2D IMAGES TO 3D MODEL GENERATION FINAL REPORT

Phoebe Owusu
Student# 1008916817
phoebe.owusu@mail.utoronto.ca

Connie Zhang
Student# 1009110999
connieyq.zhang@mail.utoronto.ca

Vanessa Poiana
Student# 1009125812
vanessa.poiana@mail.utoronto.ca

Ershey Waqar
Student# 1009121442
ershey.waqar@mail.utoronto.ca

ABSTRACT

This report outlines our team's work of generating a deep learning model to develop 3D representations of 2D images, aiming to democratize interior design and allow users to engage in 3D interior design models from multiple viewpoints. Our model implements the Realistic Synthetic 360 dataset, NeRF neural networks, baseline MLPs, and an improved accuracy rate from ray casting and positional encoding. —-Total Pages: 9

1 INTRODUCTION

The process of turning a house into a home often hinges on an individual's ability to personalize their space Perry (2014). Despite being an essential outlet for artistic expression, there are many homeowners, renters, and business owners who regard interior design as a luxury service Kim & Heo (2021). However, thanks to recent developments in deep learning, our project hopes to democratize identity-reinforcing interior design by offering an innovative option to take a batch of images from different viewpoints of any object to create a 3D model that is usable in 3D modelling software like SketchUp and Revit (Figure 1)

Figure 1: Example of how the rendered 3D objects can be used in Sketchup Roy (2022)

With this project, we hope to allow users to experiment with 3D models of decor pieces in their environment in order to get a more comprehensive understanding of how they interact with their space aesthetically. In this progress report, we will provide an overview of the team workflow and project progress, the methods of data processing and cleaning, and the baseline and primary models.

Furthermore, we will outline the background and related work demonstrating that deep learning is a natural fit for our goal. Traditional models often require trained computer vision engineers to prioritize essential features for each image analyzed Walsh et al. (2019). However, the large amount of parameters necessary to infer depth from 2D images would make this kind of approach wildly inefficient. By leveraging deep learning's highly scalable abilities to interpret complex visual data

and relationships, we aim to empower individuals to re-imagine their spaces with unprecedented flexibility and convenience.

2 ILLUSTRATION/FIGURE

The following diagram in Figure 2 illustrates the process of our 2D image to 3D representation generator based on "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis" Mildenhall1 et al. (2020),.

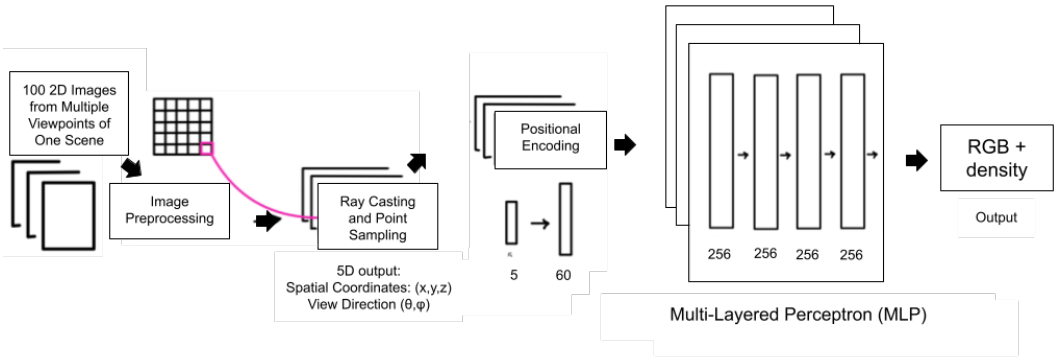

Figure 2: Network Architecture.

3 BACKGROUND & RELATED WORK

In the field of creating 3D models from 2D images, NeRF models are used to synthesize reconstructions of complex 3D scenes.

The model from the "Complementary Intrinsics from Neural Radiance Fields and CNNs for Outdoor Photometric Stereo" paper integrates NeRF and CNN into its architecture and was developed for outdoor scene relighting purposes Yang et al. (2023). The integration of these two types of neural networks greatly improved image decomposition. NeRF was used to model the 3D appearance and geometry of outdoor scenes, whereas CNN were used for intrinsic decomposition Yang et al. (2023). This architecture allowed the model to achieve a very good performance such as realistic outdoor scenes Yang et al. (2023).

PixelNeRF is a model that uses a limited set of images to create the associated 3D reconstruction of the scene Yu et al. (2021). Generally, the standard NeRF neutral network needs a large dataset of many images of an item under various angles in order to create a decent 3D reconstruction of that item. However, pixelNeRF is able to create accurate 3D reconstructions with only a few images by extracting image features from a pre-trained CNN and then using those features to condition the NeRF Yu et al. (2021).

KiloNeRF is a model that reduces computational load and memory usage, providing a much faster NeRF model Reiser et al. (2021). Standard NeRF models typically have high computational loads and, therefore, require a large amount of processing power and memory in order to produce 3D scenes Reiser et al. (2021). KiloNeRF reduces its computational load and memory usage by distributing the scene into numerous small regions and each region is then represented by a small Multi-Layer Perceptron (MLP) Reiser et al. (2021). The side by side comparison of kiloNeRF with the standard NeRF in terms of computational load can be seen in Figure 10 .

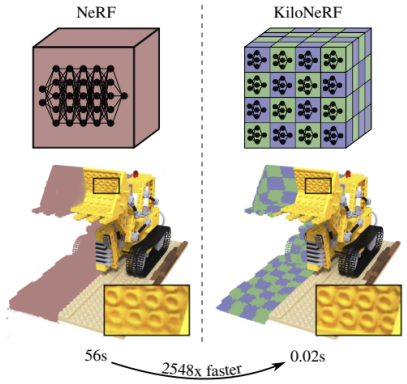

Figure 3: Comparison of KiloNeRF and standard NeRF, showing reduced computational time with KiloNeRF below the rendered LEGO truck due to its use of multiple small MLPs Reiser et al. (2021).

Mip-NeRF tackles the aliasing difficulties in NeRF models by offering multi-scale representation Barron et al. (2021). These aliasing difficulties typically occur when rendering intricate details in 3D scenes Barron et al. (2021). More specifically, Mip-NeRF implements a texture mapping technique known as mipmaps which then allows the model to continuously supply details across scales which limits aliasing artifacts Barron et al. (2021). As a result, Mip-NeRF can then handle rendering complex and fine details which improves the quality of associated 3D reconstructions.

The model from the "3D Gaussian Splatting for Real-Time Radiance Field Rendering" paper offers real-time rendering of 3D scenes using 3D Gaussian Splatting Kerbl et al. (2023). This approach is for the purpose of reducing computation load as standard NeRF models tend to have high computational loads Kerbl et al. (2023). Using a series of 3D Gaussian splats which are small ellipsoids containing color and opacity information, the model is able to represent a scene Kerbl et al. (2023). During rendering, the 3D Gaussian splats are displayed on the image plane which contributes to effective image generation Kerbl et al. (2023). From there, a neural network is used to enhance the parameters of the 3D Gaussian splats to provide improved and high-quality captures of the scene Kerbl et al. (2023).

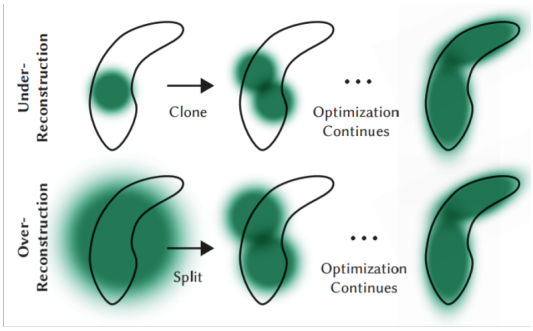

Figure 4: A demonstration of the process of 3D Gaussian splatting for creating reconstructions. Kerbl et al. (2023)

4 Data Processing

For the data used to train and evaluate our model, we initially used pre-existing NerF data from the Realistic Synthetic 360 dataset from the paper "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis" Chang et al. (2015). This dataset contains synthetic renderings of scenes with various angles and lighting, all pre-processed as seen in Figure 5 Chhibber. To reduce the spatial resolution while keeping the same 2D representation and to improve the efficiency of computations, we downsized the images so that the smallest side was 100 pixels.

In the context of computer vision or robotics, poses typically refer to the transformation matrices that describe the position and orientation of a camera or sensor in the world coordinate system. To map points from the camera's coordinate system to the world coordinate system, we pair the images with the 100 4x4 pose transformation matrices. The lego dataset was already split into training, validation, and testing data. Each subset has a corresponding "transforms.json" file containing the camera parameters of each image.

Figure 5: Three images of 'Truck' object from dataset Mildenhall1 et al. (2020)

To recreate a similar setup with our own collected data, we needed to obtain camera parameters of our images. We faced unexpected difficulties extracting the camera parameters, such as trouble running COLMAP software due to its GPU requirement. We ended up using the 'colmap2nerf.py' script from NVIDIA's GitHub for instant-ngp, which extracts frames from a video and creates an image dataset with its camera parameters Abou-Chakra (2023). The video we used was taken on an iPhone with decent camera quality, making this task easily reproducible by anyone interested. We installed necessary libraries, ffmpeg and cv2, converted the video to images specifying the extraction rate and cropping scale, and ran the colmap2nerf script (Figure 6).

Figure 6: Sample results of extracted images from video using colmap2nerf script.

With the "transforms.json" file, we performed a 70:15:15 split on the data. Unlike the pre-processed lego dataset, we had to de-noise, remove backgrounds, and normalize our images. Finally, we visualized and plotted the pre-processed images in Figure 7.

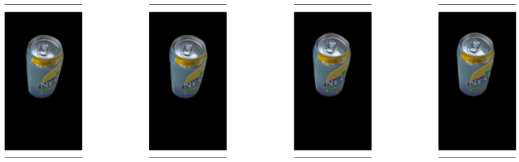

Figure 7: Sample results of extracted images from video after image pre-processing.

5 Architecture

Our primary model is heavily inspired by the paper "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis" Mildenhall1 et al. (2020), which aims to predict color and density along rays within a scene. The core idea is to represent a scene using an MLP that predicts the volume and radiance at any 3D location based on the camera's spatial coordinates and viewing direction. Unlike traditional image-based networks, NeRF models use fully-connected layers exclusively without any convolution layers. Instead of preserving local image features, NeRF models requires a continuous mapping from a 3D coordinate to their corresponding radiance values, a task which MLPs are better suited for due to their ability to handle non-grid data and model complex functions.

NeRF models process a 5D vector, which includes the spatial location (x,y,z) and viewing direction (θ, ϕ) of a point, and outputs the radiance and volume density at each pixel. To train the model, a set of images from different viewpoints of the object are required.

The steps for our network are as follows (Figure 8):

1. Collect at least 100 images from different viewpoints of the scene. For each image, obtain the camera parameters.

2. For each pixel in each image, compute the ray originating from the camera passing through the pixel.

3. Train the neural network to output the density and color at a viewpoint.

4. Render images from any new viewpoint by repeating the ray casting, point sampling, positional encoding, and neural network prediction steps for new camera parameters.

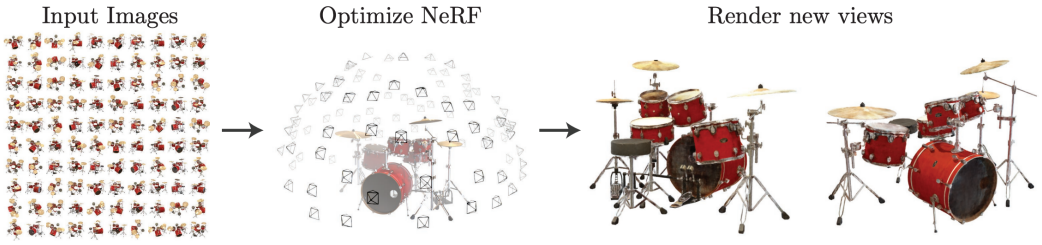

Figure 8: Multiple 2D Image from different view to 3D model using NeRF predictions for new views Mildenhall1 et al. (2020).

Training the model requires using gradient descent to minimize the difference between the generated and observed images. However, the basic model can not converge sufficiently on its own. To combat this, we used positional encoding to transform our 5D vectors. Positional encoding is applied to both the spatial coordinates and viewing direction to capture high-frequency details. This step maps the input coordinates to a higher-dimensional space using sine and cosine functions. This will take our 5 dimensional input to a 60 dimensional output which will be the input to our MLP.

Our architecture (Figure 2) consists of 4 fully connected layers, each of 256 dimensions and ReLU activation functions. The network outputs the volume density (σ) and the RGB color values for the queried 3D points and viewing directions.

6 Baseline Model

For our baseline model, we implemented a simple Multi-Layer Perceptron (MLP) model with the purpose of mapping the input coordinates of an image into related RGB values for image generation. This model will learn to predict pixel colors from input features resulting in image generation for new viewpoints.

We implemented this model through stacking 2 fully connected layers and applying ReLU activation functions for non-linearity. The final output layer reduces the number of features to 3 which relate to RGB values. A diagram of how the baseline model operates can be seen in Figure 9.

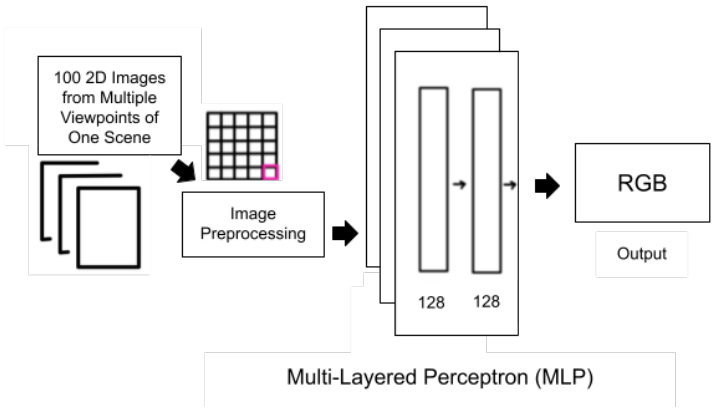

Figure 9: A diagram describing the operations of the baseline model

In terms of similarities between the baseline model and the primary neural network model, both models are MLPs. The main differences is that the primary model consists of more hidden layers and longer training. Our baseline model exhibited similar results to our primary model with a recognizable general structure of the item but more blurry (Figure 10).

To evaluate our model's performance, we used Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) values, which we plotted against the number of iterations. The MSE values are the average of the squared difference of the predicted and true pixel values. PSNR measures the quality of the rendered image compared to the ground truth, with higher values indicating better image quality. In our case, PSNR values increased linearly with the number of iterations. Similarly, MSE values decreased linearly. We stopped our training at 1000 iterations with an MSE value of 0.0033 and PSNR value of 25.24dB.

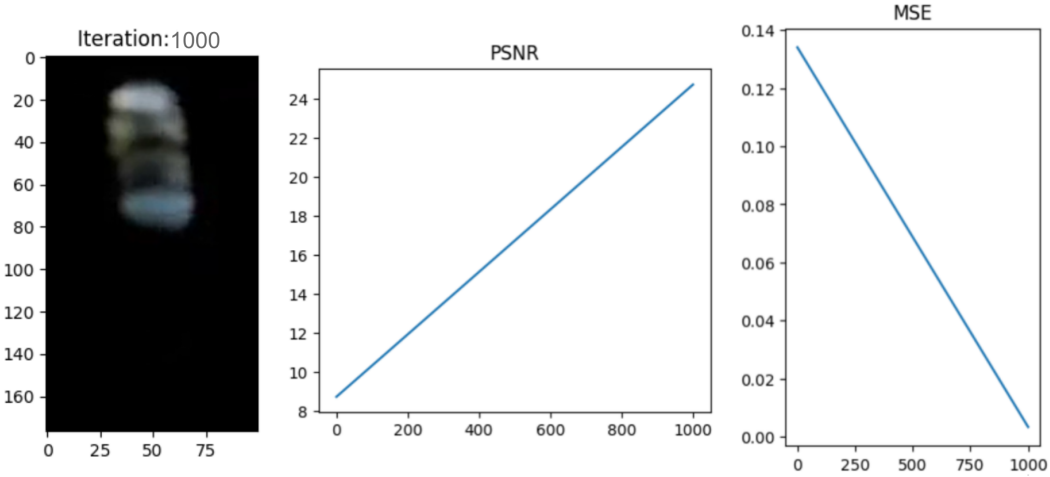

Figure 10: The PSNR and MSE values of the baseline model over the course of 1000 epochs.

7 Quantitative Results

To evaluate the performance of our primary model, we used two key metrics: Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE). Both metrics were plotted against the number of iterations to track how the model's performance evolved during training.

PSNR: In our model, the PSNR values increased significantly in the early stages of training and continued to improve, albeit at a slower rate, as the number of iterations increased (Figure 11). This trend suggests that the model quickly learned the basic structure of the scene, leading to a sharp initial improvement in image quality. However, as training progressed, the improvements became

more incremental, indicating that the model was fine-tuning the details rather than making large leaps in quality. The highest PSNR value observed was 26.8dB at 6000 iterations, reflecting the best overall image quality achieved by the model.

MSE: The MSE plot shows a steep decline in error during the initial stages of training, which corresponds to the model rapidly learning to approximate the ground truth image (Figure 11). As training continued, the rate of decline in MSE slowed, indicating that the model was refining its predictions with smaller adjustments. The lowest MSE value observed was 0.0021 at 7000 iterations. This consistent decrease in MSE, alongside the increase in PSNR, demonstrates that the model was effectively reducing errors and improving image fidelity over time.

The combination of increasing PSNR and decreasing MSE over the iterations indicates that our NeRF model was successfully learning and improving its ability to generate high-quality images. The early rapid improvements followed by more gradual refinement are typical in deep learning models as they converge on an optimal solution. While the model achieved strong performance by the end of training, the slowing rate of improvement suggests that additional iterations might yield diminishing returns, and further tuning or a more complex model might be needed for significantly better results.

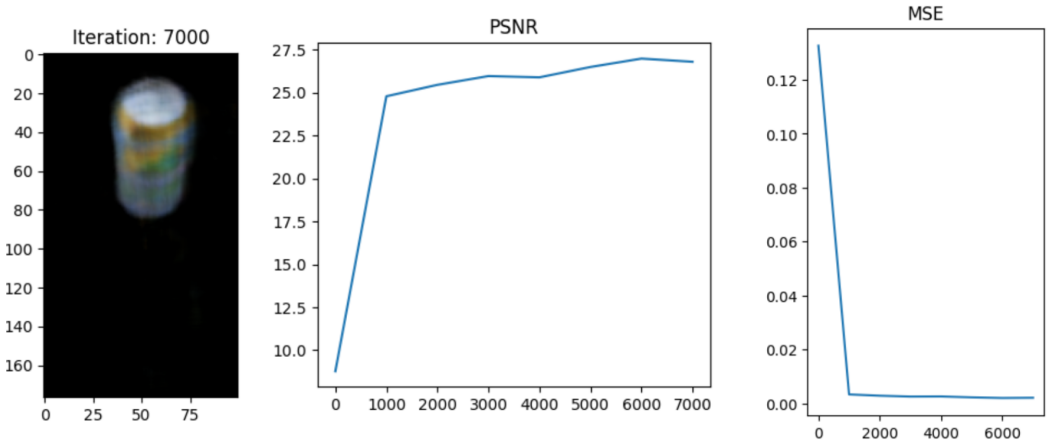

Figure 11: The PSNR and MSE values of the primary model over the course of 7000 epochs

8 Qualitative Results

The quality of the reconstructions produced by our model varied depending on the complexity of the input images. For input images with well-defined, simple structures, such as the LEGO dump truck (Figure 5), our model produced higher-quality reconstructions as seen in this video. In contrast, for more complex input images, such as a Nestea soda can with finer details (Figure 6), our model did not perform as well as seen in this video.

9 Evaluate model on new data

In our evaluation of our modified NeRF model, we aimed to ensure that the model's performance on new, unseen data accurately reflects its generalization capabilities. NeRF models are designed to hyper-tune themselves to a single scene. This means that if we want to synthesize new scenes, the entire process, including hyperparameter tuning, must be restarted from scratch. Given this limitation, we designed our evaluation to challenge the model with a dataset that diverges from the controlled environment of the original research.

We collected a diverse set of new samples, distinct from the data used during the model's development and hyperparameter tuning. This user-collected dataset featured variations in lighting conditions, scene composition, and color distributions, particularly emphasizing darker colors, which we

found to be challenging for our NeRF model. The intention was to simulate real-world conditions where the model would encounter a wide range of scenarios and focus on our intended user.

The initial evaluation revealed that the model's performance on the new data did not immediately meet the expected standards. Specifically, the model exhibited a noticeable drop in accuracy, particularly with darker colors and complex backgrounds. To address these challenges, we undertook a series of refinements:

1. **Extended Training:** We extended the training duration by approximately four times compared to the original dataset, allowing the model to better adapt to the diverse and less controlled data.
2. **Background Removal:** We implemented a preprocessing step to remove backgrounds from the images, which significantly improved the model's ability to focus on the object and yield cleaner, more accurate reconstructions.
3. **Data Selection Strategy:** Instead of selecting just a subsection of the images, we chose the most versatile and diverse set of images for training. By selecting images that were the most different from each other, we ensured that the model would perform more evenly across various angles and conditions.

After these adjustments, the model achieved a performance level on the new data that was comparable to its results on the original, controlled dataset. The model was able to generalize effectively across different scenes and lighting conditions, maintaining a high degree of accuracy and consistency.

10 Discussion

In terms of its purpose, the model is performing well. The model was created for the purpose of interior design, allowing users to get reconstructions of their furniture or decor to help them visualize how to redesign their space. The reconstructions provided by the model are not meant to exactly replicate decor but to resemble the general shape of the decor to serve as a visualization tool for planning and inspiration.

The model's performance, as indicated by the relevant plots, shows a clear trend of improvement as training progresses. Our PSNR value reinforces the conclusion that the model performed well, with it passing 27 dB which falls slightly short of the typical range for image compression (30 to 50 dB) iCue. The model's ability to accurately capture the item's general shape highlights its effectiveness for visualization purposes.

There were a few interesting scenarios with our model when training it on different inputs. We initially expected the model to work well with synthesizing scenes, given that this was the original purpose of the model and the NeRF architecture. However, we found that the model struggled with this task. To resolve this issue, we had to improve our data preprocessing strategy for the input images we captured ourselves and focus on individual objects rather than scenes. Another interesting aspect was that the model struggled with input images that featured darker colors or more complex structures. Our model may have struggled with darker colors due to their lower intensity values, which are subtle differences and might have been harder for the model to detect. Similarly, the model likely struggled with complex structures because it could not capture fine details and instead focused on the general structure or larger features, a limitation likely due to the architecture of our NeRF model. Throughout the development of our model, we gained valuable insights into the different types of NeRF models, each with its unique approach to optimizing scenes.

11 Ethical Considerations

The growing popularity of smart home technology has brought along with it many ethical concerns including "security attack, analysis of 'non-sensitive' data, improper information collection and data abuse" Chang et al. (2021). Similarly, as our project is geared towards interior-design, individuals might wish to apply it to images of their own homes and possessions. This use of personal data could threaten to violate privacy rights so it must be collected with clear consent and anonymized. In

addition, an ethical issue arising from the misuse of our model, as with smart homes, could include property surveillance and profiling Birchley et al. (2017). While our technology would not offer live feeds of individual's homes, any ability to access photos of their homes' layout and valuables could pose a threat. In order to mitigate this issue, clear usage guidelines need to be established in combination with the aforementioned anonymization for privacy protection. Furthermore, extensive use of our project might give rise to job displacements in the art and interior-design sector. While we aim to create a tool for artists that will enhance and expedite the design process, we note it might be misconstrued as a replacement for human expertise Oluwaseyi & Cena (2024). Engaging with community representatives will thus be crucial to ensure our project is a positive addition to society.

12 PROJECT DIFFICULTY/QUALITY

Our project of creating 3D reconstructions from 2D input images using a NeRF model had a variety of challenges and difficulties which we had to overcome. To begin, NeRF is a relatively new research area with ongoing advancements. Our goal was not only to grasp the fundamentals of NeRF but also to develop a unique implementation without copying on existing models, which further complicated our efforts.

While our project could be seen as easy due to our model being an MLP, the complexity of our project was raised due to the background knowlede and math required. A significant challenge was implementing key components such as positional encoding and ray-marching algorithms. We had to look into various research papers and videos to understand the mathematical foundations of these components, including the original NeRF paper Mildenhall1 et al. (2020).

Given the proejct difficulty, we believe our model the performs well taking into consideration the speed at which it renders new results and the quality. Many of the models that follow the NeRF paper require that you train your model for 24 hours before getting good results. Our model takes just under 7 minutes of training using GPU and still outputs reasonable results.

REFERENCES

Jad Abou-Chakra. Tips for training nerf models with instant neural graphics primitives. https://github.com/NVlabs/instant-ngp/blob/master/docs/nerf_dataset_tips.md, 2023.

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021. URL https://arxiv.org/abs/2103.13415.

Giles Birchley, Richard Huxtable, Madeleine Murtagh, Ruud ter Meulen, Peter Flach, and Rachael Gooberman-Hill. Smart homes, private homes? an empirical study of technology researchers' perceptions of ethical issues in developing smart-home health technologies. *BMC Medical Ethics*, 18, 2017.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *Arxiv*, Dec. 2015. doi: 10.48550/arXiv.1512.03012.

Victor Chang, Zhi Wang, Qianwen Xu, Lewis Golightly, Ben Liu, and Mitra Arami. Smart home based on internet of things and ethical issues. In *Proceedings of the 3rd International Conference on Finance, Economics, Management and IT Business - Volume 1: FEMIB,*, pp. 57–64. INSTICC, SciTePress, 2021. ISBN 978-989-758-507-4. doi: 10.5220/0010178100570064.

Akhil Chhibber. Playing around with nerf models and 2d images. https://medium.com/@akhil.chibber/playing-around-with-nerf-models-and-2d-images-3c92f353593a.

iCue. Psnr/psnr hvs/apsnr. https://vicuesoft.com/glossary/term/psnr-psnr_hvs-apsnr/.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, 42(4), July 2023. URL http://www-sop.inria.fr/reves/Basilic/2023/KKLD23.

Jeongah Kim and Wookjae Heo. Importance of interior design: An environmental mediator for perceiving life satisfaction and financial stress. *International Journal of Environmental Research and Public Health*, 18:10195, 2021. doi: 10.3390/ijerph181910195.

Ben Mildenhall1, Pratul P. Srinivasan1, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Arxiv*, 2020. doi: 10.48550/arXiv.2003.08934.

Joseph Oluwaseyi and Joshua Cena. Title: Analyzing the impact of artificial intelligence on job displacement and income inequality. *Machine Learning*, 01 2024.

Tam E. Perry. Make Mine Home: Spatial Modification With Physical and Social Implications in Older Adulthood. *The Journals of Gerontology: Series B*, 70(3):453–461, 05 2014. ISSN 1079-5014. doi: 10.1093/geronb/gbu059. URL https://doi.org/10.1093/geronb/gbu059.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps, 2021. URL https://arxiv.org/abs/2103.13744.

Clare Le Roy. How i use sketchup + layout in my interior design workflow. https://www.linkedin.com/pulse/how-i-use-sketchup-layout-my-interior-design-workflow-clare-le-roy/, 2022.

Joseph Walsh, Niall O' Mahony, Sean Campbell, Anderson Carvalho, Lenka Krpalkova, Gustavo Velasco-Hernandez, Suman Harapanahalli, and Daniel Riordan. Deep learning vs. traditional computer vision. 04 2019. ISBN 978-981-13-6209-5. doi: 10.1007/978-3-030-17795-9_10.

Siqi Yang, Xuanning Cui, Yongjie Zhu, Jiajun Tang, Si Li, Zhaofei Yu, and Boxin Shi. Complementary intrinsics from neural radiance fields and cnns for outdoor scene relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16600–16609, June 2023.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021. URL `https://arxiv.org/abs/2012.02190`.