Bundle Adjustment for 360° Pose Correction

Taru Rustagi Kaustav Mukherjee
{trustagi, kaustavm}@andrew.cmu.edu

August 24, 2026

Abstract

Many modern 3D reconstruction methods, such as NeRFs and Gaussian Splatting, have made rapid progress in reconstruction quality, but rely heavily on pixel-perfect camera poses so that photometric supervision aligns during joint training. While handheld phones and RGB-D cameras have received significant attention, less work has been done on optimizing poses from 360° video, which offers high omnidirectional overlap which enables more robust tracking than other capture methods in narrow or low-texture environments. In this project, we introduce a novel, standalone, offline pose optimizer pipeline for 360° imagery that takes coarse trajectory inputs from SLAM algorithms as initialization, and performs bundle adjustment with photometric and gaussian-splatting-aware objectives to output refined, subpixel-accurate poses that measurably boost gaussian splatting reconstruction quality. We run experiments that demonstrate, both qualitatively and quantitatively, the improvements to gaussian splats created after refining poses with our pipeline, using both self-captured data and online datasets.

1 Introduction and Problem Definition

3D reconstruction is an increasingly useful tool across a variety of industries, from entertainment, facility management, gaming, and more. One of the key techniques that have emerged in the past years is Gaussian Splatting [4], which allows for effective 3D reconstruction and for novel view synthesis from

2 Introduction

3D reconstruction is an increasingly useful tool across a variety of industries, from entertainment and facility management to gaming and virtual reality. One of the key techniques that has emerged in recent years is Gaussian Splatting [4], which allows for effective 3D reconstruction and novel view synthesis from multi-view imagery. Many modern 3D methods, including Neural Radiance Fields (NeRFs) [8] and Gaussian Splatting, have made rapid progress in reconstruction quality. However, they rely heavily on pixel-perfect camera poses so that photometric supervision aligns properly during joint training.

While handheld phones and advanced RGB-D cameras have spurred extensive work on pose estimation for these pipelines, far less attention has been paid to optimizing poses from a single 360° video. 360° capture offers high omnidirectional overlap, enabling robust tracking even in narrow or low-texture environments. This can also result in faster capture times in both easy and hard-to-reach areas, as empirically shown by Janiszewski et al. [3]. Existing open-source 360° SLAM systems (e.g., OpenVSLAM/Stella VSLAM [11]) primarily deliver coarse trajectories and perform bundle adjustment mainly for loop closures. They do not explicitly target subpixel photometric alignment, which can degrade Gaussian Splatting reconstruction quality.

Another motivation for this work is the wide availability of monocular 360° cameras. Brands like Ricoh and Insta360 (e.g., Ricoh Theta Z1, Insta360 X5) continue to release consumer devices with steadily improving hardware and ever-higher image resolutions. However, these cameras typically lack additional sensing modalities that would make per-pixel SLAM more robust (e.g., depth, odometry, IMU-grade calibration beyond basic metadata). Because 360° cameras offer broad scene coverage with minimal user effort, a pose optimizer tailored to their characteristics can substantially improve 360°-based Gaussian Splatting pipelines, especially by refining coarse trajectories into photometrically consistent, subpixel-accurate poses for better reconstruction quality.

We propose a standalone, offline pose optimizer for 360° imagery (cubemap or equirectangular/ERP) that ingests coarse poses from a SLAM system like OpenVSLAM and performs bundle adjustment explicitly optimized for Gaussian Splatting fidelity. Our optimizer incorporates photometric and GS-aware objectives to achieve subpixel color alignment consistent with GS rendering assumptions. It handles non-loop segments where graph-based loop closure offers limited leverage and incorporates projection-aware residuals for ERP/cubemap representations. The output is refined, subpixel-accurate poses that measurably boost GS reconstruction quality.

3 Related Work

3.1 360° SLAM Systems

OpenVSLAM [11] was among the first open-source SLAM packages to popularize the use of equirectangular (360°) images in SLAM pipelines. It accommodates various camera models, including perspective, fisheye, and equirectangular projections. The system is optimized for real-time localization and mapping, emphasizing bundle adjustment primarily for loop closure detection and graph correction. Stella VSLAM is an open-source fork that continues development of OpenVSLAM.

These systems prioritize computational efficiency and real-time performance, making them well-suited for robotic navigation and augmented reality applications. However, their bundle adjustment objectives focus on maintaining metric consistency and correcting drift through loop closures rather than achieving the dense photometric alignment required for high-quality novel view synthesis. The loss functions are not specifically tuned for minimizing photometric residuals across dense pixel regions, which is critical when camera poses will be used for methods like Gaussian Splatting or NeRF that rely on precise pixel-level alignment. Additionally, bundle adjustment in these systems is often applied selectively (e.g., only at loop closures or keyframes) rather than comprehensively across all frames, potentially leaving residual pose errors in non-loop segments of trajectories.

3.2 Learned Feature Detection and Matching

Traditional feature detectors like SIFT [7] have been the foundation of Structure-from-Motion pipelines for decades. SIFT provides scale and rotation invariance through a carefully designed hand-crafted pipeline involving Difference-of-Gaussians detection and gradient-based descriptors. While robust in many scenarios, SIFT can struggle with challenging conditions such as significant illumination changes, repetitive patterns, low-texture regions, or extreme viewpoint variations. Feature matching is typically performed using nearest neighbor search with ratio tests, which can produce many false matches that must be filtered with robust estimation techniques like RANSAC.

Recent learned approaches have demonstrated substantial improvements in both detection and matching. SuperPoint [1] introduced a self-supervised framework for training interest point detectors and descriptors. Unlike patch-based neural networks that operate on small image crops, SuperPoint is a fully-convolutional model that processes full-sized images and jointly computes pixel-level interest point locations and associated descriptors in one forward pass. The method uses Homographic Adaptation, where a network is first trained on synthetic shapes with ground truth corner locations (MagicPoint), then iteratively refined on real images by generating pseudo-labels through homographic warping. This self-supervised approach enables the detector to learn robust, repeatable features without manual annotation and has shown improved repeatability compared to classical methods on standard benchmarks.

SuperGlue [9] reframes feature matching as a differentiable optimal transport problem solved via graph neural networks. Rather than independently comparing feature descriptors through nearest neighbor matching, SuperGlue uses an attentional graph neural network to perform context aggregation within and across both images. The network models both self-attention (relationships between features within the same image) and cross-attention (relationships between features across images), enabling it to reason jointly about the underlying 3D scene geometry and feature correspondences. SuperGlue solves the assignment as an optimal matching problem using the Sinkhorn algorithm, incorporating a "dustbin" concept to reject unmatchable keypoints. This approach has demonstrated significantly higher precision and recall in feature matching, particularly in wide-baseline scenarios, achieving state-of-the-art results in relative pose estimation on indoor and outdoor datasets.

LightGlue [6] builds upon SuperGlue with several key improvements. It revisits multiple design decisions of SuperGlue and introduces architectural changes that improve both efficiency and accuracy. The most significant innovation

is adaptive computation: LightGlue dynamically adjusts its computational complexity based on the difficulty of the matching problem. For image pairs with large visual overlap or limited appearance change (common in video sequences), LightGlue can terminate early at shallower network layers when predictions are confident, significantly reducing inference time. Conversely, for difficult pairs with minimal overlap or large appearance changes, the network allocates more computation. This adaptivity operates through both depth pruning (early stopping at confidence thresholds) and width pruning (iterative filtering of low-confidence keypoint matches). LightGlue achieves higher accuracy than SuperGlue while being substantially faster, making it more practical for applications requiring matching across many image pairs. However, like SuperPoint and SuperGlue, LightGlue is primarily trained and evaluated on perspective images, and its performance on 360° equirectangular imagery with severe distortions near poles has not been extensively studied.

3.3 Structure-from-Motion and Bundle Adjustment

COLMAP [10] has become the standard tool for Structure-from-Motion, providing robust geometric reconstruction and global bundle adjustment. The system performs incremental reconstruction by sequentially registering new images into an existing model, using feature matching to establish 2D-3D correspondences, then estimating camera poses through PnP (Perspective-n-Point) algorithms. After registering new images, COLMAP performs bundle adjustment to jointly refine camera poses, 3D point positions, and optionally camera intrinsics by minimizing reprojection errors across all observations.

However, COLMAP's bundle adjustment still minimizes geometric reprojection error rather than photometric residuals, which may leave small alignment errors that are acceptable for traditional 3D reconstruction but problematic for photometric rendering methods.

3.4 Gaussian Splatting with Pose Optimization

Gaussian Splatting has recently emerged as a powerful alternative to implicit neural representations for novel view synthesis, offering real-time rendering through explicit 3D Gaussian primitives. Most Gaussian Splatting implementations assume camera poses are provided as input, typically obtained from COLMAP. Recent work has begun exploring joint optimization of both the 3D scene representation and camera poses.

BAD-Gaussians [12] addresses the challenge of learning Gaussian Splatting from motion-blurred images with inaccurate initial poses. The method models camera motion during exposure as a continuous trajectory in SE(3) space, then jointly optimizes the trajectory and Gaussian parameters by minimizing the photometric error between rendered and captured blurred images. This demonstrates that differentiable Gaussian Splatting can enable pose refinement through photometric objectives.

InstantSplat [2] focuses on sparse-view reconstruction by leveraging geometric priors from pre-trained models like MASt3R. It introduces Gaussian Bundle Adjustment that jointly optimizes both scene representation and camera parameters by minimizing photometric rendering error. However, the method is primarily designed for very sparse image sets (3-12 views) rather than dense video sequences. 3R-GS [5] combines 3DGS-MCMC for efficient Gaussian management with an MLP-based pose refiner and geometric constraints from feature correspondences to improve pose optimization stability.

While these methods demonstrate the feasibility of joint pose and scene optimization in Gaussian Splatting frameworks, they face challenges. Joint optimization can be unstable, potentially getting trapped in local minima where pose errors are compensated by scene geometry distortions. The optimization must carefully balance learning scene structure while simultaneously correcting pose errors, requiring careful initialization and hyperparameter tuning. Furthermore, these methods have primarily been developed and evaluated on perspective camera imagery, and their applicability to 360° equirectangular or cubemap representations remains largely unexplored. The severe distortions and wrap-around boundaries in 360° projections may introduce additional challenges for joint optimization approaches.

4 Methodology

Based on this previous work, our pipeline adds three crucial steps that improve the pose outputs from Stella VSLAM before using them with Gaussian Splatting. The key contribution is a rig-based Structure-from-Motion (SfM) method

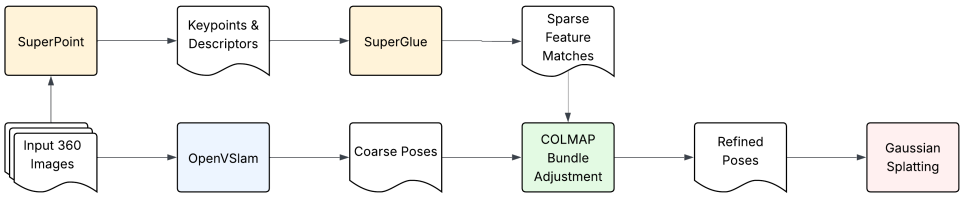


Figure 1: Full Pipeline Overview

that builds upon COLMAP's bundle adjustment process, and additional improvements are rendered through the use of SuperPoint and LightGlue, availing learning-based methods for feature detection, image descriptors, and image matching. This can be visualized in Figure 1.

4.1 Rig-Based SfM

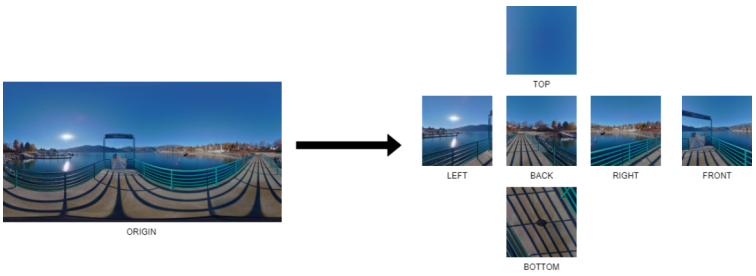


Figure 2: Cubemap from Equirectangular Image

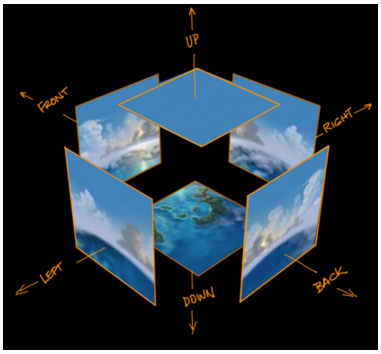


Figure 3: Rig for SfM

An equirectangular image can be decomposed into a cubemap, yielding 6 perspective images that existing bundle adjustment and SfM pipelines are used to working with, as seen in Figure 2. However, this multiplies the number of images that need to be worked with sixfold, making it a harder problem than we started with. The key observation is that these six images are constrained physically to one another, allowing for the creation of a rig as is shown in Figure 3. As the camera-to-rig extrinsics are known, this limits the number of parameters that need to be optimized for to just the 3D point position and orientation of each rig, allowing for just the tracking of the rig over time to be the optimization goal.

For the bundle adjustment, we want to minimize the sum of the squared re-projection error of 3D points onto each camera. Therefore, for every observed keypoint, we project the 3D point onto the camera, measure the pixel error, and optimize for that goal. We replace the constraints in traditional bundle adjustment with the one below:

$$T_{world \leftarrow cam_i} = T_{world \leftarrow rig_{r(i),k(i)}} \cdot T_{rig_{r(i)} \leftarrow cam_{c(i)}}$$

This enables the rig-based bundle adjustment that yields significant improvements over existing pose-estimation techniques.

4.2 SuperPoint and LightGlue

The first two steps of bundle adjustment are to detect features in each image, then match them using some form of image descriptor. In COLMAP [10], this is done using SIFT [7] as image descriptors, and a variety of configurable matching techniques. We bolster the accuracy of our pipeline by implementing two recent learning-based methods of image feature detection and description, and feature matching - SuperPoint and SuperGlue. This is enabled through our decomposition of the equirectangular input image into 6 perspective images.

SuperPoint's fully-convolutional neural network is first used to detect features in the image and assign descriptors to them. These are then passed through LightGlue [6], an attention-based improvement on SuperGlue that increases both accuracy and matching speed, whose outputs are then fed into the rig-based bundle adjustment as described above.

4.3 Gaussian Splatting

Once these features are extracted and matched, the rig constrained system is passed though bundle adjustment to yield precise poses, which are fed into a vanilla 3D Gaussian Splatting pipeline. The initial point cloud consists of the feature points triangulated from the bundle adjustment step, yielding a sparse point cloud to pass in with the optimized poses.

4.4 Data

To evaluate the 360° scenes, we initially considered utilizing Kitti360 and Matterport datasets. However, for this pipeline we utilize temporal image sequences, making the Matterport dataset unviable. Furthermore, with Kitti360, the dataset is tailored for multi-camera fusion, and stitching the dual fisheye results in a missing section over the car due to the use of 2 separate fisheye cameras recording at either side of the car. Therefore, we decided to collect our own dataset with a 360° Camera.

To first validate our hypothesis, we render 360° videos using a Unity-rendered 3D scene. This is used to validate drift and errors in the SLAM-based method and also find the best setting for real-world data capture. We then collect our own videos using a 360° camera and a outdoor walk. We utilize an Insta360 X4, collecting videos at both 5.7k and 8k resolution. We temporally sample frames from the video, passing them into our pipeline. This sampling method can be increased or decreased in density, where our experiments showed 1 frame every second co-aligned with keyframes outputted by VSlam. Our dataset consists of 8 360° video walks varying between 3-8 minutes long. We collect our videos with and without loop closure, to evaluate global bundle adjustment results on Stella VSLAM and rig based SfM.

5 Results

To evaluate the proposed pipeline, we compare gaussian splatting reconstructions generated using raw Stella VSLAM poses against those obtained after applying our rig-based bundle adjustment.

5.1 Hypothesis Validation on Synthetic Data

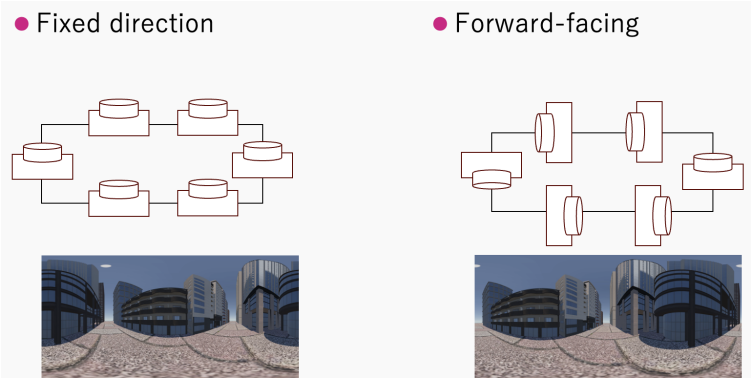


Figure 4: Explanation of direction-lock camera orientation used during capture for the rectangular trajectory.

We first examine the recovered camera trajectories to identify drift and structural inconsistencies in pose estimation. To validate our hypothesis that the estimated poses are insufficient for high-fidelity 3D reconstruction, we captured two controlled videos - a rectangular trajectory and an oval trajectory - within a virtual rendered scene in Unity.

These controlled sequences also allow us to determine which 360° camera settings produce the most stable baseline for pose tracking and estimation.

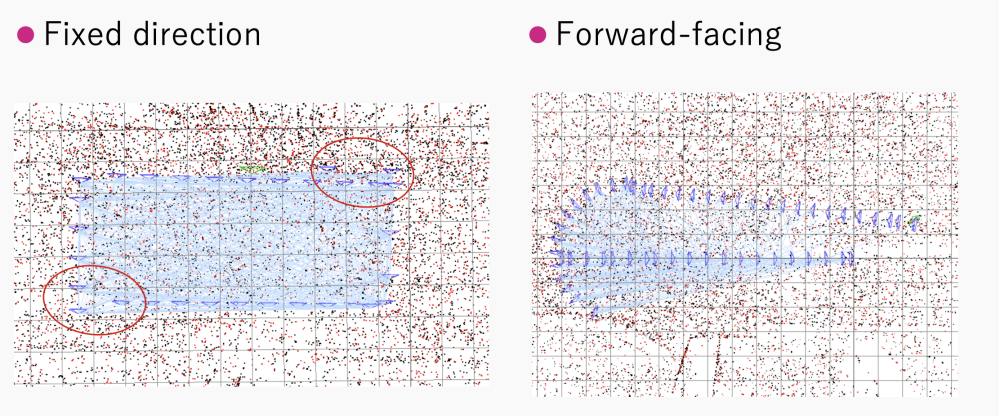


Figure 5: Stella-VSLAM trajectory outputs for a rectangular ground-truth path under direction-locked and forward-facing capture settings.

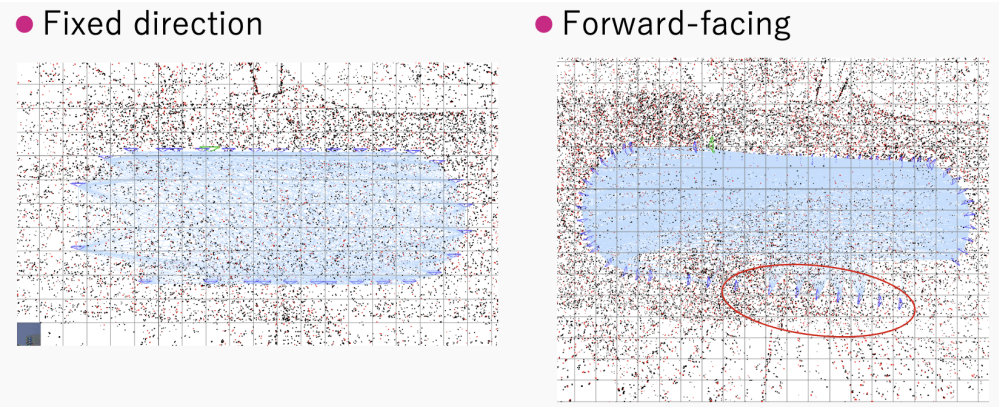


Figure 6: Stella-VSLAM trajectory outputs for an oval ground-truth path under direction-locked and forward-facing capture settings.

As Figures 5 and 6 show, direction-lock mode produces noticeably more stable pose estimates. Loop closure fails consistently in the forward-facing sequences, resulting in heavily distorted trajectories. Consequently, we use direction-lock mode for all real-world videos during quantitative evaluation.

During evaluation of the rectangular path, we also observed intermittent keypoint jumps and trajectory outliers, likely caused by noisy triangulation and unstable depth initialization. For the oval path, we see sparse tracking along the turning segment, indicating a clear graph-tracking failure that leads to local drift in relative pose estimates.

A qualitative analysis of real-world captured videos, as shown in Figure 7 revealed similar outliers in the estimated trajectories, even when direction-lock mode was enabled.

These findings validated our hypothesis and motivated further quantitative studies on our pose-refinement pipeline.

Figure 7: Example of a SLAM tracking outlier observed in real-world video capture.

5.2 Evaluation on Real-World Capture

(a) Stella-VSLAM: Example 1

(b) Stella-VSLAM: Example 2

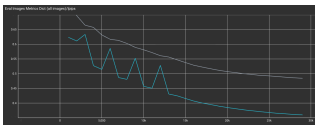
(c) Rig-based BA: Example 1

(d) Rig-based BA: Example 2

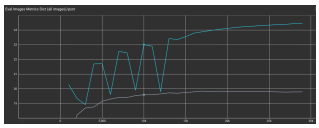
Figure 8: Ground Truth vs. Rendered Novel Views Comparison Between Stella VSLAM and Our Pipeline

For the eight videos captured in real-world conditions, we render novel views after gaussian splat training using a held-out evaluation split (one out of every eight frames). We then visually assess reconstruction fidelity and artifact suppression across both pose-estimation methods.

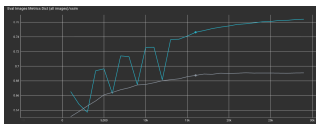
For each scene, we compute PSNR, SSIM, and LPIPS on the held-out evaluation images and report aggregated values across all eight scenes. For the largest scene, as seen in Figure 9, we additionally plot the metric curves over gaussian splat training iterations to analyze convergence behavior and compare how quickly reconstruction quality improves under raw Stella VSLAM poses versus our rig-based bundle adjustment.



(a) LPIPS



(b) PSNR



(c) SSIM

Figure 9: Comparisons between raw Stella VSLAM poses (gray) and rig-based bundle adjustment (blue).

Overall, PSNR and SSIM continue to improve up to 30k iterations for the rig-based bundle adjustment, whereas both metrics plateau early for the 360° SLAM method. While LPIPS decreases for both methods, the final values are significantly better for the rig-based bundle adjustment. The aggregated quantitative results are shown in Table 1.

Method	SSIM	PSNR	LPIPS
Direct Stella-VSLAM	0.6909	19.79	0.4844
Rig-Based BA	0.7640	24.45	0.3598

Table 1: Quantitative comparison of pose estimation methods on the held-out evaluation set.

One notable observation is that the rig-based SfM plots exhibit fluctuations early in training, which largely stabilize after approximately 15k iterations. This coincides with the pruning stage in gaussian splat training, where noisy or duplicate gaussians are removed, effectively resetting parts of the scene geometry. We discuss this behavior further in the Limitations section.

We show qualitative renderings in Figure 8 for both methods. We include two randomly selected evaluation images for each method, along with side-by-side comparisons of Stella VSLAM versus rig-based bundle adjustment for the same scene locations.

Figure 10: Comparison at same viewpoint: Stella-VSLAM (top) vs. Rig-based BA (bottom).

6 Limitations

The first limitation we noticed was the scale of the scene. Since the video feed is monocular, the outputs are always relative, not metric.

The second issue we faced were distractors in the scene, seen in Figure 11. Distractors are moving objects that cannot be reconstructed in a static scene, and result in warped results in the GS, and reduced keypoint reliability in the registration process. We did notice that the SLAM results and rig-based bundle adjustment results were significantly robust due to 360° tracking, but the metrics during evaluation in the gaussian splatting scenes were significantly affected. This was the reason for the fluctuations in the initial 15k iterations for rig-based SfM (As it was temporal, while Stella VSLAM chose keypoints with maximum features, inherently choosing frames with fewest distractors). Future works could explore masking such distractors for both bundle adjustmsnt and 3D reconstruction, potentially resulting in better results.

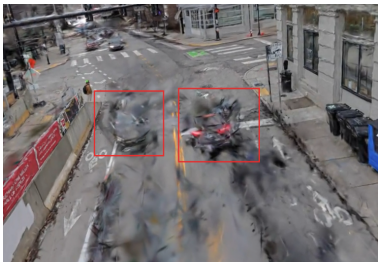


Figure 11: Distractor Reconstruction Result with Stella-VSLAM.

The third issue was computation. While rig-based BA could use constraints and incremental mapping to keep overhead low, the Stella VSLAM had a significant overhead memory and CPU usage and would crash on videos over 8 minutes long. Furthermore, the computation is exponential with equirectangular videos, so increasing the resolution from 2k, for which Stella VSLAM was designed, to the 8k videos that we used resulted in a significant increase, of up to 2 hours of processing for SLAM alone.

References

[1] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018.

[2] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2024.

[3] Mateusz Janiszewski, Masoud Torkan, Lauri Uotinen, and Mikael Rinne. Rapid photogrammetry with a 360-degree camera for tunnel mapping. *Remote Sensing*, 14(21):5494, 2022.

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[5] Joonhee Kim, Jaewoo Park, Joonghyuk Kim, et al. 3r-gs: Best practice in optimizing camera poses along with 3dgs. *arXiv preprint arXiv:2504.04294*, 2025.

[6] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed, 2023.

[7] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[9] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks, 2020.

[10] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. OpenVSLAM: A Versatile Visual SLAM Framework. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 2292–2295, New York, NY, USA, 2019. ACM.

[12] Lingzhe Zhao, Peng Wang, and Peidong Liu. Bad-gaussians: Bundle adjusted deblur gaussian splatting. *arXiv preprint arXiv:2403.11831*, 2024.