

# NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Seri Lee

09/14/21

## Abstract

We present a method that achieves state-of-the-art results for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views. Our algorithm represents a scene using a fully-connected (non-convolutional) deep network, whose input is a single continuous 5D coordinate (spatial location  $(x, y, z)$  and viewing direction  $(\theta, \phi)$  and whose output is the volume density and view-dependent emitted radiance at that spatial location. We synthesize views by querying 5D coordinates along camera rays and use classic volume rendering techniques to project the output colors and densities into an image. Because volume rendering is naturally differentiable, the only input required to optimize our representation is a set of images with known camera poses. We describe how to effectively optimize neural radiance fields to render photorealistic novel views of scenes with complicated geometry and appearance, and demonstrate results that outperform prior work on neural rendering and view synthesis. View synthesis results are best viewed as videos, so we urge readers to view our supplementary video for convincing comparisons.

## 1 Introduction

In this work, we address the long-standing problem of view synthesis in a new way by directly optimizing parameters of continuous 5D scene representation to minimize the error of rendering a set of captured images.

Figure 1: We present a method that optimizes a continuous 5D neural radiance field representation (volume density and view-dependent color at any continuous location) of a scene from a set of input images. We use techniques from volume rendering to accumulate samples of this scene representation along rays to render the scene from any viewpoint. Here, we visualize the set of 100 input views of the synthetic Drums scene randomly captured on a surrounding hemisphere, and we show two novel views rendered from our optimized NeRF representation.

We represent a static scene as a continuous 5D function that outputs the radiance emitted in each direction  $(\theta, \phi)$  at each point  $(x, y, z)$  in space, and a density at each point which act like a differential opacity controlling how much radiance is accumulated by a ray passing through  $(x, y, z)$ . Our method optimizes a deep fully connected neural network without any convolutional layers (often referred to as a multilayer perceptron or MLP) to represent this function by regressing from a single 5D coordinate  $(x, y, z, \theta, \phi)$  to a single volume density and view-dependent RGB color. To render this neural radiance field (NeRF) from a particular viewpoint we march camera rays through the scene to generate a sampled set of 3D points, use those points and their corresponding 2D viewing directions as input to the neural network to produce an output set of colors and densities, and use classical volume rendering techniques to accumulate those colors and densities into a 2D image. Because this process is naturally differentiable, we can use gradient descent to optimize this model by minimizing the error between each observed image and the corresponding views rendered from our representation. Minimizing this error across multiple views encourages the network to predict a coherent model of the scene by assigning high volume densities and accurate colors to the locations that contain the true underlying scene content.

We find that the basic implementation of optimizing a neural radiance field representation for a complex scene does not converge to a sufficiently high-resolution representation and is inefficient in the required number of samples per camera ray. We address these issues by transforming input 5D coordinate with a positional encoding that enables the MLP to represent high frequency functions, and we propose a hierarchical sampling procedure to reduce the number of queries required to adequately sample this high-frequency scene representation.

Our approach inherits the benefits of volumetric representations: both

can represent complex real-world geometry and appearance and are well suited for gradient-based optimization using projected images. Crucially, our method overcomes the prohibitive storage costs of discretized voxel grids when modeling complex scenes at high-resolutions. In summary, our technical contributions are: An approach for representing continuous scenes with complex geometry and materials as 5D neural radiance fields, parameterized as basic MLP networks. A differentiable rendering procedure based on classical volume rendering techniques, which we use to optimize these representations from standard RGB images. This includes a hierarchical sampling strategy to allocate the MLP’s capacity toward space with visible scene content. A positional encoding to map each input 5D coordinate into a higher dimensional space, which enables us to successfully optimize neural radiance fields to represent high-frequency scene content.

We demonstrate that our resulting neural radiance field method quantitatively and qualitatively outperforms state-of-the-art view synthesis methods, including works that fit neural 3D representations to scenes as well as works that train deep convolutional networks to predict sampled volumetric representations. As far as we know, this paper presents the first continuous neural scene representation that is able to render high-resolution photorealistic novel views of real objects and scenes from RGB images captured in natural settings.

## 2 Neural Radiance Field Scene Representation

We represent a continuous scene as a 5D vector-valued function whose input is a 3D location  $x = (x, y, z)$  and 2D viewing direction  $(\theta, \phi)$ , and whose output is an emitted color  $c = (r, g, b)$  and volume density  $\sigma$ . In practice, we express direction as a 3D Cartesian unit vector  $d$ . We approximate this continuous 5D scene representation with an MLP network  $F_{\Theta} : (x, d) \rightarrow (c, \sigma)$  and optimize its weights  $\Theta$  to amp from each input 5D coordinate to its corresponding volume density and directional emitted color.

We encourage the representation to be multiview consistent by restricting the network to predict the volume density  $\sigma$  as a function of only the location  $x$ , while allowing the RGB color  $c$  to be predicted as a function of both location and viewing direction. To accomplish this, the MLP  $F_{\Theta}$  first processes

Figure 2: An overview of our neural radiance field scene representation and differentiable rendering procedure. We synthesize images by sampling 5D coordinates (location and viewing direction) along camera rays, feeding those locations into an MLP to produce a color and volume density, and using volume rendering techniques to composite these values into an image. This rendering function is differentiable, so we can optimize our scene representation by minimizing the residual between synthesized and ground truth observed images.

the input 3D coordinate  $x$  with 8 fully-connected layers (using ReLU activations and 256 channels per layer), and outputs  $\sigma$  and a 256-dimensional feature vector. This feature vector is then concatenated with the camera ray’s viewing direction and passed to one additional fully-connected layer (using a ReLU activation and 128 channels) that output the view-dependent RGB color.

See Fig.3 for an example of how our method uses the input viewing direction to represent non-Lambertian effects. As shown in Fig.4, a model trained without view dependence (only  $x$  as input) has difficulty representing specularities.

### 3 Volume Rendering with Radiance Fields

Our 5D neural radiance field represents a scene as the volume density and directional emitted radiance at any point in space. We render the color of any ray passing through the scene using principles from classical volume rendering. The volume density  $\sigma(x)$  can be interpreted as the differential probability of a ray terminating at an infinitesimal particle at location  $x$ . The expected color  $C(r)$  of camera ray  $r(t) = o + td$  with near and far bounds  $t_n$  and  $t_f$  is:

$$C(r) = \int_{t_n}^{t_f} T(t) \sigma(r(t)) c(r(t), d) dt \quad (1)$$

, where

$$T(t) = \exp\left(-\int_{t_n}^t \sigma(r(s)) ds\right) \quad (2)$$

A visualization of view-dependent emitted radiance. Our neural radiance field representation outputs RGB color as a 5D function of both spatial position  $x$  and viewing direction  $d$ . Here, we visualize example directional color distributions for two spatial locations in our neural representation of the Ship scene. In (a) and (b), we show the appearance of two fixed 3D points from two different camera positions: one on the side of the ship (orange insets) and one on the surface of the water (blue insets). Our method predicts the changing specular appearance of these two 3D points, and in (c) we show how this behavior generalizes continuously across the whole hemisphere of viewing directions.

The function  $T(t)$  denotes the accumulated transmittance along the ray from  $t_n$  to  $t$ , i.e., the probability that the ray travels from  $t_n$  to  $t$  without hitting any other particle. Rendering a view from our continuous neural radiance field requires estimating this integral  $C(r)$  for a camera ray traced through each pixel of the desired virtual camera.

We numerically estimate this continuous integral using quadrature. Deterministic quadrature, which is typically used for rendering discretized voxel grids, would effectively limit our representation’s resolution because the MLP would only be queried at a fixed discrete set of locations. Instead, we use a stratified sampling approach where we partition  $[t_n, t_f]$  into  $N$  evenly-spaced bins and then draw one sample uniformly at random from within each bin:

$$t_i \sim \mathcal{U}[t_n + \frac{i-1}{N}(t_f - t_n) + \frac{i}{N}(t_f - t_n)] \quad (3)$$

Although we use a discrete set of samples to estimate the integral, stratified sampling enables us to represent a continuous scene representation because it results in the MLP being evaluated at continuous positions over the course of optimization. We use samples to estimate  $C(r)$  with the quadrature rule discussed in the volume rendering review.

$$\hat{C}(r) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i \quad (4)$$

where

$$T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \quad (5)$$

where  $\delta_i = t_{i+1} - t_i$  is the distance between adjacent samples. This function for calculating  $\hat{C}(r)$  from the set of  $(c_i, \sigma_i)$  values is trivially differentiable and reduces to traditional alpha compositing with alpha values  $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ .