Simplifying NeRF, it's NeRF or nothing

Wojciech Mazurek, Voke Brume, Shivam

December 11, 2022

Contents

1 Abstract **3**

2 Contributions **4**

2.1 Wojciech Mazurek: 4

2.2 Voke Brume: 4

2.3 Shivam: 4

3 Introduction **5**

4 Related Work **6**

5 Approach **6**

5.1 Neural Radiance Field as a continuous 3D scene representation 7

5.2 NeRF in a nutshell 8

5.2.1 Hierarchical sampling 9

5.2.2 Positional encoding 10

5.2.3 Loss function 10

5.2.4 Evaluation metrics used 11

6 Experiment **11**

6.1 Results 12

6.2 Analysis 14

7 Conclusion **14**

7.1 Future Work 15

1 Abstract

Novel view synthesis can be a complex and precise science. Camera calibration and retrieving results are rarely perfect. With the use of neural radiance fields (NeRF) and programs such as Agisoft Metashape, the entire process can be greatly simplified while still being remarkably accurate with a small data set. By combining these two processes, the ability of anyone to create photo-realistic 3D models from real-life images becomes almost trivial.

2 Contributions

2.1 Wojciech Mazurek:

1. Assisted in the generation of data
2. getting NeRF to run and generating views
3. Background research
4. Introduction section
5. Related Work section
6. Future Work section

2.2 Voke Brume:

1. Background research
2. Generating easy-to-view images
3. Experiment section
4. Paper review

2.3 Shivam:

1. Assisted in the generation of data
2. Getting NeRF to run and generating views and final rendered output
3. Background research
4. Approach Section
5. Experiment Section
6. Conclusion section

3 Introduction

In the paper by Mildenhall et al. [1], this was the first time that neural radiance fields (NeRF) were used to synthesize a view of an object. This paper is also the basis of our research. this problem, the ”novel view synthesis” task, has been around since the early nineties [2] when the goal was to just place basic texture maps on 3d objects and view interpolation. NeRFs began in 2020, using 5D vectors, 3D coordinates, and 2D viewing angles to ”generate novel views with conventional volume rendering techniques.” [3] since the 2020 work, there have been thousands of papers written on how NeRFs can better the creation of these views, from their use in VR environments[4] to their use in the medical field to better reconstruct CT scans. [5] the use of NeRFs has been instrumental in developing many 3D technologies.

The basic idea of a NeRF is the ability for it to optimize volumetric scenes on only a small set of images. [6] A set of images a little as 25 can produce results equivalent to using a few hundred images using other techniques. [1] according to their original paper, ”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”. This was the first time that this technique was ever employed and it has spawned many new research possibilities.

These NeRFs have made creating models much easier and simpler than ever. However, the problem is that this process requires using a calibrated camera and the exact coordinates and parameters. Here we propose a method of using NeRFs without calibrating any cameras; with the use of the Agisoft Metashape [7] program, typically used for game development, we can generate camera coordinates and transformation matrices without the need for camera calibration. With this program, we can get camera coordinates and then use NeRF to generate an even more accurate 3D model from this. While using this technology still takes some time, on a generally stronger rig, for example, using an RTX 2080, 30 images could be converted and trained within 20 minutes; something even stronger would be able to process this quicker. While initially, that may not sound as impressive, previous processes may have been done by hand or taken some programs multiple hours when wanting the same level of detail when not using light detection and ranging (LiDAR). [1, 3]

With this technology and its ease of use, almost anyone can create very accurate models in a relatively short amount of time. This means that the development of 3D environments, and historical preservation and study, can become much easier. By using NeRFs in combination with Metashape, the process can now be simplified so much that no matter the pictures were taken, a 3D environment can still be built from those images without prepossessing the system and having the camera information.

4 Related Work

Only a year after the initial development of NeRFs, an idea was posed to use NeRFs without any known camera parameters. [6] This would greatly simplify the use of NeRFs, allowing the ability to crowd-source image data since camera parameters are typically unknown when taking pictures when not in a lab setting. This allows for easier use of NeRFs, but it is still imperfect. The process only considers forward-facing scenes and does not account for trying to get a full view of an environment.

Other researchers focused on the use of NeRFs for dynamic scenes. This would mean that NeRFs may also be used with another dimension, time. [8] The way that this was done was by splitting the learning process into two stages "one that encodes the scene into a canonical space and another that maps this canonical representation into the deformed scene at a particular time." for many environments it is not practical to place objects in an unchanging location and take pictures with no changes. This, for example, would be very useful to use on moving items, such as animals or cars. It may be more practical to keep the camera still, and move the item around and take the images that way. This paper requires removing the background from the images and is still just a proof of concept, but it may be beneficial in the future.

Another use for NeRFs that has been developed is audio-driven synthesis or AD-NeRF.[9] This allows the user to create a mimic of someone talking by just using audio clips and images, essentially creating something similar to a deep fake but looking even more accurate. the research allowed the creation of "high-fidelity and natural results" as well as "support[ing] free adjustment of audio signals, viewing directions, and background images." The accuracy of the results was highly realistic and, as a result, somewhat frightening. What can be done with only a few videos of someone talking?

NeRFs are a very powerful tool for 3D image generation. with a lot of research being done on them. Currently, additional research is also being done in many other places. Form their use in video games [10], medicine [5], and many other fields [11, 12, 13] it is clear that NeRFs are very powerful and useful in many aspects. Currently they are just beginning to be explored, and the easier it is to use and develop with NeRFs, the better technology can get.

5 Approach

Imagine being able to record a 3D scene and afterward watch it from various angles to witness the action as it was happening at the moment of capture. We are accustomed to taking 2D pictures or films, which are then quickly stored on our phones or on the cloud. The equivalent procedure for 3D capture, however, is quite time-consuming. In the past, this process has involved capturing

numerous pictures of the area, using photogrammetry to create a detailed surface reconstruction, and then physically cleaning everything up. However, the results may be astounding and have been used, for instance, in recent interactive articles (like in the New York Times), to create a sense of place that is otherwise impossible with 2D imagery.

Following an amazing article on neural radiation fields or NeRF, one technology in particular—neural volume rendering—exploded onto the scene in 2020. With several photographs as input, this innovative technology creates a compact representation of the 3D scene in the form of a deep, fully connected neural network. Then, using this model, arbitrary perspectives of the scene can be rendered with great accuracy and detail.

5.1 Neural Radiance Field as a continuous 3D scene representation

Representing 3D surfaces

Explicit:

Figure 1: Voxels

Figure 2: Point clouds

Figure 3: Mesh

Definition of a radiance field: A radiance field is a 5-dimensional function which maps a 3D location **x** and a direction in 3D sphere **d** to a color **(r,g,b)**

$$L : R^3 * S^2 \rightarrow R^3$$

$$L(x,d) = (r,g,b)$$

So, radiance is the amount of light energy passing through a given point in space, heading in a given direction. In NeRF, there is an additional output called volume density, represented by $\sigma \in$ R

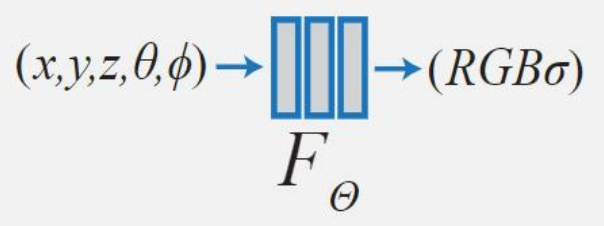

Figure 4: NeRF Network Architecture

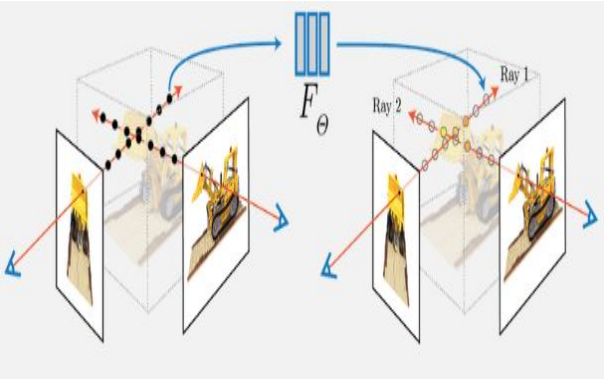

Figure 5: Volumetric rendering visualization

$$L(x,d) = (r,g,b,\sigma)$$

NeRF Idea:

Continuous neural networks as view-dependent volumetric scene representation (xyz + viewing direction d) using volumetric rendering to synthesize new views.

NeRF Volume rendering:

Volume rendering is a set of a technique used to display a 2D projection of a 3D discretely sampled data set. To render such a 2D projection of a 3D dataset, we first need to define the camera position in space relative to the volume. Then further, we need to define the RGB α where alpha stands for the opacity channel for each voxel. The main objective in volume rendering is to get a transfer function that defines RGB α for every value for every possible voxel value in a given space.

5.2 NeRF in a nutshell

Learn the radiance field of a scene based on a collection of calibrated images. It uses an MLP to learn continuous geometry and view-dependent appearance. It uses fully differentiable volume

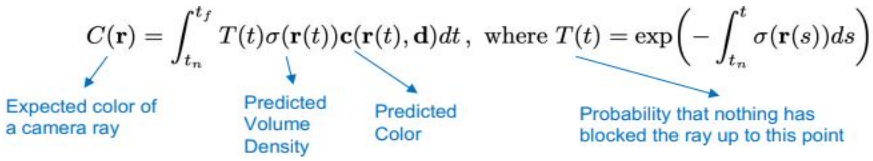

Figure 6: NeRF Volume Rendering : Generating a view from NeRF requires rendering all rays that pass through each pixel of the desired virtual camera.

rendering with reconstruction loss.

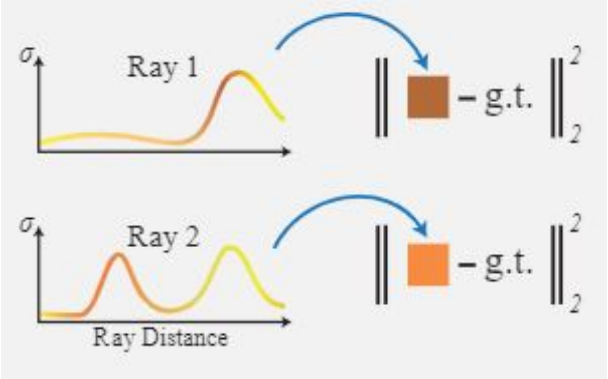

Figure 7: Volume Rendering and rendering loss

Further, it combines hierarchical sampling and positional encoding, a Fourier-basis encoding of 5D query to produce high-fidelity novel view synthesis.

5.2.1 Hierarchical sampling

NeRF uses a hierarchical structure. Two networks—the coarse network and the fine network—make up the overall network architecture. The predicted hue of a ray is assessed using N c sample points in the coarse network. It starts by optimizing from the coarse sampling, as its name suggests

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i\,, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i))$$

Figure 8: Equation for coarse network

The fine network uses $N_c + N_f = N$ sample points to calculate a ray's expected color. Its equation is the same as given in Fig. 6. We next create a more accurate sampling of points along each ray whose samples are slanted towards the important regions of the volume using the output of this "coarse" network. This process distributes additional samples to areas where we anticipate

visible content. This achieves a similar objective to significance sampling. Still, instead of treating each sample as a separate probabilistic estimate of the full integral, we discretize the sampled values as a nonuniform domain of the integration.

For each scene, a different neural continuous volume representation network is optimized. All that is needed for this is a dataset of RGB images of the scene that have been captured, along with the associated camera poses, intrinsic parameters, and scene bounds (for synthetic data, we use ground truth camera poses, intrinsics, and bounds, and for real data, we use the COLMAP structure-from-motion package [14] to estimate these parameters). We randomly select a batch of camera rays from the dataset's set of all pixels at the beginning of each optimization iteration. Then we proceed with the previously mentioned hierarchical sampling.

We then used the volume rendering procedure described in Fig. 6 to render the color of each ray from both sets of samples. N_c samples from the coarse network and $N_c + N_f$ from the fine network.

5.2.2 Positional encoding

NeRF uses positional encoding, frequently used in NLP, as opposed to the five simplistic camera parameters (Natural Language Processing). Naive input frequently performs badly when dealing with high-frequency variations in color and geometry. Positional encoding makes it easier for the network to map the input to higher-dimensional space, which helps it to improve the parameters. NeRF demonstrated that better data fitting with high-frequency fluctuation is possible when a high-frequency function maps the original input.

In simpler words, we used positional encoding to map each input 5D coordinate into a higher dimensional space. It preserves the high-frequency details.

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

Figure 9: Positional encoding

5.2.3 Loss function

Further loss is calculated, which is the total squared error between the rendered and the true pixel colors for both the coarse and fine renderings.

Here, C_c is the color for the coarse points. and C_f is the color for the finer points. This network's ultimate objective is to accurately forecast the expected color value for the ray. We can use L2-distance with the RGB data as a loss since we can estimate the ground truth ray color with

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$

Figure 10: Loss

the ground truth 3D model. Fortunately, each phase can be differentiated, allowing us to tune the network based on the expected RGB value of the rays.

According to the authors, they designed the loss function to mainly achieve two goals: 1) To optimize the coarse network 2) To optimize the fine network

5.2.4 Evaluation metrics used

PSNR (Peak Signal to Noise Ratio) Higher PSNR, lower MSE. A lower MSE implies less discrepancy between the rendered image and the ground truth image. The better the model, the greater the PSNR.

SSIM(Structural Similarity Index): It checks the structural similarity with the ground truth image model. Higher SSIM means a better model.

LPIPS(Learned Perceptual Image Patch Similarity): It determines the similarity with the view of perception using VGGNet. Lower LPIPS means a better model.

Detailed Network Architecture

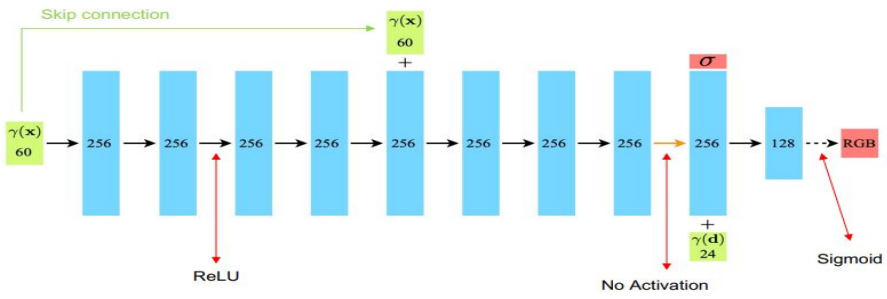

Figure 11: Network Architecture

6 Experiment

Initializing this experiment required taking multiple still images of an object and its surroundings from different observation angles. About thirty images of an office chair were taken from multiple angles in ideal lighting conditions using a Google Pixel phone. These images were then fed into Agisoft Metashape. This stand-alone software product performs photogrammetric processing of digital images and generates 3D spatial data for various applications, documentation, and visual

effects production. The outputs of Agisoft Metashape are a JSON file containing all required values for each image, such as:

- Camera 2D coordinates
- Sharpness value
- Transformation matrix

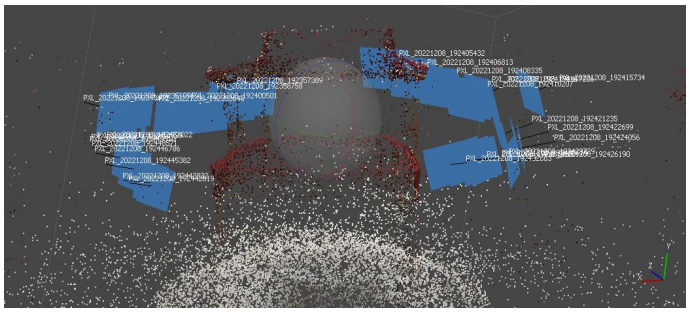

Figure 12: Result of Agisoft Metashape(1)

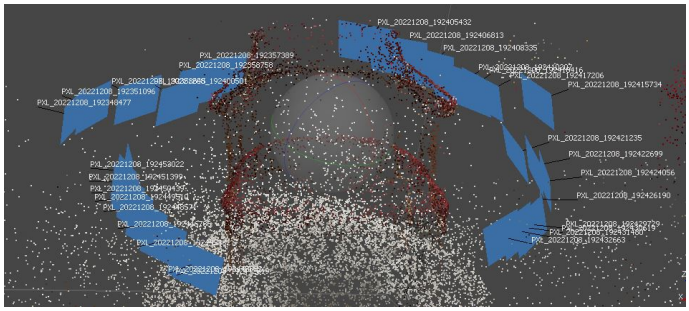

Figure 13: Result of Agisoft Metashape(2)

and an XML file containing each image's covariance values. These outputs are fed into nerfstudio, a simple API that allows for a simplified end-to-end process of creating, training, and visualizing NeRFs using modularized components. The model was trained, and the output was rendered using a 940 MX 4GB GPU and took approximately five (5) hours. The result of the training and supporting files are attached to this submission.

6.1 Results

Attention matrix

Attention is a method used in artificial neural networks that aims to imitate cognitive attention. The purpose of the impact is to encourage the network to give greater attention to the little but

significant portions of the input data by enhancing some and reducing others. Gradient descent is used to train an algorithm that determines which portion of the data is more relevant than another based on the context.

We calculated attention matrix for our model using the transformation and projection matrix for the images used.

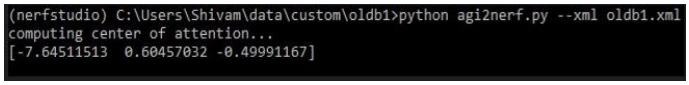

Figure 14: Attention matrix

Configuartion details of our NeRF model

1.	Total number of training iterations	30000
2.	Pipeline used	Vanilla Pipeline configurartion
3.	Train Split percentage	0.9
4.	Trained and evaluated number of rays per batch	4096
5.	Camera resolution scale factor	1.0
6.	Loss coefficients : RGB Loss coarse - 1.0	RGB Loss fine - 1.0
7.	Evaluation number of rays per chunk	32768
8.	Max resolution used	2048
9.	Number of nerf samples per ray	48
10.	Optimizer used	Adam Optimizer
11.	Learning rate	0.01
12.	Numerical stability	1e-15
13.	weight decay	0

Figure 15: A snap of depth rendered model

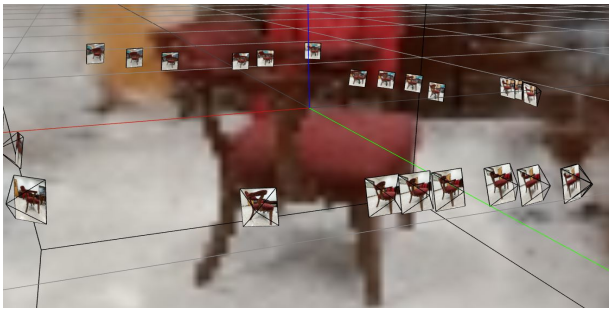

Figure 16: A snap of RGB rendering of our model

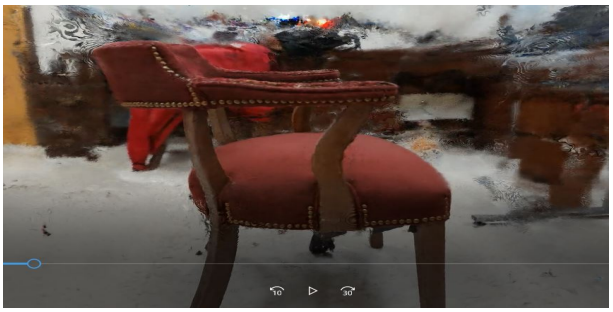

Figure 17: A snap of our final rendered model

Step (% Done)	Train Iter (time)	Train Rays / Sec	Vis Rays / Sec
1760 (5.87%)	873.643 ms	4.69 K	5.79 K
1770 (5.90%)	873.201 ms	4.69 K	5.69 K
1780 (5.93%)	872.765 ms	4.69 K	5.61 K
1790 (5.97%)	873.297 ms	4.69 K	5.63 K
1800 (6.00%)	873.633 ms	4.69 K	5.56 K
1810 (6.03%)	874.344 ms	4.68 K	5.61 K
1820 (6.07%)	873.107 ms	4.69 K	5.77 K
1830 (6.10%)	870.991 ms	4.70 K	5.80 K
1840 (6.13%)	873.262 ms	4.69 K	5.75 K
1850 (6.17%)	875.009 ms	4.68 K	5.59 K

Figure 18: Training our NeRF model

6.2 Analysis

Comparing our results with other results using Agisoft Metashape and nerfstudio, it can be inferred that the quality of image input greatly affects the quality of the rendered output. Other key parameters include the number of images and angles, proximity to the object, and lighting conditions.

7 Conclusion

NeRF showed that renderings are improved when scenes are represented as 5D neural radiance fields as opposed to the previously popular method of training deep convolutional networks to

produce discretized voxel representations. The authors anticipate that varied architectures will allow for future NeRF model optimization. NeRF's interpretability is also less good than that of earlier methods like voxel and mesh. Using both Agisoft Metashape and nerfstudio greatly simplifies the process of training and creating models. Depending on the machine that it is done on, this process can take less than an hour and still produce generally satisfactory results. With more images and data, a better result can be produced, but for anyone who can fix the models by hand after creation, it may just be quicker to generate a quick model and clean out any blemishes by hand. Otherwise, more images can be used, and an even more detailed model can be built, albeit at the price of taking more time. NeRFs have revolutionized how 3D models are created from images, and their possibilities are massive.

7.1 Future Work

There is still much that can be done with NeRFs. Future work on what has been done here may include a quantitative approach to understanding optimum parameters for rendering high-quality outputs. other possibilities may also include having a camera in place and rotating the object to get the point cloud of the item, without any background points, essentially a continuation of the dynamic NeRF. More work can done on one of its requirement of accurate camera poses to learn the scene representations. Additional work can be done on improving speed and generalizability.

References

[1] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[2] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 279–288, 1993.

[3] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[4] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun, "Fov-nerf: Foveated neural radiance fields for virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3854–3864, 2022.

[5] A. Corona-Figueroa, J. Frawley, S. Bond-Taylor, S. Bethapudi, H. P. Shum, and C. G. Willcocks, "Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray," *arXiv preprint arXiv:2202.01020*, 2022.

[6] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "Nerf–: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

[7] Agisoft, "Metashape."

[8] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021.

[9] Y. Guo, K. Chen, S. Liang, Y.-J. Liu, H. Bao, and J. Zhang, "Ad-nerf: Audio driven neural radiance fields for talking head synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5784–5794, October 2021.

[10] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. Kaplanyan, and M. Steinberger, "Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks," in *Computer Graphics Forum*, vol. 40, pp. 45–59, Wiley Online Library, 2021.

[11] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14314–14323, 2021.

[12] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5875–5884, 2021.

[13] H. Zhang, T. Dai, Y.-W. Tai, and C.-K. Tang, "Flnerf: 3d facial landmarks estimation in neural radiance fields," *arXiv preprint arXiv:2211.11202*, 2022.

[14] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.