KeyGS: A Keyframe-Centric Gaussian Splatting Method for Monocular Image Sequences

Keng-Wei Chang, Zi-Ming Wang, Shang-Hong Lai

National Tsing Hua University
percyx987654321@gapp.nthu.edu.tw, ziming614@gapp.nthu.edu.tw, lai@cs.nthu.edu.tw

Abstract

Reconstructing high-quality 3D models from sparse 2D images has garnered significant attention in computer vision. Recently, **3D Gaussian Splatting (3DGS)** has gained prominence due to its explicit representation with efficient training speed and real-time rendering capabilities. However, existing methods still heavily depend on accurate camera poses for reconstruction. Although some recent approaches attempt to train **3DGS** models without the **Structure-from-Motion (SfM)** preprocessing from monocular video datasets, these methods suffer from prolonged training times, making them impractical for many applications.

In this paper, we present an efficient framework that operates **without any depth or matching model**. Our approach initially uses **SfM** to quickly obtain rough camera poses within seconds, and then refines these poses by leveraging the dense representation in **3DGS**. This framework effectively addresses the issue of long training times. Additionally, we integrate the densification process with joint refinement and propose a **coarse-to-fine frequency-aware densification** to reconstruct different levels of details. This approach prevents camera pose estimation from being trapped in local minima or drifting due to high-frequency signals. Our method significantly reduces training time from hours to minutes while achieving more accurate novel view synthesis and camera pose estimation compared to previous methods.

Introduction

In recent years, 3D photorealistic reconstruction has gained popularity, especially with differential rendering techniques (Kerbl et al. 2023; Mildenhall et al. 2021; Edavamadathil Sivaram, Li, and Ramamoorthi 2024; Wang et al. 2021a; Xu et al. 2022). These methods use a novel approach, representing the 3D model as a differentiable volume field or a traditional representation, and optimize it through a differential rendering pipeline, leading to exceptionally high-quality reconstructions.

Notable representations include **Neural Radiance Fields** (**NeRF**) (Mildenhall et al. 2021) and the recently popular **3D Gaussian Splatting** (**3DGS**) (Kerbl et al. 2023). Both methods use volume rendering (Tagliasacchi and Mildenhall 2022), but they differ significantly in their approaches. **NeRF** employs ray-marching, which leads to slow inference due to the high computational demands of sampling along rays and feeding data to an MLP. In contrast, **3DGS** uses differential rasterization without an MLP, enabling real-time inference speeds.

In many **NeRF** and **3DGS** reconstruction pipelines, a common approach involves using software like **COLMAP** (Schonberger and Frahm 2016) for **Structure from Motion** (**SfM**) to estimate camera poses. **SfM** extracts SIFT features from images, applies the **RANSAC** algorithm for pose estimation, and performs bundle adjustment for refinement. However, this method often struggles under extreme conditions, such as noisy images, textureless regions, low resolution, or varying lighting, leading to inaccurate pose estimates. Moreover, **SfM** becomes computationally expensive with an increasing number of images due to the complexity of **RANSAC** and pairwise bundle adjustment. To overcome these challenges, some researchers focus on refining camera poses during 3D reconstruction or on methods that avoid **SfM** altogether.

To further refine camera pose estimates in **SfM**, various approaches (Lin et al. 2021; Chen, Chiu, and Liu 2024; Jeong et al. 2021; Fu et al. 2023; Liu et al. 2024; Heo et al. 2023; Park et al. 2023; Chen et al. 2023; Shi et al. 2022; Meuleman et al. 2023; Bian et al. 2023; Wang et al. 2021b; Lin et al. 2023; Sucar et al. 2021; Yan et al. 2024; Zhu et al. 2022) have been proposed to jointly refine camera poses, either starting from noisy initial poses or without any initial pose information. **Bundle-Adjusting Neural Radiance Fields (BARF)** (Lin et al. 2021) is the first **NeRF** method to use dense photometric signals for alignment. It also introduces a heuristic **coarse-to-fine strategy** that progressively increases the signal frequency to effectively refine camera poses. **Joint TensoRF** (Chen, Chiu, and Liu 2024) provides theoretical analysis indicating that image alignment may encounter gradient oscillation with high-frequency signals. To address this issue, it employs Gaussian filter to reduce frequency and utilizes a grid-based **NeRF**, **TensoRF** (Chen et al. 2022) as their representation. There are also more extreme approaches that completely avoid using any initial camera pose. **Nope-NeRF** (Bian et al. 2023) considers neighboring sequences and uses a monocular depth estimation model like **DPT** (Ranftl, Bochkovskiy, and Koltun 2021) to minimize reprojection error. Recently, with the

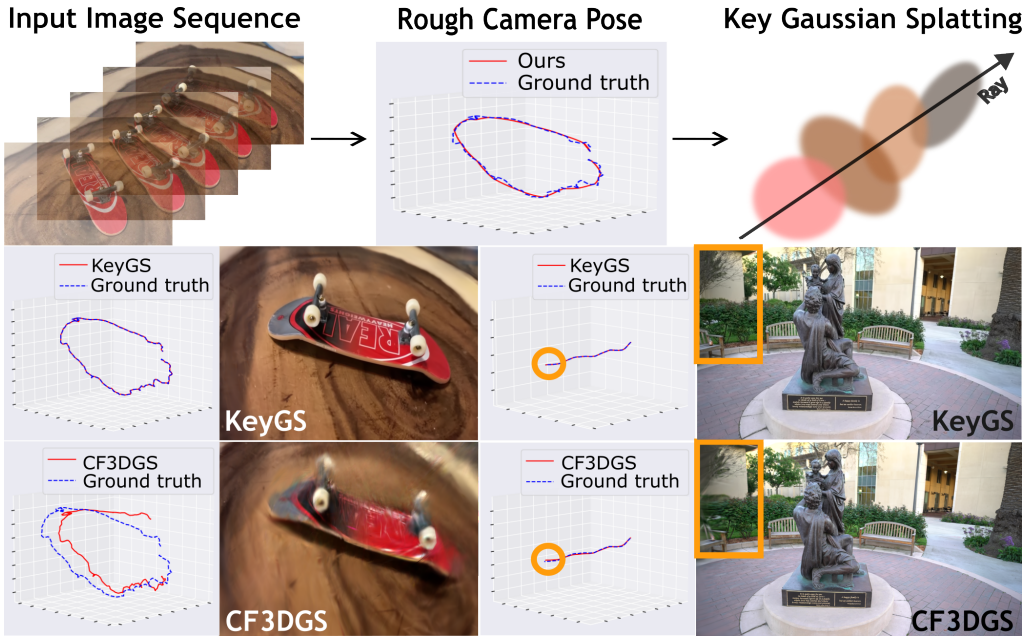


Figure 1: **KeyGS Framework**. For each sequence of images, we sub-sample $\frac{1}{N}$ images as keyframes and perform fast, albeit less accurate, sequential **SFM** with second to obtain an initial rough trajectory. We then jointly optimize the camera poses using the **KeyGS** method. Compared to **CF3DGS**, **KeyGS** continuously refines the camera poses to reduce accumulation errors that lead to localization drift. Additionally, **KeyGS** achieves detailed reconstruction by refining camera poses.

emergence of **3DGS**, **COLMAP-Free 3D Gaussian Splatting (CF3DGS)** (Fu et al. 2023) also uses the **DPT** to predict depth map as a prior to progressively register camera poses, achieving excellent performance.

However, the methods mentioned above suffer from long training times, ranging from hours to days, and NeRF-based approaches are constrained by the ray-marching rendering pipeline. **CF3DGS** employs progressive registration of camera poses and stops refining them after registration, which can lead to inefficient training speeds and accumulated errors, as illustrated by the trajectory in Figure 1.

To address these issues, we propose **KeyGS**, which uses a fast but less accurate **SfM** method to quickly obtain rough camera poses in seconds. **KeyGS** then jointly refines both pose and reconstruction, enabling complete training in approximately 10 minutes. Moreover, our approach continuously refines the camera poses to alleviate the error accumulation problem. In summary, this paper makes the following contributions:

- We propose a framework that combines **Structure-from-Motion** with **3DGS** to efficiently obtain initial rough camera poses and then refine them using **3DGS**, addressing the limitations of traditional pipelines.
- We introduce a joint refinement approach that continuously improves camera poses, mitigating error accumulation and enhancing reconstruction accuracy.
- We develop a **coarse-to-fine frequency-aware densification** technique, which builds a relationship between signal alignment and densification to refine camera poses and reduce artifacts in the reconstruction process.

Related Work

Novel view synthesis

Recent advancements in novel view synthesis have been notably propelled by the development of differential rendering. **NeRF** uses an **MLP** to model both the geometry and view-dependent appearance of scenes. By optimizing through ray marching with **volume rendering** (Tagliasacchi and Mildenhall 2022) techniques, **NeRF** achieves impressive rendering quality.

However, **NeRF** can be very inefficient because it requires a large number of sample points to be fed into a deep **MLP**, using shared weights to represent the entire scene. To improve efficiency, some works (Müller et al. 2022; Chen et al. 2022; Yu et al. 2021; Sun, Sun, and Chen 2022) utilize spatial data structures, such as grids or octrees, to optimize specific learnable features stored in these structures within 3D space. Despite these enhancements, the computationally expensive process of ray marching remains a limitation for wider applications, as it prevents real-time rendering.

To overcome these challenges, **3DGS** introduces a differential rasterization approach that represents 3D models as a set of Gaussian spheres and uses **volume rendering** to blend colors projected from these 3D Gaussians. This shift in the rendering pipeline results in higher inference speeds and a more explicit representation, garnering significant attention across various research domains.

Structure From Motion

Even methods like **NeRF** and **3DGS** typically require preprocessing to obtain accurate camera poses, as these recon-

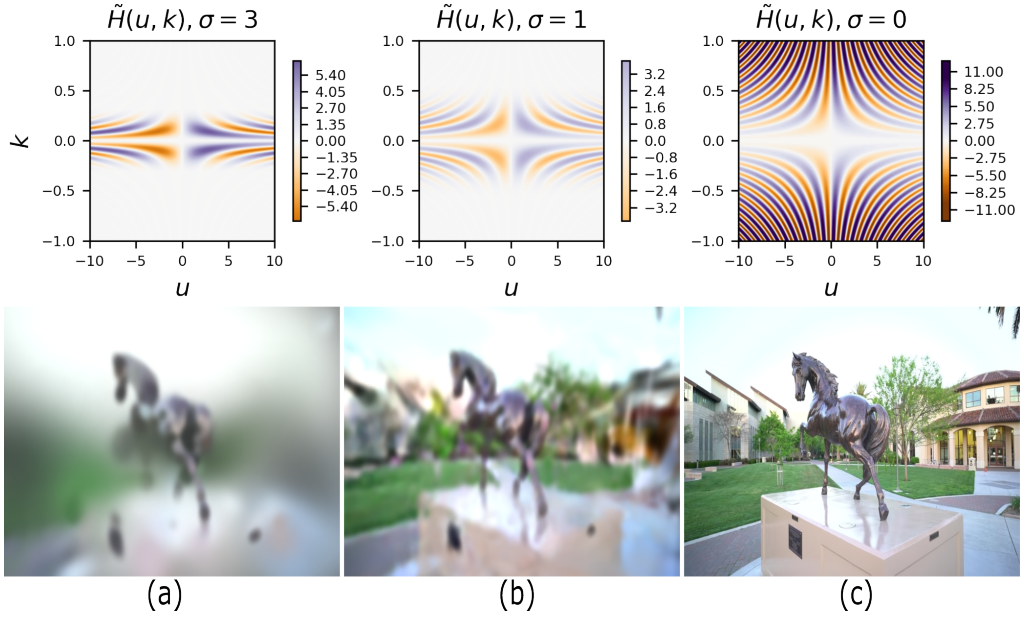


Figure 2: **Illustration of Coarse-to-Fine Frequency-Aware Densification –** The top part shows the gradient for each frequency related to alignment offset, using different scales σ for Gaussian smoothing on the Fourier kernel $\tilde{\mathcal{H}}(u, k)$. Larger σ values concentrate the gradient on low frequencies, while decreasing σ shifts it to higher frequencies. The bottom part visualizes the training process for various σ values. At high σ **(a)**, there are no details. As σ decreases, rough contours emerge **(b)**, and densification is primarily influenced by high-frequency gradients, leading to detailed structures at low σ **(c)**.

struction techniques are highly sensitive to camera pose precision. The most commonly used algorithm for camera pose estimation is **SfM**, with **COLMAP** being one of the most well-known and widely used software tools. It employs an incremental **SfM** method involving three main steps: feature extraction, feature matching with **RANSAC**, and bundle adjustment. Since it uses exhaustive mode to match and perform bundle adjustment on all pairs of images, its computational complexity can be quite high, approximately $O(n^4)$.

However, **COLMAP** offers a sequential matching mode for image sequences, which matches M neighbors and performs bundle adjustment among them, this mode is often less accurate for camera pose estimation. As a result, most tasks do not rely on this mode for registering camera poses with sequential data. In contrast, methods like **Hierarchy GS** (Kerbl et al. 2024) are designed to work at large scales, utilizing sequential data while still opting for the exhaustive mode to obtain more accurate camera poses.

Joint Refinement

Although **COLMAP** can provide nearly accurate camera poses, it can still fail under certain conditions such as noisy images, a limited number of images, or reflective surfaces. To address these issues, several models have been developed to jointly refine camera poses and improve 3D reconstruction accuracy.

BARF (Lin et al. 2021) is among the pioneering models that establish a link between camera pose estimation and 3D reconstruction. While **NeRFmm** (Wang et al. 2021b) refines intrinsic and extrinsic parameters using camera pose embeddings, **NoPeNeRF** (Bian et al. 2023) reconstructs image sequences by leveraging a depth estimator **DPT** (Ranftl, Bochkovskiy, and Koltun 2021) to obtain pseudo-depth information and jointly refines camera poses with reprojection error from neighboring images as regularization.

Similarly, **CF3DGS** (Fu et al. 2023) also utilizes a depth estimator **DPT** to initialize the 3D point cloud, which is crucial step in **3DGS**. However, **CF3DGS**(Fu et al. 2023) fixes the camera poses once they are registered, which can lead to accumulated errors in pose estimation, as shown in Figure 1. Additionally, we observe that the error accumulation problem can cause **CF3DGS**(Fu et al. 2023) to excessively split the Gaussian spheres, resulting in increased memory usage and potential crashes during training due to memory limitations.

Coarse-to-Fine Strategy

BARF (Lin et al. 2021) found that naive joint refinement fails to recover accurate poses due to difficulties in alignment caused by high-frequency components. Instead, they employ a coarse-to-fine strategy to progressively reveal the frequency components of positional encoding in vanilla **NeRF** (Mildenhall et al. 2021), which enables effective joint refinement of camera poses and scene reconstruction.

Although **BARF** (Lin et al. 2021) experimentally validates the effectiveness of the coarse-to-fine strategy, it is limited to vanilla **NeRF** (Mildenhall et al. 2021). **Joint TensoRF** (Chen, Chiu, and Liu 2024) is a **NeRF** model that focuses on joint refinement by utilizing grid-based **NeRF** (Chen et al. 2022) for reconstruction. Previous works (Heo

et al. 2023; Liu et al. 2024) have shown that grid features can be unsuitable for joint refinement of camera poses due to discretization and gradient oscillations. However, **Joint TensoRF** (Chen, Chiu, and Liu 2024) addresses this issue by providing general theoretical analysis and solution of joint refinement. They use a 1D signal f_{gt}, as an example. To align two 1D signals with a shift offset u, the gradient related to offset u is given by Equation 1:

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

where $\mathcal{H}(u,k) = 4\pi k \sin(2\pi k u)$, $\mathcal{F}[f_{gt}]$ is the Fourier transform of f_{gt}, and k is the wavenumber in the frequency domain. Intuitively, the kernel $\mathcal{H}(u,k)$ transforms the spectrum $\mathcal{F}[f_{gt}]$ into the derivative $\frac{d}{du}\mathcal{L}$. The kernel $\mathcal{H}(u,k)$ exhibits gradient oscillations at high frequencies and is quasi-convex at low frequencies. To address the challenges posed by high frequencies, the authors apply **coarse-to-fine** Gaussian smoothing to the signal. Consequently, the transform kernel $\mathcal{H}(u,k)$ becomes $\tilde{\mathcal{H}}(u,k) = \|\mathcal{F}[\mathcal{N}(0,\sigma^2)]\|^2 \cdot \mathcal{H}(u,k)$, which enhances the effectiveness of joint refinement in voxel-based NeRF, **TensoRF**. For a visual representation, we refer the reader to Figure 2. In summary, while existing methods aim to refine camera poses and use sequential image structures, they often suffer from long training times due to inefficient pipelines. This limits their application for real-world uses. Our work overcomes these challenges with the proposed **KeyGS**, an efficient framework leveraging **3DGS**. We also develop a **coarse-to-fine frequency-aware densification** strategy, inspired by **Joint TensorRF** (Chen, Chiu, and Liu 2024), for effective and practical camera pose refinement.

Proposed Method

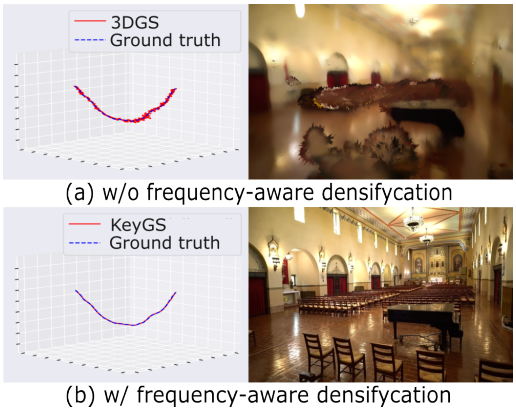
Given a sequence of images $\{I_i^{gt}\}_{i=1}^N$, our goal is to efficiently recover camera poses $\{T_i\}_{i=1}^N$ and generate a photorealistic 3D scene, simultaneously. We propose **KeyGS**, a framework designed to quickly refine initial rough camera poses. It then jointly reconstructs the scene and refines the camera poses using **3DGS** along with our coarse-to-fine frequency-aware densification technique.

Preliminary: 3D Gaussian Splatting

3DGS (Kerbl et al. 2023) represents a scene using explicit 3D Gaussians, unlike **NeRF**'s implicit representation. It uses rasterization to project the 3D Gaussians onto the image plane and applies volume rendering to blend colors. This approach is efficient for inference due to its active projection and the absence of MLP involvement.

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{2}$$

However, **3DGS** requires appropriate initial conditions, often utilizing a point cloud estimated by SfM with position $\mu \in \mathbb{R}^3$, color $c \in \mathbb{R}^3$ (parameterized by spherical harmonics), and initialized with low opacity $o \in \mathbb{R}^1$. In order to represent the 3D Gaussians in Equation 2, it employs **KNN** for estimating the covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$. During optimization, Σ is decomposed into a rotation $R \in \mathbb{R}^{3\times3}$ and a scale $S \in \mathbb{R}^{3\times3}$ to ensure it remains positive semi-definite, as shown in Equation 3.



(a) w/o frequency-aware densifycation

(b) w/ frequency-aware densifycation

Figure 3: **Comparison of naive joint refinement and our proposed method, frequency-aware densification.** (a) Applying naive joint refinement to **3DGS** results in over-splitting of the Gaussians to fit high-frequency signals. This causes the Gaussians to become spiky, making alignment more difficult and leading to oscillations in the trajectory. (b) Our proposed frequency-aware densification method uses a coarse-to-fine approach to account for gradients of different frequencies. The reconstruction results are smoother and more accurate, leading to improved camera pose recovery.

$$\Sigma = RSS^TR^T \tag{3}$$

To represent the scene using 3D Gaussians, rasterization is employed to render them as images. Given the camera view $W \in SE(3)$ and camera intrinsic $K \in \mathbb{R}^{3\times4}$, the 3D Gaussians are projected onto the image plane. This projection involves using a Taylor expansion to approximate the 3D Gaussians as 2D Gaussians on the plane. The projected mean $\mu^{2D} \in \mathbb{R}^2$ is computed by applying the camera view W and the camera intrinsic matrix K to μ. Meanwhile, the covariance $\Sigma^{2D} \in \mathbb{R}^{2\times2}$ is determined by using the Jacobian J of the camera projection, as shown in Equation 4.

$$\Sigma^{2D} = JW\Sigma W^TJ^T \tag{4}$$

Finally, using volume rendering as described in Equation 5, the colors c_i of each Gaussian are blended sequentially based on their depth, with the alpha value $\alpha_i(x) = o_i(x)G_i(x)$ associated with each Gaussian. This process results in the rendered output $\hat{C}(x)$, which is then compared to the ground truth $C^{gt}(x)$ to compute the loss.

$$\hat{C}(x) = \sum_{i=1}^{N} c_i\alpha_i(x)\prod_{j=1}^{i-1}(1-\alpha_j(x)) \tag{5}$$

KeyGS Framework

To address the efficiency issues related to the time cost of feature matching and bundle adjustment in the **SfM** problem, we use the sequential mode of **COLMAP** instead of the exhaustive mode to match features within a small neighborhood of images. This approach reduces the time complexity

from $O(n^4)$ to $O(n^2)$, significantly lowering the computational cost of **SfM**. Furthermore, since the cost is highly dependent on the number of images, we leverage the sequential structure of the image set by uniformly subsampling $\frac{1}{N}$ of the images in the sequence and applying sequential **SfM** to these keyframes. After applying **SfM**, we obtain the camera poses represented by quaternions $q_i \in \mathbb{R}^4$ and translations. For the remaining unsampled images, we perform **spherical linear interpolation (SLERP)** for quaternions (Equation 6) and linear interpolation for translations. This framework efficiently estimates rough camera poses in seconds for the trajectory, as illustrated in Figure 1.

$$Slerp(q_i, q_{i+1}, t) = \frac{\sin(1-t)\theta}{\sin(\theta)} q_i + \frac{\sin(t\theta)}{\sin(\theta)} q_{i+1} \tag{6}$$

Once the rough trajectory is obtained, we employ the **KeyGS** with our **frequency-aware densification** to the joint refinement process, utilizing dense RGB information to further refine the camera poses. This step ensures that the initial estimations are improved upon, allowing for a more accurate reconstruction of the scene.

Joint Refinement and Densification

To refine the camera pose $T_i \in SE(3)$ for each image I_i, we train **3DGS** with a learnable $SE(3)$ transformation ΔT_i to obtain the estimated pose $T_i^{\text{pred}} = \Delta T_i T_i$, Specifically, we do not parameterize ΔT_i using $\mathfrak{se}(3)$; instead, we use $\mathfrak{so}(3) \times \mathfrak{t}(3)$ to reduce the influence between rotation and translation. The image is rendered using rasterization from 3DGS, which depends on the predicted camera pose T_i^{pred} and the optimized Gaussians G. We minimize the photometric loss function $\mathcal{L}_{\text{rgb}}$, which integrates $\mathcal{L}_1$ and D-SSIM losses , balanced by a factor λ, as detailed in Equation 7. The attributes of the Gaussians and the pose refinements ΔT_i are updated through backpropagation.

$$\mathcal{L}_{\text{rgb}} = (1-\lambda)\mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}} \tag{7}$$

Another important factor in joint refinement is the densification process. In **3DGS**, Gaussians are split if the gradient with respect to their position μ^{2D}, $\frac{\partial L}{\partial \mu^{2D}}$, exceeds a certain threshold, and they are culled if their opacity, o, falls below a specified threshold. Both the refinement and densification processes are primarily influenced by the gradient term $\frac{\partial \mathcal{L}}{\partial G}\frac{\partial G}{\partial \mu^{2D}}$, as described in Equation 8. Notably, only the 2D splat affects these processes.

$$\frac{\partial L}{\partial \Delta T} = \frac{\partial L}{\partial G}\frac{\partial G}{\partial \mu^{2D}}\frac{\partial \mu^{2D}}{\partial \Delta T} \tag{8}$$

As shown in Figure 2 (c), high-frequency signals can lead to misplaced Gaussians, causing excessive splitting due to larger, oscillating gradients. Excessive splitting results in more misaligned Gaussians and an increases probability of trapping in local minima due to overfitting with incorrect poses, which complicate the optimization process and creating a vicious cycle. We depict an example for visualization in Figure 3.

Figure 4: **Coarse-to-Fine Frequency-Aware Densification**. Our method aligns signals by preventing premature Gaussian splitting at high frequencies. Although the signal may appear aligned early on, this approach suppresses gradient influence from high-frequency details. As the Gaussian filter scale decreases, the gradient shifts to high frequencies, enabling Gaussians to split and capture finer details.

Coarse To Fine Frequency-Aware Densificaion

To address gradient oscillation and excessive splitting issues, we propose a **coarse-to-fine frequency-aware** densification. First, we visualize the kernel $\tilde{\mathcal{H}}(u,k)$, as mentioned in related work, with varying scales σ of Gaussian blur in Figure 2. We observe that Gaussian smoothing not only suppresses high-frequency gradients but also adjusts the dominant gradient to different frequency levels as the scale σ changes. This approach is particularly effective for densification with misaligned Gaussians, as it provides stable gradients for low frequencies, preventing excessive splitting and enabling effective signal alignment.

$$\begin{aligned} \frac{d}{du}\mathcal{L} &= \int \|\mathcal{F}[f_{gt}]\|^2 \tilde{\mathcal{H}}(u,k)\, dk \\ &= \int \|\sum_i^N a_i \mathcal{F}[\mathcal{N}(0,\sigma^2) * G_i]\|^2 \mathcal{H}(u,k)\, dk \end{aligned} \tag{9}$$

We analyze how Gaussian blur affects the gradient with 3DGS, as described in Equation 9. Since the signal is represented by the blending of Gaussians using Equation 5, it can be viewed as linear combination of Gaussians. By leveraging the linear property of the Fourier transform, we find that applying a smooth gradient operator is equivalent to applying Gaussian blur to each Gaussian G_i.

$$G_i(x,\sigma) = \sqrt{\frac{|\Sigma|}{|\Sigma+\sigma^2 I|}} e^{-\frac{1}{2}(x-\mu)^T(\Sigma+\sigma^2 I)^{-1}(x-\mu)} \tag{10}$$

Therefore, we apply a Gaussian filter to each Gaussian G_i as defined in Equation 10 . Initially, a large-scale filter suppresses gradient influence from high-frequency details. As training progresses, we gradually reduce the filter scale σ, enhancing the ability to capture fine details. This ensures effective signal alignment. as shown in Figure 4.

Regularization

In Figure 3, we observe that Gaussians tend to fit high-frequency signals, resulting in a spiky appearance. Similar to **PhysGaussian** (Xie et al. 2024), we apply anisotropy regularization, as described in Equation 11, to control the ratio between the maximum axis $\max(S_i)$ and minimum axis $\min(S_i)$ of the Gaussians, preventing them from becoming spiky. Here, r represents the minimum ratio. Furthermore, to

Scenes	KeyGS (ours)			CF3DGS			Nope-NeRF			BARF			SC-NeRF		
	PSNR↑	SSIM↑	LPIPS↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Church	**30.62**	**0.92**	**0.06**	30.23	0.93	0.11	25.17	0.73	0.39	23.17	0.62	0.52	21.96	0.60	0.53
Barn	**34.25**	**0.95**	**0.04**	31.23	0.90	0.10	26.35	0.69	0.44	25.28	0.64	0.48	23.26	0.62	0.51
Museum	**33.46**	**0.94**	**0.03**	29.91	0.91	0.11	26.77	0.76	0.35	23.58	0.61	0.55	24.94	0.69	0.45
Family	**33.05**	**0.95**	**0.04**	31.27	0.94	0.07	26.01	0.74	0.41	23.04	0.61	0.56	22.60	0.63	0.51
Horse	33.65	**0.96**	**0.03**	**33.94**	**0.96**	0.05	27.64	0.84	0.26	24.09	0.72	0.41	25.23	0.76	0.37
Ballroom	**33.70**	0.95	**0.02**	32.47	**0.96**	0.07	25.33	0.72	0.38	20.66	0.50	0.60	22.64	0.61	0.48
Francis	**34.45**	**0.93**	**0.08**	32.72	0.91	0.14	29.48	0.80	0.38	25.85	0.69	0.57	26.46	0.73	0.49
Ignatius	**30.85**	**0.92**	**0.06**	28.43	0.90	0.09	23.96	0.61	0.47	21.78	0.47	0.60	23.00	0.55	0.53
Mean	**33.01**	**0.94**	**0.04**	31.28	0.93	0.09	26.34	0.74	0.39	23.42	0.61	0.54	23.76	0.65	0.48

Table 1: **Novel view synthesis results on Tanks and Temples.** Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. The best results are highlighted in bold.

Scenes	KeyGS (ours)			CF3DGS			Nope-NeRF			BARF			SC-NeRF		
	RPE_t↓	RPE_r↓	ATE↓	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE
Church	**0.006**	0.013	**0.000**	0.008	0.018	0.002	0.034	**0.008**	0.008	0.114	0.038	0.052	0.836	0.187	0.108
Barn	**0.008**	**0.016**	**0.001**	0.034	0.034	0.003	0.046	0.032	0.004	0.314	0.265	0.050	1.317	0.429	0.157
Museum	**0.025**	**0.025**	**0.002**	0.052	0.215	0.005	0.207	0.202	0.020	3.442	1.128	0.263	8.339	1.491	0.316
Family	**0.012**	**0.012**	**0.000**	0.022	0.024	0.002	0.047	0.015	0.001	1.371	0.591	0.115	1.171	0.499	0.142
Horse	**0.078**	**0.002**	**0.001**	0.112	0.057	0.003	0.179	0.017	0.003	1.333	0.394	0.014	1.336	0.438	0.019
Ballroom	**0.015**	**0.014**	**0.000**	0.037	0.024	0.003	0.041	0.018	0.002	0.531	0.228	0.018	0.328	0.146	0.012
Francis	**0.007**	0.016	**0.001**	0.029	0.154	0.006	0.057	**0.009**	0.005	1.321	0.558	0.082	1.233	0.483	0.192
Ignatius	**0.001**	0.010	**0.001**	0.033	0.032	0.005	0.026	**0.005**	0.002	0.736	0.324	0.029	0.533	0.240	0.085
Mean	**0.020**	**0.015**	**0.000**	0.041	0.069	0.004	0.080	0.038	0.006	1.046	0.441	0.078	1.735	0.477	0.123

Table 2: **Pose accuracy on Tanks and Temples**. COLMAP poses are used as ground truth with all methods evaluated using the same protocol. Units: RPE_r (degrees), ATE (ground truth scale), RPE_t (scaled by 100). Best results are highlighted in bold.

prevent Gaussians from being trapped in local minima due to gradient oscillations, we draw inspiration from **AbsGS** (Ye et al. 2024) and use the absolute gradient $|\frac{\partial L}{\partial \mu^{2D}}|$ to encourage Gaussian splitting, allowing them to search for a better position.

$$\mathcal{L}_{\text{aniso}} = \frac{1}{N}\sum_{i=1}^{N} \max\left\{\frac{\max(S_i)}{\min(S_i)}, r\right\} - r \tag{11}$$

Experiments

In this section, we compare our approach with existing joint refinement models: **BARF** , **CF3DGS** , **Nope-NeRF** , and **SC-NeRF** , using **the Tanks and Temples** and **CO3DV2** datasets. We also conduct an ablation study to highlight key components of our method. Moreover, we show that our method outperforms **3DGS** , even when it uses camera pose estimates from **COLMAP** , which is often regarded as ground truth to evaluate the effectiveness of pose estimation. Additional experimental results are provided in the supplementary material.

Data Preprocessing

COLMAP is the most commonly used tool for registering camera poses with **SfM**. The major computational cost arises from bundle adjustment due to the large number of images and feature points, as described in (Schonberger and Frahm 2016). We analyze the average computation time for different keyframe subsampling intervals and various downsampling resolutions using all images, as shown in Figure 5. Both options offer significant speedups compared to the exhaustive mode. However, we found that downsampling image resolution can fail in some scenes because low-resolution images lack robust feature points. Although a keyframe subsampling interval of 10 speeds up the process by over 50 times, it is not robust for outdoor scenes. Therefore, to ensure stability and obtain accurate camera poses, we use a keyframe subsample interval of 5 with full resolution in our experimental setting, it also speeds up the process by 10 to 20 times.

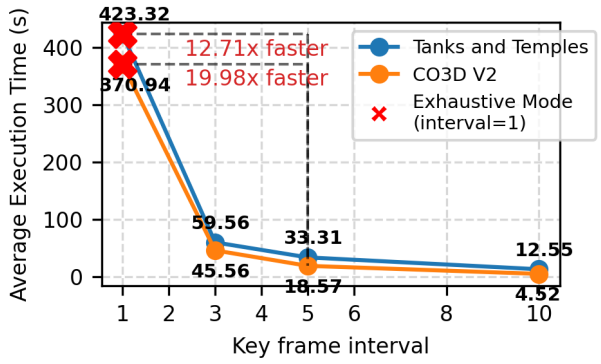


Figure 5: **KeyFrame-Centric SfM**. Our data preprocessing method can achieve a speedup of at least **10** times when using the sequential mode compared to the exhaustive mode with full images in **COLMAP**.

Scenes	KeyGS (Ours)						Nope-NeRF						CF3DGS					
	PSNR	SSIM	LPIPS	RPE_t	RPE_r	ATE	PSNR	SSIM	LPIPS	RPE_t	RPE_r	ATE	PSNR	SSIM	LPIPS	RPE_t	RPE_r	ATE
Apple	**33.53**	**0.94**	**0.07**	**0.026**	**0.116**	**0.001**	26.86	0.73	0.47	0.400	1.966	0.046	29.69	0.89	0.29	0.140	0.401	0.021
Bench	**26.35**	**0.73**	**0.30**	**0.060**	**0.332**	**0.002**	24.78	0.64	0.55	0.326	1.919	0.054	26.21	**0.73**	0.32	0.110	0.424	0.014
Hydrant	**25.33**	**0.80**	**0.15**	**0.005**	**0.042**	**0.000**	20.41	0.46	0.58	0.387	1.312	0.049	22.14	0.64	0.34	0.094	0.360	0.008
Skateboard	**32.74**	**0.93**	**0.16**	**0.029**	**0.165**	**0.001**	25.05	0.80	0.49	0.587	1.867	0.038	27.24	0.85	0.30	0.239	0.472	0.017
Teddybear	**32.67**	**0.93**	**0.09**	**0.037**	**0.120**	**0.001**	28.62	0.80	0.35	0.591	1.313	0.053	27.75	0.86	0.20	0.505	0.211	0.009
Average	**30.12**	**0.87**	**0.15**	**0.031**	**0.155**	**0.001**	25.14	0.68	0.48	0.458	0.771	1.291	26.32	0.77	0.31	0.217	0.269	0.297
Time	**10 min.**						30 hr.						2 hr.					

Table 3: **Novel view synthesis and pose accuracy on CO3D V2**. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. And COLMAP poses are used as ground truth pose. Best results are highlighted in bold.

Figure 6: **3DGS vs. KeyGS.** We show that **COLMAP** may register noisy camera poses, which can lead to failures in detailed regions, such as the tree in the Horse scene, when using **3DGS**. In contrast, our method, **KeyGS**, effectively addresses these issues through joint refinement, even when starting with rough camera poses.

Results

We use PSNR, SSIM, and LPIPS to evaluate the reconstruction results in our experiments. For camera pose accuracy, we evaluate it with RPE_t , RPE_r and ATE. More detailed evaluation metrics are given in the supplementary materials.

For the **Tanks and Temples** dataset, as shown in Tables 1 and 2, our method provides very competitive performance and achieves impressively accurate trajectories. Notably, our average **PSNR** exceeds that of **CF3DGS** by **2 dB**.

For the **CO3DV2** dataset, we followed the experimental setup in **CF3DGS**, evaluating the same five selected sequences and presenting the results in Tables 3. This dataset is more challenging due to complex trajectories and blurred images. Our method achieves a **4 dB** higher average **PSNR** than others and significantly reduces the training time cost. Moreover, our method can continuously refine camera poses, which helps prevent the accumulation of trajectory errors compared to **CF3DGS**, as illustrated in Figure 1.

Ablation Study

First, we highlight the most significant component of our strategy, as shown in Table 4. The results demonstrate that the **coarse-to-fine frequency-densification** is the core component of our method. Additionally, both the absolute gradient and anisotropy regularization further improve the performance of our approach.

To demonstrate the importance of joint refinement for camera poses, we compare our method (using a keyframe subsample interval of 5) with **3DGS** trained using ground truth camera poses from the Tanks and Temples dataset, without joint refinement. The results are shown in Table 5, and an example is depicted in Figure 6. Our method outperforms the **3DGS** with the camera poses provided by the Tanks and Temples dataset. This experiment highlights that even when the exhaustive mode in **COLMAP** is used for pose estimation, inaccuracies in COLMAP-generated camera poses can still degrade **3DGS** performance.

	C2F	**Abs.**	**Aniso Reg.**	PSNR↑	RPE_t↓	RPE_r↓
(a)	✓	✓	✓	**33.01**	**0.020**	**0.015**
(b)	✓	✓		32.94	0.020	0.015
(c)	✓		✓	31.17	0.020	0.015
(d)		✓	✓	18.13	0.935	3.022
(e)	✓			31.66	0.020	0.015
(f)				19.70	0.534	2.143

Table 4: Ablation study of the components of the proposed method on the Tanks and Temples dataset.

Setting	PSNR↑	SSIM↑	LPIPS↓
3DGS w/ COLMAP pose	30.77	0.91	0.10
Ours (rough pose + joint refine)	**33.01**	**0.94**	**0.04**

Table 5: Ablation study on leveraging exhaustive mode in COLMAP for camera pose estimation and joint refinement within the KeyGS framework.

Conclusion

In this paper, we presented **KeyGS**, an efficient framework for joint refinement of camera poses and reconstruction for monocular image sequences. We analyzed the relationship between **densification** and **joint refinement** and proposed the **coarse-to-fine frequency-aware densification** approach to address gradient oscillation from high-frequency signals. Our approach significantly outperforms previous methods with more accurate reconstruction and camera pose estimation as well as drastically reduced training times.

Acknowledgements

This work was supported in part by the National Science and Technology Council, Taiwan under grants NSTC 111-2221-E-007-106-MY3 and NSTC 113-2634-F-007 002.

References

Bian, W.; Wang, Z.; Li, K.; Bian, J.-W.; and Prisacariu, V. A. 2023. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4160–4169.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, 333–350. Springer.

Chen, B.-Y.; Chiu, W.-C.; and Liu, Y.-L. 2024. Improving Robustness for Joint Optimization of Camera Pose and Decomposed Low-Rank Tensorial Radiance Fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 990–1000.

Chen, Y.; Chen, X.; Wang, X.; Zhang, Q.; Guo, Y.; Shan, Y.; and Wang, F. 2023. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8264–8273.

Edavamadathil Sivaram, V.; Li, T.-M.; and Ramamoorthi, R. 2024. Neural Geometry Fields For Meshes. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Fu, Y.; Liu, S.; Kulkarni, A.; Kautz, J.; Efros, A. A.; and Wang, X. 2023. COLMAP-Free 3D Gaussian Splatting.

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 *International Conference on Machine Learning*, 13000–13016. PMLR.

Jeong, Y.; Ahn, S.; Choy, C.; Anandkumar, A.; Cho, M.; and Park, J. 2021. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5846–5854.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. volume 42, 139–1.

Kerbl, B.; Meuleman, A.; Kopanas, G.; Wimmer, M.; Lanvin, A.; and Drettakis, G. 2024. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. volume 43, 1–15. ACM New York, NY, USA.

Lin, C.-H.; Ma, W.-C.; Torralba, A.; and Lucey, S. 2021. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5741–5751.

Lin, Y.; Müller, T.; Tremblay, J.; Wen, B.; Tyree, S.; Evans, A.; Vela, P. A.; and Birchfield, S. 2023. Parallel inversion of neural radiance fields for robust pose estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 9377–9384. IEEE.

Liu, S.; Lin, S.; Lu, J.; Supikov, A.; and Yip, M. 2024. Baangp: Bundle-adjusting accelerated neural graphics primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 850–857.

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/CVF Conference on Computer Vision and Pattern Recognition*, 16539–16548.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. volume 65, 99–106. ACM New York, NY, USA.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. volume 41, 1–15. ACM New York, NY, USA.

Park, K.; Henzler, P.; Mildenhall, B.; Barron, J. T.; and Martin-Brualla, R. 2023. Camp: Camera preconditioning for neural radiance fields. volume 42, 1–11. ACM New York, NY, USA.

Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12179–12188.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Shi, Y.; Rong, D.; Ni, B.; Chen, C.; and Zhang, W. 2022. Garf: Geometry-aware generalized neural radiance field.

Sucar, E.; Liu, S.; Ortiz, J.; and Davison, A. J. 2021. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6229–6238.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5459–5469.

Tagliasacchi, A.; and Mildenhall, B. 2022. Volume rendering digest (for nerf).

Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; and Wang, W. 2021a. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction.

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021b. NeRF–: Neural radiance fields without known camera parameters.

Xie, T.; Zong, Z.; Qiu, Y.; Li, X.; Feng, Y.; Yang, Y.; and Jiang, C. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4389–4398.

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/CVF conference on computer vision and pattern recognition*, 5438–5448.

Yan, C.; Qu, D.; Xu, D.; Zhao, B.; Wang, Z.; Wang, D.; and Li, X. 2024. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19595–19604.

Ye, Z.; Li, W.; Liu, S.; Qiao, P.; and Dou, Y. 2024. AbsGS: Recovering fine details in 3D Gaussian Splatting. In *ACM Multimedia 2024*.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5752–5761.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12786–12796.