Enriching Unbounded Appearances for Neural Radiance Fields

Ahmad AlMughrabi[1(✉)], Umair Haroon[1], Ricardo Marques[2], and Petia Radeva[1,3]

[1] Universitat de Barcelona, 08007 Barcelona, Spain
{ahmad.almughrabi,umairharoon,petia.ivanova}@ub.edu
[2] Universitat Pompeu Fabra, Barcelona, Spain
ricardo.marques@upf.edu
[3] Institut de Neurociències, 08035 Barcelona, Spain
https://www.ub.edu/aiba/, https://www.upf.edu/web/gti/,
https://www.neurociencies.ub.edu/

Abstract. Neural radiance fields (NeRF) have recently appeared as a powerful tool for generating realistic views of objects and confined areas. Still, they face serious challenges with open scenes, where the camera has unrestricted movement, and content can appear at any distance. In such scenarios, current NeRF-inspired models frequently yield hazy or pixelated outputs, suffer slow training times, and might display irregularities because of the challenging task of reconstructing an extensive scene from a limited number of images. We propose a new framework to boost the performance of NeRF-based architectures yielding significantly superior outcomes compared to the prior work. Our solution overcomes several obstacles that affected earlier versions of NeRF, including handling multiple video inputs, selecting keyframes, and extracting poses from real-world frames that are ambiguous and symmetrical. Furthermore, we applied our framework, called "Pre-NeRF 360", to enable the use of the Nutrition5k dataset in NeRF and introduce an updated version of this dataset, known as the N5k360 dataset. The source code, the dataset, and pre-trained weights for Pre-NeRF are publicly available at (https://amughrabi.github.io/prenerf).

Keywords: NeRF · 3D Reconstruction · N5k360 Dataset

1 Introduction

Accurate 3D reconstruction from images is a fundamental problem in computer vision, underpinning applications in robotics, augmented reality, and beyond. The Nutrition 5k [45] dataset introduces unique challenges, requiring the reconstruction of objects with intricate geometries, diverse textures, and significant

R. Marques and P. Radeva—Equal supervision.

N. Gonçalves et al. (Eds.): IbPRIA 2025, LNCS 15937, pp. 80–96, 2026.
https://doi.org/10.1007/978-3-031-99565-1_7

variability in real-world capture conditions. The need for precise cameras exacerbates these challenges and poses estimation and effective keyframe selection, essential for robust and efficient reconstruction.

Traditional 3D reconstruction pipelines often rely on Structure-from-Motion (SfM) techniques [18,40] to estimate camera poses and sparse scene representations, followed by Multi-View Stereo (MVS) methods [15,17,41,47,55] to recover dense geometry. While effective in controlled environments, these methods struggle in challenging real-world scenarios, such as those presented by the Nutrition 5k dataset. Casually captured images often include repetitive textures (e.g., packaging labels) and featureless surfaces (e.g., smooth bottles), which lead to pose estimation errors. These errors propagate through the reconstruction process, significantly degrading the quality of the final 3D models.

Recent advancements, such as Neural Radiance Fields (NeRFs) [32], have demonstrated remarkable performance in high-quality image synthesis and novel view synthesis (NVS) [4,44]. NeRFs implicitly encode scene information into neural networks, enabling photorealistic rendering and surface-aware reconstruction [25,48,50]. However, these methods remain heavily dependent on accurate camera pose initialization, often using SfM systems like COLMAP [40]. When SfM fails or produces erroneous poses, NeRF-based approaches also falter, limiting their utility for practical datasets like Nutrition 5k. Additionally, large-scale datasets need efficient keyframe selection to balance computational efficiency and reconstruction fidelity.

To overcome these limitations, we propose a novel framework to tackle these challenges, integrating NeRFs with scene graph optimization to enhance camera pose estimation and enable robust keyframe selection. Our approach begins with an SfM-based pose initialization [18,40] and incorporates a joint optimization process leveraging scene graph representations. By modeling spatial relationships among frames in the scene graph, our framework effectively mitigates pose inaccuracies, even under significant outliers. Furthermore, an adaptive confidence mechanism identifies and downweights unreliable frames caused by repetitive patterns or textureless regions [5,10,52,54], ensuring that outliers do not compromise reconstruction quality.

Our method includes a structured keyframe selection strategy informed by the scene graph's topology. This strategy maximizes spatial coverage and minimizes redundancy, improving computational efficiency without sacrificing reconstruction fidelity. A coarse-to-fine optimization strategy ensures stability and efficiency, enabling global pose corrections during initial stages and refining local details in subsequent steps [7,9,22,28].

We validate our framework on the Nutrition 5k [45] and MipNeRF 360 [4] datasets, demonstrating its robustness in addressing significant pose estimation errors and achieving state-of-the-art reconstruction quality. Our experiments reveal its effectiveness in handling unstructured image capture scenarios, surpassing existing methods in geometric accuracy and computational efficiency.

2 Related Work

Rendering extensive unbounded scenes using NeRF architectures raises three crucial challenges: *parameterization*, *efficiency*, and *ambiguity* [4].

Parameterization: refers to defining a set of parameters or constraints that can represent the space of a 3D scene. Allocating more capacity to nearby objects and less to distant ones is crucial for accurate rendering unbounded scenes. NeRFs are successful in pairing specific scene types with appropriate 3D parameterizations. Scenes unbounded in all directions require different parameterizations, as explored by NeRF++ [58] and DONeRF [33], which shrink distant points towards the origin. Mip-NeRF 360 extends this idea to Mip-NeRF by presenting a way to apply smooth parameterization to volumes and introducing their parameterization for unbounded scenes.

Efficiency: Training NeRF and Mip-NeRF-like architectures for complex scenes is resource-intensive due to densely querying large MultiLayer Perceptrons (MLP)s along each ray. Mip-NeRF 360 improves efficiency by employing two MLPs: a Proposal MLP for volumetric density prediction and a NeRF MLP for rendering. This approach leverages frequent Proposal MLP evaluations and fewer NeRF MLP evaluations, increasing capacity (15×) with a modest training time increase (2×) while enhancing rendering quality by 300% [4]. Methods like distillation and compression speed up rendering [20,36,56], but not training, and Neural Sparse Voxel Fields [30] with octree acceleration provide limited training time reduction [4].

Ambiguity: A key limitation of NeRF-like architectures is their struggle to create high-quality 3D models from 2D images and generate realistic novel views [20,32,35,60]. While some approaches address issues like non-smooth surfaces and slow rendering [4], they differ from Mip-NeRF 360, which operates on continuous weights along rays rather than point samples. Many techniques also fail to handle unbounded 360-degree scenes; for example, RegNeRF [34] produces empty or dark-brown frames due to its inability to process spherical scenes effectively.

We introduce *Keyframe selection* to filter out blurry, noisy, or redundant input images to reduce ambiguity in our data model. We prioritize removing corrupted frames over recovery to minimize the number of frames while preserving essential information for real-world scenes. Methods like Deblurring [8,43], Image-Super Resolution [49], or Image Restoration [26,27] may degrade our approach by introducing artifacts, omitting small details, accumulating errors, and negatively impacting speed and memory usage. Furthermore, these methods often depend on prior knowledge, which adds constraints to NeRF-like systems.

Our Reduction Detector layer uses two filters to detect Defocus Blur and near-Image Similarity in 2D input images. Defocus Blur is handled with Fast Fourier Transformation (FFT) [1] and the Laplacian method, while near-image Similarity is detected using Perceptual Hashing (P-Hash) [21] and Hamming distance thresholding. These techniques improve data accuracy by removing

redundant, blurry, and noisy images, which are then passed to Camera Pose Estimation (CPE) for more precise pose and feature extraction in unbounded scenes.

Camera Pose Estimation is crucial for providing NeRF with the viewing direction (θ, Φ). It determines the camera's position and orientation in 3D space, vital for applications like augmented reality [2], 3D reconstruction [14], and robotics [42]. Accurate pose estimation is key to generating realistic 3D scene renderings. Techniques such as Colmap [40], SuperGlue [39], Hloc [38], and PixSfM [29] estimate poses even in challenging scenarios. Colmap, used by most NeRF-like methods, is a robust Structure from Motion (SfM) technique that addresses issues like geometric verification and outlier filtering. However, SfM struggles with symmetrical or duplicated structures [11,53], a problem Colmap also faces. Previous works [29,38,39] build on Colmap to tackle this. We integrate PixSfM into our NeRF framework, as it handles complex scene appearances with high precision, speed, and efficient memory use.

PixSfM is an advanced computer vision framework that enhances SfM accuracy by refining key-point locations and camera poses using low-level image information from multiple views. Built on Hloc [38], SuperGlue [39], and Colmap, PixSfM improves scene geometry and camera pose accuracy, even in challenging scenarios. Due to its robustness and versatility, we replace Colmap with PixSfM in our proposed data model for enhanced camera pose estimation (CPE).

In summary, in this paper, we designed a framework to address the challenges faced by NeRF-like methods in generating realistic renderings of unbounded scenes. The main **contributions** of this article are:

1. We introduce a new Reduction deduction layer in the Pre-NeRF 360 framework to address the ambiguity issue that arises in unbounded scenes. This layer incorporates two optimized filters, Defocus Blur and near-Image Similarity eliminating noisy, blurry and redundant input images.
2. We adopt an advanced and innovative camera pose estimation technique called PixSfM [29] instead of the usually used Colmap tool for Camera Pose Estimation [40]. PixSfM enables us to obtain more intricate and accurate camera pose and features, which helps any NeRF-like architectures to generate highly detailed and photo realistic renderings of unbounded scenes. We show that our framework based on PixSfM is able to extract dynamic and high-level features, and differentiate between extracted camera positions that correspond to common features, reducing significantly the ambiguity in camera pose estimation.
3. We offer an enhanced and upgraded version of the public Nutrition5k dataset called N5k360. We enabled it to be accepted for all NeRF-like architecture in order to create 3D view and rendering of the dishes. The N5k360 dataset contains all Nutrition5k dishes, and will be done publicly available after publishing the paper to be utilized with any NeRF-like architecture.
4. We conducted a comprehensive set of experiments using the Mip-NeRF360 and our proposed N5k360 dataset. In comparison with Mip-NeRF360 dataset, our proposed framework achieved slightly improved and better performance

with very fewer iterations. We validated the results using three evaluation metrics (PSNR, SSIM, LPIPS), improving the state-of-the-art performance.

The rest of this paper is structured as follows: We defined our proposed framework in Sect. 3. A thorough set of experimental results is presented in Sect. 4. Finally, we present our conclusions and future work in Sect. 5.

3 Proposed Methodology

This section presents a detailed description of our framework. Our study focuses on how to improve the NeRF input data (blurred, noisy, and redundant input images and pose estimations) in order to alleviate the ambiguity problem of NeRF-based models.

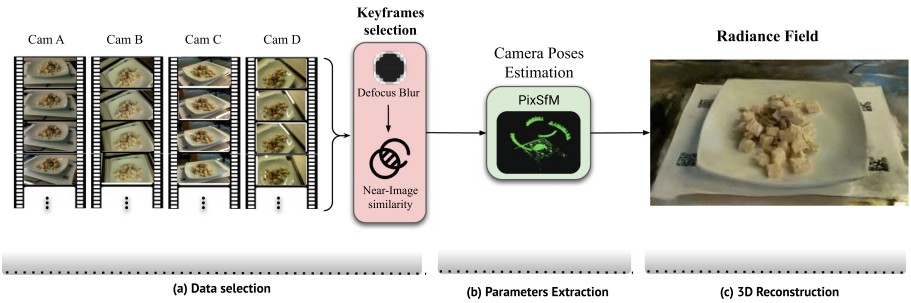

Fig. 1. Our proposed framework diagram, which outlines the entire workflow from a set of videos to any NeRF-like application. The data representation in the diagram consists of four input videos taken from the Nutrition5k dataset.

3.1 Overview

Figure 1 shows a diagram representing the general overview of our proposed approach. The input of our model is a set of videos (or a collection of images) that is used to extract a set of keyframes represented by a subsampled set of data (e.g., every k^{th} frame) from each video. Our framework is composed of three main parts: reduction detector, camera pose estimation and converter. Within the reduction detector part, **Defocus Blur reduction** module removes all the blurry frames from the list of sampled frames; followed by a **near-Image Similarity reduction** module removing the duplicate frames from the blurry-free sampled frames. Consequently, a **Camera Poses Estimation** module extracts the low detailed features and match them, resulting in the cameras' locations (i.e. poses). Lastly, a **Converter** module reformats the Camera poses information and the frames into parsable NeRF-based formats such as LLFF or Blender.

3.2 Preliminaries: NeRF for View Synthesis

NeRFs [32] can be formalized as a function F_Θ which takes as input a continuous 5D coordinate and yields a color and a density at that input location, such that:

$$F_\Theta : (\mathrm{x}, d) \rightarrow (c, \sigma)\,, \tag{1}$$

where $\mathrm{x} = (x, y, z)$ is a 3D spatial location, $d = (\theta, \Phi)$ represents a spherical direction, $c = (r, g, b)$ is the color at the 5D input coordinate and σ is the corresponding volume density.

NeRFs are typically parameterized as an MLP, which captures the 5D radiance field of a given 3D scene. This inherently volumetric information is then projected onto the image plane defined by a virtual camera to synthesize a new (unseen) photo-realistic image. The typical approach uses direct volume rendering, where virtual rays r with origin o at the virtual camera position are cast towards the scene. The number of rays equals the number of pixels in the synthetic image. Their direction d is set so that each ray $r(t) = o + td$ passes through the center of each image pixel, with $t \geq 0$ being the ray parameter determining the current position along the given ray. The color of each pixel is then computed by marching along its corresponding ray and collecting the color c and volume density σ values at a set of discrete locations. Finally, the collected values are used to compose the final pixel color. This process can be formalized as [32]:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_i\delta_i)))c_i, \;\; T_i = exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right),$$

where N refers to the number of sampled points across the ray, $\delta_i = t_{i+1} - t_i$ refers to the distance between two adjacent samples, c_i and σ_i refer to the per-point radiance and density, and T_i refers to the accumulated transmittance.

Since NeRF is differentiable, it is possible to define a loss function that allows NeRF training to minimize the mean-squared error (MSE) between the predicted renderings and the corresponding ground-truth colors:

$$\mathcal{L}_{MSE} = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\left\|\hat{C}(r) - C(r))\right\|^2$$

where $\mathcal{R}$ refers to the batch of the randomly sampled rays that belong to one or all training images, while $\hat{C}(r)$ and $C(r)$ refer to the ground truth and output color of ray r. Notably, this per-pixel optimization approach lacks holistic spatial understanding and makes NeRF sensitive to disturbance in pixel intensity.

Camera Pose Estimation: Since the input of NeRF is a continuous 5D coordinate containing a point in the space and the camera pose, CPE is used to compute the camera pose from a set of videos or frames. Assume CPE is a function called E that accept a set of frames (X): $E(X) = I(x_i, \theta, \Phi)$, and returns $I(x_i, \theta, \Phi)$, which is the estimated view direction (θ, Φ) with the associated frame, $x_i \in X$.

PixSfM [29] computes features by using deep Convolutional Neural Network (CNN)s. PixSfM provides direct alignment of low-level image information from multiple views: it first adjusts initial keypoint locations prior to any geometric estimation, and subsequently refine points and camera poses as a post-processing. In particular, PixSfM adjusts both keypoints and bundles, before and after reconstruction, by direct image alignment in a learned feature space. Exploiting this locally-dense information is significantly more accurate than geometric optimization, while deep, high-dimensional features extracted by a CNN ensure wider convergence in challenging conditions. The approach first refines the 2D keypoints only from tentative matches by optimizing a direct cost over dense feature maps. The second stage operates after SfM and refines 3D points and poses with a similar feature metric cost. Thus, the formulation elegantly combines globally-discriminative sparse matching with locally-accurate dense details. The refinement is robust to large detection noise and appearance changes, due to the optimization of the feature metric error based on dense features predicted by a neural network.

3.3 Pre-NeRF 360

Our framework aims to improve the input of the NeRF framework in order to overcome the abovementioned artifacts. Let us consider as input multiple videos X from different cameras or the same camera, as shown in Fig. 1: $X = \{X_i | i \in (1, n)\}$ where X is a set of videos, X_i is the i-th video, where $i \in [0, n]$, and n is the number of videos in X.

Key Frame Extraction: In order to minimise the redundant frames and speed up the process we first subsample the videos namely, from each video we select every k^{th} frame obtaining $X' = S(X_i, k)$. Since duplicates in frames causes instability and less robustness in the model training [61], the subsampling process reduces the likelihood of duplicates in the input data.

Defocus Blur Reduction D_{Red}: When the inputs are corrupted, such as compressed in JPEG format or blurred due to motion, the reconstructed scenes may show apparent artifacts. Corruptions often occur during the real-world capture and preprocessing stages. It potentially seems to lead to inaccurate reconstruction. Still, the more essential and interesting question —*how different corruption severity and types impact the robustness of NeRF*— is still an open field to explore.

In order to tackle this problem by removing blurry images, we propose to measure image sharpness in the frequency domain using the Fast Fourier Transform (FFT) with a Blur degree threshold h_b [12]: $X'' = \{x_i \in X' \mid \mathrm{FFT}(x_i) > h_b\}$,

Near-Image Similarity Reduction: The frame sampling approach of the key frame generates a list of images $X_{red} \subseteq X$ that still could have frames redundancy. This can occur when the camera takes longer than k^{th} frames to move while recording. We apply a **near-image similarity** [46] method to detect the duplicates among X', especially to handle too smooth or slow camera movement

behaviour. To detect near-image similarity, we apply Perceptual hashing function (P_{Hash}) [57] that generates a unique fingerprint for each frame based on its content. P_{Hash} examines the features of a frame and generates a 64-bit number fingerprint. Then, all hashes are constructed in BKTree [6] data structure to find "the closest" hashes. After that, we use Hamming Distance (HD) to find the P closest hashes corresponding to similare image frames.

Assume H denotes a hash function which takes one frame as a given input and return a binary string of length l. Assume x indicates a particular frame in X', and $\hat{x}$ denotes a modified version of this frame which is "perceptually similar" to x. Assume y denotes a frame that is "perceptually different" from x in X'. Assume x' and y' denote hash values. $0/1^l$ represents binary strings of length l. Then, the four desired properties of a perceptual hash are defined as follows:

(i) **Equal distribution (unpredictability)** of hash values, where the probability of any hash value $H(x) = x'$ is approximately $\frac{1}{2^l}$ for all $x' \in \{0,1\}^l$, with P denoting probability and l the hash code length.
(ii) **Pairwise independence**, ensuring that perceptually different frames x and y satisfy $P(H(x) = x' \mid H(y) = y') \approx P(H(x) = x')$ for all $x', y' \in \{0,1\}^l$.
(iii) **Invariance**, ensuring that perceptually similar frames x and $\hat{x}$ yield $P(H(x) = H(\hat{x})) \approx 1$.
(iv) **Distinction**, ensuring that perceptually different frames x and y yield $P(H(x) = H(y)) \approx 0$. The Hamming distance (HD), $D(u, v)$, counts the number of differing indices (i) between binary strings u and v of length l, defined as $D(u \mid H(x), v \mid H(y)) = |\{i : u_i \neq v_i, i = 1, \ldots, l\}|$. Additionally, the function $X_{\text{red}} = N(X', h_s)$ [21] identifies near-similarity frames within a threshold h_s on the HD, expressed as $X_{\text{red}} = N(\forall x_i \in X', \forall y_i \in X') = \{y_i \mid P(H(x_i), H(y_i)) \approx 1, x_i \neq y_i\}$, returning a subset $X_{\text{red}} \subseteq X'$ of duplicate frames where, for each $x_i \subseteq X_{\text{red}}$, there exists a duplicate $y_i \subseteq X'$.

In order to achieve the efficient NeRF performance, we remove X_{red} from X'' as follows: $X''' = \{x_i \in X'' \mid x_i \notin X_{\text{red}}\}$, where the frames in X''' are obtained removing all frames with a similar hash code controlled by the HD threshold h_s (frames with a smaller Hash distance than h_s are considered redundant and removed). Obtaining X''' with a distance lower than h_s leads to high frames overlapping. Note that ensuring the scene has balanced (high enough, but not redundant) data is still an open question for NeRFs.

Note that obtaining X''' is exponentially impacted by the number of frames in X''; cleaning up the de-focus blurred images before entering the near-Image Similarity Reduction step leads to significantly lower computation and better reduction speed.

To ensure compatibility with NeRF models, data from the reduction detector and camera pose estimation undergo validation and preparation. This refines camera poses to correct inconsistencies, aligning them with the scene's geometry. The validated frames and poses are formatted for NeRF inputs like LLFF or Blender, which require precise camera parameters and processed image data. These procedures guarantee input quality for NeRF synthesis, minimiz-

ing ambiguity and enabling accurate, photo-realistic 3D scenes reconstructions. Figure 1(c) shows the final result of the Pre-NeRF model.

3.4 Our Proposed Dataset: N5k360

We propose an advanced 360 food dataset named N5k360 being an improved and enhanced version of the Nutrition5k dataset compatible with all NeRF architectures. Our aim was to construct a 360° viewed 5,000 realistic food dishes that are created from Nutrition5k dataset, as mentioned in Fig. 3.

Being Nutrition5K a real dataset referring to a complex real domain, i.e., food domain, we encountered numerous issues and challenges when adapting the Nutrition5k dataset for NeRF, including low-resolution videos, errors caused by humans and cameras, and blurry and redundant images, as shown in Fig. 2. Since the dishes were recorded using four Raspberry Pi cameras, the resulting low-resolution videos harmed the PSNR value. Furthermore, the footage contained numerous blurry and redundant images due to the camera's brief recording pauses. Moreover, the videos also included some human errors, such as hands, mobile phones, and lagging in the recorded videos. Therefore, firstly, we filtered these video images from the human errors and then passed them to our proposed data model, where we applied the Keyframes selection, which filters the blurry, noisy, and redundant images followed by the CPE and NeRF.

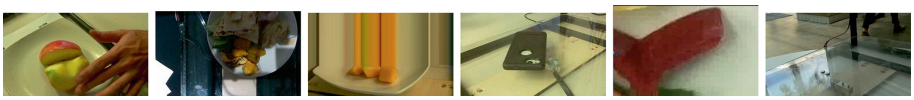

Fig. 2. Examples of challenges encountered with the Nutrition5k dataset.

To generate the N5k360 dataset, we extend our framework by including an image-rotation pipeline designed to rotate inverted frames by 180°. This pipeline aims to ensure proper synchronization of all frames captured by the recording cameras, as shown in Fig. 3. In some cases, repeating the workflow with a higher threshold h_b was necessary as the CPE process led to undesired outcomes, such as blurry frames. Insufficient features extracted from the frames caused the CPE to fail in estimating camera poses for all frames in a scene, leading to exclusion from the feature matching by the SfM.

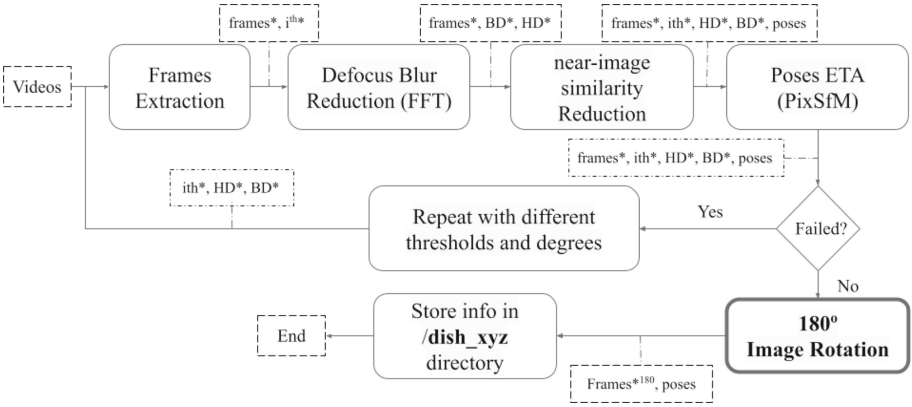

Fig. 3. Our framework with an additional step of data rotation 180°, to apply it on the Nutrition5k dish videos.

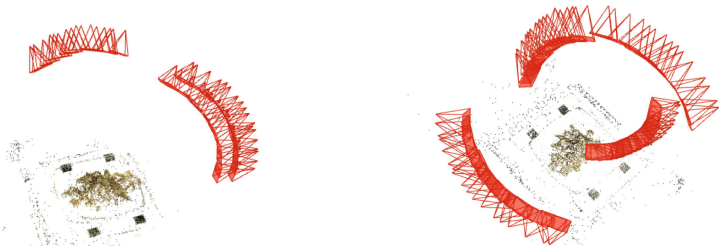

Fig. 4. Comparison of feature extraction: (Left) Colmap features using the N5k360 dataset, (Right) PixSfM features using the same dataset.

4 Experimental Results

In this section, we comment on the Mip-NeRF 360 dataset, the implementation setting, the results, and the comparison of PixSfM to Colmap CPE. Our proposed framework was evaluated on both multifaceted datasets, Mip-NeRF 360 and N5k360. As usual for NeRF evaluation, we provide the results for three error metrics, namely PSNR, SSIM, and LPIPS. The Training Time (hrs), Model Parameters (M), number of Iterations, and the Equivalent number of Iterations to TPU are also presented for all the scenes in the datasets.

4.1 Dataset

The Mip-NeRF 360 dataset [4] comprises 9 different scenes, including 5 outdoor and 4 indoor scenes. Each scene features a sophisticated central area and intricates background details. To capture these scenes, specific measures were taken to minimize photometric variations by fixing camera exposure settings, minimizing lighting changes and avoiding moving objects.

4.2 Implementation Setting

We used TPU v2 with 32 cores [23], and linear scaling rule [16] to fit our model configurations with NVIDIA GeForce RTX 3090/24G. Our model has 4096 as a batch size, a learning rate that is annealed log-linearly from 5×10^{-4} to 5×10^{-6}, and 1×10^{6} iterations. Simultaneously, we allocated 2 TPU v2 with 7 cores for each node without using the linear scaling rule for 3 days.

4.3 Pre-NeRF 360 Results

We evaluate our model on the two datasets: Mip-NeRF 360 dataset and the N5k360 dataset. In Table 1, we present the PSNR, SSIM [51], LPIPS [59], Time (hours), and Iterations mean values for all NeRF-like methods for the nine scenes. One can note that our model (backbone mipNeRF 360) outperforms all NeRF-like methods by 1.18x in LPIPS. Note that our model needs 3.17x less number of iterations. Moreover, our model outperforms Mip-NeRF 360 [4] w/GLO by 2.59%, while Mip-NeRF 2.64% in PSNR. In contrast, our model outperforms all NeRF-like methods, except Mip-NeRF 360 and Mip-NeRF 360 w/GLO in SSIM, outperforming our model by 2.02% and 1.27%, respectively. Figure 5 shows qualitative results on mip-NeRF 360 dataset and Nutrition 5k dataset.

Table 1. A quantitative comparison of our model with the SOTA on the Mip-NeRF 360 dataset. (*) denotes the NeRF-like method with Bigger MLP, while (+) denotes to NeRF-like method with Generative Latent Optimization (GLO) [31].

Method	PSNR↑	SSIM↑	LPIPS↓	Time(hrs)	Params	Iters
NeRF[13,32]	23.85	0.605	0.451	4.16	1.5M	250k
DoNeRF[33]	24.03	0.607	0.455	4.59	1.4M	250k
mip-NeRF[3]	24.04	0.616	0.441	3.17	0.7M	250k
NeRF++[58]	25.11	0.676	0.375	9.45	2.4M	250k
Deep Blending [19]	23.70	0.666	0.318	-	-	250k
Point-Based Neural Rendering [24]	23.71	0.735	0.252	-	-	250k
Stable View Synthesis [37]	25.33	0.771	0.211	-	-	250k
mip-NeRF [3] w/bigger MLP*	26.19	0.748	0.285	22.71	9.0M	250k
NeRF++ [58] w/bigger MLPs*	26.39	0.750	0.293	19.88	9.0M	250k
mip-NeRF 360 [4]	27.69	0.792	0.237	6.89	9.9M	250k
mip-NeRF 360 w/GLO [4]+	26.26	0.786	0.237	6.90	9.9M	250k
Ours	26.96	0.776	0.201	15.80	9.0M	78,75k

To validate our framework on the N5k360 dataset, we randomly selected the whole set of videos for 8 dishes from the Nutrition5k dataset and evaluated them on our proposed framework. These 8 dishes evaluations are shown in Table 2. We

present the PSNR, SSIM [51], LPIPS [59], Time (hours), and Iterations values for our model for randomly selected 8 scenes from N5k360 dataset. Our model shows in average of 24.25 PSNR, 0.81 SSIM, and 0.13 LPIPS, while the training time is 55 min for 50k iterations.

Table 2. The result of our model using randomly selected scenes from N5k360.

Dish ID	PSNR↑	SSIM↑	LPIPS↓	Time(hrs)	Params	Iters
50704750	25.55	0.85	0.133	0.55	9.0M	50k
50710793	24.53	0.80	0.12	0.55	9.0M	50k
50712459	25.57	0.82	0.11	0.55	9.0M	50k
50772617	26.86	0.86	0.11	0.55	9.0M	50k
50775219	26.86	0.83	0.11	0.55	9.0M	50k
50777256	24.99	0.82	0.11	0.55	9.0M	50k
61575996	17.54	0.60	0.22	0.55	9.0M	50k
61664061	22.10	0.81	0.11	0.55	9.0M	50k
Average	24.25	0.81	0.13	0.55	9.0M	50k

4.4 PixSfM vs Colmap CPE Comparison

Based on our experiments, we found that the classical frame-matching NeRF paradigm for CPE, i.e. Colmap, detects key points per frame once and for all, which can yield poorly-localized features and propagate significant errors to the final geometry. Furthermore, Colmap fails to detect low-level frame information, more precisely, the keypoints from multiple views, which makes Colmap's matching process failing to find any proper matches. Moreover, it gives view directions (θ, Φ) with a different range of error margins. For instance, when the camera symmetrically captured a low-level information indoor scene such as an N5k360 dish, the estimated camera poses are overlapped as shown in Fig. 4a, while PixSfM CPE can detect them precisely as shown in Fig. 4b. For instance, considering the amount of localized features, our experiments show that Colmap found fewer features as shown in 4c, while PixSfM CPE found much richer features.

(a) Bicycle PSNR=26.302, SSIM=0.670, LPIPS=0.334

(d) Garden PSNR=27.25, SSIM=0.824, LPIPS=0.174

(d) Kitchen PSNR=31.085, SSIM=0.918, LPIPS=0.062

(e) Dish(50704750) PSNR=25.553, SSIM=0.856, LPIPS=0.133

(f) Dish(50710793) PSNR=24.532, SSIM=0.780, LPIPS=0.126

(g) Dish(50712459) PSNR=25.574, SSIM=0.828, LPIPS=0.116

Fig. 5. Qualitative results of our framework, including mean PSNR, SSIM, and LPIPS metrics on both N5k360 and mip-NeRF 360 datasets.

Limitations. *Our approach may struggle with reflective or transparent surfaces due to feature matching and pose estimation challenges under dynamic lighting. Static color surfaces with limited texture may also hinder reliable pose estimation, causing inaccuracies in 3D reconstruction, especially with sparse or noisy data.*

5 Conclusions and Future Work

We developed a robust data framework for volume rendering that enhances NeRF-like models by tackling the challenges of complex data preparation. By utilizing defocus blur and detecting near-image similarities, we resolved issues in unbounded scenes and applied PixSfM for accurate camera pose estimation, which boosted NeRF's robustness. Our model surpassed existing frameworks with fewer training iterations, and we also created the N5k360 dataset, marking the first volumetric representation of food. While our results highlight the enriching the appearance of NeRF-based, questions persist regarding the effects of camera placement and the possibility of further reducing corruption. Our goal is to promote advancements in dependable NeRF for real-world applications.

Acknowledgments. This work was partially funded by the EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), and Grants PID2022141566NB-I00 (IDEATE), PDC2022-133642-I00 (DeepFoodVol), and CNS2022-135480 (A-BMC) funded by MICIU/AEI/10.13039/501100 011033, by FEDER (UE), and by European Union NextGenerationEU/ PRTR. A. AlMughrabi acknowledges the support of FPI Becas, MICINN, Spain. U. Haroon acknowledges the support of FI-SDUR Becas, MICINN, Spain.

References

1. Aravinth, S., Gopi, A., Chowdary, G.L., Bhagavath, K., Srinivas, D.R.: Implementation of blur image to sharp image conversion using laplacian approach. In: 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), pp. 339–345. IEEE (2022)
2. Baker, L., Ventura, J., Langlotz, T., Gul, S., Mills, S., Zollmann, S.: Localization and tracking of stationary users for augmented reality. Vis. Comput. 1–18 (2023)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: a multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5855–5864 (2021)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5470–5479 (2022)
5. Bian, W., Wang, Z., Li, K., Bian, J.W., Prisacariu, V.A.: Nope-nerf: Optimising neural radiance field with no pose prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4160–4169 (2023)
6. Burkhard, W.A., Keller, R.M.: Some approaches to best-match file searching. Commun. ACM **16**(4), 230–236 (1973)
7. Chen, B.Y., Chiu, W.C., Liu, Y.L.: Improving robustness for joint optimization of camera pose and decomposed low-rank tensorial radiance fields. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 990–1000 (2024)
8. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part VII, pp. 17–33. Springer (2022)
9. Chen, Y., et al.: Local-to-global registration for bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8264–8273 (2023)
10. Chng, S.F., Ramasinghe, S., Sherrah, J., Lucey, S.: Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In: European Conference on Computer Vision, pp. 264–280. Springer (2022)
11. Cui, Z., Tan, P.: Global structure-from-motion by similarity averaging. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 864–872 (2015)
12. De, K., Masilamani, V.: Image sharpness measure for blurred images in frequency domain. Procedia Eng. **64**, 149–158 (2013)
13. Deng, B., Barron, J.T., Srinivasan, P.P.: Jaxnerf: an efficient jax implementation of nerf (2020). https://github.com/google-research/google-research/tree/master/jaxnerf
14. Feng, C., Li, H., Gao, F., Zhou, B., Shen, S.: Predrecon: a prediction-boosted planning framework for fast and high-quality autonomous aerial reconstruction. arXiv preprint arXiv:2302.04488 (2023)
15. Furukawa, Y., Hernández, C., et al.: Multi-view stereo: a tutorial. Found. Trends® Comput. Graph. Vis. **9**(1-2), 1–148 (2015)
16. Goyal, P., et al.: Accurate, large minibatch SGD: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
17. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2495–2504 (2020)
18. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)
19. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Trans. Graph. (TOG) **37**(6), 1–15 (2018)
20. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5875–5884 (2021)
21. Jain, T., Lennan, C., John, Z., Tran, D.: Imagededup (2019). https://github.com/idealo/imagededup
22. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5846–5854 (2021)
23. Jouppi, N.P., et al.: In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, pp. 1–12 (2017)
24. Kopanas, G., Philip, J., Leimkühler, T., Drettakis, G.: Point-based neural rendering with per-view optimization. In: Computer Graphics Forum, vol. 40, pp. 29–43. Wiley Online Library (2021)
25. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: high-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8456–8465 (2023)
26. Liang, J., et al.: VRT: a video restoration transformer. arXiv preprint arXiv:2201.12288 (2022)
27. Liang, J., et al.: Recurrent video restoration transformer with guided deformable attention. arXiv preprint arXiv:2206.02146 (2022)
28. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5741–5751 (2021)
29. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5987–5997 (2021)
30. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Adv. Neural. Inf. Process. Syst. **33**, 15651–15663 (2020)
31. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7210–7219 (2021)
32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021)
33. Neff, T., et al.: Donerf: towards real-time rendering of compact neural radiance fields using depth oracle networks. In: Computer Graphics Forum, vol. 40, pp. 45–59. Wiley Online Library (2021)
34. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: regularizing neural radiance fields for view synthesis from sparse inputs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5480–5490 (2022)

35. Oechsle, M., Peng, S., Geiger, A.: Unisurf: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5589–5599 (2021)
36. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: speeding up neural radiance fields with thousands of tiny MLPs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14335–14345 (2021)
37. Riegler, G., Koltun, V.: Stable view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12216–12225 (2021)
38. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: robust hierarchical localization at large scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12716–12725 (2019)
39. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4938–4947 (2020)
40. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
41. Schönberger, J.L., Zheng, E., Frahm, J.-M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 501–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_31
42. Shim, S., Lee, S.W., Cho, G.C., Kim, J., Kang, S.M.: Remote robotic system for 3D measurement of concrete damage in tunnel with ground vehicle and manipulator. Comput.-Aided Civil Infrastruct. Eng. (2023)
43. Sun, L., et al.: Mefnet: multi-scale event fusion network for motion deblurring. arXiv preprint arXiv:2112.00167 (2021)
44. Tancik, M., et al.: Nerfstudio: a modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–12 (2023)
45. Thames, Q., et al.: Nutrition5k: towards automatic nutritional understanding of generic food. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8903–8911 (2021)
46. Thyagharajan, K., Kalaiarasi, G.: A review on near-duplicate detection of images using computer vision techniques. Arch. Comput. Methods Eng. **28**, 897–916 (2021)
47. Wang, F., Galliani, S., Vogel, C., Pollefeys, M.: Itermvs: iterative probability estimation for efficient multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8606–8615 (2022)
48. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)
49. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: training real-world blind super-resolution with pure synthetic data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1905–1914 (2021)
50. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: fast learning of neural implicit surfaces for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3295–3306 (2023)
51. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)

52. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf–: neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021)
53. Yan, Q., Yang, L., Zhang, L., Xiao, C.: Distinguishing the indistinguishable: exploring structural ambiguities via geodesic context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3836–3844 (2017)
54. Yan, Q., Wang, Q., Zhao, K., Chen, J., Li, B., Chu, X., Deng, F.: CF-nerf: camera parameter free neural radiance fields with incremental learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 6440–6448 (2024)
55. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 767–783 (2018)
56. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5752–5761 (2021)
57. Zauner, C.: Implementation and benchmarking of perceptual image hash functions (2010)
58. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)
59. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
60. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: neural factorization of shape and reflectance under an unknown illumination. ACM Trans. Graph. (TOG) **40**(6), 1–18 (2021)
61. Zheng, S., Song, Y., Leung, T., Goodfellow, I.: Improving the robustness of deep neural networks via stability training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4480–4488 (2016)