

## 3D Rendering Using Neural Radiance Fields

Odalis Velasco<sup>1</sup>, Andrea Pilco<sup>\*2</sup>, Samuel Peña<sup>3</sup>, Gustavo Alomia<sup>4</sup>, Briyit Vallejo<sup>5</sup>

Submitted:17/03/2024    Revised: 24/04/2024    Accepted: 01/05/2024

**Abstract:** Neural Radiance Fields (NeRF) have been applied to various tasks related to synthesizing view images of objects from multiple-view images by learning a 3D rendering model represented by neural networks. Many studies have focused on applying NeRF for volume rendering of complex objects and on optimizing NeRF with different variants. In this study, we implemented a fully automated 3D rendering process using a scanning station to generate the dataset. The images are immediately fed into TinyNeRF to obtain the volumetric representation of a Clark box gear. In our experiments, we modified the ray parameters, epochs, and the number of images fed into the neural network. The PSNR (Peak Signal-to-Noise Ratio) metric was used to measure the performance of TinyNeRF representations compared to the original images. Six tests were conducted, varying the aforementioned parameters. The highest PSNR value achieved was 25.90 dB, with a training time of 40 minutes, 120 images and 4000 epochs. The variation of the parameters was considered since the camera mounted on the scanner station is in motion, as well as the object. These variations considered the epochs, ray parameters and the number of images to improve the quality of the volumetric representation of the object, measured by PSNR.

**Keywords:** 3D Rendering, Neural Rendering, NeRF, Volumetric Rendering.

### 1. Introduction

One of the revolutionary developments in computer vision and graphics is the use of Neural Radiance Fields (NeRF) for 3D rendering and reconstruction. NeRF is used for novel view reconstruction and is a continuous, implicit radiance field parameterized with a multilayer perceptron (MLP). This new perspective of rendering procedure applies to represent the scene as a constant volumetric function. The continuous volumetric function gives the volume density and the emitted radiance from any viewpoint within that scene, this is an innovative perspective rendering technique of three-dimensional scenes from a set of 2D images [1].

Before the emergence of neural radiance fields (NeRF), the performance of 3D rendering methods was mainly based on classical computer graphics techniques like ray casting [2,3], interpolation techniques [4] and explicit representations such as voxel grids [5]. However, with the advent of neural networks like NeRF, the use of volume rendering models for reconstructing views has been growing. Most studies have primarily focused on generating synthetic images and photorealistic representations [6,7]. Some research has extended its application to large-scale outdoor scenes [8], while others have focused on the problem of synthesizing novel views of a scene from a sparse set of input views [9]. We present the state-of-the-art in NeRF studies, focusing on advancements and applications.

#### 1.1. State of the art

The studies present various adaptations of the original NeRF model. The initial research, referenced in [10], utilized NeRF with the datasets "Diffuse Synthetic 360°" and "Realistic Synthetic 360°". The "Diffuse Synthetic 360°" dataset included four

Lambertian objects with simple geometry, while the "Realistic Synthetic 360°" dataset featured eight objects with complex geometry and realistic materials. The authors demonstrated that representing scenes as 5D neural radiance fields produced superior renderings compared to deep convolutional networks outputting discretized voxel representations.

NeRF used a casting ray for each pixel, in this work [11], the authors proposed using conical frustums instead, introducing a solution called mip-NeRF. These changes improved the ability to capture fine details. The mip-NeRF was 7% faster and half the size of NeRF. Compared to NeRF, mip-NeRF reduced average error rates by 17% on the NeRF dataset. Traditional methods using neural radiation start from a homogeneous scene with straight ray paths, while in [12] the authors addressed heterogeneous volumes with piecewise constant refractive indices, resulting in curved ray paths. For a synthesis of novel views of refractive objects.

The study [13] addressed NeRF's limitation of requiring separate training for each scene by estimating dense scene representations from multiple overlapping images. The authors introduced MVSNeRF, a method combining neural volume rendering and multi-view stereo techniques, which achieved photorealistic reconstructions with just three input images and demonstrated adaptability to new scenes, achieving an average PSNR (Peak Signal-to-Noise Ratio) of 26.63 dB. In contrast, the work presented in [14] introduced the SSDNeRF model, which used a single-stage training paradigm with an end-to-end objective, incorporating a NeRF auto-decoder and a latent diffusion model simultaneously. This enabled 3D reconstruction and prior learning from sparse views. At test time, SSDNeRF could sample the diffusion prior for unconditional generation or combine it with observations of unseen objects for NeRF reconstruction. In the research [15] introduced Ced-NeRF, a method for NeRF that enhanced dynamic scene rendering through a compact and efficient approach, addressing real-time rendering challenges with hybrid rendering. Ced-NeRF employed an auxiliary network to predict scene deformations accurately, enabling efficient generation of new views with performance comparable to NeRFPlayer and K-Planes but with significantly shorter training times. In contrast, paper [16] introduced NeRF-NQA, a novel reference-free quality assessment method tailored for scenes produced by NeRF and Neural View Synthesis (NVS) techniques. NeRF-NQA integrated visual and point quality assessment modules to comprehensively evaluate the spatial and angular aspects of NVS-generated scenes. Demonstrating effectiveness and robustness, NeRF-NQA

<sup>1,2,3</sup> Facultad de Ciencias Técnicas, Universidad Internacional del Ecuador, Quito 170411, Ecuador.

ORCID ID : 0000-0003-3122-3234

ORCID ID : 0009-0002-3631-6368

ORCID ID : 0009-0007-6138-3059

<sup>4,5</sup> COMBA R&D Laboratory, Faculty of Engineering, Universidad Santiago de Cali, Cali 76001, Colombia

ORCID ID : 0000-0003-3656-2876

ORCID ID : 0009-0005-4448-9939

\* Corresponding Author Email: anpilcoat@uide.edu.ec

surpassed alternative methods in terms of accuracy and generalization, emerging as a superior option for evaluating the quality of NVS- and NeRF-generated scenes.

Furthermore, [17] presented TinyNeRF, a method that reduced the storage demand of voxel grids in voxel radiance fields by exploiting visual signal sparsity in the frequency domain. TinyNeRF employed a three-stage process: transform, pruning, and quantization. It compressed models by up to 2 orders of magnitude, reducing size from 200 MB to 2 MB, with minimal impact on synthesis speed and quality. Training time increased slightly from 3 to 6.5 minutes, with a minor degradation of 0.2 dB PSNR on the synthetic NeRF dataset. Research [18] compared different 3D reconstruction representations, including Tiny-NeRF, voxel maps, point clouds, and meshes, focusing on spatial complexity and processing time in robotics. Tiny-NeRF required three times less memory but took six times longer to compute the model.

Training NeRF without precomputed camera poses is a challenge for data acquisition. Recent developments have demonstrated that this is possible, but success has primarily been limited to forward-facing scenes, where NeRF and camera poses are jointly optimized. However, problems arise with dramatic camera movements. In [19] address this issue by incorporating undistorted monocular depth backgrounds, the implemented model called Nope-NeRF. For instance in [20], CamP focused on optimizing camera parameters to improve the fidelity of 3D scene reconstructions. It proposed a preconditioning approach that normalized the effects of camera parameter perturbations, leading to significant improvements in reconstruction quality, particularly in scenes with challenging camera motion.

Neural Radiance Fields had a wide range of applications due to their ability to generate high-quality, photorealistic 3D reconstructions from 2D images. For example, in [21] one study, researchers used a high-speed motion camera to capture image datasets of plants at various geometric levels, classifying them into three levels and assessing accuracy with the PSNR. Their findings showed that Instant-NGP excelled at reconstructing plants in the scene's center but struggled with background objects, limiting its application for large-scale phenotype acquisition. In another study [22], researchers enhanced NeRFs for large-scale scenes using geographic information system (GIS) height data, partitioning scenes into objects and background with separate neural networks and adaptive sampling to improve accuracy and speed in [23] discussed high-fidelity 3D satellite models using NeRF algorithms under different illumination conditions. For Virtual Reality (VR) and Augmented Reality (AR) [24], efficient NeRF training was explored to enhance real-time performance and quality. In the field of Cultural Heritage and Archaeology [25], NeRF was applied to preserving and studying cultural heritage sites, providing detailed 3D models for virtual tours and educational purposes. In [26], introduced NeRF-RPN, a novel object detection method that operated directly on Neural Radiance Fields, NeRF-RPN leveraged a pre-trained NeRF model to detect all object bounding boxes within a scene. By utilizing a unique voxel representation that incorporated multi-scale 3D neural volumetric features, NeRF-RPN was able to directly regress 3D bounding boxes of objects without the need to render the NeRF from any specific viewpoint. This general framework was capable of detecting objects without requiring class labels, making it highly versatile for various applications. Additionally, [27] focused on enhancing novel view synthesis for sparse high-resolution remote sensing images through novel modules, such as frequency-weighted position encoding and attention mechanisms, consistently outperforming incomplete models in terms of image generation quality metrics across diverse scenes.

Research [28] focused on the extraction of geometrically accurate meshes from NeRF-based architectures, assessment utilized chamfer distance (CD), normal consistency (NC), and PSNR metrics to gauge appearance and geometry, illustrating improvements in geometric reconstruction with this approach.

Research [29] proposed the NeRF model, which jointly optimizes camera parameters and 3D scene representation. This model was compared to the baseline NeRF, where camera parameters were estimated using COLMAP. These studies highlight the potential and ongoing advancements in NeRF technology for diverse applications.

## 1.2. Main Contributions

As evidenced by the current state of the art, numerous studies have been conducted on the evolution of NeRF from its initial concept, expanding its applicability and improving its performance through spectral rendering, instance segmentation, and camera trajectory optimization. Additionally, obtaining views of an object or scene for 3D reconstruction and rendering is a long-standing problem in computer vision and a critical prerequisite. To address these demands, a scanning station was built to automate image acquisition. Moreover, the Tiny NeRF proposed in [17] was employed to get the 3D object rendering. The entire process from image acquisition and training process was automated, streamlining the workflow and improving efficiency. This project introduced the following contributions:

- Built a scanning station consisting of a C-type arc to automatically acquire the dataset of the object.
- Solved the movements of the camera mounted on the C arc by modifying the ray parameters, neural network hyperparameters, and varying the number of images in the dataset.
- Used the PSNR metric to evaluate the results obtained.

## 1.3. Outline

This work is organized as follows: Section 2 presents the system used for image acquisition and the methodology employed for achieving 3D rendering. Section 3 details the experiments conducted and the results obtained from the modified Tiny NeRF, along with the evaluation using the PSNR metric. Section 4 provides a comprehensive discussion of the findings. Finally, Section 5 concludes the study and outlines potential future work.

# 2. Methods and Materials

In this study, neural networks were used to represent and render three-dimensional object from a set of 2D images. The process begins with capturing images of the object using a C- type arc, on which a camera was mounted. These images are then fed into the neural network. The core of the method involves adjusting ray parameters, epochs and number of images before training a TinyNeRF architecture. During 3D rendering, the output images were resized with a ratio of 4 to enhance quality and resolution. The final result is a detailed volumetric representation of the object, which, in this case, was a gear due to its intricate features. Figure 1 illustrates the methodology.

## 2.1. Scanning Station

The scanning station, depicted in Figure 2, comprises a rotating base affixed to a C-type arc upon which a computer camera was mounted. This equipment facilitated the scanning of gear with a maximum diameter of 8 cm and a maximum height of 10 cm. The system's control mechanism enables the manipulation of the camera's position and orientation relative to the object being scanned. The rotating base has a maximum stated diameter of 11.5 cm and is driven by an MG90s servomotor with a required torque of 0.215 Nm. The mechanism used to move the camera along the arc was a rack and pinion system driven by a Nema 17 stepper motor. The pinion included a module of 3, a number of teeth of 10, and a pitch diameter of 30 mm. Consequently, the number of teeth in the arc was determined based on the external diameter and the module of the pinion, resulting in 128 teeth.

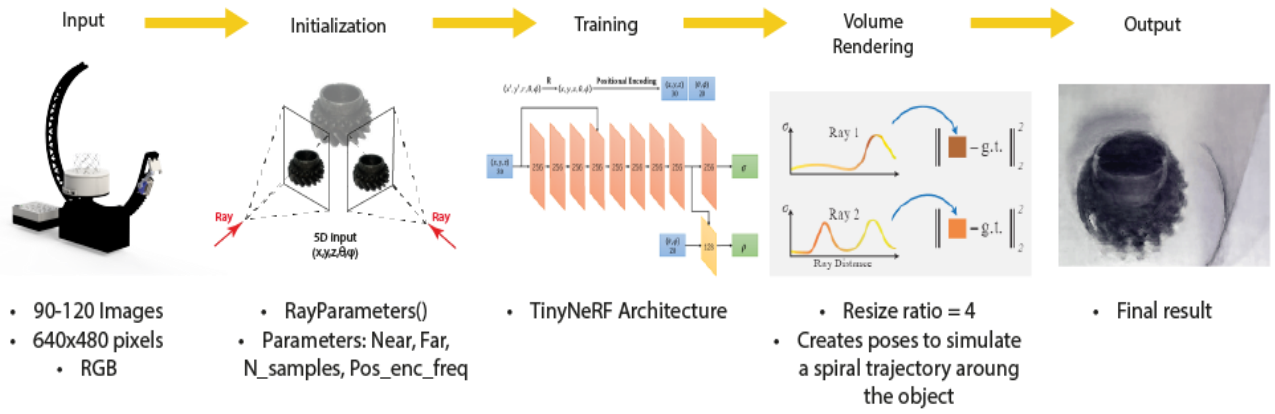


Fig. 1. Methodology for Volumetric Rendering Using Neural Radiance Fields.

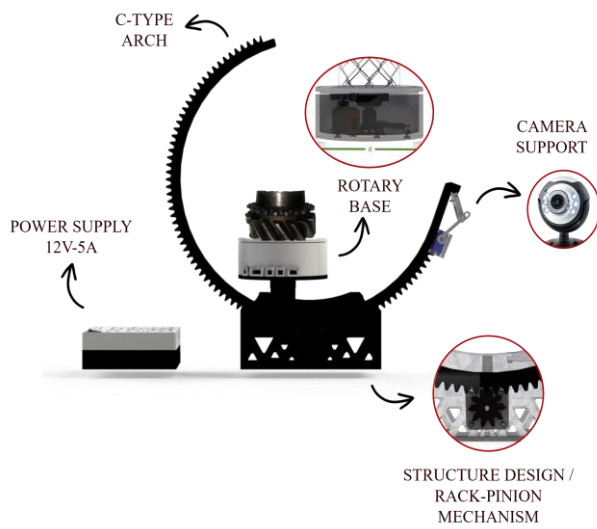


Fig. 2. Scanning station and its elements.

The selected microcontroller was an ESP8266. Figure 3 shows the communication diagram between sensors and actuators. Communication with the computer's HMI was via Wi-fi. The power supply of the prototype has a nominal voltage of 12V and a capacity of 5 amperes.

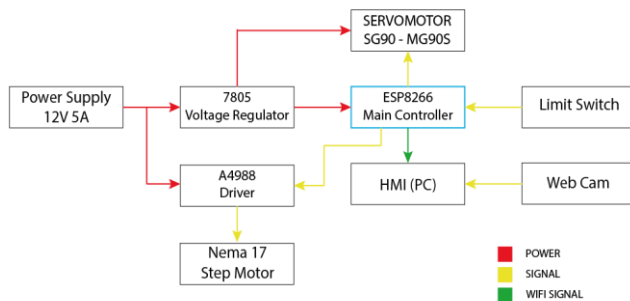


Fig. 3. Electronic configuration of volumetric rendering.

## 2.2 Image Acquisition

For image acquisition, the scan station was used. It operated in an environment with bright light and a white background. The rotary base was also white, as shown in Figure 4.



Fig. 4. Environment used for the image acquisition with the scan station.

In the scan station, a camera with the characteristics listed in Table 1 was mounted. The images had a standardized size of 640 x 480 pixels in RGB (Red, Green, Blue) color mode. Based on the dimensions of the scanned object and its distance from the camera, the camera's angle of vision plays a crucial role in dataset acquisition for neural network performance. Figure 5 illustrates that the distance of the arc in C to the base (H) is 215 mm, and the distance from the object's width to the camera (D) is 600 mm. Using equation (1), an angle of vision of 30° was calculated. This parameter significantly impacts image coverage by capturing a broader range of scene information, ensuring that the dataset includes a substantial portion of the pieces. Consequently, the camera moves in increments of 30°.

$$\alpha = 2 \times \tan^{-1}\left(\frac{H}{D}\right) \quad (1)$$

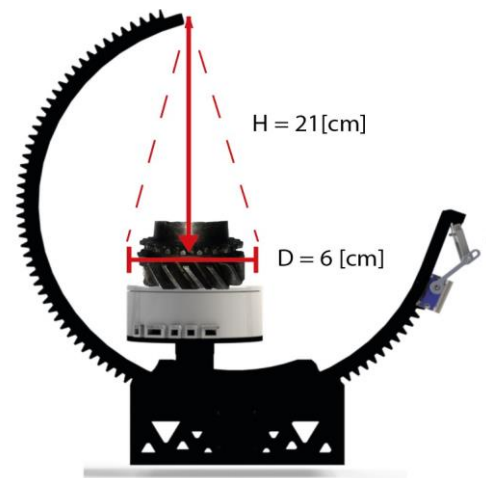


Fig. 5. Interface used to control the volumetric rendering system.

**Table 1.** Specifications of the camera used for image acquisition.

Parameters	Description Camera
ISO	100
Aperture	f/2.8
Frame Rate	30 frames per second
Predefined zoom	x10

### 2.3 Tiny NeRF

NeRF [10] is a neural rendering technique used for creating 3D objects and views. NeRF takes a 5D vector as input, consisting of 3D spatial coordinates  $(x, y, z)$  and 2D viewing directions  $(\theta, \phi)$  and outputs color (RGB) and volume density. The technique involves encoding functions that map inputs into higher-dimensional spaces, allowing the neural network to approximate complex functions. NeRF implicitly represents the properties of the 3D scene within a Multilayer Perceptron (MLP). In contrast, TinyNeRF [17] is a simplified version of NeRF. It consists of an MLP with six fully-connected ReLU layers, each with 256 filters, one fully-connected ReLU layer with 64 filters, and an output layer with four filters that express the emitted RGB with density at a certain position. The process begins by obtaining rays based on the pose and mapping 3D points, which are then fed into TinyNeRF's input. The model outputs are used to compute opacities and RGB data. The weights are subsequently calculated, and the process is repeated.

To ensure the performance and efficiency of TinyNeRF, it is important to fine-tune various aspects, including ray parameters for better sampling, training epochs for optimal learning, batch size for balanced training, ray distance for focused rendering, and the number of images. This comprehensive optimization strikes the balance between accuracy and efficiency in 3D volume creation. To achieve superior quality in volumetric rendering and ensure a more efficient scanning process, it is essential to fine-tune four key parameters within the TinyNeRF neural network:

- Near: The minimum distance from the camera's viewpoint at which the data is considered for the scene rendering
- FAR: Maximum distance from the camera's viewpoint at which the data is considered for the scene rendering
- N\_SAMPLES: Number of samples for each ray
- POS\_ENC\_FREQ: Enhance the accuracy of shapes, affecting the complexity of the model.

## 3. Experiments and Results

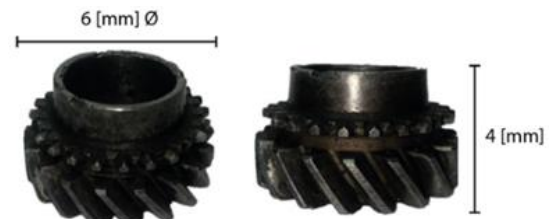
For model execution, a 7th generation Intel computer running at 2.60 GHz with 16 GB of RAM and an NVIDIA GeForce GTX 1660 Ti graphics card was used. A graphical user interface was developed, to facilitate both manual and automated control of the scanning platform mechanism. The interface, shown in Figure 6, provides control over various functions, including part 1: accessing the existing database, part 2: creating a new database, part 3: establishing a Wi-Fi connection, part 4: viewing photos that were taken, part 5: utilizing control buttons for motor movements, and part 6: monitoring the scanning and training processes. Our experiments were conducted in Python using libraries such as Pytorch for MLP and neural representations.



**Fig. 6.** Interface used to control the volumetric rendering system.

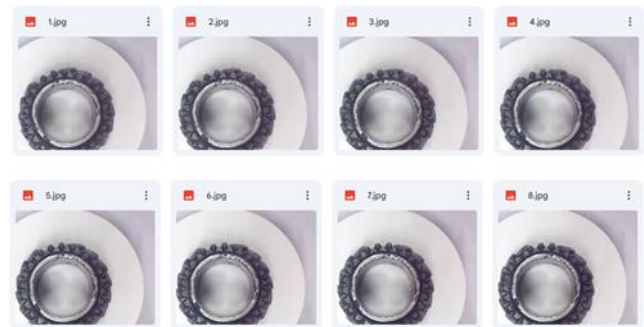
### 3.1. Dataset

All the data was obtained from the scanning station. A Clark box gear was selected due to its various details and curves in its design. All results were based solely on this object, the general dimensions are shown in Figure 7.



**Fig. 7.** Clark box gear used for volumetric rendering experiments.

The volume rendering process began when the gear was placed. The system started taking photos automatically, generating a database. Figure 8 shows the first 8 photos generated out of the dataset. The camera was positioned at the top of the arc and moved downward during the process. These images were then stored in a folder on the computer. The rotary base rotated while the camera moved along the arc. The user initially specified the number of images to be acquired during this process.



**Fig.8.** Dataset generated by the photos taken in the scan station

### 3.2. Tiny NeRF Training


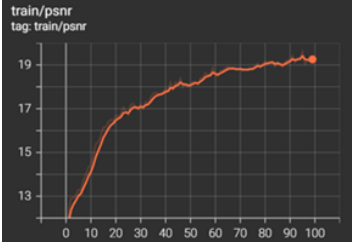
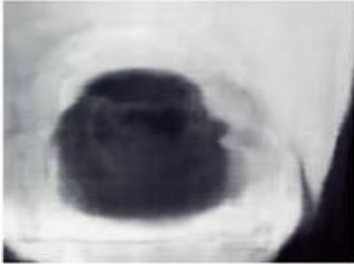
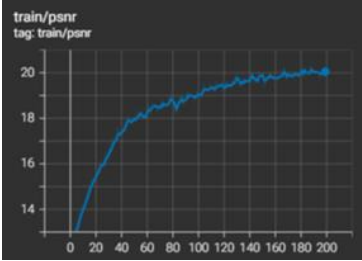
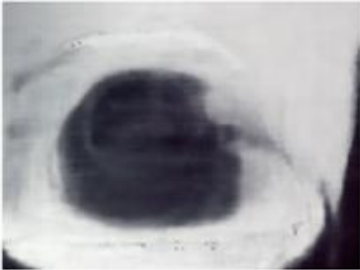
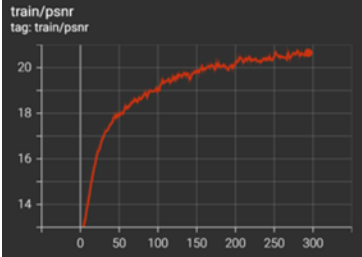

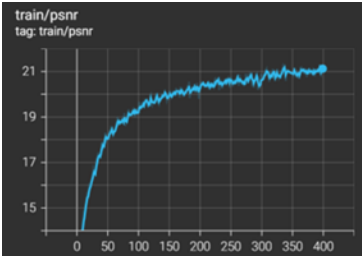
The dataset obtained with the help of the camera mounted on the C-type arc did not undergo any processing before being input into the neural network, despite the images being captured during camera movement. To address this issue and achieve the expected 3D rendering, modifications were made to the ray parameters as shown in Table 2. Initially, four tests were conducted with the following parameters: NEAR, FAR set to 0.1 and 5.0, N\_SAMPLE set to 256, DIR\_ENC\_FREQ set to 6, and EVAL\_INTERVAL set to 5. The epochs were varied by increments of 100, using PSNR as the metric and plotting the train/PSNR graph. Additionally, the number of images was 90. In the test, there was an increase in the metric from 19.12 dB to 21.16 dB as the epochs increased. As a second test to validate the model's effectiveness using images obtained from the scanning station, the number of generated images was increased to 120, and the number of epochs for




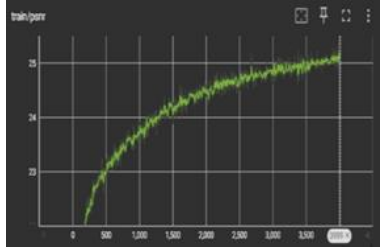

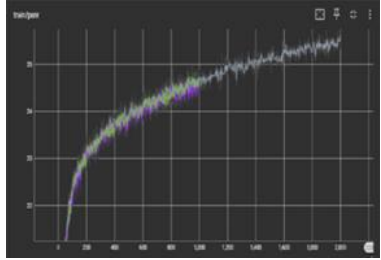
experiments ranged from 1k to 4k. The ray parameters used were NEAR, FAR: 0.0, 1.0; N\_SAMPLE: 128; DIR\_ENC\_FREQ: 4; and EVAL\_INTERVAL:

40. An increase in the PSNR metric from 25.00 dB to 25.90 dB was observed, as shown in Table 3.

**Table 2.** Results of training with a varying number of epochs with 90 images.

Number of Test	Ray Parameters	Epoch	PSNR (dB)	Time (min)	Results	Train/PSNR
1	NEAR, FAR: 0.1,5.0  N_SAMPLE:  256	100	19.12	6		
2	DIR_ENC_FREQ: 6	200	20.03	8		
3	EVAL_INTERVAL: 5	300	20.82	12		
4		400	21.16	15		

**Table 3.** Results of training with a varying number of epochs with 120 images.

<i>Number of Test</i>	<i>Ray Parameters</i>	<i>Epoch</i>	<i>PSNR (dB)</i>	<i>Time (min)</i>	<i>Results</i>	<i>Train/PSNR</i>
1	NEAR, FAR: 0.0,1.0  N_SAMPLE:  128	1k	25.00	20		
2	DIR_ENC_F REQ: 4  EVAL_INTE RVAL:  40	4k	25.90	20		

### 3.3. Evaluating peak signal-to-noise ratio (PSNR) Convergence over Epochs

For training neural networks (NN), the metric used was the peak signal-to-noise ratio (PSNR) which is used to measure the relationship between a signal and the noise that affects the representation of this signal. In this case, PSNR was used to measure how well a Tiny-NeRF representation performs compared to the original images. The PSNR values were interpreted within the following range [a3]: below 20 dB signifies low quality, 20-30 dB is considered acceptable, 30-40 dB indicates intermediate quality, 40-50 dB represents good quality, and above 50 dB is deemed excellent.

According to the results obtained, the lowest PSNR value was in the first test, which was 19.12 dB when the number of epochs was 100, indicating low quality of volume rendering. In contrast, the highest PSNR value was achieved in test 6, with a PSNR value of 25.90 dB and 4000 epochs, indicating acceptable quality of volume rendering. This result demonstrates that by varying certain ray parameters, increasing the number of epochs, and varying the number of images, the PSNR metric improves. The obtained PSNR values exhibited a clear trend of improvement with increasing epochs, indicating that the model continuously refined its ability to reconstruct the scene as the

training progressed. However, under the defined parameters, the PSNR values converged within a stable range, suggesting that further increases in epochs yielded diminishing returns in terms of model performance.

## 4. Discussion

In the work [18], a Tiny NeRF was used to obtain the dataset employed in the Pybullet simulator. This simulator allows for the creation of simulated worlds, the placement of objects, and the use of pinhole cameras to create synthetic images. The study utilized 5000 epochs, with a total time of 1,088.33 seconds, achieving a highest PSNR of 26.19 dB. In contrast, in our work, we implemented a station with a C-type arc to generate the dataset. Experiments were conducted by varying ray parameters, epochs and number of images. The highest PSNR value obtained was 25.90 dB with 4000 epochs, 120 input images and the time taken was 40 minutes (2400 seconds). which encompasses the entire duration from the initiation of photo capturing, through network training, to the generation of the rendered volume.

The comparison between the two studies highlights the efficiency of our method in achieving a comparable PSNR value with fewer epochs and a different setup. Despite the higher PSNR achieved in [18], our approach demonstrates a practical

balance between computational time and volume rendering quality, making it a viable alternative for dataset generation and neural network training. The use of a C-type arc station introduces a novel method for capturing diverse perspectives, contributing to the robustness of the dataset and the effectiveness of the training process.

## 5. Conclusions

This paper addresses the research topic of neural rendering, a field with advancements in graphics generation. We focused on rendering the volume of a gear, chosen for its intricate details, through a fully automated process. To achieve this, a station was implemented where photographs were taken to generate the dataset using a C-type arc with a mounted camera. These images are fed into TinyNeRF, a simplified version of NeRF. Six tests were conducted, varying epochs, the number of images, and ray parameters. The highest PSNR value achieved, indicating the performance of the TinyNeRF representation compared to the original images, was 25.90 dB in 40 minutes.

The dataset generation and Tiny NeRF training process are fully automated. The process begins when the piece is placed in the scan station, and the user starts the procedure through the provided interface. At the end of the process, the user can visualize the rendered volume of the piece located in the scan station, along with the PSNR value and the train/PSNR graph. Future work will utilize this implementation to continue with a machining process of the piece obtained from the volume rendering.

## Author contributions

**Name1 Surname1:** Conceptualization, Methodology, Software, Field study **Name2 Surname2:** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Name3 Surname3:** Visualization, Investigation, Writing-Reviewing and Editing.

## Conflicts of interest

The authors declare no conflicts of interest.

## References

- [1] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. Barron, and P. Srinivasan, "Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1-14.
- [2] AKM. Rabby and C. Zhang, "BeyondPixels: A Comprehensive Review of the Evolution of Neural Radiance Fields," *ArXiv*, vol. 2306.03000, 2023, pp.1-15.
- [3] J. Congote, L. Kabongo, A. Moreno, A. Segura, A. Beristain, J. Posada, and O. Ruiz, "Volume ray casting in WebGL," *Computer Graphics*, 2012, pp. 157-178.
- [4] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, pp. 423-432.
- [5] W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler, "Learning to predict 3D objects with an interpolation-based differentiable renderer," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1-11, 2019.
- [6] G. Sharma, D. Rebain, K. M. Yi, y A. Tagliasacchi, "Volumetric Rendering with Baked Quadrature Fields," *arXiv preprint arXiv:2312.02202*, 2023, pp. 1-13.
- [7] G. Mazzacca, A. Karami, S. Rigon, E. Farella, P. Trybala, and F. Remondino, "NeRF for heritage 3D reconstruction," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, no. M-2-2023, pp. 1051-1058, 2023.
- [8] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. P. Mildenhall, P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-NeRF: Scalable Large Scene Neural View Synthesis," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8238-8248.
- [9] Y. Era, R. Togo, K. Maeda, T. Ogawa and M. Haseyama, "Content-based Image Retrieval Using Effective Synthesized Images from Different Camera Views via pixelNeRF," *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*, 2022, pp. 404-405.
- [10] 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," *arXiv preprint arXiv:2003.08934*, 2020, pp. 1-12.
- [11] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855-5864.
- [12] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14124-14133.
- [13] J. I. Pan, J. W. Su, K. W. Hsiao, T. Y. Yen, and H. K. Chu, "Sampling neural radiance fields for refractive objects," *SIGGRAPH Asia 2022 Technical Communications*, 2022, pp. 1-4.
- [14] H. Chen, J. Gu, A. Chen, W. Tian, Z. Tu, L. Liu and H. Su, "Single-stage diffusion nerf: A unified approach to 3D generation and reconstruction," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2416-2425.
- [15] Y. Lin, "Ced-NeRF: A Compact and Efficient Method for Dynamic Neural Radiance Fields," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 4, 2024, pp. 3504-3512.
- [16] Q. Qu, H. Liang, X. Chen, Y. Y. Chung and Y. Shen, "NeRF-NQA: No-Reference Quality Assessment for Scenes Generated by NeRF and Neural View Synthesis Methods," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 5, May 2024, pp. 2129-2139.
- [17] T. Zhao, J. Chen, C. Leng, and J. Cheng, "Tinynerf: Towards 100 x compression of voxel radiance fields," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3588-3596.
- [18] S. Gante, J. Vasquez, M. Valencia, and M. Carbajal, "A comparison of tiny-NeRF versus spatial representations for 3D reconstruction," *arXiv preprint*

arXiv:2301.11522, 2023, pp. 1-8.

- [19] W. Bian, Z. Wang, K. Li, J. Bian, and V. A. Prisacariu, "NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1-8.
- [20] K. Park, P. Henzler, B. Mildenhall, J. T. Barron, and R. Martin-Brualla, "CamP: Camera Preconditioning for Neural Radiance Fields," arXiv:2308.10902, 2023, pp. 1-11.
- [21] K. Han, W. Xiang, and L. Yu. "Volume feature rendering for fast neural radiance field reconstruction,". *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23)* Curran Associates Inc., Red Hook,, 2024, pp. 65416–65427.
- [22] H. Aoki and T. Yamanaka, "Improving NeRF with Height Data for Utilization of GIS Data," arXiv preprint, 2023, pp. 1-8.
- [23] M. Trupti, C. Basilio, N. Van, W. Ryan, S. Todd. 3D Reconstruction of Non-cooperative Resident Space Objects using Instant NGP-accelerated NeRF and D-NeRF, 10.48550/arXiv.2301.09060, 2023, pp. 1-14.
- [24] S. Li, C. Li, W. Zhu, B. Yu, Y. Zhao, C. Wan, H. You, H. Shi, and Y. Lin, "Instant-3D: Instant Neural Radiance Field Training Towards On-Device AR/VR 3D Reconstruction," *International Symposium on Computer Architecture (ISCA)*, 2024, pp.1-13.
- [25] V. Croce, G. Caroti, L. De Luca, A. Piemonte, and P. Véron, "Neural radiance fields (NeRF): Review and potential applications to digital cultural heritage," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, 2023, pp. 453-460.
- [26] B. Hu, J. Huang, Y. Liu, Y.-W. Tai, and C.-K. Tang, "NeRF-RPN: A General Framework for Object Detection in NeRFs," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1-8.
- [27] J. Lv, J. Guo, Y. Zhang, X. Zhao, and B. Lei, "Neural Radiance Fields for High-Resolution Remote Sensing Novel View Synthesis," *Remote Sensing*, vol. 15, no. 16, 2023, pp. 3920.
- [28] M. Rakotosaona, F. Manhardt, D. Arroyo, M. Niemeyer, A. Kundu, and F. Tombari, "NeRFMeshing: Distilling Neural Radiance Fields into Geometrically-Accurate 3D Meshes," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [29] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF--: Neural Radiance Fields Without Known Camera Parameters," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. pp 1-17.