers have switched from traditional methods to learning-based neural implicit representation approaches. These methods have an impressive performance on geometry reconstruction, as the implicit function can represent the geometry continuously with relatively low memory cost and high spatial resolution [10, 21]. It is worthy of noting that NeRF [10] was originally intended to solve the problem of novel view synthesis. Meanwhile, its geometry is extracted using an arbitrary level set of the density function, which makes the geometry estimated by NeRF to be less accurate [10, 26]. To address this issue,

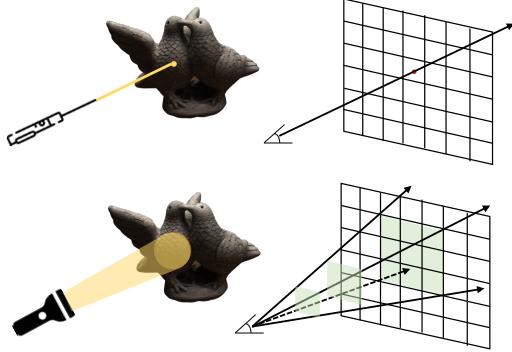


Figure 1. Traditional neural implicit representation methods work like illuminating the object by a laser pen, which provides the limited local information on the illuminated area. In contrast, BundleRecon employs a bundle of rays, which is similar to a flashlight so that the larger illuminated area provides more local information. This is beneficial to the reconstruction process.

researchers replace the density function by other implicit functions such as the occupancy network or the SDF network, from which they can obtain a more precise object representation.

For neural implicit representation methods, the sampling process is of vital importance because the selection of sampled points directly affects the quality of the reconstruction. Some studies prefer to the dense sampling near the surface area [17, 26], while others adopt a coarse-to-fine sampling process to cover a broader space [10, 12, 21]. Regardless of the sampling method, they all involve with randomly selecting pixels and using the corresponding ray to sample points. However, this commonly used approach only considers the longitudinal information along the ray while the transversal information between the rays is typically ignored. Therefore, the discrete pixels may result in a loss of information between neighboring pixels.

Inspired by the scenario of using a flashlight to illuminate an object in the dark, we propose BundleRecon to solve the above problem. As shown in Fig. 1, the traditional neural implicit representation methods work like illuminating the object by single ray. Instead, we employ a flashlight,

\*Corresponding author

i.e., a bundle of rays. This is because a larger patch of pixels in the field of view provides more useful information. This is more conducive to reconstruction process compared to a single pixel.

Additionally, we have designed bundle-based loss functions to adapt BundleRecon. By utilizing ray bundles to sample pixel patches, we extract bundle-shaped information and feed it to the neural network. Our loss functions make use of statistical information, such as mean and variance, from the bundle-shaped output, while utilizing its convolutional features for supervision.

We integrate our method into the existing method like IDR [27] and NeuS [21] and evaluate them on DTU [6] and BlendedMVS [25]. The experimental results show that using the information between neighboring pixels can enhance the reconstruction quality.

The contributions of our work are as follows:

- A ray bundle-based 3D neural reconstruction model is proposed to improve the reconstruction quality by exploiting the information between neighboring pixels.
- The bundle-based loss functions are designed for BundleRecon to further improve the reconstruction quality.
- The extensive experiments demonstrate the compatibility and effectiveness of our model, which can be used to improve the reconstruction quality of existing neural implicit multi-view reconstruction methods.

## 2. Related Work

Reconstructing objects from multiple images has been intensively studied for over thirty years. We briefly review the related work in the following.

### 2.1. Traditional Multi-view Reconstruction

Multi-view stereo is widely used in reconstruction and can be divided into point cloud reconstruction, volumetric method and depth map approach [24]. We take the depth map reconstruction as an example to introduce the pipeline of traditional multi-view reconstruction. Firstly, the multi-view input images are processed using structure from motion (SFM) [13, 16] to generate the camera parameters and sparse 3D points. Secondly, multi-view stereo (MVS) [3, 5, 14] is applied to obtain the depth information. Finally, the resulting depth maps can be fused into point clouds [9] that can be used by surface reconstruction methods [7]. Although the traditional pipeline has some inherent drawbacks, such as error accumulation through multiple steps, it treats the input image as a whole and considers the information between pixels. In contrast, many neural implicit multi-view reconstruction methods only select a limited number of discrete pixels from the input images,

thereby ignoring the information between pixels. Besides, there are some methods that explicitly represent the object. For instance, voxel grids are commonly used to representing objects [15, 20] even if the object is of arbitrary topology. However, the reconstruction quality is closely related to the resolution of the grids, which is limited by the system memory.

### 2.2. Neural Radiance Fields

The proposal of NeRF [10] has drawn increasing attention to the implicit representation of scenes. Many researchers have improved NeRF in several ways, such as positional encoding [2], sampling [2, 12, 17, 26], and rendering [1, 21, 28]. In contrast to methods that simply sample along the ray, Barron *et al.* have proposed a cone-based sampling model, named Mip-NeRF [2], which solves the aliasing problem by casting a cone from each pixel. While the shape of the cone is very similar to our ray bundle, Mip-NeRF's underlying principle is still based on sampling a single ray that corresponds to a single pixel, without considering information from surrounding pixels. In addition, there is work that applies NeRF to dynamic scenes and outdoor scenes [4, 8, 18].

NeRF was originally proposed to address the problem of novel view synthesis. However, it has since inspired researchers to explore new possibilities in 3D reconstruction. By leveraging the implicit density and color information captured in radiance fields, they discovered that it is possible to reconstruct the object using marching cubes. However, this approach often leads to poor reconstruction quality due to the fact that NeRF's density is obtained from arbitrary level sets. To overcome this limitation, researchers have begun to improve neural radiance fields by using SDF or other implicit representations to extract the geometric information of the scene, in hopes of achieving more precise 3D reconstruction results.

### 2.3. Neural Implicit Multi-view Reconstruction

Most of neural implicit multi-view reconstruction methods can be roughly categorized into two groups. The first group utilizes different implicit functions to represent the geometry while preserving the network structure of NeRF [12, 21, 26, 27]. The second category involves with designing a specialized network structure to represent complex scenes [8, 11, 22, 23], such as those featuring foreground and background elements or static and dynamic components. In this work, we focus on the first group of method. The implicit functions utilized in these works mainly fall into two categories: the occupancy network [12] and the signed distance function (SDF) network [12, 21, 26, 27]. UNISURF, proposed by Oechsle *et al.* [12], unifies the implicit surface model and volumetric radiance model and uses the occupancy network to represent geometry. Yariv *et al.* proposed

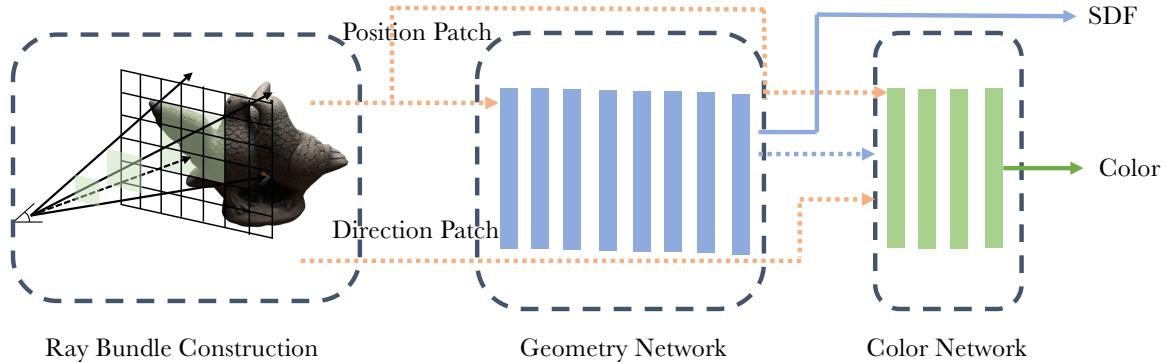


Figure 2. The BundleRecon pipeline consists of several stages. Firstly, bundle construction is performed on the input image and sampled along the rays. Secondly, the positional information is processed by the geometry network to generate the signed distance function (SDF) values and features. Finally, the direction information, along with the output of the geometry network, are fed into the color network to obtain the final color output.

IDR [27] and VolSDF [26], both of which adopt the SDF network. IDR recovers the surface by disentangling geometry and appearance, while VolSDF is an improved model based on IDR. The motivation of VolSDF is to overcome the shortcomings of neural volume rendering and neural implicit surface models. Specifically, it designs a neural volume rendering model that can effectively separate geometric information and appearance. Additionally, Wang *et al.* [21] present NeuS approach, which also employs the SDF network to represent geometry. NeuS represents density using the bell-shaped S-density function of SDF. Note that a bell-shaped density function is necessary to achieve an unbiased model effectively capturing thin structures [21].

### 3. Method

For a single pixel, the conventional approaches [12, 21, 26, 27] make use of a single ray to sample and perform neural rendering, which usually neglects the information around the pixel. In order to utilize the information from neighboring pixels, we propose a ray bundle-based method, named as BundleRecon, which is an improved sampling module that can be plugin into many existing networks.

In this section, we firstly introduce the pipeline of ray bundle-based neural reconstruction in 3.1. Then, we show how to construct a ray bundle in 3.2. The ray bundle-based loss functions will be discussed in 3.3. Finally, the implementation details are introduced in 3.4.

#### 3.1. Ray Bundle Based Neural Reconstruction

As shown in Fig. 2, BundleRecon is composed of three parts, including the ray bundle construction, the geometry network, and the color network. Given multi-view RGB images, the first step is to construct the ray bundle. To achieve this, we randomly select patches from each image to form

the ray bundle  $p \in P$ . Further details on the construction of the ray bundle can be found in Section 3.2. Once the ray bundle is constructed, we can either take surface samples using methods such as [27] or conduct dense sampling around the surface as in [21, 26]. The positional information of the sampled points  $x \in X_p$  is then fed into the geometry network, where  $X_p$  is the set of sampled points that form the ray bundle. Similarly,  $V_p$  is the set of directions of the rays that form the ray bundle.

The geometry network is defined by the signed distance function (SDF)  $f_g : \mathbb{R}^3 \rightarrow \mathbb{R}$ , which maps the spatial position of a point  $x \in \mathbb{R}^3$  to its signed distance from the object. As a result, the surface  $S$  of the object is represented by the zero-level-set of the geometry network, which can be expressed as follows

$$S = \{x | f_g(x) = 0\}. \quad (1)$$

The output of the geometry network, along with the direction information  $v \in V_p$ , is fed into the color network  $f_c : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ . This network maps the spatial position  $x \in \mathbb{R}^3$  and the ray direction  $v \in \mathbb{R}^3$ ,  $\|v\| = 1$ , to the color patch  $f_c(p)$ . In other words, given a specific point in space and the direction in which the ray travels, the color network produces a corresponding color value that is used to generate the final rendered color.

To obtain the rendered color, we perform volume rendering using the following equation,

$$\hat{C}(p) = \int_0^\infty w(t)f_c(p)dt, \quad (2)$$

where  $w(t)$  is the weight of the sampled point color.

There are various options for selecting the weights  $w(t)$ . We can adopt the unbiased and occlusion-aware weights designed by NeuS [21], or we can use regular weights in

NeRF [10]. To compute the weights, we firstly use a transformed SDF to represent the density at the sampled points, which is given by  $\sigma(x) = \alpha \cdot \rho(f_g(x))$ .  $\alpha$  is a hyperparameter and  $\rho(\cdot)$  is a transformation function, such as the cumulative distribution function of the Laplace distribution in VolSDF [26]. Once the density is defined, we can compute the probability of light traveling from the origin of the camera to the sampled point  $x(t)$  without bouncing off, which is given by

$$T(t) = \exp \left( - \int_0^t \sigma(x(s)) ds \right). \quad (3)$$

Therefore, the probability of light being reflected at sampled point  $x(t)$  and emitting color is defined as below

$$O(t) = 1 - T(t). \quad (4)$$

Thus, the weight of the sampled point color can be expressed as follows

$$w(t) = \frac{dO}{dt}(t) = \sigma(x(t))T(t). \quad (5)$$

In addition to the volume rendering mentioned above, BundleRecon can also use rendering methods in IDR [27].

$$\hat{C}(p) = L^e(\hat{x}, v^o) + \int_{\Omega} B(\hat{x}, \hat{n}, v^i, v^o) L^i(\hat{x}, v^i)(\hat{n} \cdot v^i) dv^i, \quad (6)$$

where  $B$  refers to the bidirectional reflectance distribution function (BRDF).  $\hat{x}$  is the surface point obtained by ray tracing.  $\hat{n}$  is the normal vector computed by SDF and  $\Omega$  is half sphere centered at  $\hat{n}$ .  $v^i$  and  $v^o$  represent the incoming and outgoing direction of the ray, respectively.

In summary, our model is compatible with the sampling and rendering methods for most of existing implicit models, such as NeRF [10], IDR [27], VolSDF [26], NeuS [21] and UNISURF [12].

### 3.2. Ray Bundle

We randomly select  $n$  pixels from the input image like the traditional methods [10, 12, 21, 26, 27]. Instead of simply considering each pixel in isolation, we employ it as the anchor of an  $s \times s$  pixel patch, which is named as a ray bundle. Each ray bundle consists of  $s^2$  pixels, and we construct  $n$  such bundles for a total of  $ns^2$  pixels. This is in contrast to traditional methods that only consider  $n$  rays with  $n$  pixels. As a result, our approach incurs a higher memory cost. To mitigate this, we may reduce the number of ray bundles used, however, this would inevitably compromise the quality of rendering and reconstruction, necessitating a trade-off. To reduce memory usage and enhance efficiency, when integrating BundleRecon into models [10, 21, 26] that have numerous sample points on a single ray, we can employ dense sampling for the central ray of each ray bundle.

Moreover, the sparse sampling is used for the surrounding rays. In contrast, models [27] that rely on ray tracing to obtain the surface point as the only sample point on each ray can readily incorporate BundleRecon without any changes to their sampling strategy.

As mentioned above, the ray bundle itself is a regular square patch. Fig. 3 illustrates two ways of constructing a square bundle: a fixed-size bundle, and a bundle whose size changes dynamically during training. Additionally, we can construct ray bundles using super-pixels that incorporate the semantic information.

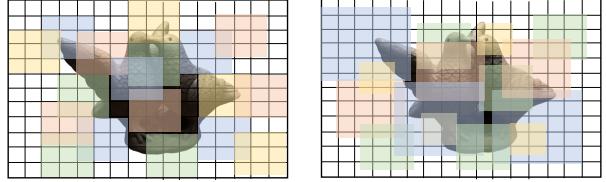


Figure 3. Each square in a different color represents a distinct epoch in the training process. The left figure indicates a fixed bundle size throughout the training, whereas the right figure depicts a varying bundle size during the training process.

Consider using a flashlight to illuminate an object. The resulting ray bundle may cover a flat area or intersect an edge, with some parts belonging to the front of the object and others to the back. In the latter case, the information on the two sides of the edge can be quite different, and it may not be necessary to treat them as a whole. To address this issue, we partition such areas based on the spatial location of the sampling points along the ray bundle. Specifically, we compute the distance between each pixel and its eight surroundings to generate a distance mask  $M_P$ . During training, this mask is used to filter out the irrelevant information.

### 3.3. Loss Function

Similar to the traditional methods [10, 12, 21, 26, 27], we employ color loss  $\mathcal{L}_c$  to supervise the color network as well as the geometry network. For a ray bundle, we have the following loss function

$$\mathcal{L}_c = \sum_{p \in P} \|\hat{C}(p) - C(p)\|_1, \quad (7)$$

where  $P$  denotes the set of all ray bundle,  $\hat{C}(p)$  is the rendered color with respect to the ray bundle  $p$ , and  $C(p)$  is the true color of the pixel patch.

To further leverage the information from neighboring pixels, we compute the mean and variance of the color patch. We denote the mean and variance operations as  $M(\cdot)$  and  $V(\cdot)$ , respectively. The loss  $\mathcal{L}_m$  and  $\mathcal{L}_v$  can be obtained as below

$$\mathcal{L}_m = \|M(\hat{C}(P)) - M(C(P))\|_2, \quad (8)$$

$$\mathcal{L}_v = \|V(\hat{C}(P)) - V(C(P))\|_2. \quad (9)$$

For color patches, we can also extract their convolutional features. To process the patch information, we need to apply the distance mask  $M_P$  to filter out irrelevant details from the color patches. We obtain the features of color patches by convolving them with a Sobel kernel, denoted as  $F(\cdot)$ . The loss function for this process is formulated as follows

$$\mathcal{L}_{conv} = \|(F(\hat{C}(P)) - F(C(P))) \cdot M_P\|_2. \quad (10)$$

### 3.4. Implementation Details

Our model can be easily integrated into the existing single ray based neural implicit models, so the architecture details can remain the same. In this paper, we demonstrate the effectiveness of our model using IDR [27] and NeuS [21]. It is noteworthy that, our model can also be plugged into other models like VolSDF [26]. During the experiment, we set the pixel patch size to be  $3 \times 3$ , which remained constant during training.  $n = 229$  ray bundles are used for sampling, which results in a batch size of 2061. We trained our model for 2000 epochs on IDR and 100K iterations on NeuS.

The loss function used in the experiment is as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_m \mathcal{L}_m + \lambda_v \mathcal{L}_v + \lambda_{conv} \mathcal{L}_{conv}, \quad (11)$$

where  $\lambda_c = 1$ ,  $\lambda_m = 5e^{-3}$ ,  $\lambda_v = 1e^{-2}$  and  $\lambda_{conv} = 5e^{-5}$ .

## 4. Experiments

### 4.1. Experimental Settings

**Datasets:** We evaluate our model on 15 scenes from the DTU dataset [6] and 4 scenes from the BlendedMVS dataset [25]. The DTU dataset includes multi-view images of various objects with the fixed camera and lighting parameters. Each image has a resolution of  $1600 \times 1200$ , and DTU also provides masks and real point cloud data, which facilitates mask processing during training and evaluation of the Chamfer distance of the reconstructed mesh. For the BlendedMVS dataset, masks and camera parameters are also provided. Their scenes have more complex backgrounds compared to DTU.

**Baseline.** We first integrate our proposed module into IDR [27] and use it as a baseline to evaluate the effectiveness of BundleRecon. Due to memory limitation, for each image, we used 229 ray bundles for the experiment. To ensure the fairness of the experiment, we lowered the number of sampled pixels of IDR to 229 for comparison as well. Besides, to validate the compatibility of BundleRecon, we integrate it into NeuS [21] to perform quantitative comparison. The number of the sampled pixels in NeuS is set to 299, and the number of ray bundle is set to 114. In this section, we use IDR\* and NeuS\* to represent our different settings from the original model.

### 4.2. Multi-view 3D Reconstruction

The qualitative results of IDR+BundleRecon is presented in Fig. 5. It is evident that the mesh generated by IDR+BundleRecon captures more details. The first and the second scenes demonstrate that the mesh generated by our model has more refined geometric textures. In the third scene, the mesh generated by IDR+BundleRecon eliminates the geometric errors in the snowman's face. In the fourth scene, the mesh generated by IDR+BundleRecon exhibits better performance in the area around the arm. One of the reconstruction details is shown in Fig. 4. Traditional models such as IDR require sampling as many pixels as possible to obtain the object details, which may lead to the incorrect geometry when the number of sampled pixels is reduced, as seen in the hole on the left. Fortunately, the incorporation of BundleRecon can effectively alleviate this issue.

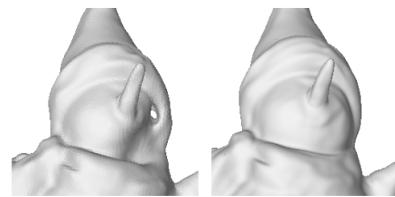


Figure 4. Comparison of reconstruction details. There are holes in the mesh generated by IDR\* (left), and the correct geometry can be obtained by integrating BundleRecon (right).

We trained NeuS+BundleRecon on the BlendedMVS dataset with masks. Moreover, the qualitative results are shown in Fig. 6. In the first scene, BundleRecon removes the geometric depressions on the dog's leg. In the second scene, we can see that BundleRecon fills the incomplete petals in the lower left corner of the jade. In the third scene, BundleRecon provides more refined hair textures. And in the last scene, BundleRecon better handles the concave part of the stone.

Table 1 summarizes the quantitative results of BundleRecon on the DTU dataset. IDR\* and IDR+BundleRecon are trained with masks, while NeuS\* and NeuS+BundleRecon are trained without masks. The table records both the chamfer distance of the reconstructed meshes and the PSNR of the rendered images, which clearly indicate that incorporating BundleRecon leads to improved performance for both IDR\* and NeuS\*. These results also demonstrate the compatibility of BundleRecon with the existing methods.

### 4.3. Ablation Study

Firstly, we investigate the effectiveness of our loss functions through ablation studies. The first experiment involves replacing the single ray with the ray bundle while still utilizing the original loss function of IDR. In exp 2 and exp 3, we introduce the additional constraints for mean and variance,

Table 1. Quantitative Comparison Results

scan	IDR*		IDR+BundleRecon		NeuS*		NeuS+BundleRecon	
	Chamfer	PSNR	Chamfer	PSNR	Chamfer	PSNR	Chamfer	PSNR
24	2.03	21.99	<b>1.82</b>	<b>22.36</b>	1.42	22.98	<b>1.31</b>	<b>24.41</b>
37	<b>2.02</b>	19.50	2.12	<b>19.95</b>	1.58	19.15	<b>1.45</b>	<b>21.99</b>
40	0.87	23.81	<b>0.85</b>	<b>24.32</b>	1.59	23.16	<b>1.18</b>	<b>25.41</b>
55	0.52	20.33	<b>0.45</b>	<b>21.63</b>	0.62	<b>20.94</b>	<b>0.54</b>	20.65
63	1.59	21.79	<b>1.31</b>	<b>22.58</b>	<b>1.79</b>	25.17	1.81	<b>25.36</b>
65	1.05	<b>23.42</b>	<b>1.03</b>	23.33	<b>0.87</b>	<b>28.15</b>	0.88	27.90
69	0.93	<b>20.84</b>	<b>0.90</b>	20.65	<b>0.70</b>	<b>25.44</b>	0.79	24.84
83	<b>1.29</b>	19.63	1.48	<b>24.32</b>	1.59	27.02	<b>1.16</b>	<b>30.91</b>
97	<b>1.46</b>	21.91	1.55	<b>22.20</b>	1.46	<b>23.87</b>	<b>1.45</b>	23.65
105	0.73	20.96	<b>0.70</b>	<b>22.41</b>	1.29	25.29	<b>1.09</b>	<b>27.30</b>
106	<b>0.74</b>	<b>20.77</b>	0.77	20.28	0.69	31.04	<b>0.64</b>	<b>31.90</b>
110	1.41	20.64	<b>1.29</b>	<b>20.88</b>	2.05	26.54	<b>1.81</b>	<b>27.02</b>
114	0.56	23.44	<b>0.38</b>	<b>25.34</b>	0.47	26.12	<b>0.42</b>	<b>26.34</b>
118	0.65	21.75	<b>0.58</b>	<b>22.96</b>	0.62	31.27	<b>0.57</b>	<b>32.66</b>
122	0.60	25.81	<b>0.57</b>	<b>25.97</b>	<b>0.63</b>	<b>30.95</b>	0.68	27.60
mean	1.10	21.77	<b>1.05</b>	<b>22.61</b>	1.16	25.81	<b>1.05</b>	<b>26.53</b>

Table 2. Ablation Study on Loss Functions.

Method	Bundle	M+V $l_1$	M+V $l_2$	Laplace	Sobel	normal	Chamfer
exp1	✓						2.07
exp2	✓	✓					2.08
exp3	✓		✓				<b>1.98</b>
exp4	✓			✓			2.15
exp5	✓			✓	✓		<b>1.82</b>

Table 3. Results on Different Ray Bundle Settings.

Method	BundleSize	BundleNum	Chamfer
exp6	$3 \times 3$	229	<b>1.82</b>
exp7	$5 \times 7$	229	1.92
exp8	$7 \times 7$	229	2.09
exp9	$3 \times 3$	57	2.15
exp10	$3 \times 3$	114	1.94
exp11	$3 \times 3$	229	<b>1.82</b>

with different norms applied for these constraints. Furthermore, in exp 4 and exp 5, we utilize the different kernels to extract convolutional features. All these experiments are conducted on DTU scan 24. The results of our ablation study on the loss functions are presented in Table 2. It can be observed that the  $l_2$  norm outperforms  $l_1$  norm in terms of mean and variance constraints. More importantly, the Sobel kernel, which introduces the first order derivative information, performs better than the Laplace convolution kernel.

We also perform the ablation study on ray bundle settings

with DTU scan 24, and present the results in Table 3. The experimental results demonstrate that enlarging the size of the ray bundle may result in poor performance with a fixed number of ray bundles. This is because the excessive information around the pixels can impede the model from converging effectively. Therefore, a larger size of ray bundle is unnecessarily better. However, we observed that increasing the number of ray bundles while keeping the bundle size fixed can lead to better reconstruction results. This indicates that, if the memory cost issue can be resolved, adopting more ray bundles could yield further improvements in reconstruction quality.

## 5. Conclusion

We present BundleRecon, a novel 3D reconstruction module for multiview implicit reconstruction that incorporates information from adjacent pixels. Our method is accompanied by a set of bundle-based loss functions to effectively constrain it. The experiments demonstrate that BundleRecon is compatible with the existing single ray based neural implicit models and can be seamlessly integrated into them to enhance their reconstruction quality.

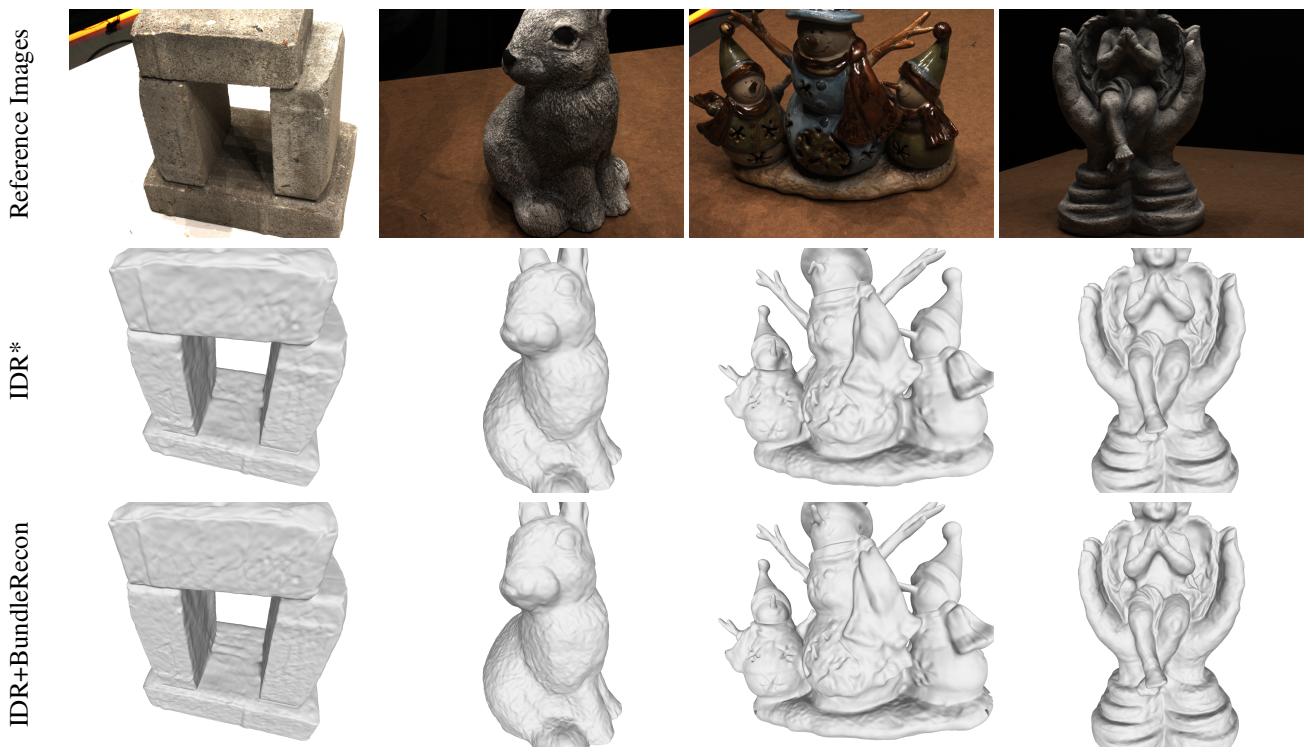


Figure 5. Qualitative comparison of IDR\* and BundleRecon on DTU

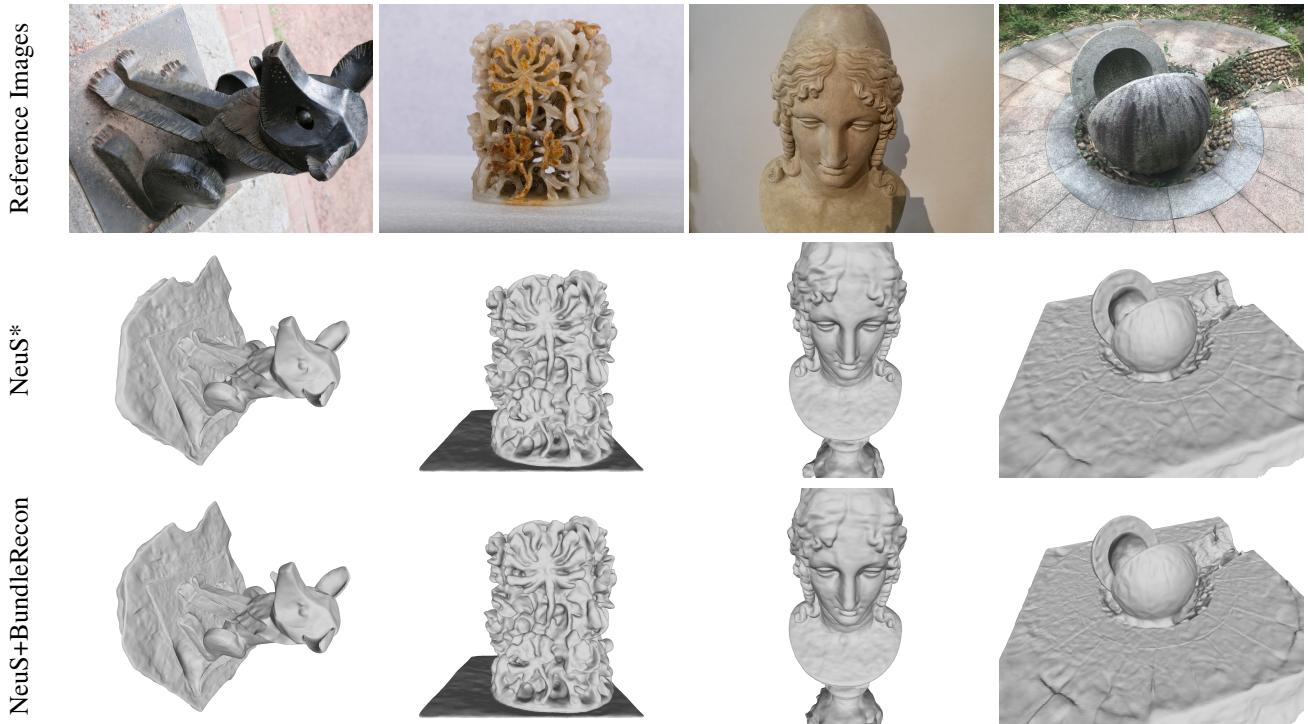


Figure 6. Qualitative comparison of NeuS\* and BundleRecon on BlendedMVS

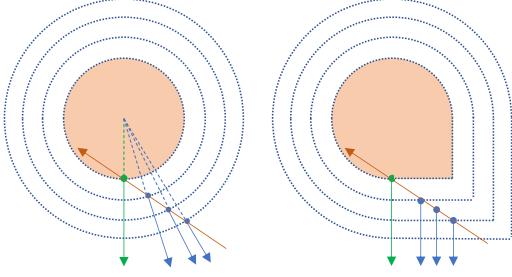


Figure 7. The green point represents the actual surface point, while the blue points represent the sampled points along the ray. As shown in the left figure, the points along the ray have different normals. If we artificially force these normal vectors to be the same, it will result in geometric distortion, as shown in the right figure.

Our future research will focus on reducing the memory cost associated with the ray bundle to match the number of single rays used in current works. This will further improve reconstruction quality. More importantly, we found that introducing ray bundles makes it possible to impose direct constraints on geometry network. Below we present a possible constraint approach, and there are still many details worth exploring.

As the training progresses, the sampled points gradually converge towards the true surface of the object. Based on the depth information and the ray direction, we can obtain the spatial coordinates of sampled points, which facilitates the normal computation. One possible way to impose constraints is to minimize the difference between the computed normal vector and the corresponding normal vector derived from SDF. However, this approach may cause geometric deformation. To illustrate this, we used 2D objects and their SDF isolines, as shown in Fig. 7. When we sampled a series of points along the ray, these points had different normal vectors, as shown on the left. When we enforced the normal vectors of these points to be the same, the geometry was deformed, as shown on the right. We believe that a possible solution to this problem is to concentrate the sampling points near the surface and add normal vector constraints at a later stage of training.

## Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants (61831015).

## References

- [1] Alex Yu and Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks, 2021. [2](#)
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. [1, 2](#)
- [3] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision-ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10*, pages 766–779. Springer, 2008. [2](#)
- [4] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022. [2](#)
- [5] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. [2](#)
- [6] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. [2, 5](#)
- [7] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006. [2](#)
- [8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. [2](#)
- [9] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. Ieee, 2007. [2](#)
- [10] 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, 4](#)
- [11] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. [2](#)
- [12] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. [1, 2, 3, 4](#)
- [13] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. [2](#)

- [14] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. [2](#)
- [15] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1067–1073. IEEE, 1997. [2](#)
- [16] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25:835–846, 2006. [2](#)
- [17] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3d reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. [1, 2](#)
- [18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. [2](#)
- [19] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason M. Saragih, Matthias Nießner, Rohit Pandey, Sean Ryan Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B. Goldman, and Michael Zollhöfer. State of the art on neural rendering. *Comput. Graph. Forum*, 39(2):701–727, 2020. [1](#)
- [20] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017. [2](#)
- [21] 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. *arXiv preprint arXiv:2106.10689*, 2021. [1, 2, 3, 4, 5](#)
- [22] Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown. Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling. In *International Conference on 3D Vision (3DV)*, 2021. [2](#)
- [23] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *International Conference on Computer Vision (ICCV)*, 10 2021. [2](#)
- [24] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. [2](#)
- [25] 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 *Proceedings of the IEEE/CVF Conference on Computer Vi-*  
*sion and Pattern Recognition*, pages 1790–1799, 2020. [2, 5](#)
- [26] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. [1, 2, 3, 4, 5](#)
- [27] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. [2, 3, 4, 5](#)
- [28] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. [2](#)
- [29] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. [1](#)