

Trini: An Efficient Representation of Dynamic Scenes for Sparse-View Camera Settings

Rishav Bhardwaj
Systems Design Engineering and Vision Science
University of Waterloo
Waterloo, Canada
rishav.bhardwaj@uwaterloo.ca

John Zelek
Systems Design Engineering
University of Waterloo
Waterloo, Canada
jzelek@uwaterloo.ca

Vasudevan Lakshminarayanan
Optometry and Vision Science
University of Waterloo
Waterloo, Canada
vengu@uwaterloo.ca

Abstract—3D reconstruction of a dynamic scene has been a challenging task in vision. Several strategies have been developed to enhance the reconstruction of dynamic scenes, with some employing tri-projection decomposition techniques that surpass D-NeRF in terms of speed and effectiveness. This paper introduces Trini, which decomposes a dynamic 3D scene into three volumes dealing with the 3D coordinates influenced by time. Each volume is further structured with four marginalized planes. These planes are then integrated with a compact MLP for rendering superior results in a seamless manner. Additionally, we incorporate a technique to efficiently determine coordinates in a set of distinct images for enhancing the reconstruction process for cases involving sparse-view camera images. The efficacy of our method outperforms other state-of-the-art techniques and is particularly evident in capturing the dynamic elements and edges present in the scene.

Keywords-dynamic 3D reconstruction;

I. INTRODUCTION

Reconstructing a 3D scene from a set of 2D images has a plethora of applications in augmented reality, virtual reality and robotics. Representing a static scene is inherently time-consuming, and capturing a dynamic scene poses an even greater challenge. Recently, there has been significant progress in 3D reconstruction using Neural Radiance Fields (NeRF) [1]. By supplying a 5D input of spatial coordinates (x, y, z) and camera view angles (θ, ϕ) to a multilayer perceptron (MLP), NeRF can effectively encode this mapping into emitted radiance values and volume density. NeRF has improved in terms of accuracy and computation time [3], [4], [5]. Remarkable speedups in training time were observed when employing the approaches proposed by [11], [12], [13]. These methods leverage explicit spatial data structures, which are then decoded by compact MLPs. Consequently, the training time is automatically reduced, as there is no longer a single MLP tasked with learning the entire scene.

NeRFs were also extended to dynamic scenes [6], [2], [7], [8]. It has been seen that using MLPs directly to reconstruct a static scene takes several hours along with great computation power. For dynamic scenes, this process is even further time consuming and complex. It sometimes takes days of GPU time to reconstruct a simple dynamic synthetic scene,

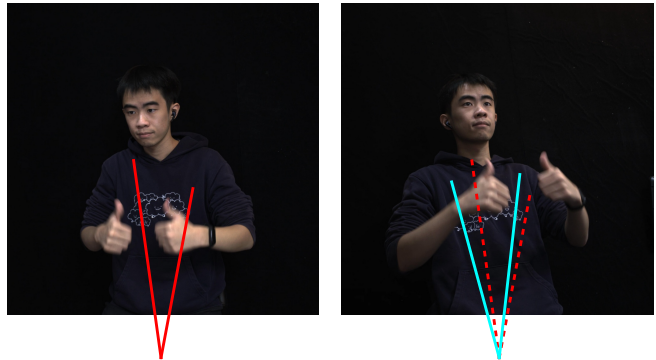


Figure 1. Given a pair of images after passing through CNNs, red dotted lines show how Tensor4D[10] follows the same set of coordinates from the first image. Our proposed roving technique enhances the flexibility in feature extraction (blue lines). It is responsible to find the best suitable new set of coordinates in the second image for the better reconstruction of the scene, check figure 2

thus making it quite impractical to be adopted for real-life applications. Undoubtedly, substantial GPU time was a necessity. Nevertheless, these findings laid the crucial foundation for advancing towards the reconstruction of dynamic 3D scenes. HexPlane [15] inspired by the work of TensorRF[11] extended its application from static scenes to dynamic scenes. The training time was significantly reduced in this case and could also be used for a real set of images. Tensor4D[10] is also like another variant of Hexplane[15] which uses nine planes instead of six. Additionally, it uses convolutional neural networks (CNNs) for the images from sparse-view camera settings. Our work extends the work of Tensor4D[10] to get better results.

Another variant for dynamic reconstruction was also introduced by TiNeuVox[17] which tried to represent scenes with time-aware voxel features. The given 3D point is first represented to a canonical space and then the voxel features are fetched. This work is a fast way to reconstruct a dynamic scene quickly. However, it does fail to render high quality images.

In this paper, we introduce Trini, a method designed to



Figure 2. The first row shows the output after employing the proposed roving technique. The second row is the output after training the network with the same number of epochs without the roving technique. Roving technique helps rendering sharper images which can be clearly seen.

efficiently represent dynamic scenes. We build upon the tri-projection decomposition technique initially employed in EG3D [9]. This approach enables us to encode the dynamic scene in a memory-efficient manner, reducing reliance solely on the MLP for scene reconstruction. As a result, both training and rendering times are significantly decreased. Trini decomposes the dynamic scene into three volumes using the tri-projection decomposition technique. Each of these volumes comprises pairs of coordinates, as well as pairs of coordinate axes concatenated with time. This results in a total of four pairs of streamlined planes within each volume. Trini encodes the 4D point in these three volumes and it is further passed through a compact MLP to obtain better results. Unlike Tensor4D [10], which uses same set of coordinates to extract features from the set of images, we employ a roving technique which enables flexibility in terms of feature selection. In the roving technique, we employ a small MLP which gives flexibility in choosing different coordinates from the set of images. This roving technique is also pictorially shown in Figure 1 and how employing this efficient technique has helped rendering sharper set of images which can be seen in Figure 2. Lastly, drawing inspiration from FreeNeRF[14], we incorporate a regularization scheme to control the frequencies of input provided to our proposed architecture during the initial epochs. By integrating the proposed decomposition and roving techniques, along with the regularization of input to the architecture, we achieve sharper renderings of novel views

at varying time points. Through comprehensive comparisons with existing state-of-the-art methods, we demonstrate the superior performance of our approach on the sparse-view camera settings Tensor4D dataset [10].

Thus, we summarize our contributions as follows:

- We introduce an architecture that efficiently decomposes dynamic 3D scenes and subsequently renders them in high-quality from novel viewpoints and times.
- Leveraging the roving technique for sparse-view camera settings, we process a set of images simultaneously. This technique proves effective in extracting features from the resulting feature maps after passing the images through CNNs.
- Through comprehensive evaluations on sparse-view camera settings, our method consistently outperforms other state-of-the-art methods across a majority of the evaluation metrics, showcasing superior rendering quality, especially on the border parts of the masked images.

II. RELATED WORK

Neural Implicit Representations. Recent advancements in 2D deep learning have pushed the boundaries into the realm of 3D. Rigid geometries can be effectively represented using methods such as point clouds [22], octrees [23], and voxels [24], [25]. Implicitly encoding 3D scenes [26], [27], [28], [29], [30], [37], [38], [39], [40] through neural networks has garnered significant attention and progress in recent research. Initial endeavors in this domain necessitated ground-

truth 3D geometry, posing a challenge due to reliance on external sensors for depth data. This limitation hindered its applicability in real-world scenarios, where suitable datasets are not readily available. Subsequent advances alleviated this requirement, relying solely on image inputs.

The introduction of NeRF[1] marked a significant milestone in 3D reconstruction. By providing 3D coordinates and camera view angles to an MLP, NeRF enabled the reconstruction of 3D scenes. Subsequent iterations and variants [16], [31], [32], [21], [33], [34] further refined and expanded upon NeRF’s capabilities, demonstrating exceptional results in novel view synthesis [35], [3], [36]. In GeoNeRF[59], the authors bifurcated their approach into two distinct phases. The first phase involves the creation of a volume using a transformer-based attention mechanism, utilizing nearby views provided as inputs. The second phase is dedicated to rendering from the generated volumes. In NSVF[41], implicit fields influenced by voxels were utilized for 3D surface reconstruction.

The versatility of NeRF extended its applications to diverse fields, including image generation [42], [43], [44], [45], [46] and human rendering [47], [48], [49]. Mixture of implicit and explicit representations [11], [56], have significantly reduced the training time significantly. In a noteworthy contribution, FreeNeRF[14] emphasized that while these approaches involve intricate mathematics and geometry for 3D scene reconstruction, such complexities can be circumvented by regulating the input frequencies provided to the network. In the initial epochs, the input was directly fed with low frequency input data only, enabling the network to quickly minimize loss. After a specific number of epochs, higher frequencies were introduced to capture finer details.

Another variant of NeRF[1], Ref-NeRF[4] focused on rendering glossy surfaces. While NeRF[1] demonstrated commendable performance, particularly on Lambertian surfaces, it faced challenges when dealing with non-Lambertian surfaces. To address this, regulators were applied to the normal vectors, enabling the accurate rendering of specular reflections. Another unique variant Deblur-NeRF[60], whose main objective is to render sharp rendering from the blurry inputs. These blurry inputs generally happen due to camera-motion or of-focus of image.

Although, an essential requirement of NeRFs[1] are dense set of input images and it faces huge performance issues if this requirement is not met. This requirement is not realistic and dense set of images of a scene are not always readily available. MixNeRF[61] gave the ability to reconstruct a scene from sparse set of camera poses. It estimates the joint distribution of RGB colors along the ray samples taken from the sparse set of camera poses.

Inspired by the work of 2D generative models [65], where a set of latent code is fed as input to the network and finally an output is retrieved from it, DeepSDF[37] and IM-

NET [38] established the signed distance fields based on the latent code given as input. Similarly ONet[26], created an occupancy field while SRN[29] encoded a representation which was decoded using LSTMs and were dependent on the cameras poses only and not on the depth information.

In TensorRF[11], the EG3D technique [9] is employed to encode the 3D scene in an efficient manner. Instead of depending on resource-intensive attention-based models or large-scale MLPs for reconstruction, this approach efficiently represents dimensions through a set of volumetric planes. This not only reduces training time, but also enhances rendering quality, reduces rendering time, and minimizes the number of parameters utilized. Mip-NeRF[16] approximated integral of the conical view frustum of the points along the ray. Exact-NeRF[62] extended the work of Mip-NeRF[16] by exploring how the results would be affected if upon no approximation of the integrated positional encoding by using the pyramid-based integral formulation.

Neural Renderings for Dynamic Scene NeRF[1] was extended to dynamic scenes to be used for real-life applications [50], [51], [52]. D-NeRF[6] stands as one of the first works in enabling NeRFs[1] to handle dynamic scenarios. Instead of directly incorporating the time variable into the input, D-NeRF[6] first transformed the input to a canonical space, followed by the use of another MLP to render the output. Although this approach yielded promising results, it was characterized by extensive training time like 30-35 hours for a synthetic dynamic scene. [50], [51], [52] are the other works that extended NeRF[1] to dynamic scenes for real-life applications. Time-aware MLPs have been introduced [17], [53], [54], [55] to reduce the training time of dynamic NeRFs. While MLPs can expedite processing time in comparison to previous architectures, their efficacy as a solution is not consistently advantageous. The reduction in time, although noticeable, is not substantial, and there is no significant improvement in rendering quality.

Certain methods, such as Space-time Neural Irradiance Fields[57], operate by taking a spatiotemporal irradiance field and subsequently estimating depth from individual frames of a video. On the other hand, approaches like [58], [8] rely on point trajectories to gain insights into the presented scene. DynIBaR [63] gives the ability to render images from novel views and times by reconstructing dynamic 3D scenes of monocular videos. It uses an MLP to estimate how a point in 3D moved with respect to time. It finds the trajectory of the point by taking the next three and previous three frames into consideration. This technique was combined with an attention-based architecture to render images. It takes almost two days to train the entire architecture for reconstructing a dynamic scene. In RoDynRF [64] the reconstruction of the dynamic scene happens without the camera poses and parameters. Since the Structure from Motion algorithms sometimes do not estimate the right pose values for complex videos, RoDynRF [64]

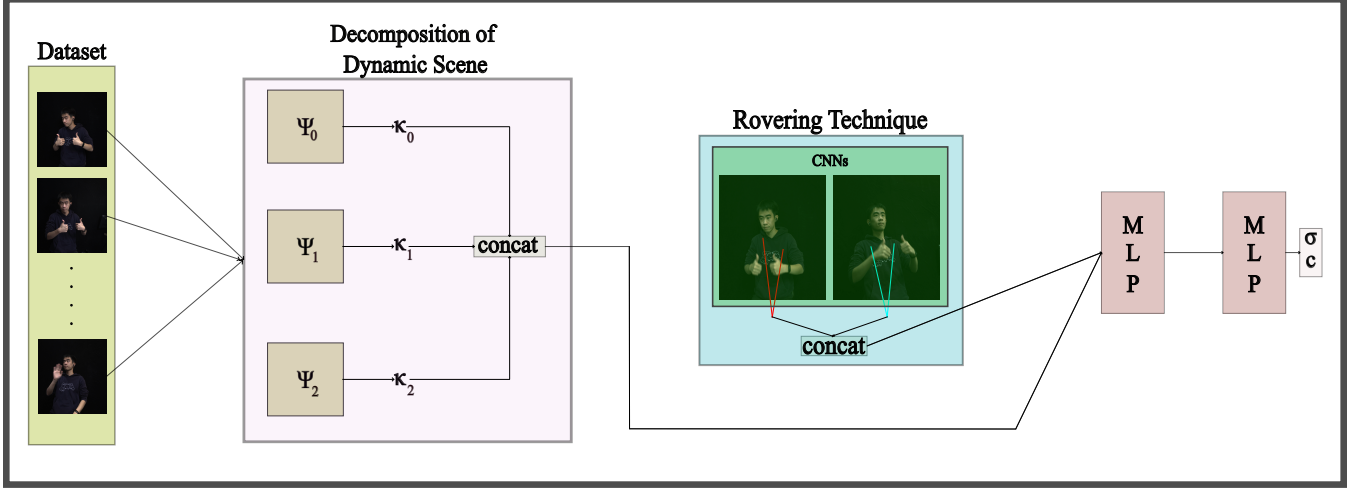


Figure 3. The comprehensive architecture is depicted for the sparse-view camera settings. Here, each Ψ represents a time-dependent volume, visually represented in Figure 4. These three volumes play a crucial role in the decomposition of the dynamic scene, followed by the concatenation of features extracted from the CNNs’ feature map using the roving technique. Finally, there are two MLPs used to decode the extracted features and render the the color and density of the 4D point.

aims to tackle this problem by estimating the camera poses and parameters as well as the static and dynamic radiance fields.

In our study, we take the idea of Hexplane[15] which involves simultaneous incorporation of all three coordinates along with time. We extend this approach to align with the principles outlined in Tensor4D[10], aiming for a more comprehensive understanding of dynamic scenes. In our exploration of image sets within sparse-view camera settings, we identify a challenge in efficiently extracting features from the feature map. To address this, we propose the roving technique, which proves effective in overcoming this limitation. Ultimately, our experiments demonstrate that our approach outperforms the work in Tensor4D[10], which served as the foundation for our research, in terms of performance, surpassing even other state-of-the-art methods.

III. METHOD

We now introduce Trini, a new architecture which renders images from novel poses and times. Figure 3 provides an overview of the model. In Section III-A, we elucidate the core concept of this paper, which enables the representation of the dynamic scene through 12 rationalized planes, organized into three volumes. Section III-B introduces the proposed roving technique. Here, a compact MLP that facilitates the extraction of features from the feature map after passing the image set through the CNNs. Just like NeRF [1], the proposed model outputs color and opacity of the points that are subject to change over time.

A. Decomposition of Dynamic Scene

Instead of representing the dynamic scene in a brute force way of 4D data, which is not only memory-intensive but

also time-consuming, we decompose this scene using the tri-projection decomposition technique. This technique leverages factorization [9] which is applied to the 3D volumes. The proposed method is an extension of the work of [10]. We represent the dynamic scene in three time-specific volumes where each volume can be expressed in the generic form as:

$$\Psi_i(a, b, c, t) = \{f(a, b), g(a, t), g(b, t), g(c, t)\} \quad (1)$$

$$q = \varrho_i^0(f(a, b), g(a, t), g(b, t)) \quad (2)$$

$$\kappa_i = \varrho_i^2(q, g(c, t)) \quad (3)$$

where $i = \{0, 1, 2\}$ denoting the volume number, a, b, c are the coordinates of the three dimensions without any specific ordering, while t denotes time. $f(\cdot, \cdot) \in \mathbb{R}^C$ is defined as the combination of two of the coordinates from the three dimensions whose features are extracted from a learnable parameter $M \in \mathbb{R}^{P \times Q \times C}$. Similarly, $g(\cdot, \cdot) \in \mathbb{R}^C$ is defined as the combination of one of the coordinates with time and the features are extracted from the learnable parameter $N \in \mathbb{R}^{G \times H \times C}$. P, Q, G, H and C are the dimensions of the learnable parameter M and N respectively with C being the same depth across both the parameters. ϱ is a compact volume specific MLP of two layers with the layer number mentioned in the superscript. κ is the output of each volume denoted in the subscript.

Each time specific volume can be represented pictorially in Figure 4. Thus for x, y, z coordinates across three dimensions and time denoted as t , we can represent it mathematically in three time specific volumes as:

$$\Psi_0(x, y, z, t) = \{f(x, y), g(x, t), g(y, t), g(z, t)\} \quad (4)$$

$$\Psi_1(y, z, x, t) = \{f(y, z), g(y, t), g(z, t), g(x, t)\} \quad (5)$$

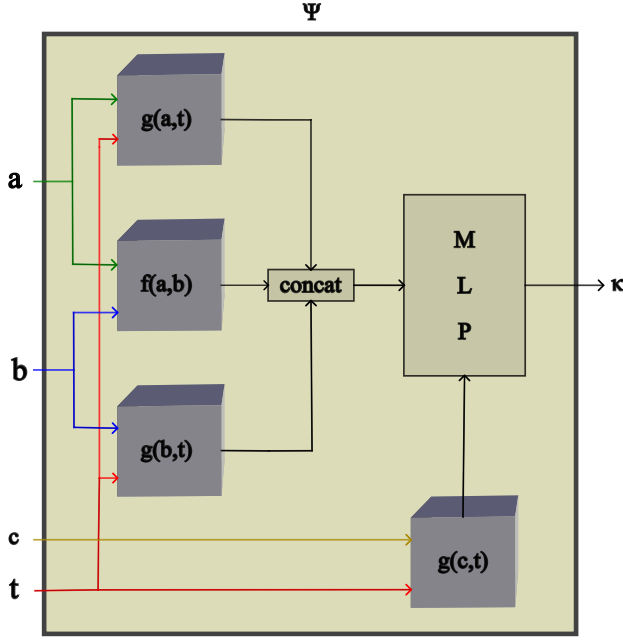


Figure 4. Generic representation of each time-dependent volumes where it consists of four planes. a , b and c are nothing but 3D coordinates that are fed as input and t is time. Mathematically this figure is represented in Eq1, Eq2 and Eq3. $g(c, t)$ is fed into the second layer of MLP, also shown in Eq3.

$$\Psi_2(z, x, y, t) = \{f(z, x), g(z, t), g(x, t), g(y, t)\} \quad (6)$$

we finally obtain the κ from these three volumes and concatenate them which can be represented as:

$$\tau = [\kappa_0, \kappa_1, \kappa_2] \quad (7)$$

where τ holds the efficient decomposed result of the three time specific volumes and $[\cdot]$ represents the concatenate operation.

B. Rovering Technique

In contrast to [10], which employs same set of coordinates for the extraction of features from the feature map, our proposed approach incorporates the use of different coordinates for every image in the set. This choice enables to offer more detailed information about the points in space that undergo changes over time. Furthermore, to enhance flexibility in feature selection, we implement a small MLP μ exclusively for the other images within the set. This MLP is tasked with retrieving the most pertinent details, as opposed to simply fetching features based on the coordinates of the first image. Please refer to Figure 1 for a visual representation. Due to the inflexibility of feature extraction in [10], the network is unable to learn the fine details which are essential for rendering sharper images. With the rovering technique, the model is pliable to choose features with ease in order

to render sharper images. Given a 3D point in space P , the other point Q for the second image in the set can be mathematically shown as:

$$p = \mu(P, A) \quad (8)$$

$$Q = P - (p \odot K) \quad (9)$$

where $A \in \mathbb{R}^E$ whose features are fetched from a learnable parameter $U \in \mathbb{R}^{F \times E}$ and F is the number of images in the dataset. Similarly, $K \in \mathbb{R}^3$ is extracted from a learnable parameter $V \in \mathbb{R}^{F \times 3}$ and \odot is element-wise product.

Employing this rovering technique effectively increases the quality of the rendered images from novel viewpoints, as illustrated in Figure 2.

IV. TRINI FOR SPARSE-VIEW CAMERA SETTINGS

In this section, we demonstrate the working of Trini for sparse-view camera settings. In Section IV-A, we integrate the dynamic scene decomposition outlined in Section III-A with the rovering process over image pairs described in Section III-B. In Section IV-B, we show the loss functions used in order to reconstruct the scene in an effective way.

A. Sparse-View Camera Dataset

In sparse-view camera settings, we initially break down the scene using the tri-projection decomposition technique, yielding τ as shown in Equation 7. Subsequently, the image set undergoes processing through the CNNs, and features are extracted from the resultant feature map. This is done after obtaining the 3D points of the image pairs using the rovering technique as described in Section III-B.

We combine the features derived from the tri-projection decomposition technique with the feature map from the CNNs. Additionally, we include the time frame of the image, as well as the mean and cone embeddings obtained after processing the 3D point through Mip-NeRF's [16] positional encoding. This can be expressed mathematically as:

$$v, \sigma = \Gamma_{ref}(\tau, \delta, m, t, c_e) \quad (10)$$

where Γ_{ref} is the MLP, δ denotes the features extracted from the CNNs after employing the rovering technique, m and c_e are the mean and cone embeddings from [16] and t is the time. v and σ correspond to the resulting features and density.

Drawing inspiration from FreeNeRF [14], we regulate the frequency of concatenation between mean during the initial epochs, and increasing it later on. Finally we pass the high dimensional v , m and viewing direction (θ) to color MLP (Γ_{color}) which can be expressed as:

$$c = \Gamma_{color}(v, m, \theta) \quad (11)$$

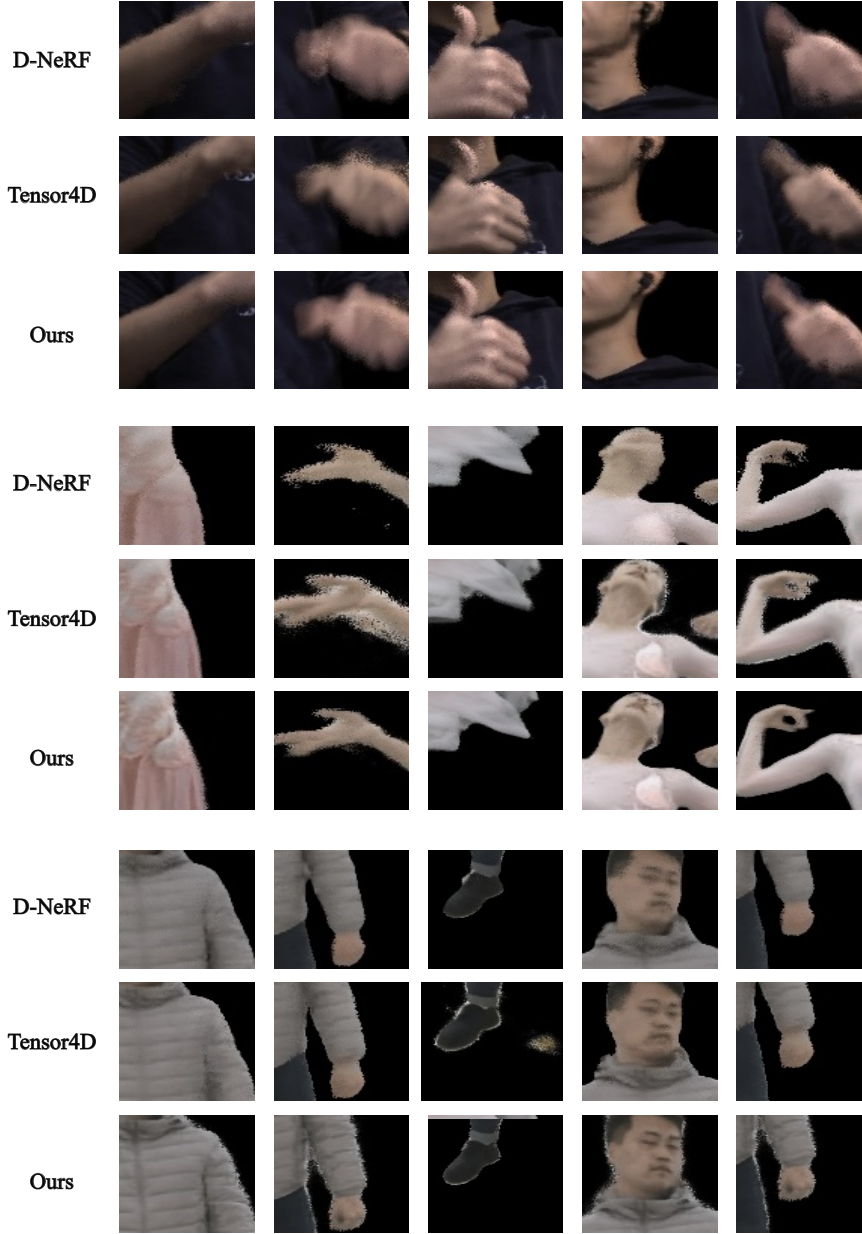


Figure 5. We show how our proposed method outperforms other state-of-the-art methods across the three different datasets. All the significant differences in images shown in this figure are obtained by rendering images from novel poses. The other methods used are D-NeRF[6] and Tensor4D[10].

B. Loss Function

We employ the same loss function for all the dataset. Firstly we use the color loss which is represented mathematically as:

$$L_{color} = ||C_{pred} - C_{GT}|| \quad (12)$$

where C_{pred} and C_{GT} are the predicted and ground truth value of the 3D point.

Subsequently, we apply the surface constraint loss on the planes to get smoother surfaces.

$$L_{grad} = |||g||_2 - 1||_2 \quad (13)$$

where g are the gradients from the Γ_{ref} .

V. EXPERIMENTS

We assess the performance of our proposed method, Trini, on Tensor4D [10] dataset. The Tensor4D dataset

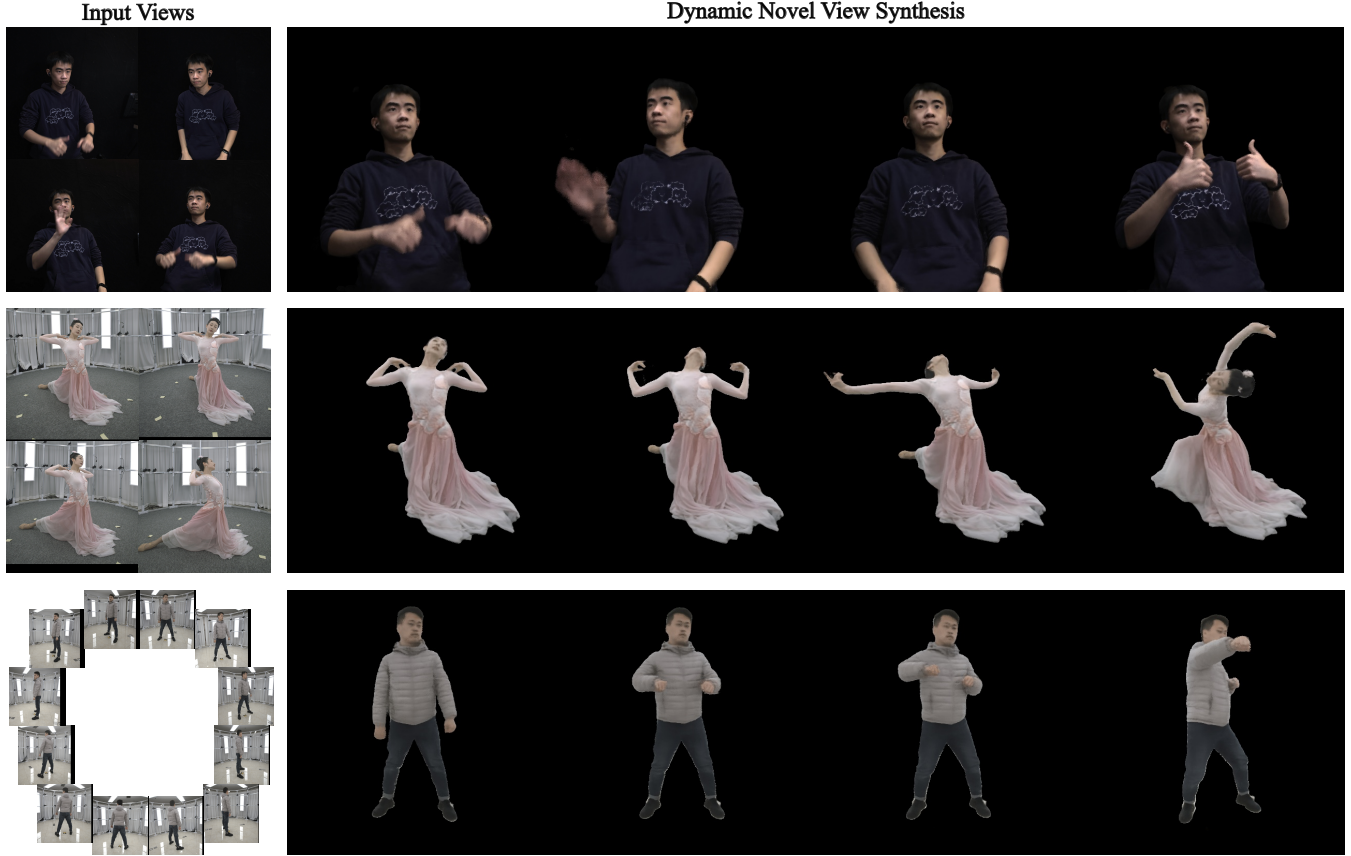


Figure 6. The leftmost columns show the set of the images that are given as input. Also, we show the camera poses that are taken into consideration to reconstruct the dynamic scene. The other columns are the high quality images rendered from the proposed method from the novel poses and times of the dynamic scenes.

Table I
QUANTITATIVE COMPARISON ON THUMBSUP_V4 AND BOXING_V12 DATASET (SPARSE-CAMERA VIEW SETTINGS). **BEST** AND **SECOND** RESULTS ARE IN THE HIGHLIGHTS.

Model	Dance				Boxing			
	PSNR \uparrow	SSIM \uparrow	MSE \downarrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	MSE \downarrow	LPIPS \downarrow
D-NeRF[6]	24.47	0.846	0.0058	0.092	22.96	0.765	0.0068	0.127
TiNeuVox[17]	22.33	0.780	0.0087	0.136	21.29	0.744	0.0083	0.145
NeuS-T[18]	23.54	0.816	0.0083	0.132	21.11	0.738	0.0055	0.169
Tensor4D[10]	28.12	0.922	0.0026	0.049	26.28	0.937	0.0027	0.033
Ours	31.53	0.969	0.0011	0.031	28.86	0.951	0.0008	0.019

[10] comprises data captured from a sparse-view imaging system. In our study, we specifically utilize a subset of this dataset which includes four views, providing both a front-facing perspective and a comprehensive 360° view from 12 evenly spaced cameras and all cameras in the dataset are synchronized and calibrated. The "thumbsup_v4" and "dance_v4" datasets consists of 120 and 400 images respectively with front view facing from four cameras. While "boxing_v12" has 360 images from 12 cameras providing a 360° view.

For a fair comparison between our proposed method and other state-of-the-art approaches, all training and novel view

rendering were conducted on a single Nvidia Titan V GPU with 12 gigabytes of memory. All reported results and training times are specific to this GPU configuration. We utilized the PyTorch library [19] for the implementation.

A. Dynamic Novel View Synthesis Results

We present a comparative analysis of our work with existing state-of-the-art methods, including D-NeRF [6], TiNeuVox [17], Tensor4D [10], and NeuS-T, an extension of NeuS [18] that incorporates an additional time variable.

In Figure 6, we demonstrate Trini's performance in sparse-view camera settings. It excels in reconstructing dynamic

Table II
QUANTITATIVE COMPARISON ON THUMBSUP_v4 DATASET. THE SYNTHESIS QUALITY ACROSS DIFFERENT METRICS, TRAINING TIME, NUMBER OF ITERATIONS AND NUMBER OF PARAMETERS USED ARE REPORTED. **BEST** AND **SECOND** RESULTS ARE IN THE HIGHLIGHTS.

Model	PSNR \uparrow	SSIM \uparrow	MSE \downarrow	LPIPS \downarrow	Training Time	#iterations	#parameters
D-NeRF[6]	25.11	0.878	0.0034	0.174	28h	500K	4.8M
TiNeuVox[17]	22.64	0.832	0.0054	0.195	27mins	10K	102M
NeuS-T[18]	24.71	0.859	0.0045	0.172	31h	500K	5M
Tensor4D[10]	27.94	0.897	0.0019	0.038	34mins	10K	43M
Ours	29.72	0.927	0.0021	0.029	73mins	24K	23M

Table III
QUANTITATIVE COMPARISON ON THUMBSUP_v4 DATASET ONLY ON THE **BORDERS** OF THE MASKED PART OF THE IMAGE. THE SYNTHESIS QUALITY PSNR, TRAINING TIME, NUMBER OF ITERATIONS AND NUMBER OF PARAMETERS USED ARE REPORTED.

Model	PSNR	Time	#Iter	#Params
D-NeRF[6]	23.89	111h	3000K	4.8M
TiNeuVox[17]	21.66	10h	200K	102M
NeuS-T[18]	22.93	122h	3000K	5M
Tensor4D[10]	26.75	11h	200K	43M
Ours	28.19	12h	240K	23M

scenes from novel viewpoints and times, even in complex scenarios, while maintaining the desired high quality.

In Table I, we present a quantitative comparison for "dance_v4" and "boxing_v12" dataset. The tables highlight the best and second-best results. We have employed evaluation metrics including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean-Square Error (MSE), and the perceptual quality measure (LPIPS) [20]. The results unequivocally demonstrate Trini's superiority over other state-of-the-art methods.

In Table II, we provide a comprehensive quantitative comparison, considering various metrics alongside training time, number of iterations, and method size for the "thumbsup_v4" dataset. The best and second-best results have been highlighted and Trini outperforms other state-of-the-art methods across most of the evaluation metrics.

In Table III, we extend this comparison to the "dance_v4" dataset, focusing specifically on the borders of the masked image section. It is evidently clear that Trini consistently achieves sharper border details compared to other methods, all while maintaining superior image quality.

In Figure 5, we pictorially differentiate how Trini renders high quality images from novel viewpoints and time, compared to the other state-of-art methods.

In Figure 2, we demonstrate how the implementation of a straightforward roving technique, which involves a compact MLP and a few parameters tailored to the specific images in the dataset, effectively leads to a noticeable improvement in the quality of rendered novel images. This experiment was conducted on other scenes in the dataset, and consistent results were achieved each time.

Limitations. This work is dependent on bounding boxes in order to differentiate between the background and fore-

ground.

VI. CONCLUSION

We have presented Trini, a novel approach for representing a dynamic 3D scene in a total of 12 streamlined planes bifurcated using three volumes. Additionally, we propose an innovative roving technique that efficiently extracts features from the image set in a sparse-view camera setting, resulting in sharper images with significantly reduced artifacts in rendered images from novel views. Our method demonstrates comparable or superior results across different evaluation metrics, with a substantial to on par reduction in training time compared to other state-of-the-art methods. We anticipate that Trini will make a significant contribution towards expediting and improving the quality of rendering in the field of 3D research.

REFERENCES

- [1] Mildenhall, Ben and Srinivasan, Pratul P and Tancik, Matthew and Barron, Jonathan T and Ramamoorthi, Ravi and Ng, Ren, *Nerf: Representing scenes as neural radiance fields for view synthesis*. Communications of the ACM, 65, 99–106, 2021. 1, 3, 4
- [2] Gao, Chen and Saraf, Ayush and Kopf, Johannes and Huang, Jia-Bin. *Dynamic view synthesis from dynamic monocular video*. Proceedings of the IEEE/CVF International Conference on Computer Vision. 5712–5721, 2021 1
- [3] Yu, Alex and Ye, Vickie and Tancik, Matthew and Kanazawa, Angjoo. *pixelnerf: Neural radiance fields from one or few images*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4578–4587, 2021. 1, 3
- [4] Verbin, Dor and Hedman, Peter and Mildenhall, Ben and Zickler, Todd and Barron, Jonathan T and Srinivasan, Pratul P. *Ref-nerf: Structured view-dependent appearance for neural radiance fields*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5481–5490, 2022. 1, 3
- [5] Reiser, Christian and Peng, Songyou and Liao, Yiyi and Geiger, Andreas. *Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps*. Proceedings of the IEEE/CVF International Conference on Computer Vision, 14335–14345, 2021 1
- [6] Pumarola, Albert and Corona, Enric and Pons-Moll, Gerard and Moreno-Noguer, Francesc. *D-nerf: Neural radiance fields for dynamic scenes*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10318–10327, 2021. 1, 3, 6, 7, 8

- [7] Park, Keunhong and Sinha, Utkarsh and Barron, Jonathan T and Bouaziz, Sofien and Goldman, Dan B and Seitz, Steven M and Martin-Brualla, Ricardo. *Nerfies: Deformable neural radiance fields*. Proceedings of the IEEE/CVF International Conference on Computer Vision. 5865–5874, 2021. 1
- [8] Du, Yilun and Zhang, Yinan and Yu, Hong-Xing and Tenenbaum, Joshua B and Wu, Jiajun. *Neural radiance flow for 4d view synthesis and video processing*. IEEE/CVF International Conference on Computer Vision (ICCV). 14304–14314, 2021. 1, 3
- [9] Chan, Eric R and Lin, Connor Z and Chan, Matthew A and Nagano, Koki and Pan, Boxiao and De Mello, Shalini and Gallo, Orazio and Guibas, Leonidas J and Tremblay, Jonathan and Khamis, Sameh and others. *Efficient geometry-aware 3D generative adversarial networks*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 16123–16133, 2022. 2, 3, 4
- [10] Shao, Ruizhi and Zheng, Zerong and Tu, Hanzhang and Liu, Boning and Zhang, Hongwen and Liu, Yebin. *Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 16632–16642, 2023. 1, 2, 4, 5, 6, 7, 8
- [11] Chen, Anpei and Xu, Zexiang and Geiger, Andreas and Yu, Jingyi and Su, Hao. *Tensorf: Tensorial radiance fields*. European Conference on Computer Vision. Springer. 333–350. 2022 1, 3
- [12] Sun, Cheng and Sun, Min and Chen, Hwann-Tzong. *Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5459–5469, 2022. 1
- [13] Yu, Alex and Li, Ruilong and Tancik, Matthew and Li, Hao and Ng, Ren and Kanazawa, Angjoo. *Plenotrees for real-time rendering of neural radiance fields*. Proceedings of the IEEE/CVF International Conference on Computer Vision. 5752–5761, 2021. 1
- [14] Yang, Jiawei and Pavone, Marco and Wang, Yue. *FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8254–8263, 2023. 2, 3, 5
- [15] Cao, Ang and Johnson, Justin. *Hexplane: A fast representation for dynamic scenes*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 130–141, 2023. 1, 4
- [16] Jonathan T. Barron and Ben Mildenhall and Matthew Tancik and Peter Hedman and Ricardo Martin-Brualla and Pratul P. Srinivasan. *Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields*. ICCV, 2021 3, 5
- [17] Fang, Jiemin and Yi, Taoran and Wang, Xinggang and Xie, Lingxi and Zhang, Xiaopeng and Liu, Wenyu and Nießner, Matthias and Tian, Qi. *Fast dynamic radiance fields with time-aware neural voxels*. SIGGRAPH Asia 2022 Conference Papers. 1–9, 2022. 1, 3, 7, 8
- [18] Wang, Peng and Liu, Lingjie and Liu, Yuan and Theobalt, Christian and Komura, Taku and Wang, Wenping. *Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction*. arXiv preprint arXiv:2106.10689. 2021. 7, 8
- [19] Paszke, Adam and Gross, Sam and Massa, Francisco and Lerer, Adam and Bradbury, James and Chanan, Gregory and Killeen, Trevor and Lin, Zeming and Gimelshein, Natalia and Antiga, Luca and others. *Pytorch: An imperative style, high-performance deep learning library*. Advances in neural information processing systems. 2019. 7
- [20] Zhang, Richard and Isola, Phillip and Efros, Alexei A and Shechtman, Eli and Wang, Oliver. *The unreasonable effectiveness of deep features as a perceptual metric*. Proceedings of the IEEE conference on computer vision and pattern recognition. 586–595, 2018. 8
- [21] Niemeyer, Michael and Barron, Jonathan T and Mildenhall, Ben and Sajjadi, Mehdi SM and Geiger, Andreas and Radwan, Noha. *Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5480–5490, 2022. 3
- [22] Fan, Haoqiang and Su, Hao and Guibas, Leonidas J. *A point set generation network for 3d object reconstruction from a single image*. Proceedings of the IEEE conference on computer vision and pattern recognition. 605–613, 2017. 2
- [23] Wang, Peng-Shuai and Liu, Yang and Guo, Yu-Xiao and Sun, Chun-Yu and Tong, Xin. *O-cnn: Octree-based convolutional neural networks for 3d shape analysis*. ACM Transactions On Graphics (TOG). 36, 4, 1–11, 2017. 2
- [24] Yan, Xinchun and Yang, Jimei and Yumer, Ersin and Guo, Yijie and Lee, Honglak. *Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision*. Advances in neural information processing systems. 29, 2016. 2
- [25] Girdhar, Rohit and Fouhey, David F and Rodriguez, Mikel and Gupta, Abhinav. *Learning a predictable and generative vector representation for objects*. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14. 484–499, 2016. 2
- [26] Mescheder, Lars and Oechsle, Michael and Niemeyer, Michael and Nowozin, Sebastian and Geiger, Andreas. *Occupancy networks: Learning 3d reconstruction in function space*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 4460–4470, 2019. 2, 3
- [27] Niemeyer, Michael and Mescheder, Lars and Oechsle, Michael and Geiger, Andreas. *Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 3504–3515, 2020. 2
- [28] Sitzmann, Vincent and Thies, Justus and Heide, Felix and Nießner, Matthias and Wetzstein, Gordon and Zollhofer, Michael. *Deepvoxels: Learning persistent 3d feature embeddings*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2437–2446, 2019. 2

- [29] Sitzmann, Vincent and Zollhöfer, Michael and Wetzstein, Gordon. *Scene representation networks: Continuous 3d-structure-aware neural scene representations*. Advances in Neural Information Processing Systems. 32, 2019. 2, 3
- [30] Sun, Jiaming and Chen, Xi and Wang, Qianqian and Li, Zhengqi and Averbuch-Elor, Hadar and Zhou, Xiaowei and Snavely, Noah. *Neural 3d reconstruction in the wild*. ACM SIGGRAPH 2022 Conference Proceedings. 1–9, 2022. 2
- [31] Barron, Jonathan T and Mildenhall, Ben and Verbin, Dor and Srinivasan, Pratul P and Hedman, Peter. *Mip-nerf 360: Unbounded anti-aliased neural radiance fields*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5470–5479, 2022. 3
- [32] Mildenhall, Ben and Hedman, Peter and Martin-Brualla, Ricardo and Srinivasan, Pratul P and Barron, Jonathan T. *Nerf in the dark: High dynamic range view synthesis from noisy raw images*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 16190–16199, 2022. 3
- [33] Tancik, Matthew and Casser, Vincent and Yan, Xichen and Pradhan, Sabeek and Mildenhall, Ben and Srinivasan, Pratul P and Barron, Jonathan T and Kretschmar, Henrik. *Block-nerf: Scalable large scene neural view synthesis*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8248–8258, 2022. 3
- [34] Yen-Chen, Lin and Florence, Pete and Barron, Jonathan T and Lin, Tsung-Yi and Rodriguez, Alberto and Isola, Phillip. *Nerf-supervision: Learning dense object descriptors from neural radiance fields*. International Conference on Robotics and Automation (ICRA), 6496–6503, 2022. 3
- [35] Chen, Anpei and Xu, Zexiang and Zhao, Fuqiang and Zhang, Xiaoshuai and Xiang, Fanbo and Yu, Jingyi and Su, Hao. *Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo*. Proceedings of the IEEE/CVF International Conference on Computer Vision. 14124–14133, 2021. 3
- [36] Zhou, Zhizhuo and Tulsiani, Shubham. *Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 12588–12597, 2023. 3
- [37] Park, Jeong Joon and Florence, Peter and Straub, Julian and Newcombe, Richard and Lovegrove, Steven. *DeepSDF: Learning continuous signed distance functions for shape representation*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 165–174, 2019. 2, 3
- [38] Chen, Zhiqin and Zhang, Hao. *Learning implicit fields for generative shape modeling*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5939–5948, 2019. 2, 3
- [39] Chibane, Julian and Alldieck, Thiemo and Pons-Moll, Gerard. *Implicit functions in feature space for 3d shape reconstruction and completion*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 6970–6981, 2020. 2
- [40] Xu, Qiangeng and Wang, Weiyue and Ceylan, Duygu and Mech, Radomir and Neumann, Ulrich. *Disn: Deep implicit surface network for high-quality single-view 3d reconstruction*. Advances in neural information processing systems. 32, 2019. 2
- [41] Liu, Lingjie and Gu, Jiatao and Zaw Lin, Kyaw and Chua, Tat-Seng and Theobalt, Christian. *Neural sparse voxel fields*. Advances in Neural Information Processing Systems. 33, 15651–15663, 2020. 3
- [42] Chan, Eric R and Monteiro, Marco and Kellnhofer, Petr and Wu, Jiajun and Wetzstein, Gordon. *pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis*. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 5799–5809, 2021. 3
- [43] Gu, Jiatao and Liu, Lingjie and Wang, Peng and Theobalt, Christian. *Stylenet: A style-based 3d-aware generator for high-resolution image synthesis*. arXiv preprint arXiv:2110.08985, 2021. 3
- [44] Jang, Wonbong and Agapito, Lourdes. *Codenerf: Disentangled neural radiance fields for object categories*. Proceedings of the IEEE/CVF International Conference on Computer Vision. 12949–12958, 2021. 3
- [45] Schwarz, Katja and Liao, Yiyi and Niemeyer, Michael and Geiger, Andreas. *Graf: Generative radiance fields for 3d-aware image synthesis*. Advances in Neural Information Processing Systems. 33, 20154–20166, 2020. 3
- [46] Sun, Jingxiang and Wang, Xuan and Zhang, Yong and Li, Xiaoyu and Zhang, Qi and Liu, Yebin and Wang, Jue. *Fen-erf: Face editing in neural radiance fields*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7672–7682, 2022. 3
- [47] Shao, Ruizhi and Zhang, Hongwen and Zhang, He and Chen, Mingjia and Cao, Yan-Pei and Yu, Tao and Liu, Yebin. *Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 15872–15882, 2022. 3
- [48] Sun, Guoxing and Chen, Xin and Chen, Yizhang and Pang, Anqi and Lin, Pei and Jiang, Yuheng and Xu, Lan and Yu, Jingyi and Wang, Jingya. *Neural free-viewpoint performance rendering under complex human-object interactions*. Proceedings of the 29th ACM International Conference on Multimedia. 4651–4660, 2021. 3
- [49] Wang, Liao and Wang, Ziyu and Lin, Pei and Jiang, Yuheng and Suo, Xin and Wu, Minye and Xu, Lan and Yu, Jingyi. *ibutter: Neural interactive bullet time generator for human free-viewpoint rendering*. Proceedings of the 29th ACM International Conference on Multimedia. 4641–4650, 2021. 3
- [50] Li, Ruilong and Tanke, Julian and Vo, Minh and Zollhöfer, Michael and Gall, Jürgen and Kanazawa, Angjoo and Lassner, Christoph. *Tava: Template-free animatable volumetric actors*. European Conference on Computer Vision. 419–436, 2022. 3
- [51] Ost, Julian and Mannan, Fahim and Thuerey, Nils and Knodt, Julian and Heide, Felix. *Neural scene graphs for dynamic scenes*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2856–2865, 2021. 3

- [52] Peng, Sida and Zhang, Yuanqing and Xu, Yinghao and Wang, Qianqian and Shuai, Qing and Bao, Hujun and Zhou, Xiaowei. *Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 9054–9063, 2021. 3
- [53] Gan, Wanshui and Xu, Hongbin and Huang, Yi and Chen, Shifeng and Yokoya, Naoto. *V4d: Voxel for 4d novel view synthesis*. IEEE Transactions on Visualization and Computer Graphics. 2023. 3
- [54] Guo, Xiang and Chen, Guanying and Dai, Yuchao and Ye, Xiaoqing and Sun, Jiadai and Tan, Xiao and Ding, Errui. *Neural deformable voxel grid for fast optimization of dynamic view synthesis*. Proceedings of the Asian Conference on Computer Vision. 3757–3775, 2022. 3
- [55] Liu, Jia-Wei and Cao, Yan-Pei and Mao, Weijia and Zhang, Wenqiao and Zhang, David Junhao and Keppo, Jussi and Shan, Ying and Qie, Xiaohu and Shou, Mike Zheng. *Devrf: Fast deformable voxel radiance fields for dynamic scenes*. Advances in Neural Information Processing Systems. 35, 36762–36775, 2022. 3
- [56] Fridovich-Keil, Sara and Yu, Alex and Tancik, Matthew and Chen, Qinlong and Recht, Benjamin and Kanazawa, Angjoo. *Plenoxels: Radiance fields without neural networks*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5501–5510, 2022. 3
- [57] Xian, Wenqi and Huang, Jia-Bin and Kopf, Johannes and Kim, Changil. *Space-time neural irradiance fields for free-viewpoint video*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9421–9431, 2021. 3
- [58] Wang, Chaoyang and Eckart, Ben and Lucey, Simon and Gallo, Orazio. *Neural trajectory fields for dynamic novel view synthesis*. arXiv preprint arXiv:2105.05994, 2021. 3
- [59] Johari, Mohammad Mahdi and Lepoittevin, Yann and Fleuret, François. *Geonerf: Generalizing nerf with geometry priors*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 18365–18375, 2022. 3
- [60] Ma, Li and Li, Xiaoyu and Liao, Jing and Zhang, Qi and Wang, Xuan and Wang, Jue and Pedro V. Sander. *Deblur-NeRF: Neural Radiance Fields from Blurry Images*. arXiv preprint arXiv:2111.14292, 2021. 3
- [61] Seo, Seunghyeon and Han, Donghoon and Chang, Yeonjin and Kwak, Nojun. *MixNeRF: Modeling a Ray With Mixture Density for Novel View Synthesis From Sparse Inputs*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 20659–20668, 2023. 3
- [62] Isaac-Medina, Brian KS and Willcocks, Chris G and Breckon, Toby P. *Exact-NeRF: An Exploration of a Precise Volumetric Parameterization for Neural Radiance Fields*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 66–75, 2023. 3
- [63] Li, Zhengqi and Wang, Qianqian and Cole, Forrester and Tucker, Richard and Snavely, Noah. *DynIBaR: Neural Dynamic Image-Based Rendering*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. 3
- [64] Liu, Yu-Lun and Gao, Chen and Meuleman, Andreas and Tseng, Hung-Yu and Saraf, Ayush and Kim, Changil and Chuang, Yung-Yu and Kopf, Johannes and Huang, Jia-Bin. *Robust Dynamic Radiance Fields*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 3
- [65] Goodfellow, Ian and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua. *Generative adversarial nets*. Advances in neural information processing systems. 27, 2014. 3