RefiNeRF: Modelling dynamic neural radiance fields with inconsistent or missing camera parameters

Shuja Khalid
University of Toronto
27 King's College Cir, Toronto, ON M5S
skhalid@cs.toronto.edu

Frank Rudzicz
University of Toronto
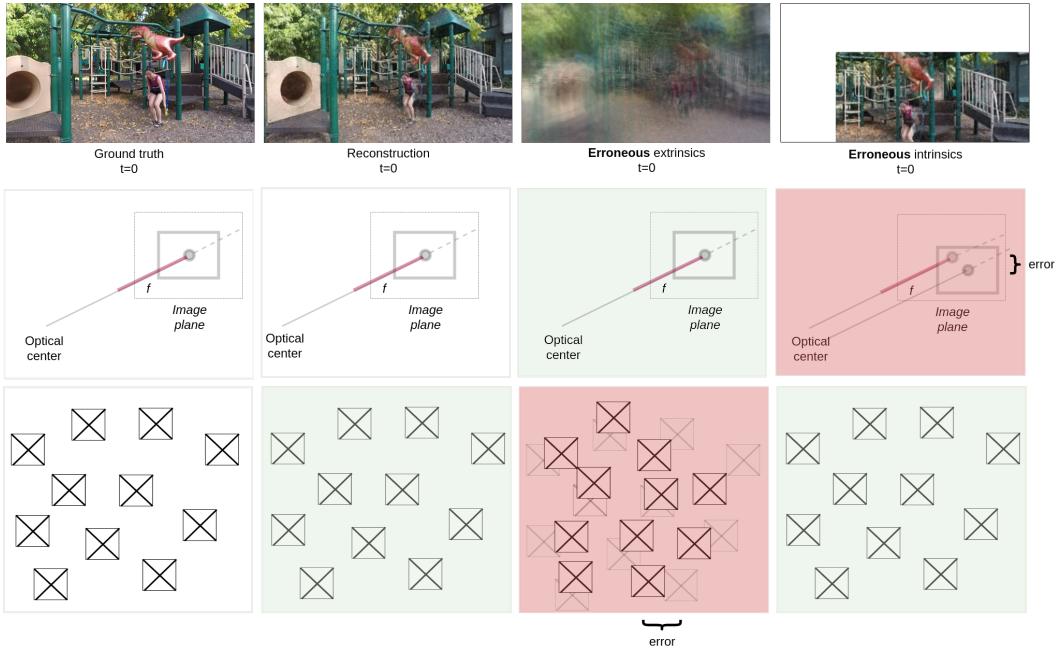27 King's College Cir, Toronto, ON M5S
frank@cs.toronto.edu

Figure 1: Our proposed approach learns camera parameters using a simple photometric loss using a learning scheduler and is easy to incorporate in both static and dynamic frameworks.

Abstract

Novel view synthesis (NVS) is a challenging task in computer vision that involves synthesizing new views of a scene from a limited set of input images. Neural Radiance Fields (NeRF) have emerged as a powerful approach to address this problem, but they require accurate knowledge of camera intrinsic and extrinsic parameters. Traditionally, structure-from-motion (SfM) and multi-view stereo (MVS) approaches have been used to extract camera parameters, but these methods can be unreliable and may fail in certain cases. In this paper, we propose a novel technique that leverages unposed images from dynamic datasets, such as the NVIDIA dynamic scenes dataset, to learn camera parameters directly from data. Our approach is highly exten-sible and can be integrated into existing NeRF architectures with minimal modifications. We demonstrate the effectiveness of our method on a variety of static and dynamic scenes and show that it outperforms traditional SfM and MVS approaches. The code for our method is publicly available at https://github.com/redacted/refinerf. Our approach offers a promising new direction for improving the accuracy and robustness of NVS using NeRF, and we anticipate that it will be a valuable tool for a wide range of applications in computer vision and graphics.

1. Introduction

The classical neural radiance field design methodology treats the presence of camera parameters as an afterthought. The time taken for generating six-degree-of-freedom pose information isn't included in most publications [23, 22, 19, 25] and there is a lack of data, discussing the effect of these parameters on downstream metrics. In this paper, we study the effect of non-existent or erroneous camera parameters and present a modular framework to help address these issues, with minimal computational overhead. Our results show that our approach leads to improved novel-view synthesis metrics compared to state-of-the-art approaches [20, 36, 29].

Camera parameters Camera parameters are the intrinsic and extrinsic properties that define how a camera captures images in a scene. The intrinsic parameters describe the internal characteristics of the camera, such as its focal length, image sensor size, and distortion coefficients. The extrinsic parameters describe the camera's position and orientation in the world, relative to the scene being captured. For the purposes of this paper, we are interested in learning the following camera parameters:

Focal length: the distance between the lens and the image sensor when the lens is focused at infinity. *Image sensor size*: the dimensions of the image sensor that captures the image. *Principal point*: the point where the optical axis intersects the image plane. *Lens distortion*: the amount of distortion that the lens introduces into the image. *Translation*: the position of the camera in 3D space relative to the scene being captured. *Rotation*: the orientation of the camera in 3D space relative to the scene being captured.

Together, these camera parameters define the camera model that can be used to relate 3D points in the world to their corresponding 2D image points in the camera's image plane. This relationship is fundamental to many computer vision tasks such as object recognition, tracking, and 3D reconstruction.

The camera intrinsic parameter matrix, K, can be computed as:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where f_x and f_y are the focal lengths of the camera in the x and y directions, respectively, c_x and c_y are the coordinates of the camera's principal point in the image plane, and s is the skew coefficient.

The intrinsic parameters are defined as a matrix with values f_x, f_y, c_x, c_y, and s. The focal lengths f_x and f_y represent the distance between the camera's lens and the image plane, while the principal point coordinates c_x and c_y represent the intersection of the optical axis with the image plane.

The skew coefficient s accounts for non-orthogonality between the axes of the image plane.

The camera extrinsic parameter matrix, P, can be computed as:

$$P = \begin{bmatrix} R_x(\theta_x)R_y(\theta_y)R_z(\theta_z) & \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2}$$

COLMAP COLMAP [29] is by far the most commonly used approach for predicting camera intrinsics and 6-degree-of-freedom pose information, but it is imperfect. It uses a collection of images to generate high-quality 3D representations and the input images, camera parameters and 3D points in a scene (also known as a 'bundle') are optimized using key-points extracted from the image using non-linear least squares optimization. Since COLMAP is such a crucial component of novel-view synthesis, we consider its failure mechanisms:

- Lack of texture and distinct features: If the images in the input collection contain insufficient texture or distinct features, it can be difficult for COLMAP to accurately reconstruct the 3D structure of the scene.

- Overlapping images: If the images in the input collection overlap too much, COLMAP may struggle to disambiguate the different structures in the scene and reconstruct them accurately.

- Image quality: Poor image quality, such as low resolution, low contrast, or large amounts of noise, can make it difficult for COLMAP to detect features and match them across images.

- Image orientation: If the images in the input collection are not well-oriented, with large amounts of camera rotation or camera tilt, COLMAP may have trouble reconstructing a consistent 3D model.

- Initialization: The accuracy of the reconstruction depends heavily on the initial guess for the camera poses and 3D points, and if this guess is not close enough to the true values, COLMAP may converge to a suboptimal solution or fail to converge at all.

As we move towards real-world applications and away from idealized static setups, correctly determining these parameters is paramount and poses many questions. Can we effectively extract camera pose information from monocular point sources? What should be done if COLMAP fails to find suitable key-points from the image, and can't register images? This paper attempts to answer these questions by building on some landmark papers by Wang *et al.* [36] and Lin *et al.* [20]. Our contributions are:
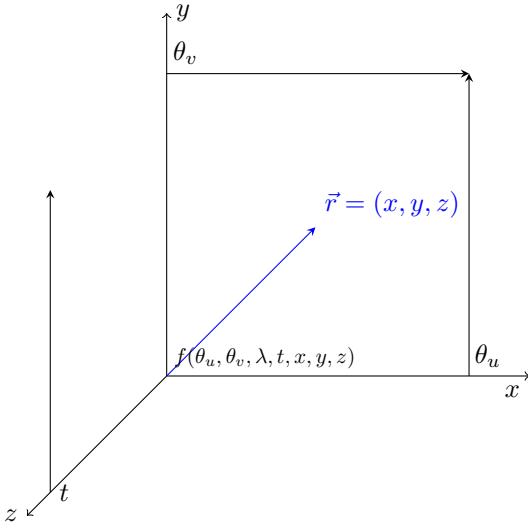
Figure 2: A 7D general representation of the Plenoptic function featuring a ray of light in 3D space extending towards an object

- We provide a refining technique that converges to optimal camera parameters in cases where the COLMAP predictions are erroneous

- We outline a scheduling technique and initializations that provide estimates of camera parameters in cases where COLMAP fails completely

- We compare and contrast the effectiveness of these initializations against state-of-the-art bundle adjustment techniques such as NeRF– [36], BARF [20], COLMAP [29]

- We conduct extensive ablation studies to study the effect of noise on both intrinsic and extrinsic camera parameters

Plenoptic function A plenoptic function is a mathematical representation used in computational photography to describe the light field, which is the amount of light traveling in every direction in a given scene. It provides information about the light rays in a scene, including their direction, position, and intensity, and can be used to generate 2D images or perform post-capture adjustments such as refocusing and perspective correction. We illustrate this function in Figure 2. For the purposes of this paper, we model a 6D plenoptic function $(\theta_u, \theta_v, t, x, y, z)$.

2. Related Works

We cover the extent of existing Neural Radiance Field (NeRF) modelling techniques and distinguish between the approaches used for capturing camera parameters.

3D visual scene representation is the desired form of visual scene understanding and is more representative than its 2D counterparts. Significant progress has been made in recent years to model complex geometries using methods such as point-clouds [11, 3, 39, 32], voxels [30, 47, 40, 16], octrees [38, 45], or various computed tomography algorithms [6]. These computationally expensive techniques have served as a bottleneck for true 3D understanding and most tasks require strong priors [7, 28, 14] or existing templates [25, 10, 42, 41, 37]. In NeRFs [23] implicit functions are used for representing scenes by implicitly encoding photometric attributes such as colour, surface illumination, opacity, etc., using shallow neural nets [21, 1, 26, 22]. Since its advent, there has been an explosion in self-supervised learning of scenes and their constituents. A simple pixel-wise photometric reconstruction loss is leveraged to train the models in an end-to-end manner. The models can be broadly categorized as follows:

Static scenes Some methods in the implicit representation paradigm have generated highly detailed scenes using limited images [26, 25, 19, 8, 15]. The impressive results presented in the aforementioned papers demonstrate the potential of well-designed representations, but do not directly apply to in-the-wild scenes. The reason is that the input images, while sparse, are well-posed, capture a $360°$ panoramic view of the scene, and come with pre-computed pose information. In contrast, in-the-wild scenes are typically captured from a monocular source without pose information. Pose information is usually inferred using off-the-shelf models that use *structure-from-motion*, which can be non-deterministic and prone to error [29].

Dynamic scenes Some approaches extend the implicit representation paradigm to include time τ, particularly in-the-wild scenes that are under-constrained and require off-the-shelf models. Flow-based methods [19, 8, 15] use additional inputs such as depth estimation [18], optical flow [2, 33], and semantic segmentation [9] to constrain the scene. Deformation-based approaches [27, 34] have also been used to model dynamic scenes without relying on pre-trained models or pre-designed priors. However, these approaches do not generalize well to in-the-wild scenes. Our approach combines flow-based and deformation-based methods to set the state-of-the-art on the NVIDIA dynamic scenes dataset.

Learning camera parameters Intrinsic and extrinsic parameters of a camera are a pre-requisite of neural radiance fields. These parameters are traditionally extracted from off-the-shelf structure-from-motion software such as COLMAP and requires additional computational overhead. The more frames in a scene, the higher the computational
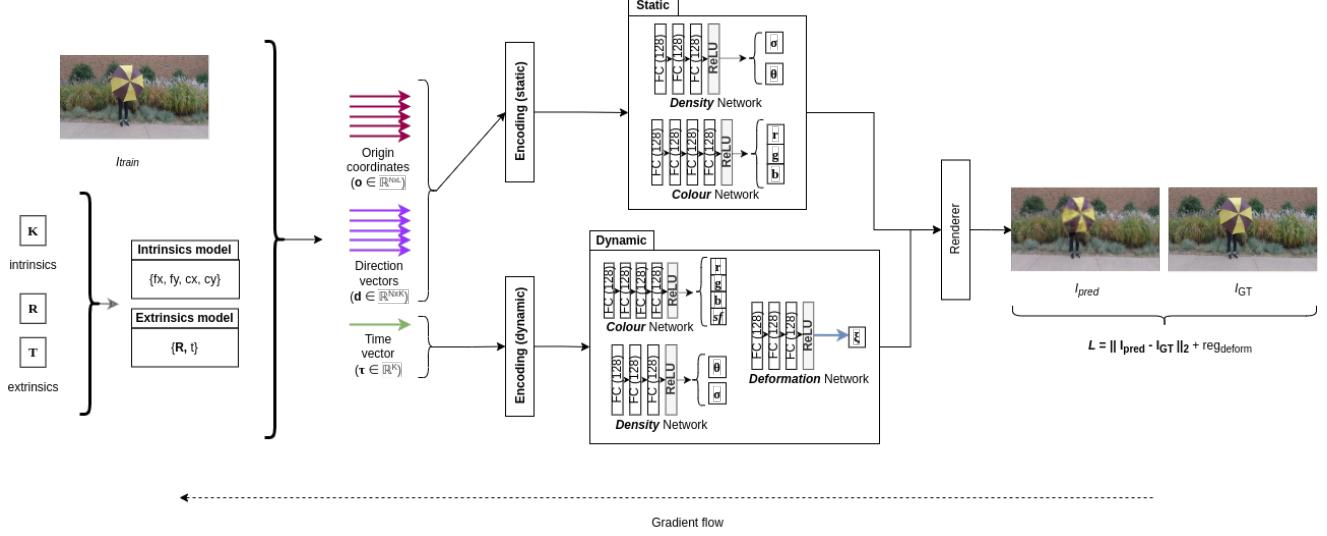
Figure 3: Our proposed end-to-end trainable architecture. We use density and colour networks (*top*) to model a static representation of the scene, and density, colour, and deformation networks (*bottom*) to model the motion-centric pixels in the image. Each set of representations are trained separately and the final image consists of the fused output.

burden. Some work attempted to address these concerns in static scenes such as iNeRF [43], NeRF– [36], and BARF [20]. However, those studies were limited as they only considered static scenes and very favourable learning conditions. We introduce a paradigm for dynamic images by considering the static and dynamic portions of the image separately as inspired by Khalid *et al* [15].

We also leverage multi-resolution encoding, which has significantly improved reconstructions by encoding data as a multi-resolution subset of high-frequency embeddings, as measured by commonly-used reconstruction metrics, Learned Perceptual Image Patch Similarity (LPIPS) [46], structural similarity (SSIM) [5], and peak signal-to-noise ratio (PSNR) [13].
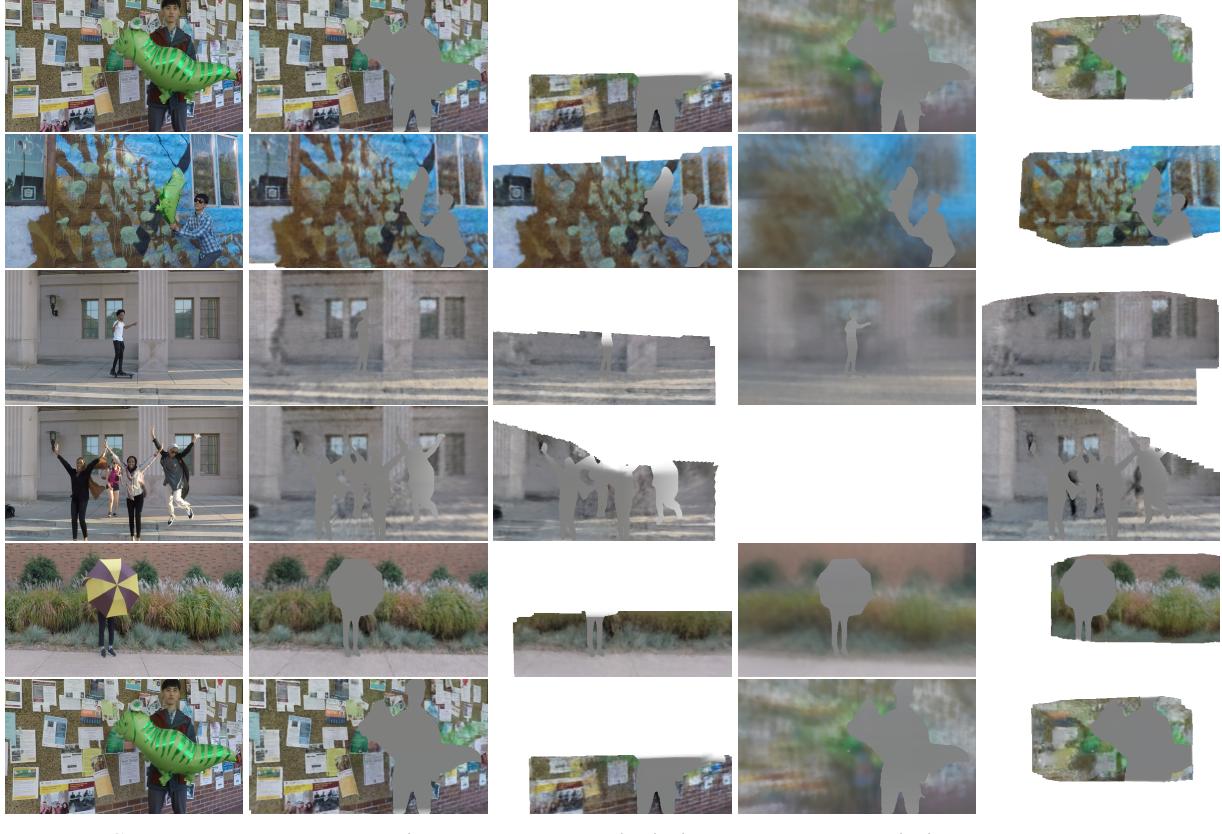
3. Preliminary

3.1. Datasets

NVIDIA dynamic scenes dataset The NVIDIA Dynamic Scenes Dataset [44] is a high-quality dataset to support the development of AI algorithms for 3D understanding problems. The dataset consists of synchronized high-resolution RGB data captured using 24 cameras in a variety of real-world urban and suburban environments with individuals performing a variety of tasks. The dataset includes over 8 scenes, with a total of more than 1000 frames, and it is designed to provide a diverse and challenging set of scenes for researchers. The goal of the dataset is to help advance the state-of-the-art in areas such as 3D scene understanding and novel view or time synthesis.

Cholec80 The Cholec80 dataset [35] is a medical image dataset that consists of 80 video recordings of laparoscopic cholecystectomy surgeries. Laparoscopic cholecystectomy is a minimally invasive surgical procedure performed to remove the gallbladder. The dataset was created to provide a relatively large-scale and high-quality dataset for computer-aided surgical navigation and robot-assisted surgery systems. The videos in the Cholec80 dataset have been captured with high temporal and spatial resolution, and they cover a wide range of surgical scenarios, instrument movements, and physiological variations. The dataset provides a valuable resource for researchers working on developing algorithms for real-time surgical navigation, instrument tracking, and autonomous robotic surgery. We extract short clips from this dataset and evaluate it on the novel scene synthesis problem. The smooth textures of internal tissue in this dataset makes it challenging for COLMAP to produce accurate camera parameter estimates.

Grid encoding Grid encoding is a method of compressing and representing light field data in a computationally efficient manner. It involves dividing the light field into a grid of micro-images and encoding the light rays passing through each micro-image as separate elements in the grid. The encoded data can be stored and processed more efficiently than raw light field data, allowing for faster rendering of images and other operations. Grid encoding is a key component of many light field imaging techniques and is used in various applications, including virtual and augmented reality, 3D imaging, and computational photogra-

| GT | reconstruction | erroneous intrinsics | erroneous extrinsics | erroneous params. |

Figure 4: **Qualitative** results: We show erronoeus predictions generated by COLMAP. Erroneous params refers to both erroneous intrinsics and extrinsics.

phy. The illustrations presented in this paper are generated using grid-encoding representation developed by Muller *et al.* [24].

In neural radiance fields, grid positional and frequency encoding arer often used to encode the spatial location and frequency information of the scene. Grid positional (3) and frequency encoding(4) are defined thus:

$$
\mathbf{f}_{\text{pos}}(\mathbf{p}) = \\
[\sin(\mathbf{W}_1\mathbf{p}), \cos(\mathbf{W}_1\mathbf{p}), \dots, \\
\sin(\mathbf{W}_n\mathbf{p}), \cos(\mathbf{W}_n\mathbf{p})] \quad (3)
$$

$$
\mathbf{f}_{\text{freq}}(\mathbf{p}) = \\
[\sin(2^0\pi\mathbf{W}_1\mathbf{p}), \cos(2^0\pi\mathbf{W}_1\mathbf{p}), \dots, \\
\sin(2^{n-1}\pi\mathbf{W}_n\mathbf{p}), \cos(2^{n-1}\pi\mathbf{W}_n\mathbf{p})] \quad (4)
$$

where $\mathbf{p}$ is the 3D spatial location of a point in the scene, $\mathbf{W}_i$ is a learnable weight matrix of size $3 \times d_i$, where d_i is the number of frequencies along the ith axis, and n is the number of frequency bands.

The grid positional encoding uses a sine-cosine pair for each weight matrix $\mathbf{W}_i$ to encode the spatial location of the point along the corresponding axis. The grid frequency encoding uses the sine-cosine pair for each weight matrix $\mathbf{W}_i$ to encode the frequency information of the point along the corresponding axis, with increasing frequency bands in powers of two.

These encodings are concatenated to form the final encoding vector $\mathbf{f}_{\text{grid}}(\mathbf{p})$, which is then used as input to the neural network to predict the radiance of the point.

The concatenation of the grid positional and frequency encodings is:

$$
\mathbf{f}\text{grid}(\mathbf{p}) = [\mathbf{f}\text{pos}(\mathbf{p}), \mathbf{f}\text{freq}(\mathbf{p})] \quad (5)
$$

3.2. Pose-free Estimation

Pose-free estimation refers to the process of estimating the properties of an object or scene without requiring explicit information about its pose, or position and orienta-

Scene	PSNR					SSIM					LPIPS				
	colmap	NeRF–	BARF	Ours	delta	colmap	NeRF–	BARF	Ours	delta	colmap	NeRF–	BARF	Ours	delta
Balloon1	16.785	14.019	14.019	14.821	0.802	0.584	0.416	0.416	0.441	0.025	0.172	0.330	0.320	0.299	-0.021
Balloon2	19.656	16.369	16.352	16.581	0.229	0.666	0.554	0.554	0.602	0.048	0.161	0.340	0.341	0.305	-0.036
Jumping	18.423	5.443	5.446	13.224	7.778	0.709	0.542	0.548	0.611	0.063	0.168	0.243	0.241	0.218	-0.023
Umbrella	19.171	18.465	18.458	18.673	0.215	0.587	0.566	0.566	0.584	0.018	0.207	0.288	0.290	0.266	-0.024
Skating	22.437	19.782	19.784	20.229	0.445	0.799	0.743	0.743	0.761	0.018	0.094	0.147	0.147	0.132	-0.015
Playground	21.684	15.003	15.001	17.804	2.803	0.806	0.379	0.378	0.556	0.178	0.137	0.342	0.341	0.259	-0.082

Table 1: We compare and contrast our approach to existing methods such as *BARF* and *NeRF–*, the delta values are calculated by subtracting our values to that of BARF, which consistently outperforms NeRF– on this dataset.

Scene	PSNR			SSIM			LPIPS		
	colmap	Ours	delta	colmap	Ours	delta	colmap	Ours	delta
Balloon1	16.203	16.167	-0.036	0.606	0.592	-0.014	0.181	0.189	0.008
Balloon2	19.488	19.826	0.338	0.675	0.678	0.003	0.156	0.156	0.000
Jumping	18.077	18.180	0.103	0.709	0.710	0.001	0.148	0.146	-0.002
Umbrella	19.216	19.232	0.016	0.589	0.587	-0.002	0.168	0.169	0.001
Skating	22.740	22.774	0.034	0.803	0.803	0.000	0.067	0.065	-0.002
Playground	22.770	22.720	-0.050	0.879	0.877	-0.002	0.081	0.082	0.001

Table 2: **Ablation**: Intrinsics

Scene	PSNR			SSIM			LPIPS		
	colmap	Ours	delta	colmap	Ours	delta	colmap	Ours	delta
Balloon1	16.173	16.291	0.118	0.604	0.598	-0.006	0.181	0.184	0.003
Balloon2	19.483	19.867	0.384	0.675	0.677	0.002	0.155	0.155	0.000
Jumping	17.942	18.283	0.341	0.712	0.739	0.027	0.139	0.144	0.005
Umbrella	19.240	19.270	0.030	0.589	0.586	-0.003	0.169	0.174	0.005
Skating	22.730	22.899	0.169	0.803	0.805	0.002	0.067	0.066	-0.001
Playground	22.796	22.588	-0.208	0.879	0.872	-0.007	0.082	0.086	0.004

Table 3: **Ablation**: Extrinsics

tion in space. This is particularly useful in computer vision and computer graphics, where the pose of an object can be difficult to measure or observe. Pose-free estimation algorithms use features such as texture, colour, or shape to determine the properties of an object, rather than relying on explicit information about its pose. These algorithms are used in applications such as object recognition, tracking, and 3D reconstruction. By eliminating the need for pose information, pose-free estimation methods can be more robust, flexible, and computationally efficient than traditional pose-based methods.

3.3. Metrics

We assess our proposed framework from two perspectives. First, to gauge the quality of novel view rendering, we use commonly employed metrics such as Peak Signal-to-Noise Ratio (PSNR) [13], Structural Similarity Index Measure (SSIM) [5], and Learned Perceptual Image Patch Similarity (LPIPS) [46]. Second, we evaluate the precision of the optimized camera parameters, encompassing the focal length, rotation, and translation. Regarding focal length assessment, we report the absolute error in terms of pixels. For camera poses, we adhere to the evaluation protocol of Absolute Trajectory Error (ATE) [31].

3.4. Learning Strategy

Our proposed approach is closely inspired by the methodologies presented in BARF [20] and NeRF– [36]. BARF uses a scheduled approach for including higher frequency positional encoding parameters during image reconstruction. High-frequency positional encodings are essential as they allow for high-fidelity images [4] and have become commonplace in scene rendering tasks [25, 22, 15]. We opt for a slightly different approach in which we learn

how the frequency components should be applied during reconstruction.

In the absence of pose information, or in cases where COLMAP fails to estimate camera parameters, we introduce learnable intrinsic and extrinsics parameters. In doing so, the NeRF model is able to generate realistic renderings. Since our dynamic deformation model, inspired by Khalid *et al.* [15], is designed to train the *static* and *dynamic* models separately, we introduce a scheduled training methodology in Algorithm 1.

Algorithm 1 Camera parameter update

Require: Input data $N_{all}, N_s, N_c, \mathcal{F}_\xi, \mathcal{F}_s, \mathcal{F}_c$
Ensure: Output $\mathcal{F}_\xi$

1: **procedure** PARAMUPDATE(**X**)
2:     Initialize trainable camera params $\mathcal{R}, t, f_x, f_y$
3:     **for** $i = 1$ **to** N_{all} **do**
4:         Forward pass and calculate reconstruction loss
5:         **if** $i \leq N_s$ **then**
6:             Update $\mathcal{F}_s$
7:         **else**
8:             Update $\mathcal{F}_\xi$
9:         **end if**
10:        **if** $i \leq N_c$ **then**
11:            Update cam. param.: $R \leftarrow R + \nabla R$
12:            Update cam. param.: $t \leftarrow t + \nabla t$
13:            Update $\mathcal{F}_c$
14:        **end if**
15:     **end for**
16: **end procedure**

(a) *Scene*: **Balloon1**

(b) *Scene*: **Playground**

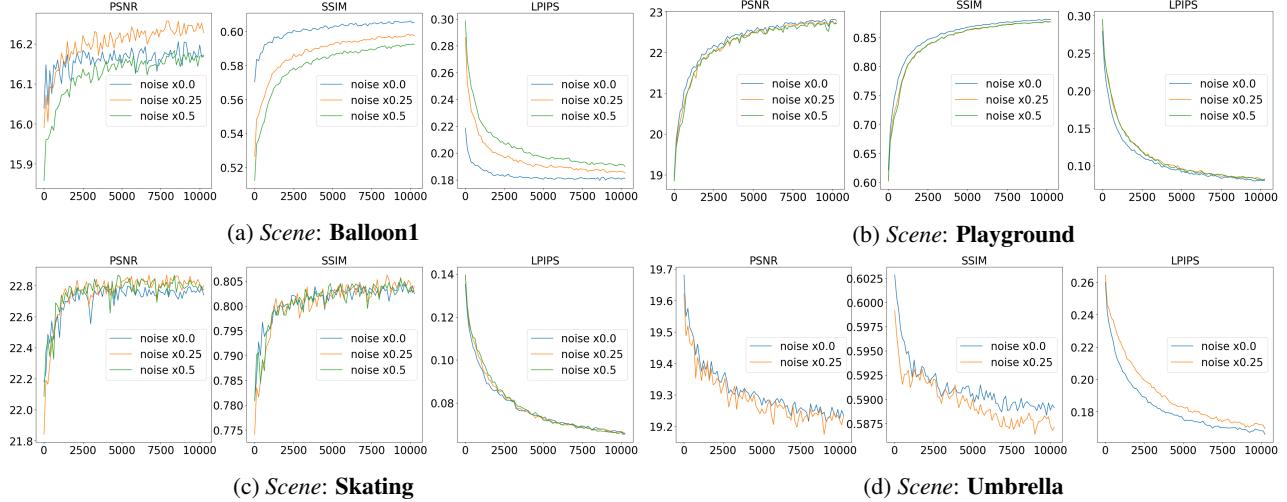(c) *Scene*: **Skating**

(d) *Scene*: **Umbrella**

Figure 5: **Quantitative** results: Training dynamics using increments of perturbations when predicting camera intrinsics. We perturb the camera intrinsics by $\pm 50\%$ in increments of $\pm 25\%$
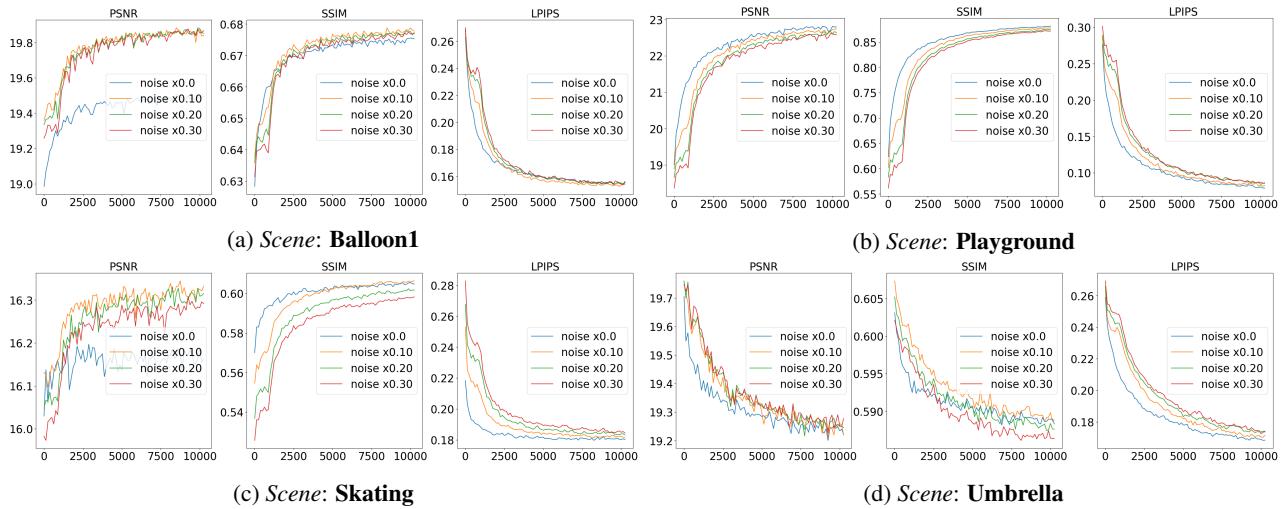

(a) *Scene*: **Balloon1**

(b) *Scene*: **Playground**

(c) *Scene*: **Skating**

(d) *Scene*: **Umbrella**

Figure 6: **Quantitative** results: Training dynamics using increments of perturbations when predicting camera extrinsics. We perturb the camera extrinsics by $\pm 30\%$ in increments of $\pm 10\%$

3.5. Implementation

Our implementation is based on the framework provided by Khalid *et al.* [15], with a few modifications for enhanced computation efficiency. Specifically, we do not include the effect of the dynamic component in our calculations, as the static representation is used in the analysis, ee keep the hidden layer dimension at 256, and we sample only 4096 pixels from each input image and 128 points along each ray. We use Kaiming initialization [12] for the NeRF model and initialize all cameras to the origin, looking in the $-z$ direction, with the focal length (f) set to the image width. To optimize the NeRF, camera poses, and focal lengths, we employ three separate Adam optimizers, all with an initial

learning rate of 0.001. The learning rate of the NeRF model decays every 100 epochs by multiplying it by 0.997 (equivalent to stair-cased exponential decay), while the learning rates of the pose and focal length parameters decay every 10 epochs with a multiplier of 0.9. Unless otherwise specified, all models are trained for $10,000$ epochs. Further technical details are provided in the supplementary material.

4. Experiments

We validate the effectiveness of our approach by predicting poses from scratch and conducting extensive ablation studies for the novel-view synthesis task.
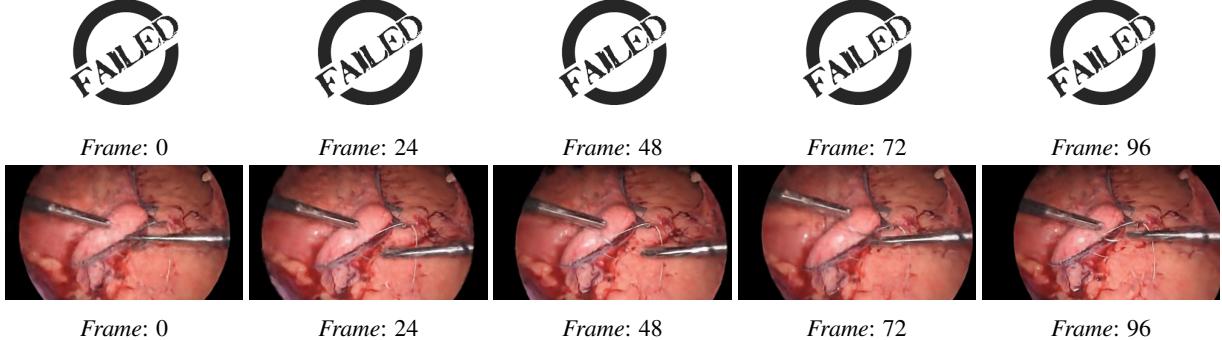
| Frame: 0 | Frame: 24 | Frame: 48 | Frame: 72 | Frame: 96 |

| Frame: 0 | Frame: 24 | Frame: 48 | Frame: 72 | Frame: 96 |

Figure 7: **Qualitative** results: We include results on a clip extracted from the Cholec80 dataset. *Top row* The lack of rich textures makes COLMAP generate predictions that result in failed reconstructions. *Bottom row* Our proposed approach allows for novel view synthesis with a fixed camera pose in extremely challenging environments.

4.1. Pose-free estimation

Quantitative We compare our technique to BARF and NeRF–, each of which attempt to learn camera parameters directly through gradients generated from a photometric loss. We show that our approach, which incorporates a simple initialization and scheduling methodology to generate realistic renderings even in the absence of camera parameters. As illustrated in Table 3.1, our method outperforms the existing state-of-the-art, and approaches in which values are generated by simple COLMAP initialization. This indicates that our approach isn't a replacement for COLMAP-based generalizations but can improve existing predictions or provide adequate estimates in case COLMAP fails for some of the reasons mentioned earlier. Figure 4 shows qualitative results.

4.2. Ablation Study

Camera poses We conduct an ablation study to illustrate the refinement capabilities of our proposed method. We treat the predictions of intrinsics and extrinsics separately. In Tables 2 and 3, we capture the novel view synthesis results of training a model end-to-end and perturbing the ground truth camera pose information. We perturb the rotational parameters by $\pm 30°$ in increments of $\pm 10\%$. The translational components are perturbed by $\pm 30\%$ in increments of $\pm 10\%$. For both intrinsics and extrinsics, we notice that the refinement process can actually improve reconstruction metrics, as illustrated in Figures 5 and 6. The appendix includes training dynamics for all of the scenes in the NVIDIA dynamic scenes dataset.

We repeat the same procedure by perturbing the camera intrinsics by $\pm 50\%$ in increments of $\pm 25\%$. We similarly observe the model's tendency to improve novel-view synthesis metrics if the gradient is allowed to flow through the camera parameters, using the proposed learning scheme in Alg. 1.

Cholec80 We present our results on the Cholec80 dataset to show the generalizability of our proposed approach to extremely challenging real-world environments. The Cholec80 dataset, due to its relatively uniform and texture-less image content, suffers from erroneous COLMAP estimates. This type of forward-facing data captured using a monocular camera, which is typical of various real-world applications, benefits from our proposed approach and produces a fixed-view camera reconstruction of the scene. We present these results in Figure 7. We sample 15 frames/s for this reconstruction and are only able to capture 6 seconds worth of content. We intend to push the limits of this temporal novel-view synthesis in future work.

5. Conclusion

We introduce refiNeRF, a straightforward, modular, and effective technique for training radiance fields for novel-view synthesis when dealing with imperfect camera poses. We assess the importance of refining coarse representations made by COLMAP and present a technique for jointly registering and reconstructing coordinate-based scene representations. Our experiments indicate that refiNeRF can effectively learn 3D scene representations from scratch while correcting significant camera pose misalignment.

Although refiNeRF shows promising results for both static and dynamic scenes, it shares the same limitations as the original NeRF approach, such as slow optimization and rendering, the requirement of dense 3D sampling, and dependence on heuristic coarse-to-fine scheduling strategies. Through the application of recent advancements such as iNGP [24] and vaxNeRF [17], we attempt to bypass some of the aforementioned limitations and believe that this framework will accelerate the widespread adoption of NeRFs.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. *Advances in neural information processing systems*, 34:26289–26301, 2021.

[2] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995.

[3] Saifullahi Aminu Bello, Shangshu Yu, Cheng Wang, Jibril Muhmmad Adam, and Jonathan Li. Deep learning on 3d point clouds. *Remote Sensing*, 12(11):1729, 2020.

[4] Antoine Beyeler, Michela Paganini, Benjamin Marlin, and Anton Osokin. Frequency positional encodings for efficient representation learning in vision. In *International Conference on Learning Representations*, 2021.

[5] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2011.

[6] Thorsten M Buzug. Computed tomography. In *Springer handbook of medical technology*, pages 311–342. Springer, 2011.

[7] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.

[8] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.

[9] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron, 2018.

[10] Jianfei Guo, Zhiyuan Yang, Xi Lin, and Qingfu Zhang. Template nerf: Towards modeling dense shape correspondences from category-specific object images. *arXiv preprint arXiv:2111.04237*, 2021.

[11] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[13] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[14] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022.

[15] Shuja Khalid and Frank Rudzicz. wildnerf: Complete view synthesis of in-the-wild dynamic scenes captured using sparse monocular data. *arXiv preprint arXiv:2209.10399*, 2022.

[16] Byung-soo Kim, Pushmeet Kohli, and Silvio Savarese. 3d scene understanding by voxel-crf. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1425–1432, 2013.

[17] Naruya Kondo, Yuya Ikeda, Andrea Tagliasacchi, Yutaka Matsuo, Yoichi Ochiai, and Shixiang Shane Gu. Vaxnerf: Revisiting the classic for voxel-accelerated neural radiance field. *arXiv preprint arXiv:2111.13112*, 2021.

[18] Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

[19] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[20] 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.

[21] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[22] 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.

[23] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[24] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.

[25] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[26] 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.

[27] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[28] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022.

[29] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[30] Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3069, 2018.

[31] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012.

[32] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017.

[33] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[34] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021.

[35] Andriyoga Twinanda, Sherif M Shehata, Didier Mutter, Jacques Marescaux, and Michel de Mathelin. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2016.

[36] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[37] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619, 2021.

[38] Jane Wilhelms and Allen Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics (TOG)*, 11(3):201–227, 1992.

[39] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[41] Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown. Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling. In *2021 International Conference on 3D Vision (3DV)*, pages 962–971. IEEE, 2021.

[42] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019.

[43] 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.

[44] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020.

[45] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[47] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.