Relightable Neural Radiance Fields for Novel View Synthesis

Malena I. Mahoney
mahon479@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

This paper describes relighting neural radiance fields for novel view synthesis. View synthesis is the problem of using input images with corresponding camera angles to produce a photorealistic 3D model of an environment and its objects. Neural radiance fields (NeRFs) were created as a solution to view synthesis. Neural radiance field models work well for generating realistic 3D models from 2D image inputs; however, they do not support changing the lighting or placing the objects from the input images into different environments. The problem comes from the fact that NeRFs rely on a neural network that is essentially overfitted to the original environment used in the training. This means an object in a given scene cannot be placed into a different scene using the NeRF neural network model. A new model, relightable neural radiance fields (ReNeRFs), has been proposed to combat this issue. ReNeRFs have the ability to control the lighting of an object and place it into novel environments using an image-based relighting approach.

Keywords: view synthesis, neural networks, NeRF, ReNeRF, 3D scene representation, image-based relighting

1 Introduction

Imagine you are designing a video game that includes a stuffed dog as an object in your virtual world. Instead of creating the stuffed dog virtually, you would like to use a real-life stuffed dog as a basis for your video game dog. To achieve this, you must create some sort of 3D model of that dog so you can easily view it from a wide variety of viewpoints within your game.

View synthesis is the problem of using input images with corresponding camera angles to produce a photorealistic three-dimensional model of an environment and its objects. Past solutions to this problem have been unsuccessful in producing photorealistic quality for new viewpoints, which is desired in view synthesis. Reflective properties of materials and complex geometry, like skin and fur, are the keys to photorealism in scenes but can be hard to achieve. [6]

Neural radiance fields (NeRFs) represent realistic 3D scenes with impressive photorealism given simple images from the real world. With the current NeRF models, a video taken via a smartphone is all it takes to make a visual representation of the 3D world. The scenes created by NeRFs are interactive due to their ability to use controlled and arbitrary viewpoints for synthesizing new imagery within a 3D scene. Many applications such as feature films, video games, and virtual reality rely on photorealistic 3D scenes to provide audiences and users with the proper immersive experience. This can be accomplished using NeRFs simply by populating the virtual environments of these applications with real-world photos and then creating a NeRF model. [6]

The current NeRF models produce great 3D scenes with fully controllable viewpoints; however, the scenes are limited in controlling certain environmental features like lighting. Control over the lighting of an object would be considered a desirable trait for a 3D scene in the applications mentioned above as it makes the scene adaptable to various lighting conditions that can occur within a film or video game. A new technique, relightable neural radiance fields (ReNeRFs), overcomes this limitation as it can relight the scene for any lighting condition in addition to allowing full control over the camera viewpoint. ReNeRF is based on the technique of image-based relighting (IBRL). This approach requires more than just capturing a video with a smartphone, as was done for the standard NeRF model inputs. It involves capturing many photos of a scene under one-light-at-a-time (OLAT) conditions and then combining those images to produce the desired lighting condition. OLAT conditions are like different lamps in various spaces of a room and only one being turned on at a time. This model is learned by employing in-studio photogrammetry[1] with only a few area light sources. [7]

Figure 1 (left) shows input images for a ReNeRF model; notice the different lighting conditions in each image. The shadow moves around the toy dog based on where the light source is located. Images like these are inputted into the ReNeRF model to produce the images found on the right of Figure 1. The dog can be placed in different environments, each with distinct lighting conditions, as seen in the top three scenes on the right of Figure 1. The middle scene shows the ability of novel environment placement as well as changing the viewpoint. The bottom scenes show the idea of nearfield lighting which will be discussed further later in Section 4. [7]

[1] Photogrammetry is the process of capturing images of an object from many different angles in a controlled studio environment in order to produce a 3D model of the object. [3]

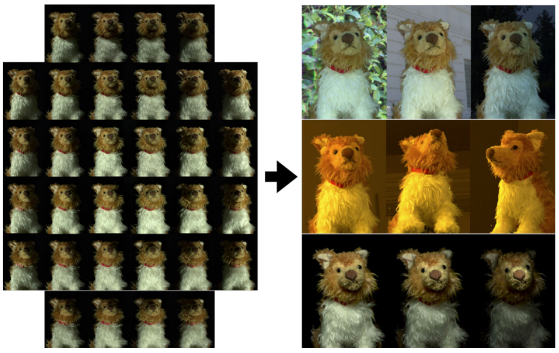

Figure 1. Relightable neural radiance field trained on images of an object with various views and lighting conditions (left). A trained relightable neural radiance field provide full control over viewpoint and lighting conditions along with the ability to place the object into novel environments.[7]

In the next section I'll describe the basics of neural networks (Section 2), which are the basis of both NeRF and ReNeRF models. Section 3 describes standard NeRF models and volume rendering, and Section 4 covers the ReNeRF model proposed by Xu et al. [7]. Results (Section 5) and conclusions (Section 6) will conclude this paper.

2 Neural Networks

The solutions to view synthesis presented in this paper use neural networks to model real-life objects. A neural network is a computer model that processes information found in input data and produces outputs desired by the creator. Modeled loosely after the human brain, neural networks process and move data using interconnected neurons. Each neuron takes inputs and performs calculations to produce an output that is passed to its connected neurons. There are many different types of neural networks but for the purposes of this paper, we will focus on multilayer perceptrons (MLPs). This section will describe multilayer perceptrons and their training process as presented in Sejal Jaiswal [2].

2.1 Multilayer Perceptrons

Multilayer perceptrons organize the neurons into different layers: an input layer, one or more hidden layers, and an output layer. The input layer neurons take the original data and pass it to the hidden layers. The hidden layer and output layer neurons contain functions called activation functions. These functions determine the range of the output and its behavior in response to different input values. Activation functions help the model learn complex patterns in the data. The neurons in the hidden layers perform their respective activations functions and pass the results to the output layer. The output layer takes the values provided by the last hidden layer and applies another activation function to compute the final output value.

As mentioned above, the neurons in neural networks are connected and each of these connections have a weight attached to them. When data leaves a neuron and enters another, a weight is applied to the value. These weights correspond to how strong the connection is between each neuron. If the weight multiplying the first neuron's output is high, it will have more of an influence on the second neuron's input. The initial weight values are set by the creator of the neural network, usually at random and the values are then updated through a process called training.

2.2 Neural Network Training

Deep neural networks go through a training process to ensure that the model produces accurate results. Training data with inputs and their expected outputs is provided to the neural network during this process. This means the desired output is known and can be compared to what is predicted by the model to find differences. The goal is that once training is complete, the model can take new inputs (not seen in the training data) and produce the desired output.

Multilayer perceptrons are trained using backpropagation which is an algorithm that adjusts weights with the goal of minimizing the loss function. A loss function measures the difference between the model's predictions and the true target values from the training data. To begin this training process, the training inputs are sent through the model with the initial weights. The loss function is then computed using the outputs and the data is sent backwards through the model to calculate the gradient of the loss function. The gradient tells the model how to adjust the weights to minimize loss and they are updated accordingly. The training inputs are sent back through the model and the process is repeated until the loss is minimized sufficiently.

3 Neural Radiance Fields

As described by Mildenhall et al. [6], neural radiance fields (NeRFs) are photorealistic continuous scenes generated by a multilayer perceptron that is trained on multiple camera angles of a given scene. The MLP is used to generate the color and density of each new view and then a volume rendering technique uses the MLP outputs to render the new view of the scene. Continuity of a scene is important for virtual flythroughs where the user is changing viewpoints constantly. The continuity allows for the photorealistic quality of new viewpoints as it does not store captures of the scene but rather updates the scene in real time given the updated views. The scenes created by NeRFs are meant to be interactive so a user can change to any viewpoint at any time and get continuous visual updates. [6]

3.1 Scene Generation

The neural radiance field scene is generated by a multilayer perceptron combined with volume rendering. The MLP takes

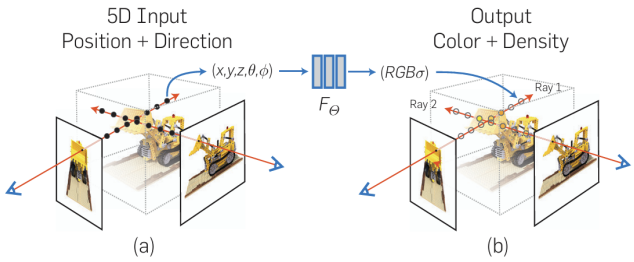

Figure 2. An overview of our neural radiance field scene representation. The 3D coordinates and viewing direction are inputted into the model. The output is the color and density at each of the inputted points. Volume rendering uses these outputs to portray the final pixel of that part of the scene. [6]

a five-dimensional input consisting of a three-dimensional location $x = (x, y, z)$ and the 3D view direction (θ, ϕ). The output is a volume density σ and a color $c = (r, g, b)$. Once this MLP is trained (see Section 2.2), the volume density and directional emitted color can be generated for any given 3D point and viewing direction within the scene. Volume density of a point is how solid or transparent it is. For example, a point that is in empty space within the scene has a low or non-existent density since there is no object there, and a point found on an object in the scene has a higher density. A higher density means the emitted color will actually be viewable to the human eye. [6]

Each input for the MLP is generated by placing a camera ray from the given viewing direction, as shown in Figure 2, and collecting the set of three-dimensional coordinates along that ray. The neural network outputs the set of colors and densities for each input which are then rendered into a three-dimensional scene using volume rendering techniques. [6]

3.2 Volume Rendering

Volume rendering allows the scene's color to be viewable by users. Using the volume densities and directionally emitted radiance generated by the neural network, volume rendering creates the three-dimensional image output. Volume rendering estimates the amount of light that makes it out of the volume at each camera angle.

For each camera viewpoint there is a ray that runs through the scene, each with their own discrete points. Figure 2 (left) shows these discrete points as black dots along the ray. Volume rendering estimates the volume density and color on each point along that array. Figure 2 (right) shows the discrete points with their color and densities. The densities represent how solid or translucent the point is. Once the colors and densities along each point on the ray have been computed, the points' colors and densities are combined to produce one point with the proper color and density. This single point is a pixel that the viewer sees. [5]

4 Relightable Neural Radiance Fields

NeRFs provide a solution to novel view synthesis but still have limitations. Each NeRF can only produce the scene represented in the training images. This means things like the lighting and the environment of the scene cannot be changed. For real-world applications, like films and video games, it is important to have the option of changing things like the lighting and environment of a scene depending on what a character or player does. In a video game, for example, a player may move a flashlight around a scene, continuously changing the lighting conditions and angles at which the objects within the scene are being viewed. A player may also move an object into new environments within the virtual world that has a different lighting condition than its previous placement. Relightable radiance fields (ReNeRFs) are capable of relighting objects and placing them in novel environments.

The relightable radiance field (ReNeRF) model as explained in Xu et al. [7] uses a technique called image-based relighting (IBRL). The idea behind this technique is to light an object from different one-light-at-a-time (OLAT) directions and produce two-dimensional images of the object under novel light conditions. Lighting the object with a few small area-OLATs and constructing a NeRF model that disentangles point-OLATs allows for the relighting of the object from novel viewpoints and light positions. All material in this section will be based on Xu et al. [7] unless otherwise noted.

The following sections include information on capturing images and data preparation (Section 4.1), image generation (Section 4.2) using rendering, neural network architecture (Section 4.3) for the ReNeRF model, more general lighting (Section 4.4), and training (Section 4.5).

4.1 Capturing and Preprocessing Input Images

The input images for ReNeRFs are captured using existing photogrammetry techniques. This means the images are taken within a controlled studio environment. The setup for collecting these input images includes 32 area light sources (LED bars) and 10 video cameras, arranged around the frontal hemisphere of a small capture volume, as seen in Figure 3 (left). A total of 34 lighting conditions are captured by the LED bars, operating as OLATs, including "full ON" and "full OFF" conditions. An example of 34 OLAT lighting conditions can be seen in Figure 3 (right). To acquire a depth estimation that will be used in training (Section 4.5), there are 4 cameras referred to as stereo pairs that capture the depth of the object.

4.2 Image Generation

As discussed in Section 3.1, the NeRF model takes a three-dimensional location and the 3D view direction from the input images. To allow for control over lighting, the ReNeRF model takes those same inputs along with a collection of small point lights. The point lights are found on the area light sources (LED bars) described above in Section 4.1. Each LED

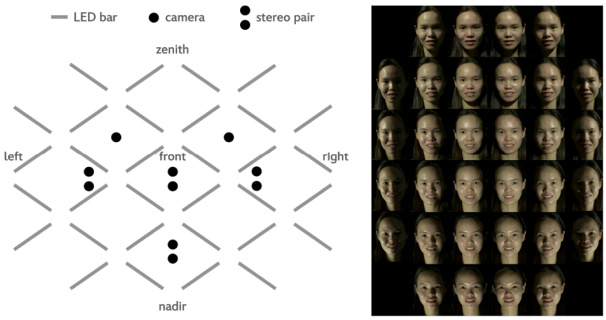

Figure 3. Layout of 10 video cameras and 32 area lights (LED bars) placed in front of the object (left). 32 area OLAT-images are captured using one-light-at-a-time technique (right). [7]

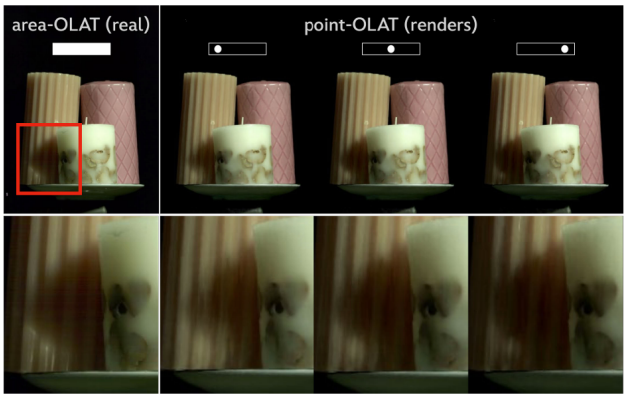

Figure 4. Image captured under the light of a single LED bar (top left), and 3 point light sources (top right). There are slight changes in the shadows using the different point light sources showing more complex lighting conditions. [7]

bar has three point lights on it: one on the left, one on the right, and one in the middle. These point light sources allow for the model to handle more complex lighting conditions as seen in Figure 4. The model calculates the radiance, which is the light that is emitted, reflected, and transmitted in the scene for each of the point lights, and incorporates that into the volume rendering procedure from the NeRF model to produce the relightable scene.

Figure 5 shows the light dependent volume rendering used in ReNeRF. The pixel ray is defined as ω_0, the discrete points are x_i, p is the individual point light source, and the light direction at each point is ω_i. The RGB value for each pixel is a combination of the x_i points at their corresponding ω_i values along the pixel ray. The RGB is computed for every pixel in the scene to provide the completely interactive virtual scene. If there are multiple point light sources, the contributions of those sources are added together for each point along the pixel array to produce the final RGB for the pixel. This provides full control over lighting conditions.

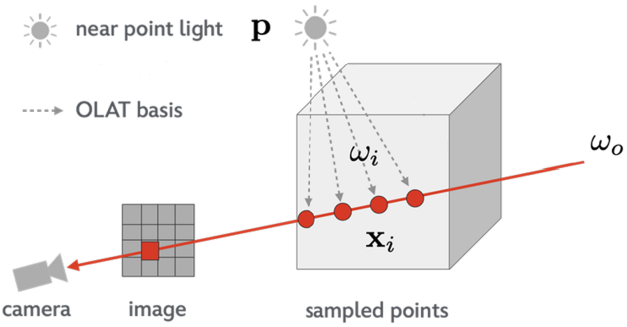

Figure 5. Volumetric rendering with light direction ω_i, pixel ray ω_0, discrete points x_i, and individual point light source p. The gray cube represents the full scene. Adapted from [7].

4.3 Architecture of the Relightable Neural Radiance Field Model

Similar to previous NeRF models, ReNeRF uses a multilayer perceptron (MLP) to discover the color and densities of each point in the scene to generate the complete scene. The difference is the type of color that is generated. Standard NeRF models output a simple color whereas the ReNeRF model provides a more complex color output. The neural network for ReNeRF has the same inputs as standard NeRF models (see Section 3.1) with the addition of the point light sources found on the LED bars. The architecture of the MLP, as seen in Figure 6, includes three main parts: a NeRF MLP, an OLAT MLP, and a spherical codebook of learned OLAT codes.

The NeRF MLP comes from the standard NeRF model described in Section 3.1. The ReNeRF model uses the same density output as the standard NeRF. The 3D point and viewing direction are inputted into the NeRF MLP and it outputs the density and a vector of neural features that will be used by the OLAT MLP to produce the diffuse and specular colors.

The diffuse color and specular colors are decided by the OLAT MLP with the help of learned point-OLAT code derived from the point light sources. This point-OLAT code is learned through training and represents the lighting conditions at each point in the scene. The full set of lighting conditions are stored in the spherical codebook. Think of the spherical codebook as a globe that tracks how the sun lights up different cities. On this globe, the sun is the point light source p, and x_i, a 3D point in the scene, is a city. The model asks the codebook for $\omega_i(x_i, p)$ which is the direction of sunlight from p to x_i. The codebook provides the lighting condition (OLAT code) at that city given that direction of sunlight. This lighting information is passed to the OLAT MLP which processes the data to generate the color of the given point, viewing direction, and lighting direction. The color generated for each point by the OLAT MLP is a combination of the diffuse and specular colors which will be described below.

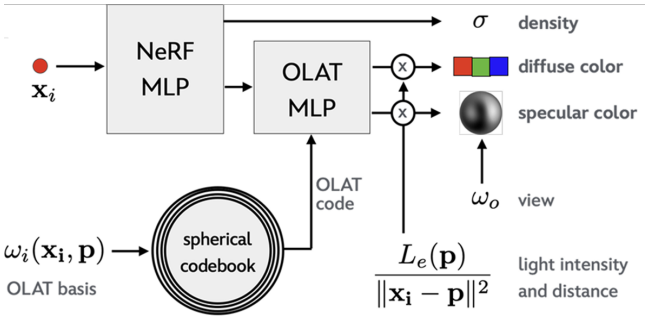

Figure 6. Architecture includes standard NeRF MLP, an OLAT MLP that is conditioned on NeRF features and a learned spherical codebook associated with ω_i. The output comprises of density, diffuse color, and specular color. Adapted from [7].

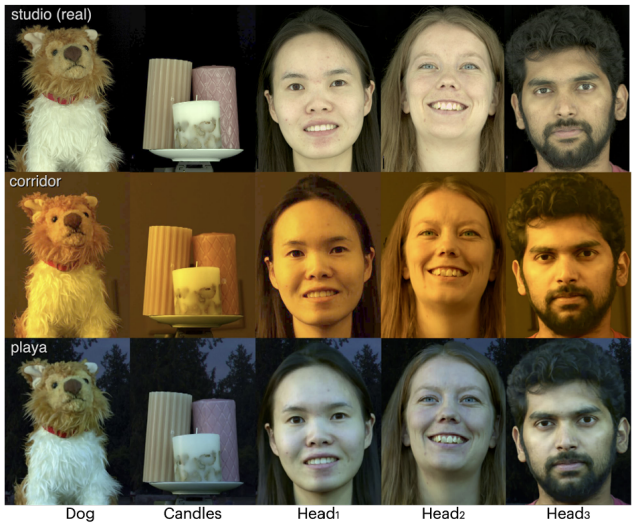

Figure 7. Object rendered under two novel lighting environments using the ReNeRF model and its environmental map abilities. The top row shows the FULL-ON capture of the object taken in the controlled studio setting to be used for reference. [7]

As mentioned previously, the ReNeRF model outputs a more complex form of color than the standard NeRF model. Specifically, the model generates the radiance of each point which is the transmittance, reflectance, and emittance of light. The density output corresponds to the transmittance which is the amount of light that passes through the object, so how solid or translucent the object is. Diffuse color and specular color cover the reflectance aspect of radiance. Notice in Figure 6, specular color is view dependent since the light reflecting off of a smooth surface will be going in a different direction depending on how you are looking at it. Think of a clear lake with a mountain reflection, different parts and colors of the mountain will be seen depending on where the viewpoint is. The emittance part of radiance comes from the fraction $L_e(p)/\|x_i - p\|^2$ seen in Figure 6. The numerator is the emittance of the point light source and the denominator is the distance. This fraction allows the model to handle nearfield lighting which is the idea that the closer the light gets to an object, the more shadowed the area around the object will be.

4.4 Novel Environments and Area Lights

The ReNeRF architecture shown in Figure 6 can only render a scene with a single point light p. Image-based relighting (IBRL) allows for renderings with an area light or novel environments. IBRL makes a scene for each point light source and combines them to create the final scene. The combinations of the different scenes occur in image-space where the diffuse and specular color outputs are added together. This means the NeRF MLP in Figure 6 evaluates each 3D point only once and those outputs are sent through the OLAT MLP multiple times, once for each point source.

4.5 Training

Each of the ReNeRF models are trained on a total of 320 images for each object, 32 different lighting conditions taken from 10 different camera views. The images are taken using the technique described in Section 4.1.

As discussed in Section 2.2, loss functions guide the model to accurate results for each parameter. ReNeRFs have three parts to minimize in their loss function: rendering loss, density field constraints, and regularization losses. Rendering loss ensures that each pixel color generated by volume rendering is accurate. The density field constraints check whether the model is correctly identifying empty space versus solid objects. Regularization losses are implemented to prevent the model from overfitting to the training data. An overfitted model essentially learns too much from training data so it can only perform well on the training data and not with new input data.

ReNeRFs are trained over 200,000 iterations meaning the training data are inputted into the model and the weights are updated 200,000 times. The first 150,000 have a single point light source at the center of each LED bar. Two additional point light sources are added on the left and right of the LED bar for the last 50,000 to make the model adaptable to more complex lighting conditions.

5 Results

Trained on both real and synthetic datasets, the ReNeRF method demonstrated success in generating photorealistic renderings of various objects under varying viewpoints and lighting conditions. ReNeRF extended the novel view synthesis abilities of NeRFs by utilizing the technique of image-based relighting and training on images captured via the in-studio photogrammetry setup shown in Figure 3 (left).

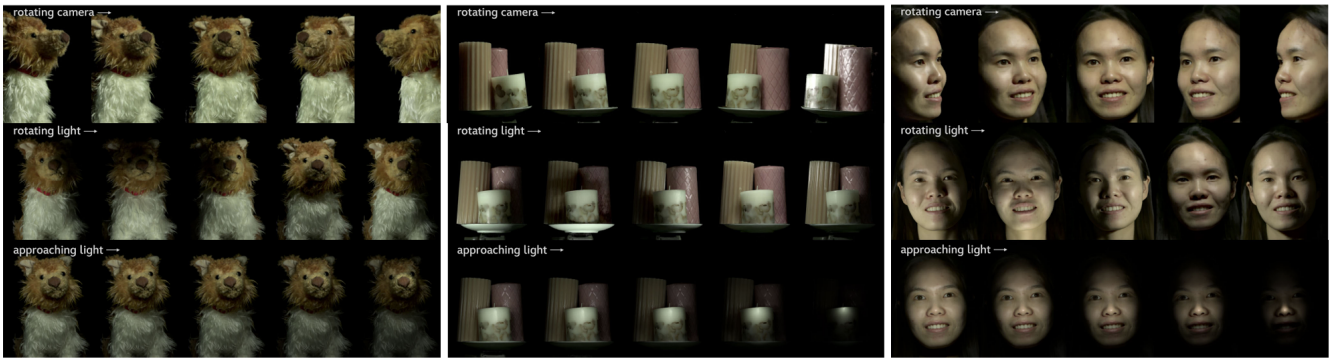

Figure 8. Scenes rendered by ReNeRF model: a furry dog, a plate with candles, and a human head. Shows the ability of ReNeRFs to generate scenes under novel views, novel lighting conditions, and under a near point light approaching the object. [7]

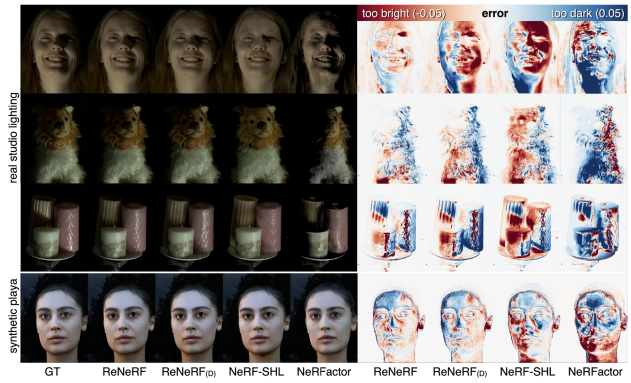

Figure 9. Measure of relighting error on real datasets (top) and a synthetic dataset for a novel environment (bottom). The more pronounced the color in the right panel, the more error that rendering has. ReNeRFs generally have smaller errors compared to the other models. [7]

ReNeRF is able to portray complex materials including shiny eyes, fur, semi-translucent skin, etc. with different lighting conditions. Realistic renderings with different lighting conditions of objects containing these complex materials can be found in Figure 8. Notice how well the fur of the toy dog is captured for every lighting condition and viewpoint. The data-driven nature of ReNeRF and its lack of simplifying assumptions for lighting allow continuously varying lighting direction and light distance from the object during virtual fly-throughs. Another limitation of NeRFs that was fixed with ReNeRFs is the ability to place objects in new environments that have their own distinct lighting. Figure 7 shows different objects placed in different environments. The top row shows the images taken via in-studio photogrammetry and the following two show novel environments. The middle column of Figure 7 provides a great example of how well the ReNeRF model handles semi-translucent skin. For more examples of ReNeRF results, see [1].

For comparison, ReNeRF was evaluated alongside three other models: ReNeRF$_{(D)}$, NeRF-SHL [4], and NeRFactor [8]. The main difference between these models and the ReNeRF model discussed in this paper is that ReNeRF$_{(D)}$, NeRF-SHL, and NeRFactor all assume distant lighting meaning they do not calculate the distance between the light source and the object. The far left column of Figure 9 is the ground truth to which each model's rendering is compared to. Each pixel in each rending is compared to the ground truth to produce the errors shown on the right of Figure 9. The more pronounced the color on the error plot, the more error it has. Notice the ReNeRF model does perform better than the others as it generally has less error. [7]

6 Conclusion

This paper described relightable neural radiance fields (ReNeRFs) as an improvement to standard NeRF in novel view synthesis. Standard NeRFs generate interactive virtual scenes with high photorealism given only images and their corresponding camera angle. Though very effective in producing 3D scenes, scenes generated via NeRF cannot be placed under different lighting conditions. Using multilayer perceptrons and image-based relighting, ReNeRF scenes have full lighting control (Figure 8) and are able to be placed in any given environment as seen in Figure 7. Trained on images captured in a controlled studio setting under various lighting conditions, ReNeRF scenes are free of lighting and environment assumptions. A trained ReNeRF model can create a continuous interactive virtual scene that supports real time updates with changing viewpoints and lighting conditions. ReNeRFs are practical assets to applications with virtual environments such as films, video games, and virtual reality as the scenes produced can easily populate such environments. [7]

Acknowledgments

I would like to thank my advisor, Nic McPhee, along with Brian Mitchell, for all the help with the research and development of this paper.

References

[1] DisneyResearchHub. 2023. *ReNeRF:Relightable Neural Radiance Fields with Nearfield Lighting*. Youtube. https://www.youtube.com/watch?v=iPBesfjNVXM

[2] Sejal Jaiswal. 2024. Multilayer perceptrons in Machine Learning: A comprehensive guide. https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning

[3] Handy Kosasih. 2024. What Is Photogrammetry? Understanding the Basics, Methods, and Applications. https://interscaleedu.com/en/blog/technology/what-is-photogrammetry/

[4] Gengyan Li, Abhimitra Meka, Franziska Mueller, Marcel C. Buehler, Otmar Hilliges, and Thabo Beeler. 2022. EyeNeRF: a hybrid representation for photorealistic synthesis, animation and relighting of human eyes. *ACM Transactions on Graphics* 41, 4 (July 2022), 1–16. https://doi.org/10.1145/3528223.3530130

[5] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. *[ECCV 2020] NeRF: Neural Radiance Fields (10 min talk)*. Youtube. https://www.youtube.com/watch?v=LRAqeM8EjOo

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (Dec. 2021), 99–106. https://doi.org/10.1145/3503250

[7] Yingyan Xu, Gaspard Zoss, Prashanth Chandran, Markus Gross, Derek Bradley, and Paulo Gotardo. 2023. ReNeRF: Relightable Neural Radiance Fields with Nearfield Lighting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 22524–22534. https://doi.org/10.1109/ICCV51070.2023.02064

[8] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. 2021. NeRFactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics* 40, 6 (Dec. 2021), 1–18. https://doi.org/10.1145/3478513.3480496