An Effective Scheme to Accelerate NeRF for Web Applications Using Hash-based Caching and Precomputed Features

OkHwan Bae and Chung-Pyo Hong*

Division of Computer Engineering, Hoseo University, Republic of Korea
E-mail: foem954@gmail.com; cphong@hoseo.edu
**Corresponding Author*

Received 30 October 2024; Accepted 27 November 2024

Abstract

In recent years, 3D reconstruction and rendering technologies have become increasingly important in various web-based applications within the field of web technology. In particular, with the emergence of technologies such as WebGL and WebGPU, which enable real-time 3D content rendering in web browsers, immersive experiences and interactions on the web have been significantly enhanced. These technologies are widely used in applications such as 3D visualization of virtual products or 3D exploration of building interiors on real estate websites. Through these advancements, users can experience 3D content directly in their browsers without the need to install additional software, greatly expanding the possibilities of the web. Amidst this trend, the neural radiance field (NeRF) has garnered attention as a cutting-edge technology that improves the accuracy of 3D reconstruction and rendering.

NeRF is a technique widely used in computer vision and graphics for reconstructing 3D spaces from 2D images taken from multiple viewpoints. By predicting the color and density of each pixel, NeRF captures the complex 3D structure and optical properties of a scene, enabling highly accurate 3D reconstructions. However, NeRF's primary limitation is the time-consuming

Journal of Web Engineering, Vol. 23_7, 1041–1056.
doi: 10.13052/jwe1540-9589.2376

nature of both the training and inference processes. Research efforts to address this issue have focused on two key areas: optimizing network architectures and training procedures to accelerate scene learning, and improving inference speed for faster rendering. While progress has been made in enhancing training speed, challenges remain in improving the inference process.

To address these limitations, we propose a two-step approach to significantly improve NeRF's performance. First, we optimize the training phase through a multi-resolution hash encoding technique, reducing the computational complexity and speeding up the learning process. Second, we accelerate the inference phase by caching the input data of the NeRF MLP, which allows for faster rendering without sacrificing quality. Our experimental results demonstrate that this approach reduces training time by 68.42% and increases inference speed by 98.18%.

Keywords: Neural radiance field, multiresolution hash encoding, 3D reconstruction, cache.

1 Introduction

In recent years, technologies for reconstructing 3D structures from 2D images have made significant advancements in computer vision, providing substantial benefits for WebGL and WebGPU-based web applications. These technologies are essential for enhancing various web-based applications, with ongoing research aimed at generating more precise and realistic 3D models to improve user experience and interactivity on the web. Among these, NeRF (neural radiance field) [1] is a representative technique that enables high quality 3D reconstruction, focusing on accurately restoring complex 3D scenes using 2D images captured from multiple viewpoints. NeRF has attracted significant attention in both computer vision and graphics, and various studies are being conducted to improve its performance.

The core concept of NeRF is to interpret each pixel in 2D images as a ray and learn how these rays are distributed in 3D space to reconstruct the 3D structure and optical properties of a scene. NeRF predicts the color and density of each ray through a neural network, modeling the complex 3D structure of the scene based on this information. Through this method, NeRF can restore color and ray information at each point in the 3D model, allowing the scene to be rendered from various angles.

However, despite its excellent performance, NeRF has some significant limitations [2–6]. The most prominent issue is its high computational cost. NeRF uses deep learning neural networks to extract rays from pixels in the input images and reconstruct the scene, which requires complex computations such as camera calibration and volume rendering. In particular, to generate high-quality 3D models, all points in the scene must be densely sampled, demanding substantial computational resources. As a result, NeRF's training and inference times are very long, with complex scenes taking hours to learn, and generating new viewpoint images during inference also requires considerable time.

To address these issues, recent research has focused on either reducing training time or improving inference speed. Research aimed at reducing training time primarily focuses on optimizing NeRF's network architecture or training algorithms to enhance scene learning efficiency. These studies significantly improve NeRF's training speed, but limitations remain in terms of inference speed. Conversely, studies focused on improving inference speed mainly aim to quickly generate new viewpoint images using pre-trained NeRF models, but the issue of training time remains unresolved.

Thus, there is a need for methods that can improve both training time and inference speed simultaneously. In this study, we propose a method to significantly enhance both the training and inference performance of NeRF by introducing a multi-resolution hash encoding technique. Existing methods directly input the spatial coordinates of each pixel into the network, which can result in complex computations. In contrast, multi-resolution hash encoding transforms spatial coordinates into a more efficient low-dimensional space, allowing NeRF's MLP to learn the 3D spatial coordinates more quickly. This significantly reduces training time.

Additionally, we introduced a caching technique for NeRF's MLP input values to significantly improve inference speed. When generating images from a new viewpoint, NeRF generally performs complex calculations for the rays passing through the MLP, resulting in high computational costs. However, similar values often recur within the same space, so previously computed values can be cached and reused, reducing unnecessary computations. This approach greatly reduces the computational load during inference.

The structure of this paper is as follows: Section 2 introduces existing studies aimed at overcoming the limitations of NeRF, and Section 3 explains how to improve training and inference speed using multi-resolution hash

encoding and caching. Section 4 describes the experimental process and results, and Section 5 presents the conclusion.

2 Related Work

In this section, we present a variety of related studies that overcome the short introductory limitations for NeRF. 3D reconstruction, which generates actual 3D models from 2D images taken from various angles, is a crucial technology, and active research is being conducted in the fields of computer vision and graphics [7, 8]. A neural radiance field (NeRF) is a representative method proposed to solve this 3D reconstruction problem, using volume rendering to reconstruct 3D scenes based on images from different perspectives. NeRF calculates the rays for each pixel using camera calibration data and models the radiance field of the spatial coordinates these rays pass through with a neural network. This allows high-quality 3D reconstructions even in scenes with complex geometric structures and optical properties. However, NeRF suffers from a high computational cost and long training times, leading to very slow training and inference (rendering) speeds, which limits its application in real-world scenarios. For instance, training a single scene can take tens of hours or even days, making it unsuitable for real-time applications. To overcome these limitations, various studies have been proposed.

2.1 Enhancements in Training Efficiency

MVSNeRF [9] is a study that reconstructs the neural radiance field using a multi-view stereo (MVS) approach. This method introduces the plane sweep cost volume used in MVS to effectively infer geometric information. As a result, it can accurately reconstruct scenes using only three input views, combining fast network inference with physically based volume rendering to achieve high-quality 3D reconstructions. MVSNeRF is also highly data-efficient, capturing fine details in complex scenes even with a small number of input images. Furthermore, it introduces an efficient training algorithm to reduce computational costs and improve practicality.

Plenoxels [10] is another method that significantly improves training speed by representing the neural radiance field with a sparse 3D grid and spherical harmonics, without the use of neural networks. This study divides the space into grids and directly optimizes the density and radiance properties at each grid's location. Trilinear interpolation maintains continuity between grids, and regularization techniques are applied to prevent overfitting and

enhance generalization performance. As a result, Plenoxels reduced the training time from around 1.6 days in the original NeRF to about 11 minutes while improving the PSNR (peak signal-to-noise ratio) performance from 31.15 to 31.83. This greatly improves memory usage and computational efficiency, bringing real-time 3D graphics applications closer to reality.

PointNeRF [11] is a study that models continuous volumetric radiance fields using 3D neural point clouds. This method generates an initial point cloud using a pre-trained deep learning-based MVS network and efficiently samples rays in empty space through traditional point rendering techniques. During the volume rendering process, the characteristics of each point are predicted with a neural network, and the point cloud is continuously adjusted through geometric inference. PointNeRF was able to reconstruct scenes that took more than 20 hours to train in traditional NeRF in just 20 to 40 minutes with higher quality. PointNeRF allows efficient learning even in scenes with complex geometric structures, and its compatibility with point-based sensor data makes it useful in fields like autonomous driving and robotics.

Instant neural graphics primitives (Instant-NGP) [12] is a study that drastically reduces NeRF's training time by embedding input coordinates through multi-resolution hash encoding. This method uses a small-scale neural network but embeds the spatial coordinates in a hash table to effectively capture the details of complex scenes. By utilizing multi-resolution hash maps, it can learn information at various scales simultaneously, allowing high-quality 3D models to be generated in just a few seconds.

However, these studies mainly focus on reducing training times, and there is still room for improvement in inference and rendering speeds. NeRF's inference stage still requires complex computations, limiting its use in real-time applications.

2.2 Advances in Real-time Inference

NeRF suffers from slow inference and rendering speeds due to the need for repeatedly invoking the neural network for each ray, which limits its applicability in real-time applications. Several studies have been proposed to address this issue of slow inference speed.

PlenOctrees [13] is a method that enables real-time rendering by converting a pre-trained NeRF model into a precomputed octree-based 3D representation. This approach uses spherical harmonics to predict radiance and removes view dependency. The octree structure allows hierarchical storage of spatial information, enabling fast lookups during ray tracing. While

traditional NeRF takes about 30 seconds to render a single image, PlenOctrees significantly improves rendering speed to about 6 ms on an NVIDIA V100 GPU. Furthermore, octree optimization maintains or enhances visual quality.

MobileNeRF [14] is a study aimed at developing a NeRF model that can run efficiently on mobile devices by simplifying the complex neural network structure and optimizing the rendering process. This method uses texture polygons, which include binary opacity and feature vectors, to synthesize images from new viewpoints. After rendering polygons with a Z-buffer, a small view-dependent MLP maps the feature values of each pixel to color values. This allows for immediate and efficient rendering without the need for a GPU, even on various devices with a very small MLP. MobileNeRF contributes to providing high-quality 3D content for mobile AR/VR applications.

Additionally, FastNeRF [15] proposes a method to maximize inference speed by caching the NeRF model's inference values. By separating position and view direction, it spatially samples the position function and stores it in a precomputed cache. The view direction function is computed online and combined with the cached data, reducing the number of neural network calls per ray and maximizing parallel processing on the GPU. As a result, FastNeRF achieves over 200 frames per second rendering speed while maintaining high-quality 3D reconstruction, enabling real-time rendering.

However, while these studies have effectively reduced inference times, the training speed remains at the same level as traditional NeRF, requiring long training times. Thus, improving both training and inference speeds simultaneously is a crucial challenge for the practical realization of real-time 3D reconstruction.

In this paper, we propose a 3D reconstruction method based on 2D images that integrates key techniques from various studies aimed at improving both training and inference speeds. This method improves upon previous research by enhancing both training and inference speeds simultaneously.

3 Proposed Scheme

In this section, we describe the machine learning framework designed to accelerate 3D rendering in NeRF. The proposed scheme leverages hash-based optimization during the training phase to improve memory efficiency, while the inference stage is enhanced by a caching mechanism to expedite

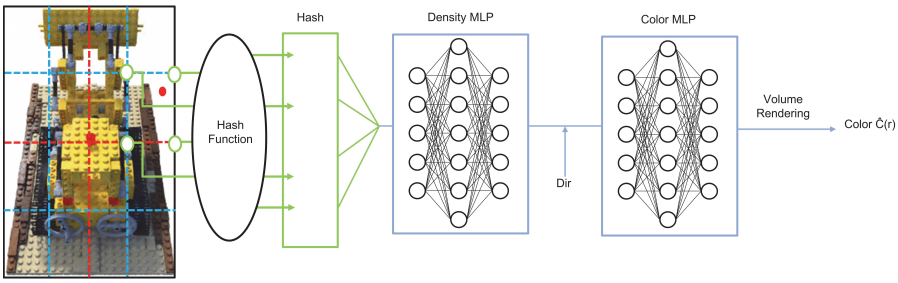

Figure 1 Hash based training optimization.

the rendering process. The details of the hash-based learning model and the caching strategies for faster inference are then explained.

3.1 Optimizing Training with Hash-based Feature Encoding

In this paper, we introduce a multi-resolution global bounding box and a multi-resolution hash encoding method to accelerate the learning speed. Figure 1 is a pipeline of learning optimization methods through hash-based feature encoding.

The multi-resolution global bounding box scheme splits the space into different resolutions, allowing the spatial position of the input coordinates to be encoded efficiently. Specifically, the entire 3D space is divided into grids of multiple resolutions, each grid cell being assigned a unique index number. Then, divide the input 3D coordinates and direction information by the total 3D space size for scaling, add 0.5 to move it to the range [0, 1], and make sure each coordinate is within the range [0.5, 0.5]. This significantly improves the learning speed by excluding coordinates that deviate from the 3D space from the operation. Based on this mask, the input coordinates are scaled by the grid resolution N to calculate the position on each grid to calculate the index of the grid cell containing the coordinates. This allows the spatial information of the input coordinates to be represented across multiple scales, effectively capturing spatial features of varying sizes, from small to large structures.

These index numbers are then used to apply multi-resolution hash encoding via a hash function (1).

$$\mathrm{h(x)} = \left(\bigoplus_{\mathrm{i}=1}^{\mathrm{d}} \mathrm{x_i}\pi_\mathrm{i} \right) \bmod T \tag{1}$$

The hash function maps the high-dimensional index space into a limited-size hash table, efficiently managing memory usage. Each index is

transformed into a key for the hash table via the hash function, and the hash table stores a feature vector corresponding to each key. These feature vectors are embedded into an F-dimensional space and, in this study, we set $F = 2$, balancing computational efficiency and memory usage. This shows that even with small feature vectors, sufficient expressiveness can be achieved.

Next, the feature vectors of neighboring grid points surrounding the input coordinate are combined using trilinear interpolation. Trilinear interpolation is a method that computes a weighted sum of the feature vectors from the eight adjacent grid points in 3D space, allowing for a smooth and continuous spatial feature representation. This provides a refined feature expression for the input coordinates.

In addition, directional information is incorporated into the interpolated feature vectors. Directional information is essential for accurately modeling surface properties and reflection characteristics of objects by considering the directionality of light rays. This directional information is combined with the interpolated spatial features and fed into the model, enabling it to learn color variations based on direction.

The combined feature representations are then fed into two separate MLPs: density MLP and color MLP. Density MLP takes spatial features as input and predicts the density at a given location. Color MLP takes both spatial and directional features as input and predicts color information at those locations and directions. MLP is compactly designed to improve computational efficiency and accelerate the learning process. Each MLP consists of three FC layers with 64 hidden units.

Finally, volume rendering is applied to create 2D images at new time points using predicted density and color values. Volume rendering calculates the accumulated color, density, and transparency as light rays pass through space, resulting in realistic images.

Combining multi-resolution global bounding boxes and multi-resolution hash encoding methods, we efficiently represent the spatial position of the input coordinates on multiple scales, improving the model's expressiveness while minimizing memory usage. This improves the computational efficiency of the learning process and enables fast generation of high-quality images.

3.2 Accelerating Inference via Caching and Precomputation

In this section, we introduce a cache technique to accelerate the inference speed of NeRF. Figure 2 shows an acceleration by cacheting pre-calculated inference values of NeRF in the inference stage.

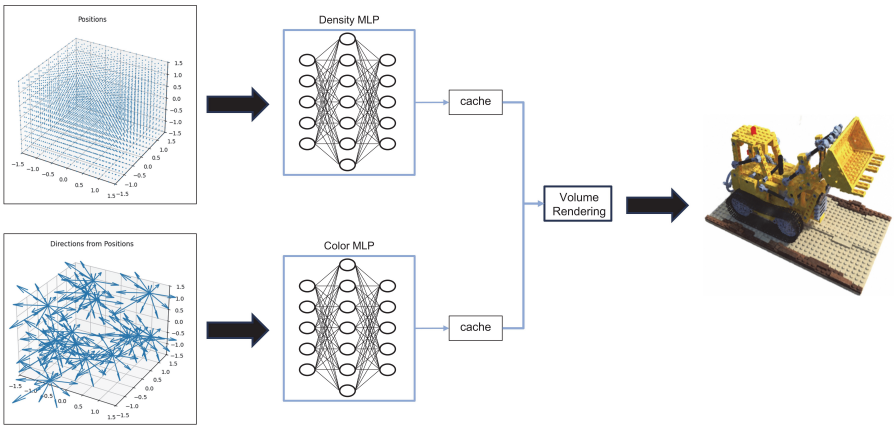

Figure 2 Accelerating inference with caching and precomputation.

Using a NeRF model trained with multi-resolution hash encoding, we divide the global bounding grid space into two: one for position space and the other for direction space. Each space is partitioned into grids of sizes N_p and N_d for the position and direction spaces, respectively.

In this process, we efficiently represent the entire 3D space and ray directions by dividing them into grid cells. Each grid cell has a unique index, allowing quick retrieval of information about positions or directions. By partitioning the space into grids, we can efficiently manage data for specific positions or directions without complex computations.

Then, we precompute the density and color values for each position and direction and store them in cache space. This cache table contains density and color information for various combinations of positions and directions, enabling us to quickly obtain the necessary values during inference without performing complex neural network computations. This significantly reduces the model's inference time using precomputed data.

When new input values are given, such as a new camera position or ray direction, instead of performing complex inference with the existing NeRF model, we quickly retrieve the density and color values corresponding to the input position and direction from the cache space. If the exact coordinates are not available in the cache, we calculate the required values using trilinear interpolation with the values of adjacent grid cells.

This method allows for fast inference without MLP computations, achieving rendering speeds close to real-time. It addresses the issue where existing NeRF models could not respond in real-time due to complex neural network computations.

The rapidly obtained density and color values are used to perform volume rendering, generating the final 2D image from a new viewpoint.

This approach can also efficiently manage memory usage by leveraging the cache space. You can optimize system resources by adjusting the cache size as needed or prioritizing important data. Additionally, it significantly reduces computational complexity during the inference stage. By enabling rapid image generation while maintaining the quality of the trained model, it avoids complex neural network computations. This is crucial for real-time applications and allows for real-time 3D content rendering in web browsers.

4 Evaluation

In this section, we present the experimental results comparing the proposed scheme with the conventional scheme. The experiments were conducted in two parts. The first part evaluates the efficiency in both the training and inference phases. The second part assesses the rendering quality using the peak signal-to-noise ratio (PSNR) as a metric. For a detailed comparison, we showcase the rendered images from the proposed scheme, the conventional scheme, and the ground truth, highlighting the visual differences.

The experiments were conducted in an environment with a single Nvidia RTX 3060 GPU. This ensures that the comparison between the proposed and conventional schemes is consistent and fair, taking place under the same hardware conditions.

Additionally, we analyze the impact of cache size in the proposed scheme by measuring its influence on both inference time and PSNR. We also provide a graphical representation illustrating how variations in cache size affect both the efficiency and quality metrics. This visualization highlights the balance between cache size, speed, and rendering quality in the accelerated rendering process.

4.1 Efficiency Evaluation: Training and Inference Comparison

In this section, we evaluate the efficiency of the proposed NeRF model by comparing its training and inference performance with that of the conventional NeRF model.

As shown in Figure 3, the proposed NeRF model has improved training speed by 68.42% compared to the conventional NeRF. This significant enhancement is primarily due to the use of multi-resolution hash encoding

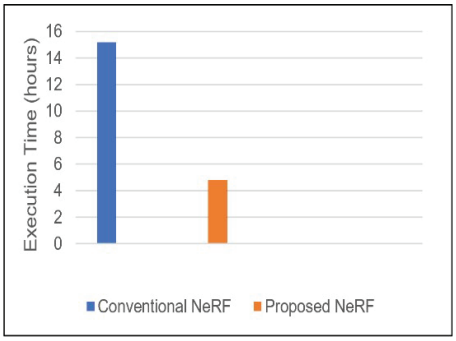

Figure 3 Comparison of training time.

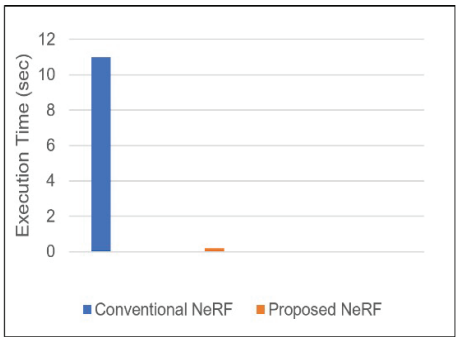

Figure 4 Comparison of rendering time.

and the division of the global bounding grid space into position and direction grids.

Furthermore, as depicted in Figure 4, the inference (rendering) speed of the proposed NeRF model has impressively improved by 98.18%. This dramatic increase enables rendering capabilities close to real-time, which is a substantial advancement over existing NeRF models that often suffer from slow inference due to complex neural network computations. By retrieving precomputed values from the cache and using trilinear interpolation when necessary, our model bypasses the need for time-consuming MLP computations during inference.

These efficiency improvements are not only theoretical but also have practical implications for applications that require rapid rendering of 3D scenes, such as real-time simulations in web browsers.

4.2 Quality Assessment: PSNR-based Rendering Accuracy

As shown in Figure 5, although the proposed NeRF model has significantly improved training and inference speeds compared to the conventional NeRF, the PSNR indices are 24.36 (proposed model) and 26.29 (conventional model), respectively, indicating that it maintains image quality similar to that of the conventional NeRF model. This means that the proposed model enhances efficiency while delivering excellent performance without any degradation in image quality, which is an important achievement that greatly improves its potential for use in real-time applications.

Finally, as shown in Table 1, the proposed NeRF model has demonstrated improvements of 68.42% in training speed and 98.18% in inference time compared to the conventional NeRF. Additionally, the PSNR indices were 24.36 and 26.29, respectively, indicating that while significantly improving training and inference speeds, the proposed model maintains image quality similar to that of the conventional NeRF model.

4.3 Impact of Cache Size on Efficiency and Quality

As can be seen in Figure 6 and Table 2, the performance of the proposed NeRF model is significantly affected by the cache size, resulting in variations

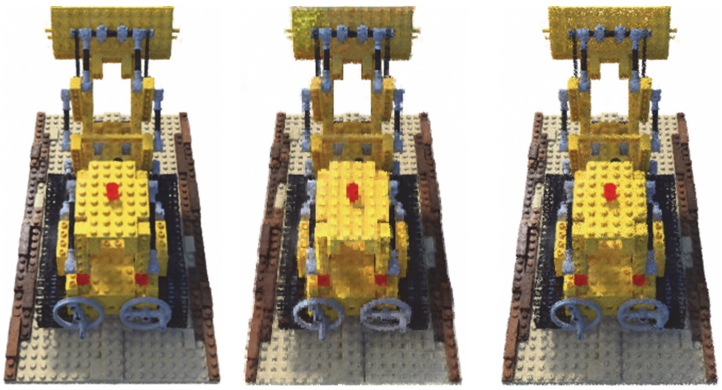

Figure 5 The result of the proposed NeRF.

Table 1 Comparison between the proposed NeRF model and the conventional model

	Training	Rendering	PSNR
Proposed NeRF	4.8 h	0.2 s	24.36
Conventional NeRF	15.2 h	11.2 s	26.29

Figure 6 Comparison of results of proposed NeRF by cache size.

Table 2 PSNR comparison of proposed NeRF by cache size

	512^3	256^3	128^3
PSNR	26.81	25.42	24.15
Rendering	1.4 s	1.1 s	0.8 s

in both output quality and rendering speed. Specifically, by setting the cache size divided into 512, 256, and 128 counts along the x, y, and z axes respectively, we observed PSNR indices of 26.81, 25.42, and 24.15, indicating that a larger cache size produces higher-quality outputs. On the other hand, rendering times were measured at 1.4 seconds, 1.1 seconds, and 0.8 seconds, demonstrating that smaller cache sizes result in faster rendering speeds. This is because memory usage and data access times vary depending on the cache size, and these results indicate that by adjusting the cache size, one can balance the output image quality and rendering speed. Therefore, it is important to select the optimal cache size according to the specific application or system resource constraints.

5 Conclusion

In recent years, 3D reconstruction and rendering techniques have become increasingly important in various web-based applications in the field of web technology. In particular, techniques that enable real-time 3D content rendering in web browsers have greatly improved immersive experiences and interactions in the web. These techniques can greatly expand the potential of the web through 3D visualization of virtual products. Amid this trend, NeRF (neural radiance field) has emerged as a state-of-the-art technology that improves the accuracy of 3D reconstruction and rendering.

NeRF has gained considerable attention in the field of 3D space rendering due to its ability to reconstruct highly accurate 3D scenes from 2D images taken from multiple viewpoints. By predicting the color and density of each pixel, NeRF can capture the complex 3D structure and optical properties of a scene, enabling photorealistic 3D reconstructions. Despite its capabilities, NeRF's main limitation lies in its slow training and inference processes.

In this paper, we addressed the limitations of traditional NeRF regarding its training and inference performance. To overcome these challenges, we proposed a two-step approach aimed at significantly improving NeRF's overall efficiency. First, we optimized the training phase by employing a multi-resolution hash encoding technique, which reduces computational complexity and accelerates the learning process. Second, we enhanced the inference phase by caching the input data of the NeRF MLP, enabling faster rendering without compromising quality. Our experimental results demonstrated that this approach reduces training time by 68.42% and increases inference speed by 98.18%.

Acknowledgement

This research was supported by the Academic Research Fund of Hoseo University in 2021 (20210481).

References

[1] Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." Communications of the ACM 65.1 (2021): 99–106.

[2] Wang, Guangming, et al. "NeRF in Robotics: A Survey." arXiv preprint arXiv:2405.01333 (2024).

[3] Gao, Ruicheng, and Yue Qi. "A Brief Review on Differentiable Rendering: Recent Advances and Challenges." Electronics 13.17 (2024): 3546.

[4] Cai, Jintong, and Huimin Lu. "NeRF-based Multi-View Synthesis Techniques: A Survey." 2024 International Wireless Communications and Mobile Computing (IWCMC). IEEE, 2024.

[5] Zhu, Fang, et al. "Deep review and analysis of recent nerfs." APSIPA Transactions on Signal and Information Processing 12.1 (2023).

[6] Rabby, A. K. M., and Chengcui Zhang. "BeyondPixels: A comprehensive review of the evolution of neural radiance fields." arXiv preprint arXiv:2306.03000 (2023).

[7] Jin, Yiwei, Diqiong Jiang, and Ming Cai. "3d reconstruction using deep learning: a survey." Communications in Information and Systems 20.4 (2020): 389–413.

[8] Vora, Suhani, et al. "Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes." arXiv preprint arXiv:2111.13260 (2021).

[9] MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo

[10] Fridovich-Keil, Sara, et al. "Plenoxels: Radiance fields without neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[11] Xu, Qiangeng, et al. "Point-nerf: Point-based neural radiance fields." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[12] Müller, Thomas, et al. "Instant neural graphics primitives with a multiresolution hash encoding." ACM transactions on graphics (TOG) 41.4 (2022): 1–15.

[13] Yu, Alex, et al. "Plenoctrees for real-time rendering of neural radiance fields." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[14] Chen, Zhiqin, et al. "Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[15] Garbin, Stephan J., et al. "Fastnerf: High-fidelity neural rendering at 200fps." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

Biographies

OkHwan Bae holds a Master's degree in computer engineering from Hoseo University, Asan, Korea. His research areas include computer vision, deep learning, and reinforcement learning.

Chung-Pyo Hong received his B.Sc. and M.Sc. degrees in computer science from Yonsei University, Seoul, Korea, in 2004 and 2006, respectively. In 2012, he received his Ph.D. degree in computer science from Yonsei University, Seoul, Korea. He is currently an associate professor of Computer Engineering at Hoseo University, Asan, Korea. His research interests include machine learning, explainable AI, and data science.