The Visual Computer
https://doi.org/10.1007/s00371-023-03070-y

ORIGINAL ARTICLE

Rendering real-world unbounded scenes with cars by learning positional bias

Jiaxiong Qiu[1] · Ze-Xin Yin[1] · Ming-Ming Cheng[1] · Bo Ren[1]

Accepted: 16 August 2023

Abstract
In real-world unbounded outdoor scenes with cars, there are various specular reflections caused by the surrounding environment appearing on the reflective surfaces of cars. Background regions of unbounded scenes encode inherent ambiguity of rendering, and specular reflections on cars violates the multi-view consistency. NeRF++ struggles in these scenes because of the enormous ambiguity. To deal with the challenges of rendering unbounded scenes with cars, we present a novel module to strengthen the capability of the basic model in this task. We propose to learn the positional bias between sampled points along a camera ray and target points along the incident light by multi-layer perceptrons to reconstitute the input points and view direction with regularization constraints for physical rendering. Considering the variety of materials and textures in unbounded scenes, we implicitly separate learned foreground colors into two components, diffuse and specular colors, to acquire smooth results. Our module improves basic models by 2.5% on average SSIM in our extensive experiments, produces more photo-realistic novel views of real-world unbounded scenes than other compared methods, and achieves the physical color editing of cars.

Keywords Unbounded scenes · Specular reflections · Positional bias · Regularization

1 Introduction

View synthesis is a fundamental and challenging task in computer vision and graphics. It requires consumer cameras to capture sparse views of a scene and generates photo-realistic images from a free viewpoint. This makes view synthesis able to provide realistic navigation in the metaverse and provide more reliable data augmentation for deep models of autonomous driving.

Neural radiance field (NeRF) [10] has become a popular view synthesis technique that adopts an implicit volumetric function to represent a scene. This function is parameterized by fully connected networks that map spatial points $\mathbf{x}$ sampled along camera rays and the view direction to the volume density and color. This approach produces promising results of realistic view synthesis with varying resolutions. However, NeRF struggles in unbounded scenes, where images are captured at free viewpoints, and the bounds of each view are various and unpredictable. The content of unbounded scenes may appear at an arbitrary distance. This inherent ambiguity causes the performance of NeRF in novel views of unbounded scenes to degrade dramatically. NeRF++ [32] tackles this issue and proposes an inverted sphere parameterization to extend NeRF in unbounded scenes. It splits an unbounded scene into two components: foreground and background. Then, it uses two neural networks to model each component and combine them to be the rendered image. NeuS [26] adopts the scheme of NeRF++ to deal with the background of scenes and construct effective weights for rendering the foreground.

Although NeRF++ handles the inherent ambiguity of unbounded scenes well, it is still limited by other ambiguities caused by specific objects which violates the multi-view consistency. In this paper, we focus on more challenging unbounded scenes where cars exist. As Figs. 1 and 2 show, the surface of a car is composed of highly reflective materials and encodes apparent specular reflections (e.g., trees and

✉ Bo Ren
rb@nankai.edu.cn

Jiaxiong Qiu
qiujiaxiong727@gmail.com

Ze-Xin Yin
Zexin.Yin.cn@gmail.com

Ming-Ming Cheng
cmm@nankai.edu.cn

[1] TMCC, College of Computer Science, Nankai University, Tianjin 300000, China

Published online: 12 September 2023

clouds) from the surrounding environment. Specular reflections on the same region of a car present different contents in different views according to the Fresnel effect. This issue prevents NeRF-based models from interpolating the accurate color of cars at novel viewpoints.

Recent Phong-based methods [4, 21, 24, 34] introduce the bidirectional reflectance function (BRDF) [7, 8] to address objects with reflective surfaces. These methods trace the incident light with normal and use perfect object masks or synthetic objects without complex backgrounds to evaluate their performance. Ref-NeRF [24] adopts the gradient of the volume density [4, 21] in reference to spatial locations as the ground-truth normal of scenes. However, when putting cars into real-world unbounded scenes, the gradient of the volume density occurs massive errors and noise as shown in Fig. 2a. A small noise of normal causes a large distance deviation of source objects which produce the content of specular reflections in the background. So, this issue causes more ambiguity in our task.

Phong-based methods consider that the incident light path is the optimal solution for tackling the ambiguity from specular reflections. Based on the above observations, it is hard to trace the actual incident light directly in our task. Thus, we propose to use neural networks to find this solution without the surface normal. As Fig. 2b shows, two incident lights emitted from objects in the background are captured at two viewpoints. We consider the positional bias (e.g., Δp and $\Delta p^{'}$) between the target point along the incident light and a

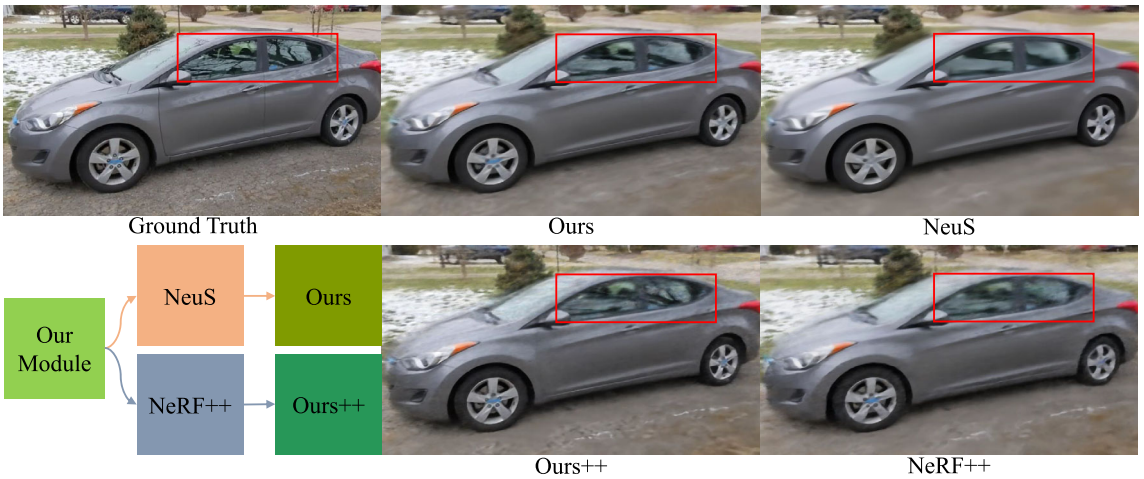

Fig. 1 We propose a novel module to improve the performance of basic models (e.g., NeuS [26] and NeRF++ [32]) for rendering unbounded scenes with cars. Our module is integrated into basic models and preserves fine details of specular reflections on the car

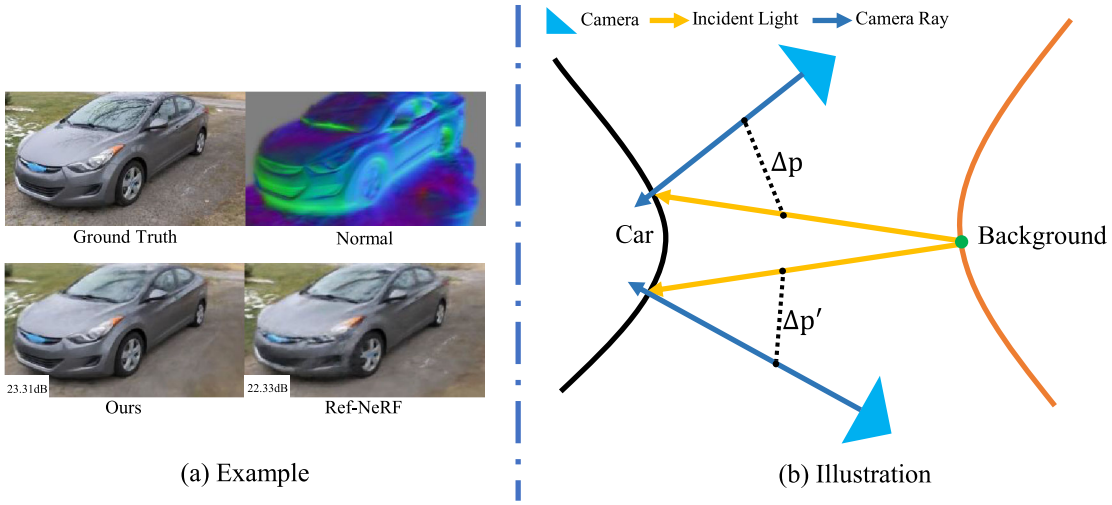

Fig. 2 **a** Phong-based methods rely on the quality of the normal, which is estimated from the gradient of the volume density and degrades in unbounded scenes with cars. Our method is built on the positional bias instead of the normal and generates more accurate result than the SOTA phong-based method ([24]). **b** Illustration of the positional bias. We propose to implicitly learn the positional bias (e.g., Δp and $\Delta p^{'}$) between a ray and the target incident light path for retrieving points along the incident light path

point sampled along a camera ray. Each point along a camera ray related to reflection can be mapped to the incident light by a positional bias at each viewpoint. Inspired by deformable NeRF-based methods [11, 13, 27], we propose to retrieve the positional bias from sampled points along rays by neural networks. Specifically, we reconstitute these points by adding their 3D positions with the learned positional bias.

To improve the efficiency of networks for retrieving the positional bias, we design a searching space with a novel regularizer. To ensure reconstituted points from a ray along the same incident light, we propose a novel regularization term to adjust the distribution of reconstituted points. Moreover, inspired by Ref-NeRF [24], we separate the learned color into two components: diffuse and specular, to acquire smooth results in the foreground because of various materials and textures in it. Based on these novel schemes, we packed them as a novel module. We integrate it into two relatively basic models (NeuS and NeRF++). Our extensive experiments demonstrate that our module can produce meaningful performance gain of basic models in unbounded scenes with cars and be generalized to other reflective objects. Particularly, our module faithfully separates diffuse and specular colors and achieves the physical color editing of cars in real-world unbounded scenes.

To summarize, our main contributions are:

1. We design a novel module to strengthen the capability of NeRF-based models in unbounded scenes with cars, by learning the positional bias from the sampled points along rays.
2. We propose a regularizer to make neural networks retrieve the positional bias efficiently and a regularization term to make reconstituted points distributed along the target light path for physical rendering.
3. Our module facilitates two basic models rendering more accurate novel views in unbounded scenes with reflective objects.

2 Related works

2.1 Reflective objects rendering

Reflective objects are ubiquitous in real-world scenes. However, they are hard to be rendered correctly by a renderer without accurate ray-tracing. Synthesizing novel views in scenes with reflections is a challenging task. Image-based rendering approaches [5, 6, 14, 20, 25, 29] take captured images as the input and tackle this task through careful preprocessing and postprocessing. They conduct the reflection decomposition of each image. These methods need careful preprocessing and postprocessing and rely on additional prior information in well-bounded scenes. Our module is only supervised by the captured images and focuses on more challenging unbounded scenes. Eikonal fields [3] focus on the refractive objects which mainly contain the refracted light with a curve path. Our work is proposed to model the reflected light on the car surface with a straight line path. Recent inverse rendering methods [4, 21, 28, 31, 34, 35] are based on BRDF. They recover the reflectance of an object with simple BRDF and local illumination. These assumptions are ill-suited in our task because of the complex environment. NeRS [31] proposes to recover the shape and texture of an object from a sphere template by introducing the Phong model [12] with perfect object masks. Ref-NeRF [24] designs a parameterization of the view direction to render scenes with specular reflections. Our module handles transparent objects well and improves the rendering quality of basic models in unbounded scenes.

2.2 Unbounded scenes rendering

In unbounded scenes, images are captured by a free-moving camera, and objects exist at any distance from the camera. Learning-based IBR algorithms [16–18] has also been successfully applied in rendering unbounded scenes. However, they suffer from the 3D reconstruction quality of COLMAP [19], which is not stable in unbounded scenes. NeRF++ [32] splits unbounded scenes into foreground and background and uses two neural networks to model each part. Mip-NeRF 360 [2] also designs a scene parameterization like NeRF++ for Mip-NeRF [1] with large parameters; the key idea of it is the online distillation. In contrast with these methods, our module is proposed to help basic NeRF-based models tackle the ambiguity caused by cars in unbounded scenes rather than the inherent ambiguity of unbounded scenes.

3 Method

Given a set of input images and camera parameters of each view, we aim to render photo-realistic unbounded scenes with cars in free views. We design a novel module to achieve our goal and integrate our module into NeuS [26] and NeRF++ [32] separately to evaluate its effectiveness.

Unbounded scenes with cars encode the ambiguity from the background and specular reflections on cars. We focus on the foreground and propose to learn the positional bias for rendering specular reflections physically. An overview of our framework is shown in Fig. 3, which consists of two parts: the foreground region and the background region. Each part produces a color image of the corresponding region of a view. In this paper, we concentrate on the ambiguity caused by specular reflections on cars, i.e., the foreground region. The parameterization scheme by splitting foreground and background regions follows NeRF++ [32]. For the background

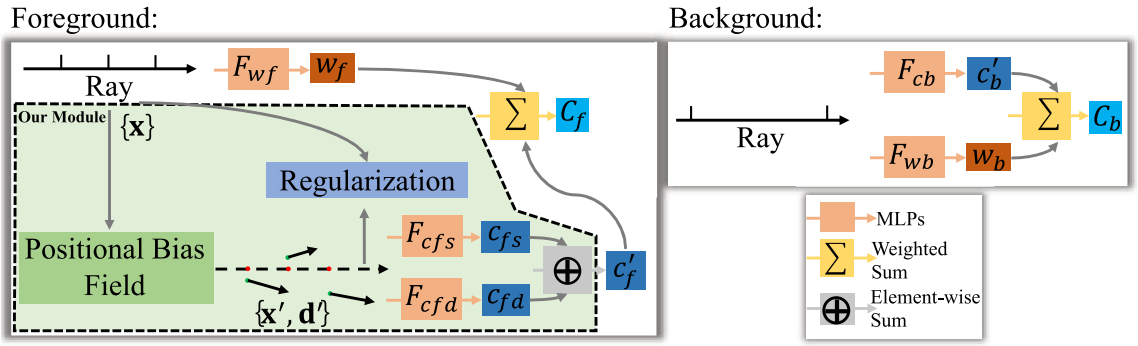

Fig. 3 An overview of our framework. In our work, an unbounded scene is decomposed into two parts: foreground and background. Our method focuses on the foreground part and consists of three parts: positional bias field (Sect. 3.2), regularization (Sect. 3.3) and rendering (Sect. 3.4)

region, we adopt the pipeline of NeRF [10] to generate the rendered background image $\mathbf{C}_\mathrm{b}$. More importantly, we propose several schemes to help basic models tackle this task. Specifically, we extract spatial points $\mathbf{x}$ and the view direction $\mathbf{d}$ of rays from camera parameters at first. $\mathbf{x}$ is also the input of the positional bias field we proposed to reconstitute points and the view direction with a novel regularizer. The output points $\mathbf{x}^{'}$ and view direction $\mathbf{d}^{'}$ are fed into two MLPs to implicitly predict specular $\mathbf{c}_\mathrm{fs}$ and diffuse $\mathbf{c}_\mathrm{fd}$ color values, which then summed as the learned foreground color $\mathbf{c}_\mathrm{f}^{'}$. Then, we feed $\mathbf{x}^{'}$ and $\mathbf{d}^{'}$ into a MLP F_wf to acquire densities, then estimate weights w_f of rendering the foreground image. We can acquire the rendered foreground image $\mathbf{C}_\mathrm{f}$ from w_f and $\mathbf{c}_\mathrm{f}^{'}$ by a weighted summation. Finally, the rendered foreground and background images are added to be supervised by the captured RGB image.

3.1 Preliminaries

As described by NeRF ([10]), a ray $\mathbf{r}$ extracted from camera parameters can be presented as:

$$\mathbf{r} = \mathbf{c} + t\mathbf{d}. \tag{1}$$

where $\mathbf{c}$ is the camera center, $\mathbf{d}$ is the view direction and t is the depth along this ray which sampled as spatial points $\mathbf{x}$. NeRF builds an implicit function, which maps the 3D position of points $\mathbf{x} = (x, y, z)$ to colors $\mathbf{c}$ and weights $\mathbf{w}$ of rendering. In practice, the positional encoding scheme [23] is applied in NeRF and projects $\mathbf{x}$ to a higher-dimensional space by sine and cosine functions with increasing frequencies. This makes MLPs model high-frequency details of captured images.

The color of $\mathbf{r}$ is determined by the summation:

$$\begin{cases} C(r) = \sum_{i=1}^{n} T_i (1 - \exp(-\sigma_i \Delta t_i))\mathbf{c}_i \\ T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_i \Delta t_i \right). \end{cases} \tag{2}$$

where n is the number of sampled points along a camera ray, $\Delta t = t_{i+1} - t_i$, σ and $\mathbf{c}$ are learned by neural networks. NeRF++ [32] proposes an inverted sphere parameterization to boost the capability of NeRF in unbounded scenes. It partitions the scene into two components with spatial points: the inside unit sphere and the outside inverted sphere, to represent the foreground and the background separately. Specifically, spatial points $\{\mathbf{x}|(x^2 + y^2 + z^2) \leq 1\}$ are considered in the foreground and other points are in the outer volume. Given that $t \in (0, t^{'})$ is inside the sphere and $t \in (t^{'}, \infty)$ means the region outside the sphere, then Eq. 2 can be rewritten as:

$$\begin{aligned} C(r) = &\sum_{i=1}^{n_{t'}} (1 - \exp(-\sigma_i \Delta t_i))\mathbf{c}_{fi} \\ &+ T_{n_{t'}} \sum_{i=n_{t'}}^{n_\infty} (1 - \exp(-\sigma_i \Delta t_i))\mathbf{c}_{bi}. \end{aligned} \tag{3}$$

where $T_{n_{t'}}$ is the accumulated transmittance of sampled points inside the sphere.

Based on the implicit volume rendering [10, 32] and surface rendering [30], NeuS [26] builds an appropriate connection between the implicit signed distance function (SDF) and rendered colors. It uses the same scheme of NeRF++ to handle the background scene and proposes constructing weights of rendering from the implicit SDF for the foreground scene and achieves promising scene surfaces and renderings.

3.2 Positional bias field

A car mainly comprises semi-transparent windows and a highly reflective car body. Both specular reflection and refraction appear on semi-transparent windows. On the car body, the specular reflection is distorted on the surface. So, the actual ray tracing is hard to be correctly modeled in a real-world scene with cars without the accurate surface normal and material information of cars. Due to the ambiguity

Fig. 4 Effect of the reconstituted view direction $\mathbf{d}^{'}$. It maintains the consistency of the camera center after reconstituting spatial points and preserves the thin structures of cars

of each training view caused by specular reflections on cars, synthesizing novel views of real-world unbounded scenes with cars is an ill-posed problem.

MirrorNeRF [27] warps points into a canonical space with a carefully designed latent code to handle position deviates of a specified mirror. It proposes to implicitly retrieve the error pattern in a deformable field. For rendering cars by NeRF-based methods, specular reflections break the consistency between adjacent views and confuse neural networks to generate inaccurate specular reflections of novel views. Inspired by MirrorNeRF, we consider implicitly restoring this consistency in a deformable field by retrieving positional bias. Specifically, we use a neural network $\Phi(\mathbf{x}) \rightarrow \Delta\mathbf{x}$ to build this field and predict a 3D position deviation for each sampled point.

We focus on the foreground points, i.e., $\mathbf{x} = (x, y, z)|$ $x, y, z \in [-1, 1]$. Thus, we set the last activation function of Φ with Tanh, and then the reconstituted points $\mathbf{x}^{'}$ can be presented as:

$$\mathbf{x}^{'} = \frac{(\mathbf{x} + \Delta\mathbf{x})}{2}. \tag{4}$$

$\mathbf{x}^{'}$ also belongs to the foreground. To remove the ambiguity of the camera center $\mathbf{c}$ after reconstituting spatial points, we reconstitute the view direction $\mathbf{d}$ according to Eq. 1:

$$\mathbf{d}^{'} = \frac{\mathbf{x}^{'} - \mathbf{c}}{t}. \tag{5}$$

As Fig. 4 shows, reconstituting the view direction preserves thin structures of the car and reduces noisy specular reflections.

3.3 Regularization

Although the positional bias field provides a canonical space to interpolate specular reflections in novel views, the interpolation accuracy depends on the implicit representations learned from reconstituted points. However, the efficiency of retrieving an appropriate positional bias is limited by the searching space. Ideally, the reconstituted points for rendering specular reflections should be on the path of the reflected

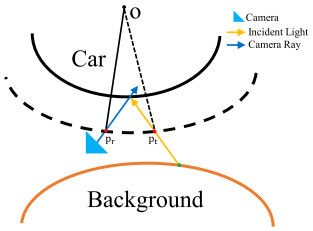

Fig. 5 Notation of the proposed searching space. The dotted curve is our designed searching space

light. In terms of the previous section, the default searching space is the entire foreground (i.e., a sphere). This searching space makes neural networks easily suffer from local minima. To improve the efficiency of retrieving the incident light path, we reduce the searching space of each sampled point from the whole sphere to a spherical surface.

We design a novel searching space to address this issue by introducing a regularizer. As illustrated in Fig. 5, p_r is a reference point on a camera ray and p_t is the target point on the incident light path. For each $\mathbf{x}_i = p_\mathrm{r}$ of n sampled foreground points, we estimate the distance $l_i = \sqrt{x^2 + y^2 + z^2}$ between it and the origin point o of the world coordinate system. Then, we take l_i as the radius of a sphere. Given the distance $l_i^{'}$ between the reconstituted points $\mathbf{x}_i^{'}$ and o, we tie $l_i^{'}$ by using a penalty:

$$\Gamma_p = \frac{1}{n}\sum_i \|l_i^{'} - l_i\|^2. \tag{6}$$

The spatial position of $\mathbf{x}^{'}$ is constrained on the same sphere with $\mathbf{x}$. This scheme ensures that each p_r corresponds to an optimal point p_t and encourages neural networks to seek p_t on a spherical surface.

Physically, the incident light travels in a line. Hence, the retrieved points also ideally distribute along a line. We propose the second regularization term to make the retrieved points satisfy this constraint. We aim to physically map the sampled points along a ray to the incident light by neural networks, so the reconstituted points from the same ray should also distribute along the same line. To achieve this, we propose to penalize the reconstituted points from a ray that is

Fig. 6 Effect of Γ_l. Keeping the reconstituted points along a line makes neural networks trace the target light physically and render more accurate specular reflections

non-collinear. We set the vector $\mathbf{q}_i = \mathbf{x}_{i+1}^{'} - \mathbf{x}_i^{'}$, then this regularization term can be written as:

$$\Gamma_l = \frac{1}{n}\sum_{i=1}^{n-1} \mathbf{q}_{i-1} \otimes \mathbf{q}_i. \tag{7}$$

where $\otimes$ is the cross product. This regularization advises neural networks to reconstitute points along the target light path for rendering more photo-realistic specular reflections. Figure 6 illustrates that our model can render more accurate specular reflections with the aid of Γ_l.

3.4 Rendering

Inspired by the traditional Phong model, the outgoing radiance L_{out} on a car surface can be modeled as $L_{\text{out}} = L_{\text{spec}} + L_{\text{diff}}$, where L_{spec} represents the specular part and L_{diff} represents the diffuse part. These parts are generated from the incident light. To implicitly trace the light, we assume L_{in} can be implicitly retrieved from L_{out} by the whole optimization process. In our work, we propose the positional bias field F_{p} to map the sampled points x to the target points $x^{'}$ and reconstitute the view direction $d^{'}$. With the help of regularization and neural networks $\{F_{\text{cs}}, F_{\text{cd}}\}$, we model the specular color c_{s} of L_{spec} by $c_{\text{s}} = F_{\text{cs}}(x^{'}, d^{'})$, and the diffuse color c_{d} of L_{diff} by $c_{\text{d}} = F_{\text{cd}}(x^{'})$. Two parts are linearly combined to be supervised by the captured image and adaptively optimized by training $\{F_{\text{cs}}, F_{\text{cd}}\}$. We add the diffuse color $\mathbf{c}_{\text{fd}}$ with the specular color $\mathbf{c}_{\text{fs}}$ of the foreground region to be the foreground color $\mathbf{c}_{\text{f}}^{'}$:

$$\mathbf{c}_{\text{f}}^{'} = \mathbf{c}_{\text{fd}} + \mathbf{c}_{\text{fs}}. \tag{8}$$

In terms of Eqn. 3, we can acquire the final rendered image C by:

$$C = C_{\text{f}} + C_{\text{b}}. \tag{9}$$

where C_{f} is the foreground image and C_{b} is the background image.

3.5 Loss Function

In this paper, we embed our module in NeuS [26] and NeRF++ [32]. We adopt the same loss function of each basic model and combine it with our proposed regularization. Given the batch size b, the loss function of NeuS is defined as:

$$\begin{cases} \mathcal{L}_r = \dfrac{1}{nb}\displaystyle\sum_{k,i}(|\nabla f(p_{k,i})| - 1)^2, \\ \mathcal{L}_c = \dfrac{1}{b}\displaystyle\sum_{i} \|C_i, \tilde{C}_i\|. \end{cases} \tag{10}$$

where f is the implicit SDF and $\tilde{C}$ is the ground-truth color. The loss function of NeRF++ is defined as:

$$\mathcal{L}_c = \frac{1}{b}\sum_{i} \|C_i, \tilde{C}_i\|^2. \tag{11}$$

Then, the whole loss function applied in this paper can be written as:

$$\mathcal{L} = \mathcal{L}_c + \alpha\mathcal{L}_r + \varphi_p\Gamma_p + \varphi_l\Gamma_l. \tag{12}$$

In practice, we set $\alpha = 0.1$ of NeuS and $\varphi_p = 1.0$, $\varphi_l = 0.01$ by default.

4 Experiments

To evaluate our module, we embed it in NeuS and NeRF++ with fair settings and acquired two results ('Ours' and 'Ours++') of each basic model in challenging unbounded scenes captured around cars. We conduct comprehensive experiments with comparisons among other approaches on eleven scenes quantitatively and qualitatively. As in NeRF++, the camera parameters of each scene are estimated by the publicly available tool COLMAP [19]. Then, they are normalized and recentralized to ensure the origin point o of the world coordinate system close to the car.

4.1 Datasets

4.1.1 CO3D and IBR dataset

CO3D dataset [15] is a large-scale dataset with real-world multi-view images of common object categories captured by a phone camera. We select four unbounded scenes ('Car-1','Car-2','Car-3' and 'Car-4') with cars from the CO3D dataset. In addition, we select two unbounded scenes ('Car-5' and 'Car-6') with cars from the IBR dataset [18], which capture images with a GoPro. In other NeRF-based papers, they usually evaluate their methods in almost six scenes. Based on this observation, we also collect six scenes from different datasets for evaluating the performance and robustness of our method.

4.1.2 Tanks and temples dataset

Tanks and Temples dataset [9] consists of hand-held captured scenes. To explore the effectiveness of our module in unbounded scenes with other reflective objects, we select a scene named 'Horse' from this dataset. Moreover, we use the same split between the additional four training and test scenes of this dataset for evaluating the performance of our module under the photometric variation during training.

4.2 Implementation details

We implement our module with several MLPs. For the positional bias field, $\Phi(\mathbf{x})$ is parameterized by an MLP, which consists of 4 linear layers. To decompose the foreground color, we use two MLPs to learn specular and diffuse colors separately. We follow the implementation details of basic models to integrate our module. For training each model, the batch size of rays at an iteration is 1024, and each model is trained for 200k iterations on a single NVIDIA Tesla V100 GPU. The optimizer and the scheduler of the learning rate are set from the released codes of each model.

4.3 Compared Methods

We evaluate our module against NeRF by the widely-used released codes of NeRF[1] with normalized device coordinates. We also evaluate our module against Stable View Synthesis [17] by the officially released codes[2], it is trained on extra scenes and represents the state of the art of IBR in rendering unbounded scenes with prior geometry information. We use the officially released codes[3] [4] of NeuS and NeRF++ as basic models and compare them separately. For current methods Ref-NeRF and Mip-NeRF 360, which are built on Mip-NeRF [1] and based on 3D conical frustums instead of rays. We adopt their officially released codes[5] with the same setting of other methods for fair comparisons.

Table 1 Quantitative comparison of test views between our method and previous methods on the CO3D and IBR datasets

Methods	PSNR↑	SSIM↑	LPIPS↓	#Params
NeRF	21.87	0.589	0.531	1.2M
Stable view synthesis	21.40	0.730	**0.382**	12.3M
NeuS	23.78	0.762	0.473	1.4M
NeRF++	24.43	0.774	0.446	2.4M
Ours	24.65	0.790	0.441	1.8M
Ours++	**24.80**	**0.798**	0.412	3.4M

Best metrics are highlighted

Table 2 Quantitative comparison of test views between our method, Ref-NeRF ([24]) and Mip-NeRF 360 ([2]) on 'Car_6'

Methods	PSNR↑	SSIM↑	LPIPS↓
Ref-NeRF	22.47	0.667	0.626
Mip-NeRF 360	26.81	0.862	0.446
Ours++	**27.68**	0.864	0.441
Ours 360	27.09	**0.871**	**0.421**

Best metrics are highlighted

4.4 Metrics

We use three traditional image similarity metrics for quantitative evaluation of all methods: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). We also compute the Learned Perceptual Image Patch Similarity (LPIPS) [33], which is related to the human perceptual distance. These metrics are estimated between the output results of each model and ground-truth images in novel views on average of a scene.

4.5 Comparisons

Table 1 shows quantitative results of our method and previous models in novel views on the CO3D and IBR datasets. The performance of Stable View Synthesis(SVS) [17] relies on the 3D surfaces of scenes, which decrease dramatically in the unbounded scene with cars because of the inherent ambiguity and the reflective ambiguity. When embedded with our module, NeuS (Ours) achieves over 0.8 dB better than the basic model on PSNR, nearly 3% SSIM gain with lower LPIPS on average. It also attains more precise results with fewer

[1] https://github.com/yenchenlin/nerf-pytorch.

[2] https://github.com/isl-org/StableViewSynthesis.

[3] https://github.com/Totoro97/NeuS.

[4] https://github.com/Kai-46/nerfplusplus.

[5] https://github.com/google-research/multinerf.

Table 3 Quantitative comparison of test views between our method and basic models in each scene of the CO3D and IBR dataset

Scene	PSNR↑				SSIM↑				LPIPS↓			
	NeuS	NeRF++	Ours	Ours++	NeuS	NeRF++	Ours	Ours++	NeuS	NeRF++	Ours	Ours++
Car-1	21.48	21.62	**21.90**	21.82	0.756	0.741	**0.773**	0.770	0.419	0.410	0.382	**0.364**
Car-2	**22.90**	22.61	22.85	22.74	0.668	0.652	0.671	**0.690**	0.452	0.461	0.443	**0.417**
Car-3	25.58	25.76	26.02	**26.42**	0.829	0.829	0.834	**0.853**	0.411	0.383	0.407	**0.352**
Car-4	22.47	26.05	**26.31**	26.09	0.683	0.790	**0.807**	0.797	0.531	0.401	0.398	**0.389**
Car-5	23.87	23.84	**24.06**	**24.06**	0.809	0.802	**0.821**	0.816	0.528	0.525	**0.508**	0.510
Car-6	26.39	26.67	26.77	**27.68**	0.828	0.832	0.836	**0.864**	0.495	0.493	0.497	**0.441**

Best metrics are highlighted

Fig. 7 Qualitative comparison of test views between our method and previous approaches on the CO3D and IBR datasets. The best metrics on PSNR are highlighted. Our module helps basic models interpolate more accurate specular reflections on cars with fine details

parameters than NeRF++. NeRF++ (Ours++) strengthens the performance of NeRF++ on both PSNR and SSIM, with the second-lowest LPIPS. When compared with recent methods based on Mip-NeRF, as Table 2 presents, our method (Ours++) outperforms Ref-NeRF and achieves over 0.8dB better than Mip-NeRF 360 on PSNR. We apply our proposed schemes (Ours 360) to Mip-NeRF 360 (only reconstituting the view direction) and also improve its performance on all metrics.

We present quantitative results of each scene on the CO3D and IBR datasets in Table 3. The first group consists of scenes captured by the camera motion with a higher degree of freedom than the camera motion of the second group. Our module significantly helps basic models render more accurate novel views in each scene. Especially, our module improves the PSNR of NeuS by 3.7dB in 'Car-4' and NeRF++ by 1dB in 'Car-6'. Figure 7 shows the qualitative results of test views in three scenes. SVS sometimes renders clearly visible reflections but fails to interpolate accurate specular reflections on cars. NeRF suffers from massive noise in the output results. NeuS synthesizes over-smooth images, and some regions of highlights on cars are missing. Our module tackles these problems of NeuS and recovers correct highlights. NeRF++ generates noisy reflections with a lack of fine details. Our module preserves more details of reflections than NeRF++.

4.6 Ablation study

Unbounded scenes with cars encode much ambiguity in different views. Rendering these scenes is a challenging task. To explore the impact of each component of our module on the performance of basic models, we conduct a sufficient ablation study on 'Car-4' by disabling each component separately. Table 4 shows quantitative results of different settings for NeuS equipped with our module (Ours). With each component dropped, the performance degrades reasonably.

4.6.1 Effect of positional bias field

We propose to reconstitute the sampled points along rays for retrieving appropriate points along the incident light path. When the positional bias field is disabled, the performance of our model degrades over 1dB on PSNR and 3.5% on SSIM. This demonstrates the effectiveness of the positional bias field.

4.6.2 Effect of reconstituting view direction

The view direction is defined as the direction vector from the camera center to the sampled points according to Eq. 1. When acquiring the reconstituted points through the positional bias field, we can reconstitute the view direction to remove the ambiguity of the camera center. This helps neural networks generate more accurate results.

Table 4 Ablation study of our module on the 'Car-4'

Settings	PSNR↑	SSIM↑	LPIPS↓
Basic Model	22.47	0.683	0.531
No $\mathbf{d}'$, w/ $\mathbf{d}$	25.78	0.793	0.411
No positional bias field	25.27	0.772	0.448
No regularization	25.65	0.788	0.420
No diffuse	25.58	0.787	0.419
Ours	**26.31**	**0.807**	**0.398**

Best metrics are highlighted

4.6.3 Effect of regularization

We design a searching space for retrieving appropriate points efficiently by a novel regularizer and propose a regularization term to adjust the distribution of reconstituted points for more accurate reflection interpolation. To determine whether the regularization is a significant scheme for rendering unbounded scenes with cars, we disable it from our full model. The performance of the model without the regularization on PSNR drops by 0.6dB when compared to our full model. This demonstrates that regularization plays an important role in our tasks.

4.6.4 Effect of diffuse color

We decompose the learned foreground color into diffuse color and specular color to reduce the effect of various materials and textures of cars for rendering. The diffuse color encodes the essential information of cars with less ambiguity and makes neural networks can pay more attention to rendering more complicated specular colors. To verify the necessity of this decomposition, we disable the diffuse path with other components of our module enabled. The performance of the model without the diffuse path on all metrics degrades. However, due to the robust positional bias field, it also achieves the 3.1dB gain on PSNR when compared to the basic model.

4.6.5 Effect of parameters

We use α,φ_l and φ_p to adjust weights of each regularization term for better performance. $\alpha = 0.1$ is adopted from NeuS. To evaluate the effect of other parameters, we first set $\varphi_p = 1.0$ and change φ_l. Then, we set $\varphi_l = 0.01$ and change φ_p. The results are presented in Table 6.

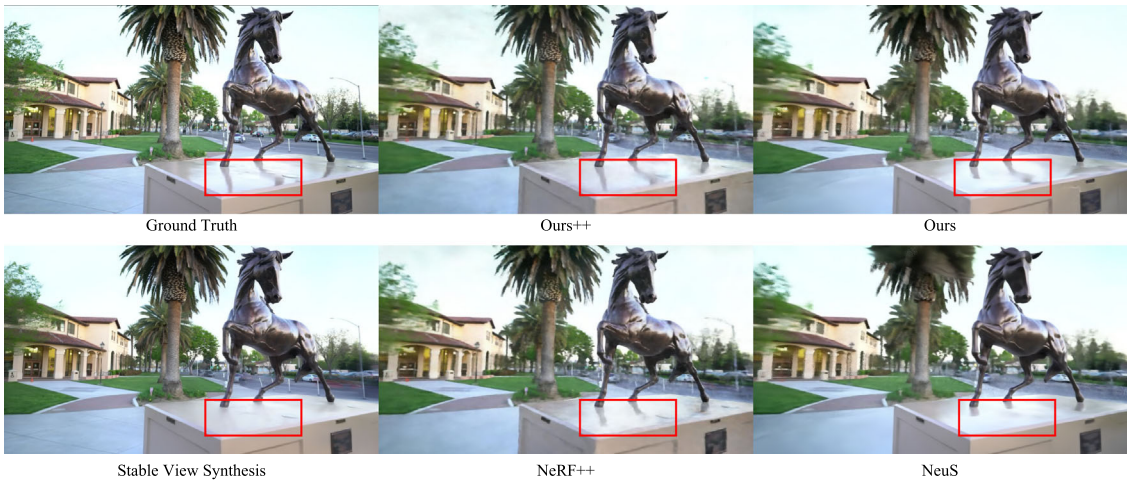

Fig. 8 We evaluate our module on an unbounded scene with a reflective table on the 'Horse' of Tanks and Temples dataset. Basic models embedded with our module interpolate more accurate specular reflections on the table, where the SOTA method fails

Table 5 Quantitative comparison of test views between our method and previous approaches on the 'Horse' of Tanks and Temples dataset

Methods	PSNR↑	SSIM↑	LPIPS↓
NeRF	17.37	0.568	0.554
Stable view synthesis	22.46	**0.921**	**0.148**
NeuS	21.72	0.826	0.380
NeRF++	22.49	0.838	0.363
Ours	21.84	0.836	0.369
Ours++	**22.63**	0.851	0.353

Best metrics are highlighted

4.7 Generalization

We also explore the generalization ability of our module by replacing cars with other reflective objects in unbounded scenes. We select a scene named 'Horse' of Tanks and Temples dataset with a reflective table. Figure 8 shows qualitative results of our method, baselines, and the SOTA model. Our module successfully helps basic models interpolate more accurate reflections on the tabletop in this scene. Table 5 presents quantitative results of novel views in this scene. We integrate our module into NeRF++ and achieve the best performance on PSNR when compared with previous approaches.

5 Discussion

5.1 Appearances

Decomposing the diffuse color and the specular color from real-world reflective objects without any prior is an ill-posed problem. As Fig. 9 shows, our module faithfully separates the diffuse and specular color from the rendered foreground color in real-world unbounded scenes. The diffuse color is produced from the reconstituted points, and specular reflections are missing. The specular color is learned from the reconstituted points combined with the view direction and contains actual specular reflections. During training the model embedded with our module, we only adopt the captured image to supervise the final rendered image. So the diffuse and specular colors are separated implicitly. This also illustrates that our points bias field can model foreground scenes physically. Our method decomposes the car into diffuse and specular parts without corresponding ground-truth data, generating a high-fidelity diffuse part is an ill-posed problem. In the unbounded scenes, the sampled points behind the car also affect the quality of the diffuse part. In the first row of Fig. 10, our model considers the red light as the diffuse part because the light is not on. In the last row of Fig. 10, the light of the red car is considered as the specular part.

5.2 More details and comparisons

We test the running time of our method and basic models on a single GPU. The resolution of the testing image is 947 × 536. NeuS costs 64.3 s per image, and ours costs 72.8 s per image. NeRF++ costs 77.2 s per image, and ours++ costs 95.8 s per image. For the view consistency, our method is stable under changing views, we show the continuously changing views and estimate working regions from the optical flow by RAFT [22] in Fig. 11. The positional bias field transforms spatial sampled points from the fixed volumetric space to a deformable space, for tracing the incident light more effectively and physically. The temporal consistency is learned during the optimization. We also illustrate the positional bias field visually in Fig. 12.

Fig. 9 Our module can edit the foreground of a scene by modifying the diffuse color. 'Ours_g' means our result with green diffuse color. 'Ours_p' means our result with purple diffuse color

Fig. 10 Example of appearances from the diffuse and specular paths in our module. These appearances of the same view contain the same background color. Our module separates meaningful diffuse and specular parts from the ground truth

Based on the above observations, we can modify the learned diffuse color and then edit the foreground of scenes. Figure 10 shows a visualization of editing a red car by changing the channel order of the diffuse color. The surface color of this car is changed physically, while the rendered specular reflections still appear on cars. This indicates that besides interpolating accurate reflections in novel views, our module can supply more reliable augmentation of a car by physically editing its color.

We further compare our method with relative models in four scenes of the Tanks and Temples dataset. These scenes suffer from photometric variation across images, and there are few reflective objects in these scenes. So, these scenes are ill-suited to our goal. As Table. 7 presents, our module facilitates NeRF++ outperforming Mip-NeRF, which has much larger parameters. SVS achieves the best performance because it can predict the photometric variation of these scenes.

Table 6 Effect of parameters of regularization terms

φ_l	0.01	0.25	0.5	0.75
PSNR	26.31	25.23	24.80	23.69
φ_p	1.0	0.7	0.5	0.3
PSNR	26.31	26.16	25.10	26.18

5.3 Limitations

We zoom in several regions of the car to visualize the specular reflection in Figs. 4, 6, and 7. Although our module achieves more accurate renderings than basic models in unbounded scenes with cars, several details of specular reflections are still missing. One possible optimization of further work is enhancing the representation ability of neural networks for rendering the specular color.

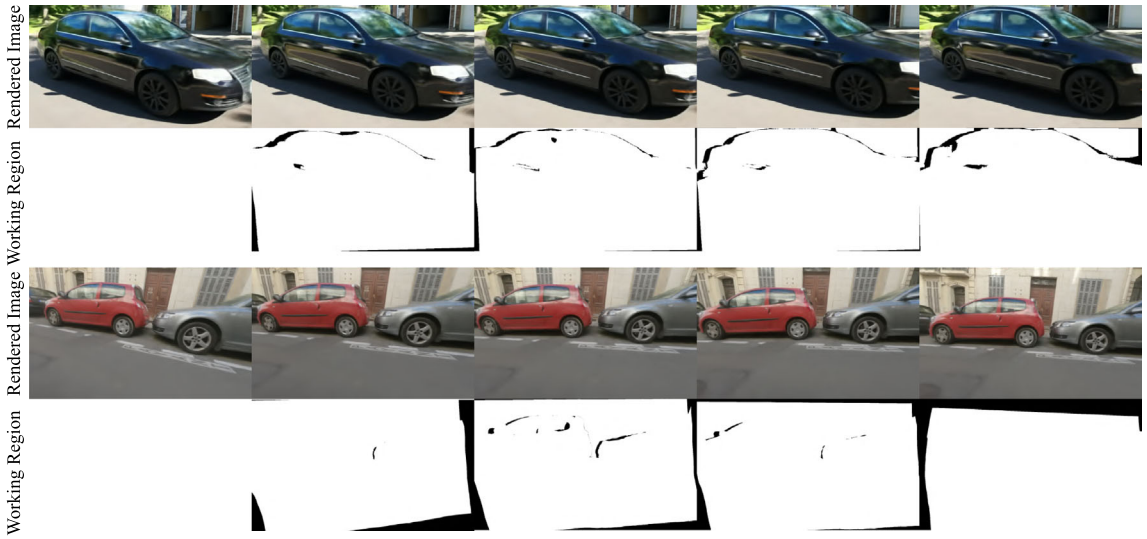

Fig. 11 View consistency. From left to right, the rendered images are continuously changing views. The white regions are working regions of our model

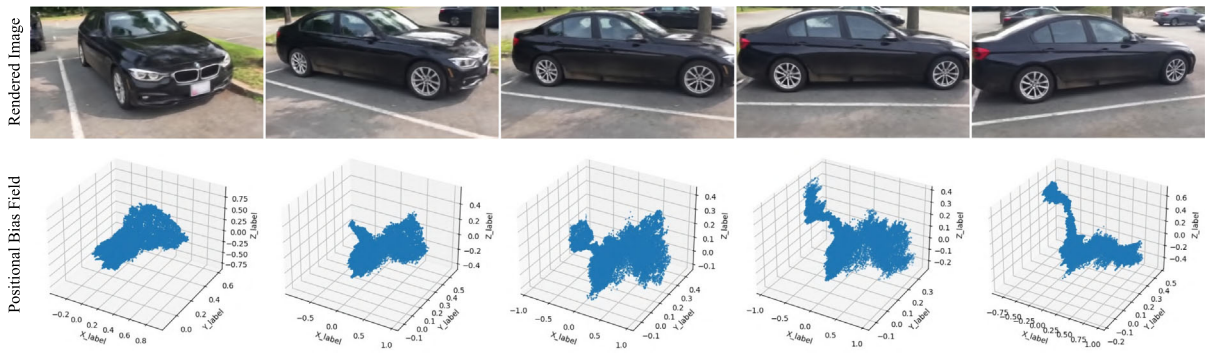

Fig. 12 Visualization of the positional bias field. Top: rendered novel views. Bottom: transformed spatial points in the positional bias field

Table 7 Quantitative comparison of test views between our method and relative approaches on the Tanks and Temples dataset [9]

Methods	PSNR↑	SSIM↑	LPIPS↓	#Params
NeRF	18.72	0.609	0.473	1.2M
NeRF++	19.32	0.647	0.425	2.4M
Mip-NeRF	19.85	0.697	0.340	9.0M
Stable View Synthesis	**21.13**	**0.777**	**0.209**	12.3M
Ours++	20.03	0.696	0.459	3.4M

Best metrics are highlighted

6 Conclusion

In this work, we aim to render accurate novel views in unbounded scenes with cars and have proposed a novel module to facilitate basic models tackling this task. We focus on the ambiguity caused by specular reflections on cars. In our module, we propose a novel positional bias field learned from the sampled points along rays with two effective regularization terms, for retrieving the positional bias between the sampled points and the target incident light path. We implicitly decompose the foreground color into diffuse color and specular color for acquiring smooth results. We conduct extensive experiments on real-world datasets for evaluation. Our module significantly improves the performance of basic models in terms of quantitative and qualitative comparison. Moreover, our module faithfully separates the diffuse and specular parts in the foreground and makes the basic model that can edit scenes physically.

Author Contributions J-XQ contributed to conceiving, designing the analysis and writing; Z-XY performed data collection; BR performed writing—review and editing; and M-MC performed supervision.

Funding This work is supported by the National Key Research and Development Program of China Grant (No.2018AAA0100400), NSFC (No.61922046) and NSFC (No.62132012).

Data availability We used two common datasets in this work: CO3D [15] (https://ai.facebook.com/datasets/CO3D\discretionary--dataset/), IBR [18] (https://gitlab.inria.fr/sibr/projects/semantic\discretionary--

reflections/semantic_reflections/). and Tanks and Temples datasets [9] (https://www.tanksandtemples.org/).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., et al.: (2021a) 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
2. Barron, J.T., Mildenhall, B., Verbin, D., et al.: (2021b) Mip-nerf 360: unbounded anti-aliased neural radiance fields. arXiv preprint arXiv:2111.12077
3. Bemana, M., Myszkowski, K., Revall Frisvad, J., et al.: (2022) Eikonal fields for refractive novel-view synthesis. In: ACM SIGGRAPH 2022 Conference Proceedings, pp. 1–9
4. Boss, M., Braun, R., Jampani, V., et al.: (2021) Nerd: neural reflectance decomposition from image collections. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12684–12694
5. Firmino, A., Frisvad, J.R., Jensen, H.W.: Progressive Denoising of Monte Carlo Rendered Images. In: Computer Graphics Forum, pp. 1–11. Wiley (2022)
6. Guo, Y.C., Kang, D., Bao, L., et al.: (2021) Nerfren: neural radiance fields with reflections. arXiv preprint arXiv:2111.15234
7. Immel, D.S., Cohen, M.F., Greenberg, D.P.: A radiosity method for non-diffuse environments. ACM Siggraph. Comput. Graph. **20**(4), 133–142 (1986)
8. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp. 143–150 (1986)
9. Knapitsch, A., Park, J., Zhou, Q.Y., et al.: Tanks and temples: benchmarking large-scale scene reconstruction. ACM Trans. Graph. (ToG) **36**(4), 1–13 (2017)
10. Mildenhall, B., Srinivasan, P.P., Tancik, M., et al.: Nerf: representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision, pp. 405–421. Springer, (2020)
11. Park, K., Sinha, U., Barron, J.T., et al.: Nerfies: deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5865–5874 (2021)
12. Phong, B.T.: Illumination for computer generated pictures. Commun. ACM **18**(6), 311–317 (1975)
13. Pumarola, A., Corona, E., Pons-Moll, G., et al.: 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)
14. Qiu, J., Zhu, Y., Jiang, P.T., et al.: Rdnerf: relative depth guided nerf for dense free view synthesis. Vis. Comput. (2023). https://doi.org/10.1007/s00371-023-02863-5
15. Reizenstein, J., Shapovalov, R., Henzler, P., et al.: Common objects in 3d: large-scale learning and evaluation of real-life 3d category reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10901–10911 (2021)
16. Riegler, G., Koltun, V.: Free view synthesis. In: European Conference on Computer Vision, pp 623–640 . Springer (2020)
17. Riegler, G., Koltun, V.: Stable view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12216–12225 (2021)
18. Rodriguez, S., Prakash, S., Hedman, P., et al.: Image-based rendering of cars using semantic labels and approximate reflection flow. Proc. ACM Comput. Graph. Interact. Tech. **3** (2020)
19. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
20. Sinha, S.N., Kopf, J., Goesele, M., et al.: Image-based rendering for scenes with reflections. ACM Trans. Graph. (TOG) **31**(4), 1–10 (2012)
21. Srinivasan, P.P., Deng, B., Zhang, X., et al.: Nerv: neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7495–7504 (2021)
22. Teed, Z., Deng, J.: Raft: recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, pp. 402–419. Springer (2020)
23. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in neural information processing systems 30 (2017)
24. Verbin, D., Hedman, P., Mildenhall, B., et al.: Ref-nerf: structured view-dependent appearance for neural radiance fields. arXiv preprint arXiv:2112.03907 (2021)
25. Vicini, D., Adler, D., Novák, J., et al.: Denoising Deep Monte Carlo Renderings. In: Computer Graphics Forum, pp. 316–327. Wiley (2019)
26. Wang, P., Liu, L., Liu, Y., et al.: Neus: learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021a)
27. Wang, Z., Wang, L., Zhao, F., et al.: Mirrornerf: one-shot neural portrait radiance field from multi-mirror catadioptric imaging. In: 2021 IEEE International Conference on Computational Photography (ICCP), IEEE, pp. 1–12 (2021b)
28. Wu, H., Hu, Z., Li, L., et al.: Nefii: Inverse rendering for reflectance decomposition with near-field indirect illumination. arXiv preprint arXiv:2303.16617 (2023)
29. Xu, J., Wu, X., Zhu, Z., et al.: Scalable image-based indoor scene rendering with reflections. ACM Trans. Graph. (TOG) **40**(4), 1–14 (2021)
30. Yariv, L., Kasten, Y., Moran, D., et al.: Multiview neural surface reconstruction by disentangling geometry and appearance. Adv. Neural Inf. Process. Syst. **33**, 2492–2502 (2020)
31. Zhang, J., Yang, G., Tulsiani, S., et al.: Ners: neural reflectance surfaces for sparse-view 3d reconstruction in the wild. Adv. Neural Inf. Process. Syst. **34**, 29835–29847 (2021)
32. Zhang, K., Riegler, G., Snavely, N., et al.: Nerf++: analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)
33. Zhang, R., Isola, P., Efros, A.A., et al.: 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)
34. Zhang, X., Srinivasan, P.P., Deng, B., et al.: Nerfactor: neural factorization of shape and reflectance under an unknown illumination. ACM Trans. Graph. (TOG) **40**(6), 1–18 (2021)
35. Zhang, Y., Sun, J., He, X., et al.: (2022) Modeling indirect illumination for inverse rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18643–18652

Jiaxiong Qiu is a PhD student from the College of Computer Science at Nankai University. He received his master degree supervised at University of Electronic Science and Technology of China in 2020. He obtained his bachelor degree at Dalian Maritime University in 2017. His research interests include computer vision, computer graphics, robotics, and deep learning.

Ze-Xin Yin is a graduate student at the College of Computer Science, Nankai University. He received his bachelor's degree from Xidian University of Computer Science and Technology in 2021. His research interests include 3D computer vision and deep learning.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then, he was 2 years of research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, and CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TIP.

Bo Ren received the PhD degree from Tsinghua University in 2015. He is currently an associate professor in the College of Computer Science, Nankai University, Tianjin. His research interests include physically based simulation and 3D scene reconstruction and analysis.