CPT-VR: Improving Surface Rendering via Closest Point Transform with View-Reflection Appearance

Zhipeng Hu[1,3], Yongqiang Zhang[1,3,*], Chen Liu[2], Lincheng Li[3,*], Sida Peng[3], Xiaowei Zhou[3], Changjie Fan[1], and Xin Yu[2]

[1] NetEase Fuxi AI Lab, Hangzhou, China
{zhangyongqiang02,zphu,lilincheng,fanchangjie}@corp.netease.com
[2] The University of Queensland, Queensland, Australia
chen.liu7@uqconnect.edu.au, xin.yu@uq.edu.au
[3] Zhejiang University, Hangzhou, China
{pengsida,xwzhou}@zju.edu.cn

Abstract. Differentiable surface rendering has significantly advanced 3D reconstruction. Existing surface rendering methods assume that the local surface is planar, and thus employ linear approximation based on the Singed Distance Field (SDF) values to predict the point on the surface. However, this assumption overlooks the inherently irregular and non-planar nature of object surfaces in the real world. Consequently, the approximate points tend to deviate from the zero-level set, affecting the fidelity of the reconstructed shape. In this paper, we propose a novel surface rendering method termed **CPT-VR**, which leverages the **C**loset **P**oint **T**ransform (CPT) and **V**iew and **R**eflection direction vectors to enhance the quality of reconstruction. Specifically, leveraging the physical property of CPT that accurately projects points near the surface onto the zero-level set, we correct the deviated points, thus achieving an accurate geometry representation. Based on our accurate geometry representation, incorporating the reflection vector into our method can facilitate the appearance modeling of specular regions. Moreover, to enable our method to no longer be dependent on any prior knowledge of the background, we present a background model to learn the background appearance. Compared to previous state-of-the-art methods, CPT-VR achieves better surface reconstruction quality, even for cases with complex structures and specular highlights.

Keywords: Surface Rendering · 3D Reconstruction · Differentiable Rendering · Closest Point Transform

1 Introduction

Differentiable rendering plays a crucial role in 3D content creation, especially showing greater promise in multi-view 3D reconstruction and text-to-3D generation. Recent advancements have utilized differentiable rendering for attaining

* Corresponding authors.

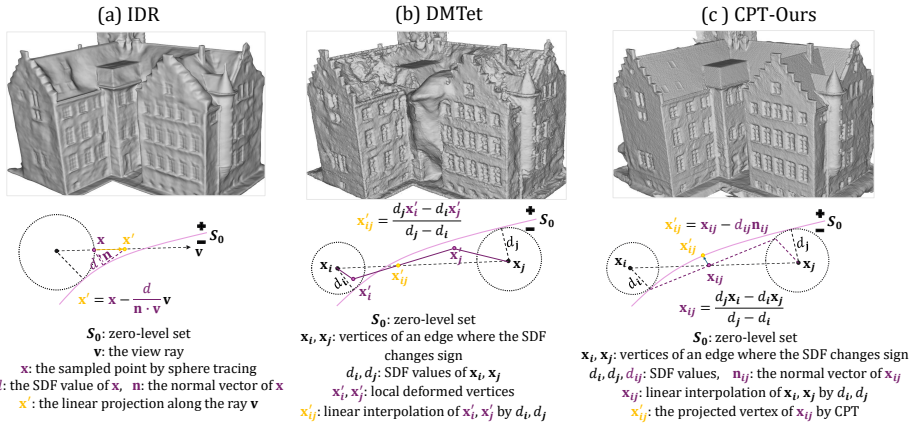

Fig. 1: Illustration of the surface point sampling methods and their corresponding construction results. (a) IDR [34] first employs a sphere tracing method to identify the sampled point x and then leverages the SDF value as well as the normal vector to obtain the approximate point $x^{'}$. (b) DMTet [19] detects surfaces through the observation of sign changes in the SDF at the vertices of a tetrahedron. Then, it employs linear interpolation to approximate the points $x^{'}_{ij}$ on the zero-level set. Compared with them, our method (c) utilizes CPT to project the interpolated points x_{ij} on the zero-level set, and thus attain the transformed points $x^{'}_{ij}$. The approximate points $x^{'}$ of IDR and $x^{'}_{ij}$ of DMTet are deviated from the zero-level set, while $x^{'}_{ij}$ obtained from our method exactly on the object surface. This implies our method is able to obtain a more accurate geometry representation and gradient backpropagation.

the Singed Distance Field (SDF) and object appearance. These methods can be divided into two types: volume rendering and surface rendering. In volume rendering, SDF is first converted into density values that are then rendered via the NeRF framework [17, 38]. While these methods consistently generate reliable geometry reconstruction, the dense points sampling along the ray increases the rendering time cost. Conversely, surface rendering methods only sample one point along the ray, markedly optimizing rendering speeds.

Existing surface rendering methods, such as IDR (Implicit Differentiable Renderer) [34] and DMTet (Deep Marching Tetrahedra) [19] achieve computational efficiency by assuming the object surface can be considered linear or at least can be approximated as linear within a local scope. Based on this assumption, they employ linear interpolation to determine the points at specific locations on the surface. As shown in Fig. 1, IDR employs sphere tracing [7] to identify points close to the surface along a ray and then relies on the SDF values and the normal vector to calculate the intersection points between rays and the surface. DMTet identifies the surfaces by observing changes in the sign of the SDF at the vertices of a tetrahedron and then leverages linear interpolation to approximate the surface points. However, as demonstrated in Fig. 1, their approximate points always deviate from the object surface (*i.e.,* zero-level set), since the object surface is mostly irregular and non-planar. Hence, modeling geometry representations based on these deviated points would further lead to inaccurate gradient back-

propagation, resulting in the reconstructed surfaces losing a lot of details, as illustrated by the reconstruction results in Fig. 1 (a) and (b).

To address the above problem, we introduce a novel surface rendering approach, termed **CPT-VR**, which is grounded in the **C**loset **P**oint **T**ransform algorithm and augmented with **V**iew-**R**elfection appearance modeling. In a nutshell, in the geometry representation modeling process, we conduct the Closet Point Transform (CPT) for deviated points to enhance the geometry representation. In the appearance modeling process, we intend to incorporate the View and Reflection vectors and thus attain better appearance representation for cases with specular highlights. To the best of our knowledge, we are the first attempt to correct the deviated points via CPT in the surface rendering task.

To be specific, in the geometry representation modeling process, we first devise a neural SDF network to compute SDF values for the entire 3D space. Based on the SDF values, we employ the Marching Cubes [14] to attain the vertices close to the surface of the object. As suggested in Fig. 1 (c), most of the obtained vertices are deviated from the object's surface rather than on it. Hence, we leverage CPT to calculate the closest point on the object surface for each deviated vertex, thus obtaining accurate and differentiable vertices. Subsequently, we utilize a feature-based rasterizer to render this triangle mesh in each view. In this fashion, we obtain the corresponding 3D position, geometry feature, and normal vector of each pixel in an image.

In the appearance representation modeling process, we focus on improving the geometry quality for specular highlights. Existing methods such as IDR [34] and NeuS2 [25] rely solely on the view direction to capture the surface information of objects, which cannot handle the object with highlights well. Ref-NeRF [21] demonstrates that replacing the view direction with the reflection direction can mitigate the impact of highlights. However, since the reflection vector calculated based on the deviated points is inaccurate, directly incorporating these reflection vectors into existing surface rendering methods will lead to significant backpropagation errors. Consequently, the model will not only be unable to handle the specular highlights but will also diminish its capability to reconstruct non-specular areas. In contrast, benefiting from our accurate point sampling on the zero-level set based on CPT, incorporating view-reflection vectors in our framework can facilitate the appearance modeling of specular regions.

Moreover, we found that existing surface rendering methods [1,22,34] require prior background knowledge, such as foreground masks, or the background image, to precisely process and render foreground objects, which constrains their flexibility and applicability in various scenes. To solve this problem, we present a background model to obtain the background appearance representation. Then we attain the final appearance modeling by blending the foreground and background appearance. Extensive experiments conducted on DTU [9] and BlendedMVS [31] demonstrate that our method achieves state-of-the-art geometry quality and surface rendering. Additionally, we also illustrate that our method is effective and reliable in cases with complex structures and specular highlights. In summary, our contributions are three-fold:

- We present a novel differentiable surface rendering method CPT-VR which shows superior performance in reconstructing the detailed structure of objects. With the CPT algorithm, we can address the approximate point deviation issue caused by linear interpolation.
- We incorporate the view and reflection vectors to model the foreground appearance representation, facilitating our method to be robust against specular highlights.
- We develop a simple background model to generate the background appearance representation, enabling our method to no longer be dependent on any prior background knowledge.

2 Related Work

Surface Rendering with SDF. IDR [34] introduces a system that learns surfaces in a self-supervised manner. It first identifies points close to the surface via sphere tracing and then finds approximate intersection points on the surface by projecting them along the view ray. Although IDR can learn the 3D shape and appearance from images, it relies heavily on accurate masks. To mitigate this issue, MVSDF [37] supervises the SDF network with depth maps generated by Vis-MVSNET [36]. It also uses relaxed masks generated by probability maps to learn a surface indicator. RegSDF [35] further introduces the Hessian regularization and the minimal surface constraint. Some works [1, 22] use specialized reparameterization techniques and design weighting functions along a ray based on SDF values, thus they can propagate gradients to intermediate points from sphere tracing. These reparameterization-based methods are similar to methods based on volume rendering, but they still need a given background: either black or a pre-defined environment map.

Rasterization-based Rendering. Recently, there have been attempts to utilize the differentiable rasterizer for surface rendering. NVDIFFREC [18] and NVDIFFRECmc [8] adapt DMTet [19] to extract a mesh from an implicit SDF and then render the mesh by the differentiable rasterizer Nvdiffrast [10]. While these methods are efficient, their reliance on the linear approximation in DMTet limits their ability to recover shapes with complex structures. Meanwhile, NIE [16] propagates neural implicit surfaces according to a flow field from a certain energy function. ENS [23] combines neural deformation fields with rasterization rendering, allowing deform an initial shape into a target surface progressively. Moreover, NDS [29] and FastMESH [39] focus on optimizing a rough mesh derived from priors through differentiable rasterization.

Volume Rendering of SDF. Several methods [2, 20, 24, 32] have been developed for learning SDF through volume rendering. VolSDF [32] and NeuS [24] leverage Laplace or Logistic cumulative distribution function of SDF to represent the volume density and sample dense points along a ray based on SDF values. Based on these methods, some works [2,3,12,26–28,40,41] aim to further enhance the geometry details. Among them, Geo-Neus [3] and TUVR [40] improve SDF accuracy by introducing additional supervision from point clouds. On the training efficiency front, Voxurf [30] and Vox-Surf [11] speed up the training process

through voxel-based representations. NeuS2 [25] speeds up NeuS [24] by simplifying the second-order derivatives calculation of the SDF network constructed based on a multi-resolution hash encoding.

Hybrid Volume-rasterization Rendering. Recently, some rendering methods have been developed based on a hybrid volume-rasterization way. BakedSDF [33] first learns the SDF and appearance of a scene based on VolSDF [32] and transforms the SDF into a mesh. Then the transformed mesh is refined using a rasterizer pipeline. VMesh [6] takes a comprehensive approach by initially training a combination of a neural SDF and a density field through volume rendering. Then the mesh undergoes further refinement by DMTet [19] and a differentiable rasterizer [10], ensuring accurate surface representation.

3 Method

3.1 Preliminaries

Signed Distance Field (SDF) is a mathematical function f that measures the shortest distance from a point $\mathbf{x}$ to an object surface, indicating whether the point is outside (positive distance) or inside (negative distance) the surface. The surface is identified by the zero-level set $\mathcal{S}$, where SDF equals to zero, providing a concise description of the object's surface:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\}. \tag{1}$$

Neural SDF denotes a neural network that models SDFs for 3D objects. In this work, we employ SDFnet which leverages the Multilayer Perceptrons (MLPs) enhanced with a multi-resolution hash encoding to model SDF [25]:

$$(d, \mathbf{z}) = f_\theta(\mathbf{x}, h_\Omega(\mathbf{x})). \tag{2}$$

Here, d is the predicted SDF value, $\mathbf{z}$ signifies the predicted geometry feature, f_θ denotes an MLP parameterized by weights θ, and $h_\Omega(\mathbf{x})$ is the multi-resolution hash encoding of the input point $\mathbf{x}$. Moreover, the normal vector $\mathbf{n}$ of $\mathbf{x}$ is derived as the normalized gradient of the implicit surface:

$$\mathbf{n} = \nabla_\mathbf{x} d / \|\nabla_\mathbf{x} d\|. \tag{3}$$

To ensure training stability, our SDFnet employs an optimization schedule that progressively incorporates the level of detail and the numerical gradients [12].

3.2 Geometry Representation Modeling

Geometry Representation via SDF. As suggested in Fig. 2, we first leverage our SDFnet to compute the SDF values for the entire 3D space. Based on SDF values, we utilize the isosurface algorithm Marching Cubes to extract and represent the triangle mesh $\mathcal{G} = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ and $\mathcal{F}$ indicate the vertices and faces respectively. However, this method assumes that the surface changes

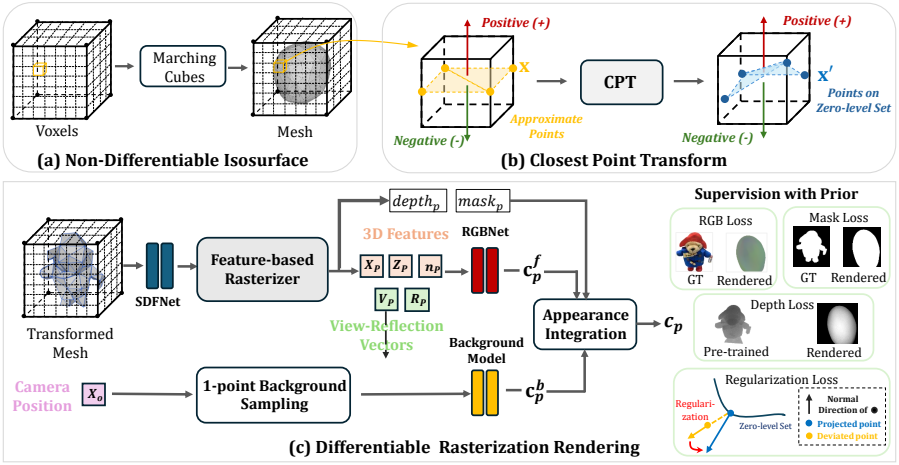

Fig. 2: Illustration of our proposed method CPT-VR. In (a), we first extract the mesh via Marching Cubes, which does not involve gradient propagation. Then we leverage CPT to project the approximate points on the zero-level set (b). Subsequently, we render the transformed mesh via a rasterizer in (c). Specifically, we incorporate the view-reflection vectors into the foreground appearance modeling stage via RGBNet. Additionally, we employ the Background Model to attain the background information. Finally, we integrate the foreground and background appearance to generate the rendering results. We adopt four supervisions to optimize these neural networks.

linearly inside the voxel, overlooking the complex convex structure of the object surface. Hence, the linearly interpolated vertices tend to deviate from the zero-level set and the triangle mesh may not accurately reflect the complex structure of the object surface.

Closest Point Transform. To project these deviated vertices onto the zero-level set, we devise the closet point transform (CPT) to SDF. Specifically, CPT first utilizes the norm vector $\mathbf{n}$ within the signed distance field to identify the projected direction of a deviated point. It then moves the deviated point along the opposite direction of $\mathbf{n}$ by a distance d equal to its SDF value. Mathematically, for a vertex $\mathbf{x} \in \mathcal{V}$, this transformation process is expressed as follows:

$$\mathbf{x}' = \mathbf{x} - d\mathbf{n}. \tag{4}$$

After that, we obtain the transformed mesh $\mathcal{G}' = (\mathcal{V}', \mathcal{F})$.

3.3 Feature-based Rasterisation

After obtaining the transformed mesh, we first employ a feature-based differentiable rasterizer Nvdiffrast [10] to render triangle meshes in each view. Specifically, as suggested in Fig. 2 (c), for a pixel $p(x, y)$, its projected primitive triangle face within the mesh is represented as $(\mathbf{x}'_0, \mathbf{x}'_1, \mathbf{x}'_2)$. The rasterizer computes the barycentric coordinates u, v, depth D_p, and visible mask M_p for this pixel as

follows:

$$(u, v, D_p, M_p) = \mathcal{R}(\mathcal{G}', p(x, y), K, R, \mathbf{t}), \tag{5}$$

where $\mathcal{R}$ denotes the rasterizer, $\mathcal{G}'$ represents the transformed mesh, and K and $R, \mathbf{t}$ are the camera's intrinsic and extrinsic parameters, respectively.

Subsequently, each projected vertex $\mathbf{x}'$ from the set $\mathcal{V}'$ is input into SDFnet. SDFnet outputs $(d', \mathbf{z}', \mathbf{n}')$, encapsulating the distance, geometric feature, and the normal vector associated with each vertex. These outputs are then interpolated for a pixel $p(x, y)$ based on its barycentric coordinates u, v:

$$\begin{aligned} \mathbf{x}_p &= u\mathbf{x}'_0 + v\mathbf{x}'_1 + (1-u-v)\mathbf{x}'_2, \\ \mathbf{z}_p &= u\mathbf{z}'_0 + v\mathbf{z}'_1 + (1-u-v)\mathbf{z}'_2, \\ \mathbf{n}_p &= u\mathbf{n}'_0 + v\mathbf{n}'_1 + (1-u-v)\mathbf{n}'_2, \end{aligned} \tag{6}$$

where $(\mathbf{x_p}, \mathbf{z_p}, \mathbf{n_p})$ denote the interpolated features.

3.4 Appearance Representation Modeling

In this work, we divide the appearance modeling process into two parts:

Foreground Appearance Modeling based on View-Reflection Vectors. To better capture the appearance of objects with specular highlights, we incorporate the view direction and reflection direction into the foreground modeling process. Specifically, we employ the RGBnet g^f_γ, a Multilayer Perceptron (MLP) with weights γ to determine the foreground color. RGBnet takes the interpolated 3D features $(\mathbf{x_p}, \mathbf{z_p}, \mathbf{n_p})$, view direction $\mathbf{v}_p$, and reflection direction $\mathbf{r}_p$ as input, and outputs the RGB color $\mathbf{c}^f_p$ for the foreground. The process is formulated as:

$$\mathbf{c}^f_p = g^f_\gamma(\mathbf{x_p}, \mathbf{z_p}, \mathbf{n_p}, Y(\mathbf{v_p}), Y(\mathbf{r_p})), \tag{7}$$

where $Y(\cdot)$ denotes Spherical Harmonics encoding. Here, the reflection direction $\mathbf{r_p}$ is defined as the mirror vector of the view direction $\mathbf{v_p}$ relative to the normal vector $\mathbf{n_p}$, calculated as:

$$\mathbf{r_p} = 2(-\mathbf{v_p} \cdot \mathbf{n_p})\mathbf{n_p} + \mathbf{v_p}. \tag{8}$$

Spherical harmonics are orthogonal bases on the unit sphere, and their mathematical definition and physical properties make them ideal for encoding direction frequencies compared to 3D space (xyz) frequencies.

Background Appearance Modeling. To enable the model to reconstruct objects without any background priors, we present a one-point background model composed of a Multilayer Perceptron (MLP) g^b_π with weights π, alongside an additional multi-resolution hash encoding h^b_Ω. The modeling process of the background appearance $\mathbf{c}^b_p$ is formulated as:

$$\mathbf{c}^b_p = g^b_\pi(\mathbf{x}^b_p, h^b_\Omega(\mathbf{x}^b_p), Y(\mathbf{v}_p)), \tag{9}$$

where $\mathbf{x}^b_p$ represents the background sampling point, determined by the camera location $\mathbf{x}_o$, the veiw direction $\mathbf{v}_p$ and the far distance F_p of the sampling

boundary. $\mathbf{x}_p^b$ is mathematically expressed as:

$$\mathbf{x}_p^b = \mathbf{x}_o + 2F_p\mathbf{v}_p. \tag{10}$$

Following NeuS [24], the foreground is normalized within a unit sphere centered at $\mathbf{0}$. As indicated in Fig. 3, the far distance F_p is calculated as:

$$m = -\mathbf{x}_o \cdot \mathbf{v}_p; F_p = m + 1. \tag{11}$$

This approach facilitates a more versatile background representation, compensating for the absence of masks.

Appearance Integration. Finally, we integrate the background appearance $\mathbf{c}_p^b$ and foreground appearance $\mathbf{c}_p^f$ to attain the rendered color, defined as:

$$\mathbf{c}_p = M_p\mathbf{c}_p^f + (1 - M_p)\mathbf{c}_p^b. \tag{12}$$

Overall, our method enables surface rendering of objects without any known background information, offering greater flexibility and adaptability.

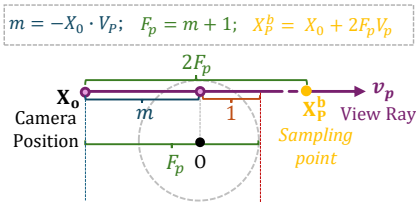

Fig. 3: Illustration of the one-point background sampling.

3.5 Training Objectives

Color loss. We optimize the CPT-VR model by ensuring the colors it generates match real image colors as closely as possible. This is done by calculating the L1 error, which is the average difference between the colors $\hat{\mathbf{c}}_i$ produced by the model and the ground-truth colors $\mathbf{c}_i$ for all pixels P:

$$\mathcal{L}_C = \frac{1}{P}\sum_i^P \|\hat{\mathbf{c}}_i - \mathbf{c}_i\|_1. \tag{13}$$

Regularization of SDFnet. A well-defined neural SDF exhibits Lipschitz continuity [13] and its gradient adheres to the Eikonal equation [4], ensuring smoothness and stability of the generated field. A valid SDF field holds the property that the normal of a point near the surface and the normal of its projected point remain consistent. For effective closest point transform, we directly constrain our SDFnet based on this normal consistency property [15]. Specifically, it aligns the normal of a deviated point $\mathbf{n}$ with the normal $\mathbf{n}'$ of its projected point, enhancing stability for surface rendering via the closest point transform of SDF. Mathematically, it is represented by:

$$\mathcal{L}_{nc} = 1 - \mathbf{n} \cdot \mathbf{n}'. \tag{14}$$

Mask Loss. If the mask of the foreground region is available, we can further refine the rendering effect of CPT-VR by including a mask loss. The mask constraint is formulated as:

$$\mathcal{L}_M = \frac{1}{P}\sum_i^P (\hat{M}_i - M_i)^2, \tag{15}$$

where the $\hat{M}_i$ and the M_i represent the predicted mask and the ground truth mask, respectively.

Depth loss. Inspired by previous works [37] and [35], we utilize depth maps generated by the pre-trained network CasMVSNet [5] to guide our model. We minimize the squared difference (MSE) between the rendered depths ($\hat{D}_i$) and those from the pre-trained network (D_i), weighted by the confidence $prob_i$ from CasMVSNet, as follows:

$$\mathcal{L}_D = \frac{1}{P} \sum_{i}^{P} prob_i (\hat{D}_i - D_i)^2, \tag{16}$$

where the confidence ranges from 0 to 1, and depths considered unreliable are given a confidence of 0.

Total loss. Our final metric combines all these aspects, adjusting their importance with specific weights λ:

$$\mathcal{L} = \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_{nc}\mathcal{L}_{nc} + \lambda_M\mathcal{L}_M + \lambda_D\mathcal{L}_D. \tag{17}$$

Here, λ_{rgb}, λ_{nc}, λ_M, and λ_D are empirically set to 3, 0.01, 1, and 1, respectively.

4 Experiments

4.1 Experimental Settings

Implementation details. For the DTU dataset, we train CPT-VR for 10,000 iterations with a batch size of one image, targeting a Marching Cubes resolution of 256^3. In each training batch, we render one view at a resolution of 800 $\times$ 600 pixels. Note that, when we only utilize depth loss and color loss for supervision, we employ a higher Marching Cubes resolution of 512^3 and render images at 1600 $\times$ 1200 pixels. For BlendedMVS, CPT-VR is trained for 20,000 iterations with a batch size of one image and a Marching Cubes resolution of 256^3, rendering images at 768 $\times$ 576 pixels. Moreover, the initial learning rate of our model is set at 0.005, with a final epoch decay factor of 0.1 for DTU and 0.005 for BlendedMVS. More details about the architecture and the isosurface configuration are provided *in the supplementary material.*

Datasets. All experiments are conducted on DTU [9] and BlendedMVS [31]. Objects in DTU are collected in a controlled lab setting, with ground truth data that includes images, masks, and point clouds. For our training and evaluation, we leverage 15 objects identified by IDR with challenges such as specular highlights, delicate structures, and areas hidden from view. Meanwhile, BlendedMVS encompasses a diverse range of objects and scenes captured in natural settings. From this dataset, we select 7 difficult objects as NeuS [24] and the 'Bull' (with specular highlights) to further assess the model performance in reconstructing objects against complex backgrounds.

4.2 Quantitative Evaluation

Analysis on DTU. We benchmark our method against the state-of-the-art methods on the DTU dataset. DTU dataset contains 128 scans captured in the laboratory. In this work, we evaluate the surface mesh using the same set of scans as in [34, 37]. As denoted in Tab. 2, our method consistently outperforms other surface rendering methods on most scans. Note that, our method achieves the lowest value in mean *Chamfer Distance* (0.59), which is 16.9% lower than the second-best surface rendering method RegSDF (0.71). Since SDFnet in our method has a similar structure to it in NeuS2, we also compare NeuS2 and rencent volume rendering methods [26, 41] with CPT-VR. As demonstrated in Tab. 2, whether the two methods incorporate background information from the object mask (w/ M) or use our proposed one-point background model (w/ BG1) to attain background information, our method surpasses NeuS2 on the vast majority of ScanIDs.

Table 1: Comparison of the average rendering time on DTU. The rendered image resolution of all methods is 1600×1200, and the resolution of Marching Cubes is 512^3. The rendering time of our method is significantly lower than that of others.

Method	Rendering Time (s)
IDR	27.233
NeuS2 (w /M)	1.979
Ours (w/ M)	0.057
Ours (w/ BG1)	0.104

Table 2: Quantitative results on DTU dataset. Here, we evaluate all methods on *Chamfer Distance (CD)*. The lower the *CD* value, the better the performance. For a fair comparison, we also present the results of the volume rendering method NeuS2 since our SDFnet has a similar structure to it. "w/ M" means providing the prior background knowledge by masks, while "w/ BG1" means employing our one-point background model without any prior background knowledge. The best and second-best results are highlighted by different colors. The results illustrated that CPT-VR attains better performance among most ScanIDs.

ScanID	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
DMTet [19]	1.84	1.73	1.05	0.61	1.66	1.37	0.68	1.28	3.82	1.07	0.91	1.15	0.37	0.54	0.55	1.24
IDR [34]	1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.90	0.42	0.51	0.53	0.90
MVSDF [37]	0.83	1.76	0.88	0.44	1.11	0.90	0.75	1.26	1.02	1.35	0.87	0.84	0.34	0.47	0.46	0.89
FastMESH [39]	0.65	1.48	0.57	0.40	1.48	0.77	0.56	0.86	0.84	0.94	0.72	0.81	0.52	0.49	0.54	0.77
RegSDF [35]	0.59	1.41	0.63	0.42	1.34	0.62	0.59	0.89	0.91	1.02	0.60	0.59	0.29	0.40	0.38	0.71
HF-NeuS [26] (w/ M)	1.11	1.28	0.61	0.47	0.97	0.68	0.62	1.34	0.91	0.73	0.53	1.82	0.38	0.54	0.51	0.83
LOD-NeuS [41] (w/ M)	0.65	0.91	0.37	0.48	1.05	0.87	0.82	1.22	0.95	0.69	0.56	1.30	0.42	0.58	0.57	0.76
NeuS2 [25] (w/ M)	0.56	0.76	0.49	0.37	0.92	0.71	0.76	1.22	1.08	0.63	0.59	0.89	0.40	0.48	0.55	0.70
NeuS2 [25] (w/ BG1)	0.69	0.86	0.83	0.34	1.05	0.68	0.65	1.05	1.12	0.67	0.58	1.41	0.37	0.50	0.52	0.75
Ours (w/ M)	0.55	0.72	0.35	0.38	0.85	0.59	0.54	0.81	0.83	0.67	0.53	0.59	0.32	0.38	0.37	0.56
Ours (w/ BG1)	0.43	0.73	0.36	0.32	0.93	0.61	0.61	0.89	1.02	0.68	0.48	0.73	0.32	0.37	0.37	0.59

Moreover, we also measure the rendered results with *PSNR* to the ground truths. As indicated in Tab. 3, with the introduction of background information (BG1: ✗), our method achieves the highest *PSNR* values on most ScanIDs. Additionally, it can be observed that when using the one-point background model, both NeuS2 and CPT-VR are able to achieve similar or better performance compared to leveraging background information from masks (BG1: ✓).

Table 3: Quantitative comparison of 2D view synthesis on DTU dataset. We measure all methods based on the metric *PSNR*. A higher *PSNR* indicates better synthesis quality. Here, B1 represents whether to introduce extra background knowledge into the method. ✗ means to leverage the background information from masks, while ✓ indicates utilizing our one-point background model without introducing any background information. We color code the best and second-best results. Compared to NeuS2, our method achieves much better appearance reconstruction results.

ScanID	BG1	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
NeuS2	✗	28.44	27.14	29.70	29.67	31.75	27.83	24.84	31.24	26.86	30.57	26.05	28.93	28.98	27.82	32.48	28.82
NeuS2	✓	27.71	24.99	27.51	28.89	20.55	30.14	29.01	27.70	26.16	31.09	33.10	30.69	30.48	32.77	34.10	28.99
Ours	✗	28.15	27.31	28.54	29.52	34.04	29.83	29.00	35.92	29.32	33.08	29.45	31.16	30.77	30.50	38.38	31.00
Ours	✓	27.30	25.59	26.88	27.77	30.91	30.06	29.28	28.05	26.68	28.92	32.03	30.18	29.02	34.02	34.02	29.38

Table 4: Quantitative comparison on BlendedMVS dataset. We measure the method on the metric *Chamfer Distance (CD)*. The lower *CD* values indicate better performance. For a fair comparison, we adopt our one-point background model to NeuS2. Compared to NeuS2, CPT-VR attains better geometry reconstruction results among most of the categories.

Method	Bear	Clock	Dog	Durian	Man	Sculpture	Bull	Stone	Mean
NeuS2	0.38	0.47	0.95	0.47	0.46	0.40	0.65	0.38	0.52
Ours	0.35	0.45	0.68	0.48	0.46	0.39	0.45	0.38	0.45

Analysis on BlendedMVS. As illustrated in Tab. 4, we compare our method with NeuS2 on the eight selected categories with complex backgrounds. Since the scales are unknown in BlendedMVS, we scale the bounding box of each object to 100 units to measure their performance on *Chamfer Distance (CD)*. CPT-VR achieves the best performance in the categories of 'Bear', 'Clock', 'Dog', 'Sculpture', and 'Bull', with the *CD* value for 'Man' and 'Stone' being on par with NeuS2, and only 0.01 behind NeuS2 for 'Durian'. Moreover, CPT-VR outperforms NeuS2 by 14.8%. These results demonstrate the advantage of our method in reconstructing objects within complex environments.

Analysis of Method Efficiency. We evaluate the rendering time of IDR, NeuS2, and CPT-VR on DTU. The results in Tab. 1 suggest the rendering time of IDR (24.233s) and NeuS2 (1.97s) is lengthy. Compared with them, our method achieves real-time rendering under the cases without introducing background priors being about 19 times faster than NeuS2. This is primarily because NeuS2 utilizes the dense sampling method in the volume rendering process.

4.3 Qualitative Evaluation

Impact of Specular Highlights. Fig. 4 illustrates the qualitative results on the DTU dataset. We compare our CPT-VR(w/M) model with IDR [34], DMTet [19], MVSDF [37], and NeuS2 [25]. As shown by the close-up regions in Fig. 4, the specular regions constructed by other methods exhibit obvious indentations, while our method can construct these regions with high quality.

Impact of Complex Backgrounds. Fig. 5 demonstrates the visual results of reconstructing objects with complex backgrounds. Since existing surface render-

ing methods cannot handle the cases without any background prior, we specifically evaluate NeuS2 and our method on BlendedMVS. As the elliptical area demonstrated, our method can effectively model the geometry of foreground objects without introducing background information, particularly the details on the surface of objects. Moreover, the rendering results of our method closely resemble the input image.

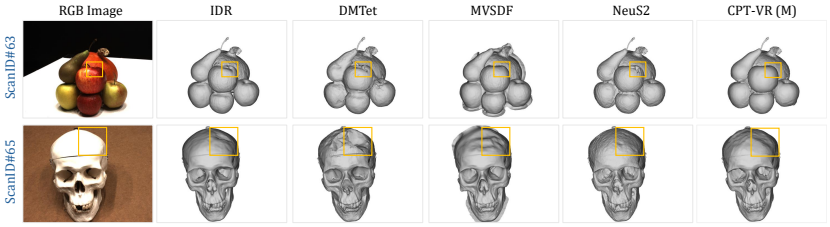

Fig. 4: Qualitative comparisons on objects in DTU with specular highlights. The close-ups in yellow highlight the specific regions for comparison. The geometry reconstruction results demonstrate that our method is robust against specular highlights.

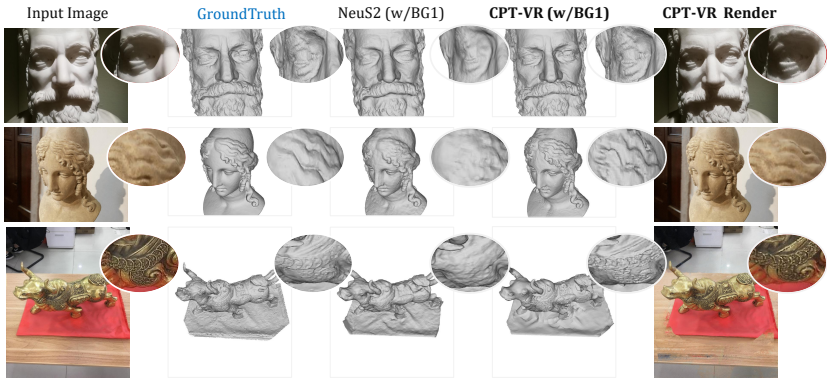

Fig. 5: Qualitative results of objects with complex background on BlendedMVS. We outline the close-up regions for detailed comparison. It is evident that without the mask guidance, our method also exhibits strong capabilities in modeling foreground objects, especially for the finer-grained surface structures.

Impact of the Sampling Point Amount. Tab. 5 suggests the impact of varying the number of sampling points and their distances. The optimal result is achieved when sampling one point at $2F_p$, and the CD is 0.66. Conversely, sampling either too closely (at F_p) or too distantly (at $1000F_p$) causes the sampled points to stray outside the desired background range, leading to suboptimal

results with CD values of 0.68 and 0.73, respectively. Additionally, increasing the number of sampling points in our approach tends to accentuate background details, thereby negatively impacting the accuracy of foreground object modeling. More background points would render the model to prioritize background fitting over the foreground, thus generating inferior results.

4.4 Ablation Study

As shown in Tab. 6, we set five settings to demonstrate the superiority of our method design. From the results of CPT-V-M, CPT-R-M, and CPT-VR-M, it is evident that introducing view-ray directions brings greater benefits to geometry reconstruction than introducing the two vectors alone. Meanwhile, the comparable *CD* values of CPT-VR-M and CPT-VR-BG1 demonstrate that the one-point background model can largely eliminate the negative impact of complex backgrounds in images on rendering the target object, enabling the model to no longer be dependent on any background knowledge. Moreover, the results of CPT-VR-BG1 and CPT-VR-BG1-D also verify that introducing depth supervision can further boost the reconstruction quality of our method.

Table 5: Impact of the Number of Background Points. Here, Num. indicates the sampling amount of points. Pos. is the sampling positions of points along the ray direction. F_p is defined in Sec. 3.4. All settings are evaluated on *Chamfer Distance (CD)*. Sampling one point at $2F_p$ attains the best performance.

Method	Num.	Pos.	*CD* ↓
CPT-VR	2	$2F_p$, $1000F_p$	0.74
CPT-VR	2	F_p, $2F_p$	0.67
CPT-VR	1	$1000F_p$	0.73
CPT-VR	1	F_p	0.68
CPT-VR	1	$2F_p$	0.66

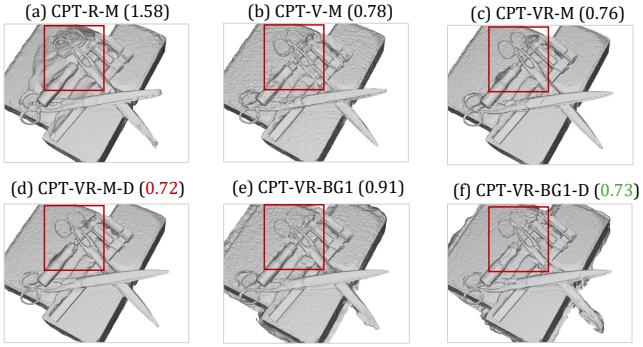

Fig. 6: Qualitative results of ablation studies on DTU. The close-ups in red highlight the specific regions for comparison. The content in the bracket is the Chamfer Distance, indicating the reconstruction quality. Incorporating the reflection vector enhances the ability of the model to simulate the appearance representation on specular regions. In addition, when utilizing the background model to attain the background information, introducing depth supervision greatly improves the reconstruction quality.

Table 6: Ablation study of our design choices. We evaluate all designs on *Chamfer Distance (CD)*. Results demonstrate the effectiveness of all our design choices and the importance of using accurate geometry representation when adding the reflection vector for appearance modeling.

Settings	Design Choices						*CD*↓
	View Vector	Reflection Vector	Background Model	$\mathcal{L}_C$	$\mathcal{L}_M$	$\mathcal{L}_D$	
CPT-V-M		✓		✓	✓		0.72
CPT-R-M	✓			✓	✓		0.83
CPT-VR-M	✓	✓		✓	✓		0.65
CPT-VR-BG1	✓	✓	✓	✓			0.66
CPT-VR-BG1-D	✓	✓	✓	✓		✓	0.59

As illustrated in Sec. 1, we argue that only when the surface sampling points are accurate can the introduction of the reflection vector achieve better appearance modeling of specular regions. To further demonstrate that, we incorporate the reflection vector into the surface rendering method DMTet [19] and IDR [34]. Tab. 7 indicates that introducing the reflection vector reduces the geometry reconstruction ability of DMTet (*CD* rises from 1.24 to 1.63) and IDR (*CD* rises from 0.90 to 1.68) Conversely, the sampling surface points of our method with CPT are more accurate. The accurate points facilitate the modeling of the reflection vector, avoiding a greater backpropagation error. Thus, incorporating the reflection vector into our method enables our model more robust against specular highlights. Fig. 6 also demonstrates the superiority of our design choices.

Table 7: Impact of view-reflection vectors. All methods are evaluated with masks on *Chamfer Distance (CD)*. With the reflection vector, the performance of our method has significantly improved, while DMTet and IDR become worse.

Method	IDR	IDR	DMTet	DMTet	Ours	Ours	Ours
View Vector (V)	✓	✓	✓	✓	✓		✓
Reflection Vector (R)		✓		✓		✓	✓
CD↓	0.90	1.68	1.24	1.63	0.72	0.83	0.65

5 Conclusions

In this paper, we present a novel surface rendering method **CPT-VR**. Based on the CPT algorithm, we correct the deviated points approximated by linear interpolation, enhancing the geometric accuracy. Meanwhile, benefiting from the accurate geometry representation, we incorporate view-reflection vectors into the appearance molding process, which enables our method against the specular highlights. Moreover, we devise a background model to enable our model to handle cases with complex backgrounds without any background knowledge. Experiments on the popular 3D reconstruction datasets demonstrate the superiority of our method in surface rendering, especially for cases with complex structures and specular highlights. We hope that our designs can provide some insights for future works.

References

1. Bangaru, S.P., Gharbi, M., Luan, F., Li, T.M., Sunkavalli, K., Hasan, M., Bi, S., Xu, Z., Bernstein, G., Durand, F.: Differentiable rendering of neural sdfs through reparameterization. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–9 (2022)
2. Darmon, F., Bascle, B., Devaux, J.C., Monasse, P., Aubry, M.: Improving neural implicit surfaces geometry with patch warping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6260–6269 (2022)
3. Fu, Q., Xu, Q., Ong, Y.S., Tao, W.: Geo-neus: geometry-consistent neural implicit surfaces learning for multi-view reconstruction. arXiv preprint arXiv:2205.15848 (2022)
4. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: International Conference on Machine Learning. pp. 3789–3799. PMLR (2020)
5. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2495–2504 (2020)
6. Guo, Y.C., Cao, Y.P., Wang, C., He, Y., Shan, Y., Qie, X., Zhang, S.H.: Vmesh: Hybrid volume-mesh representation for efficient view synthesis. arXiv preprint arXiv:2303.16184 (2023)
7. Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. The Visual Computer **12**(10), 527–545 (1996)
8. Hasselgren, J., Hofmann, N., Munkberg, J.: Shape, light, and material decomposition from images using monte carlo rendering and denoising. Advances in Neural Information Processing Systems **35**, 22856–22869 (2022)
9. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 406–413 (2014)
10. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics (TOG) **39**(6), 1–14 (2020)
11. Li, H., Yang, X., Zhai, H., Liu, Y., Bao, H., Zhang, G.: Vox-surf: Voxel-based implicit surface representation. IEEE Transactions on Visualization and Computer Graphics (2022)
12. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465 (2023)
13. Liu, H.T.D., Williams, F., Jacobson, A., Fidler, S., Litany, O.: Learning smooth neural functions via lipschitz regularization. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–13 (2022)
14. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
15. Ma, B., Zhou, J., Liu, Y.S., Han, Z.: Towards better gradient consistency for neural signed distance functions via level set alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17724–17734 (2023)
16. Mehta, I., Chandraker, M., Ramamoorthi, R.: A level set theory for neural implicit evolution under explicit flows. In: European Conference on Computer Vision. pp. 711–729. Springer (2022)

17. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I. pp. 405–421 (2020)
18. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting triangular 3d models, materials, and lighting from images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8280–8290 (2022)
19. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. Advances in Neural Information Processing Systems **34**, 6087–6101 (2021)
20. Sun, J., Chen, X., Wang, Q., Li, Z., Averbuch-Elor, H., Zhou, X., Snavely, N.: Neural 3d reconstruction in the wild. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–9 (2022)
21. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5481–5490. IEEE (2022)
22. Vicini, D., Speierer, S., Jakob, W.: Differentiable signed distance function rendering. ACM Transactions on Graphics (TOG) **41**(4), 1–18 (2022)
23. Walker, T., Mariotti, O., Vaxman, A., Bilen, H.: Explicit neural surfaces: Learning continuous geometry with deformation fields. arXiv preprint arXiv:2306.02956 (2023)
24. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. Advances in Neural Information Processing Systems **34**, 27171–27183 (2021)
25. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3295–3306 (2023)
26. Wang, Y., Skorokhodov, I., Wonka, P.: Hf-neus: Improved surface reconstruction using high-frequency details. Advances in Neural Information Processing Systems **35**, 1966–1978 (2022)
27. Wang, Y., Skorokhodov, I., Wonka, P.: Improved surface reconstruction using high-frequency details. arXiv preprint arXiv:2206.07850 (2022)
28. Wang, Y., Skorokhodov, I., Wonka, P.: Pet-neus: Positional encoding tri-planes for neural surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12598–12607 (2023)
29. Worchel, M., Diaz, R., Hu, W., Schreer, O., Feldmann, I., Eisert, P.: Multi-view mesh reconstruction with neural deferred shading. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6187–6197 (2022)
30. Wu, T., Wang, J., Pan, X., Xu, X., Theobalt, C., Liu, Z., Lin, D.: Voxurf: Voxel-based efficient and accurate neural surface reconstruction. arXiv preprint arXiv:2208.12697 (2022)
31. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1790–1799 (2020)
32. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems **34**, 4805–4815 (2021)

33. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B.: Bakedsdf: Meshing neural sdfs for real-time view synthesis. arXiv preprint arXiv:2302.14859 (2023)
34. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems **33**, 2492–2502 (2020)
35. Zhang, J., Yao, Y., Li, S., Fang, T., McKinnon, D., Tsin, Y., Quan, L.: Critical regularizations for neural surface reconstruction in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6270–6279 (2022)
36. Zhang, J., Yao, Y., Li, S., Luo, Z., Fang, T.: Visibility-aware multi-view stereo network. British Machine Vision Conference (BMVC) (2020)
37. Zhang, J., Yao, Y., Quan, L.: Learning signed distance field for multi-view surface reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6525–6534 (2021)
38. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)
39. Zhang, Y., Zhu, J., Lin, L.: Fastmesh: Fast surface reconstruction by hexagonal mesh-based neural rendering. arXiv preprint arXiv:2305.17858 (2023)
40. Zhang, Y., Hu, Z., Wu, H., Zhao, M., Li, L., Zou, Z., Fan, C.: Towards unbiased volume rendering of neural implicit surfaces with geometry priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4359–4368 (2023)
41. Zhuang, Y., Zhang, Q., Feng, Y., Zhu, H., Yao, Y., Li, X., Cao, Y.P., Shan, Y., Cao, X.: Anti-aliased neural implicit surfaces with encoding level of detail. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023)