



BeyondPixels: A Comprehensive Review of the Evolution of Neural Radiance Fields

AKM Shahariar Azad Rabby 

Dept. of Computer Science
The University of Alabama at Birmingham
Birmingham, AL, USA
arabby@uab.edu

Chengcui Zhang 

Dept. of Computer Science
The University of Alabama at Birmingham
Birmingham, AL, USA
czhang02@uab.edu

Abstract—Neural rendering combines ideas from classical computer graphics and machine learning to synthesize images from real-world observations. NeRF, short for Neural Radiance Fields, is a recent innovation that uses AI algorithms to create 3D objects from 2D images. By leveraging an interpolation approach, NeRF can produce new 3D reconstructed views of complicated scenes. Rather than directly restoring the whole 3D scene geometry, NeRF generates a volumetric representation called a “radiance field,” which is capable of creating color and density for every point within the relevant 3D space. The broad appeal and notoriety of NeRF make it imperative to examine the existing research on the topic comprehensively. While previous surveys on 3D rendering have primarily focused on traditional computer vision-based or deep learning-based approaches, only a handful of them discuss the potential of NeRF. However, such surveys have predominantly focused on NeRF’s early contributions and have not explored its full potential. NeRF is a relatively new technique continuously being investigated for its capabilities and limitations. This survey reviews recent advances in NeRF and categorizes them according to their architectural designs, especially in the field of novel view synthesis.

Index Terms—Neural Radiance Field, NeRF, Computer Vision Survey, Novel View Synthesis, Neural Rendering, 3D Reconstruction, Differentiable Rendering

I. INTRODUCTION

Image-based view synthesis techniques are widely applied to computer graphics and computer vision. One of the pressing concerns of these techniques is representing a 3D model or scene using the information of the input 2D images. Neural Radiance Field or NeRF [1], is a novel technique that trains AI algorithms to generate 3D objects from 2D images. NeRF can render novel 3D reconstructed views of complex scenes utilizing an interpolation approach between scenes. However, instead of directly recovering the entire 3D scene geometry, NeRF computes a “radiance field,” a volumetric representation that generates color and density for each point in the concerned 3D space.

NeRF uses a deep neural network to represent complex scenes in a fully-connected, non-convolutional way. The input to the network is a continuous 5D coordinate, while the output is the volume density and view-dependent emitted radiance at that particular spatial location. The method uses classic volume rendering techniques to synthesize views and is optimized using a set of images with known camera poses.

The method outperforms previous work on neural rendering and view synthesis and can render novel photorealistic views of scenes with complicated geometry and appearance. The inventors of NeRF propose a new method for view synthesis by directly optimizing the parameters of a continuous 5D scene representation. They introduce a positional encoding and a hierarchical sampling strategy to address the inefficiency of the basic implementation. Using input viewing direction, their method can represent non-Lambertian effects such as specularities. NeRF uses a separate neural continuous volume representation network for each scene. NeRF presents a new method for view synthesis and provides quantitative and qualitative results and ablation studies to demonstrate its superior performance compared to prior works. NeRF also discusses how the work improves upon previous approaches that use multilayer perceptrons (MLP) to represent objects and scenes as continuous functions and the potential for more progress in efficiently optimizing and rendering NeRF.

NeRF has several advantages compared to previous approaches, including:

- High-quality 3D reconstructions: NeRF can create high-quality 3D reconstructions of complex scenes, including fine surface details and reflections.
- View synthesis: NeRF can synthesize novel views of a scene from a small number of input images, allowing for virtual walkthroughs of a scene from any viewpoint.
- Continuous representation: NeRF provides a continuous representation of a scene that can be efficiently queried at any point, enabling applications such as object manipulation and rendering.
- Unsupervised training: NeRF can be trained unsupervised, meaning it can learn to reconstruct a scene without explicit supervision.
- Wide applicability: NeRF can be applied to a wide range of scenarios, including outdoor scenes, indoor scenes, and even microscopic structures.

A generous amount of research based on 3D scene representation has further enhanced and extended NeRF and achieved satisfactory accuracy along with efficiency [2]. This new and emerging field of research can be divided into two broad categories based on its specific purpose: the analysis

and improvement of NeRF itself [3–5] and the extensions based on the NeRF framework [6–8]. Over time, researchers have developed various innovating techniques to overcome the limitations of NeRF, described in more detail as follows.

Quality and Scalability: The representation of NeRF suffers from vulnerability to sampling and aliasing problems, which can lead to significant artifacts in the synthesized images. Aliasing artifacts are the visual artifacts that appear when NeRF is used to render images of scenes that contain sharp edges or textures. These artifacts are caused by the limited sampling of the radiance field, which leads to the inaccurate reconstruction of these features. Researchers tried to overcome this issue using a cone instead of rays and to avoid ray sampling in the empty scene space [9–11]. One of the biggest challenges with NeRF is its slow training speed. Training a NeRF model requires a large amount of memory and computing power. NeRF is also slow in rendering, particularly at high resolutions. This can make it impractical for real-time applications or generating animations with multiple frames. Many researchers have focused on methods such as dividing the scene into smaller manageable blocks, voxel-based representations, and super-sampling, etc., to accelerate NeRF [12–15].

High-Quality Large Dataset: One limitation of NeRF is that it requires a large dataset of high-quality images to train the network. This dataset needs to capture the scene from various viewpoints, which can be challenging and time-consuming. The dataset’s quality also plays a critical role in the network’s performance. If the dataset is noisy or contains artifacts, it can significantly impact the quality of synthesized views. Furthermore, as the scene’s complexity increases, the dataset demands also increase, making it challenging to use NeRF for large-scale scenes. Researchers tried to focus on these limitations and improve the performance of NeRF by adding prior conditions to the model. Prior can play a dual role in image processing. Prior can be used to assist in the reconstruction of images, or can be employed in a generative fashion to produce new images, with prior as an additional input feature or a label used to guide the model’s predictions. For example, in image generation, a conditional deep-learning model might be trained to generate images of specific objects or scenes based on a given set of prior such as an object class or scene description [16–19]. Furthermore, some researchers also tried the one-few-zero shot approach to solve these limitations. These different forms of learning allow NeRF to generate accurate 3D models with limited or no data, making them useful in real-world applications [20–22].

Representing Articulated Objects: NeRF has shown remarkable progress in representing complex and highly-detailed static scenes using deep learning models. However, one of the significant limitations of NeRF is its inability to effectively represent articulated objects, such as humans or animals with multiple moving parts, where the object’s geometry is highly dynamic and can change with respect to the viewpoint. The problem arises due to the fact that NeRF represents the scene as a continuous volumetric function and assumes that the scene

is static. When applied to articulated objects, this assumption breaks down, resulting in artifacts in the synthesized images. In such cases, the explicit modeling of each object part using a set of parametric functions is not feasible, and hence, the representation of these objects requires new approaches. An emerging trend is the application of the NeRF for articulated models of people or cats. Such models represent the shape of the object in the image using rigid parts that are interconnected [23–26].

Scene Editing: One other limitation of NeRF is the difficulty in editing the scene after it has been rendered. The reason is that NeRF represents the scene as a continuous function and does not store explicit geometry information. This makes it challenging to modify or remove objects in the scene or to change their properties, such as texture or lighting. While some recent works have attempted to address this limitation by proposing methods for interactive scene editing, these works are still in their infancy and would require significant improvements to become practical tools for artists and designers [27–30].

NeRF has also been applied in several other related fields with excellent results [31–33], including the field of audio, where it has found use due to the similarity of sound propagation to rays in a volume [34, 35].

While there have been several surveys and research papers discussing the traditional computer vision-based [36–39] and conventional deep learning-based approaches [40–43] to 3D rendering, only a few of them have discussed NeRF [44]. This is because NeRF is a relatively new technique that has only been introduced recently, and its full potential and limitations are still being explored. Traditional computer vision and deep learning approaches have been limited to generating photorealistic images with intricate geometries using discrete representations such as triangle meshes or voxel grids. In contrast, NeRF has shown remarkable potential in generating highly realistic images with precise color and geometric position. By tracing camera rays through a scene, NeRF obtains a set of sampled 3D points, which are utilized alongside their corresponding 2D viewing directions as input to a neural network. The output set comprises colors and densities, which collectively generate visually accurate scene representations. Also, NeRF requires fewer images to generate a 3D scene than traditional computer vision and deep learning approaches, reducing the rendering time. Researchers in traditional computer vision societies focused on mathematical solutions such as Marching Cubes and Space partitioning, while Marching Cubes [45–47] seems the most popular algorithm for surface rendering. Some researchers have also used other methods such as Contour Filter [48] and various interpolation methods [49]. In deep learning, during the early times, CNN was used by several for surface rendering, while for volume rendering, traditional computer vision algorithms were still used [50, 51]. Later, new traditional computer vision-based algorithms have been used for volume rendering, such as space partitioning, occupancy networks [52], shape partitioning, and subspace parameterization [53].

In this survey, After outlining the detailed explanation of the theory behind NeRF in Section 2, we categorize and analyze the existing research works in Section 3 based on the abovementioned classification. We discuss the strengths and limitations of NeRF-based methods, including their advantages in terms of representation capabilities, rendering quality, and generalization to new scenes, as well as challenges in terms of interpretability, computational efficiency, and scalability in Section 4. Finally, we highlight the potential future research directions of NeRF-based methods, such as improving the interpretability, efficiency, and scalability of the method.

II. BACKGROUND

A. Understating NeRF

Neural Radiance Fields (NeRF) [1] is a groundbreaking technique in computer graphics that allows for the photorealistic rendering of 3D scenes. Developed in 2020 by researchers at the University of California, Berkeley, NeRF uses deep neural networks to model the 3D geometry and appearance of objects in a scene, enabling the creation of high-quality visualizations that are difficult or impossible to achieve using traditional rendering techniques.

The key idea behind NeRF is to represent the appearance of a scene as a function of 3D position and viewing direction, known as the radiance field. The radiance field describes how light travels through the scene and interacts with its surfaces and can be used to generate images from arbitrary viewpoints. The “neural” part of the name refers to the fact that the radiance field is learned using a neural network. The NeRF algorithm involves several steps: data acquisition, network training, and rendering. Figure 1 provides an overview of the NeRF scene representation and differentiable rendering procedure. It shows the steps involved in synthesizing images, which include sampling 5D coordinates along camera rays, using an MLP to generate color and volume density, and compositing these values into an image using volume rendering techniques.

In the following, we will explore the basics of NeRF, how it works, and its applications in computer vision and computer graphics.

1) *Basics of NeRF*: NeRF is a deep learning technique that learns to model a scene’s 3D geometry and appearance by capturing the relationship between the scene’s geometry and appearance using a neural network. The neural network takes a set of 2D images of the scene from different viewpoints as input and outputs a 3D representation of the scene. This representation can render new views of the scene from any viewpoint, resulting in highly photorealistic renderings.

2) *Data Acquisition*: The first step in the NeRF algorithm is to acquire a set of 2D images of the scene from different viewpoints. Next, these images are used to train the neural network to model relationship between the 3D geometry and appearance of the scene. Ideally, the images should cover a wide range of viewpoints and lighting conditions to ensure the neural network can capture the full range of variability in the scene.

3) *Network Training*: The next step in the NeRF algorithm is to train a neural network to model the relationship between the 3D geometry and appearance of the scene. Specifically, the neural network takes as input a pair of coordinates (x, y) that corresponds to a 2D point in the image plane and a viewing direction vector and outputs a corresponding 3D point in the scene and its associated color. This process is repeated for every pixel in the input images, resulting in a set of 3D points and colors that can be used to render the scene from any viewpoint.

The training process involves optimizing the parameters of the neural network to minimize the difference between the rendered image and the ground truth image. This is done by computing the rendering equation, which describes how light travels through a scene and interacts with its surfaces, and using it to generate a synthetic image that is compared to the ground truth image.

4) *Rendering*: The final step in the NeRF algorithm is to use the trained neural network to render new views of the scene from arbitrary viewpoints. This is done by querying the neural network for the radiance field at each point in the 3D space and using it to generate an image of the scene from the desired viewpoint.

5) *Applications*: NeRF has already been used in various applications, and its potential for future applications is vast. For example, NeRF could generate realistic 3D models of archaeological sites, allowing researchers to explore and study them in new ways. It could also generate virtual try-on experiences for online shopping, allowing customers to see how clothes would look on them before making a purchase. NeRF has a wide range of applications in computer graphics, including virtual reality, augmented reality, video games, and movie production. By enabling the creation of highly realistic 3D visualizations, NeRF has the potential to revolutionize how we create and interact with digital content.

One of the most promising applications of NeRF is in virtual and augmented reality. NeRF can create immersive experience indistinguishable from reality by generating real-time photorealistic renderings of virtual environments. This has the potential to transform how we interact with virtual content, making it possible to create realistic simulations of real-world scenarios for training, education, and entertainment.

Another application of NeRF is in video game development. Using NeRF to create highly realistic environments and characters, game developers can create more immersive and engaging experience for players. This can lead to a more enjoyable and satisfying gaming experience, which can, in turn, drive increased engagement and revenue for game developers.

In movie production, NeRF can be used to create highly realistic special effects and CGI scenes. By modeling the appearance and geometry of objects and scenes using neural networks, filmmakers can create highly detailed and realistic visualizations that are difficult or impossible to achieve using traditional rendering techniques.

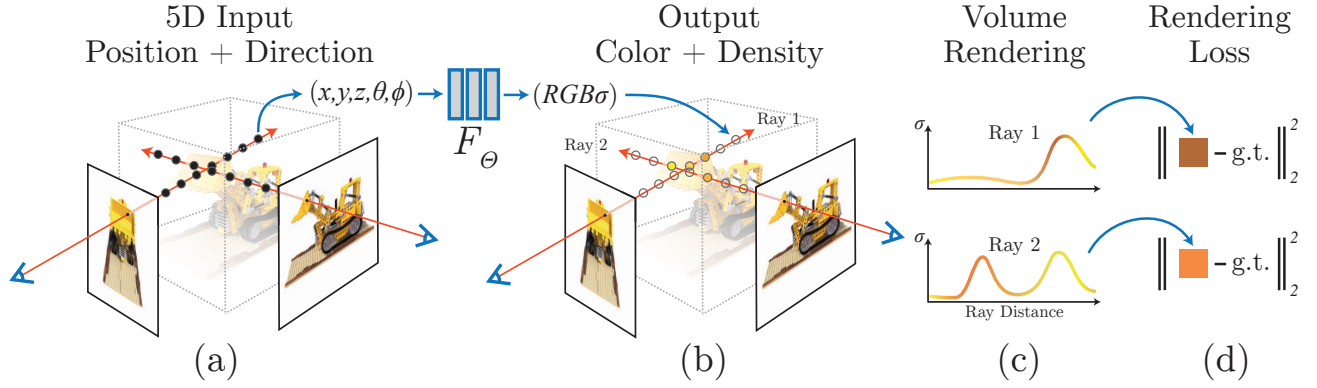


Figure 1: An overview of the process of rendering and training NeRF. The first step (a) involves selecting sampling points for each pixel in an image that needs to be synthesized. The next step (b) involves using NeRF’s MLP(s) to generate densities and colors at the selected sampling points, followed by volume rendering techniques to composite these values and generate an image (c). Since this rendering function is differentiable, scene representation is optimized by minimizing the difference between the synthesized and observed ground truth images (d). This image is adapted from [1].

B. Literature Review

NeRF has received significant attention in the computer vision research community in recent years, and a number of survey papers have been published to provide a comprehensive overview of the advances made in this area. In the survey paper [54], the authors discuss the advantages of NeRF over other rendering techniques. The first survey paper to focus on NeRF is a pre-print by Dellaert et al. [55], which references a relatively small number of publications at the time and categorizes those papers into different types. In contrast, Tewari et al. [56] wrote a comprehensive state-of-the-art report on advances in neural rendering with a specific focus on NeRF models. This paper includes several influential NeRF papers and other cutting-edge rendering techniques. The authors provide detailed descriptions and analyses of the discussed papers, making the report a valuable resource for researchers and practitioners in this field. Gao et al. [57] published a survey paper regarded as one of the most comprehensive surveys of NeRF. To organize the vast amount of NeRF literature, the authors classify the literature into six categories based on their citation counts and GitHub stars. In addition, the authors discuss the datasets and evaluation metrics used in various NeRF papers. While the authors have attempted to cover a large number of papers, the survey provides less detailed information on many papers, making it difficult to follow without expert knowledge. Furthermore, the survey lacks a clear technical justification for the categories, which may leave readers wondering about the rationale behind the categorization. In addition, there are websites [58–61] that periodically summarize and categorize NeRF papers from top computer science conferences. However, these resources generally offer only brief summaries but not comprehensive surveys.

Our survey stands apart from previous surveys in that we focus exclusively on papers related to NeRF, providing comprehensive and detailed summaries and comparisons of each selected work. To aid future researchers, we propose a new classification system for NeRF models based on the challenges those model address to encourage further research to overcome these challenges. By extensively covering the state-of-the-art NeRF-based papers presented at top computer science conferences, our survey offers a comprehensive and up-to-date resource for researchers and practitioners interested in this cutting-edge technology.

III. ADVANCEMENT OF NeRF

A. Quality and Scalability

The NeRF method creates a volumetric representation of a scene using input images and camera poses, which can result in undersampling and aliasing artifacts. In addition, the training process is slow, and rendering can take a long time. Researchers have tried to overcome these issues by using cones instead of rays and implementing methods such as dividing the scene into smaller blocks or using voxel-based representations, as described below.

Mip-NeRF [9] is a novel improvement of NeRF, designed to represent 3D objects as a continuous function. While NeRF was an innovative solution to the problem of rendering 3D objects with high accuracy, it was plagued by sampling and aliasing problems, which led to reduced precision in the resulting image. Mip-NeRF addresses these issues by introducing cone tracing instead of the ray tracing used in NeRF, which allows for a more accurate representation of each pixel. The cone is divided into conical frustums [62], and a representation of the volume covered by each frustum is created using integrated positional encoding (IPE) features, which replaces NeRF’s two separate “coarse” and “fine” MLPs

with a single multiscale MLP, reducing the model size and making training and evaluation faster. IPE is a technique that allows neural networks to take advantage of the geometric structure of the data. In the case of Mip-NeRF, IPE features are used to encode the position and orientation of each point in 3D space, allowing the network to understand better the geometry of the scene being rendered. Mip-NeRF also uses a scale-aware structure, which enables the network to consider the scale of the objects in the scene. This is important because it ensures the network can accurately represent objects of different sizes and resolutions. The scale-aware structure is achieved using a multiscale MLP, which allows the network to handle different scales of detail simultaneously. As a result, Mip-NeRF achieves better accuracy than NeRF, especially in scenes viewed at various resolutions. It reduces error rates by 60% compared to NeRF and is faster by 7% with only half the number of parameters. This is achieved by using a scale-aware structure and merging the separate “coarse” and “fine” MLPs of NeRF into a single MLP. This innovation has the added benefit of reducing the complexity of the model and the computational overhead, making Mip-NeRF more efficient than its predecessor.

Point-NeRF [63] is a new approach to NeRFs that address the computational challenges of training and rendering NeRFs. Point-NeRF represents the scene as a collection of points, each of which has a corresponding neural network that predicts the color and appearance of the point. This representation is much more efficient than traditional NeRFs, representing the scene as a continuous volume. Point-NeRF combines deep multiview stereo (MVS) [64] methods with neural 3D point clouds to generate high-quality view synthesis results. This method can be initialized via direct inference of a pre-trained deep network to produce a neural point cloud, which can be fine-tuned to surpass the visual quality of NeRF with a training time that is 30 times faster than actual NeRF. The representation used by Point-NeRF involves a neural point cloud with confidence values to represent the volume density and view-dependent radiance at any 3D location. A PointNet-like [65] neural network processes each neighboring neural point and regresses the view-dependent radiance and volume density. The inverse-distance weight aggregates neural features, and confidence values are used in the point removal process. Unlike NeRF, Point-NeRF operates entirely in 3D and avoids sampling empty scene space. The authors propose a pipeline for efficiently reconstructing point-based radiance fields using a neural network to generate and optimize a point-based field per scene with point growing and pruning techniques. Using a feed-forward neural network for efficient reconstruction, a neural generation module is then used to predict all neural point properties, including point locations, features, and point confidence. To generate 3D point locations, deep MVS methods are used, which involve cost volume-based [66] 3D CNNs, and a 2D CNN is used to extract neural 2D image feature maps from each image. Point clouds from multiple viewpoints are combined to obtain the final neural point cloud, and the point generation networks and representation networks are trained

end-to-end with a rendering loss. The pipeline significantly saves per-scene fitting time and achieves high-quality rendering. While the pipeline can generate an initial radiance field for a new scene, the authors propose optimizing the neural point cloud and MLPs in the representation to improve the quality further. However, the initial point cloud may contain holes and outliers that degrade rendering quality. To address this, the authors propose point pruning and growing techniques to gradually improve geometry modeling and rendering quality. Point pruning removes unnecessary outlier points based on point confidence values, while point growing adds new points to cover missing scene geometry in empty regions. The experiments demonstrate that Point-NeRF can achieve state-of-the-art results on several datasets and handle errors and outliers via a novel pruning and growing mechanism. Using Point-NeRF, high-quality, photorealistic images of scenes can be generated from novel viewpoints, including challenging scenes with complex geometry and lighting. Moreover, The authors shows that Point-NeRF is robust to noise, missing data, and outliers.

While Point-NeRF is a promising new development, at about the same time NeRFusion [67] came out to introduce a new neural framework for large-scale radiance field reconstruction. The proposed framework uses recurrent neural modules to reconstruct a large sparse radiance field incrementally from a long RGB image sequence. It is pre-trained across scenes, making it generalizable and efficient. An MLP network is used to regress the volume density and view-dependent radiance at any given 3D location from this volume, which can be used to synthesize images at novel target viewpoints. The radiance field representation is reconstructed locally and globally from image sequences using a shared MLP network, and both local and global volumes are modeled in the canonical world space. The authors propose a deep neural network that uses multiple images for per-frame reconstruction to improve scene geometry. The authors use deep MVS techniques to build a cost volume from image features and regress a neural feature volume. The authors construct volumes in the canonical world coordinate frame (WCF) [68] to align it with the final global volume output. Voxel pruning maximizes memory and rendering efficiency by removing non-essential voxels. The authors propose a method for rendering novel views of a scene using a NeRF represented by a sparse neural volume. They train their pipeline using rendering supervision with ground truth images without extra geometry supervision. The authors first pre-train the local reconstruction network and the radiance field decoder using a loss that makes the network learn to predict reasonable local neural volumes. Then the entire pipeline is trained with the local reconstruction network, fusion network, and radiance field decoder network together using a rendering loss. The resulting networks can produce a high-quality radiance field and realistic rendering results. The reconstructed radiance field can also be optimized per scene to boost the quality of the rendering. The proposed method uses a global neural volume fusion network to fuse local feature volumes from

each frame into a global volume. The fusion process uses GRUs (Gated Recurrent Unit) [69] with sparse 3D CNNs to recurrently fuse the per-frame local reconstruction and output high-quality global radiance fields. The GRU update gate and reset gate determine how much information is from the previous global volume. Current local volume is incorporated into the new global features, allowing the module to adaptively improve the global scene reconstruction while keeping the representation consistent. The current approach is suitable for large-scale indoor scenes but may need to be more efficient for unbounded outdoor scenes with foreground objects and distant backgrounds. This is because a uniform grid is used for the entire scene in the current method. Future work may address this by using per-view reconstruction in disparity space or applying spherical coordinates for regions at long distances. For the current pipeline, input frames are sampled uniformly, but a precise input view selection technique may be necessary to address various camera motions.

NeRFs are not able to deform. This is because NeRFs are trained on a fixed set of images, and they cannot be updated to reflect changes in the scene. DRF-Cages, on the other hand, can be deformed by manipulating the cage. This makes DRF-Cages a more versatile tool for representing and rendering 3D scenes. It works by first constructing a triangular mesh around the object that you want to deform. This mesh is called the "cage". The cage is then deformed using a standard mesh deformation algorithm. Once the cage has been deformed, the radiance field is updated to reflect the new shape of the object. The cage is generated by converting the optimized radiance field into a fine mesh and creating the corresponding cage for the generated mesh. However, inaccurate cage predictions due to complex shapes or fine details require manual resolution based on automatically generated cages. A simple cage can be built using 3D software for simple deformation or movement. The position of a point inside the cage is represented by cage coordinates, which are calculated based on the point's relative position to the cage vertices. The cage can be deformed by manipulating its vertices, and the new position of a point for the deformed cage can be calculated using the same cage coordinates. The radiance field accompanies the cage and is deformed along with it. Deformed-to-canonical mapping is used to map sampling points from the deformed space to the canonical space for color and density computations. The deformed radiance field is divided into three parts: outside both the canonical cage and deformed cage, inside the canonical cages but outside the deformed cages, and inside the deformed cages. The radiance field remains unchanged before and after deformation for the position outside the cages. The spatial position and view direction are mapped to the canonical space to query the color and density from the canonical radiance field for points inside the deformed cage. While the proposed method achieves promising results, it faces some challenges. One issue is the difficulty in representing detailed cage shapes while keeping a small number of cage vertices. The authors also highlight the need for more effective methods for automatic cage generation

from 3D scenes, requiring manual refinement, especially for real scenes with backgrounds. Failure cases of the approach include artifacts observed when parts of the scene are not well modeled due to occlusion or when parts of the object get drastically deformed. The first issue is challenging to address, as traditional optimization methods cannot handle unseen parts during training. However, occlusion-aware scene modeling or scene completion techniques may help.

NeRF is slower to render and harder to train, whereas FastNeRF [70] is faster to render and easier to train. The FastNeRF architecture splits the original NeRF neural network into two separate networks that only depend on positions and directions, respectively. By caching the output of these two functions, the rendering process is significantly accelerated, resulting in three orders of magnitude improvement in performance without compromising visual quality. The standard NeRF model requires a cache size of 5600 Terabytes, which is too large for consumer-grade hardware. In contrast, FastNeRF architecture reduces the cache size to 54 GB, making it feasible for consumer-grade hardware. The choice of k and l (the number of samples for the position and the direction) depends on the scene size and the expected image resolution. In many scenarios, a smaller cache is sufficient, which lowers the memory requirement even further. The authors have provided formulas in the supplementary materials to calculate cache sizes for both network architectures. The key advantage of FastNeRF is its ability to cache the outputs of the network, resulting in significant performance improvements without compromising visual quality. The method is 3000 times faster than the original NeRF and at least an order of magnitude faster than existing work while maintaining visual quality. The authors show that the factorization of the original NeRF into two separate functions provides other benefits, such as reducing the number of network parameters, enabling better control of the network, and simplifying the training process. FastNeRF's architecture is flexible and compatible with other techniques, such as multi-resolution representations and volume rendering. The authors provide several examples of the application of FastNeRF, including free-viewpoint videos and virtual reality. One of the limitations of FastNeRF is that it assumes that the scene is static and does not change over time. The authors acknowledge this limitation and suggest that future work could explore the extension of FastNeRF to dynamic scenes. Additionally, the authors note that the memory bandwidth of the GPU limits FastNeRF's performance and that further optimization may be possible.

Several studies indicate that it is possible to achieve real-time rendering by converting the neural representation into a discrete one after training [70–72]. However, when compared to KiloNeRF [73], these methods use a considerably higher amount of GPU memory, which could make them less appropriate for larger scenes. KiloNeRF, which speeds up Neural Radiance Fields with thousands of tiny MLPs, addresses these limitations by introducing a method for accelerating the rendering process of NeRF that synthesizes high-quality novel views of a scene. The method builds upon NeRF by

representing a scene as a collection of volumetric density and color values using a multi-layer perceptron (MLP) to encode densities and colors at any continuous 3D position in the scene. KiloNeRF subdivides the scene into a uniform resolution grid and assigns many small and independent MLPs, each tasked to represent the part of the scene that falls within a specific 3D cell. The KiloNeRF model is first trained by distilling the knowledge of an ordinary NeRF model and is then fine-tuned on training images with a photometric loss. Regularization strategies are applied to the weights and biases of the last two layers of the network, which are responsible for view-dependent modeling of color, to avoid artifacts in empty spaces without sacrificing visual quality. Equidistant points are sampled along the ray to reduce further the number of points evaluated by the network, and empty space skipping (ESS) and early ray termination (ERT) are used in this sampling process. ESS is implemented by populating an occupancy grid with binary values to indicate whether a cell contains any content, and the network is only evaluated if the cell is occupied. ERT is used to terminate ray evaluation when the transmittance value becomes close to zero. Grouping network queries by distance from the camera is used to implement ERT. The prototype implementation of KiloNeRF is based on PyTorch [74], MAGMA [75], and Thrust. Custom CUDA kernels were developed for various tasks, including sampling, empty space skipping, early ray termination, positional encoding, network evaluation, and alpha blending. To handle the simultaneous queries of thousands of networks, a custom routine was developed that fuses the entire network evaluation into a single CUDA kernel. The input batch is ordered to ensure the same network processes subsequent inputs. The use of MAGMA is essential for the proper handling of the simultaneous query of thousands of networks. The KiloNeRF method assumes a bounded scene, a limitation that must be addressed in future work. Nevertheless, the proposed method has shown impressive results, enabling the rendering of high-quality novel views of a scene at a faster speed than previous methods.

Trained on the dataset created from images captured from multiple self-driving vehicle drives, Block-NeRF [76] synthesizes views of large 3D scenes using neural networks. This technology is based on NeRF but with a twist: it divides the scene into smaller, manageable blocks to overcome the memory limitations of the original NeRF. The result is a variation of NeRF that can represent large-scale environments. The Block-NeRF paper proposed a solution for addressing the challenges in rendering large-scale environments by dividing them into individually trained Block-NeRFs, which are rendered and combined dynamically at inference time. The placement and size of these blocks are crucial to achieving full coverage of the target environment. One way to ensure coverage is to place a block at each intersection, with each block covering 75% of the way until it meets the next intersection. This leads to a 50% overlap between adjacent blocks and allows for easier appearance alignment. Additional blocks can be introduced where necessary to connect intersections, and training data

for each block can be kept within its intended bounds using a geographical filter. The block size can be variable, and other placement heuristics are possible if the entire environment is covered. This approach provides maximum flexibility, scales up to arbitrarily large environments, and allows for updates or introductions of new regions in a piecewise manner. To compute a targeted view, only a subset of the Block-NeRFs are rendered and composited based on their geographic location compared to the camera. Appearance matching ensures consistency between different Block-NeRFs, and the optimized appearance is iteratively propagated through the scene. This process is continued from the root Block-NeRF, and multiple surrounding blocks are considered when computing the loss. The authors also found that by providing the camera’s exposure information to the appearance prediction section of the model, NeRF can effectively counteract any visual discrepancies. The benefits of Block-NeRF are numerous. First, it can be used to explore scenes from new viewpoints, which can be extremely helpful for scientific research or creative expression. Second, it can be used to create highly realistic virtual environments for gaming or training purposes. Third, Block-NeRF has the potential to revolutionize the way we experience virtual reality by providing a more immersive and realistic environment. Finally, the model provides maximum flexibility, scales up to arbitrarily large environments, and allows for updates or introductions of new regions in a piecewise manner. There are, however, some limitations to the Block-NeRF model. One issue is that the model filters out transient objects during training via masking, which can cause artifacts in the resulting renderings. Additionally, the model has an issue with sampling distant objects with lower density, which leads to blurrier reconstructions.

While BlockNeRF uses a fixed grid of blocks, MegaNeRF [77] uses a dynamic grid that is adapted to the scene being rendered. This makes MegaNeRF a more scalable and efficient framework from large-scale visual captures using NeRF. Mega-NeRF proposes a framework for creating interactive 3D environments from large-scale visual captures using NeRF. The authors address the challenges of scaling up NeRFs for this purpose, including modeling thousands of images with varying lighting conditions, large model capacities, and slow rendering speeds. Mega-NeRF utilizes a sparse network structure and a geometric clustering algorithm for training and rendering large-scale scenes. The scene is decomposed into cells with centroids, and a corresponding set of model weights is initialized. A two-stage hierarchical sampling procedure with positional encoding is employed to record high-frequency details. Mega-NeRF generates opacity and color for a specific position, direction, and appearance embedding using the model weights nearest to the query point. The scene is tessellated into a top-down 2D grid to assign point queries to centroids at inference time. The scene is further subdivided into a foreground volume and a background volume, which is modeled with separate Mega-NeRFs. The camera’s altitude measurements terminate rays near ground level for efficient sampling. Each Mega-NeRF submodule can be trained in parallel with no

inter-module communication since each is a self-contained MLP. Only possibly relevant pixels that cross spatial cells are added to the trainset for each submodule to minimize the trainset size. A small overlap factor is included between cells to minimize visual artifacts near boundaries. Once NeRF gains a coarse understanding of the scene, irrelevant pixels/rays that do not contribute to a particular NeRF due to an intervening occluder can be pruned away, further reducing trainset sizes by 2x. The authors also propose a novel interactive rendering method and explore caching and temporal coherence to speed up rendering. In particular, they precompute a coarse cache of opacity and color and dynamically subdivide the tree throughout the interactive visualization to quickly produce an initial view and perform additional rounds of model sampling to refine the image. A final round of guided ray sampling is performed to further improve rendering quality. Rays are rendered in a single pass, and transmittance is accumulated along the ray to further accelerate the process. Evaluation of their method on existing datasets and their drone footage shows improvements in training speed and PSNR. Mega-NeRF also introduces a novel dataset of high-definition images collected through drone recordings covering an area of 100,000 square meters around an industrial complex, containing thousands of images. The proposed framework provides an effective and scalable solution for creating interactive 3D environments from large-scale visual captures using NeRFs. Mega-NeRF significantly improves training speed and rendering quality, making it a promising approach for real-time virtual fly-throughs in large environments. Additionally, the novel dataset introduced in this paper provides a valuable resource for future research in this area.

DeVRF [78] describe their new method for modeling non-rigid scenes using explicit and discrete voxel-based representations. Their approach involves modeling both the scene’s 3D canonical space and the 4D deformation field. The volumetric representation allows for an efficient query of deformation, density, and color of any 3D point at any time step in the scene. To learn the 3D volumetric canonical prior, a static-to-dynamic learning paradigm is employed from multi-view static images, which is then transferred to dynamic radiance field reconstruction. This prior provides essential knowledge of the 3D geometry and appearance of the target dynamic scene. The 4D voxel deformation field represents the motion of the deformable scene. Scene properties can be obtained by querying the scene properties of their corresponding canonical points through trilinear interpolation in the volumetric canonical space. Pixel colors can be calculated through volume rendering with the sampled scene properties along each ray. However, training DeVRF due to its large number of model parameters can lead to overfitting and suboptimal solutions. To address this, the authors propose a coarse-to-fine training strategy that progressively increases the resolution of the 4D voxel deformation field. This approach improves training efficiency and accuracy by removing most suboptimal solutions. Additionally, The authors use several loss functions to regularize the learned deformation field, including a re-rendering loss

to minimize the photometric MSE loss between observed and rendered pixels, a 4D deformation cycle consistency to enforce consistent deformation, optical flow supervision to ensure the induced flow matches the estimated flow, and total variation regularization to enforce smooth motion between neighboring voxels. The authors note that while DeVRF can quickly reconstruct deformable radiance fields, it has a large number of parameters. It also does not synchronously optimize the 3D canonical space prior to the second stage, which may limit its ability to model significant deformations.

Around the same time period of DeVRF, a group of researchers led by Li presented a novel framework called SteerNeRF [79], which utilizes the smoothness of the viewpoint trajectory to accelerate the rendering of NeRFs while preserving image quality. The rendering time of NeRF is proportional to the amount of computation required, which can be reduced by decreasing the number of samples on each ray. Existing acceleration strategies focus on pre-caching the output using a voxel grid, leading to large memory consumption. The proposed solution is to reduce the number of pixels and the number of samples for volume rendering by utilizing the smooth viewpoint trajectory. This solution can work in conjunction with existing methods to achieve high-speed rendering while maintaining visual quality. The process of recovering the output image uses a rendering buffer consisting of previous feature maps and depth maps combined with the current feature map at the current viewpoint, using a 2D neural renderer. The previous frames are reprojected to align with the current frame, and a lightweight 2D convolutional network is used for fast inference. A modified U-Net [80] is used, with a reduced number of convolution layers for high-resolution features and an increased depth of convolution layers for low-resolution features. An off-the-shelf inference acceleration toolbox, NVIDIA TensorRT, is used to optimize the neural renderer and reduce the inference time. The method uses a shallow U-Net as the 2D neural renderer and leverages a pre-trained NeRF model to synthesize more viewpoints as the pseudo-ground truth to address the overfitting problem when the number of training views is relatively small.

MobileNeRF [81] uses a new encoding scheme that is more compact than previous methods. This allows Mobile-NeRF to represent the same scene with less data, which makes it more efficient for rendering on mobile devices with limited memory. Mobile-NeRF aims to overcome the drawbacks of conventional NeRF rendering by presenting a new representation of NeRF that uses textured polygons, which can synthesize novel images efficiently using standard rendering pipelines. The MobileNeRF representation consists of textured polygons with opacity and feature vectors stored in the texture atlas, and a lightweight MLP running in a GLSL fragment shader [71] produces the output color. The training setup involves a fixed mesh topology and three optimized MLPs to minimize the mean squared error between predicted and ground truth colors of training image pixels. The predicted color is obtained through alpha compositing using radiance and opacity, both obtained by evaluating the MLPs at specific

positions. The representation is converted to a polygonal mesh after binarization and fine-tuning. Only partially visible quads are stored, and a texture image is created with a patch allocated for each quad. The discrete opacity and features are baked into the texture map by iterating over its pixels and converting the pixel coordinates to 3D coordinates. The rendering system is highly optimized for GPUs and can run at interactive frame rates on various devices. An interactive viewer is available as an HTML webpage with JavaScript rendered by WebGL via the threejs library.

B. High-Quality Large Dataset

NeRF requires a large dataset of high-quality images, which can be challenging to obtain and affect the quality of synthesized views. Researchers have tried to address this issue by adding prior conditions to the model, which can assist in image reconstruction or be used in a generative fashion to produce new images. Additionally, one-few-zero shot learning can generate 3D models with limited or no data, making NeRF useful in real-world applications. Priors can improve neural view synthesis and enable the reconstruction of images from sparse collections. In a generative manner, priors can generate new images based on prior information.

CAMPARI [82] is a more versatile approach that can generate objects using a broader range of input data. CAMPARI is a new approach for photorealistic image synthesis using deep generative models. While recent works have proposed 3D-aware models that render 3D scenes to the image plane for better consistency, this approach poses a challenge because it requires camera modeling. The authors address this challenge by proposing a solution that involves learning a camera generator and the image generator for a more principled approach to 3D-aware image synthesis. The model is unique in its approach to learning a camera generator and the image generator, allowing for more complex camera distributions without tuning. The scene is decomposed into the foreground and background for efficient and disentangled representation. As a result, the model can learn 3D-consistent representations and faithfully recover the camera distribution. Furthermore, it can generate new scenes with control over the camera viewpoint and shape and appearance during test time. The authors propose using stratified sampling and injecting prior knowledge into the sampling process to approximate numerical integration. The authors find that the model tends to reduce the camera pose range for better image quality, but this tradeoff can be avoided by keeping the camera generator fixed. The authors also find that the generated 3D representations are multi-view consistent but not as expected, resulting in “inverted faces.” To address this issue, the authors plan to investigate incorporating stronger 3D shape biases into the generator model. The authors compare their proposed model (CAMPARI) to baseline models and find that CAMPARI performs similarly or better, despite not requiring camera parameter tuning. The baselines depend on the camera parameters being correctly tuned, and their performance drops if the camera distribution of the data is not matched. The authors attribute this to the multi-view

inconsistencies introduced by the baselines. CAMPARI is able to learn the camera distribution and achieve 3D-consistent results without a learnable projection, making it more robust. The authors plan to investigate further how to incorporate stronger 3D shape biases into the generator model.

The latest research by Yin et al. [83] in implicit neural 3D representation has introduced a novel approach called CoCo-INR that addresses a significant limitation of the current state-of-the-art technique, NeRF. Existing methods require a dense set of calibrated views to produce high-quality 3D scenes, which makes them less effective with fewer input views. To overcome this limitation, CoCo-INR proposes injecting prior information into the coordinate-based network to enhance the feature representation and reduce the dependence on massive calibrated images. The proposed approach employs two attention modules: codebook attention and coordinate attention. The codebook attention extracts valuable prototypes from the codebook, while the coordinate attention enables each coordinate to query representative features from the prototypes. Integrating prior information through these attention mechanisms results in more accurate and efficient 3D representations. The CoCo-INR approach has been applied to multi-view scene reconstruction and novel view synthesis. The authors have used a combination of geometry and appearance networks to predict the scene, following the NeRF++ [84] framework. The networks are trained by minimizing the difference between the rendered colors and the ground truth colors without 3D supervision. The loss function includes a color rendering loss and an Eikonal term to regularize the geometry network. The injection of prior information through CoCo-INR provides several advantages over existing methods. First, it reduces the dependency on calibrated views, which is particularly important in scenarios where there are limitations on the number of views that can be obtained. Second, the approach enhances the quality of 3D representation and improves the accuracy of the predicted scenes. Finally, using attention mechanisms in the proposed approach provides a more efficient way to integrate prior information, which could benefit various applications.

Previous efforts have been made to make NeRF generalize to new scenes with only one or a few views, these approaches still require multiple views. Pix2NeRF [85] propose a pipeline for generating NeRF from a single input image in response to this challenge. Pix2NeRF is based on π -GAN [86], a generative model for 3D image synthesis that maps latent codes to radiance fields, and a reconstruction objective that includes an autoencoder with an encoder and π -GAN generator. What sets Pix2NeRF apart is that it is unsupervised, able to be trained with independent images, and has potential applications in 3D avatar generation, novel view synthesis, and super-resolution. However, the current limitations of Pix2NeRF are that it only works for one category per dataset and cannot generalize to new categories. To address this issue, alternative research directions are suggested, including local conditional fields similar to PixelNeRF [87], and GRF [88], which can generalize to unseen categories, multi-instance, and even real-world scenes. Pix2NeRF is not limited to using π -GAN as its backbone,

and newer generative NeRF models like EG3D [89] could potentially achieve better visual quality. Although architecture search, particularly with respect to the encoder, remains a challenging problem, the authors suggest utilizing more mature encoder architectures from 2D GAN feed-forward inversion literature, such as pixel2style2pixel [90], to potentially improve the performance of Pix2NeRF significantly. The authors expect their method to become a strong baseline for future research in this area and have demonstrated its superiority over naive GAN inversion methods through extensive ablation studies. Ultimately, Pix2NeRF offers a promising solution for generating NeRFs from a single input image, with the potential to revolutionize various fields, including computer graphics, virtual reality, and robotics.

3D generative models using coordinate-based MLPs to represent 3D radiance fields are currently state-of-the-art. However, this approach can lead to slow performance and changes in geometry or appearance when the camera pose is changed. The paper VoxGRAF [91] proposes an alternative approach using sparse voxel grid representations for efficient and consistent generative modeling. The authors replace the coordinate-based MLP in the 3D generator with a sparse 3D Convolutional Neural Network (CNN). The model consists of a 3D foreground generator and a 2D background generator. The foreground generator is based on the StyleGAN2 [92] architecture and uses a latent code and camera pose to map color and density values onto a sparse voxel grid. The background generator maps the latent code to a background image. Finally, the two images are combined using alpha composition to create the final image. The authors compared their method with state-of-the-art baselines and found that their approach resulted in more consistent multi-view results. Additionally, the authors compared their method with state-of-the-art methods that do and do not have a neural rendering pipeline. Rendering times are also reported in milliseconds per image, with the ability to separate scene generation and rendering, which is helpful for real-time applications. This approach has the potential to be faster and more efficient, leading to more accurate and consistent 3D generative models, especially in real-time applications.

VoxGRAF is a fast and efficient approach that can generate consistent multi-view results. However, it is not as accurate as HyperNeRFGAN [93], which can produce more realistic 3D objects. The authors present a new approach to 3D object representation that addresses the limitations of standard voxel and point cloud methods. The authors propose using NeRFs to synthesize 3D scenes from a small set of 2D images. The authors introduce the HyperNeRFGAN model, which combines the hypernetworks paradigm and NeRF representation. The generator takes a sample from a base distribution and returns the parameters for the NeRF model, which transforms spatial locations into emitted colors and volume density. The model is trained using the StyleGANv2 objective, and the noise vector is transformed to obtain the weights of the NeRF model. The generator produces 2D images with a 3D-aware NeRF representation to ensure accurate 3D object creation. The model has

a unique NeRF representation for each object and outperforms other approaches. However, the model’s limitation is that it uses only 2D images and not 3D information. Future work will include adding 3D mesh structure information to improve the model’s capabilities. Overall, the HyperNeRFGAN model presents a promising approach to 3D object representation that could have important applications in fields such as computer graphics, computer vision, and virtual reality.

LOLNeRF [94] addresses the limitation of NeRF by learning a generative 3D model from a single 2D image. The method allows the 3D structure to be rendered from different views without requiring multi-view data. To achieve this, all images in the dataset are aligned to a canonical pose, and a shared generative model with an autoencoder framework [95] is used. This allows the method to be trained with arbitrary image sizes by separating training complexity from image resolution. The method outperforms adversarial methods in representing object appearance and predicting geometry without geometric supervision. To improve generalization, two models are trained, one for foreground objects and one for the background. The method models shape as solid surfaces, improving the predicted shapes’ quality. The training process is optimized to reconstruct images from datasets and find optimal latent representations for each image without the need for rendering entire images or patches. Camera parameters are estimated using a pre-trained network that extracts 2D landmarks from the images. The model is capable of generating new 3D reconstructions given a pre-trained model and a latent code. It can be trained with arbitrary image sizes without increasing memory requirements during training and outperforms adversarial methods in representing the appearance of objects from the learned category. However, LOLNeRF relies on other methods for semantic information extraction and can lead to failure cases when the estimated pose or segmentation needs to be corrected. While the auto-decoder framework has advantages over GANs, it cannot maximize image “plausibility.” Future work aims to improve image quality by adding adversarial training to the existing method. The method can potentially augment image quality and improve the perceptual quality of images rendered from novel latent codes. This could be a possible direction for future work.

Traditional 3D reconstruction methods require a large number of input views to reconstruct a 3D scene accurately. DietNeRF [96] overcomes this limitation by incorporating prior knowledge from a pre-trained image encoder to guide the optimization process of the NeRF model. This prior knowledge comes in the form of a semantic consistency loss, which ensures that the high-level semantic features of the observed and rendered views are similar. Semantic consistency loss allows the DietNeRF model to capture stable high-level semantics across different viewpoints. The authors evaluate two sources of supervision for representation learning in DietNeRF: a CLIP model and visual classifiers pre-trained on ImageNet images. The CLIP model produces normalized image embeddings, while the visual classifiers use Vision Transformer architec-

tures that extract features from image patches and produce a single, global embedding vector. The pre-trained CLIP model used in DietNeRF is trained on millions of images with captions, providing rich supervision for image representations. The captions provide both semantically sparse and dense learning signals, which helps the image representation capture fine-grained details and high-level semantics. During training, the DietNeRF model is improved for efficiency and quality by using low-resolution semantic consistency renderings and sampling poses from a continuous distribution, LSC minimization, and mixed precision computation and memory-saving techniques. These optimizations allow the model to produce high-quality 3D reconstructions from a small number of input images while also reducing the computational requirements of the model. In experiments, DietNeRF is shown to significantly improve the quality of few-shot view synthesis, allowing it to render novel views with as few as one observed image. The model can also produce plausible completions of unobserved regions, making it useful for various applications, including virtual and augmented reality, robotics, and autonomous driving.

The multi-mirror catadioptric imaging system allows MirrorNeRF: One-shot Neural Portrait Radiance Field from Multi-mirror Catadioptric Imaging [97] to reconstruct a 3D scene from a single high-resolution image. MirrorNeRF introduces a novel approach to rendering photo-realistic human portraits for VR/AR applications. The traditional methods for capturing multi-view settings can be expensive and time-consuming. Still, MirrorNeRF provides a low-cost catadioptric imaging system with a single high-resolution camera that enables one-shot portrait reconstruction and rendering. The proposed approach uses a sphere mirror array arranged in a hexagonal pattern and a green screen as a background for foreground segmentation. The target subject is positioned in front of the mirror array for one-shot capture. The system calibrates using red calibration points to obtain the relative pose between the camera and the mirror array. The restored rays sample the plenoptic function of the mirror array, and the system associates image pixels with corresponding 3D points on the mirror surfaces to sample and reconstruct the captured scene. The proposed approach introduces a Neural Warping Radiance Field (NeWRF) to restore the 3D scene and produce photo-realistic images from arbitrary viewpoints. The NeWRF addresses the misalignment caused by manual arrangement and coarse calibration. Additionally, a density regularization scheme is proposed to improve rendering quality and training efficiency by leveraging inherent geometry information in a self-supervised manner. The warping displacement field is conditioned on the mirror to handle the misalignment caused by mirror position deviations. Experimental results demonstrate that the proposed approach is effective and robust in achieving photo-realistic, high-quality, and free-viewpoint rendering of human portraits. However, there are some limitations to the MirrorNeRF system. The requirement that all mirrors be almost on a plane means that the system can only reconstruct the front view of the subject. A high-resolution camera and

non-skewed viewing angle are also necessary to achieve high-fidelity results. The training process takes hours for a static scene, but the system’s requirements can be reduced to make it portable. However, the system is not suitable for dynamic scenes; incorporating neural priors or techniques to speed up training or making the neural warping field deformable in time series may lead to improvements.

Previous dynamic NeRF methods require dense images as input and can only model a single identity, which limits their applicability. FDNerF [98] requires only a small number of dynamic images and can be used with different people, making it more flexible than existing methods. Using a novel Conditional Feature Warping (CFW) module in the 2D feature space to handle inconsistencies in the dynamic frames and warp them to the desired expression, FDNerF eliminates the need for dense images and can model different identities with few-shot inputs. The CFW module is conditioned on expression and constrained by 3D geometry and consists of three sub-networks: a ResNet-like feature encoding network, a semantic mapping network, and a conditional warping network. The ResNet-like feature encoding network extracts features from the input frames, and the semantic mapping network maps these features to the target expression. The conditional warping network uses a motion descriptor to guide the warping process, and the aligned feature volumes are obtained through bilinear interpolation sampling. The warping fields are constrained by the 3D geometry represented in the jointly trained NeRF. To reconstruct 3D faces with a desired expression, the Radiance Field Reconstruction method is used. The process first extracts aligned feature vectors from the target feature volumes. Then it feeds them into the reconstruction module to estimate each spatial point’s color and density values. The feature vectors, along with the position and direction of the point, are processed by a neural network to produce the final results. The user can set the number of input frames flexibly, and the method eliminates the geometric discrepancy between views by adjusting color-related parameters only in the later stages of the network. To render images, FDNerF uses a hierarchical sampling strategy and two NeRF networks for coarse and fine reconstructions. The densities estimated by the coarse network are used for important sampling of query points in the fine network, allowing for efficient rendering and high-quality results. However, the authors note that there are limitations to the method, such as inconsistencies in non-face regions and lighting, which can cause blurriness in the result. The authors suggest that future works may overcome these limitations with separate warping fields for different parts and a reflectance field. While there are limitations to the method, the author’s suggestions for future works provide a promising direction for further improvements.

Unlike traditional methods that require 3D supervision, Jain et al. [99] utilize neural rendering and multi-modal image and text representations to generate objects. The method employs image-text models trained on large captioned image datasets to guide the generation process. The method optimizes a NeRF for high scores with a target caption based on a pre-trained

CLIP model [100]. Dream Fields trains a Radiance Field that renders images with high semantic similarity to a given text prompt, using pre-trained image-text retrieval models. The approach also incorporates simple geometric priors to improve fidelity and visual quality. An MLP with parameters optimizes the approach to produce outputs representing a scene’s differential volume density and color at every 3D point, solely dependent on the 3D coordinates, not the camera’s viewing direction. The method uses segments spaced at equal intervals with random jittering along the ray to compute the transmittance for rendering an image. To train Dream Field in a zero-shot setting, the approach randomly samples poses and uses a CLIP network to measure the similarity between the rendered image and the provided caption. The image and text encoders used in the CLIP network are from CLIP, and a baseline Locked Image-Text Tuning (LiT) ViT B/32 model [101], both trained contrastively on large datasets of captioned images. The proposed method supports 3D data augmentations by uniformly sampling camera azimuth in 360 degrees around the scene, and camera elevation, focal length, and distance from the subject can also be augmented. To improve coherence and reduce artifacts and spurious density, the method regularizes the opacity of Dream Field renderings by maximizing the average transmittance of rays passing through the volume. The transmittance loss is defined by the probability of light passing through N discrete segments of the ray and is compared to baseline sparsity regularizers. The Dream Fields approach can produce realistic, multi-view consistent object geometry and color from various natural language captions despite the scarcity of diverse, captioned 3D data. However, iterative optimization can be expensive, and using the same prompt for all perspectives can result in repeated patterns on multiple sides of an object. The image-text models used to score renderings are not perfect and can inherit harmful biases. Finally, Dream Fields does not target complex scene generation and does not handle scene layout. Dream Fields has shown promising results in producing realistic, multi-view consistent object geometry and color.

C. Representing Articulated Objects

Dealing with articulated objects in NeRF is a challenging problem in computer vision research. While NeRF has shown promising results in representing static scenes, it struggles to capture the dynamic and complex nature of articulated objects such as humans or animals. One of the main reasons for this is NeRF’s assumption of a static scene represented as a continuous volumetric function, which fails to consider the motion and deformation of the object’s geometry. Additionally, modeling each part of an articulated object explicitly using a set of parametric functions is not practical and hence requires new approaches for representation. Articulated object recognition is an active area of research, and recent works are applying the concept of NeRF for articulated models of people or cats, representing the shape of the object in the image using interconnected rigid parts.

iNeRF [23] introduces a new framework for 6 DoF pose estimation, which uses an observed image, an initial pose estimate, and a 3D object or scene as input to estimate the camera pose. Unlike traditional NeRF models that optimize weights using a set of given camera poses and image observations, iNeRF aims to recover the camera pose given the weights and the image by inverting the trained NeRF. The NeRF model’s ability to render an image observation by taking an estimated camera pose is utilized to minimize the photometric loss function and update the pose. The authors discuss different strategies for sampling rays in a NeRF optimization procedure to estimate the camera pose given an observed image. The authors’ found that sampling all pixels in the image is not feasible due to memory limitations and explore three strategies for selecting a smaller set of rays to compute the loss function. Random sampling is found to be ineffective when the batch size of rays is small, as most randomly-sampled pixels correspond to textureless regions. Interest point sampling, which samples from localized points of interest, converges faster but is prone to local minima. To address this, the authors propose interest region sampling, which samples from dilated masks centered on the interest points, and find that it speeds up the optimization when the batch size of rays is small. However, iNeRF has limitations as it does not model lighting and occlusion, negatively affecting its performance. One possible solution proposed in the paper is to include appearance variation with transient latent codes and optimize these codes along with the camera pose in iNeRF. The method takes approximately 20 seconds to run 100 optimization steps, making it unsuitable for real-time use. Improvements in NeRF’s rendering speed could mitigate this issue. The iNeRF method has succeeded in synthetic and real-world scenarios, including category-level object pose estimation. The authors believe that iNeRF can improve NeRF by estimating camera poses and integrating additional training data.

NeRF is limited in its ability to represent articulated objects due to the non-linear relationship between the kinematic representation [102] of 3D articulations and the resulting radiance field. NARF [103] addresses this limitation using a neural network to transform the 3D location and 2D viewing direction into density and RGB color value. The density controls the radiance of a ray passing through a location, and the network consists of two ReLU MLP networks, one for volume density and another for RGB color. NARF proposes a pose-conditioned NeRF model that can extend the representation capacity of NeRF to articulated objects that a kinematic model can describe. The kinematic model of the articulated object is represented by a tree structure of joints and bones, with each joint having its own transformation matrix. However, implicit transformations and part dependency issues limit the naive approach of concatenating the pose configuration vector as the model input. To overcome these issues, the object is decomposed into rigid object parts, each with its own local coordinate system defined by the rigid transformation obtained through forward kinematics. A rigidly transformed neural

radiance field (RT-NeRF) is used to model each part, and a single unified NeRF is trained to encode multiple parts while avoiding the part dependency issue. Estimating the radiance field is carried out in an object coordinate system where the density is constant with respect to a local 3D location. The authors presented two basic solutions for NARF: Part-Wise NARF ($NARF_P$) and Holistic NARF ($NARF_H$), and analyzed their pros and cons. Part-Wise, NARF decomposes the object into parts and trains a separate RT-NeRF for each part. This approach allows for better handling of part dependencies but can be computationally expensive. Holistic NARF models the entire object as a single RT-NeRF, which is computationally efficient but can lead to issues with implicit transformations. The authors propose a final solution, called Disentangled NARF, which combines the advantages of both Part-Wise and Holistic NARF. Disentangled NARF decomposes the object into parts, but the RT-NeRF for each part is trained in a common latent space, allowing for better handling of part dependencies while avoiding implicit transformations. The network uses a hierarchical volume sampling strategy to estimate the densities and colors of samples and then renders the color and mask of each camera ray using volume rendering. Experiments show that the proposed method is efficient and can generalize well to novel poses.

Animatable NeRF [104] extends NeRF by adding the ability to control the pose of the scene. The approach leverages the strengths of NeRF and the parameterized human model SMPL [105] to create high-quality human reconstructions. The authors optimize NeRF and SMPL parameters for better results and faster convergence by introducing pose-guided deformation and analysis-by-synthesis. The method works by mapping 3D position, shape, and pose into color and density. The authors use the SMPL model to model human appearance and geometry. The method aims to handle human movements between different frames by transforming the 3D position in the observation space into canonical space. This is achieved using the SMPL model to enable the explicit transformation of spatial points and reduce reliance on diverse input poses to generalize to unseen poses. The template pose is defined as X-pose, and the transformation of a point from observation space to canonical space is defined using the SMPL vertex set near that point. The blend skinning weight is used to characterize the movement patterns of a vertex along with the SMPL joints and to strengthen the movement impact of the nearest neighbor. The authors use volume rendering techniques to render the NeRF into a 2D image and obtain pixel colors by accumulating the colors and densities along the corresponding camera ray. The authors approximate the continuous integration by sampling points between the near and far planes along the camera ray and use a prior 3D mask to provide geometric guidance and deal with ambiguity during pose-guided deformation. The authors introduce an assumption for learning a more accurate NeRF that densities should be zero for points far from the surface of the human mesh and use a distance threshold to limit the distance between the sample point and the SMPL surface in the observation space.

The proposed method learns an animatable NeRF for human subjects by explicitly deforming the observation space under the guidance of SMPL transformations. However, inaccurate human body estimation can lead to blurry results. To address this problem, the SMPL parameters are fine-tuned during training using VIBE [106] to estimate the parameters for each frame as initialization of variables which will be optimized during training. This approach helps to obtain clearer and sharper results. Animatable NeRF can handle complex poses, but noticeable detail loss is observed compared to simple training videos due to inaccurate SMPL estimations and struggles to handle loose or non-rigid clothes, and slow movements and simple poses are recommended for high-quality rendering. The method cannot reconstruct invisible parts, so the input video should cover the whole body. The authors believe that animatable NeRF can have broad applications in various fields, such as film, virtual reality, and gaming. The proposed method can be extended to other objects or animals, with some pose estimation and deformation modifications. In the future, authors plan to explore ways to incorporate temporal coherence into the method and improve non-rigid objects' handling. The authors also plan to investigate further the use of generative adversarial networks (GANs) to enhance the visual quality of the reconstructed human avatars.

HumanNeRF [107] introduces a modified version of NeRF that incorporates human motion and is capable of generalizing to unseen poses and identities. HumanNeRF introduces a new method for generating high-quality and photo-realistic images of dynamic humans using sparse RGB streams. This is achieved through a modified version of NeRF that incorporates human motion and can be generalized. An aggregated pixel alignment feature is used to blend multi-view input images and a non-rigid deformation field to learn subtle displacement, which warps the human body from the current time frame to a common canonical pose. This results in a model that can predict the volume density and color at a given point before deformation using Dynamic Human Volume Rendering. However, there may be artifacts and imperfections when transferring motion to an unseen person due to the limited training data and diversity among different identities and scenes. To address this issue, the authors suggest a fast fine-tuning solution where the network is trained on various performers, and the feature blending network is frozen. When given an unseen subject, the network parameters of the deformation field and the generalizable NeRF are optimized. A neural blending scheme is introduced to refine the textures produced by NeRF rendering in multi-view settings, as The authors can contain artifacts and lack high-frequency details. This is done by first rendering a depth map from the target view, and each point in the map is back-projected into the neighboring views to fetch the colors and visibility. At training time, depth maps are rendered from synthetic human model datasets, and the corresponding image features are extracted. A neural appearance blending network is used to compute the blending weights, and the final color for each point is obtained by taking the dot product of the blending weights and the fetched colors

from the neighboring views. The generalizable NeRF module and the appearance blending network are trained separately. The network is optimized using a color loss and a silhouette loss, with the total loss being a combination of the two. The HumanNeRF approach is promising for generating high-quality and photo-realistic images of dynamic humans using sparse RGB streams. This method has potential applications in various fields, such as virtual reality, gaming, entertainment, education, and immersive telepresence. The fine-tuning solution proposed by the authors addresses the issue of limited training data and diversity among different identities and scenes. The neural blending scheme provides a solution to refine the textures produced by NeRF rendering in multi-view settings. The proposed approach effectively generates photo-realistic free-view synthesis, even for challenging human poses and motions, and can be used in various fields, such as VR/AR for gaming, entertainment, education, and immersive telepresence.

Generating realistic images of humans that can handle complex poses and lighting conditions is a challenging task. BANMo [108] presents a method to generate high-quality, animatable 3D models of flexible objects using multiple casual RGB videos. The method uses implicit neural functions to represent the 3D geometry and appearance of the object in a canonical space and a neural blend skinning model to restrict the object’s deformation space. BANMo models deformable objects in a canonical time-invariant rest pose space using implicit functions to represent the 3D shape, color, and dense semantic embeddings. The method uses Multilayer Perceptron (MLP) networks to predict the properties of a 3D point in the canonical space. The MLP network maps 3D points to a canonical feature embedding that can be matched by pixels from different viewpoints and lighting conditions. BANMo also employs a pair of time-dependent warping functions for forward and backward mapping and uses volume rendering to render images that account for deformation. It computes expected surface intersection and 2D re-projection to render 2D flow. To approximate articulated body motion, BANMo defines mappings between 3D points at different times by blending rigid transformations of 3D coordinates of bones. Poses are represented with angle-axis rotations and 3D translations, and skinning weights are assigned to bones using a skinning weight function that is conditioned on explicit 3D Gaussian ellipsoids. The method uses a coarse and fine component to model fine geometry’s skinning weights. BANMo uses canonical feature embedding to match 3D points with their corresponding pixel features in different time instances, allowing for joint optimization of shape, articulation, and embeddings. The embedding of a canonical 3D point is computed with an MLP that ensures the output 3D descriptor matches 2D descriptors of corresponding pixels across multiple views. The canonical embedding is self-supervised by enforcing consistency between feature matching and geometric warping. The canonical embeddings provide strong cues for the registration of pixels and ensure a coherent reconstruction with observations from multiple videos. BANMo optimizes

various parameters, including MLPs, learnable codes, and pixel embeddings, to minimize three types of losses: reconstruction losses, geometric feature registration losses, and a 3D cycle consistency regularization loss. The method uses an initialization strategy for root body poses to improve optimization. It provides a rough per-frame initialization of root poses using a network called PoseNet, which takes a DensePose CSE feature image as input and predicts the root pose. BANMo utilizes a pre-trained DensePose-CSE [109] to obtain a rough root body pose registration. A generic relative root pose estimator is needed to build generic deformable 3D model reconstruction pipelines. However, the optimization process is computationally expensive and needs to be sped up in future work.

D. Scene Editing

NeRF has limitations in its ability to edit scenes, such as adding or removing objects, and changing lighting or materials. This is because NeRF learns to represent a scene as a dense field of radiance, which means that any changes to the scene would require changes to the radiance field. This can be computationally expensive and time-consuming. Additionally, NeRF is trained on a dataset of images that have been captured in a specific way, so it may not be able to generalize to scenes that have been captured in a different way.

Liu et al. [110] presents a novel method for enabling users to edit and control an implicit continuous volumetric representation of a 3D object. The authors investigate three types of user edits: changing the appearance of a local part, modifying the local shape, and transferring color or shape from a target object instance. The network is based on a conditional radiance field that extends the NeRF representation, with latent vectors over shape and appearance. The representation is trained over a set of shapes belonging to a class, and each shape instance is represented by latent shape and appearance vectors. The authors describe the network architecture, training, and editing procedures in detail. The proposed model includes a shared shape network, an instance-specific shape network, a fusion shape network, and two separate networks for density and radiance predictions. The aim is to optimize a loss function to achieve accurate and efficient editing, where the edited radiance field should reflect the user’s desired change. The strategies involve updating specific layers of the network to speed up the optimization process and reduce the computational cost and subsampling user constraints during training. Feature caching is also applied to avoid unnecessary computation during optimization, reducing rendering time for a high-resolution image by 7.8x. These strategies achieve both accuracy and efficiency in editing the instance. To obtain the color at a pixel location for a desired camera location, the paper uses the “over” compositing operation. Users select a color and create a foreground mask to indicate where the color should be applied and can optionally create a background mask to leave the color unchanged. The model then updates the neural network and latent color vector to respect the user’s

constraints, using a reconstruction loss that calculates the squared Euclidean distance between output colors and the target foreground and background colors. A regularization term is also used to discourage large deviations from the original model, and the color editing loss is the sum of the reconstruction loss and regularization loss. The loss is optimized over the latent color vector and the neural network. There are two shape editing operations: shape part removal and shape part addition. The user scribbles over the desired removal region to remove a part in a rendered view. To construct the editing example, the regions corresponding to the foreground mask are whitened out, and a density-based loss is optimized, encouraging the inferred densities to be sparse. To add a part, the user selects a new instance and copies a local region into the original instance by scribbling. The modified regions are defined as the foreground region, the unmodified regions as the background region, and a reconstruction and regularization loss are optimized. The authors note that the method fails to reconstruct novel object instances that differ from other class instances. The authors are optimistic that NeRF rendering time improvements will help to reduce the interactive time. It takes over a minute for a user to get feedback on their shape edit, with the bulk of the computation spent on rendering views rather than editing itself. While there are limitations to the method, the authors are optimistic that NeRF rendering time improvements will help to reduce the interactive time.

In recent years, implicit neural rendering techniques have shown promising results for generating novel views of a scene. However, existing methods typically encode the entire scene as a whole, which can limit the method’s ability to edit high-level features of a scene, such as an object identity. To address this limitation, Yang et al. [111] propose a novel neural scene rendering method to generate high-quality novel views of a clustered, real-world scene with editing capabilities. The proposed method uses a two-pathway architecture consisting of the scene and object branches. The scene branch encodes the scene geometry and appearance, while the object branch encodes each individual object in the scene. The authors apply positional encoding to both the scene voxel feature and space coordinate for each point sampled along the camera ray to generate the hybrid space embedding. The scene branch function outputs the opacity and color of the scene, while the object branch function takes in the embedded object voxel feature and object activation code to output the color and opacity for the desired object. The object activation code is unique to each object and acts as a condition for the object branch to produce the desired output while everything else remains empty. To handle occlusion, the authors propose a 3D guard mask that identifies occlusion regions and stops the gradient applied to the occluded region during training. The model is jointly optimized for the scene branch and object branch, with the total loss defined as the sum of the losses of the two branches. The editable scene rendering pipeline is divided into the background and

object stages. In the background stage, the scene color and opacity are obtained while pruning the point sampling at the target region. In the object stage, rays are shot at the target objects, and the object-specific color and opacity are transformed to the desired location. Finally, all opacity and colors are aggregated, and pixel colors are rendered with quadrature rules. The proposed method is the first editable neural scene rendering method that supports high-quality novel view rendering and object manipulation. The method shows competitive performance for static scene novel-view synthesis and object-level editing. The authors’ extensive experiments demonstrate that the method can achieve realistic rendering and enable high-level editing tasks, such as moving or adding furniture. Overall, the proposed method represents a significant step forward in the field of implicit neural rendering and has the potential to open up new possibilities for realistic and interactive 3D graphics applications.

(D²NeRF) [112] is a new approach for generating high-quality NeRF models of static scenes. D²NeRF learns a 3D scene representation using separate NeRFs for moving objects and the static background. The static component uses NeRF, representing the scene as a spatial-dependent density and spatial-view-dependent radiance. In contrast, the dynamic component uses HyperNeRF [113] to capture scenes with non-rigid motion and topological changes accurately. The two models are composited to calculate the color of a camera ray by integrating radiance according to volumetric rendering within a predefined depth range. Regularizers are applied self-supervised to ensure correct separation between static and dynamic objects. Regularizers include binary entropy loss, skewed entropy loss, ray regularization, and static regularization. NeRFs face challenges in modeling standalone shadows. Therefore, the proposed solution incorporates shadows as a reduction in the radiance of the static scene using a shadow ratio that scales down the radiance of the scene. This is penalized for avoiding over-explaining dark regions of the scene. Shadows cast from dynamic objects onto other dynamic objects do not need explicit modeling. The proposed method relies on precise camera registration for successful decoupling and reconstruction. However, view-dependent radiance changes caused by reflective surfaces may be misinterpreted as dynamic effects, leading to incorrect decoupling. Removing texture-less targets that move within a small range is also difficult, as the motion clues are ambiguous. The authors demonstrate that their method performs better than existing approaches on a new dataset containing various dynamic objects and shadows. Overall, the proposed approach presents a novel way to decouple moving objects from the static background in a monocular video in a self-supervised fashion, using separate radiance fields for the static and dynamic portions of the scene.

Creating realistic and interactive virtual scenes is a complex task that requires the ability to manipulate and edit the underlying representations of these scenes. The NeRF have

shown great potential for this purpose, but editing them is difficult due to their non-object-centric nature. In response, Kobayashi et al. [114] proposes distilled feature fields (DFFs) as a new approach to enable local and interactive editing of NeRFs without the need for retraining. DFFs extend NeRFs by adding decoders for other quantities of interest, such as semantic labels, to compute density and view-dependent color. The authors propose a 3D zero-shot segmentation approach that uses open-set text labels or other feature queries to create a feature branch that models a 3D feature field describing the semantics of each spatial point. The authors supervise the feature field using a pre-trained pixel-level image encoder as a teacher network and optimize it by minimizing the difference between rendered features and the teacher’s features. To maintain the quality of reconstructed geometry, a stop-gradient is applied to density in rendering features. The DFF model calculates the probability of a label for a point in 3D space, and this segmentation can be used for interactive editing without retraining. This method allows for blending two NeRF scenes based on a segmentation field, and various edits can be applied to specific regions using this segmentation. The method also enables complex edits, including optimization-based methods, and allows for selective editing of specific regions in multi-object scenes. However, the DFF framework has two limitations. Firstly, the performance of the distillation process is limited by the teacher model and the resolution of the teacher encoders. This limitation may result from coarse-grained DFFs that cannot understand specific text queries. Secondly, the DFF relies on 3D reconstruction by NeRF, which can result in geometry errors in radiance fields that make the supervision of DFFs noisy.

IV. DISCUSSION

The NeRF approach has demonstrated impressive results in both the quality and efficiency of rendering photorealistic images of 3D scenes. Its use of a neural network representation allows for capturing complex, non-linear relationships in the data, resulting in a more accurate and detailed representation of the scene. This implicit representation also allows for more efficient rendering processes, eliminating the need for pre-processing steps such as voxelization or meshing. One of the most notable advantages of the NeRF approach is its ability to produce high-quality images of novel views of a scene, even under challenging lighting conditions. This capability is particularly valuable in applications such as virtual reality, robotics, and autonomous vehicles, where synthesizing views from arbitrary viewpoints is critical. In addition, NeRF is able to capture complex light transport effects, such as reflections and refractions, that are challenging for other methods to reproduce. Combined with its ability to handle variations in input data, it makes NeRF a promising method for view synthesis and other computer vision applications. Finally, the success of NeRF has inspired further research and development in the field of implicit representation learning and view synthesis,

which may lead to further breakthroughs in computer vision and graphics.

Table I presents a comparison of different methods for image reconstruction using various evaluation metrics such as PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and LPIPS (Learned Perceptual Image Patch Similarity). A higher PSNR indicates a more faithful reconstruction. SSIM is a measure of the similarity between two images. It is calculated based on the local statistics of the images. A higher SSIM indicates a more similar image. LPIPS is a measure of the perceptual similarity between two images. It is calculated using a deep neural network that has been trained to distinguish between real and fake images. A lower LPIPS indicates a more perceptually similar image.

Looking at the PSNR metric, we can see that most of the models perform poorly on the DTU dataset, with NeRF having the lowest score of 8.00, while on other datasets, models such as PointNeRF, Mip-NeRF, and HumanNeRF perform much better with scores of 33.31, 33.09, and 36.01, respectively.

The SSIM metric also shows similar results, with NeRF performing the worst on the DTU dataset, while PointNeRF, NuroFusion, and HumanNeRF perform much better on their respective datasets.

The LPIPS metric shows that most of the models have low scores, indicating that there is room for improvement in the neural rendering methods. However, some models such as KiloNeRF, Edit-NeRF, and NeRF-Editing have lower LPIPS scores, indicating that they perform better in terms of perceptual similarity.

Table I: Comparison table of various NeRF methods evaluated on different datasets based on PSNR, SSIM, and LPIPS scores.

Paper	PSNR	SSIM	LPIPS	Dataset Used
NeRF [1]	8.00	0.286	0.703	DTU [115]
CoCo-INR [83]	26.738	0.852	0.298	DTU [115]
DietNeRF [96]	14.242	0.481	0.487	DTU [115]
PointNeRF [63]	33.31	0.978	0.049	NeRF Synthetics [1]
NuroFusion [67]	31.25	0.953	0.069	NeRF Synthetics [1]
FastNerf [70]	29.155	0.936	0.053	NeRF Synthetics [1]
KiloNeRF [73]	31.00	0.95	0.03	NeRF Synthetics [1]
SteerNeRF [79]	30.97	0.948	0.065	NeRF Synthetics [1]
MobileNeRF [81]	30.90	0.947	0.062	Syntatic 360 [1]
Mip-NeRF [9]	33.09	0.961	0.043	Blander [9]
Mega-NeRF [77]	22.08	0.628	0.489	UrbanScene3d [116]
Pix2NeRF [85]	18.14	0.84	-	ShapeNet-SRN [117]
Block-NeRF [76]	23.60	0.649	0.0417	Alamo Square dataset
LOLNeRF [94]	25.3	0.836	0.491	CelebA-HQ [118]
FDNeRF [98]	24.847	0.821	0.142	VoxCelebdataset [119]
Edit-NeRF [110]	37.67	-	0.022	PhotoShapes [120]
NeRF-Editing [121]	29.62	0.975	0.024	Mixamo [122]
D ² NeRF [112]	34.14	0.979	0.090	Bag
DFFs [114]	32.85	0.932	0.162	Replicadataset
LearningObject [111]	15.0607	0.585	0.522	ToyDesk
NARF [103]	30.86	0.9586	-	THuman
HumanNeRF [107]	36.01	0.9897	0.0356	Multi-view dataset[107]

NeRF has become a popular tool for synthesizing photorealistic 3D scenes from 2D images. However, there are still some challenges and limitations associated with the NeRF approach; one of the major drawbacks of NeRF is its memory requirements, which can be a significant bottleneck for practical applications. The high memory consumption of NeRF

is due to the need to store the neural network parameters, which can be challenging for large-scale scenes or high-resolution images. The computational expense of NeRF is also a limitation, particularly during training. Evaluating the neural network for each ray in the scene is time-consuming, and techniques proposed to accelerate the training of NeRF may come at the cost of decreased performance. Another limitation of NeRF is its inability to handle dynamic scenes or moving objects. NeRF only learns a single representation of the scene, which needs to be improved to handle changes in the scene over time. Attempts to extend NeRF to handle dynamic scenes are an active research area. In addition, NeRF is sensitive to the quality of the input images, as it relies on accurate depth and camera pose estimates, which may be challenging to obtain in practice, especially for complex scenes. Finally, NeRF needs more interpretability, making understanding why certain views are synthesized correctly or incorrectly challenging. This is because NeRF uses a black-box neural network to represent the scene, which makes it challenging to analyze the internal workings of the model. Developing techniques for understanding and visualizing the learned representations of NeRF is an important direction for future research. Furthermore, NeRF's slow inference speeds, reliance on accurate pose estimation and multiple views, and limited effectiveness in scenarios with sparse views or poor camera calibration are additional limitations that must be addressed. Continued research efforts are needed to overcome these limitations and unlock NeRF's full potential.

V. CONCLUSION

The Neural Radiance Fields (NeRF) method has shown great potential for solving the challenging problem of image-based view synthesis. It provides a powerful and flexible representation of the 3D scene geometry and appearance using a continuous implicit function defined by a neural network. Our review has highlighted the various extensions and improvements to the original NeRF method, demonstrating the potential of this approach to solve more complex application scenarios. The extensions and improvements to the original NeRF method have made it more versatile, efficient, and capable of solving complex application scenarios. However, the high computational cost and lack of interpretability remain significant challenges that must be addressed in future research. Despite these limitations, NeRF has opened up new possibilities for virtual and augmented reality, gaming, and robotics. To further advance the field, the future research direction for NeRF can focus on improving interpretability, reducing computational overheads to achieve real-time rendering, exploring new application scenarios, and enhancing the scalability and versatility of the method. The NeRF has undoubtedly paved the way for new research directions and possibilities, making it an exciting time for computer vision.

REFERENCES

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng.

NeRF: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, dec 2021.

[2] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf efficient neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12902–12911, 2022.

[3] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *CoRR*, abs/2102.07064, 2021.

[4] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215. IEEE, 2021.

[5] Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pages 5742–5752. PMLR, 2021.

[6] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13144–13152, 2021.

[7] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[8] Michael Niemeyer and Andreas Geiger. GIRAFFE: representing scenes as compositional generative neural feature fields. *CoRR*, abs/2011.12100, 2020.

[9] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[10] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022.

[11] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[12] Thomas Müller, Alex Evans, Christoph Schied, and

- Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.
- [13] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.
 - [14] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qin-hong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *CoRR*, abs/2112.05131, 2021.
 - [15] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.
 - [16] Abril Corona-Figueroa, Jonathan Frawley, Sam Bond-Taylor, Sarath Bethapudi, Hubert PH Shum, and Chris G Willcocks. Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 3843–3848. IEEE, 2022.
 - [17] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6361, 2021.
 - [18] Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pages 5742–5752. PMLR, 2021.
 - [19] Connor Lin, Niloy Mitra, Gordon Wetzstein, Leonidas J Guibas, and Paul Guerrero. Neuform: Adaptive overfitting for neural shape editing. *Advances in Neural Information Processing Systems*, 35:15217–15229, 2022.
 - [20] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.
 - [21] Young Chun Ahn, Seokhwan Jang, Sungheon Park, Ji-Yeon Kim, and Nahyup Kang. Panerf: Pseudo-view augmentation for improved neural radiance fields based on few-shot inputs, 2022.
 - [22] Norman Müller, Andrea Simonelli, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Autorf: Learning 3d object radiance fields from single view observations, 2022.
 - [23] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.
 - [24] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021.
 - [25] Fu Li, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation, 2022.
 - [26] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, June 2022.
 - [27] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.
 - [28] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022.
 - [29] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcíński, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18623–18632, 2022.
 - [30] Yi-Ling Qiao, Alexander Gao, and Ming Lin. Neu-physics: Editable neural geometry and physics from monocular videos. *Advances in Neural Information Processing Systems*, 35:12841–12854, 2022.
 - [31] Danny Driess, Ingmar Schubert, Pete Florence, Yunzhu Li, and Marc Toussaint. Reinforcement learning with neural radiance fields, 2022.
 - [32] Chengliang Zhong, Peixing You, Xiaoxue Chen, Hao Zhao, Fuchun Sun, Guyue Zhou, Xiaodong Mu, Chuang Gan, and Wenbing Huang. Snake: Shape-aware neural 3d keypoint field, 2022.
 - [33] XRNeRF Contributors. Openxrlab neural radiance field toolbox and benchmark. <https://github.com/openxrlab/xrnerf>, 2022.
 - [34] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35:3165–3177, 2022.
 - [35] Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. *Advances in Neural Information Processing Systems*, 35:8144–8158, 2022.
 - [36] M Aharchi and M Ait Kbir. A review on 3d reconstruction techniques from 2d images. In *The Proceedings of the Third International Conference on Smart City Applications*, pages 510–522. Springer, 2019.

- [37] Hanry Ham, Julian Wesley, and Hendra Hendra. Computer vision based 3d reconstruction: A review. *International Journal of Electrical and Computer Engineering*, 9(4):2394, 2019.
- [38] Usman Khan, AmanUllah Yasin, Muhammad Abid, Imran Shafi, and Shoab A Khan. A methodological review of 3d reconstruction techniques in tomographic imaging. *Journal of Medical Systems*, 42(10):1–12, 2018.
- [39] Zhaoxi Pan, Song Tian, Mengzhao Guo, Jianxun Zhang, Ningbo Yu, and Yunwei Xin. Comparison of medical image 3d reconstruction rendering methods for robot-assisted surgery. In *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 94–99. IEEE, 2017.
- [40] Anny Yuniarti and Nanik Suciati. A review of deep learning techniques for 3d reconstruction of 2d images. In *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, pages 327–331. IEEE, 2019.
- [41] Hanene Ben Yedder, Ben Cardoen, and Ghassan Hamarneh. Deep learning for biomedical image reconstruction: A survey. *Artificial intelligence review*, 54(1):215–251, 2021.
- [42] Emmanuel Ahishakiye, Martin Bastiaan Van Gijzen, Julius Tumwiine, Ruth Wario, and Johnes Obungoloch. A survey on deep learning in medical image reconstruction. *Intelligent Medicine*, 1(03):118–127, 2021.
- [43] Muhammad Saif Ullah Khan, Alain Pagani, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Three-dimensional reconstruction from a single rgb image using deep learning: A review. *Journal of Imaging*, 8(9):225, 2022.
- [44] Lu Chen, Sida Peng, and Xiaowei Zhou. Towards efficient and photorealistic 3d human reconstruction: a brief survey. *Visual Informatics*, 5(4):11–19, 2021.
- [45] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.
- [46] Pratomo Adhi Nugroho, Dwi Kurnia Basuki, and Riyanto Sigit. 3d heart image reconstruction and visualization with marching cubes algorithm. In *2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC)*, pages 35–41. IEEE, 2016.
- [47] Abidin Çalışkan and Ulus Çevik. Three-dimensional modeling in medical image processing by using fractal geometry. *Journal of Computers*, 2017.
- [48] Mahani Hafizah, Tan Kok, and Eko Supriyanto. Development of 3d image reconstruction based on untracked 2d fetal phantom ultrasound images using vtk. *WSEAS transactions on signal processing*, 6(4):145–154, 2010.
- [49] Somoballi Ghoshal, Sourav Banu, Amlan Chakrabarti, Susmita Sur-Kolay, and Alok Pandit. 3d reconstruction of spine image from 2d mri slices along one axis. *IET Image Processing*, 14(12):2746–2755, 2020.
- [50] Qiufu Li and Linlin Shen. 3d neuron reconstruction in tangled neuronal image with deep networks. *IEEE transactions on medical imaging*, 39(2):425–435, 2019.
- [51] Fei Tong, Megumi Nakao, Shuqiong Wu, Mitsuhiko Nakamura, and Tetsuya Matsuda. X-ray2shape: reconstruction of 3d liver shape from a single 2d projection image. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1608–1611. IEEE, 2020.
- [52] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.
- [53] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.
- [54] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020.
- [55] Frank Dellaert and Lin Yen-Chen. Neural volume rendering: Nerf and beyond. *arXiv preprint arXiv:2101.05204*, 2020.
- [56] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul P. Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Nießner, Jonathan T. Barron, Gordon Wetstein, Michael Zollhöfer, and Vladislav Golyanik. Advances in neural rendering. *CoRR*, abs/2111.05849, 2021.
- [57] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.
- [58] Frank Dellaert. Nerf explosion 2020. <https://dellaert.github.io/NeRF>, Dec 2020. (Accessed on 12/12/2022).
- [59] Frank Dellaert. Nerf at iccv 2021. <https://dellaert.github.io/NeRF21>, Oct 2021. (Accessed on 12/13/2022).
- [60] Frank Dellaert. Nerf at cvpr 2022. <https://dellaert.github.io/NeRF22>, Jun 2022. (Accessed on 01/23/2023).
- [61] Mark Boss. Nerf at neurips 2022. https://markboss.me/post/nerf_at_neurips22, Sep 2022. (Accessed on 02/13/2023).
- [62] Frustum - wikipedia. <https://en.wikipedia.org/wiki/Frustum>. (Accessed on 03/08/2023).
- [63] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*

- Pattern Recognition*, pages 5438–5448, 2022.
- [64] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.
 - [65] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
 - [66] Introduction to cost volume. <https://academic-accelerator.com/Manuscript-Generator/Cost-Volume>. (Accessed on 03/29/2023).
 - [67] X. Zhang, S. Bi, K. Sunkavalli, H. Su, and Z. Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5439–5448, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.
 - [68] compas_fab - coordinate frames. https://gramaziokohler.github.io/compas_fab/latest/examples/01_fundamentals/02_coordinate_frames.html. (Accessed on 03/08/2023).
 - [69] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.
 - [70] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021.
 - [71] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021.
 - [72] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.
 - [73] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *International Conference on Computer Vision (ICCV)*, 2021.
 - [74] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
 - [75] Ahmad Abdelfattah, Azzam Haidar, Stanimire Tomov, and Jack Dongarra. Novel hpc techniques to batch execution of many variable size blas computations on gpus. In *Proceedings of the International Conference on Supercomputing*, pages 1–10, 2017.
 - [76] Matthew Tancik, Vincent Casser, Xinchun Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.
 - [77] H. Turki, D. Ramanan, and M. Satyanarayanan. Meganerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.
 - [78] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes, 2022.
 - [79] Sicheng Li, Hao Li, Yue Wang, Yiyi Liao, and Lu Yu. Steernerf: Accelerating nerf rendering via smooth view-point trajectory. *ArXiv*, abs/2212.08476, 2022.
 - [80] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
 - [81] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures, 2022.
 - [82] Michael Niemeyer and Andreas Geiger. Campari: Camera-aware decomposed generative neural radiance fields. In *2021 International Conference on 3D Vision (3DV)*, pages 951–961. IEEE, 2021.
 - [83] Fukun Yin, Wen Liu, Zilong Huang, Pei Cheng, Tao Chen, and Gang YU. Coordinates are not lonely—codebook prior helps implicit neural 3d representations. *arXiv preprint arXiv:2210.11170*, 2022.
 - [84] Kai Zhang, Gernot Riegler, Noah Snively, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *CoRR*, abs/2010.07492, 2020.
 - [85] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In

- Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3981–3990, June 2022.
- [86] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.
 - [87] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4585. IEEE, 2021.
 - [88] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021.
 - [89] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022.
 - [90] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.
 - [91] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. VoxGRAF: Fast 3d-aware image synthesis with sparse voxel grids. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
 - [92] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
 - [93] Adam Kania, Artur Kasymov, Maciej Zięba, and Przemysław Spurek. Hypernerf-gan: Hypernetwork approach to 3d nerf gan, 2023.
 - [94] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022.
 - [95] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.
 - [96] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.
 - [97] Ziyu Wang, Liao Wang, Fuqiang Zhao, Minye Wu, Lan Xu, and Jingyi Yu. Mirrornerf: One-shot neural portrait radiance field from multi-mirror catadioptric imaging. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2021.
 - [98] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Fdnerf: Few-shot dynamic neural radiance fields for face reconstruction and expression editing. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.
 - [99] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022.
 - [100] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
 - [101] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.
 - [102] Kinematic models | smithsonian institution. <https://www.si.edu/spotlight/kinematic-models>. (Accessed on 04/09/2023).
 - [103] Atsuhiko Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5762–5772, 2021.
 - [104] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular rgb videos. *arXiv preprint arXiv:2106.13629*, 2021.
 - [105] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.
 - [106] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020.
 - [107] F. Zhao, W. Yang, J. Zhang, P. Lin, Y. Zhang, J. Yu, and L. Xu. Humannerf: Efficiently generated human radiance field from sparse inputs. In *2022 IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7733–7743, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.
- [108] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873, 2022.
 - [109] Natalia Neverova, David Novotny, Marc Szafraniec, Vasil Khalidov, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. *Advances in Neural Information Processing Systems*, 33:17258–17270, 2020.
 - [110] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.
 - [111] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.
 - [112] Tianhao Walter Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D²nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
 - [113] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.
 - [114] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
 - [115] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.
 - [116] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: the urbanscene3d dataset. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 93–109. Springer, 2022.
 - [117] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.
 - [118] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.
 - [119] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *CoRR*, abs/1706.08612, 2017.
 - [120] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *ACM Trans. Graph.*, 37(6), dec 2018.
 - [121] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022.
 - [122] Adobe Inc. Mixamo. <https://www.mixamo.com>. (Accessed on 04/12/2023).