SS-NeRF: Physically Based Sparse Spectral Rendering With Neural Radiance Field

Ru Li, *Member, IEEE*, Jia Liu, Guanghui Liu, *Senior Member, IEEE*, Shengping Zhang,
Bing Zeng, *Fellow, IEEE*, and Shuaicheng Liu, *Senior Member, IEEE*

***Abstract*—In this paper, we propose SS-NeRF, the end-to-end Neural Radiance Field (NeRF)-based architectures for high-quality physically based rendering with sparse inputs. We modify the classical spectral rendering into two main steps, 1) the generation of a series of spectrum maps spanning different wavelengths, 2) the combination of these spectrum maps for the RGB output. The proposed architecture follows these two steps through the proposed multi-layer perceptron (MLP)-based architecture (SpectralMLP) and spectrum attention UNet (SAUNet). Given the ray origin and the ray direction, the SpectralMLP constructs the spectral radiance field to obtain spectrum maps of novel views, which are then sent to the SAUNet to produce RGB images of white-light illumination. Applying NeRF to build up the spectral rendering is a more physically-based way from the perspective of ray-tracing. Further, the spectral radiance fields decompose difficult scenes and improve the performance of NeRF-based methods. Previous baseline, such as SpectralNeRF, outperforms recent methods in synthesizing novel views but requires relatively dense viewpoints for accurate scene reconstruction. To tackle this, we propose SS-NeRF to enhance the detail of scene representation with sparse inputs. In SS-NeRF, we first design the depth-aware continuity to optimize the reconstruction based on single-view depth predictions. Then, the geometric-projected consistency is introduced to optimize the multi-view geometry alignment. Additionally, we introduce a superpixel-aligned consistency to ensure that the average color within each superpixel region remains consistent. Comprehensive experimental results demonstrate that the proposed method is superior to recent state-of-the-art methods when synthesizing new views on both synthetic and real-world datasets.**

***Index Terms*—Spectral rendering, neural radiance field, sparse scene reconstruction.**

I. Introduction

NEWTON found that white light can be dispersed into a series of spectrums with various colors from red to violet by passing light through the glass prism [1]. After that, the research on spectral theory developed rapidly [2], [3]. Until now, the application of spectral images has involved various aspects of daily life, including object detection [4], face recognition [5], and so on [6], [7]. Spectral images can record and reveal the electromagnetic radiation intensity information of objects, which is an important interdisciplinary subject mainly involving physics and chemistry [8].

Spectral rendering is a fundamental problem in computer graphics, which can understand the absorption, reflection, and other interactions with objects and has been used to generate photo-realistic images [9]. Conventional spectral rendering involves two transformations: 1) the spectral power distribution L to the XYZ image, achieved by using integral operation through the visible light; 2) the XYZ image to the RGB image, realized by the conversion matrix [10]. Based on the transformations of $L \rightarrow \text{XYZ} \rightarrow \text{RGB}$, physically-based spectral rendering has been widely researched over the past decades [11], [12], [13] where light transport is modeled with multiple wavelengths instead of just using the red, green, and blue components. Such methods predict photo-realistic rendering so that no effect that contributes to the interaction of light with a scene is neglected. However, they merely generate one image of the current viewpoint and are limited in representing scenes.

Recently, Neural Radiance Field (NeRF) is designed to render compelling images of 3D scenes from novel viewpoints [14]. NeRF-based methods achieve photo-realistic rendering of scenes by encoding the volumetric density and color of a scene within the weights of a coordinate-based multi-layer perceptron (MLP). Subsequently, a series of works focused on recovering the radiance field using deep neural networks [15], [16], [17], [18]. This approach has enabled significant progress toward photo-realistic view synthesis and can solve the limitation that spectral rendering cannot represent the overall scene.

Applying NeRF-based architecture to achieve the physically-based spectral rendering involves constructing the spectral radiance field for the scene information along the ray, which facilitates the rendering process and keeps the rendering pipeline simple yet effective by employing a physically-based multi-spectral integral calculation for the $L \rightarrow \text{XYZ} \rightarrow \text{RGB}$ conversion. For spectral rendering, applying the neural radiance field to learn the spectrum maps and using the integration of spectral bands are more physically-based ways. Spectral radiance fields model light transport across multiple wavelengths, which captures the real-world interactions of light

Received 21 January 2025; revised 4 August 2025; accepted 9 September 2025. Date of publication 17 September 2025; date of current version 3 December 2025. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62372091, Grant 62272134, and Grant 62402136, in part by the the National Natural Science Foundation of Shandong Province under Grant ZR2024QF064, and in part by Hainan Province Science and Technology Plan Projects under Grant ZDYF2025(LALH)001 and Grant ZDYF2025(LALH)003. Recommended for acceptance by B. Shi. *(Corresponding authors: Shuaicheng Liu; Shengping Zhang.)*

Ru Li and Shengping Zhang are with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: liru@hit.edu.cn; s.zhang@hit.edu.cn).

Jia Liu, Guanghui Liu, Bing Zeng, and Shuaicheng Liu are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: liushuaicheng@uestc.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2025.3611376

with different materials compared to traditional RGB-based methods that rely on only three color channels. This allows our method to preserve fine-grained characteristics. For NeRF-based methods, decomposing scenes into multiple spectral components helps disentangle complex scene structures, which is superior to previous NeRF-based methods in terms of geometry and texture reconstruction under complex scenes and image quality of the novel viewpoint synthesis. The motivation, that is, the need for spectral radiance fields is: spectral information can provide more details on the material constitution of objects in the scene, which has been utilized in classic vision tasks, such as material classification [19]. The idea of importing spectral information to rendering is a new perspective, which may provide inspiration for rendering and vision tasks.

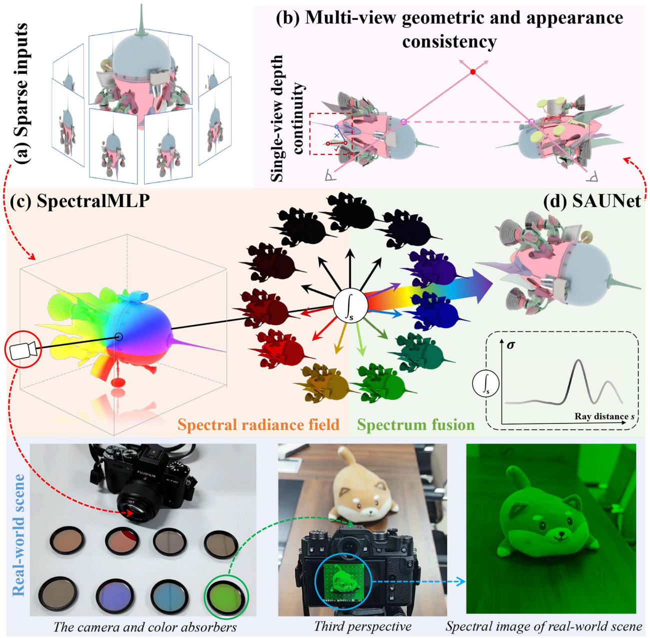

(a) Sparse inputs
(b) Multi-view geometric and appearance consistency
Single-view depth continuity
(c) SpectralMLP
(d) SAUNet
Spectral radiance field
Spectrum fusion
σ
Ray distance s
Real-world scene
The camera and color absorbers
Third perspective
Spectral image of real-world scene

Fig. 1. The SS-NeRF builds up the process of spectral rendering based on the (a) sparse inputs. The first step in (c) samples spectrum points along the ray and uses volume rendering to generate spectrum maps. The second step in (d) fuses discrete spectrum maps to obtain RGB images. Considering the incomplete supervision from sparse views, the SS-NeRF introduces the single-view depth continuity mechanism and the multi-view geometric and appearance consistency constraints in (b) to achieve more realistic and coherent renderings with sparse inputs.

Following the variant rendering pipeline, we design an MLP-based architecture, termed SpectralMLP, which maps a 5D input coordinate (3D position and 2D viewing direction) to the scene representation (volume density and spectral radiance). Volume rendering is applied to compose these values into discrete spectrum maps. In order to better extract the spectral information, the spectrum attention (SA) module is designed to explore the correlations between spectrum maps. Then, SAUNet is proposed to combine these spectrum maps into high-quality RGB images.

While SpectralNeRF [20] demonstrates the effectiveness of incorporating spectral radiance fields for physically-based rendering under dense input views, it relies heavily on the availability of a number of training images to achieve high-quality reconstruction. This assumption is often impractical in real-world scenarios, especially when capturing spectral images across multiple wavelengths. Many researchers focus on addressing this problem using sparse views through methods such as depth regularization [21], [22], large-scale pre-training [23], [24], 2D pre-trained models [25], [26], and frequency annealing [27], making it a critical area of interest. However, these methods mainly focus on RGB inputs and often rely on pre-training or pre-trained models, which may not generalize well to spectral data or lead to collapsed reconstructions and overly smooth visual outcomes.

In this paper, we extend the SpectralNeRF [20] to Sparse SpectralNeRF (SS-NeRF), which enables physically-based rendering under sparse NeRF settings (Fig. 1(a)). SS-NeRF is specifically motivated by the need to achieve high-fidelity novel view synthesis under sparse input views, a much more challenging setting where traditional construction methods tend to suffer from geometric collapse, blurry appearance, and inconsistent color across views. SS-NeRF addresses the challenges of defective geometry and appearance caused by limited observations using several key mechanisms in Fig. 1(b), including the depth-aware continuity, the geometric-projected consistency, and the superpixel-aligned constraint. The depth-aware continuity guarantees that the depth map predicted by NeRF remains consistent with the depth prior generated by the DPT model [28], while also preserving spatial smoothness within each single-view. The geometric consistency ensures that the geometric structure is accurately maintained across different views, reducing discrepancies and enhancing the overall geometric coherence. Meanwhile, the superpixel-aligned harmony segments the images into superpixels, guaranteeing consistent average color within each region between different views.

The contribution of SpectralNeRF [20] includes:

- We propose SpectralNeRF, that builds up physically-based rendering with NeRF from the spectral perspective, which leads to mutual enhancement of spectral rendering and NeRF-based methods.
- We design the SAUNet to fuse the discrete spectrum maps to generate high-quality RGB images, which can approach the integral calculation in spectral rendering.
- We render 8 spectral datasets and capture 2 real-world scenes with spectrum maps and RGB images, and provide comprehensive comparisons of these datasets with several NeRF-based methods to demonstrate the superiority.

In comparison to SpectralNeRF, we make the following new contributions in SS-NeRF:

- We propose SS-NeRF, a few-shot view synthesis framework that builds upon SpectralNeRF to handle sparse input views, addressing the challenges of limited observations and improving the quality of scene representation.
- We design the depth-aware continuity that enforces the spatial coherence within single-view depth maps, the geometric-projected consistency that ensures multi-view geometric alignment, and the superpixel-aligned constraint that reduces color discrepancies.
- We further render 2 spectral datasets and capture 8 real-world scenes with spectrum maps and RGB images, providing the detailed render configurations for the synthetic

datasets and capture guidelines for the real-world scenes. The effectiveness of SS-NeRF is validated on these datasets through extensive experiments.

II. Related Works

Neural Radiance Field for 3D Scenes: Using the neural network to represent a 3D scene and generate novel views with weights of MLP or other network parameters has become a hot topic. Previous methods address the issue with explicit discrete representations [29], [30], [31]. These methods are limited in applying gradient-based techniques to optimize mesh geometry and topology. Since Mildenhall et al. introduced the differentiable volumetric rendering technique to optimize a neural radiance field [14], a number of studies have been carried out to dive deeper into NeRF-based architecture, including the detail preservation methods [15], [16], [32], the faster training and inference of NeRF [33], [34], the extension from image to video [35], [36], the refractive novel-view synthesis [37], the dynamic scenes [38], the LiDAR scenes [39], the infrared and spectral scenes [40], and the event cameras [41]. Poggi et al. proposed X-NeRF to generate infrared and spectral images directly using MLP. However, they did not consider the process of spectral rendering and the combination of spectral images [40]. It is challenging for these methods to represent scenes with complex textures. To solve such problems, we present a novel NeRF-based architecture, which introduces spectral information into the radiance field to simplify complicated scenes.

3D Scene Reconstruction with Sparse Inputs: There is a growing number of NeRF-based studies concentrated on synthesizing novel views from sparse inputs, which can mainly be divided into three categories: 1) leveraging pre-trained models, which train a NeRF model on pre-selected scenes and fine-tune on the target scene [23], [24], [26]; 2) imposing constraints on the continuity of object geometry and semantics [21], [22], [42]; and 3) incorporating depth supervision [43], [44], [45]. For example, GeCoNeRF [46] leverages rendered depth maps at unobserved viewpoints to warp sparse input images at the patch level. This method improves consistency in geometry reconstruction under sparse views. 3D Gaussian Splatting (3DGS) is a recent technique for representing 3D scenes from multiple camera perspectives by projecting data onto a volumetric grid using Gaussian functions [47]. Recent advancements in sparse GS-based reconstruction, such as DNGaussian [48] and SparseGS [49], have enhanced sparse 3DGS through depth regularization. FewViewGS [50] proposes a multi-stage training scheme with pre-training and tuning to avoid overfitting. However, it struggles to render texture-rich regions accurately when novel views diverge significantly from the input views, primarily due to the absence of suitable appearance constraints. Moreover, sparse input views will result in insufficient optimization, potentially leading to collapsed reconstructions. For instance, GeCoNeRF [46], SparseNeRF [45], and DNGaussian [48] are sensitive to incorrect depth cues and may overlook subtle local depth variations. In this paper, we address these limitations by introducing the depth-aware continuity, the geometric-projected consistency, and the superpixel-aligned constraint, which effectively prevent inconsistencies across views and improve reconstruction fidelity in scenarios with sparse inputs.

Spectral Rendering: With the development of computing power, various rendering technologies are proposed to obtain photo-realistic images [51], [52], [53]. Spectral rendering is a more physically correct technique that indeed models a scene's light transport with real wavelengths [54]. Over the past decades, many physically-based spectral rendering methods are proposed, including stochastic sampling over the visible light [11], representing spectral information using Fourier coefficients [55], and sampling in the spatial domain [11]. These methods are designed for canonical ray-tracing rendering pipelines, which might be time-consuming when rendering scenes with complicated geometry. We consider the scene information along the ray and take advantage of NeRF-based architecture to combine the neural radiance field and spectral information to perform physically-based spectral rendering.

III. Preliminaries of Spectral Rendering

Given the CIE tristimulus values X, Y, and Z, the CIE color matching functions $f_X(\lambda)$, $f_Y(\lambda)$ and $f_Z(\lambda)$ involve the influence of light with wavelength λ to the three values [56], which were defined by measuring the mean color perception of a sample of human observers over the visual range from $\lambda_{\text{violet}} = 380$ to $\lambda_{\text{red}} = 780$ nanometer (nm). The following equation calculates the CIE X, Y, and Z values for light with wavelength λ:

$$\begin{cases} X = \kappa \sum f_X(\lambda) L(\lambda) \Delta\lambda \\ Y = \kappa \sum f_Y(\lambda) L(\lambda) \Delta\lambda \\ Z = \kappa \sum f_Z(\lambda) L(\lambda) \Delta\lambda, \end{cases} \tag{1}$$

where κ is a normalizing constant, $\sum$ represents the summation of visible light, $\Delta\lambda$ represents the sampling interval, and L denotes the spectral power distribution of the light source from the direction $(\theta_{\text{v}}, \varphi_{\text{v}})$ of observation x:

$$L(x, \theta_{\text{v}}, \varphi_{\text{v}}, \lambda) = \int_{\Omega} f_{\text{r}}(x, \theta, \varphi, \theta_{\text{v}}, \varphi_{\text{v}}, \lambda)\, R_{\text{i}}(x, \theta, \varphi, \lambda) \cos\theta \mathrm{d}\omega, \tag{2}$$

where R_{i} represents the radiance from direction (θ, φ) to point x, Ω is the hemispherical space on the surface where point x is located, f_{r} represents the bidirectional reflectance distribution function (BRDF), which is determined by the reflection characteristics of the material at point x, $\mathrm{d}\omega$ is a solid angle. Note that, (1) is calculated with the form of summation to estimate the original continuous integral.

To obtain a colorimetrically correct RGB image, the X, Y, and Z values are transformed to the sRGB color space using:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.133 & -1.616 & -0.490 \\ -0.978 & 1.916 & 0.033 \\ 0.072 & -0.229 & 1.405 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{3}$$

Due to the inconsistency of application environments, there are various XYZ → RGB conversion methods [10]. The matrix M^{c} listed here is computationally convenient.

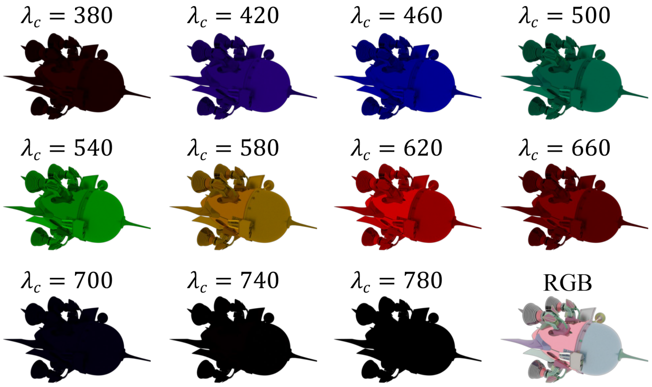


Fig. 2. The RGB spectrum maps of different wavelengths and the RGB image of white-light illumination.

Obtaining L from the rendering machine is difficult. In order to simplify the rendering pipeline of $L \rightarrow \text{XYZ} \rightarrow \text{RGB}$, we embed (3) to (1) to obtain the RGB spectrum maps corresponding to the wavelengths. Linearly combining the two equations to generate the RGB spectrum maps is reasonable on the one hand, and is simple yet effective on the other hand. The RGB spectrum maps corresponding to the wavelengths can be formulated as:

$$\begin{cases} R_\lambda = (M^c_{11} f_X(\lambda) + M^c_{12} f_Y(\lambda) + M^c_{13} f_Z(\lambda)) L(\lambda)\Delta\lambda \\ G_\lambda = (M^c_{21} f_X(\lambda) + M^c_{22} f_Y(\lambda) + M^c_{23} f_Z(\lambda)) L(\lambda)\Delta\lambda \\ B_\lambda = (M^c_{31} f_X(\lambda) + M^c_{32} f_Y(\lambda) + M^c_{33} f_Z(\lambda)) L(\lambda)\Delta\lambda. \end{cases} \quad (4)$$

Finally, the RGB spectrum maps are combined to generate the RGB image of white-light illumination:

$$\begin{cases} R = \kappa \sum R_\lambda \\ G = \kappa \sum G_\lambda \\ B = \kappa \sum B_\lambda. \end{cases} \quad (5)$$

Fig. 2 shows an example that includes 11 RGB spectrum maps and one RGB image rendered with our hypothesis by Mitsuba [57]. The λ_c in Fig. 2 represents the center of the sampling interval of spectral illuminants.

Furthermore, to verify the feasibility of the variant spectral rendering pipeline in (4) and (5), considering the spectral power difference and the physical theory, the least square (LSQ) method is utilized to obtain the adapted weights for spectrum maps of wavelength bands. Specifically, we acquire a linear fit of the RGB image with spectrum maps by solving an optimization problem: $\arg\min ||Aw - b||^2$, where A is the stack of vectors of flattened spectrum maps, w denotes the weight vector to be solved, and b represents the vector of flattened target RGB image. Then, the linear fit result C_{lsq} can be calculated through $C_{\text{lsq}} = \sum_i^{\lambda_{\text{total}}} w_i \cdot \mathbf{S}_{\lambda_i}$, where $i \in \{1, 2, \ldots, \lambda_{\text{total}}\}$ and λ_{total} denotes the number of spectrum maps, $\mathbf{S}_{\lambda_i}$ represents the i^{th} spectrum map.

The estimated weights curve should be approximately proportional to the spectral power distribution $L(\lambda)$ of D65 except for bands near 380 nm and 780 nm which tend to be black. Corresponding curves are shown in Fig. 3. The almost identical tendency of the spectral weights and the spectral power distribution demonstrates the rationality of the proposed variant spectral rendering pipeline. Fig. 4 further presents the visual results of the linear weighted sum of spectrum maps and the ground truth images. We can find that the least square method is almost right. Even though there is a tiny color difference on the scraper of the digger (red boxes), the linear weighted sum is a decent approximation to the image rendered with white-light illumination, which proves the correctness of our variant form of spectral rendering.

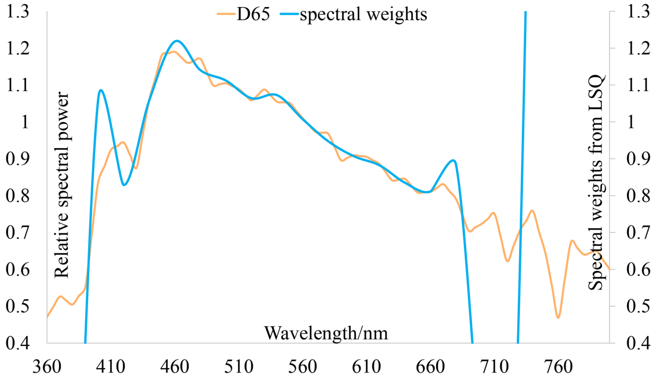


Fig. 3. The correlations between the spectral power distribution of D65 and the spectral weights.

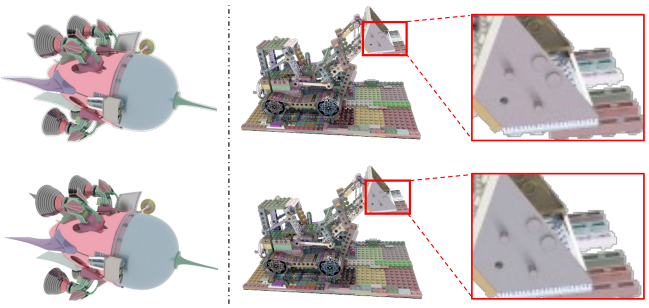


Fig. 4. The first row shows the results of the least square method and the second row is the rendered ground truth. They are similar, indicating that the variant spectral rendering pipeline is reasonable and effective.

IV. Method

We propose an end-to-end NeRF-based architecture to achieve the physically-based spectral rendering with sparse inputs. As shown in Fig. 5, the baseline includes two modules operating the RGB spectrum map rendering (Fig. 5(a)) and the spectrum fusion (Fig. 5(c)). The first module is an MLP-based network to produce spectrum maps according to the given ray origin $\mathbf{o}$ and ray direction $\mathbf{d}$. The second module fuses the discrete spectrum maps to obtain an RGB image of white-light illumination, which applies the attention mechanism to better extract useful information from spectrum maps to approach the integral calculation in spectral rendering. Note that, we generate s_{num} discrete spectrum maps by uniform sampling through the visible light $380nm$–$780nm$. The overall color of the generated spectrum maps conforms to the distribution of the CIE color matching function in Fig. 5(b). Both spectral rendering and NeRF-based methods will be improved through the proposed pipeline. As for sparse input views, three complementary constraints in Fig. 5(e), (f), and (g) are designed in

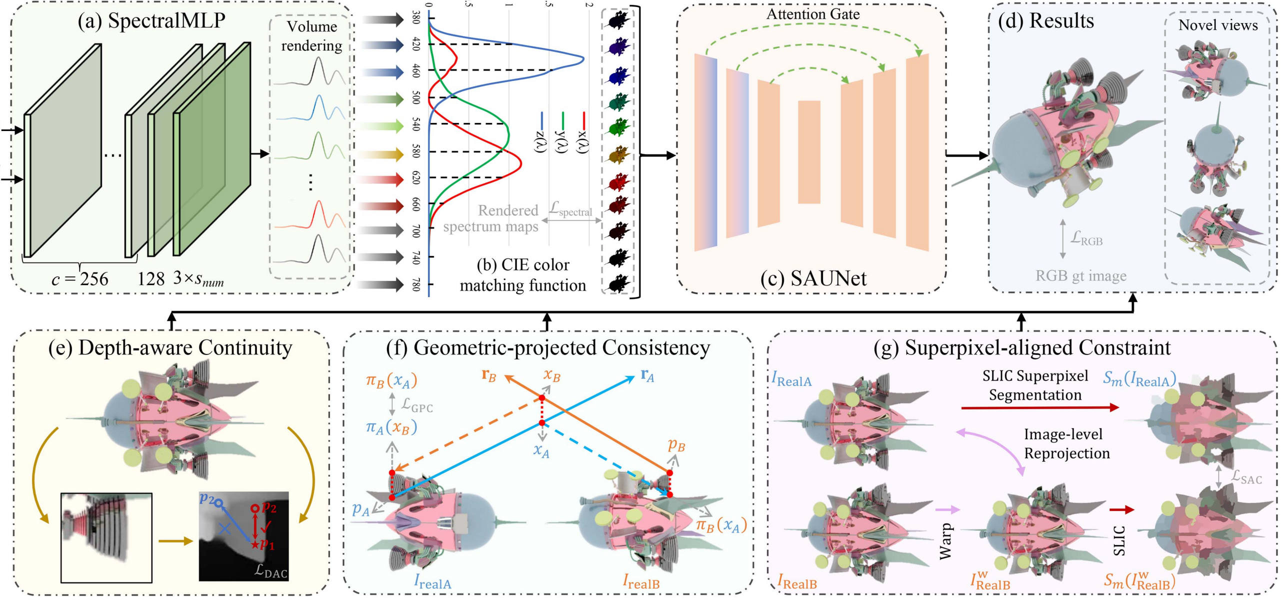


Fig. 5. An overview of the main architecture. We first design (a) SpectralMLP to construct the spectral radiance field and generate s_{num} RGB spectrum maps with novel views using volume rendering. The generated spectrum maps are constrained by the rendered spectral images. The color of the spectrum maps matches the distribution of the CIE color matching function in (b). The (c) SAUNet combines the discrete spectrum maps to produce high-quality RGB outputs, constrained by the rendered RGB images. As for sparse inputs, we design the (e) depth-aware continuity to enforce spatial smoothness with a single-view, the (f) geometric-projected consistency to ensure the multi-view geometric coherence, and the (g) superpixel-aligned constraint to preserve texture and color integrity. $\mathbf{o}$ is the ray origin, $\mathbf{d}$ is the ray direction and c represents the channels of different layers in SpectralMLP.

SS-NeRF to improve the reconstruction quality under limited supervision.

A. Spectral Radiance Field

We represent the scene as spectral radiance fields within bounded 3D volumes. For a given ray origin $\mathbf{o} = (x, y, z)$ and ray direction $\mathbf{d} = (\theta, \phi)$, we propose the SpectralMLP $F_{\mathbf{\Theta}}$ to generate the spectral radiance s_{λ_i} and the density σ of the ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$. SpectralMLP $F_{\mathbf{\Theta}}$ achieves the mapping from $(\mathbf{o}, \mathbf{d})$ to (s_{λ_i}, σ), which is defined as:

$$(\mathbf{s}_{\lambda_i}, \sigma) = F_{\mathbf{\Theta}}\left(\gamma(\mathbf{o}), \gamma(\mathbf{d})\right), \tag{6}$$

where $i \in \{1, 2, \ldots, s_{\text{num}}\}$, γ represents the positional encoding [58] that maps the inputs into higher dimensional frequency space, which is applied separately to each of the three coordinate values in $\mathbf{x}$ and to the three components of the direction unit vector $\mathbf{d}$. The s_{num} is set to 11 to achieve the balance between performance and efficiency.

The SpectralMLP finally outputs s_{num} spectral radiance and one σ value. As for different wavelength λ, the density of each point is same, but the spectral radiance is different. Volume rendering [59] is then applied to render the spectral radiance s_{λ_i} of each ray passing through the scene. The spectrum value $\widehat{\mathbf{S}}_{\lambda_i}(\mathbf{r})$ of ray $\mathbf{r}(t)$ is computed as:

$$\widehat{\mathbf{S}}_{\lambda_i}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{s}_{\lambda_i}(\mathbf{r}(t), \mathbf{d})dt, \tag{7}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(p))dp)$, t denotes a position along the ray, t_n and t_f are the near and far boundary.

B. SAUNet

Theoretically, according to (5), directly combining the spectrum maps of every wavelength among the visible light is practicable. However, in our implementation, such an operation is insufficient because the spectral datasets are extremely sparse. Therefore, we propose the spectrum attention UNet (SAUNet) (Fig. 6(a)) to learn the correlations of spectrum maps and generate high-quality RGB outputs. Applying the SAUNet can imitate the original integral operation better and produce results close to the ground truth. We first introduce the attention gate [60] (Fig. 6(c)) to refine features from the encoder. The high-level features before outputting to the next decoder stage are guided by the low-level features from the encoder using the attention mechanism. We further design a spectrum attention (SA) module to better explore the correlations of spectrum maps.

The standard residual convolutional network is insufficient for extracting the spectral dependencies [61]. We modify the standard residual blocks and introduce the SA module (Fig. 6(b)) to better combine the discrete spectrum maps to make the results close to the integral calculation in spectral rendering. Specifically, three 1×1 convolutional blocks are first used to reorganize and reweight the importance of spectrum maps. The channel attention (CA) [62] is then introduced to focus on inter-spectral feature fusion by the attention mechanism in the channel dimension.

SAUNet contains 3 encoders and 3 decoders. Skip connections with attention gates pass feature maps from each encoder to decoder. Feature maps from low-levels contain more detailed information of spectrum maps. The SA module is placed in the first two encoders to best use the spectral information because

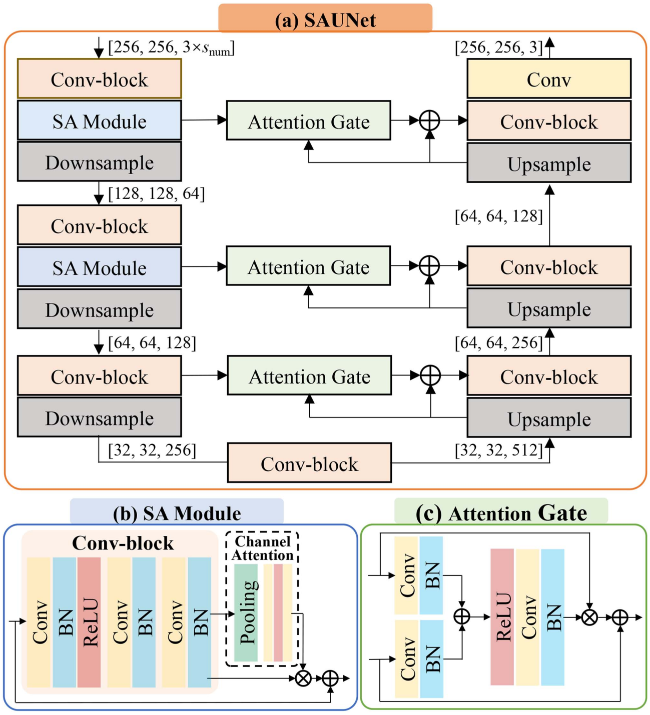


Fig. 6. The detailed architecture of SAUNet. The SA module is applied to the first two encoders to better explore the correlations between spectrum maps.

higher-level features are more abstract, which may affect the ability of the network to explore the correlations. The fusion process is defined as:

$$\widehat{C} = \text{SAUNet}\left(\widehat{\mathbf{S}}_{\lambda_i}\right), \tag{8}$$

where $\widehat{C}$ represents the final output RGB image.

C. Depth-Aware Continuity

Precise geometry facilitates both spectral and color prediction in the radiance fields. The depth-aware continuity first uses the depth maps predicted from a pre-trained monocular depth estimator as supervision to encourage geometric accuracy within the single-view. Specifically, the pre-trained dense prediction Transformer (DPT) is used to generate the monocular depth map $\boldsymbol{D}$ for each training view [28]. Subsequently, to alleviate the constraint posed by inconsistencies in absolute depth values, a softened depth constraint based on Pearson correlation [63] is introduced, which mitigates the scale ambiguity between the rendered depth $\boldsymbol{D}_{\text{ren}}$ and the estimated depth $\boldsymbol{D}_{\text{est}}$:

$$\mathcal{L}^{\text{s}}_{\text{DAC}} = \frac{\text{Cov}\left(\boldsymbol{D}_{\text{ren}}, \boldsymbol{D}_{\text{est}}\right)}{\sqrt{\text{Var}\left(\boldsymbol{D}_{\text{ren}}\right)\text{Var}\left(\boldsymbol{D}_{\text{est}}\right)}}. \tag{9}$$

The monocular depth constraint guarantees that the predicted depth map estimated by NeRF is consistent with the depth map form DPT. However, it does not enforce spatial continuity of the depth map. Therefore, we incorporate spatial continuity priors from the depth model DPT. If two neighboring pixels exhibit similar depth values in the DPT output, we constrain their corresponding depth values predicted by NeRF to also be continuous, as shown in Fig. 5(e). The spatial continuity is given by:

$$\mathcal{L}^{\text{c}}_{\text{DAC}} = \sum_{p_1} \sum_{\boldsymbol{D}^{p_2}_{\text{est}} \in \text{KNN}\left(\boldsymbol{D}^{p_1}_{\text{est}}\right)} \max\left(\left|\boldsymbol{D}^{p_1}_{\text{ren}} - \boldsymbol{D}^{p_2}_{\text{ren}}\right| - m, 0\right), \tag{10}$$

where p_1 and p_2 are randomly sampled depth pixels, m is a small margin that tolerates minor variations in neighboring depths. KNN$(\cdot)$ returns k-nearest neighbors measured by depth values within a superpixel. In our implementation, we set m to 1.0×10^{-4}.

The depth-aware continuity combines the single-view depth supervision and spatial continuity term:

$$\mathcal{L}_{\text{DAC}} = L^{\text{s}}_{\text{DAC}} + \mathcal{L}^{\text{c}}_{\text{DAC}}, \tag{11}$$

where $L^{\text{s}}_{\text{DAC}}$ enforces the depth accuracy, and $L^{\text{c}}_{\text{DAC}}$ promotes local smoothness guided by spatial priors.

D. Geometric-Projected Consistency

The geometric-projected consistency is introduced to ensure geometric accuracy by quantifying the discrepancy between corresponding rays. Ideally, the ray $\mathbf{r}_A$ and $\mathbf{r}_B$ should intersect in the 3D space if the camera parameters are correctly calibrated. However, misalignment errors will result in a measurable distance between the rays, as shown in Fig. 5(f). Let x_A represent a point in 3D space along ray $\mathbf{r}_A$, parameterized as: $x_A(t_A) = \mathbf{o_A} + t_A\mathbf{d_A}$, and x_B represent a corresponding point in 3D space along ray $\mathbf{r}_B$, parameterized as: $x_B(t_B) = \mathbf{o_B} + t_B\mathbf{d_B}$. The shortest distance between these two rays can be expressed as:

$$d = \frac{|(\mathbf{o_B} + t_B\mathbf{d_B} - \mathbf{o_A}) \times \mathbf{d_A}|}{||\mathbf{d_A}||^2}. \tag{12}$$

To calculate a normalized distance that reflects geometric consistency in the image planes, we project the 3D points x_A and x_B onto images I_{realA} and I_{realB}. This accounts for the correspondence of the projected points to their respective image pixels p_A and p_B. The projected distance is defined as:

$$\mathcal{L}_{\text{GPC}} = \frac{\|\pi_A(x_B) - p_A\| + \|\pi_B(x_A) - p_B\|}{2}, \tag{13}$$

where $\pi(\cdot)$ represents a projection function that ensures equal contribution from each correspondence, regardless of their distance from the cameras. A smaller $\mathcal{L}_{\text{GPC}}$ implies better alignment between the 3D points and their image correspondences, indicating higher geometric consistency across views.

E. Superpixel-Aligned Constraint

In addition to the geometric-projected consistency, we introduce the superpixel-aligned constraint to ensure appearance consistency across different perspectives. This constraint addresses the challenges posed by subtle color variations between camera views. The process begins with a rough alignment step to mitigate perspective inconsistencies, where the real view I_{realB} is warped to align with I_{realA}. This initial alignment, which serves as a foundation for further refinement, is achieved by using SuperPoint [64] for feature extraction, LightGlue [65] for feature matching, and RANSAC [66] for outliers rejection. Then, we minimize the color differences between warped real view $I^{\text{w}}_{\text{realB}}$ and the real view I_{realA}.

While this rough alignment helps reduce large-scale inconsistencies, directly constraining pixel-wise differences is unreasonable due to minor color variations between different views [14]. To address this, we employ superpixel-aligned segmentation as a preprocessing step for both the warped view $I^{\text{w}}_{\text{realB}}$ and the reference view I_{realA}, as illustrated in Fig. 5(g). Specifically, we utilize the Simple Linear Iterative Clustering (SLIC) algorithm [67], which segments images into uniformly sized regions, with uniform color, brightness, and texture. The superpixel-aligned constraint calculates the differences of the average values among superpixel region S_m:

$$\mathcal{L}_{\text{SAC}} = \frac{1}{M}\sum_{m=1}^{M} ||\text{avg}\left(S_m\left(I^{\text{w}}_{\text{realB}}\right)\right) - \text{avg}\left(S_m\left(I_{\text{realA}}\right)\right)||_2^2, \tag{14}$$

where M denotes the total number of superpixels. In our implementation, we segment images into 400 superpixels to achieve a balance between computational efficiency and spatial coherence. Unlike GeCoNeRF [46], which operates at a regular patch-wise level, our superpixel-based constraint offers a more compact and perceptually aligned representation of the appearance distribution by enforcing consistency at the irregular region level. This formulation better captures semantic boundaries and reduces color discrepancies across views. An ablation study of comparing the superpixel-based constraint with a patch-based alternative is provided in Section VI-C. In addition, by focusing on region-level features rather than individual pixel values, the superpixel-aligned constraint enforces the rationality of appearance alignment.

F. Optimization

The objective function includes the following five items: 1) the weighted spectrum map reconstruction loss $\mathcal{L}_{\text{spectral}}$, which pushes the SpectralMLP to produce desired spectrum maps; 2) the RGB reconstruction loss $\mathcal{L}_{\text{RGB}}$, which optimizes the SAUNet to generate high-quality RGB images; 3) the depth-aware continuity loss $\mathcal{L}_{\text{DAC}}$, which guarantees that the rendered depth map remains consistent with the predicted depth map, while also preserving spatial smoothness within the rendered depth map; 4) the geometric-projected consistency $\mathcal{L}_{\text{GPC}}$, which minimizes the discrepancies in 3D geometric consistency; and 5) the superpixel-aligned constraint $\mathcal{L}_{\text{SAC}}$, which enhances the appearance coherence between different views. The full objective function is described as:

$$\mathcal{L} = \mathcal{L}_{\text{spectral}} + \lambda_{\text{RGB}}\mathcal{L}_{\text{RGB}} + \lambda_{\text{DAC}}\mathcal{L}_{\text{DAC}} + \lambda_{\text{GPC}}\mathcal{L}_{\text{GPC}} + \lambda_{\text{SAC}}\mathcal{L}_{\text{SAC}}, \tag{15}$$

where λ_{RGB}, λ_{DAC}, λ_{GPC}, and λ_{SAC} are hyper-parameters to balance the contributions of the different components. In our implementation, we empirically set $\lambda_{\text{RGB}} = 1.1$, $\lambda_{\text{DAC}} = 0.02$, $\lambda_{\text{GPC}} = 0.01$ and $\lambda_{\text{SAC}} = 0.1$.

Weighted Spectrum Map Reconstruction Loss: We found the image quality of generated spectrum maps is different, with its power concentrated in wavelength near 380nm and 780nm tends to be black, resulting in higher PSNR scores and closer $\mathcal{L}_1$ distance. Therefore, we propose the weighted spectral reconstruction loss $\mathcal{L}_{\text{spectral}}$ to better acquire useful information from more informative spectrum maps distributed in the middle of visible light. Similar to NeRF, we simultaneously optimize a coarse model and a fine model. The weighted spectrum map reconstruction loss is defined as:

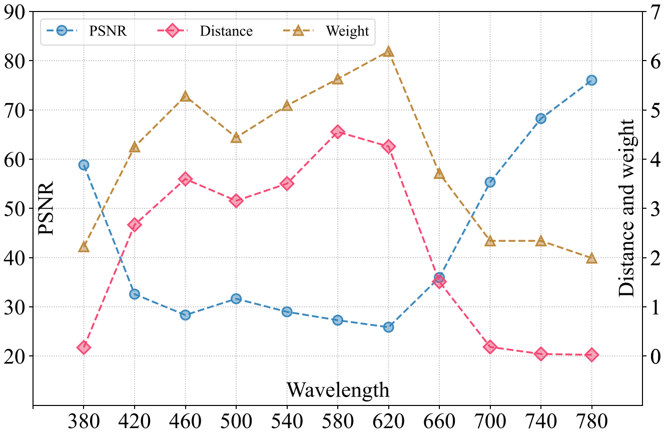


Fig. 7. We increase the weight in the middle range of the wavelengths while decreasing it at two ends for the compensation of SpectralMLP.

$$\mathcal{L}_{\text{spectral}} = \sum_{i}^{s_{\text{num}}} w_s \cdot \sum_{\mathbf{r}\in\mathcal{R}(\mathbf{P})} \Big(||\widehat{\mathbf{S}}^c_{\lambda_i}(\mathbf{r}) - \mathbf{S}_{\lambda_i}(\mathbf{r})||_2^2 + ||\widehat{\mathbf{S}}^f_{\lambda_i}(\mathbf{r}) - \mathbf{S}_{\lambda_i}(\mathbf{r})||_2^2 \Big), \tag{16}$$

where s_{num} is the number of spectrum maps, $\mathbf{S}_{\lambda_i}$ represents the spectrum maps, $\mathcal{R}(\mathbf{P})$ is a set of camera rays at target position $\mathbf{P}$, $\widehat{\mathbf{S}}^c$ and $\widehat{\mathbf{S}}^f$ represent the spectrum maps generated in the coarse stage and fine stage, and w_s are the weights that are correlated to the PSNR scores of spectrum maps:

$$w_s = 2^{P_{\text{max}}/P_\lambda}, \tag{17}$$

where P_{max} is the maximum PSNR value of s_{num} spectrum maps, and P_λ is the PSNR value of wavelength λ. Fig. 7 illustrates the distributions of the PSNR, the $\mathcal{L}_1$ pixel distance, and the weights. We give more penalties to the SpectralMLP network if it cannot satisfactorily reconstruct these informative spectrum maps distributed in the middle of the visible light.

RGB Reconstruction Loss: The RGB reconstruction loss $\mathcal{L}_{\text{RGB}}$ minimizes the difference between the predicted RGB image $\widehat{C}$ and the rendered RGB ground truth C. The $\mathcal{L}_2$ distance is adopted as the loss function, written as:

$$\mathcal{L}_{\text{RGB}} = ||\widehat{C} - C||_2^2. \tag{18}$$

V. Datasets and Implementations

A. Datasets

Spectral Illumination: We render our synthetic scenes with multiple spectral illuminants to generate spectral images. To acquire spectral illuminants, we divide the wavelength range of the light source spectrum in Mitsuba [57] from 360 nm to

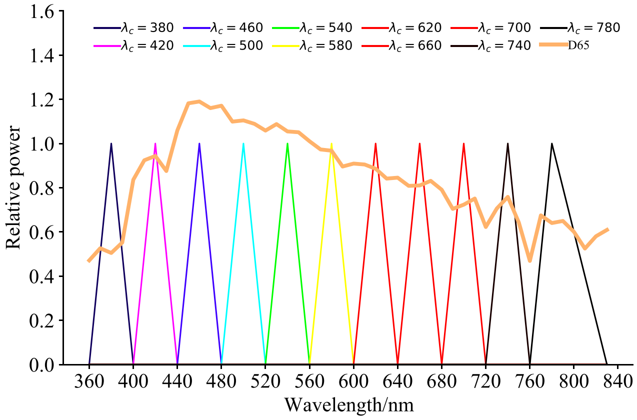


Fig. 8. The spectral power distribution of CIE standard illuminant D65 and 11 spectral illuminants. λ_c represents the midpoint of each interval.

830 nm into 11 adjacent intervals. As for the first 10 spectral intervals, the relative power at the midpoint λ_c is set to a certain value. The maximum power of the last interval (760 nm to 830 nm) is set at 780 nm because 780 nm is a commonly used termination value of visible light. Simultaneously, the power at two endpoints for each interval is set to 0. After a simple linear interpolation, we obtain 11 curves as the spectral power distributions of the spectral illuminants. In addition, we use the CIE standard illuminant D65 as the default white light source for scenes in our dataset. The D65 light source is an artificial light source that simulates daylight, and its emission spectrum conforms to the average midday light of European and Pacific countries. The spectral power distributions of the 11 spectral illuminants and the D65 is illustrated in Fig. 8.

The Spectral Reflectance and the Camera Response: Spectral rendering refers to a physically correct modeling approach for a scene's light transport process, which constructs light source emission and object material reflectance with real wavelengths, just the same as how the color works in the real world. We choose Mitsuba as the renderer to synthesize our spectral datasets because Mitsuba is a research-oriented rendering system with specific support for spectral rendering. Other common traditional renderers, such as Eevee and Cycles in Blender [68], render a scene with simply red, green, and blue components, whose rendering pipeline is a crude approximation of the physical color system. According to (2), when rendering our spectral datasets, some materials are assigned with the generated random spectral responses and the bidirectional scattering distribution function (BSDF) of the plastics and the metals. Other materials are physically-based rendering image textures and assigned with plausible smooth spectral responses corresponding to each color using spectral upsampling algorithms in Mitsuba. The camera sensor response is fixed as the normal RGB sensitivity curves. The measured spectral power distributions are converted to the linear RGB values based on the CIE 1931 XYZ color matching curves. Then, the spectrum maps of multiple wavelengths are rendered by ray-tracing rendering algorithms in Mitsuba while changing the spectral power distribution of light sources.

Synthetic Scenes: As shown in Fig. 9, we first render 8 scenes with models located in the middle of the field, among which 4 scenes are designed without background and their viewpoints are captured on a sphere surrounding the models (the first two columns of Fig. 9). One case with different wavelengths is shown in the top row of Fig. 10. The following 4 scenes use texture to construct the walls and floors, and their viewpoints are sampled on the upper hemisphere. The second row of Fig. 10 shows one such example across different wavelengths. Then, we render 2 indoor forward-facing scenes in the fifth column, where the camera viewpoints are spatially constrained and limited in perspective. The example is presented in the third row of Fig. 10.

Real-world Scenes: We capture 10 forward-facing real datasets in a sealed room (the last five columns of Fig. 9) using a camera and 8 color absorbers whose center wavelengths range from $400nm$ to $750nm$ with the interval of $50nm$. Different color absorbers are covered to the camera lens to obtain the spectral images. Their camera poses are generated using the COLMAP [69]. The last row of Fig. 10 presents the captured wavelengths of one real-world scene.

B. Implementation Details

We implement the SpectralMLP on top of NeRF [14], which uses an eight-layer MLP with 256 channels and ReLU activation to predict the density σ, and following two fully-connected layers with 128 and $3 \times s_{\text{num}}$ channels to obtain the spectral radiance. We use a batch of 2048 rays, and sample 64 points along each ray in the coarse model and 128 points in the fine model, respectively. We use PyTorch [70] to implement our model and train it using the Adam optimizer [71] with learning rate of 5×10^{-4} and 0.001 for the SpectralMLP and the SAUNet, respectively. Other Adam hyperparameters are left at default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 0.7$.

Similar to general sparse methods, we use 8 training views for 360° scenes designed without background, 24 training views for 360° scenes with background, and 6 training views for forward-facing scenes. DPT Hybrid [28] model is selected as the monocular depth estimator for the depth-aware continuity. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU.

VI. Experiments

We conduct comprehensive experiments to verify the effectiveness of our methods. Specifically, we compare SS-NeRF with: 1) FreeNeRF [72], a Frequency regularized method; 2) SparseNeRF [45], which exploits depth priors from real-world inaccurate observations; and 3) DNGaussian [48], a recent 3DGS-based sparse method.

A. Quantitative Comparisons

We report quantitative performance using PSNR (higher is better), SSIM (higher is better), as well as LPIPS [73] (lower is better). Table I shows the comparison results of SS-NeRF with several representative NeRF-based and GS-based methods. NeRF-based baselines, although yielding relatively low errors in certain regions, often result in overly smooth synthesized views. In contrast, GS-based methods generally perform well on synthetic datasets but often struggle to accurately reconstruct

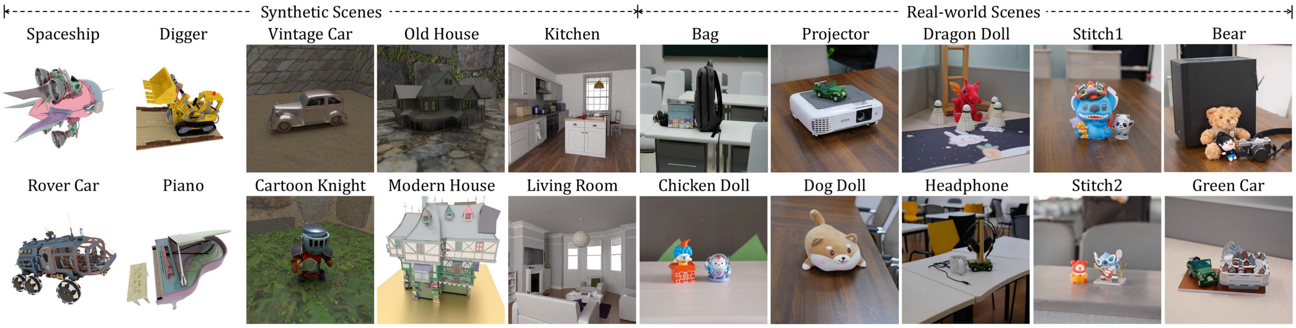


Fig. 9. Scenes used for our datasets. The 20 scenes include 10 synthetic scenes and 10 real-world scenes. The synthetic scenes consist of both 360° scenes without and with background, as well as forward-facing scenes.

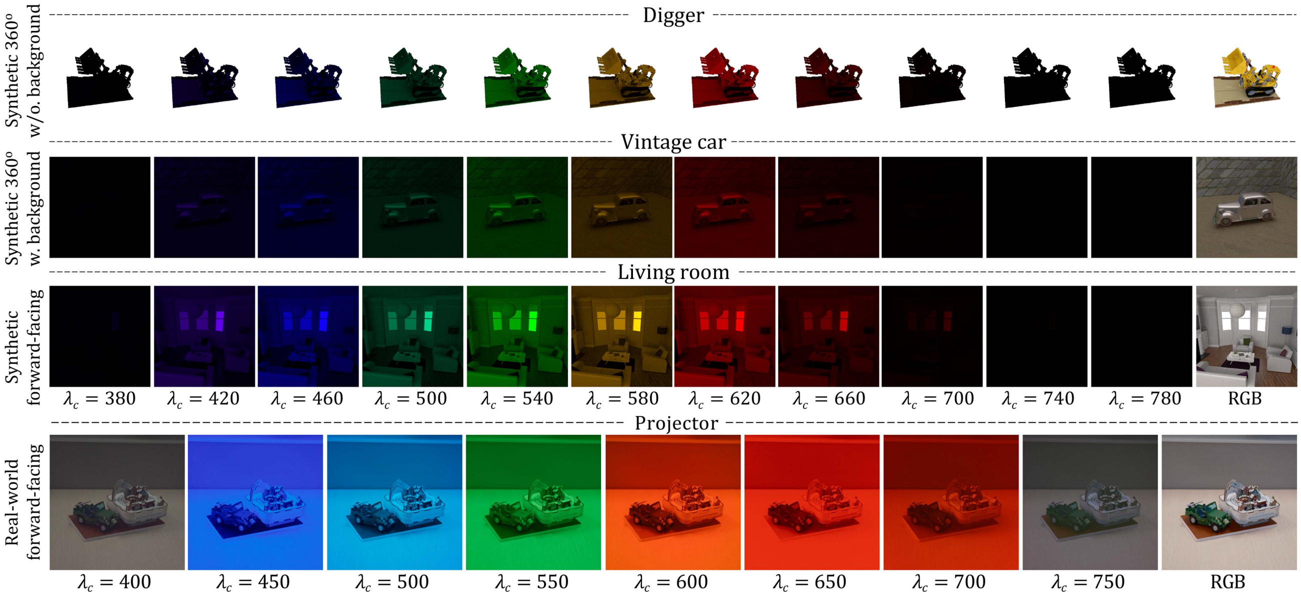


Fig. 10. Our datasets with different wavelengths.

complex geometries in real-world scenes, frequently producing blurred or ambiguous outputs. By incorporating the designed mechanisms, SS-NeRF achieves superior average scores, excelling in both geometric fidelity and detailed appearance reconstruction. The results demonstrate its ability to produce high-quality outputs and maintain consistency across different viewpoints in sparse input scenarios.

B. Qualitative Comparisons

Fig. 11 shows the visual comparisons with representative sparse reconstruction methods. The FreeNeRF [72], while differing less from ground truth in the error maps, often interpolates colors to unknown viewpoints, generating smooth synthesized views that lack high-frequency details. DNGaussian [48] struggles to maintain the consistency of the geometry at the edges or other unobserved angles, leading to significant differences compared to the real image. In contrast, the proposed SS-NeRF effectively addresses these challenges and captures fine details.

We then present several generated spectrum maps in Fig. 12. Fig. 13 further provides comparisons in the CIE 1931 xy chromaticity space, which offers a perceptually meaningful view of color reproduction accuracy. The synthesized RGB images are first linearized and then converted to the CIE XYZ color space using the standard sRGB → XYZ transformation [74], followed by projection onto the 2D xy chromaticity plane. As shown in Fig. 13, the RGB-based methods exhibit deviations from the ground truth, especially in regions with complex materials. In contrast, our method achieves xy distributions that are more tightly clustered around the ground truth.

C. Ablation Studies

The effectiveness of Main Components: We conduct ablations on several components to understand how these modules work, including s_{num}, attention gate, and SA module. The results are shown in Table II. First, as shown in Table II(a) and (b), introducing the spectral radiance fields can effectively improve the performance. Second, as shown in Table II(f) and (g), removing the attention gate (AG) will degrade the results. Third, Table II(c), (d), (e), and (g) show the results when embedding the SA module to different encoder blocks. The SA module is

TABLE I
QUANTITATIVE COMPARISONS OF SS-NERF WITH OTHER NERF-BASED AND GS-BASED SPARSE METHODS IN TERMS OF PSNR, SSIM, AND LPIPS. **ORANGE** INDICATES THE BEST PERFORMANCE AND **YELLOW** REFERS TO THE SECOND BEST RESULT

	Spaceship			Rover Car			Digger			Piano		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	24.302	0.8832	0.0982	23.801	0.8803	0.0994	25.528	0.8952	0.0901	24.729	0.8859	0.0925
SparseNeRF	24.011	0.8770	0.1131	23.479	0.8711	0.1216	25.196	0.8913	0.1015	24.210	0.8790	0.1072
DNGaussian	24.336	0.8867	0.0886	23.715	0.8765	0.0957	25.605	0.8970	0.0852	24.725	0.8891	0.0868
SS-NeRF	24.861	0.8973	0.0855	24.048	0.8866	0.0898	26.113	0.9095	0.0826	24.914	0.8917	0.0872

	Vintage Car			Cartoon Knight			Old House			Modern House		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	24.855	0.7093	0.3058	24.896	0.7132	0.2987	21.755	0.6823	0.3301	23.281	0.7480	0.2234
SparseNeRF	25.024	0.7273	0.2952	24.973	0.7268	0.2928	21.896	0.6856	0.3285	23.356	0.7495	0.2209
DNGaussian	25.569	0.7725	0.2187	25.893	0.7753	0.2102	22.991	0.7312	0.2445	24.158	0.7556	0.2182
SS-NeRF	25.787	0.7784	0.2057	25.968	0.7716	0.2025	22.880	0.7347	0.2325	24.240	0.7562	0.2192

	Kitchen			Living Room			Bag			Chicken Doll		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	24.735	0.8277	0.1884	24.523	0.8323	0.1968	23.899	0.7639	0.2004	23.590	0.7468	0.2125
SparseNeRF	24.881	0.8342	0.1756	24.715	0.8229	0.1823	24.083	0.7650	0.2009	23.712	0.7455	0.2019
DNGaussian	23.881	0.7889	0.2045	23.291	0.7932	0.2127	23.910	0.7528	0.2181	23.939	0.7470	0.2101
SS-NeRF	25.576	0.8562	0.1631	25.260	0.8527	0.1727	24.516	0.7714	0.1933	23.935	0.7521	0.1974

	Projector			Dog Doll			Dragon Doll			Headphone		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	23.832	0.7792	0.1754	19.655	0.6119	0.2228	24.735	0.8234	0.1587	24.191	0.8067	0.1712
SparseNeRF	23.758	0.7846	0.2021	19.876	0.6243	0.2495	24.672	0.8191	0.1654	24.025	0.8024	0.1768
DNGaussian	22.589	0.7657	0.1985	19.137	0.5925	0.2544	23.171	0.7896	0.1789	23.061	0.7713	0.1893
SS-NeRF	24.379	0.8249	0.1763	20.415	0.7116	0.2080	25.168	0.8319	0.1399	24.575	0.8142	0.1521

	Stitch1			Stitch2			Bear			Green Car		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FreeNeRF	23.517	0.7953	0.1819	23.767	0.7996	0.1701	23.669	0.8001	0.1838	24.893	0.8186	0.1636
SparseNeRF	23.684	0.7967	0.1882	24.165	0.8105	0.1621	23.651	0.8007	0.1816	24.919	0.8181	0.1528
DNGaussian	23.226	0.7899	0.1994	23.385	0.8023	0.1615	22.912	0.7926	0.1957	23.129	0.8052	0.1687
SS-NeRF	24.096	0.8054	0.1775	24.394	0.8070	0.1598	23.855	0.8035	0.1825	25.206	0.8221	0.1472

TABLE II
ABLATION STUDIES OF DIFFERENT COMPONENTS. AG IS THE ATTENTION GATE. SA IS THE SPECTRUM ATTENTION MODULE. E1, E2, AND E3 ARE THE FIRST, SECOND, AND THIRD ENCODER BLOCKS

	s_{num}	AG	SA	PSNR↑	SSIM↑	LPIPS↓
(a)	0			23.317	0.8265	0.2014
(b)	11			24.578	0.8343	0.1809
(c)	11	✓		24.692	0.8357	0.1688
(d)	11	✓	E1	24.755	0.8359	0.1691
(e)	11	✓	E1+E2+E3	24.930	0.8360	0.1690
(f)	11		E1+E2	25.079	0.8389	0.1572
(g)	11	✓	E1+E2	25.151	0.8392	0.1577

TABLE III
PERFORMANCE WITH DIFFERENT CONFIGURATIONS IN SS-NERF

Methods	DAC	GPC	SAC	w_s	PSNR ↑	SSIM ↑	LPIPS ↓
(a)				✓	24.147	0.8219	0.1943
(b)	✓			✓	24.252	0.8263	0.1941
(c)		✓		✓	24.944	0.8315	0.1807
(d)			✓	✓	25.112	0.8341	0.1691
(e)	✓	✓	✓		24.993	0.8332	0.1715
(f)		✓	✓	✓	25.129	0.8388	0.1577
(g)	✓	✓	✓	✓	25.151	0.8392	0.1577

placed in the first two encoders to best explore the correlations of spectrum maps.

The Impact of DAC, GPC, SAC, and w_s: We further perform ablation studies to validate the effectiveness of the key modules in SS-NeRF, including the Depth-Aware Continuity (DAC), the Geometric-Projected Consistency (GPC), the Superpixel-Aligned Constraint (SAC), and the spectral weights w_s in (16). First, results in Table III(a) and (b) demonstrate the effectiveness of DAC in improving depth prediction and reconstruction quality. Second, Table III(a) and (c) show that incorporating GPC leads to better structural accuracy by enhancing alignment between the predicted images and the ground truth. Furthermore, Table III(a) and (d) highlight the importance of the SAC in preserving appearance fidelity across views, maintaining the visual integrity of textures and colors. Their complementary contributions in Table III(e) and (f) enable high-quality reconstructions and enhance the performance of SS-NeRF. Finally, as shown in Table III(e) and (g), removing the weights w_s in (16) also degrades the performance of novel view synthesis. This confirms that emphasizing the contribution of perceptually informative spectrum bands can learn more meaningful representations.

The Influence of Different Materials: We also conduct the ablation study when selecting different materials to construct the 3D models. Fig. 14 shows the Digger model with different materials, among which the first one is the original model, the second one applies the PNG decals to the objects, and the

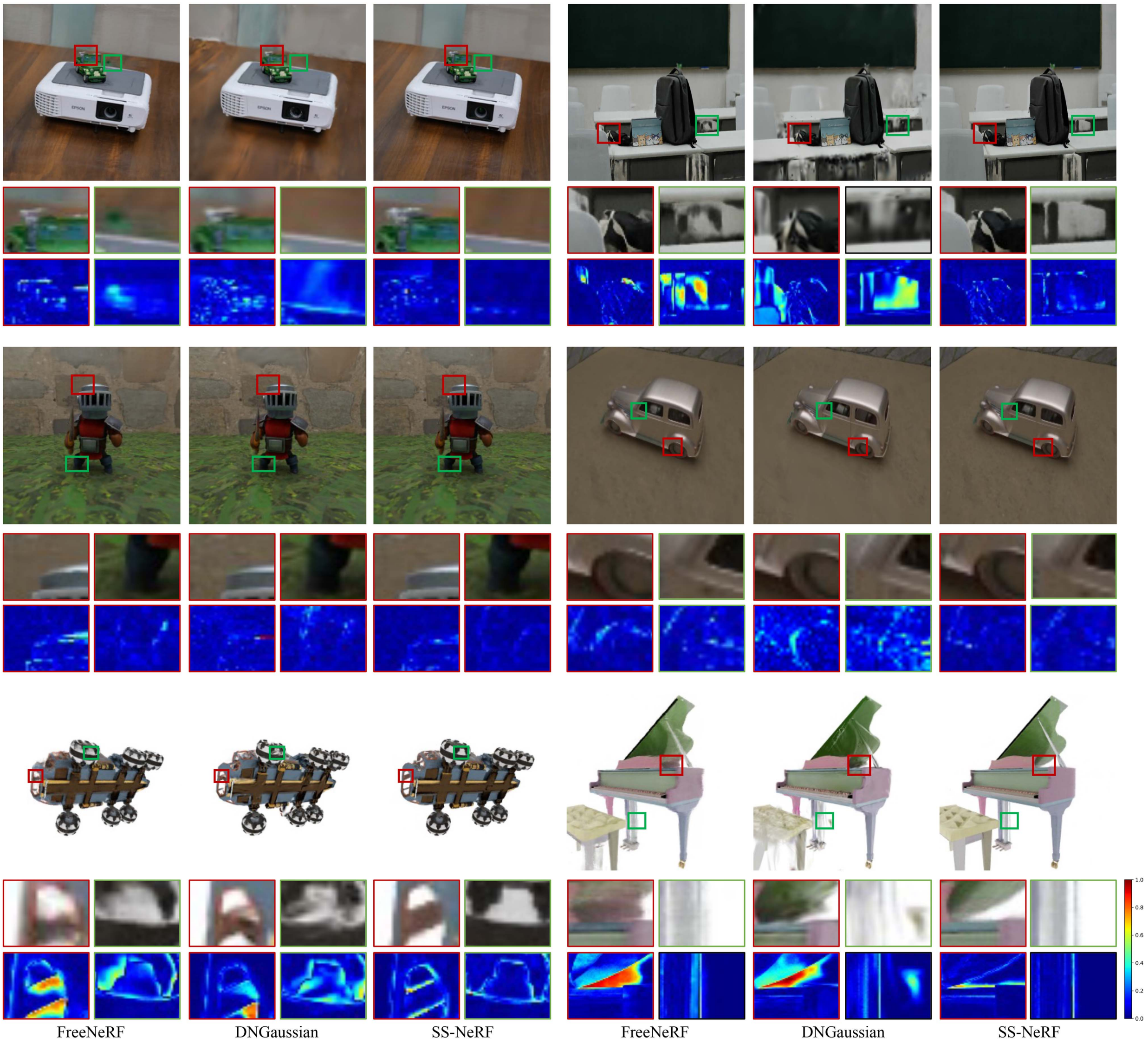


Fig. 11. Comparisons between SS-NeRF and sparse scene reconstruction methods on real-world and synthetic datasets. The top two cases are real-world scenes, the middle two cases are 360° scenes with background, and the bottom two cases are 360° scenes without background.

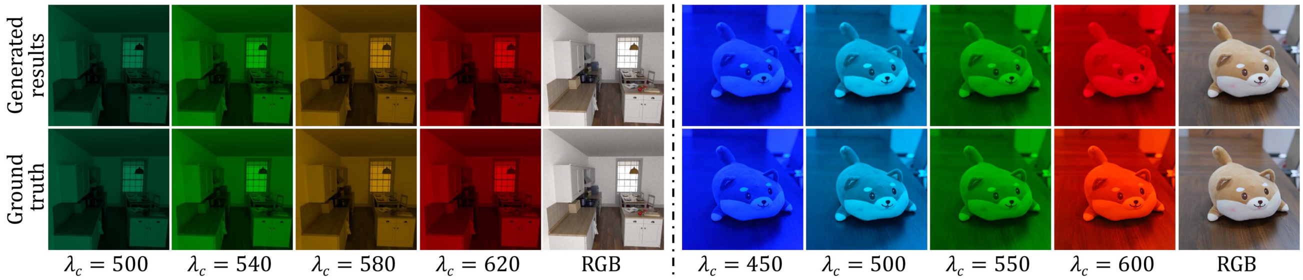
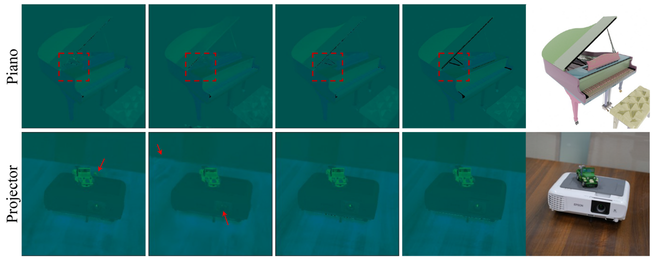


Fig. 12. Comparisons between the generated spectrum maps and the ground truth for the synthetic dataset (left) and the real-world dataset (right), illustrating the high fidelity of spectral reconstruction.

last one indicates that we randomly assign spectral materials to objects, which will greatly increase the complexity of 3D models. Table IV shows the quantitative performance. The results of the 3D model with PNG decals and the mixed materials are obviously inferior to the default material. The performance of mixed materials is lower than the PNG decals because the reflectance of PNG decals is first defined using RGB values and then converted to spectral characteristics by a low-dimensional parametric model [75]. The operation makes the PNG decals contain less information than the mixed native spectral materials



(a) FreeNeRF (b) DNGaussian (c) SS-NeRF (d) Ground Truth

Fig. 13. Comparisons between SS-NeRF and sparse scene reconstruction methods in the CIE 1931 xy chromaticity space.

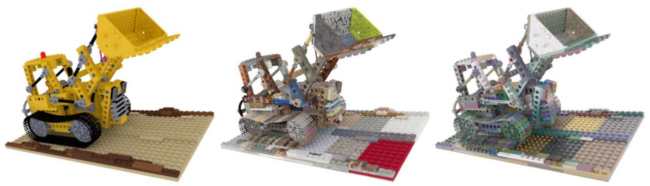


Fig. 14. The Digger model with original materials, PNG decals, and mixed materials.

TABLE IV
ABLATION EXPERIMENTS OF DIFFERENT MATERIALS

	NeRF+	Ours	Ours with PNG decals	Ours with mixed materials
PSNR ↑	22.583	24.048	23.663	23.159
SSIM ↑	0.8792	0.8866	0.8848	0.8817
LPIPS ↓	0.1215	0.0898	0.0935	0.1018

TABLE V
RESULTS WITH DIFFERENT λ_{DAC}, λ_{GPC}, AND λ_{SAC} VALUES

	λ_{DAC}, λ_{GPC} and λ_{SAC}	PSNR ↑	SSIM ↑	LPIPS ↓
(a)	0.01, 0.01, and 0.1	25.091	0.8385	**0.1574**
(b)	0.05, 0.01, and 0.1	24.942	0.8377	0.1612
(c)	0.02, 0.005, and 0.1	24.730	0.8370	0.1603
(d)	0.02, 0.05, and 0.1	24.960	0.8381	0.1626
(e)	0.02, 0.01, and 0.05	24.821	0.8367	0.1611
(f)	0.02, 0.01, and 0.5	24.815	0.8372	0.1605
(g)	0.02, 0.01, and 0.1 (Ours)	**25.151**	**0.8392**	0.1577

and is easier for networks to learn. Note that NeRF+ refers to the original NeRF enhanced with our techniques, such as DAC, GPC, and SAC, enabling it to handle sparse input views. However, its performance across all three material types remains inferior to our proposed SS-NeRF.

The Selection of Hyperparameters: We conduct additional ablation experiments to assess the impact of different hyperparameter settings for the λ_{DAC}, λ_{GPC}, and λ_{SAC}. Results demonstrate that the optimal configuration is $\lambda_{\text{DAC}} = 0.02$, $\lambda_{\text{GPC}} = 0.01$, and $\lambda_{\text{SAC}} = 0.1$. As shown in Table V(a), (b), and (g), changing λ_{DAC} from the optimal 0.02 to 0.01 or 0.05 leads to performance degradation, indicating the sensitivity of depth-aware regularization. Similarly, Table V(c), (d), and (g) illustrate that modifying λ_{GPC} to 0.005 or 0.5 also leads to suboptimal performance, showing that improper geometric constraints negatively impact structural consistency. Furthermore, as shown in Table V(e), (f), and (g), we observe that both too small λ_{SAC} (e.g., 0.05) and too large λ_{SAC} (e.g., 0.5) appearance weights degrade the rendering quality. These findings suggest that inaccurate settings can adversely affect the overall rendering fidelity. The optimal configuration achieves a good trade-off among depth alignment, geometric consistency, and appearance coherence.

TABLE VI
COMPARISONS OF SUPERPIXEL-BASED SEGMENTATION AND PATCH-BASED SEGMENTATION

	PSNR ↑	SSIM ↑	LPIPS ↓
Patch-based segmentation	24.825	0.8377	0.1624
Superpixel-based segmentation	25.151	0.8392	0.1577

The Effectiveness of Superpixel-based Segmentation: Superpixel-based segmentation plays a crucial role in improving the perceptual consistency and spatial coherence of appearance modeling. Unlike regular patch-wise constraints [46] that enforce consistency over uniformly sampled, fixed-size regions (generally 15×15), superpixel-based methods adaptively group pixels based on low-level visual features such as color and texture. This results in spatially coherent and semantically meaningful regions that often align well with object contours and material boundaries. Results in Table VI demonstrate the effectiveness of superpixel-based segmentation.

VII. CONCLUSION

We have proposed the SS-NeRF, an end-to-end NeRF-based architecture designed for physically-based spectral rendering with sparse inputs. We modified the traditional spectral rendering pipeline into two steps and designed SpectralMLP and SAUNet to build up the two steps. Furthermore, we incorporated the depth-aware continuity, the geometric-projected consistency, and the superpixel-aligned constraint to enhance the scene reconstruction. The proposed methods can generate high-quality RGB output of white-light illumination. Comprehensive experiments have demonstrated the superiority of the SS-NeRF. In the future, we plan to extend the NeRF-based architectures by integrating advanced frameworks like Gaussian splatting to further enhance reconstruction accuracy and rendering speed, facilitating broader real-time applications.

REFERENCES

[1] I. Newton, “A serie’s of quere’s propounded by Mr Isaac Newton, to be determin’d by experiments, positively and directly concluding his new theory of light and colours; and here recommended to the industry of the lovers of experimental philosophy, as they were generously imparted to the publisher in a letter of the said Mr Newtons of july 8.1672,” *Philos. Trans. Roy. Soc. London*, vol. 7, no. 85, pp. 5004–5007, 1672.

[2] R. Pickholtz, D. Schilling, and L. Milstein, “Theory of spread-spectrum communications—A tutorial,” *IEEE Trans. Commun.*, vol. COM-30, no. 5, pp. 855–884, May 1982.

[3] B. Helffer, *Spectral Theory and its Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[4] J. Liang, J. Zhou, L. Tong, X. Bai, and B. Wang, “Material based salient object detection from hyperspectral images,” *Pattern Recognit.*, vol. 76, pp. 476–490, 2018.

[5] M. Uzair, A. Mahmood, and A. Mian, “Hyperspectral face recognition with spatiospectral information fusion and PLS regression,” *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1127–1137, Mar. 2015.

[6] R. Dian, A. Guo, and S. Li, “Zero-shot hyperspectral sharpening,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12650–12666, Oct. 2023.

[7] D. Hong et al., “SpectralGPT: Spectral remote sensing foundation model,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, Aug. 2024.

[8] L. Bertrand, M. Thoury, P. Gueriau, É. Anheim, and S. Cohen, “Deciphering the chemistry of cultural heritage: Targeting material properties by coupling spectral imaging with image analysis,” *Accounts Chem. Res.*, vol. 54, no. 13, pp. 2823–2832, 2021.

[9] M. S. Peercy, “Linear color representations for full speed spectral rendering,” in *Proc. Annu. Conf. Comput. Graph. Interactive Techn.*, 1993, pp. 191–198.

[10] B. Smits, “An RGB to spectrum conversion for reflectances,” *J. Graph. Tools*, vol. 4, no. 4, pp. 11–22, 1999.

[11] S. Watanabe, S. Kanamori, S. Ikeda, B. Raytchev, T. Tamaki, and K. Kaneda, “Performance improvement of physically based spectral rendering using stochastic sampling,” in *Proc. Int. Workshop Comput. Color Imag.*, 2013, pp. 184–198.

[12] Y. Sun, F. D. Fracchia, M. S. Drew, and T. W. Calvert, “A spectrally based framework for realistic image synthesis,” *Vis. Comput.*, vol. 17, no. 7, pp. 429–444, 2001.

[13] G. Wu, Y. Liu, L. Fang, and T. Chai, “Revisiting light field rendering with deep anti-aliasing neural network,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5430–5444, Sep. 2022.

[14] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[15] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P.P. Srinivasan, “Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 5855–5864.

[16] J. T. Barron, B. Mildenhall, D. Verbin, P.P. Srinivasan, and P. Hedman, “Mip-NeRF 360: Unbounded anti-aliased neural radiance fields,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5470–5479.

[17] Z. Yan, C. Li, and G. H. Lee, “NeRF-DS: Neural radiance fields for dynamic specular objects,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8285–8295.

[18] Y.-H. Huang, Y.-P. Cao, Y.-K. Lai, Y. Shan, and L. Gao, “NeRF-Texture: Synthesizing neural radiance field textures,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 9, pp. 5986–6000, Sep. 2024.

[19] D. Jumanazarov, J. Koo, H. F. Poulsen, U. L. Olsen, and M. Iovea, “Significance of the spectral correction of photon counting detector response in material classification from spectral X-ray CT,” *J. Med. Imag.*, vol. 9, no. 3, pp. 34504–34504, 2022.

[20] R. Li, J. Liu, G. Liu, S. Zhang, B. Zeng, and S. Liu, “SpectralNeRF: Physically based spectral rendering with neural radiance field,” in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 3154–3162.

[21] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, “RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5480–5490.

[22] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, “SPARF: Neural radiance fields from sparse and noisy poses,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4190–4200.

[23] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “PixelNeRF: Neural radiance fields from one or few images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4578–4587.

[24] A. Chen et al., “MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14124–14133.

[25] A. Jain, M. Tancik, and P. Abbeel, “Putting NeRF on a diet: Semantically consistent few-shot view synthesis,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 5885–5894.

[26] Z. Ni, P. Yang, W. Yang, H. Wang, L. Ma, and S. Kwong, “ColNeRF: Collaboration for generalizable sparse input neural radiance field,” in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 4325–4333.

[27] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, “Depth anything: Unleashing the power of large-scale unlabeled data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10371–10 381.

[28] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12179–12 188.

[29] Q. Dai, Y. Song, and Y. Xin, “Random-accessible volume data compression with regression function,” in *Proc. Int. Conf. Comput.-Aided Des. Comput. Graph.*, 2015, pp. 137–142.

[30] D. Wu, S.-T. Xia, and Y. Wang, “Adversarial weight perturbation helps robust generalization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 2958–2969.

[31] B. Mildenhall et al., “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” *ACM Trans. Graph*, vol. 38, no. 4, pp. 1–14, 2019.

[32] T. Chen, P. Wang, Z. Fan, and Z. Wang, “Aug-NeRF: Training stronger neural radiance fields with triple-level physically-grounded augmentations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15191–15 202.

[33] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “NeRF in the wild: Neural radiance fields for unconstrained photo collections,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7210–7219.

[34] C. Reiser, S. Peng, Y. Liao, and A. Geiger, “KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14335–14 345.

[35] Z. Li, S. Niklaus, N. Snavely, and O. Wang, “Neural scene flow fields for space-time view synthesis of dynamic scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6498–6508.

[36] Z. Li, Q. Wang, F. Cole, R. Tucker, and N. Snavely, “DynIBaR: Neural dynamic image-based rendering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4273–4284.

[37] M. Bemana, K. Myszkowski, J. Revall Frisvad, H.-P. Seidel, and T. Ritschel, “Eikonal fields for refractive novel-view synthesis,” in *Proc. ACM Trans. Graph Conf.*, 2022, pp. 1–9.

[38] A. Cao and J. Johnson, “HexPlane: A fast representation for dynamic scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 130–141.

[39] S. Huang et al., “Neural LiDAR fields for novel view synthesis,” 2023, *arXiv:2305.01643*.

[40] M. Poggi, P. Z. Ramirez, F. Tosi, S. Salti, S. Mattoccia, and L. Di Stefano, “Cross-spectral neural radiance fields,” in *Proc. Int. Conf. 3D Vis.*, 2022, pp. 606–616.

[41] V. Rudnev, M. Elgharib, C. Theobalt, and V. Golyanik, “EventNeRF: Neural radiance fields from a single colour event camera,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4992–5002.

[42] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang, “SparseNeuS: Fast generalizable neural surface reconstruction from sparse views,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 210–227.

[43] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised NeRF: Fewer views and faster training for free,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12882–12 891.

[44] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, “MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 25018–250 32.

[45] G. Wang, Z. Chen, C. C. Loy, and Z. Liu, “SparseNeRF: Distilling depth ranking for few-shot novel view synthesis,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 9065–9076.

[46] M.-S. Kwak, J. Song, and S. Kim, “GeCoNeRF: Few-shot neural radiance fields via geometric consistency,” in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 18023–180 36.

[47] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3D gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph*, vol. 42, no. 4, pp. 1–14, 2023.

[48] J. Li et al., “DNGaussian: Optimizing sparse-view 3D gaussian radiance fields with global-local depth normalization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20775–20 785.

[49] H. Xiong, S. Muttukuru, R. Upadhyay, P. Chari, and A. Kadambi, “SparseGS: Real-time 360° sparse view synthesis using Gaussian splatting,” 2023, *arXiv:2312.00206*.

[50] R. Yin, V. Yugay, Y. Li, S. Karaoglu, and T. Gevers, “FewViewGS: Gaussian splatting with few view matching and multi-stage training,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 127204–127 225.

[51] P. Dai, Z. Li, Y. Zhang, S. Liu, and B. Zeng, “PBR-Net: Imitating physically based rendering using deep neural network,” *IEEE Trans. Image Process.*, vol. 29, pp. 5980–5992, 2020.

[52] R. Li, P. Dai, G. Liu, S. Zhang, B. Zeng, and S. Liu, "PBR-GAN: Imitating physically based rendering with generative adversarial networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1827–1840, Mar. 2024.

[53] T. Hu, X. Xu, S. Liu, and J. Jia, "Point2Pix: Photo-realistic point cloud rendering via neural radiance fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8349–8358.

[54] K. D. A. C. A. Wilkie et al., "Tone reproduction and physically based spectral rendering," in *Proc. Eurographics Conf.*, 2002, pp. 101–123.

[55] C. Peters, S. Merzbach, J. Hanika, and C. Dachsbacher, "Using moments to represent bounded signals for spectral rendering," *ACM Trans. Graph*, vol. 38, no. 4, pp. 1–14, 2019.

[56] Color and V. R. Labs, 1995. [Online]. Available: http://cvrl.ioo.ucl.ac.uk

[57] Mitsuba, 2010. Accessed: 2010. [Online]. Available: http://www.mitsuba-renderer.org/

[58] N. Rahaman et al., "On the spectral bias of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5301–5310.

[59] M. Levoy, "Efficient ray tracing of volume data," *ACM Trans. Graph*, vol. 9, no. 3, pp. 245–261, 1990.

[60] O. Oktay et al., "Attention U-net: Learning where to look for the pancreas," 2018, *arXiv: 1804.03999*.

[61] J. Jiang, C. Wang, X. Liu, K. Jiang, and J. Ma, "From less to more: Spectral splitting and aggregation network for hyperspectral face super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 267–276.

[62] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[63] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang, "FSGS: Real-time few-shot view synthesis using gaussian splatting," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 145–163.

[64] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 224–236.

[65] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local feature matching at light speed," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 17627–17 638.

[66] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[67] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[68] Blender, 1994. Accessed: 1994. [Online]. Available: http://www.blender.org/

[69] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[70] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–11.

[72] J. Yang, M. Pavone, and Y. Wang, "FreeNeRF: Improving few-shot neural rendering with free frequency regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8254–8263.

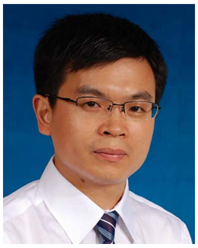
[73] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

[74] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes, "Proposal for a standard default color space for the Internet—sRGB," in *Proc. Color Imag. Conf.*, 1996, pp. 238–245.

[75] W. Jakob and J. Hanika, "A low-dimensional function space for efficient spectral upsampling," *Comput. Graph. Fourm*, vol. 38, no. 2, pp. 147–155, 2019.

Ru Li (Member, IEEE) received the BE degree in electronic information engineering from the China University of Petroleum, Qingdao, China, in 2016, and the PhD degree from the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2022. She is currently a lecture with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. Her research interests include computer vision and computer graphics.

Jia Liu received the BE and MSc degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2021 and 2024, respectively. Currently, he is working toward the PhD degree with the University of Electronic Science and Technology of China. His research interests include computer vision and computer graphics.



Guanghui Liu (Senior Member, IEEE) received the MSc and PhD degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2002 and 2005, respectively. In 2005, he joined Samsung Electronics, Seoul, South Korea, as a senior engineer. In 2009, he became an associate professor with the School of Electronics Engineering, UESTC, where he has been a full professor since 2014 and is currently with the School of Information and Communication Engineering. His general research interests include multimedia, remote sensing, and wireless communication.

Shengping Zhang received the PhD degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. He had been a postdoctoral research associate with Brown University and with Hong Kong Baptist University, and a visiting student researcher with the University of California at Berkeley. His research interests include deep learning and its applications in computer vision.

Bing Zeng (Fellow, IEEE) received the BE and MSc degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the PhD degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991. He worked as a post-doctoral fellow with the University of Toronto from September 1991 to July 1992, and as a researcher with Concordia University from August 1992 to January 1993. Then, he joined The Hong Kong University of Science and Technology (HKUST). He returned to UESTC in Summer 2013. At UESTC, he works on image and video processing, 3D and multiview video technology, and visual Big Data.

Shuaicheng Liu (Senior Member, IEEE) received the BE degree from Sichuan University, Chengdu, China, in 2008, and the MSc and PhD degrees from the National University of Singapore, Singapore, in 2014 and 2010, respectively. In 2015, he joined the University of Electronic Science and Technology of China (UESTC) and is currently a professor with the Institute of Image Processing, School of Information and Communication Engineering, Chengdu. He works on computer vision, computer graphics and computational imaging related problems, with applications in mobile photography and videography.