



Refractive Neural Rendering of Underwater Scenes via Complex Refractive Surface Reconstruction

Xiaoqiang Zhang¹, Zhixin Zhang¹, Lingyan Ran², and Xinmin Li³(✉)

¹ School of Information and Control Engineering, Southwest University of Science and Technology, Mianyang 621010, Sichuan, China

² School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, Shaanxi, China

³ College of Computer Science, Chengdu University, Chengdu 610106, Sichuan, China

lixinmin@cdu.edu.cn

Abstract. The vanilla NeRF and its subsequent variants usually assume that the path of the scene ray is a straight line. For scenes with refractive media, *e.g.*, underwater scenes, the refracted scene rays would violate the assumption and bring geometrically incorrect modeling of the rendering procedure, resulting in low rendering quality. In particular, for underwater scenes with complicated refractive surfaces, an accurate ray-by-ray estimation of the refractive surface normal would be crucial for rendering the refracted rays. To address the limitations in current underwater neural rendering techniques, in this paper, we propose to realize high-quality neural rendering of scenes with refractive media while accurately reconstructing the complex refractive surface. Firstly, to accurately model the scene refraction, a multilayer perceptron is utilized to estimate the distance from the camera center to the refractive surface, which is used to fit a Bezier surface to achieve more accurate normal vector estimation. Secondly, a learnable parameter is designed to control the ratio of coarse-to-fine sampling and achieve adaptive sampling of the refracted ray. Furthermore, to enhance underwater low-frequency view-dependent information, the spherical harmonics is adopted for directional position encoding of the refracted ray. Extensive experimental results on public and self-built datasets demonstrate the rendering quality and surface reconstruction accuracy in various refractive scenes with complex refractive surfaces.

Keywords: Neural rendering · Refraction · Adaptive sampling · Spherical harmonics encoding

1 Introduction

The recent advent of neural rendering and neural radiance fields (NeRF) [9] has attracted much attention in the field of graphics and vision. The NeRF and

its subsequent variants usually leverage multi-view images [15] and can achieve excellent rendering quality for novel view synthesis, while implicitly reconstructing 3D scenes based on a small number of sparse input images. In vanilla NeRF and most of its variants [2, 3, 10], it is usually assumed that there is only one refractive medium in the scene, *e.g.*, the air, which restricts the application scenes for neural rendering techniques. For scenes with multiple refractive media, such as underwater scenes, the path of the ray from the scene object to the camera would be refracted, which violates the basic assumptions in NeRF.

Several NeRF-based methods [6, 12, 17, 19] have attempted to address the above problem in underwater scene rendering via specialized designs. LB-NeRF [6] and D-NeRF [12] designs a deformation layer or deformation field, which can account for the non-Lambertian scenes. However, it is difficult to model the ray-by-ray refraction deformation. NeReF [17] reconstructs both the radiance field and the corresponding density field, accumulating refraction points and their surface normals along each ray during queries. The water-free images of the same scene are needed for model training, which restricts the application of the algorithm. NeRFrac [19] employs a multilayer perceptron (MLP) to approximate the ray-to-surface distances, and computes approximate surface normals using neighboring surface points, with the assumption that the local neighboring surface are flat. For complex refractive surfaces, the flat assumption may lead to an estimation error.

Inspired by [19], in this paper, a refractive neural rendering method is proposed based on accurate reconstruction of complex refractive surfaces. We assume that there are two refractive media in the scene, which is typical in underwater imaging scenarios. A particular ray in such a scene could be divide into two segments, *i.e.*, ray segment in the air **Ray_{air}** and ray segment in the water **Ray_{water}**. The **Ray_{air}** is implicitly utilized to reconstruct the complex refractive surface. In particular, an MLP is designed to model the distance that **Ray_{air}** travels from the ray origin to the refractive surface. A Bezier surface is fitted locally to estimate the surface normal, which would yield more accurate normal compared to flat-plane or polynomial approximations. For **Ray_{water}**, we consider the underwater scene object distributions when sampling, *i.e.*, simple scenes require fewer fine samples, while complex scenes necessitate more. A learnable parameter that dynamically adjusts the ratio between coarse and fine sampling points is designed to adapt to diverse scene requirements. Finally, to enhance the ability to learn underwater low-frequency view-dependent color variations, spherical harmonic (SH) positional encoding is applied to the directional information on the sampled points on **Ray_{water}**. Quantitative evaluations of the rendering quality and refractive surface reconstruction quality on public [19] and self-build synthetic dataset demonstrate that the proposed approach outperforms the relevant baseline approaches. In summary, the main contributions of this paper are as follows.

- A ray-by-ray refractive surface reconstruction method that employ Bezier surface fitting to solve for high-precision normals on complex refractive surfaces;

- An adaptive coarse-to-fine sampling ratio control strategy that dynamically adjusts the sampling distribution based on scene complexity;
- A hybrid encoding scheme combining SH and sinusoidal positional encoding, preserving high-frequency signal modeling while improving low-frequency signal fitting.

2 Related Works

Neural Rendering is an important application of the combination of machine learning and computer graphics, of which NeRF [9] is the pioneering work. It designs an end-to-end model that uses the MLP to represent static scenes to achieve novel view synthesis. The MLP uses the coordinates and directions of 3D points in space as the input to predict the volume density and radiance values of these points, which are used for rendering specified view via volume rendering. The variants of NeRF would focus the the improvement of rendering quality and efficiency. Mip-NeRF [2] uses a conical interface instead of original ray for sampling, achieving the effects of deblurring and anti-aliasing, while the application scenarios is expanded in Mip-NeRF360 [3]. DS-NeRF [5] utilizes the sparse pointclouds to supervise NeRF training and can achieve high-quality novel view synthesis even with a small amount of input images. Instant-NGP [10] divides the space into multi-resolution grids. It only learns the information of grid vertices. Information of other positions are obtained from linear interpolation, therefore reducing the network size and improving the training efficiency. Tensorf [4] represents the radiance field with a four-dimensional tensor and decomposes it into multiple low-rank components, simultaneously improving rendering quality and efficiency. IBRNet [14] constructs the viewpoint interpolation function and builds the core network framework through ray transformation and MLP, which is capable of synthesizing complex new views.

Novel View Synthesis of Underwater Scenes. NeRF and most of its improved algorithms are based on the assumption that ray does not deflect, but some optical phenomena do not conform to this assumption, the most typical of which is the refraction of ray. The goal of WaterNeRF [13] is to render underwater scenes, but the camera is placed in water. NeReF [17] proposed a novel neural refractive field, which can simultaneously estimate the surface position and normal vector direction of transparent fluids and effectively reconstruct the refracted surface. However, training data of scenes with/without water is needed for loss calculation. Nerfies [11] models the deformation field through MLPs and maps the 3D points in the observation coordinate system to the canonical coordinate system of the standard NeRF model. D-NeRF also uses deformation fields to learn about displacements in real scenarios. Bai *et al.* [1] adopt refractive calibration approach of Zhang *et al.*'s [21] method, and propose to render the underwater scene with the assumption of flat refractive geometry. NeRFrac [19] assumes a curved refractive surface, and uses an MLP to model the refractive field, which is the prediction of the distance from the ray to the refractive surface. The refracted ray is then calculated and used for NeRF training and rendering.

3 Preliminaries

3.1 Refractive Geometry

Suppose that a light ray travels through two refractive media, with the refractive index of incident medium of λ_1 and that of the refracted medium of λ_2 , the Snell's law describes the relation between the incident angle θ_1 and the refracted angle θ_2 :

$$\lambda_1 \sin \theta_1 = \lambda_2 \sin \theta_2. \quad (1)$$

Let $\mathbf{I} \in \mathbb{R}^3$ be the directional vector of the incident ray, $\mathbf{N} \in \mathbb{R}^3$ be that of the refractive surface normal at the position where the refraction occurs, the incident angle θ_1 can be calculated via $\theta_1 = \arccos \frac{\mathbf{I} \cdot \mathbf{N}}{\|\mathbf{I}\| \|\mathbf{N}\|}$, the refracted ray $\mathbf{T} \in \mathbb{R}^3$ is then obtained by

$$\mathbf{T} = \frac{\lambda_1}{\lambda_2} \mathbf{I} + \left(\frac{\lambda_1}{\lambda_2} \cos \theta_1 - \sqrt{1 - \frac{\lambda_1^2}{\lambda_2^2} \sin^2 \theta_1} \right) \mathbf{N}, \quad (2)$$

where all vectors are expressed in the camera coordinate system. Throughout the paper, we assume that the camera is in the air, with a refractive index of 1.00, and the scene object is in the water, with a refractive index of 1.33.

3.2 Volume Rendering

NeRF implicitly represents a scene via an MLP. First, the origin $\mathbf{o}$ and direction $\mathbf{w}$ of a ray are determined based on the intrinsic and extrinsic parameters of the camera. Then, sample along the direction of the ray $\mathbf{w}$. The position coordinates $\mathbf{x}$ and direction vectors (θ, ϕ) of the sampled 3D points are inputted into the neural network to obtain the corresponding volume density σ and radiance value $\mathbf{c} = (r, g, b)$. Finally, render to obtain the result of the pixels corresponding to the rays via

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t), \mathbf{w}) dt, \quad (3)$$

where t_n and t_f represent the near clipping plane and the far clipping plane, respectively. $T(t)$ is the cumulative transparency, *i.e.*, the probability that the ray does not collide with other particles from t_n to t . $T(t)$ is given by: $T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s)) ds\right)$. The photometric loss in Eq. (4) is utilized to optimize the network parameters.

$$\mathcal{L} = \sum_{\mathbf{r} \in R} \|C(\mathbf{r}) - C_{gt}(\mathbf{r})\|_2^2, \quad (4)$$

where R are all rays in the batch and $C_{gt}(\mathbf{r})$ is the ground-truth color.

4 Proposed Method

In this section, we elaborate on the main pipeline and the specific steps of the proposed approach. The objective is to achieve high-quality neural rendering of scenes with refractive media, while reconstructing the complex refractive surfaces. As illustrated in Fig. 1, we assume a scene with two refractive media, *e.g.*, air and water. An array of cameras is placed on the water, with a complex or wavy refractive surface. The algorithm pipeline comprises 3 key components: (1) **Ray-by-ray refractive surface reconstruction**. It takes the origin and direction of the ray in the air Ray_{air} as input. An MLP is adopted to predict the distance from the ray origin to the refractive surface. Based on the estimated distance, a Bezier surface is then fitted the local region to derive the refractive normal vector. (2) **Adaptive hierarchical sampling of the refracted ray**, in which a learnable parameter controls the ratio between coarse and all sampling, enabling adaptive sampling of refracted rays $\text{Ray}_{\text{water}}$. (3) **Directional encoding with spherical harmonics**. For the direction vectors of sampled points, low-frequency components are encoded using 2nd-order spherical harmonics, which helps to learn low-frequency view-dependent color variations. The positional encoded 3D point coordinates and direction vectors are fed into a neural radiance field MLP to predict volume density and color values, which are later used for rendering. The rendered color is to calculate $\mathcal{L}$ in Eq. (4), which is used to optimize the two mentioned MLP.

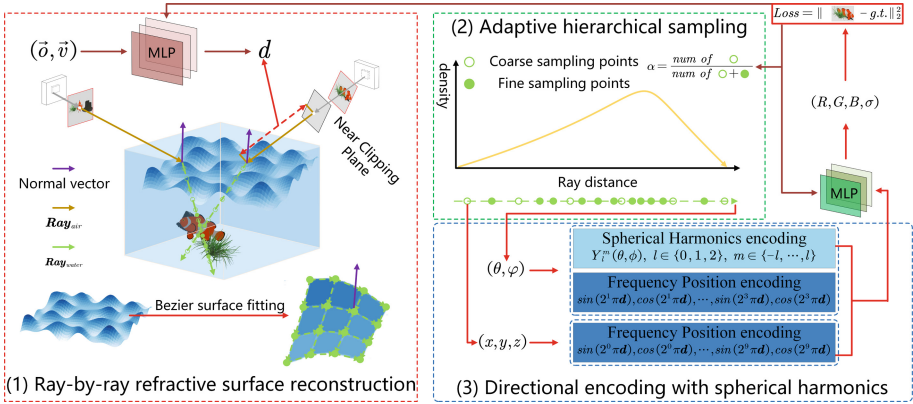


Fig. 1. The pipeline of the proposed approach. The main modules of the method include: (1) Ray-by-ray refractive surface reconstruction (2) Adaptive hierarchical sampling of the refracted ray and (3) Directional encoding of the sampled points with spherical harmonics. All parameters are learned via minimizing the photometric loss.

4.1 Ray-by-Ray Refractive Surface Reconstruction

As shown in Eq. (2), the surface normal plays an important role in computing the direction of the refracted ray. In addition, to fully model the refraction,

the position where the refraction occurs is also needed. For a complex or wavy refractive surface, the two parameters are difficult to model globally. Therefore, in this subsection, we would estimate the two parameters in a ray-by-ray fashion.

For each incident ray in the air $\mathbf{Ray}_{air}$, the origin $\mathbf{o}_c$ and the direction $\mathbf{v}_c$ can be determined via the calibrated camera intrinsic and extrinsic parameters. The ray can be expressed parametrically as $\mathbf{r}_c(t) = \mathbf{o}_c + t\mathbf{v}_c$, where t is the parameter. The subscript c means that vectors are expressed in the average camera coordinate system.

Following [9, 19], we then transform the ray coordinates to the normalized device coordinates (NDC) system, with the transformed ray origins $\mathbf{o}$ and direction vector of $\mathbf{v}$. For each ray, an MLP is designed to predict the distances d of how long the ray travels to reach the refractive surface. The origin $\mathbf{o}$ and direction $\mathbf{v}$ are first encoded using positional encoding and then fed into an 8-layer MLP to predict d . Applying the ray parametric equation, the position of the refraction point $\mathbf{P}_R$ can be computed as:

$$\mathbf{P}_R = \mathbf{o} + d\mathbf{v}. \quad (5)$$

By obtaining the position of the refraction point $\mathbf{P}_R$ of the entire refractive surface, a Bicubic Bezier surface fitting is adopted to fit the local surface. The surface defined by a grid of $4 \times 4 = 16$ control points, which are obtained using a local neighborhood of 4-by-4 refraction points:

$$\mathbf{S}(m, n) = \sum_{i=0}^3 \sum_{j=0}^3 \mathbf{P}_{i,j} B_i^3(m) B_j^3(n) \quad m, n \in [0, 1] \quad (6)$$

where $\mathbf{P}_{i,j}$ denotes the control point grid, (m, n) represents the parametric coordinates on the surface, used to locate points on the surface in parametric space, and $B_i^3(m)$ and $B_j^3(n)$ are the cubic Bernstein basis functions.

After fitting the local surface, the normal vector can be derived from the cross product of the partial derivatives along the two parametric directions:

$$\mathbf{N}(m, n) = \frac{\partial \mathbf{S}}{\partial m} \otimes \frac{\partial \mathbf{S}}{\partial n}, \quad (7)$$

where the partial derivatives are computed as follows:

$$\begin{cases} \frac{\partial \mathbf{S}}{\partial m} = \sum_{i=0}^3 \sum_{j=0}^3 \mathbf{P}_{i,j} \frac{dB_i^3(m)}{dm} B_j^3(n), \\ \frac{\partial \mathbf{S}}{\partial n} = \sum_{i=0}^3 \sum_{j=0}^3 \mathbf{P}_{i,j} B_i^3(m) \frac{dB_j^3(n)}{dn}. \end{cases} \quad (8)$$

With the fitted refractive surface, *i.e.*, the distance d and the normal vector, Snell's law can be applied to model the refracted ray. It should be noted that Eq. (2) is expressed in the camera coordinate system. Because the transformation from the camera coordinate system to the NDC system is non-Euclidean, it is invalid to apply Eq. (2) in NDC system. To address this, we first transform $\mathbf{v}$ and $\mathbf{P}_R$ from NDC system back to the average camera coordinate system. In the average camera coordinate system, we use the Bezier surface fitting to estimate the predicted normal vector, then calculate the refracted ray direction $\mathbf{v}'_c$. Finally, we transform $\mathbf{v}'_c$ into NDC system, denoted as $\mathbf{v}'$.

4.2 Adaptive Hierarchical Sampling of the Refracted Ray

In scenes with refractive media, based on $\mathbf{P}_R$, the refracted ray $\mathbf{r}'$ can be expressed as

$$\mathbf{r}'(t) = \mathbf{P}_R + t\mathbf{v}', \quad (9)$$

where $\mathbf{v}'$ denotes the direction of the refracted ray, and t a scalar is a parameter.

Then, sampling is performed along the refracted ray. Conventional sampling approaches usually are applied in a coarse-to-fine fashion, with a fixed number of 128 sampled points in the coarse stage, and that of 64 points in the fine stage. The fixed number may not be adapted to different scene requirements for sample point distribution.

To address this, we introduce an optimizable parameter α , initialized to 0.5. In practical applications, the number of coarse sampling points N_c is calculated as $N_c = \lfloor \alpha N_{points} \rfloor$, where N_{points} represents the total number of sampling points, and $\lfloor \cdot \rfloor$ indicates rounding down to the nearest integer. Consequently, the number of fine sampling points N_f is given by $N_f = N_{points} - N_c$. α influences the point allocation between coarse and fine sampling during the forward pass and is updated based on the loss $\mathcal{L}$ in Equation (4) during the backward propagation phase.

4.3 Positional Encoding Strategy

Before feeding the position coordinates and the direction vectors of sampled points on the refracted ray into the NeRF MLP to predict scene point radiance and density, a positional encoding is usually applied. For input $\mathbf{x} \in \mathbb{R}^D$, conventional sinusoidal positional encoding is defined as:

$$\gamma(\mathbf{x}) = [\sin(2^0\pi\mathbf{x}), \cos(2^0\pi\mathbf{x}), \dots, \sin(2^{L-1}\pi\mathbf{x}), \cos(2^{L-1}\pi\mathbf{x})], \quad (10)$$

where $L = 10$ for position coordinates and $L = 4$ for direction vectors.

Considering the importance of ray directions of the refracted rays in the refractive scene, in our approach, an optimized encoding is applied for the direction vectors. For the **low-frequency components**, a 2nd-order Spherical-Harmonics (SH) basis is adopted, which encodes low-frequency directional information. SH demonstrate superior fitting capability for low-frequency signals, enabling more accurate representation of smooth variations in illumination and color [18]. The 2nd-order SH comprises 9 basis functions, which are defined as:

$$Y_l^m(\theta, \phi), \quad l \in \{0, 1, 2\}, \quad m \in \{-l, \dots, l\}, \quad (11)$$

where θ and ϕ represent the polar and azimuthal angles of the directional vector, respectively. For the **high-frequency components**, The sinusoidal encoding is preserved but with adjusted frequency bands (1,2,3), resulting in an 18-dimensional high-frequency encoding,

$$\gamma(\mathbf{d}) = [SH_2(\mathbf{d}), \sin(2^1\pi\mathbf{d}), \cos(2^1\pi\mathbf{d}), \dots, \sin(2^3\pi\mathbf{d}), \cos(2^3\pi\mathbf{d})]. \quad (12)$$

The positional encoded sample points are finally fed into a NeRF MLP to predict the volume density and radiance value, with volume rendering in Eq. (3), the photometric loss $\mathcal{L}$ in Eq. (4) is minimized to optimize the MLP in Sect. 4.1, α in Sect. 4.2 and the NeRF MLP. The Adam optimizer with a initial learning rate of 5e-4, moment parameters of 0.9 and 0.999, is employed for parameter updates. The learning rate decays exponentially, reducing to 10% at 250,000 steps.

5 Experimental Results

This section outlines the experimental results showcasing our approach’s performance. Initially, we quantitatively assess rendering performance using the public dataset from [19]. Next, we evaluate the accuracy of estimated refractive surfaces—including surface normals and coordinates—on a synthetic dataset developed using the Taichi programming toolkit [7]. Lastly, we present an ablation study to affirm the importance of our design choices.

5.1 Evaluation Metrics

The main goal of the proposed approach is twofold, *i.e.*, rendering high-quality novel views in multiple refractive media scenes, while reconstructing the refractive surfaces. Therefore, the approach is quantitatively evaluated in the two related aspects mentioned above.

Rendering quality evaluation metrics include the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [16], Learned Perceptual Image Patch Similarity (LPIPS) [20] and the Average [2] of the above three metrics, which are typical metrics for rendering.

Refractive surface reconstruction evaluation metrics includes refractive surface coordinate error S_{error} and surface normal error θ_{error} . For S_{error} , we use the RMSE error between the predicted surface coordinates at position (i, j) , $\mathbf{P}_{pred}(i, j)$, and the ground-truth surface coordinates $\mathbf{P}_{gt}(i, j)$:

$$S_{error} = \sqrt{\frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \|\mathbf{P}_{pred}(i, j) - \mathbf{P}_{real}(i, j)\|^2}, \quad (13)$$

where $H \times W$ represents the image resolution. The surface normal error θ_{error} represents the average of the angular differences between the predicted surface normal and the ground-truth: θ_{error} is defined as follows:

$$\theta_{error} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \arccos \frac{\mathbf{n}_{pred}(i, j) \cdot \mathbf{n}_{gt}(i, j)}{\|\mathbf{n}_{pred}(i, j)\| \cdot \|\mathbf{n}_{gt}(i, j)\|}, \quad (14)$$

where $\mathbf{n}_{pred}(i, j)$ and $\mathbf{n}_{gt}(i, j)$ are the predicted and the ground truth surface normal at position (i, j) , respectively.

5.2 Quantitative and Qualitative Analysis on Public Dataset

To evaluate the performance of the proposed method, we choose to utilize the dataset from [19], which contains two synthetic data, namely “hybrid” and “primary sine”, and two real data, namely “redflower” and “tree”. The dataset assumes a forward-facing camera setup similar to that of [8], with a camera number of 3×3 . Following [19], the 8 side views are used for training, while the center view is used for testing. It should be noted that in [19], the ground truths of the refractive surface are not provided. Hence, we only evaluate the rendering quality.

Table 1. Quantitative results on public datasets [19]. $\uparrow$ indicates higher metric is better, $\downarrow$ indicates lower metric is better. The best performance is in bold.

Method	Real - redflower				Real - tree			
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$
NeRF [9]	23.781	0.710	0.415	0.098	28.253	0.727	0.484	0.072
Mip-NeRF [2]	24.430	0.732	0.401	0.091	28.765	0.753	0.469	0.068
DS-NeRF [5]	24.238	0.741	0.373	0.089	28.688	0.762	0.442	0.066
TensorF [4]	24.412	0.769	0.336	0.084	28.209	0.758	0.360	0.064
NeRFrac [19]	25.918	0.837	0.239	0.063	30.385	0.834	0.315	0.049
Ours	26.298	0.851	0.232	0.059	30.492	0.842	0.296	0.047
Method	Synthetic - Hybrid				Synthetic - primary sine			
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$
NeRF [9]	20.514	0.691	0.386	0.124	21.012	0.721	0.360	0.115
Mip-NeRF [2]	20.423	0.672	0.404	0.128	21.073	0.703	0.378	0.117
DS-NeRF [5]	21.742	0.710	0.429	0.116	22.486	0.735	0.408	0.106
TensorF [4]	20.642	0.608	0.442	0.134	21.248	0.649	0.421	0.123
NeRFrac [19]	32.581	0.935	0.159	0.028	33.517	0.941	0.150	0.025
Ours	33.363	0.938	0.154	0.026	34.069	0.942	0.149	0.024

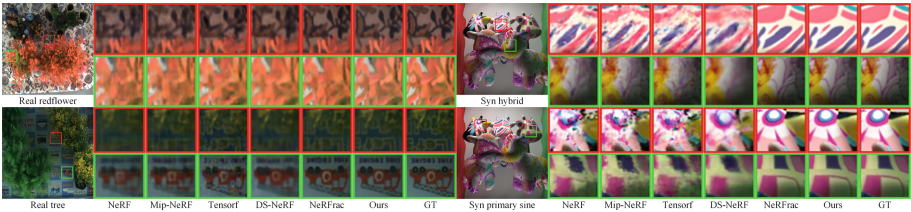


Fig. 2. Qualitative analysis on public datasets. The left image displays the original data with two annotated regions of interest (RoI), while the right side compares the rendering details of different methods within the RoI alongside the corresponding ground truth.

To demonstrate the performance of the proposed approach, the rendering quality of different methods is quantitatively compared, including the vanilla

NeRF [9] and its variants [2] focusing on rendering quality, those focusing on rendering quality and efficiency [4, 5] and NeRFrac [19]. Table 1 shows the quantitative comparisons among all approaches, while the bold text represents the best metrics. It can be seen that the proposed method achieves the best performance on all 4 metrics among all 4 data.

To further demonstrate the algorithm performance, the qualitative analysis of the rendering quality is provided in Fig. 2 of all 4 data. The highlighted details of the rendering result of different methods show the quality of different approaches with respect to the ground truths.

5.3 Quantitative and Qualitative Analysis on Synthetic Dataset

To further evaluate the quality of refractive surface reconstruction, along with rendering quality, a synthetic dataset is generated via Taichi [7]. Similar to those of the public dataset [19], the synthetic data is also rendered using a 3×3 camera arrangement. To verify the performance with different refractive surfaces, 4 distinct and complicated refractive surface geometries are utilized in the experiments, namely surface A (Radial Wave), surface B (Oblique Sine Wave), surface C (Gauss Packet) and surface D (Gauss Modulated). The equations for the above surfaces are given below,

$$A : z = z_0 + a \sin \left(w \sqrt{(x - x_0)^2 + (y - y_0)^2} \right), \quad (15)$$

$$B : z = z_0 + a \sin (w (x + y)), \quad (16)$$

$$C : z = z_0 + a \sin \left(w \sqrt{(x - x_0)^2 + (y - y_0)^2} \right) \cdot e^{-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}}, \quad (17)$$

$$D : z = z_0 + a \sin (w (x + y)) \cdot e^{-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}}, \quad (18)$$

where a denotes the wave amplitude, w the frequency, and (x_0, y_0) the wave center.

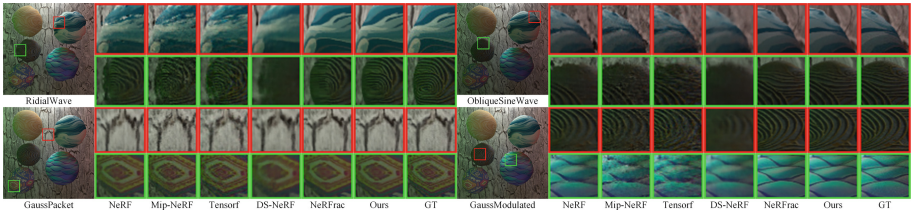


Fig. 3. Results on synthetic datasets. The results on synthetic datasets are demonstrated. Both NeRFrac and our method mitigate artifacts and blurring in novel-view rendering by learning refraction-induced deformations. Compared to NeRFrac, our approach achieves superior detail preservation. For instance, on the RadialWave dataset, our method reconstructs finer details (e.g., the brown sphere) with higher fidelity.

Quantitative and qualitative evaluations of rendering quality are on the above synthetic datasets are given in Table 2 and Fig. 3, respectively. From Table 2, it can be seen that the proposed approach achieves the best performances in terms of the 4 metrics under 4 different surfaces. Specifically, compared to NeRFrac [19], our approach demonstrates an average improvement of 2.907% in PSNR, 2.169% in SSIM, and an average reduction of 8.551% in LPIPS and 11.798% in Average. Figure 3 illustrates the rendering details under 4 different refractive surfaces.

Table 2. Quantitative evaluations of rendering quality on synthetic datasets. The proposed method achieves superior performance on visual metrics compared to other baseline approaches.

Method	RadialWave				ObliqueSineWave			
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$
NeRF [9]	22.624	0.460	0.443	0.121	23.908	0.541	0.404	0.104
Mip-NeRF [2]	22.105	0.397	0.464	0.130	23.742	0.489	0.417	0.108
DS-NeRF [5]	23.504	0.471	0.494	0.117	25.498	0.554	0.458	0.095
TensorF [4]	23.046	0.430	0.451	0.119	24.898	0.530	0.408	0.097
NeRFrac [19]	29.176	0.799	0.243	0.051	31.708	0.863	0.170	0.035
Ours	29.423	0.813	0.226	0.048	33.835	0.901	0.147	0.027
Method	GaussPacket				GaussModulated			
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$
NeRF [9]	27.808	0.758	0.192	0.054	24.201	0.606	0.300	0.089
Mip-NeRF [2]	26.669	0.701	0.201	0.062	23.564	0.525	0.318	0.099
DS-NeRF [5]	26.721	0.669	0.301	0.072	25.264	0.574	0.380	0.090
TensorF [4]	26.790	0.687	0.206	0.062	24.480	0.549	0.322	0.092
NeRFrac [19]	32.757	0.894	0.138	0.029	30.992	0.859	0.171	0.037
Ours	33.463	0.903	0.128	0.026	31.587	0.872	0.160	0.034

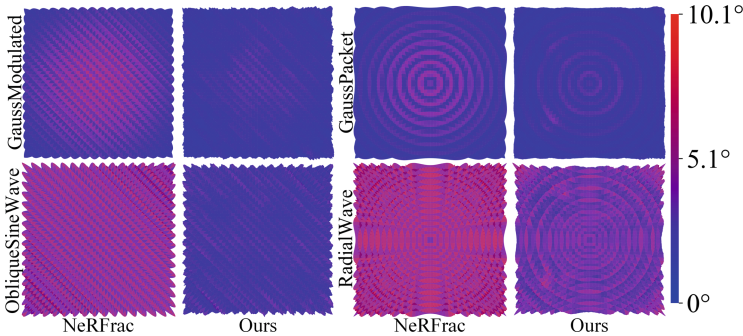


Fig. 4. Visualization results of normal error comparisons on the synthetic datasets for different refractive surfaces. For all 4 complicated surfaces, the proposed approach would estimate more accurate surface normals.

Table 3. Quantitative evaluations for refractive surface reconstruction. Both methods yield comparable refractive surface coordinate errors. For the surface normal error, the proposed method achieves higher precision in both mean error and the standard deviation.

Method	GaussModulated		GaussPacket	
	$S_{error} \downarrow$	$\theta_{error}(^{\circ}) \downarrow$	$S_{error} \downarrow$	$\theta_{error}(^{\circ}) \downarrow$
NeRFrac [19]	0.0026 $\pm$ 0.0011	2.8842 $\pm$ 1.7135	0.0032 $\pm$ 0.0011	2.0366 $\pm$ 1.2246
Ours	0.0028 $\pm$ 0.0011	1.4674 $\pm$ 0.8307	0.0028 $\pm$ 0.0009	1.2612 $\pm$ 0.7650
Method	ObliqueSineWave		RadialWave	
	$S_{error} \downarrow$	$\theta_{error}(^{\circ}) \downarrow$	$S_{error} \downarrow$	$\theta_{error}(^{\circ}) \downarrow$
NeRFrac [19]	0.0020 $\pm$ 0.0010	4.6525 $\pm$ 2.2430	0.0025 $\pm$ 0.0011	5.5071 $\pm$ 2.6597
Ours	0.0019 $\pm$ 0.0009	2.0488 $\pm$ 0.9845	0.0026 $\pm$ 0.0011	3.4824 $\pm$ 1.7315

Furthermore, we evaluate the surface reconstruction performance using S_{error} and θ_{error} . The quantitative evaluations and comparisons with [19] are given in Table 3. For both metrics, in addition to the mean value of S_{error} and θ_{error} , the standard deviation of errors of all rays is also listed. For S_{error} , since both NeRFrac [19] and the proposed method employ similar surface coordinate reconstruction approaches, the metrics for surface coordinate S_{error} are similar. However, for θ_{error} , it can be seen that the proposed method outperforms [19] in terms of mean and standard deviation, reflecting the advantages of our approach. It is noteworthy that, for scenes with refractive media, the accuracy of normal direction is crucial when computing the refraction rays. To further show the accuracy of the estimated normal, the normal errors for 4 different refractive surfaces of each pixel are shown in Fig. 4. It can be seen that a more accurate and robust result can be obtained from the proposed method.

During the experiments, we found that the convergence values of α would differ for scenes of different refractive surfaces. For scenes with complex surfaces, *e.g.*, “GaussPacket” and “GaussModulated”, the corresponding α is 0.611 and 0.613, respectively. Compare with the datasets used in Sect. 5.2, “real tree” and “real redflower”, the surfaces are simple, α would converge to 0.773 and 0.774, respectively. By allowing α to converge to different values, our approach balances the distinct sampling demands posed by various types of refractive surfaces.

5.4 Ablation Study

In the ablation study, we investigate the necessity of several key designs in the proposed approach and demonstrate their importance in achieving better visual rendering quality. We focus on three components: (1) surface fitting (SF), (2) adaptive sampling (AS), and (3) directional encoding (DE). In the ablation study, the vanilla NeRFrac [19] is adopted as the baseline approach. The experiments were conducted on the Oblique Sine Wave dataset to validate the contribution of each module.

Table 4. Quantitative results of Ablation study. In the table, the baseline model is from [19]. SF stands for surface fitting, AS stands for adaptive sampling, and DE stands for directional encoding, which are the three important modules in the proposed method.

Model	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Average $\downarrow$
Baseline	31.708	0.863	0.170	0.035
Baseline + SF	33.192	0.886	0.164	0.030
Baseline + SF + AS	33.415	0.895	0.156	0.028
Baseline + SF + AS + DE	33.835	0.901	0.147	0.027

Table 4 gives the quantitative results. By adding the SF module for more accurate normal vector estimations, there is a 4.680% improvement in PSNR, a 2.665% increase in SSIM, a 3.529% reduction in LPIPS, and a 14.286% decrease in the Average metric. Further adoption of the AS module would make the model focusing on high-complexity regions, and the metrics are further optimized, with the PSNR improved by 0.672%, SSIM increased by 1.016%, LPIPS decreased by 4.878%, and the Average metric dropped by 6.667%. Finally, the design for spherical harmonic (SH) encoding for “low-frequency component” would enhance directional feature representation. The metrics continue to improve, with the PSNR increased by 1.257%, SSIM rose by 0.670%, LPIPS decreased by 5.769%, and the Average metric reduced by 3.571%. For the time cost in iteration, the baseline takes 47.67 ms while the proposed method takes 45.56 ms.

6 Conclusion

This work introduces a novel framework to address high-quality novel view synthesis in scenes with refractive media through complex refractive surfaces under sparse training data. The proposed approach incorporates three key modules. (1) Refractive surface modeling, which can fit complex refractive surfaces via MLP based distances and Bezier surface fitting to achieve precise normal estimation of surfaces. (2) Adaptive hierarchical sampling, which dynamically adjusts the coarse-to-fine sampling ratio based on scene complexity. (3) Directional encoding with spherical harmonics, which replaces the 0th-frequency component of directional encoding to learn low-frequency view-dependent color variations. Extensive experiments conducted on public and self-build synthetic datasets demonstrate that the proposed approach achieves superior performance compared to existing neural rendering methods.

Acknowledgements. This work is supported in part by the Natural Science Foundation of China under Grant 62201479 and Grant 62476226, in part by the Postgraduate Innovation Fund Project by Southwest University of Science and Technology under Grant 25ycx2031, in part by Key Laboratory of Pattern Recognition and Intelligent Information Processing by Chengdu University under Grant MSSB-2025.

References

1. Bai, T., et al.: Neural rendering of underwater scenes under flat refractive geometry. In: Fifteenth International Conference on Information Optics and Photonics (CIOP 2024), vol. 13418, pp. 584–594. SPIE (2024)
2. Barron, J.T., et al.: Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5855–5864 (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5470–5479 (2022)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: TensorRF: Tensorial radiance fields. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV, vol. 13692, pp. 333–350. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-19824-3_20
5. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised NeRF: fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12882–12891 (2022)
6. Fujitomi, T., et al.: LB-NeRF: light bending neural radiance fields for transparent medium. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 2142–2146. IEEE (2022)
7. Hu, Y., Li, T.M., Anderson, L., Ragan-Kelley, J., Durand, F.: Taichi: a language for high-performance computation on spatially sparse data structures. ACM Trans. on Graph. (TOG) **38**(6), 1–16 (2019)
8. Mildenhall, B.: Local light field fusion: practical view synthesis with prescriptive sampling guidelines. ACM Trans. on Graph. (ToG) **38**(4), 1–14 (2019)
9. Mildenhall, B., et al.: NeRF: representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision, pp. 405–421 (2020)
10. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. on Graph. (TOG) **41**(4), 1–15 (2022)
11. Park, K., et al.: NeRFies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5865–5874 (2021)
12. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10318–10327 (2021)
13. Sethuraman, A.V., Ramanagopal, M.S., Skinner, K.A.: WaterNeRF: neural radiance fields for underwater scenes. In: OCEANS 2023-MTS/IEEE US Gulf Coast, pp. 1–7. IEEE (2023)
14. Wang, Q., et al.: Ibrnet: learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2021)
15. Wang, Y., et al.: Mesh-guided multi-view stereo with pyramid architecture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2039–2048 (2020)
16. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)

17. Wang, Z., et al.: NeReF: neural refractive field for fluid surface reconstruction and rendering. In: 2023 IEEE International Conference on Computational Photography (ICCP), pp. 1–11. IEEE (2023)
18. Wu, L., et al.: Neural directional encoding for efficient and accurate view-dependent appearance modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21157–21166 (2024)
19. Zhan, Y., Nobuhara, S., Nishino, K., Zheng, Y.: NeRFrac: neural radiance fields through refractive surface. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 18402–18412 (2023)
20. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
21. Zhang, X., Zhong, L., Leng, Q., Ran, L., Chu, H.: Underwater light field camera calibration based on multi-layer flat refractive model and multi-projection-center model. *Acta. Optica. Sinica.* **42**(12), 228–236 (2022)