Improving Robustness for Joint Optimization of Camera Poses and Decomposed Low-Rank Tensorial Radiance Fields

Anonymous submission

Abstract

In this paper, we propose an algorithm that allows joint refinement of camera pose and scene geometry represented by decomposed low-rank tensor, using only 2D images as supervision. First, we conduct a pilot study based on a 1D signal and relate our findings to 3D scenarios, where the naive joint pose optimization on voxel-based NeRFs can easily lead to sub-optimal solutions. Moreover, based on the analysis of the frequency spectrum, we propose to apply convolutional Gaussian filters on 2D and 3D radiance fields for a coarse-to-fine training schedule that enables joint camera pose optimization. Leveraging the decomposition property in decomposed low-rank tensor, our method achieves an equivalent effect to brute-force 3D convolution with only incurring little computational overhead. To further improve the robustness and stability of joint optimization, we also propose techniques of smoothed 2D supervision, randomly scaled kernel parameters, and edge-guided loss mask. Extensive quantitative and qualitative evaluations demonstrate that our proposed framework achieves superior performance in novel view synthesis as well as rapid convergence for optimization.

1 Introduction

In recent years, neural rendering has become a widely-used method for high-quality novel view synthesis. NeRF as a pioneer work (Mildenhall et al. 2020) represents a 3D radiance field as an implicit continuous function built upon multilayer perceptrons (MLPs) which is trained with differentiable volume rendering. While achieving excellent synthesis quality, NeRF suffers from training/inference inefficiency due to dense evaluation of the computationally expensive MLPs.

To this end, voxel-based methods built upon the explicit scene representation of 3D voxel grid (Sun, Sun, and Chen 2022; Fridovich-Keil et al. 2022; Liu et al. 2020) are proposed to achieve faster training and provide better rendering quality than the original MLP-based NeRF, hence becoming the more preferred choices for downstream applications.

Nevertheless, maintaining a dense 3D voxel grid is in turn memory intensive, thus still restricting wider applications of voxel-based methods. Fortunately, TensoRF (Chen et al. 2022) proposes to tackle such memory-intensive issue of the voxel grid via replacing the dense 3D grid with *decomposed low-rank tensor*. TensoRF achieves a high data compression ratio and low computational cost at the same time while also achieving state-of-the-art performance. Providing a win-win situation on memory usage and computational efficiency, the decomposed low-rank tensor architecture has been widely adopted in many recent works (Xu et al. 2023; Fridovich-Keil et al. 2023; Goel et al. 2022; Han and Xiang 2023; Shao et al. 2023; Tang et al. 2022; Meuleman et al. 2023).

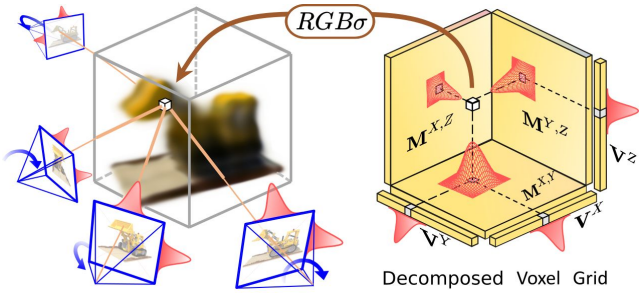

Figure 1: **Robust joint pose refinement on decomposed tensor.** Our method enables joint optimization of **camera poses** and **decomposed voxel representation** by applying efficient *separable component-wise convolution* of **Gaussian filters** on 3D tensor volume and 2D supervision images.

On the other hand, the effectiveness of NeRF (and most of the aforementioned works) hinges on precise camera poses of input images, which are often calculated using Structure-from-Motion (SfM) algorithms like COLMAP (Schönberger and Frahm 2016). While some works (Wang et al. 2021; Lin et al. 2021; Chng et al. 2022) aim to bypass the slow and occasionally inaccurate COLMAP process by optimizing camera pose and scene representation jointly on the original MLP-based NeRF, their success is often tied to the spectral bias (Yüce et al. 2022) of the MLP architecture which ensures the smoothness of 3D radiance field early in training. Voxel-based methods, however, lack such properties and can overemphasize sharp edges, making naive joint optimization problematic as getting trapped in local optima (Fig. 2 (a)).

In this work, we present simple yet effective methods for refining the camera pose and the 3D scene using decomposed low-rank tensors (cf. Fig. 1). We identify that controlling the frequency spectrum is vital for pose alignment, while directly realizing such control in a dense 3D grid could be nontrivial/challenging as well as computationally demanding. To this end, we introduce an efficient 3D filtering method using *component-wise separable convolution* for enabling the spectral control. To ensure stability in

the optimization process, we propose several techniques, including *smoothed 2D supervision*, *randomly scaled kernel paramter*, and the *edge-guided loss mask*. These techniques are experimentally proven crucial for successful pose refinement in our ablation studies. In results, our proposed method requires only 50k training iterations, where all the previous methods typically needs 200k iterations (e.g. the overall training time is reduced to 25%, compared to previous MLP-based methods). The main reason behind this advantage is that the separable component-wise convolution mechanism gives us very sensitive and accurate control on the spectrum of the 3D radiance field, allowing us to schedule the coarse-to-fine training more effectively. Moreover, our method performs favorably against state-of-the-art methods on novel view synthesis. Our contributions are three-fold:

- We derive and analyze the convergence of naive camera pose optimization with voxel-based radiance field reconstruction, and propose a learning strategy built upon the component-wise convolution where the coarse-to-fine training schedule is ensured.
- To enhance the robustness of our joint optimization, we introduce techniques of smoothed 2D supervision, scaled kernel parameters, and the edge-guided loss mask.
- Training time drops by 25% versus existing MLP-based methods, with requiring only 50k iterations against 200k of previous methods. Results show state-of-the-art performance in novel view synthesis with unknown pose.

2 Related Work

Accelerating Neural Rendering. As the seminal work of neural rendering, NeRF adopts MLPs to construct the implicit representation of the 3D scene, providing high-quality view synthesis but having a time-consuming training process due to the computational demands of MLPs. For addressing such issue, different variants of NeRF are proposed to use custom spatial data structures where the scene information is distributed only locally thus aiding faster training and rendering, in which those spatial data structures include *point cloud* (Xu et al. 2022; Hu et al. 2023), *space partitioning tree* (Wang et al. 2022; Yu et al. 2021), *triangular mesh* (Chen et al. 2023b; Kulhanek and Sattler 2023), and *voxel gird* (Sun, Sun, and Chen 2022; Fridovich-Keil et al. 2022; Liu et al. 2020; Hedman et al. 2021). Among these variants, the voxel grid has become more popular due to its easy implementation and quality reconstruction. However, as scene dimensions grow, the memory usage of the voxel grid becomes inefficient. To address this, (Müller et al. 2022) recommends compressing the grid via hash encoding, while (Chen et al. 2022; Fridovich-Keil et al. 2023) suggest adopting tensor decomposition for 3D feature compression, in which our method is mainly based on (Chen et al. 2022) but can be adaptable to other tensor decomposition-based voxel structures like K-Planes (Fridovich-Keil et al. 2023).

Joint Pose Estimation on MLP-based NeRFs. (Wang et al. 2021) is one of the first NeRF-based attempts to tackle the joint problem of estimating camera poses and learning 3D scene representation by directly adjusting camera pose using

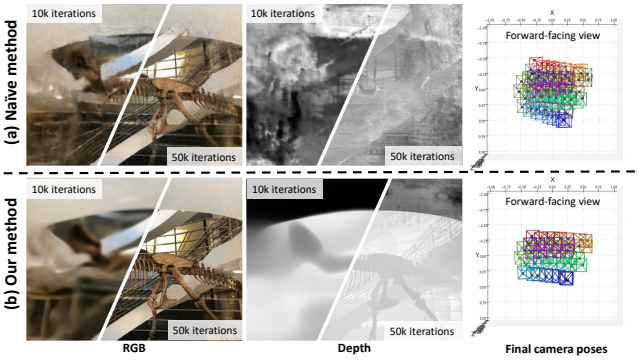

Figure 2: **Comparison of naive joint pose optimization and our proposed method on voxel-based NeRFs.** (a) Naively applying joint optimization on voxel-based NeRFs leads to dramatic failure as premature high-frequency signals in the voxel volume would curse the camera poses to stuck in local minima. (b) We propose a computationally effective manner to directly control the spectrum of the radiance field by performing *separable component-wise convolution* of Gaussian filters on the decomposed tensor. The proposed training scheme allows the joint optimization to converge successfully to a better solution.

gradient propagation on neural radiance fields. The robustness of such joint optimization is further enhanced by (Lin et al. 2021; Chng et al. 2022), where they propose various methods to smooth the pose gradient derived from the underlying MLP. (Chen et al. 2023a) further increases the noise tolerance by a specially designed local-global joint alignment approach. Our method also tackles joint problems but is specifically designed for the voxel-based NeRF built upon the decomposed low-rank tensor architecture.

Pose Estimation on Decomposed Low-rank Tensors. There do exist works that optimize camera pose on decomposed low-rank tensor (Liu et al. 2023; Meuleman et al. 2023) but require rich additional geometry clues (e.g., depth map and optical flow). To our best knowledge, we are the first attempt to jointly optimize the camera pose and the *decomposed low-rank tensor* using only 2D image supervision.

Pose Estimation on Multi-Resolution Hash Encoding. Aside from decomposed low-rank tensor, *multi-resolution hash encoding* is another compressed voxel-based architecture proposed by (Müller et al. 2022). Along with such a choice of architecture, recently (Heo et al. 2023) has proposed to address the joint optimization of camera pose and multi-resolution hash encoding. They suggest a new interpolation scheme that provides smooth gradients hence preventing gradient fluctuation in the hash volume, along with a curriculum learning scheme that controls the learning rate of the hash table at each resolution level. Although achieving impressive results on joint optimization, the effectiveness of their method is limited to multi-resolution hash encoding and is not applicable to *decomposed low-rank tensor*.

3 Our Proposed Method

3.1 Joint Refinement of 3D Scenes and Poses

Volume Rendering for Radiance Field Reconstruction. Based on the setting of neural volume rendering in NeRF, the

radiance fields respective for geometry and appearance for a 3D scene are represented via two functions (implemented by MLPs): $F_\sigma : \mathbb{R}^3 \rightarrow \mathbb{R}^1$ and $F_c : \mathbb{R}^6 \rightarrow \mathbb{R}^3$, where F_σ returns the volume density of an input 3D coordinate, while F_c outputs the color at an input 3D coordinate given a 3D viewing direction. For rendering a pixel on 2D coordinate u with its homogeneous form $\bar{u} = [u; 1]^\top$, we first sample a sequence of N 3D-coordinates $\{s_n\}_{n=1\cdots N}$ along the camera ray defined by the camera center $\vec{c} \in \mathbb{R}^3$ and the ray direction $\vec{d_u} = K^{-1}\bar{u}$,

$$\{s_n\}_{n=1\cdots N} = s(\vec{c}, \vec{d_u}) = \{\vec{c} + t_n \cdot \vec{d_u}\}_{n=1\cdots N}, \tag{1}$$

where K is the intrinsic camera matrix and $\{t_n\}_{n=1\cdots N}$ are N samples equidistantly distributed along the depth axis in between the near and far planes of the view frustum. The resultant color of the pixel is obtained by integrating through the density field F_σ and color field F_c using the volume rendering equation (Kajiya and Von Herzen 1984; Mildenhall et al. 2020), where we denote the *discretized volume rendering intregral* by a function $\mathbf{V}$:

$$\mathbf{V}(F_\sigma, F_c, s(\vec{c}, \vec{d_u})) = \sum_{s_n \in s(\vec{c}, \vec{d})} T_n \cdot \alpha_n \cdot \mathbf{C}_n, \tag{2}$$

where $T_n = exp(-\sum_{j=1}^{n} \delta_j F_\sigma(s_j))$ represents accumulated transmittance prior to s_n, $\alpha_n = 1 - exp(-\delta_n F_\sigma(s_n))$ represents the opacity of sample s_n, and $\mathbf{C}_n = F_c(s_n,\ \vec{d_u}\ )$ represents the color of sample s_n, and $\delta_j = \|s_j - s_{j-1}\|$ is the euclidean distance between two adjacent samples.

In the typical setting of NeRF, given a set of L 2D-images $\mathbf{I} = \{I_1, \cdots, I_L\}$ with their corresponding camera poses $\mathbf{P} = \{P_1, \cdots, P_L\} \in \mathfrak{se}(3)$ Lie algebra as input, we aim to reconstruct the 3D scene represented by F_σ^* and F_c^*, via minimizing the loss $\mathcal{L}_{\text{rec}}$ of 2D photometric reconstruction with the gradient-based optimization algorithm, in which

$$\mathcal{L}_{\text{rec}}(F_\sigma, F_c) = \sum_{i=1}^{L} \sum_{u \in \mathbf{U}} \|\mathbf{V}(F_\sigma, F_c, \mathcal{W}_{\text{3d}}(P_i, s(\vec{0}, \vec{d_u}))) - I_{iu}\|, \tag{3}$$

where $\mathbf{U}$ is the set of all possible 2D coordinates in the input images, $I_{iu} \in \mathbb{R}^3$ is the RGB color of pixel location u on training image I_i, warping function $\mathcal{W}_{\text{3d}}(P, \_) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ performs rigid 3D transformation parameterized by $P \in \mathfrak{se}(3)$ Lie algerbra, and $\mathcal{W}_{\text{3d}}(P, s(\vec{0}, \vec{d_u}))$ maps each sample 3D coordinate in canonical ray ($\vec{c} = \vec{0}$) into a 3D sample coordinate of camera ray with pose P. Note that this is an ill-posed reconstruction problem that suffers from shape-radiance ambiguity (Zhang et al. 2020).

3D Joint Optimization. When it comes to jointly estimating camera poses (where the camera poses $\mathbf{P}$ are also unknown) and learning scene representation (Lin et al. 2021; Chng et al. 2022; Chen et al. 2023a; Heo et al. 2023), the problem is even more ill-defined with the objective now being extended from Eq. 3 and defined as:

$$\mathcal{L}_{\text{joint}}(F_\sigma, F_c, \mathbf{P}) = \sum_{i=1}^{L} \sum_{u \in \mathbf{U}} \|\mathbf{V}(F_\sigma, F_c, \mathcal{W}_{\text{3d}}(P_i, s(\vec{0}, \vec{d_u}))) - I_{iu}\|. \tag{4}$$

Such joint optimization is highly influenced by the structural bias of the underlying representation of $\{F_\sigma, F_c\}$, which we will conduct a pilot study with a simpler 1D case in Sec. 3.2.

3.2 Gaussian Filter on 1D Signal Alignment

Here we aim to analyze the effect of the signal spectrum (spectrum of F_c, F_σ, and $\mathbf{I}$ in Eq. 4) on the joint optimization process. We begin by reducing 3D joint optimization of camera pose and scene reconstruction into a simpler 1D counterpart of signal alignment.

1D Signal Alignment. Let us consider a target ground truth 1D signal f_{GT} (assuming the signal to be continuous, bounded, and have finite support), which we aim to reconstruct and align with. We are given randomly translated versions f_1, f_2 of the ground truth signal f_{GT}, where $f_1 = \mathcal{W}_{\text{1d}}(f_{GT}, p_1)$, $f_2 = \mathcal{W}_{\text{1d}}(f_{GT}, p_2)$ with having $\mathcal{W}_{\text{1d}}$ a signal translation operation defined as $\mathcal{W}_{\text{1d}}(f, p)(x) = f(p - x)$, and p_1, p_2 are the translation values.

Although the reconstruction is trivial in such a 1D setting, in order to mimic the case of 3D joint optimization, we attempt to estimate a signal g as well as the translation values q_1 and q_2 via adopting the iterative gradient-based optimization on the reconstruction loss.

$$\begin{aligned}\mathcal{L}_{\text{1d}}(g, q_1, q_2) &= \sum_{i \in [1,2]} \int \|\mathcal{W}_{\text{1d}}(g, q_i)(x) - f_i(x)\|^2 dx \\ &= \sum_{i \in [1,2]} \int \|g(x) - f_{GT}(x - p_i + q_i)\|^2 dx.\end{aligned} \tag{5}$$

Note that Eq. 5 and Eq. 4 are analogous in terms of their structure/formulation, where the difference only lies in the dimensionality. And $\mathcal{L}_{\text{1d}}$ achieves the optimum whenever $q_1 - q_2 = p_1 - p_2$ and $g = \mathcal{W}_{\text{1d}}(f_{GT}, p_1 - q_1)$. Please check Figure 3(a) for a simple visual representation of Equation 5, where f_1 and f_2 are connected to g by the reconstruction loss $\mathcal{L}_{\text{1d}}$ (i.e blue arrows), whose gradients are used to update g and the translation values $\{q_1, q_1\}$.

Connection between 1D Signal Alignment and 3D Joint Optimization. The formulation of 1D signal alignment effectively simulates the "local phenomenon" of joint camera pose alignment and 3D scene reconstruction on a 2D cross-section: As shown in Figure 3(c), where we consider two neighboring camera poses as well as a cross-section in the 3D space passing through both camera planes and intersecting with each camera plane on a projected straight line, the RGB color values on such projected lines correspond to the 1D shifted ground truth signals f_1, f_2 in Equation 5, and the value of the radiance field on the cross-section corresponds to reconstructed signal g in Equation 5. Similar to the loss $\mathcal{L}_{\text{1d}}$ in Equation 5, the projected lines on the camera planes and the corresponding cross-section in the 3D radiance field are connected by the volume rendering function V and reconstruction loss $\mathcal{L}_{\text{joint}}$ in Equation 4. As a result, the complete 3D joint optimization can be intuitively viewed as simultaneously performing many 1D signal analyses on the superposition of all possible combinations of camera poses and cross-sections.

Spectrum Analysis and Effect of Gaussian Filtering on 1D Signal Alignment. If we assume rapid convergence of signal g (which means g achieves local optima w.r.t current q_1, q_2 whenever we update q_1, q_2.), we find that the problem

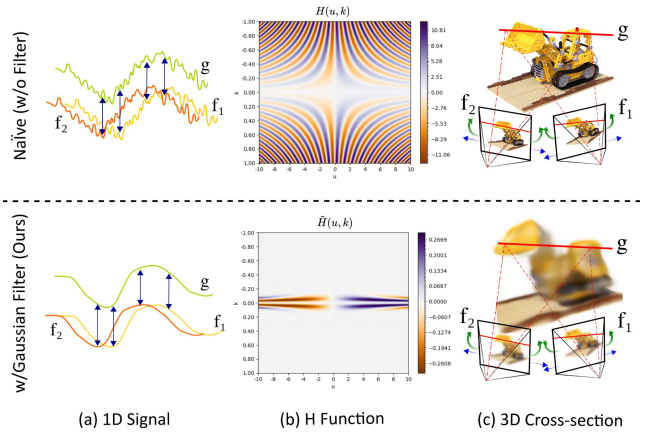

Figure 3: **Spectrum analysis and effect of Gaussian filtering on 1D signal alignment.** (a) 1D signal alignment comparison: noisy signals can get trapped in local optima without Gaussian filtering. (b)(*Top*) Visualization of $H(u,k)$ in Eq. 7, which shows alternating signs as k departs from 0, causing misdirection in gradient-based optimization if there has too much high-frequency energy in the signal. (b)(*Bottom*) Visualization of $\tilde{H}(u,k)$ in Eq. 8, which is the modulated version of $H(u,k)$ with the help of Gaussian filter $\mathcal{N}$. (c) 1D alignment relates to 3D joint optimization in Eq. 4, where effective pose refinement stems from the 1D alignment in specific cross-sections, with the red lines in 3D scene correlating to horizontal shifts (blue arrows) and rotations (green arrows).

is equivalent to pure alignment between two ground-truth signals (cf. our supplement for detailed derivation):

$$\mathcal{L}_{1d}(q_1, q_2) = \int \|f_{GT}(x) - f_{GT}(x+u)\|^2 dx, \tag{6}$$

where $u = (p_1 - p_2) - (q_1 - q_2)$ is the shift between two ground truth signals, which has an initialization of $p_1 - p_2$, and we aim for u to reach 0 with gradient-based optimization.

Moreover, by analyzing the relationship between f_{GT} and the optimization gradient $\frac{d}{du}\mathcal{L}_{1d}$ in terms of their spectral properties, we get the following result (cf. our supplement for detailed derivation):

$$\frac{d}{du}\mathcal{L}_{1d} = \int \| \mathfrak{F}[f_{GT}] \|^2 \cdot H(u,k)\, dk, \tag{7}$$

where $H(u,k) = 4\pi k\ sin(2\pi k u)$, $\mathfrak{F}[f_{GT}]$ is Fourier transform of f_{GT}, and k is the wavenumber in frequency domain.

Particularly, we are interested in the sign of $\frac{d}{du}\mathcal{L}_{1d}$ which determines the direction of our iterative optimization. We plot the value of $H(u,k)$ in Fig. 3(b)(*Top*), where we can observe that the sign of H is well-behaved when the magnitude of k is small (here well-behaving means the direction of the gradient is able to let u descend to 0, i.e., being positive when $u > 0$ and negative when $u < 0$). However, when k increases, the sign of H quickly begins to alternate, and the magnitude increases, which causes the gradient to be large and noisy. Hence high-frequency signals with a spreading spectrum can easily lead the optimization process to get stuck in the local optima.

To this end, we demonstrate that applying a Gaussian filter on the signal f_{GT} effectively mitigates the sign-alternating issue of the original H function. Specifically, we show that filtering the input signal is equivalent to modulating H by a Gaussian window (cf. our supplement for derivation):

$$\frac{d}{du}\tilde{\mathcal{L}}_{1d} = \int \| \mathfrak{F}[f_{GT}] \|^2 \cdot \tilde{H}(u,k)\, dk, \tag{8}$$

where $\tilde{H}(u,k) = \| \mathfrak{F}[\mathcal{N}] \|^2 \cdot H(u,k)$. Here $\tilde{\mathcal{L}}_{1d}$ basically is $\mathcal{L}_{1d}$ calculated with Gaussian convoluted signal $\mathcal{N} * f_{GT}$, and $\mathfrak{F}[\mathcal{N}]$ denotes the Fourier transform of the Gaussian kernel $\mathcal{N}$. In Fig. 3(b)(*Bottom*), we plot the modulated $\tilde{H}(u,k)$, with observing that the misbehave region is suppressed (note that we set the variance of $\mathcal{N}$ to 4 here). The gradient descent will likely converge to $u = 0$ once the initial magnitude of u is less than 6.0. The region where $\frac{d}{du}\tilde{\mathcal{L}}_{1d}$ does well-behave is *quasi-convex* and is guaranteed to converge to global optima given suitable learning rate that prevents us from getting stuck on saddle points. Our analysis agrees with the motivation behind the coarse-to-fine training schedule of (Lin et al. 2021) and (Heo et al. 2023). Specifically, observing that the well-behaved region in $H(u,k)$ grows wider as u approaches 0 (cf. Fig. 3(b)(*Top*)), which means that we can loosen the filtering strength of Gaussian kernel as u approaches 0, leading to larger and more accurate gradient.

3.3 2D Planar Image Alignment

In addition to the 3D joint optimization problem, previous works (Lin et al. 2021; Chng et al. 2022) also consider a 2D image patches alignment task as a simpler example of joint optimization, in which there are L overlapping image patches $\mathbf{I}_{2d} = \{I_1, \cdots, I_L\}$ cropped from a single ground truth image I_{gt} before being transformed by 2D homography. The homography transforms are parameterized by $\mathbf{P}_{2d} = \{P_1, \cdots, P_L\} \in \mathfrak{sl}(3)$ and initialized as $\vec{0}$. Analogously to Equation 4, our objective is to jointly optimize the 2D image content $F_{2d} : \mathbb{R}^2 \to \mathbb{R}^2$ and per-patch homography warps $\mathbf{P}_{2d}$ by the reconstruction loss. Joint optimization can be formulated as:

$$\mathcal{L}_{2d}(F_{2d}, \mathbf{P}_{2d}) = \sum_{i=1}^{L} \sum_{u \in \mathbf{U}_{2d}} \|F_{2d}(\mathcal{W}_{2d}(P_i, u)) - I_{iu}\|^2, \tag{9}$$

where $\mathbf{U}_{2d}$ is the set of all possible 2D coordinates in the image patches, I_{iu} is the color of pixel at location u on input image patch I_i, warp function $\mathcal{W}_{2d}(P_i, \_) : \mathbb{R}^2 \to \mathbb{R}^2$ performs 2D homography transformation parameterized by $P_i \in \mathfrak{sl}(3)$ Lie algebra, and $\mathcal{W}_{2d}(P_i, u)$ maps 2D coordinate u on I_{gt} into a transformed 2D coordinate on patch I_i. Notice the strong structural correspondence among Eq. 5 (1D alignment), Eq. 9 (2D alignment), and Eq. 4 (3D alignment), the three problems share similar computational property.

We parameterize F_{2d} by a 2D decomposed low-rank tensor $\mathrm{T}_{2d} \in \mathbb{R}^{h \times w}$, where w, h are the dimensions of the image. Motivated by our analysis in Section 3.2, we filter T_{2d} with 2D Gaussian kernel to avoid overfitting.

$$F_{2d}(\mathbf{x}) = (\mathcal{N}_{2d} *_{2d} \mathrm{T}_{2d})(\mathbf{x}) = (N_{2d} *_{2d} (\sum_{r=1}^{R} \mathbf{v}_r^X \otimes \mathbf{v}_r^Y))(\mathbf{x}), \tag{10}$$

where $\mathbf{x} \in \mathbb{R}^2$ is 2D pixel coordinates, $\mathcal{N}_{2d}$ is 2D gaus-

sian kernel, $*_{2d}$ is the convolution operator, and $\otimes$ denotes outer product between the 1D vector components $\mathbf{v}_r^X \in \mathbb{R}^w, \mathbf{v}_r^Y \in \mathbb{R}^h$. "$(\mathbf{x})$" at the end of the expressions means bilinearly interpolating the preceding discrete 2D volume with continuous coordinate $\mathbf{x}$. Our method outperforms the naïve tensor method and previous methods (Lin et al. 2021; Chng et al. 2022), experiment results are shown at Sec. 4.1.

The width of Gaussian kernel $\mathcal{N}_{2d}$ is controlled by an exponential coarse-to-fine training schedule that changes continuously (see the supplementary material for details about kernel schedule). In order to support continuous changing width on a discrete Gaussian kernel, the kernel is generated by the following rule:

$$\mathcal{N}_{1d}(\sigma) = \begin{cases} \bigoplus_{x=1}^{L_{\mathcal{N}}} min(1, \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}) & \text{if } \sigma > 0.0001 \\ \delta[x] & \text{otherwise,} \end{cases} \tag{11}$$

$$\mathcal{N}_{2d}(\sigma) = \mathcal{N}_{1d}(\sigma)^\top \otimes \mathcal{N}_{1d}(\sigma),$$

where $L_{\mathcal{N}}$ is the size of the discrete kernel, 1D kernel $\mathcal{N}_{1d}(\sigma) \in \mathbb{R}^{L_{\mathcal{N}}}$ is discretely sampled from continuous Gaussian distribution and clamped to a max value of 1.0 before being concatenated into a vector by $\oplus$ operator. To avoid numerical instability, when $\sigma < 0.001$, we assign $\mathcal{N}_{1d}(\sigma)$ to be discrete impluse function. 2D kernel $\mathcal{N}_{2d}(\sigma) \in \mathbb{R}^{L_{\mathcal{N}} \times L_{\mathcal{N}}}$ is generated by outer product of two 1D kernels.

3.4 Decomposed Low-Rank Tensor

This section describes the decomposed low-rank tensor proposed by TensoRF (Chen et al. 2022), in which two different types of tensor decomposition are considered: CP-decomposition and VM-decomposition. In our discussion, we focus on *VM-decomposition*, although our method is also naturally applicable to CP-decomposition.

To represent the 3D density field F_σ, we store the information in a 3D tensor $\mathrm{T}_\sigma \in \mathbb{R}^{I \times J \times K}$. F_σ is defined simply as component-wise interpolation of T_σ

$$\mathrm{T}_\sigma = \sum_{r=1}^{\mathbf{R}} \mathbf{v}_{\sigma,r}^X \otimes \mathbf{M}_{\sigma,r}^{Y,Z} + \mathbf{v}_{\sigma,r}^Y \otimes \mathbf{M}_{\sigma,r}^{X,Z} + \mathbf{v}_{\sigma,r}^Z \otimes \mathbf{M}_{\sigma,r}^{X,Y}, \tag{12}$$

where $\mathbf{R}$ is the number of components in the decomposition, $(\mathbf{V}_r^X, \mathbf{V}_r^Y, \mathbf{V}_r^Z) \in (\mathbb{R}^I, \mathbb{R}^J, \mathbb{R}^K)$ are 1D vector-components for axes (X, Y, Z) repectively, $(\mathbf{M}_r^{Y,Z}, \mathbf{M}_r^{X,Z}, \mathbf{M}_r^{X,Y}) \in (\mathbb{R}^{J \times K}, \mathbb{R}^{I \times K}, \mathbb{R}^{I \times J})$ are 2D matrix-components for axes $(Y\text{-}X, X\text{-}Z, X\text{-}Y)$ repectively, operator $\otimes$ denotes the outer product between vector and matrix.

To represent the 3D color field F_c, the information queried from 3D feature tensor $\mathrm{T} \in \mathbb{R}^{I \times J \times K \times G}$ is decoded by a small MLP S into RGB color value (G is the input feature dimension of S). The implementation can be formulated as

$$\begin{aligned} F_c(\mathbf{x}, \vec{d}) &= \mathbf{S}(\mathrm{T}_c(\mathbf{x}), \vec{d}) \\ \mathrm{T}_c &= \sum_{r=1}^{\mathbf{R}} \mathbf{v}_{c,r}^X \otimes \mathbf{M}_{c,r}^{Y,Z} \otimes \mathbf{b}_r^X + \\ &\quad \mathbf{v}_{c,r}^Y \otimes \mathbf{M}_{c,r}^{X,Z} \otimes \mathbf{b}_r^Y + \mathbf{v}_{c,r}^Z \otimes \mathbf{M}_{c,r}^{X,Y} \otimes \mathbf{b}_r^Z. \end{aligned} \tag{13}$$

$T_c(\mathbf{x})$ denotes the component-wise linear-interpolation of tensor volume T_c on 3D coordinate $\mathbf{x}$. $\vec{d}$ is the viewing direction of the current ray. $\mathbf{v}_{c,r}$ and $\mathbf{M}_{c,r}$ have the same shape as their $\mathbf{v}_{\sigma,r}$ and $\mathbf{M}_{\sigma,r}$ counterparts, $\mathbf{b}_r^X, \mathbf{b}_r^Y, \mathbf{b}_r^X \in \mathbb{R}^G$ are feature components to expand the feature axis of T_c.

3.5 Separable Component-Wise Convolution

As theoretically analyzed in Sec. 3.2 and empirically shown in Fig. 2(a), naïvely applying low-rank decomposed tensor to joint camera pose optimization results in suboptimal reconstruction quality and inaccurate poses. Therefore, we propose to limit the spectrum of the radiance field F_σ and F_c with a coarse-to-fine training schedule.

If we naïvely convolve the 3D Gaussian kernel with our 3D volume T_σ, (as in the 2D planar case of Eq. 10), we would have to reconstruct the whole 3D tensor before applying convolution, destroying the space compression advantage of decomposed low-rank tensor, see Eq. 14.

$$F_\sigma(x, y, z) = (\mathcal{N}_{3d} * T_\sigma)(x, y, z), \tag{14}$$

where $*_{3d}$ denotes 3D convolution, $\mathcal{N}_{3d}$ is the 3D Gaussian filter defined by $\mathcal{N}_{1d} \otimes \mathcal{N}_{2d}$. Under this setting, the time complexity is $O(I \cdot J \cdot K \cdot L_{\mathcal{N}}^3)$ and space complexity $O(I \cdot J \cdot K)$, where $L_{\mathcal{N}}$ is the size of 3D Gaussian kernel in each dimension.

To achieve computationally efficient convolution on the 3D decomposed low-rank tensor volume, we perform our proposed *separable component-wise convolution*, by taking advantage of the following identity (whose correctness will be proven in the supplementary material).

$$\tilde{\mathrm{T}}_\sigma = \sum_{r=1}^{\mathbf{R}} \tilde{\mathbf{v}}_{\sigma,r}^X \otimes \tilde{\mathbf{M}}_{\sigma,r}^{Y,Z} + \tilde{\mathbf{v}}_{\sigma,r}^Y \otimes \tilde{\mathbf{M}}_{\sigma,r}^{X,Z} + \tilde{\mathbf{v}}_{\sigma,r}^Z \otimes \tilde{\mathbf{M}}_{\sigma,r}^{X,Y}, \tag{15}$$

where $\tilde{\mathrm{T}}_\sigma = (\mathcal{N}_{3d} *_{3d} \mathrm{T}_\sigma)$ denotes the 3D Gaussian convoluted tensor volume, $\tilde{\mathbf{v}}_{\sigma,r} = (\mathcal{N}_{1d} *_{1d} \tilde{\mathbf{v}}_{\sigma,r})$ denotes the 1D Gaussian convoluted vector component, and $\tilde{\mathbf{M}}_{\sigma,r} = (\mathcal{N}_{2d} *_{2d} \tilde{\mathbf{M}}_{\sigma,r})$ denotes the 2D Gaussian convoluted matrix component. In other words, **the 3D convoluted tensor can be expressed as the composition of individually convoluted components**, which allows us to distribute the 3D Gaussian convolution across the individual components of the decomposed low-rank tensor. Similar to Sec. 3.4, the value of the density field is component-wised linearly sampled from the Gaussian convoluted components, i.e., $\tilde{F}_\sigma(\mathbf{x}) = \tilde{T}_\sigma(\mathbf{x})$.

Similarly, the spectral restricted version of the color field F_c can be obtained as

$$\begin{aligned} \tilde{F}_c(\mathbf{x}, \vec{d}) &= \mathbf{S}(\tilde{\mathrm{T}}_c(\mathbf{x}), \vec{d}) \\ \tilde{\mathrm{T}}_c &= \sum_{r=1}^{\mathbf{R}} \tilde{\mathbf{v}}_{c,r}^X \otimes \tilde{\mathbf{M}}_{c,r}^{Y,Z} \otimes \mathbf{b}_r^X + \\ &\quad \tilde{\mathbf{v}}_{c,r}^Y \otimes \tilde{\mathbf{M}}_{c,r}^{X,Z} \otimes \mathbf{b}_r^Y + \tilde{\mathbf{v}}_{c,r}^Z \otimes \tilde{\mathbf{M}}_{c,r}^{X,Y} \otimes \mathbf{b}_r^Z. \end{aligned} \tag{16}$$

With *separable component-wise convolution*, the time complexity required is $O(I \cdot J \cdot L_{\mathcal{N}} + J \cdot K \cdot L_{\mathcal{N}} + K \cdot I \cdot L_{\mathcal{N}})$ for computing convoluted components (assuming that we separate 2D Gaussian convolution on matrix components into 1D Gaussian convolutions), and $O(\mathbf{R})$ for each query sample (same as the original decomposed tensor in (Chen et al. 2022)), drastically reducing the computation required for filtering 3D radiance field F_σ and F_c.

3.6 Techniques for Increasing Pose Robustness

Here we summarize our improvements on naïve decomposed low-rank tensors that improve joint camera pose optimization and radiance field reconstruction.

Coarse-to-Fine 3D schedule. Using efficient 3D convolution algorithm in Sec. 3.5. During training, we apply a coarse-to-fine schedule on the 3D radiance field $\tilde{F}_\sigma, \tilde{F}_c$ by controlling the kernel parameter (σ of Eq. 11) of the Gaussian kernel, which is exponentially reduced to 0 at 10k iterations and remains 0 afterward.

Smoothed 2D Supervision. Inspired by the analysis in Sec. 3.2, we discovered that blurring the 2D training image with a parallel set of scheduled 2D Gaussian kernels also helps the joint optimization. On the one hand, smoothed supervision images produce smoothed image gradients and stabilize the camera alignment. On the other hand, smoothed training image also helps to restrict the spectrum of the learned 3D scene. The Gaussian schedule for smoothing 2D training images is similar to that of the 3D Gaussian.

Randomly Scaled Kernel Parameter. Scenes in the real world have different geometry structures (different depths and object shapes); hence, it is hard to get optimal reconstruction quality with a single manually designed kernel schedule. To mitigate the issue, we propose scaling the kernel parameter (σ of Gaussian) by a uniformly sampled factor between 0 and 1 in each training iteration. Random scales are sampled independently among 3D Gaussian kernels (for the radiance field) and 2D Gaussian kernels (for training images), allowing combinations of different-sized kernels to guide the joint optimization.

Edge-Guided Loss. Since the signal for pose alignment comes from the edge regions of the training images. We locate the edge regions of smoothed training images with the Sobel filter (Kanopoulos, Vasanthavada, and Baker 1988), and scale the reconstruction loss of pixels inside the edge regions by 1.5x. Empirically we apply this edge-guided scale alternately on every other training iteration.

4 Experiments

Although our method is applicable to various decomposed low-rank tensor implementations, in this section, we validate our proposed method using TensoRF (Chen et al. 2022) with inaccurate or unknown camera poses.

We evaluate our proposed method against three previous works **BARF** (Lin et al. 2021), **GARF** (Chng et al. 2022), and **HASH** (Heo et al. 2023). Since the implementation of GARF and HASH are unavailable, we directly use the results reported in their paper for comparison. We compare these methods on the **planar image alignment** task and novel view synthesis task on **NeRF-Synthetic** and **LLFF** dataset. We provide detailed implementation details and experimental setup in the supplementary material.

4.1 Results

Planar Image Alignment (2D). In Fig. 4 we compare our method (i.e., 2D TensoRF + 2D Gaussian) with naïve 2D TensoRF implementation (Chen et al. 2022) and BARF (Lin et al. 2021). Quantitative results are reported in Tab. 1, including $\mathfrak{sl}(3)$ warp error and patch PSNR. These results demonstrate the effectiveness of Gaussian filtering in joint optimization, verifying the analysis in Sec. 3.2.

Methods	$\mathfrak{sl}(3)$ error ↓	patch PSNR ↑
BARF	0.0105	35.19
Naïve 2D TensoRF	0.5912	20.80
2D TensoRF + 2D Gaussian	**0.0023**	**40.70**

Table 1: **Quantitative results of planar image alignment.**

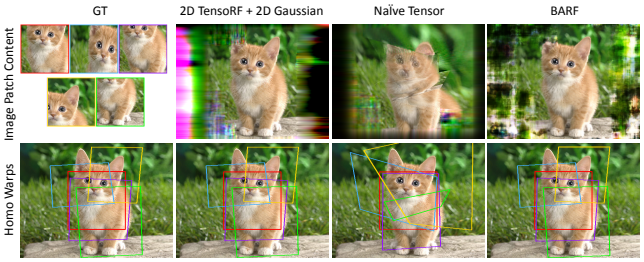

Figure 4: **Qualitative comparisons of the 2D image patch alignment.** *2D TensoRF + 2D Gaussian* successfully registers accurate warping parameters, verifying the analysis of Gaussian filtering on joint optimization.

NeRF (3D): Synthetic Object & Real World Objects. Tab. 2 reports the pose error and novel-view synthesis quality of the NeRF-Synthetic dataset. Our method achieves the smallest pose error in 5 out of 8 scenes and achieves the best reconstruction quality in all eight scenes, and the quantitative results are shown in Fig. 5.

Tab. 3 reports the pose error and novel-view synthesis quality of the LLFF dataset. Our method achieves pose error on par with previous methods and produces the best average view synthesis quality. Our method also scores the best LPIPS in 7 out of 8 scenes, indicating that our method produces perceptually more natural novel-view synthesis.

Note that we achieve state-of-the-art results within only 20% to 25% of training iterations, while all other competing methods train for 200k iterations.

4.2 Ablation Component Analysis

In Tab. 4, we report the effect of each proposed component on the pose error and PSNR of the optimization results. The results are average across all real-world scenes in the LLFF dataset. In (a) (b), we show the effect of *randomly scaled kernel* described in Sec. 3.6. In (b)(c), we show the effectiveness of *edge guided loss* (Sec. 3.6). Finally, in (c)(d)(e), we show the necessity of Gaussian filtering on both 2D supervising images and 3D radiance field represented by a decomposed tensor grid, which validates the analysis in Sec. 3.2.

4.3 Time Complexity

In Fig. 6, we compare with previous methods on average PSNR and training iterations in the *Synthetic NeRF* dataset.

Scene	Camera Pose Registration								View Synthesis Quality											
	Rotation (°) ↓				Translation ↓				PSNR ↑				SSIM ↑				LPIPS ↓			
	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours
Chair	0.113	0.096	**0.085**	0.8743	0.549	0.428	**0.365**	3.501	31.32	31.16	31.95	**35.227**	0.959	0.954	0.962	**0.984**	0.042	0.044	0.036	**0.0125**
Drum	0.052	0.043	0.041	**0.037**	0.232	0.225	0.214	**0.118**	24.15	23.91	24.16	**25.78**	0.909	0.900	0.912	**0.934**	0.097	0.099	0.087	**0.051**
Ficus	0.081	0.085	0.079	**0.050**	0.461	0.474	0.479	**0.173**	26.29	26.26	28.31	**31.37**	0.935	0.934	0.943	**0.978**	0.057	0.058	0.051	**0.014**
Hotdog	0.235	0.248	0.229	**0.105**	1.123	1.308	1.123	**0.499**	34.69	34.54	35.41	**37.18**	0.972	0.970	0.981	**0.982**	0.029	0.032	0.027	**0.013**
Lego	0.101	0.082	0.071	**0.049**	0.299	0.291	0.272	**0.100**	29.29	28.33	31.65	**34.23**	0.925	0.927	0.973	**0.981**	0.051	0.050	0.036	**0.010**
Materials	**0.842**	0.844	0.852	0.854	**2.688**	2.692	2.743	2.69	27.91	27.84	27.14	**29.04**	0.941	0.936	0.911	**0.951**	0.059	0.058	0.062	**0.031**
Mic	0.070	0.071	**0.068**	1.177	0.293	0.301	**0.287**	5.00	31.39	31.18	32.33	**32.50**	0.971	0.969	0.975	**0.976**	0.047	0.048	0.043	**0.024**
Ship	0.073	0.075	0.079	**0.058**	0.310	0.326	0.287	**0.167**	27.64	27.50	27.92	**31.98**	0.862	0.849	0.879	**0.903**	0.119	0.132	0.110	**0.053**
Mean	0.195	0.193	**0.189**	0.400	0.744	0.756	**0.722**	1.533	28.96	28.84	29.86	**32.07**	0.935	0.930	0.943	**0.961**	0.063	0.065	0.056	**0.026**

Table 2: **Quantitative results on the NeRF-Synthetic dataset.** Our method achieves the best average novel-view synthesis quality and the best pose error in 5 out of 8 scenes. Notice that our method converges within 40k iterations, while all previous methods train for 200k iterations.

Scene	Camera Pose Registration								View Synthesis Quality											
	Rotation (°) ↓				Translation ↓				PSNR ↑				SSIM ↑				LPIPS ↓			
	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours
Fern	0.470	0.191	**0.110**	0.472	0.250	**0.102**	**0.102**	0.199	24.51	23.79	24.62	**26.17**	0.740	0.710	0.743	**0.842**	0.290	0.311	0.285	**0.101**
Flower	0.460	**0.251**	0.301	1.375	0.220	0.224	**0.211**	0.389	**26.40**	23.37	25.19	25.62	0.790	0.698	0.744	**0.810**	0.110	0.211	0.128	**0.105**
Fortress	**0.030**	0.479	0.211	0.449	0.270	0.364	**0.241**	0.419	29.09	29.08	**30.14**	29.68	0.820	0.823	**0.901**	0.882	0.150	0.132	0.098	**0.065**
Horns	**0.030**	0.304	0.049	0.386	0.210	0.222	**0.209**	0.251	22.54	22.78	**22.97**	22.84	0.690	0.727	0.736	**0.819**	0.330	0.298	0.290	**0.159**
Leaves	**0.130**	1.272	0.840	1.99	0.230	0.249	**0.228**	0.397	19.72	18.78	19.45	**21.24**	0.610	0.537	0.607	**0.753**	0.270	0.353	0.269	**0.240**
Orchids	0.430	0.627	0.399	**0.279**	0.410	0.404	0.386	**0.340**	19.37	19.45	20.02	**20.57**	0.570	0.574	0.610	**0.698**	0.260	0.291	0.213	**0.141**
Room	0.270	0.320	0.271	**0.188**	0.200	0.270	0.213	0.191	31.90	31.95	**32.73**	31.87	0.940	0.949	**0.968**	0.936	0.130	0.099	**0.098**	0.105
T-Rex	**0.420**	1.138	0.894	0.523	**0.360**	0.720	0.474	0.416	22.86	22.55	23.19	**24.19**	0.800	0.767	0.866	**0.878**	0.190	0.206	0.183	**0.082**
Mean	**0.280**	0.573	0.384	0.709	0.269	0.331	**0.258**	0.325	24.55	23.97	24.79	**25.27**	0.745	0.723	0.772	**0.827**	0.216	0.227	0.197	**0.112**

Table 3: **Quantitative results on the LLFF dataset.** Our method achieves the best average novel-view synthesis quality and best LPIPS in 7 out of 8 scenes. Our method converges within 50k iterations, while all previous methods train for 200k iterations.

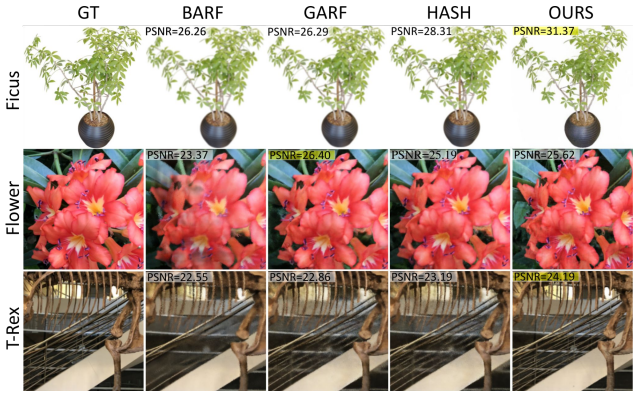

Figure 5: **Visual comparisons of novel view synthesis.**

	3D Gaussian	2D Gaussian	Randomized Kernel	Edge Guided Loss	Rotation (°) ↓	Translation ↓	PSNR ↑
(a)	✓	✓	✓	✓	**0.72**	**0.33**	**25.36**
(b)	✓	✓		✓	1.00	0.371	25.25
(c)	✓	✓			1.91	0.939	25.12
(d)	✓				33.00	12.7	20.10
(e)		✓			26.25	8.9	19.73
(d)					23.29	9.4	23.97

Table 4: **Ablation study of the components of the proposed method on the real-world LLFF dataset.**

The figure shows two advantages of our method: (1) rapid convergence and (2) high-quality novel view synthesis.

The early-stage blurry supervision can hinder detailed structure reconstruction later in the optimization, impacting the final result quality. Our method resolves this problem by applying 3D convolution with directly controllable kernel parameters on a single 3D tensor grid, which enables

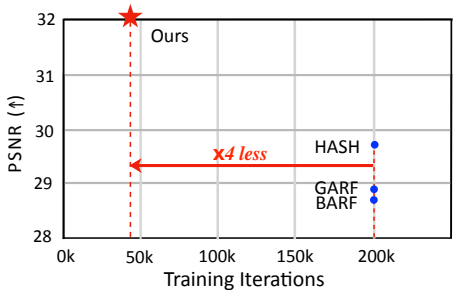

Figure 6: **PSNR and training iterations comparison.**

smooth (by continuously changing the Gaussian kernel parameter) and rapid transition (by exponential kernel schedule) of the 3D content across the coarse-to-fine spectrum domains. Furthermore, using a single tensor volume (as opposed to multiple volumes of different resolution levels as in (Heo et al. 2023)) encourages parameter reuse throughout the kernel schedule, which also supports rapid convergence.

5 Conclusion

We propose an algorithm for the simultaneous refinement of camera poses and scene geometry represented by decomposed low-rank tensor, using only 2D images as supervision. We derive and analyze the frequency property of the 1D signal to show that naïve joint pose optimization on voxel-based NeRFs can easily lead to sub-optimal solutions. Based on the analysis, we propose a separable component-wise convolution mechanism that enables camera pose alignment in 2D and 3D radiance fields. Comprehensive evaluations demonstrate our proposed framework's state-of-the-art performance and rapid convergence without known poses.

References

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chen, Y.; Chen, X.; Wang, X.; Zhang, Q.; Guo, Y.; Shan, Y.; and Wang, F. 2023a. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Z.; Funkhouser, T.; Hedman, P.; and Tagliasacchi, A. 2023b. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chng, S.-F.; Ramasinghe, S.; Sherrah, J.; and Lucey, S. 2022. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance Fields without Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goel, R.; Dhawal, S.; Saini, S.; and Narayanan, P. J. 2022. StyleTRF: Stylizing Tensorial Radiance Fields. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*.

Han, K.; and Xiang, W. 2023. Multiscale Tensor Decomposition and Rendering Equation Encoding for View Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hedman, P.; Srinivasan, P. P.; Mildenhall, B.; Barron, J. T.; and Debevec, P. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Heo, H.; Kim, T.; Lee, J.; Lee, J.; Kim, S.; Kim, H. J.; and Kim, J.-H. 2023. Robust Camera Pose Refinement for Multi-Resolution Hash Encoding. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Hu, T.; Xu, X.; Chu, R.; and Jia, J. 2023. TriVol: Point Cloud Rendering via Triple Volumes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kajiya, J. T.; and Von Herzen, B. P. 1984. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*.

Kanopoulos, N.; Vasanthavada, N.; and Baker, R. L. 1988. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits*.

Kulhanek, J.; and Sattler, T. 2023. Tetra-NeRF: Representing Neural Radiance Fields Using Tetrahedra. *arXiv preprint arXiv:2304.09987*.

Lin, C.-H.; Ma, W.-C.; Torralba, A.; and Lucey, S. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Liu, L.; Gu, J.; Lin, K. Z.; Chua, T.-S.; and Theobalt, C. 2020. Neural Sparse Voxel Fields. *Advances in Neural Information Processing Systems (NeurIPS)*.

Liu, Y.-L.; Gao, C.; Meuleman, A.; Tseng, H.-Y.; Saraf, A.; Kim, C.; Chuang, Y.-Y.; Kopf, J.; and Huang, J.-B. 2023. Robust Dynamic Radiance Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Meuleman, A.; Liu, Y.-L.; Gao, C.; Huang, J.-B.; Kim, C.; Kim, M. H.; and Kopf, J. 2023. Progressively Optimized Local Radiance Fields for Robust View Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (TOG)*.

Schönberger, J. L.; and Frahm, J.-M. 2016. Structure-from-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shao, R.; Zheng, Z.; Tu, H.; Liu, B.; Zhang, H.; and Liu, Y. 2023. Tensor4D: Efficient Neural 4D Decomposition for High-Fidelity Dynamic Reconstruction and Rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, C.; Sun, M.; and Chen, H. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, J.; Chen, X.; Wang, J.; and Zeng, G. 2022. Compressible-Composable NeRF via Rank-residual Decomposition. *Advances in Neural Information Processing Systems*.

Wang, L.; Zhang, J.; Liu, X.; Zhao, F.; Zhang, Y.; Zhang, Y.; Wu, M.; Yu, J.; and Xu, L. 2022. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021. NeRF−−: Neural Radiance Fields Without Known Camera Parameters. *arXiv preprint arXiv:2102.07064*.

Xu, Q.; Xu, Z.; Philip, J.; Bi, S.; Shu, Z.; Sunkavalli, K.; and Neumann, U. 2022. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Y.; Wang, L.; Zhao, X.; Zhang, H.; and Liu, Y. 2023. AvatarMAV: Fast 3D Head Avatar Reconstruction Using Motion-Aware Neural Voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021. PlenOctrees for Real-time Rendering of Neural Radiance Fields. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yüce, G.; Ortiz-Jiménez, G.; Besbinar, B.; and Frossard, P. 2022. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.