Cross-Ray Neural Radiance Fields for Novel-view Synthesis from Unconstrained Image Collections

Yifan Yang, Shuhai Zhang, Zixiong Huang, Yubing Zhang, Mingkui Tan
1South China University of Technology 2Pazhou Lab 3Guangzhou Shiyuan Electronics Co., Ltd 4Key Laboratory of Big Data and Intelligent Robot, Ministry of Education

Abstract

Neural Radiance Fields (NeRF) is a revolutionary approach for rendering scenes by sampling a single ray per pixel and it has demonstrated impressive capabilities in novel-view synthesis from static scene images. However, in practice, we usually need to recover NeRF from unconstrained image collections, which poses two challenges: 1) the images often have dynamic changes in appearance because of different capturing time and camera settings; 2) the images may contain transient objects such as humans and cars, leading to occlusion and ghosting artifacts. Conventional approaches seek to address these challenges by locally utilizing a single ray to synthesize a color of a pixel. In contrast, humans typically perceive appearance and objects by globally utilizing information across multiple pixels. To mimic the perception process of humans, in this paper, we propose Cross-Ray NeRF (CR-NeRF) that leverages interactive information across multiple rays to synthesize occlusion-free novel views with the same appearances as the images. Specifically, to model varying appearances, we first propose to represent multiple rays with a novel cross-ray feature and then recover the appearance by fusing global statistics, i.e., feature covariance of the rays and the image appearance. Moreover, to avoid occlusion introduced by transient objects, we propose a transient objects handler and introduce a grid sampling strategy for masking out the transient objects. We theoretically find that leveraging correlation across multiple rays promotes capturing more global information. Moreover, extensive experimental results on large real-world datasets verify the effectiveness of CR-NeRF. The code and data can be found at https://github.com/YifYang993/CR-NeRF-PyTorch.git.

1. Introduction

Novel-view synthesis is a long-standing problem in computer vision that has paved the way for numerous applications such as virtual reality and digital humans [13, 45]. More recently, the emergence of Neural Radiance Fields (NeRF) has driven the field forward, as it has shown significant performance in reconstructing 3D geometry [50] and recovering the appearance [3, 35, 1] from multi-view image sets. However, NeRF assumes that the images do not have variable appearances and moving objects [31] (called the static scene assumption c.f. Sec. 3), which leads to significant performance degradation on large-scale Internet image collections. To expand the scope of NeRF, we aim to exploit the collections and provide a 3D immersive experience through which we can visit international landmarks such as the Brandenburg Gate, and the Trevi Fountain from different viewpoints and times of one day.

To achieve this, we address the problem of recovering an appearance-controllable and anti-occlusion NeRF from unconstrained image collections. In other words, by reconstructing the NeRF representation, we control the appearance of the scene based on photos with various photometric conditions, while eliminating occlusions caused by the images. Although providing a sense of immersion, reconstructing NeRF with these images faces the following two challenges. 1) Varying appearances: Imaging two tourists who take photos in the same viewpoint but under various conditions, e.g., different capturing times, diverse weather (e.g., sunny, rainy, and foggy), and different camera settings (e.g., aperture, shutter, and ISO). This varying condition causes that although multiple photographs are taken of the same scene, they look dramatically different. 2) Transient occlusion: Even with a constant appearance, transient objects such as cars and Pedestrians may obscure the scene. Since these objects are usually captured by only one photographer, it is usually impractical to reconstruct these objects in high quality. The above challenges conflict with the static-scene
assumption of NeRF and result in inaccurate reconstruction that leads to over-smoothing and ghosting artifacts [31].

Recently, several attempts (NeRF-W [31]; Ha-NeRF [6]) have been proposed to address the aforementioned challenges. From Fig. 1(a), NeRF-W and Ha-NeRF leverage a single-ray manner, wherein a single camera ray (i.e., a beam of light extending from a camera through a pixel on an image plane into a 3D scene) serves as input. This manner then involves considering appearance and occlusion factors and subsequently synthesizing each color of pixel of a novel view independently. One potential issue of this manner is its reliance on local information (e.g., information of a single image pixel) of every single ray for recognizing appearance and transient objects. In contrast, humans tend to utilize global information (e.g., information across multiple image pixels), which provides a more comprehensive understanding of an object to observe its appearance and handle occlusion. Motivated by this, we propose to tackle varying appearance and transient objects with a cross-ray paradigm (see Fig. 1(b)), wherein we utilize global information from multiple rays to recover the appearance and handle transient objects. Subsequently, we synthesize a region of a novel view simultaneously. Based on the cross-ray paradigm, we propose a Cross-Ray Neural Radiance Fields (CR-NeRF), which comprises two components: 1) To model variable appearances, we propose to represent information of multiple rays with a novel cross-ray feature. We then fuse the cross-ray feature and an appearance embedding via a cross-ray transformation network using global statistics, e.g., feature covariance of the cross-ray. The fused feature is fed to a decoder to obtain colors of several pixels simultaneously. 2) To handle transient objects, we propose a unique perspective of handling transient objects as a segmentation problem, through which we detect transient objects by considering global information of an image region. From this perspective, we segment the unconstrained images for a visibility map of the objects. To avoid computation overhead, we introduce a grid sample strategy that samples the segmented maps to pair with the input rays. We theoretically analyze that leveraging correlation across multiple rays promotes capturing more global information.

We summarize our contributions in three folds:

- A new cross-ray paradigm for novel-view synthesis from unconstrained photo collections: We find that existing methods fall short of producing satisfactory visual outcomes from unconstrained photo collections via a single-ray-level paradigm, primarily due to the neglect of the potential cooperative interaction among multiple rays. To address this, we propose a novel cross-ray paradigm, which exploits the global information across multiple rays.
- An interactive and global scheme for addressing varying appearances: Unlike existing methods that process each ray independently, we represent multiple rays by introducing a cross-ray feature, which facilitates the interaction among rays through feature covariance. This enables us to inject a global informative appearance representation into the scene, resulting in more realistic and efficient appearance modeling. Our theoretical analysis demonstrates the necessity of considering multiple rays for appearance modeling.
- A novel segmentation technique for processing transient objects: We reformulate the transient object problem as a segmentation problem. We use global information of an unconstrained image to segment a visibility map. Moreover, we apply grid sampling to pair the map with multiple rays. Empirical results show that CR-NeRF eliminates the transient objects in reconstructed images.

2. Related Works

Neural rendering. Neural rendering applies deep learning with computer graphic technologies to render images and reconstruct 3D scenes. Recent advances seek to apply learning-based technology to generate representations such as signed distance field [34, 27, 59, 19], point clouds [10, 17, 26], voxels [37, 15, 57] and occupancy fields [58, 32, 38], which are then applied for rendering novel views. With the remarkable performance, NeRF [33] has attracted attention from the neural rendering community. More recently, NeRF has been extended to represent a time-series of scenes [25, 39, 22], handle high-resolution settings [18, 52], address relighting [43], and reconstruct large-scale environments [46, 47, 48]. Notably, one limitation of NeRF is that it assumes the scene is static, which faces challenges of varying appearance and presence of transient objects in unconstrained image collections. To alleviate this, NeRF-W [31] and Ha-NeRF [6] focus on addressing the challenges by processing each ray of a scene indepen-
Novel-view synthesis. Synthesizing views from a novel viewpoint has long been a fundamental problem in computer vision and computer graphics. Traditionally, novel views can be synthesized through 4D light field strategy [21, 53, 5]. However, the strategy requires a dense camera array for capturing data, which is usually impractical. Since collecting a sparse set of images is efficient, view synthesis research takes advantage of geometry structure [2, 7] to aid in constructing novel views with limited input. With the flourish of deep learning, deep neural networks have been leveraged to estimate the scene geometry (e.g., point clouds [54, 40], depth map [44, 28], multiple-layer image [11, 55]). Although leveraging the geometry enhances the quality of novel views, the estimation is usually without ground truth supervision and usually is not accurate enough. To circumvent the difficulty of estimating precise geometry, we propose to utilize an implicit function, i.e., neural radiance fields (NeRF) [33] for novel-view synthesis.

3. Preliminaries

Neural Radiance Fields (NeRF) [33] implicitly represents a static 3D scene with multilayer perceptron (MLP) and then produces a novel view via volume rendering (VR) [9]. NeRF generates a pixel color of a novel view from a camera ray independently. In this sense, we can describe the rendering process w.r.t. a single camera ray \( r(t) = o + td \) which is cast from a camera center \( o \) in the direction \( d \) and passes through a pixel on an image plane w.r.t. the novel view. We sample \( n \) ray points \( \{r(t_i)\}_{i=1}^n \) along \( r \) between a given near plane \( t_n \) and a far plane \( t_f \). For each ray point \( r(t_i) \), we query the MLP at a 3D position \( x_i = (x, y, z) \) and a viewing orientation \( d_i = (d_x, d_y, d_z) \) to obtain a color \( k_i = (r, g, b) \) and a density \( \sigma_i \) via equations: \( x_i \rightarrow \{F^i, \sigma_i\}, \{F^i, d_i\} \rightarrow k_i \), where \( F^i \) denotes a ray-point-level feature regarding the ray point \( i \). To learn high-frequency information, position encoding [33] is employed to \( x_i \) and \( d_i \). Typically, \( o \) and \( d \) are estimated by structure from motion approaches [41, 49] from multi-view images regarding the 3D scene.

To approximate the color \( \hat{c}(r) \) of the pixel of a reference image, NeRF accumulates \( n \) ray points \( \{r(t_i)\}_{i=1}^n \) along the ray \( r \) into the \( \hat{c}(r) \) via VR [9]:

\[
\hat{c}(r) = \sum_{i=1}^{n} \varphi_i \alpha_i k_i, \quad \varphi_i = \exp(-\sum_{l=1}^{i-1} \sigma_l \delta_l), \quad \alpha_i = 1 - \exp(-\sigma_i \delta_i) .
\]

(1)

Here, \( \alpha_i \) is the probability of the ray that terminates at \( r(t_i) \); \( \varphi_i \) is the accumulated transmittance from the near plane \( t_n \) to \( r(t_i) \); \( \delta_i = t_{i+1} - t_i \) is distance between two adjacent ray points. The MLP is optimized via minimizing the loss: \( \mathcal{L} = \| \hat{c}(r) - c(r) \|^2 \), where \( c(r) \) denotes the ground truth color of a pixel w.r.t. the ray \( r \).

Limitations of NeRF on novel-view synthesis from unconstrained collections. Given an unconstrained collection of a scene, we seek to reconstruct the scene whose
appearance can be modified according to a new image, while removing transient objects. Since NeRF assumes the lighting in the scene is constant over time and there are no moving objects or changes in lighting during the time that the input images are captured (called static scene assumption [31]), NeRF is limited to effectively modeling the geometry and appearance of static scenes only. To address the limitation of NeRF, recent advances [31, 6] synthesize novel views on single-ray level (see Fig. 1 (a)) following equation:

\[ \{x_i, d_i, F_{i0}\}_{i=1}^n \rightarrow \hat{c}_n, \]

where \( F_{i0} \) is image-level conditional embedding of \( I_0 \). From Eqn. 2, existing methods [31, 6] generate the color \( \hat{c}_n \) of each pixel by processing the corresponding single ray independently. This manner ignores global information among multiple rays, leading to inaccurate appearance modeling (see our empirical studies in Fig. 3 and Fig. 4).

4. Cross-Ray Neural Radiance Fields

Given unconstrained photo collections of a scene, we seek to reconstruct the scene whose appearance can be modified based on a new image, while removing transient objects. This task is challenging due to the existence of variable appearances and transient occlusions in the photo collections. To address this, intuitively by that a human usually detects appearance and transient objects by considering global information (e.g., information across several image pixels) rather than local information (e.g., information of a single image pixel), we propose a Cross-Ray Neural Radiance Fields (CR-NeRF) that exploits global information across multiple camera rays, which correspond to several pixels of an image, to address both challenges.

As shown in Fig. 2 and Alg. 1, CR-NeRF consists of two components: 1) Cross ray appearance modeling (c.f. Sec. 4.1). To model varying appearances, we first sample a grid of rays using a grid sampling strategy [42]. Next, we represent the rays with a novel cross-ray feature \( F_{cr} \). We then inject an appearance embedding \( \mathcal{F}_a \) into \( F_{cr} \) via a learned transformation network. The fused feature is fed to a decoder for obtaining colors of multiple pixels simultaneously. We theoretically analyze the necessity of considering multiple rays and thus design an appearance loss \( L_a \) for cross-ray appearance modeling. 2) Cross-ray transient objects handling (c.f. Sec. 4.2). To handle transient objects, we deploy a segmentation network for generating a visibility map regarding transient objects. To pair the map with the rays, we also apply the grid sampling strategy on the maps. We devise an occlusion loss \( L_t \) for transient handling.

The overall optimization of our proposed CR-NeRF minimizes the following objective function:

\[ L_{overall} = L_a + \lambda L_t, \]

where \( \lambda \) is a hyper-parameter for balancing the appearance loss \( L_a \) (see Eqn. 8) and the occlusion loss \( L_t \) (see Eqn. 9).

4.1. Cross-Ray Appearance Modeling

To adapt CR-NeRF to variable appearance through a global perspective, we modify the scene by leveraging multiple rays and the appearance of the unconstrained images.

Representing scene information with multiple rays. To model appearance from a multi-view observation, we first represent scene information using multiple rays. To this end, we propose a novel cross-ray feature \( F_{cr} \) with equations:

\[
\{\{F_{ij}, \sigma_{ij}\}_{i=1}^n\}_{j=1}^m = \{\text{MLP}_{\theta}(\{x_{ij}, d_{ij}\}_{i=1}^n)\}_{j=1}^m, \\
F_{cr} = \{\text{VR}((\{F_{ij}, \sigma_{ij}, \delta_{ij}\}_{i=1}^n))\}_{j=1}^m.
\]

Besides, we obtain an appearance feature \( F_a \) of an appearance image \( I_0 \) by \( F_a = E_{\theta} (I_0) \). With \( F_{cr} \) and \( F_a \), it is critical to find an effective fusion manner to inject image appearance into the scene representation.

Injecting appearance into scene representation. The key of our cross-ray appearance modeling is to exploit the potential cooperative relationship among the cross-ray features \( F_{cr} \) to facilitate appearance modeling from the given appearance image \( I_0 \) to the scene representation. In other words, we seek a transformation operation that can transfer the style from a reference image and also retain the essential content during training. To this end, we learn a transformation \( T \) to align the transferred cross-ray features \( T(F_{cr}) \) and the appearance feature \( F_a \) with an auxiliary identity term, which is formulated as below,

\[
\min_T \mathbb{E}_{F_{cr}, F_a} \|T(F_{cr}) - F_a\|^2 + \beta \|P(T(F_{cr}) - F_a)\|^2, \tag{5}
\]

where \( \beta \) is a trade-off parameter and \( P \) is a constant matrix for matching the transformed feature \( T(F_{cr}) \) and \( F_{cr} \). Next, we theoretically analyze the necessity of considering multiple rays to solve Problem (5) for appearance modeling.

Necessity of considering multiple rays for appearance modeling. We consider a Gaussian case that can provide some insights to devise an effective approach to inject the appearance into the scene representation. To this end, we assume the two features \( F_{cr} \) and \( F_{cr} \) are following two Gaussian distributions and \( T \) is a linear transformation that rigorously matches two distributions. We provide a closed-form solution to Problem (5) under this assumption as follows.

\[
T = P^{-1} \Sigma_{cr}^{-1/2} \left( \Sigma_{cr}^{-1/2} P \Sigma_{a} P \Sigma_{cr}^{-1/2} \right)^{1/2} \Sigma_{cr}^{-1/2}. \tag{6}
\]
Algorithm 1: The training pipeline of CR-NeRF.

Input: $m$ rays, a reference image $I_n$, a multilayer perceptron MLP$_{\eta_1}$, an appearance encoder $E_{\theta_2}$, a transformation net $T_{\theta_3}$, a decoder $D_{\theta_4}$, a content encoder $E_{\theta_5}$, and a segmentation net $S_{\theta_6}$.

Output: The estimated colors of $m$ pixels of a novel view.

1. while not converged do
   2. Generate cross-ray features $\mathcal{F}^{cr}$ and an appearance feature $\mathcal{F}^a$ with $E_{\theta_5}$ and MLP$_{\eta_1}$ by Eqn. (4).
   3. Obtain the loss $L_a$ for modeling appearance with $E_{\theta_2}$, $T_{\theta_3}$, $D_{\theta_4}$ and $E_{\theta_5}$ by Eqn. (8). $\triangleright$ c.f. Sec. 4.1
   4. Obtain the visibility map $M$ for masking out transient objects with and $S_{\theta_6}$ by Eqn. (9).
   5. Obtain the loss $L_i$ for handling transient with $T_{\theta_3}$, $D_{\theta_4}$ and $S_{\theta_6}$ by Eqn. (10). $\triangleright$ c.f. Sec. 4.2
   6. Obtain the overall loss of $L_{overall} = L_a + \lambda L_i$.
   7. Update the parameters $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_5, \theta_6\}$ by descending the gradient: $\nabla_{\Theta} L_{overall}$

End.

Proposition 1 suggests the transformation matrix $T$ is determined by the covariance of $\mathcal{F}^{cr}$ and $\mathcal{F}^a$ given $P$, which is consistent with the conclusion in [23, 29]. Inspired by this, we can construct a neural network to learn the appearance transformation $T$ by feeding the covariances of $\mathcal{F}^a$ and $\mathcal{F}^{cr}$.

Specifically, we adopt an effective transformation network following Li et al. [23] which is defined as:

$$T(\mathcal{F}^{cr}) = \mathbf{T} \hat{\mathcal{F}}^{cr},$$

where $\mathbf{T} = \text{Cov}(\mathcal{F}^{cr})\text{Cov}(\mathcal{F}^a))$, $\hat{\mathcal{F}}^{cr} = \phi_1(\mathcal{F}^{cr})$, $\hat{\mathcal{F}}^a = \phi_2(\mathcal{F}^a)$.

Here, $\phi_1$, $\phi_2$, and $\phi_3$ are non-linear mappings parameterized by convolutional neural networks (CNN) that can express richer embedding to prepare for appearance modeling. Intuitively, we consider multiple rays when modeling appearance to employ multi-view information. The correlation between the feature maps of these different views, which can be given by the covariance, is able to capture more global texture information for a given appearance image [12, 24, 8], thus facilitating better appearance modeling for a scene.

**Loss function $L_a$ for varying appearance modeling:** To generate a novel-view image with a satisfactory appearance from the transformed feature $T_{\theta_3}(\mathcal{F}^{cr})$, we need to enforce a decoder $D_{\theta_4}$ into the training process of appearance modeling. Inspired by the formulation in Problem (5), we provide the loss function for appearance modeling as:

$$L_a = ||E_{\theta_5}[D_{\theta_4}(T_{\theta_3}(\mathcal{F}^{cr}))] - \mathcal{F}^a||^2_2 + \beta||E_{\theta_5}[D_{\theta_4}(T_{\theta_3}(\mathcal{F}^{cr}))] - E_{\theta_5}[D_{\theta_4}(\mathcal{F}^{cr})]||^2_2,$$

where $\mathcal{F}^{cr}$ is obtained with an MLP$_{\eta_1}$ by Eqn. 4. Here, we use a tailored encoder $E_{\theta_5}$ to model the transformed feature $\mathbf{PT}(\mathcal{F}^{cr})$ so that the content of the transformed image closely matches its original counterpart. In this way, we can synthesize a novel-view image by $I_n = D_{\theta_4}(T_{\theta_3}(\mathcal{F}^{cr}))$.

### 4.2. Transient Objects Handling

To deal with transient objects caused by unconstrained photo collections for novel-view synthesis, we propose a new perspective, i.e., obtaining the visibility map of transient objects by segmenting the reference image $I_n$. With the receptive fields of a deep segmentation network [30], the interactions of different pixels and rays are facilitated, thus introducing more global information.

To accurately detect transient objects, we start by exploring a pre-trained Mask R-CNN model [14] and a pre-trained DeepLabV3 model [4] that are capable of effectively segmenting common objects such as tourists and cars, etc. We observe that although the models properly segment the common objects, the reconstruction error is amplified (see empirical studies in Sect. 5.2). The possible reason is that the target transient objects are not limited to common objects, more objects (e.g., shadows of tourists in Fig. 6) should also be taken into consideration.

In this sense, we choose a learning-based manner to select which objects to segment and therefore deploy a light-weight segmentation network $S_{\theta_6}$ following [56]. Since we cannot sample all rays that interact with $I_n$ due to limited GPU memory in the training phase, naively processing all rays of transient objects (i.e., $S_{\theta_6}(I_n)$) is therefore not applicable. Hence, we apply a grid sampling strategy (GS) [42] which samples $S_{\theta_6}(I_n)$ to pair with $m$ rays (see Fig. 2). The whole process for estimating $M$ is:

$$M = \text{GS}(S_{\theta_6}(I_n)),$$

where $S_{\theta_6} : \mathbb{R}^{3 \times h_{cr1} \times h_{cr2}} \rightarrow \mathbb{R}^{3 \times h_{cr1} \times h_{cr2}}$ are heights and width of $I_n$. Here, $S_{\theta_6}$ learns a visibility map $M$ without the supervision of ground truth segmentation masks. During training, we set $m$ to be smaller than $h_{cr1} \times h_{cr2}$ for saving computational overhead.

**Loss function $L_i$ for eliminating transient objects:** The loss function for handling transient objects is:

$$L_i = ||(1 - M) \odot (I_n - I_\alpha)|^2_1 + \lambda_0||M||^2_2,$$

where $\odot$ denotes element-wise multiplication. The loss $L_i$ aims to mask out transient objects via $M$. To prevent our transient network from masking everything, we follow Ha-NeRF to add $\lambda_0||M||^2_2$ as a regularization term.

### 4.3. Difference of CR-NeRF with Existing Methods

To model varying appearances, Ha-NeRF and NeRF-W process each single ray independently by Eqn. 2. To handle transient objects, NeRF-W implements an additional MLP for rendering transient objects by Eqn. 2. Ha-NeRF
estimates a visibility map by separately utilizing a UV coordinate and a conditional feature of a reference image. Differently, CR-NeRF considers information across multiple rays. Specifically, CR-NeRF takes \( m \) rays as input, fuses them with a conditional feature, and generates a region of an image simultaneously (see Fig. 1 (b)). A recent work i.e., 4K-NeRF To capture ray correlation, leverages depth-modulated convolutions. In contrast, CR-NeRF captures the covariance of different rays. We theoretically (c.f. Sec. 4.1) and empirically (see details in the appendix) analyze the necessity of considering multiple rays.

### 5. Experiments

**Implementation details.** We implement our approach using PyTorch [36] and train our networks with Adam [20] optimizer. For a fair comparison, we follow all common hyper-parameter settings same as Ha-NeRF [6], e.g., setting the number of input rays, learning rate, \( \lambda \) and height and width of fully connected layers to 1024, 5 \times 10^{-3}, 1 \times 10^{-3}, 8 and 256, respectively. We set \( \beta \) to 1 \times 10^{-5}. For a thorough study, we downscale the original image of each dataset by 2 times (R/2) and 4 times (R/4). During inference, we omit the segmentation network \( S_{\theta_A} \), see more details about the inference of CR-NeRF in the appendix.

**Datasets, metrics, and comparison methods.** Following Ha-NeRF [6], we evaluate our proposed method on three datasets: Brandenburg Gate, Sacre Coeur, and Trevi Fountain. For visual inspection, we present rendered images generated from the same set of input views. We also report quantitative results based on PSNR, SSIM [51], and LPIPS [60, 16]. We evaluate our proposed method against NeRF [33], NeRF-W [31], Ha-NeRF [6]. For ablation studies, we construct several variants of our CR-NeRF: 1) CR-NeRF-R replaces the cross-ray features from CR-NeRF with a ray-point-level features, i.e., features of ray points along multiple rays; 2) CR-NeRF-B is constructed upon CR-NeRF without the cross-rays appearance modeling module and transient handling module; 3) CR-NeRF-A eliminates the cross-rays appearance modeling module only and 4) CR-NeRF-T removes the transient handling module.

<table>
<thead>
<tr>
<th>R / 4</th>
<th>Brandenburg Gate</th>
<th>Sacre Coeur</th>
<th>Trevi Fountain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR (↑)</td>
<td>SSIM (↑)</td>
<td>LPIPS (↓)</td>
</tr>
<tr>
<td>NeRF</td>
<td>19.62</td>
<td>0.8200</td>
<td>0.1455</td>
</tr>
<tr>
<td>NeRF-W*</td>
<td>24.00</td>
<td>0.8758</td>
<td>0.1332</td>
</tr>
<tr>
<td>Ha-NeRF</td>
<td>24.58</td>
<td>0.8829</td>
<td>0.0927</td>
</tr>
<tr>
<td>CR-NeRF (Ours)</td>
<td>26.18</td>
<td>0.8937</td>
<td>0.0840</td>
</tr>
<tr>
<td>CR-NeRF (Ours)</td>
<td><strong>26.86</strong></td>
<td><strong>0.9069</strong></td>
<td><strong>0.0733</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R / 2</th>
<th>Brandenburg Gate</th>
<th>Sacre Coeur</th>
<th>Trevi Fountain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR (↑)</td>
<td>SSIM (↑)</td>
<td>LPIPS (↓)</td>
</tr>
<tr>
<td>NeRF</td>
<td>18.90</td>
<td>0.8159</td>
<td>0.2316</td>
</tr>
<tr>
<td>NeRF-W*</td>
<td>24.17</td>
<td>0.8905</td>
<td>0.1670</td>
</tr>
<tr>
<td>Ha-NeRF</td>
<td>24.04</td>
<td>0.8773</td>
<td>0.1391</td>
</tr>
<tr>
<td>CR-NeRF (Ours)</td>
<td>25.94</td>
<td>0.8929</td>
<td>0.1378</td>
</tr>
<tr>
<td>CR-NeRF (Ours)</td>
<td><strong>26.53</strong></td>
<td><strong>0.9003</strong></td>
<td><strong>0.1060</strong></td>
</tr>
</tbody>
</table>

Table 1. Quantitative experimental results on three real-world datasets under two resolution settings, i.e., downsampling original image resolution by 2 (R/2) and 4 (R/4). The bold and the underlined numbers indicate the best and second-best results, respectively.

**5.1. Comparison Experiments**

**Quantitative experiments.** We conduct extensive experiments on Brandenburg Gate, Sacre Coeur, and Trevi Fountain datasets. We follow Ha-NeRF with the image resolution setting of 2 \times downscaling (R/2) and further evaluate the effectiveness of our CR-NeRF on 4 \times downscaling (R/4). As demonstrated in Tab. 1, we observe that vanilla NeRF performs worst among all methods, since NeRF assumes the scene behind the training images is static. By modeling the style embedding and handling the transient objects, NeRF-W and Ha-NeRF achieve competitive performance in terms of PSNR, SSIM, and LPIPS. Note that NeRF-W optimizes its style embedding on test images since NeRF-W can not transfer to unseen test images directly. Thus, the comparison with NeRF-W is unfair. Even with the unfair comparison, thanks to the cross-ray manner, our CR-NeRF outperforms NeRF-W and Ha-NeRF on Brandenburg and Trevi under two downscaling settings.

**Qualitative experiments.** We summarize the qualitative results of all comparison methods in Fig. 3. We observe that NeRF produces foggy artifacts and inaccurate appearance. NeRF-W and Ha-NeRF are able to reconstruct a more promising 3D geometry and model appearance from the ground truth image. However, the reconstructed geometry is not accurate enough, e.g., the shape of the green plant and ghost effects around the pillar in Brandenburg, the cavity in Sacre. Besides, the transferred appearance is not realistic enough, e.g., sunshine on statues in Sacre, and the color of blue sky and grey roof in Trevi. Differently, our CR-NeRF introduces a cross-ray paradigm and therefore achieves more
realistic appearance modeling and reconstructs a consistent geometry by suppressing transient objects.

Comparison of appearance modeling. We investigate the appearance modeling ability of our CR-NeRF in Fig. 4. We observe that 1) CR-NeRF captures appearance information more accurately than Ha-NeRF, especially towards recovering appearances from images with high-frequency information, e.g., green sky, blue sky, red building, sunlight on the gate. 2) CR-NeRF successfully removes transient objects such as tourists and cars while retaining static objects such as roads and buildings.

5.2. Ablation Studies

Ablation of appearance module and transient module. We summarize the ablation studies of CR-NeRF on Brandenburg, Sacre, and Trevi dataset in Tab. 2. We observe that CR-NeRF-A and CR-NeRF-T outperform CR-NeRF-B. and CR-NeRF exceeds all variants, indicating the effectiveness of our Appearance Module and Transient Module.

Cross-ray manner and fusing level. We study the ef-
effectiveness of the cross-ray manner and the fusing level by comparing with our baseline CR-NeRF-R quantitatively in Tab. 1 and qualitatively in the appendix. From Tab. 1, CR-NeRF-R achieves a competitive performance on three datasets, which shows the superiority of leveraging various rays. Moreover, our proposed CR-NeRF outperforms CR-NeRF-R consistently on all datasets. We assume that compared with the cross-ray-point features, the granularity of the cross-ray features \( F_{cr} \) is closer to that of the image-level conditional features. Therefore, feature fusion is more effective. We provide qualitative results in the appendix.

### 5.3. Further Experiments

**Unseen appearance modeling.** Our proposed CR-NeRF is able to deal with unseen appearance images thanks to the ability of our cross-ray appearance modeling handler. As shown in Fig. 5, our CR-NeRF captures the whole range appearance (e.g., the blue and purple appearance in the last two columns in Brandenburg and Trevi fountain datasets) of the given style image more accurately compared with Ha-NeRF. Moreover, our CR-NeRF synthesizes a more consistent appearance than images generated by Ha-NeRF (e.g., the sudden bright light on the sky of the second column in Brandenburg dataset). Note that NeRF-W needs to optimize its appearance embedding on each test image by pixel-level supervision, thus NeRF-W cannot be directly applied to unseen appearance modeling.

**Inference time on multiple images.** When dealing with multiple images of various appearances with fixed camera position, the inference efficiency of our CR-NeRF exceeds Ha-NeRF significantly (i.e., 2.12 seconds vs 24.09 seconds in Tab. 3). The reason is that our CR-NeRF generates cross-ray features \( F_{cr} \) only once by using a NeRF backbone and synthesizes various appearances by fusing \( F_{cr} \) and appearance embedding of each image. In contrast, Ha-NeRF requires the use of its NeRF backbone for each estimation. For efficiency, we modify Ha-NeRF by saving its interim results. However, since the interim results of Ha-NeRF occupy a large amount of GPU memory beyond the capacity of the single TITAN Xp GPU, moving the results to the host memory requires additional I/O time.

**Transient objects handling.** We observe that simply masking common objects harms reconstruction performance. Specifically, we use a pre-trained DeepLabV3 and a pre-trained Mask R-CNN that produce promising segmentation results for common objects such as pedestrians and cars (we carefully choose the categories for estimation to avoid masking out static objects). However, performance degrades when combining CR-NeRF-A with these two networks (see Tab. 4). Considering that the transient handler of CR-NeRF is trained without the supervision of ground truth visibility maps, our estimated visibility maps are inevitably less accurate than the pre-trained network on the common objects (see the appendix for more details). We assume the definition of transient objects is still an open question and we leave it to our future work.
6. Conclusion

In this paper, we address novel-view synthesis from unconstrained images by considering the information of multiple rays within a scene. The unconstrained scenario introduces the varying appearances and transient objects in the images. We propose a novel cross-ray paradigm for the task by leveraging global interactive information across multiple rays. Guided by the paradigm, to address the variable appearance, we propose to represent information of multiple rays with cross-ray features and then inject an appearance of each image via fuse feature covariance of the rays and the image appearance. To handle transient objects, we propose a novel perspective of handling transient objects via image segmentation on multiple rays. Based on this, we estimate and grid sample a visibility map to pair with the rays. Extensive experimental results on large real-world datasets show the effectiveness of our proposed method.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC) (62072190), National Natural Science Foundation of China (NSFC) (61836003) (key project), Program for Guangdong Introducing Innovative and Entrepreneurial Teams 2017ZT07X183. and grid sample a visibility map to pair with the rays. Extensive experimental results on large real-world datasets show the effectiveness of our proposed method.

References


