UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Leveraging Depth Maps and 3D Gaussian Splatting for Camera Pose Recovery and 3D Scene Reconstruction

Author:
Ostap Hembara

Supervisor:
Denys Rozumnyi

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2024

Declaration of Authorship

I, Ostap Hembara, declare that this thesis titled, “Leveraging Depth Maps and 3D Gaussian Splatting for Camera Pose Recovery and 3D Scene Reconstruction” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Leveraging Depth Maps and 3D Gaussian Splatting for Camera Pose Recovery and 3D Scene Reconstruction

by Ostap Hembara

Abstract

In recent years, the field of 3D scene reconstruction has witnessed significant advancements, fueled by growing interest in applications ranging from augmented reality to autonomous navigation. A key component of this progress has been the development of Neural Radiance Fields (NeRF), which have revolutionized the way we render and interact with 3D environments. Despite these advancements, the process of camera pose estimation remains a bottleneck, often requiring extensive computational resources and time. This thesis introduces an innovative approach that leverages 3D Gaussian Splatting, a technique that provides a more explicit representation during both the rendering and training phases, enhancing the efficiency and clarity of 3D reconstructions. Specifically, we focus on a method that utilizes estimated monocular depth maps to recover camera poses, which are then used to reconstruct the 3D scene. This methodology not only simplifies the traditional pipeline by obviating the need for direct pose estimation but also improves the speed of the reconstruction process. We evaluate our approach using both synthetic and real-world datasets, in order to see it performance in different scenarios.

Acknowledgements

I extend my deepest gratitude to those valiant individuals who continue to fight against Russia, whose sacrifices have made it possible for me and many others to continue our research and academic pursuits in these challenging times of war.

I am thankful to my project advisor, Denys Rozumnyi, for his invaluable guidance and unwavering support throughout this project. His expertise and encouragement have been crucial in shaping this research.

My sincere appreciation also goes to Reface, whose generous provision of computational resources has been essential for the conduct of this study.

I would also like to thank the organizers and donors of the study program for making this research possible.

Last but not least, many thanks to my wife for her support and for the many hours spent listening to my ideas and providing feedback that was crucial in my pursuit.

Contents

Declaration of Authorship ii

Abstract iii

Acknowledgements iv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Objective . . . 2
- 1.3 Thesis Structure . . . 2

2 Related Work 3
- 2.1 Traditional 3D Reconstruction Methods . . . 3
- 2.2 Exploring Neural Radiance Fields . . . 4
- 2.3 Point-Based Rendering and 3D Gaussian Splattings . . . 4
- 2.4 Camera Pose Recovery Methods . . . 5

3 Method 8
- 3.1 3D Gaussian Spattings . . . 8
- 3.2 Relative Pose Estimation with Local 3DGS . . . 10
- 3.3 Adjusted Poses with Global 3DGS . . . 13

4 Data 16
- 4.1 Real World Data (CO3D) . . . 16
- 4.2 Synthetic Data . . . 17

5 Experiments 19
- 5.1 Implementation Details . . . 19
  - 5.1.1 Data Preprocessing . . . 20
  - 5.1.2 Training Pipeline and Parameters . . . 20
  - 5.1.3 Evaluation Metrics . . . 22
- 5.2 Comparing Pose Recovery Strategies . . . 24
  - 5.2.1 Local 3DGS . . . 24
  - 5.2.2 Local and Global 3DGS . . . 25
  - 5.2.3 Opacity Filtering . . . 25
  - 5.2.4 Simultaneous Optimization of Camera Poses and Global 3DGS 26
  - 5.2.5 Results . . . 27

6 Conclusions 30
- 6.1 Experiments Summary . . . 30
- 6.2 Limitations . . . 31
- 6.3 Contribution . . . 31
- 6.4 Future Work . . . 32

Bibliography **33**

List of Figures

3.1 Example of 3DGSs of different scale, orientation and center that lie along the ray[Yurkova, 2023]. 9
3.2 From initial unprotected point cloud, the optimization process renders and generates a set of 3DGS G_0. Optimization scheme provided by authors [Bernhard Kerbl, 2023] 11
3.3 Relative transforms $(q_0, t_0), (q_1, t_1), ...(q_{n-1}, t_{n-1})$ between timestemps $I_0, I_1, ..., I_n$. Image adapted from [Sweeney, n.d.] 12
3.4 Overall optimization pipeline scheme. First we fit local 3DGS G_{L_t} for each frame I_t, then we estimate local relative transforms R_{L_t}, T_{L_t} using photometric loss. After, we optimize global 3DGS Γ_t with t cameras and adjust global relative transform R_{G_t}, T_{G_t} btw I_0 and I_t. 14

4.1 Overview of the CO3D dataset featuring various objects from multiple views showed by the authors[Reizenstein et al., 2021]. 16
4.2 Example of C03D dataset images, mask and labeled camera poses using SFM methods given by the authors of the dataset[Reizenstein et al., 2021]. 17
4.3 Example of custom synthetic dataset with moving trajectorie around the cow object. RGB axices represent camera's orientation, red is right vector, blue is forward and green is up. 18

5.1 Unprojected pcd in camera space for I_0 frame **car** sequence. Here axises represent camera orientation with blue being forward, red is right and green is up vectors. 21
5.2 Before and after fitting local 3DGS for I_0 frame **car** sequence. 21
5.3 Procrustes analysis applied to 2 unaligned set of points. a) scales 2 sets to same size; b) shifting to same position; c) aligning orientation. Image provided by [*Procrustes analysis* n.d.] 23
5.4 Recovered trajectories for video sequences using only local 3DGS; a) cow synthethic with ground truth depth; b) cow synthethic using estimated depth; c) teddybear_co3d sequence; blue line is ground truth trajectory and red is recovered; 24
5.5 Filtering 3DGS with different opacity treshholds 26
5.6 Recovered trajectories for video sequences using local and global 3DGS with simultaneous optimization; a) cow synthethic with ground truth depth; b) cow synthethic using estimated depth; c) teddybear_co3d sequence; blue line is ground truth trajectory and red is recovered; . . 26
5.7 Recovered trajectories for each video sequence and novel-view synthesis. Blue trajectories are ground truth and red ones are estimated. On left novel-view images generated from model and on right side are ground truth images. 29

List of Tables

5.1 Comparison of $\mathbf{RPE_{rot}}$ metrics for different strategies across video sequences. $\mathbf{RPE_{rot}}$ units of measure are degrees. 27
5.2 Comparison of $\mathbf{RPE_{trans}}$ metrics for different strategies across video sequences 27
5.3 Comparison of **SSIM** metrics for different strategies across video sequences 28
5.4 Comparison of **PSNR** metrics for different strategies across video sequences 28

To my grandmother and grandfather, who would sacrifice everything so that I could have something.

Chapter 1

Introduction

1.1 Motivation

In recent years, the development of technologies [Gaurav Chaurasia and Drettakis, 2013; Georgios Kopanas and Drettakis, 2021] capable of reconstructing 3D scenes in real-time has gathered increasing attention due to its vast array of applications. These applications span various fields such as augmented and virtual reality, where users interact with seamless integrations of digital and real-world elements[Zhiwen Fan, 2023], autonomous vehicle navigation which relies on accurate environmental modeling for safe operation, and cultural heritage preservation where delicate artifacts and sites can be digitally preserved in great detail. The ability to render these reconstructions in real-time significantly enhances user experience and operational efficiency, making the pursuit of more advanced 3D reconstruction methods a priority in Computer Vision research field.

The subject of 3D reconstruction is divided into several categories, each with its advantages and disadvantages:

Traditional Methods: These methods typically rely on feature extraction and matching across multiple images to estimate depth and reconstruct scenes. Techniques such as stereo vision and structure from motion (SfM) fall into this category. While effective, they often suffer from high computational costs and can struggle with textureless surfaces or repetitive patterns.

Neural Radiance Fields (NeRF): NeRF has emerged as a powerful tool for high-fidelity 3D rendering by utilizing deep learning to model the volumetric scene function[Alex Yu, 2021]. This technique excels in handling complex lighting and fine details but requires substantial computational resources and processing time, making it less feasible for real-time applications[Truong et al., 2023].

3D Gaussian Splattings: As an evolution in rendering technology, 3D Gaussian Splattings offer a novel approach by using Gaussian kernels to project 3D points onto 2D planes, effectively allowing for efficient and dynamic scene rendering[Charatan et al., 2023]. This method has shown promising results in terms of both speed and quality of the reconstructions, making it particularly suitable for real-time applications[Paliwal et al., 2024].

Among the mentioned methods, 3D Gaussian Splattings (3DGS) stand out due to their demonstrated ability to efficiently balance rendering quality with computational speed[Chung, Oh, and Lee, 2023]. The explicit representation provided by 3DGS aids in clearer and more accurate scene reconstructions. However, a significant challenge in fully leveraging the potential of 3D Gaussian Splattings lies in the prerequisite of known camera poses. Traditional methods for determining these poses, such as using COLMAP[Schönberger and Frahm, 2016], are computationally

intensive and can be impractical in real-time scenarios. This creates a bottleneck that hinders the broader application of 3DGS in real-time 3D reconstruction.

Since accurate camera poses are essential for reconstructing scenes using 3D Gaussian Splatting, this work aims to develop new method to recover camera poses more efficiently and with less computation. By leveraging estimated monocular depth maps, we propose a novel approach to deduce camera poses without the need for traditional, resource-heavy algorithms. By significantly reducing computational load and rendering time, this advancement opens up new possibilities for real-time 3D scene reconstruction

1.2 Research Objective

In this master thesis, we outline ideas to boost the efficiency and accuracy of 3D scene reconstruction using 3D Gaussian Splatting. Our study concentrates on simplifying the reconstruction process, by eliminating the typically required step of camera pose recovery using conventional methods. We focus on the following goals:

- employing 3D Gaussian Splatting for effective 3D scene reconstruction, capitalizing on its ability to deliver high-quality outputs with reduced computational demands and training time;
- seeking to avoid traditional camera pose estimation methods due to their computational intensity and time consumption;
- leveraging depth maps which are generated from monocular images by a depth estimation model;
- use synthetic data with known ground truth depth and real-world data to evaluate performance in different scenarios.

1.3 Thesis Structure

In Chapter 2 we review relevant methods for 3D scene reconstruction and pose recovery. Suggested method and theory that describes it outlined in Chapter 3. Description of custom generated data and real-world data can be found in Chapter 4. Chapter 5 provides detail overview about experiment settings, data preprocessing and result metrics. Last but not list, conclusions and future work ideas is given in Chapter 6.

Chapter 2

Related Work

2.1 Traditional 3D Reconstruction Methods

Traditional 3D reconstruction methods have established a robust framework for generating three-dimensional models from two-dimensional data. These techniques, which have evolved significantly over the years, are characterized by their robust handling of semantic information and intricate multi-staged processes that add layers of complexity to their implementation.

The advent of Structure-from-Motion (SfM) technology, notably advanced by [Snavely, Seitz, and Szeliski, 2006], marked a significant development in the field. SfM facilitates the creation of a sparse point cloud by analyzing a collection of photographs to estimate camera positions and orientations in 3D space. This initial stage is crucial for understanding the basic structure of the scene but is generally not sufficient for full 3D reconstruction. This limitation led to the integration of multi-view stereo (MVS) techniques, as seen in the work of [Galliani, Lasinger, and Schindler, 2015], which build upon the sparse data provided by SfM to create more detailed and comprehensive 3D models. These models are constructed by meticulously aligning and merging multiple images from different viewpoints, thereby enhancing the depth and realism of the reconstruction.

Following the construction of 3D models, the process of view synthesis begins, where novel views of the scene are generated by re-projecting and blending input images based on the reconstructed geometry. This stage is pivotal in applications such as virtual reality, filmmaking, and architectural visualization, where lifelike renderings of virtual scenes are required. Techniques developed by researchers like [Schönberger et al., 2016b; Schönberger et al., 2016a]. Xu and Tao, 2019 have shown impressive capabilities in synthesizing new views by effectively utilizing the underlying geometric data. However, despite their successes, these methods often struggle with issues such as unreconstructed regions—areas where the geometry has not been fully captured—and "over-reconstruction," where erroneous geometry is produced.

Recent advances in neural rendering, as highlighted by [Tewari et al., 2022] have started addressing these challenges more effectively. Neural rendering techniques leverage deep learning to refine the synthesis process, reducing artifacts and improving the quality of the reconstructed views. Additionally, these methods mitigate the need for extensive computational resources traditionally required for storing and processing large sets of input images, particularly on GPU-intensive tasks.

Overall, traditional 3D reconstruction methods form a complex, multistage pipeline that starts from basic geometrical estimation to sophisticated view synthesis.

2.2 Exploring Neural Radiance Fields

Neural Radiance Fields (NeRFs) have emerged as a groundbreaking approach in the field of 3D scene reconstruction, particularly known for their ability to synthesize highly realistic images from sparse views of a scene. Developed by [Mildenhall et al., 2020], NeRFs use a fully connected deep neural network to model the volumetric scene implicitly.

At its core, a NeRF represents a scene using a continuous 5D function that maps spatial coordinates (x, y, z) and viewing directions (Θ, Φ) to color (RGB) and density (σ). The model takes a set of sparse 2D images of a scene, each associated with camera parameters, and learns to predict the color and opacity of points in space as seen from specific viewpoints. During training, the network optimizes a loss function that minimizes the difference between the observed colors in the images and the colors predicted by the model for corresponding camera rays[Lin et al., 2021].

The process involves casting rays through each pixel in the training images and sampling points along these rays. For each sample point, the NeRF model predicts both color and density. The color and density values are then used to compute the final pixel color through a differentiable rendering technique known as volume rendering. This technique uses the classic "alpha compositing" approach to blend colors along a ray, considering the accumulated opacity.

NeRF employs a simple but deep fully connected neural network architecture with several layers of ReLU activations. The input to the network includes the 3D coordinates and 2D direction parameters, which are first transformed using positional encoding to allow the model to capture higher frequency details in the scene[Jeong et al., 2021]. This encoding transforms each input coordinate into a higher-dimensional space, enabling the network to learn fine spatial variations more effectively.

Despite the simplicity of the architecture, the key to NeRF's effectiveness lies in its training strategy and the use of hierarchical volume sampling techniques to efficiently approximate the integral over the volume rendering equation. This hierarchical approach speeds up convergence and enhances detail capture by focusing more network capacity on areas with complex geometry or high variation in appearance.

NeRFs can generate highly realistic and detailed 3D renderings from a limited set of images, surpassing the quality of traditional methods. They excel at synthesizing novel views of a scene with precise handling of occlusions and complex lighting[Barron et al., 2021].

NeRFs require significant computational resources and time to train, often needing multiple GPUs and hours to days of processing time. Due to the intensive computation required during inference, NeRFs are generally not suitable for real-time applications[Zhang et al., 2020].

In summary, NeRFs represent a significant advance in the field of 3D scene reconstruction, offering unparalleled rendering quality at the cost of high computational demands and long training times. As research progresses, ongoing efforts aim to address these drawbacks, making NeRFs more practical for a broader range of applications.

2.3 Point-Based Rendering and 3D Gaussian Splattings

Point-based rendering methods[Franke et al., 2024], and particularly 3D Gaussian Splattings, have brought significant advancements to the field of 3D reconstruction

by focusing on efficiency and adaptability. 3DGS, a novel approach within this category, has revolutionized the process by addressing some of the critical limitations of traditional and other contemporary rendering techniques[Kopanas et al., 2021].

They work by projecting 3D points onto a 2D plane, using Gaussian kernels to manage the splatting process. This technique treats each point in a point cloud as a center of a 3DGS, blending these points smoothly in the image space to form a continuous surface representation. The use of Gaussian kernels helps in handling overlaps between points, smoothing out the resulting image and filling gaps effectively.

This approach differs from traditional mesh-based methods by eliminating the need to construct complex polygonal meshes. Instead, 3D Gaussian Splattings rely directly on the raw point clouds, which are simpler to manipulate and can be dynamically adjusted without complex computations. This direct utilization of point clouds not only simplifies the processing pipeline but also reduces the computational overhead associated with mesh processing[Wu et al., 2023].

Unlike methods that require dense mesh or volumetric representations, 3D Gaussian Splattings maintain a relatively low memory footprint. This efficiency is particularly beneficial when dealing with large datasets or when operating within hardware with limited memory resources[Zhang et al., 2024].

One of the most significant advantages of 3DGS is the minimal training required compared to deep learning-based methods like NeRFs. This feature, coupled with its fast rendering capabilities, makes a method suitable for applications needing quick movement through the scene, such as interactive 3D applications[Müller et al., 2022].

The simplicity and efficiency of 3D Gaussian Splattings allow for easier integration into real-time systems such as game engines and web browsers. This compatibility opens up possibilities for real-time interactive 3D visualizations on the web and in gaming, where users can experience high-quality 3D environments without the need for extensive computing resources.

2.4 Camera Pose Recovery Methods

Traditional methods of 3D scene reconstruction typically rely on accurate camera pose estimation to align and integrate multiple views into a coherent 3D model. This often involves complex photogrammetric software like COLMAP[Schönberger and Frahm, 2016], which utilizes feature extraction, feature matching, and bundle adjustment techniques to deduce camera positions and orientations. However, there are innovative approaches that circumvent the need for direct camera pose estimation, leveraging alternative data sources and methodologies to simplify the reconstruction process.

Gaussian SLAM[Yugay et al., 2023] utilizes the concept of 3D Gaussian Splattings within a SLAM framework to enhance both the mapping and localization processes. This method smoothness the data representation and aids in the interpretation of scene structure. The 3DGS representation helps to efficiently manage and merge overlapping data points from different viewpoints, enhancing the robustness of the map construction.

The primary advantage of Gaussian SLAM is its ability to handle large-scale environments with a high degree of detail. By employing Gaussian kernels, this method effectively reduces noise and fills gaps in the data, which are common issues in traditional SLAM systems that rely solely on raw point clouds or mesh data. Additionally, the 3DGS approach allows for continuous updates to the map with

minimal computational overhead, making it suitable for dynamic environments and mobile platforms such as robots and autonomous vehicles.

SplaTAM (Splat, Track and Map)[Keetha et al., 2024] is a sophisticated approach to SLAM that utilizes 3D Gaussian Splatting techniques specifically adapted for dense RGB-D data. This method differentiates itself from other SLAM approaches by focusing on high-density, real-time mapping and tracking using RGB-D sensors, which provide both color (RGB) and depth (D) data simultaneously. SplaTAM integrates these data types into a cohesive system that leverages the advantages of 3D Gaussian Splatting for enhanced environmental mapping and navigation. While both SplaTAM and Gaussian SLAM utilize 3DGS techniques, their applications and methodologies have distinct differences. SplaTAM emphasizes real-time tracking and mapping capabilities, ideal for applications requiring immediate feedback and interaction, such as augmented reality or robotic navigation in dynamically changing environments.

The NoPE-Nerf method, introduced by [Wenjing Bian, 2022], marks a substantial development in the Neural Radiance Fields (NeRF) technology by eliminating the need for pre-determined camera poses. This innovative approach integrates pose estimation directly into the NeRF optimization pipeline, allowing it to learn the geometry of the scene and the viewpoints of the images simultaneously. What sets NoPE-Nerf apart is its unique framework which employs an unsupervised learning strategy to infer camera poses directly from image data. This method uses differential rendering as a feedback mechanism to iteratively adjust both the 3D scene parameters and the associated camera poses. By treating pose estimation as an integral part of the NeRF optimization process, NoPE-Nerf avoids the common pitfalls associated with incorrect or imprecise initial pose estimates that can significantly degrade the quality of the reconstructed scene. NoPE-Nerf also introduces a regularization scheme designed to prevent the convergence to degenerate or implausible camera configurations, a common issue in unsupervised pose estimation scenarios. This regularization is crucial for maintaining the integrity of the camera trajectories and ensuring that the pose estimates evolve in a physically plausible manner throughout the learning process.

Lu-Nerf, developed by [Cheng et al., 2023], introduces a novel framework for scene and pose estimation by leveraging the concept of local Neural Radiance Fields (NeRFs) that are initially unposed. This method focuses on synchronizing multiple local NeRFs to reconstruct a scene and estimate camera poses simultaneously. The core idea is to divide the scene into smaller segments, each represented by a local NeRF, and then optimize these segments together to achieve global consistency in both geometry and appearance. This approach not only enhances the efficiency of the NeRF-based reconstruction process but also improves its scalability and flexibility in handling diverse and complex scenes. Lu-Nerf is particularly effective in environments where traditional global NeRF applications might struggle with computational overhead or require excessive fine-tuning.

NeRF–, proposed by [Wang et al., 2022], tackles the challenge of constructing Neural Radiance Fields without pre-known camera parameters. This method innovatively modifies the traditional NeRF approach by incorporating a self-calibrating mechanism that estimates camera poses in conjunction with the scene's 3D geometry. The key advancement here is the method's capability to adaptively refine camera pose estimates through a continuous feedback loop that aligns the generated 3D model with observed image data. This capability makes NeRF– highly applicable in situations where camera metadata is unavailable or unreliable, such as in historical image datasets or in applications where manual calibration is impractical.

Recent advancements in 3D reconstruction technology have led to the development of COLMAP-free 3D Gaussian Splattings, as described by [Fu et al., 2023]. This innovative method leverages depth information extracted directly from images, bypassing the traditional need for explicitly calculating camera poses. By using depth maps to inform the placement and shaping of 3D Gaussian Splattings, this approach simplifies the reconstruction process, significantly reducing its complexity and computational demands. COLMAP-free 3D Gaussian Splattings capitalize on both local and global 3DGS models to refine the scene's structure iteratively. Initially, local 3DGS models are constructed for individual images using the depth data to approximate the scene's geometry at a granular level. These local models are crucial for capturing fine details and subtle nuances of the scene's local geometry. As the reconstruction progresses, these local models are integrated into a global 3DGS framework. This global model aggregates information across multiple views, enhancing the overall consistency and coherence of the reconstructed scene. The process involves a strategic densification of 3DGSs, where additional 3DGSs are gradually added to the model to increase its resolution and detail, especially in areas where the initial depth data may be sparse or noisy.

The main advantage of using depth maps for pose recovery lies in its simplicity and speed. Unlike traditional methods that require multiple stages of processing and optimization, depth-based pose recovery can be integrated directly into the rendering pipeline of 3DGS. This integration not only speeds up the reconstruction process but also reduces the potential for errors that are often introduced during multi-step processing.

Chapter 3

Method

In this chapter, we explore the methods we've used for 3D scene reconstruction with 3D Gaussian Splattings, partially inspired by similar techniques from previously published research, notably the approach described in the paper on Fu et al., 2023. Recognizing the influence of this foundational work is essential. However, due to a lack of detailed explanations and unavailable code at the time of our study, we encountered significant gaps in understanding how to implement these methods fully.

To address these gaps, we developed our own version of the methodology. This chapter will detail our approach, highlighting where our methods align with the established techniques and where we've introduced new ideas or modifications. Our goal is to clearly differentiate our unique contributions from the baseline methods, offering a transparent view of how we've built on and diverged from previous works.

3.1 3D Gaussian Spattings

In the development of our approach for a 3D scene reconstruction, we incorporate a novel method known as 3D Gaussian Splattings (3DGS), which offers a distinct way to model and render complex scenes efficiently. Here we detail the main aspects of 3DGS and its integration into our methodology, emphasizing its parametric nature and the advantages it brings to 3D reconstruction tasks.

3DGS models the scene as a collection of 3D Gaussians, each defined by a set of parameters that explicitly describe its geometric and optical characteristics. This method contrasts with implicit forms of representation such as those used in Neural Radiance Fields (NeRFs), providing a more direct and manipulable model of the scene. Each Gaussian in 3DGS is characterized by a center (mean) point, The Gaussian function $G(x)$ in 3DGS is defined as follows:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)} \tag{3.1}$$

where μ is the mean (center) point, and Σ is the covariance matrix that defines the spatial distribution and orientation of the Gaussian in 3D space. This parametrisations represent anisotropic 3D elipses which are rasterized along the ray.

Each Gaussian is parameterized by several components that together describe its contribution to the scene:

- **Center Position** $\mu \in \mathbb{R}^3$: Specifies the central location of the Gaussian in the 3D space.

FIGURE 3.1: Example of 3DGSs of different scale, orientation and center that lie along the ray[Yurkova, 2023].

- **Color Representation**: Utilizes spherical harmonics (SH) coefficients $c \in \mathbb{R}^k$ to encode the color information, where k represents the degrees of freedom in the color model.
- **Rotation Factor** $r \in \mathbb{R}^4$: Defined in quaternion terms to manage the orientation of the Gaussian.
- **Scale Factor** $s \in \mathbb{R}^3$: Determines the size of the Gaussian along each axis, forming an ellipsoidal shape.
- **Opacity** $\alpha \in \mathbb{R}$: Controls the transparency of the Gaussian, affecting how it blends with other elements in the scene.

The rendering process involves projecting these 3D Gaussians onto a 2D image plane using the camera's view transformation W. The projection modifies the 3D covariance matrix into a 2D form:

$$\Sigma_{2D} = JW\Sigma W^\top J^\top \tag{3.2}$$

where J represents the Jacobian of the affine transformation approximating the projective transformation.

Each pixel's color and opacity result from alpha-blending the contributions of all Gaussians that influence that pixel, computed as:

$$C_{\text{pix}} = \sum_{i=1}^{N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{3.3}$$

This formula integrates the density and color of points, derived from their spherical harmonics and opacity parameters, through the Gaussian-defined ellipsoidal influence.

To perform scene reconstruction, the initialized 3DGS points are fitted to the observed data by optimizing these parameters against a photometric loss, using the differentiable rendering equation. Our approach enhances this process by integrating estimated camera poses instead of relying solely on ground-truth data, allowing for dynamic adaptation to varying observational conditions.

In our study, we primarily utilized the implementation provided by the authors as outlined in Bernhard Kerbl, 2023. Specifically, we adopted their approach of initializing the scene using a point cloud. We also applied heuristic values from the implementation, which have proven to accelerate the 3D reconstruction process for

individual frames. This methodological choice ensures a more efficient and reliable framework for our analysis.

3.2 Relative Pose Estimation with Local 3DGS

In our methodology, we employ the concept of local 3D Gaussian Splattings (3DGS), inspired by the techniques described in the Fu et al., 2023, to facilitate robust relative transformations between frames. This approach leverages the initial generation of a local 3D point cloud from depth and image data, which serves as the basis for subsequent 3D Gaussian modeling and optimization. The use of local 3DGS is crucial for establishing precise extrinsic transformations that are foundational to the accurate alignment and reconstruction of successive frames in a 3D space. By focusing on local model fitting, we ensure that each frame is evaluated independently, reducing the propagation of error across the sequence and enhancing the fidelity of the scene reconstruction.

While we utilize the standard 3DGS optimization pipeline Bernhard Kerbl, 2023 as a basis for our process as done in Fu et al., 2023, we have implemented specific modifications to better suit our reconstruction needs. Notably, we have adjusted the pipeline to include a densification step, which is initiated earlier in the optimization process to enhance the model's accuracy and detail from the outset. Additionally, unlike the default method, we have chosen to eliminate the step of resetting the opacity during optimization of local 3DGS. This change aims to expedite the overfitting process to the initial frame, allowing for faster convergence while still maintaining high accuracy in the fit between the modeled Gaussians and the observed scene data.

To establish an initial extrinsic transformation between two frames, we utilize local 3DGS. This technique begins by using the image I_0 and its corresponding depth map D_0 to unproject points into camera space, creating an initial 3D point cloud. These points form the foundation for fitting a local 3DGS model, which is overfitted to the scene. An essential step in this process involves employing metrical depth, which is crucial for ensuring consistent and comparable depth measurements across different views. Metrical depth, expressed in real-world units such as meters, allows for precise scaling and accurate positioning of objects within the 3D space, thus preserving the geometric fidelity of the scene.

For quantifying the fit between the 3 model and the scene, we apply a loss formula that combines the Structural Similarity Index (SSIM) and photometric L1 loss, effectively measuring discrepancies between the rendered 3DGS image G_0 and the original image I_0:

$$\text{Loss} = \lambda_1 \cdot \text{SSIM}(G_0, I_0) + \lambda_2 \cdot \|G_0 - I_0\|_1 \tag{3.4}$$

This loss function helps refine the Gaussian parameters to closely match the observed data in I_0, ensuring that the representation G_0 is tightly coupled with the actual scene's appearance and structure.

Once we have an overfitted representation of G_0, the next step involves estimating a 6-degree transformation comprising 3 degrees of rotation and 3 degrees of translation to find the relative pose transition into the next frame I_1. This transformation is optimized by minimizing the photometric loss same as in (3.4), but instead of I_0 - I_1, effectively aligning G_0 with I_1 based on observed image features. It is important to mention that we freeze all parameters of 3DGS during optimization of relative transform.

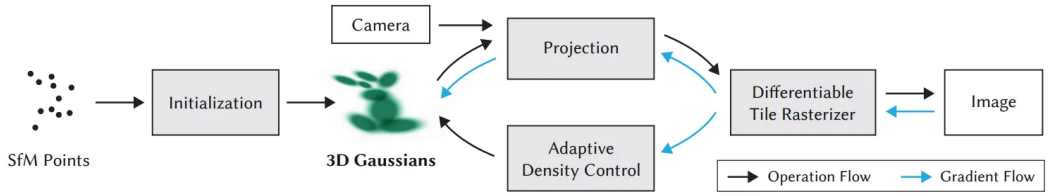

FIGURE 3.2: From initial unprotected point cloud, the optimization process renders and generates a set of 3DGS G_0. Optimization scheme provided by authors [Bernhard Kerbl, 2023]

Relative Pose Parameters

Similar to the strategy employed in Fu et al., 2023, we optimize camera transformations using a translation vector and a quaternion representation for rotation. Originally, Fu et al., 2023 did not specify details on how gradients were propagated to learn these parameters. Due to the limitations of the rasterizer used in the 3D Gaussian Splattings Bernhard Kerbl, 2023, we are unable to directly pass learnable rotation **R** and translation **T** parameters to the rasterizer, as gradient propagation is restricted to the parameters of the 3DGS models themselves. Consequently, rather than moving the camera during optimization, we adjust the scene itself by transforming the 3DGS parameters.

To address these challenges, we optimize the rotation component using a quaternion representation, expressed as

$$q = a + bi + cj + dk \tag{3.5}$$

where a is the real part, and b, c, d are the imaginary parts. This quaternion representation enables continuous and smooth optimization of scene rotation, capable of representing rotations exceeding 360 degrees. Unlike rotation matrices, which cannot be interpolated smoothly and may accumulate numerical errors, or Euler angles, which suffer from discontinuities and loss spikes at the 0 and 360-degree boundaries, quaternions avoid these issues. The ability to interpolate between quaternion states makes them particularly suitable for optimizing 3D reconstructions, allowing for more nuanced and precise control over the rotation of the scene elements

The translation component is optimized as a vector,

$$t = (t_x, t_y, t_z) \tag{3.6}$$

simplifying the model's learning process for spatial adjustments. This approach of directly optimizing rotation (**q**) and translation (**t**) parameters as part of the model ensures that the gradients can effectively propagate through these transformations, enabling robust learning and accurate pose estimation in complex 3D reconstruction tasks.

Applying Relative Pose to 3DGS Parameters

In our optimization process, we tackle the challenge of aligning local 3D Gaussian Splattings (3DGS) for each timestep t with the subsequent camera frame using a relative transformation. Our aim is to identify the transformation Θ_t that minimizes the discrepancy between the rendered image and the following frame. This transformation encompasses six degrees of freedom (6DoF), covering both rotation and translation components. However, a significant problem arises from our inability to directly apply these transformations to the rasterizer parameters, making it crucial

to handle these transformations with precision across the 3DGS parameters, which we detail further in this chapter. This careful application of transformations represents a key contribution of our work, filling a gap left by the original work Fu et al., 2023, which did not provide detailed implementation insights. The optimization formula can be expressed as:

$$\Theta_{\text{argmin}} = \arg\min_{\Theta} \text{Loss}\left(R(G_t \otimes \Theta_t), I_{t+1}\right) \tag{3.7}$$

Here, G_t represents the optimized local 3DGS at time t, Θ_t is the 6DoF transformation applied to G_t, and R denotes the rendering function that projects G_t transformed by Θ_t onto the image plane. I_{t+1} is the subsequent image frame against which the rendered image is compared.

To accurately simulate the effect of rendering Gaussian parameters with specific camera transformations, we methodically apply these transformations to specififc parameters of our 3D Gaussians. The focus is on adjusting the centers and the rotational attributes of the Gaussians to align with the new camera orientations and positions.

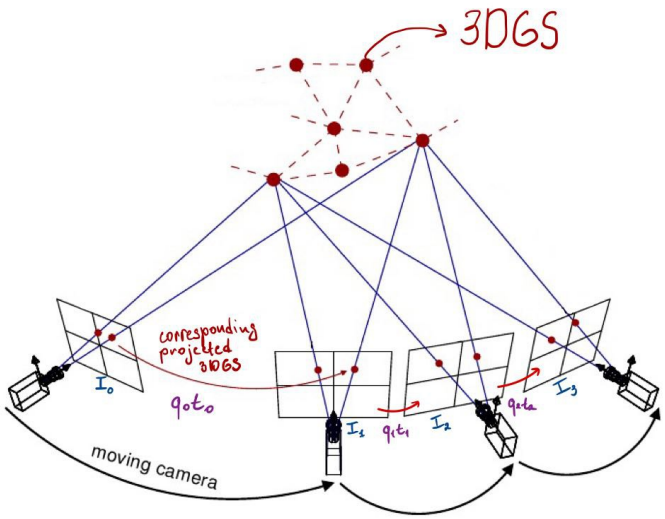

FIGURE 3.3: Relative transforms $(q_0, t_0), (q_1, t_1), ...(q_{n-1}, t_{n-1})$ between timestemps $I_0, I_1, ..., I_n$. Image adapted from [Sweeney, n.d.]

The centers of the Gaussians, denoted as μ, are adapted by translating them by the vector $\mathbf{t}$ and rotating them using the quaternion $\mathbf{q}$. This adjustment aligns μ with the camera's new viewpoint, ensuring that the Gaussian centers are correctly positioned in the transformed scene. Additionally, the rotation parameters of the Gaussians, indicated by $\mathbf{r}$, are updated according to $\mathbf{q}$ to maintain the correct orientation of the ellipsoids relative to the camera's altered perspective.

To avoid complexities associated with spherical harmonics during rendering process, we have limited our model to using only the first degree of spherical harmonics

for color representation. This decision simplifies the model by reducing it to managing basic RGB values, which significantly eases the learning process. Since spherical harmonics are view-dependent and defined in world space, they require adjustments when the camera orientation changes. By restricting our model to the first degree, we eliminate the need to rotate higher-order spherical harmonics, thereby enabling more efficient updates to these parameters.

These transformations ensure that our 3D Gaussians are not only precisely positioned but also maintain their visual consistency across various camera views.

3.3 Adjusted Poses with Global 3DGS

We have found the concept of utilizing a global 3D Gaussian Splattings framework, as demonstrated in the Fu et al., 2023, to be particularly beneficial. This global approach assists in adjusting the trajectory according to the camera views that have already been learned, enhancing the accuracy of the model over time.

In our implementation, we have developed our own version of the optimization pipeline to enhance this global strategy. For instance, we adopt a practice of resetting the opacity with each new frame. Additionally, we continue to densify the global 3DGS until the end of the process, a technique suggested in the original paper but not explicitly detailed.

A key question that arises in the context of global adjustments is whether to apply the adjustments solely to the new camera when it arrives or to all cameras retrospectively whenever a new frame is added. Fu et al., 2023 does not clearly specify this, and to address this ambiguity, we have implemented both approaches in our system. By doing so, we can compare their effectiveness directly in our experiments, providing concrete results that highlight the benefits and drawbacks of each method

We propose maintaining a global scene representation, denoted as Γ, which is continuously updated and extended with each new frame. This global representation Γ integrates information from all previous frames, providing a robust reference that helps in correcting and stabilizing the estimated poses over time. By comparing and adjusting the newly estimated relative poses against Γ, we can significantly reduce the error accumulation and align the trajectory more closely with the ground truth. This method ensures that each frame not only contributes to but also benefits from a cumulative and coherent global understanding of the scene, leading to more accurate and stable long-term navigation and mapping.

Optimization of global 3DGS

Upon establishing the relative poses between frames using local 3D Gaussian Splattings (3DGS), the next critical step involves updating the global scene representation to reflect these changes. This update process begins with the integration of the newly processed images I_t and I_{t+1} into the global 3DGS. To refine and enhance the accuracy of the global scene model, we undertake an optimization process that runs for N iterations, employing the densification strategy as outlined in the original 3DGS paper. This approach incrementally builds the density and detail of the global model, ensuring a comprehensive representation of the scene as more data becomes available.

An essential part of this updating process is resetting the opacity values for all the 3DGS in the global set after each frame is added. This reset is crucial as it allows us to prioritize Gaussian points that consistently appear across multiple images, thus reinforcing the contributions from elements that maintain visibility and relevance throughout the scene's evolution. Initially, the overfitted local 3DGS from the first

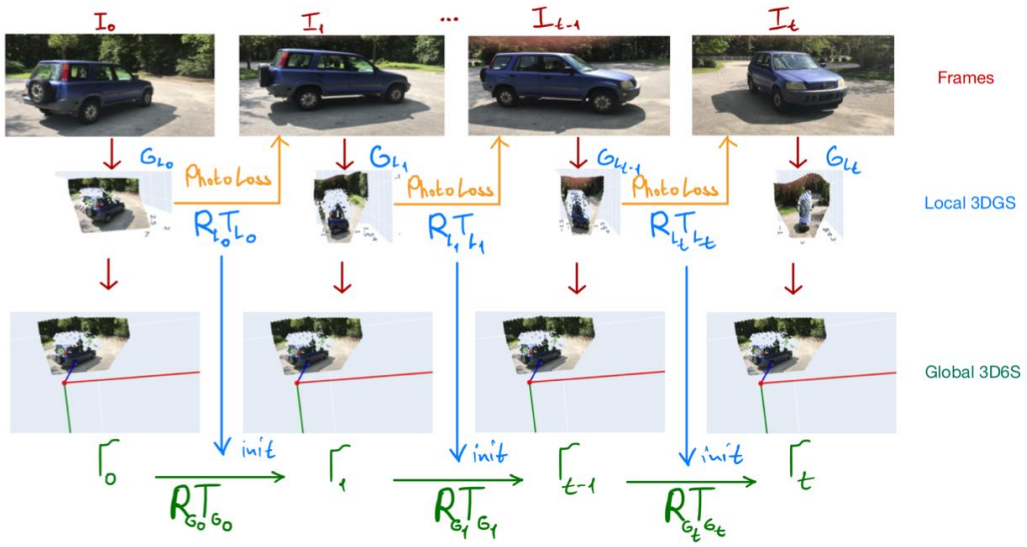

FIGURE 3.4: Overall optimization pipeline scheme. First we fit local 3DGS G_{L_t} for each frame I_t, then we estimate local relative transforms R_{L_t}, T_{L_t} using photometric loss. After, we optimize global 3DGS Γ_t with t cameras and adjust global relative transform R_{G_t}, T_{G_t} btw I_0 and I_t.

frame, G_0, serves as the seed for the initial point cloud, providing a baseline from which the global representation can grow.

Following the update of the global 3DGS, we revisit and adjust the relative poses between I_t and I_{t+1}. Initially, this iterative refinement may not seem significantly beneficial since the model is still heavily dependent on the early local transformations from the local 3DGS. However, as more frames, such as I_{t+2}, are integrated, the global model becomes increasingly robust. This enhancement reduces reliance on the initial local estimates and improves the stability and accuracy of the pose estimation process over time. As new frames continue to be added, the global representation builds a more comprehensive and reliable depiction of the scene, thereby reinforcing the effectiveness of the global adjustments in the pose estimation.

In contrast to local 3DGS, where Gaussians may not always align accurately with the scene's surface details, the global 3DGS develops a more coherent and aligned set of 3D Gaussian Splattings. This alignment is due to the iterative integration and optimization processes that incorporate surface details more accurately over time. As a result, the global scene representation not only captures the broad structural features of the environment but also refines these features to align more closely with the physical reality of the scene, thereby providing a robust foundation for accurate global pose estimation.

Progressive Densification

In our exploration of 3D scene reconstruction using global 3D Gaussian Splattings (3DGS), we have identified a recurring issue of underfitting when integrating new frames into the global model. This challenge becomes apparent when the global 3DGS does not densify sufficiently to accurately represent the dynamics introduced by newer frames. Consequently, even after optimizing the global set with a new frame I_t, the pose adjustment for the subsequent frame I_{t+1} may be hindered due to a suboptimal fit to the most recent frames, which significantly influence the pose

estimation process.

While the Fu et al., 2023 suggests using gradient accumulation to mitigate this issue, we propose an alternative approach to enhance the integration and fitting process. After estimating the relative transformation between I_t and I_{t+1}, we initiate the fitting of a local 3DGS for frame I_{t+1}, then apply the calculated transformation Θ_t to these Gaussians. To refine the integration, we implement a filtering process that retains only those 3DGSs with an opacity greater than 0.8. This threshold was determined experimentally to effectively highlight Gaussians that sit on the edges of objects—areas crucial for defining object boundaries and particularly susceptible to changes as the camera moves, capturing significant perspective shifts and structural changes.

By selectively integrating these high-opacity edge 3DGS into the global scene, we ensure that these critical features are emphasized, thereby accelerating the integration of new frames into the global model. This method not only enhances the fit of subsequent frames but also improves the overall temporal coherence of the scene reconstruction, ensuring that each new frame adapts more seamlessly into the global context.

Chapter 4

Data

4.1 Real World Data (CO3D)

FIGURE 4.1: Overview of the CO3D dataset featuring various objects from multiple views showed by the authors[Reizenstein et al., 2021].

CO3D[Reizenstein et al., 2021] is an object-centered dataset that features a diverse array of objects captured from various trajectories, providing a wide range of viewing angles and perspectives. Each object in the dataset is typically represented in a sequence of about 200 frames per video. This extensive collection allows for robust testing and evaluation of 3D reconstruction algorithms across different object types and motion dynamics.

The dataset not only includes RGB images but also provides masks for each frame, which are crucial for focusing the reconstruction process on the object of interest by filtering out background noise and irrelevant details. These masks enhance the precision of the reconstruction by ensuring that the algorithms are primarily processing the relevant object data.

In our research, we utilize the CO3D dataset as a key component of our testing framework. The real-world data provided by CO3D allows us to evaluate our approach under practical conditions. Additionally, to conduct a reliable evaluation, we use COLMAP-estimated camera poses as a proxy for ground truth. These poses offer a benchmark against which we can measure the accuracy of the camera poses recovered by our method. This is essential for validating the effectiveness of our pose estimation technique, especially in scenarios where exact ground truth data may not be available.

The CO3D dataset also includes metrics about quality of the videos, which provide insights into the visibility and clarity of the objects across different frames.

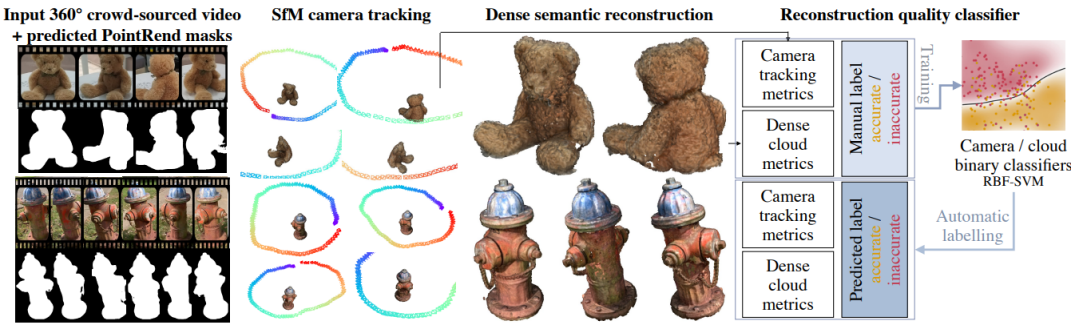

FIGURE 4.2: Example of C03D dataset images, mask and labeled camera poses using SFM methods given by the authors of the dataset[Reizenstein et al., 2021].

These metrics are valuable for selecting suitable video sequences for testing, ensuring that the data used in our experiments are of high quality and represent the challenges typical in real-world scenarios.

We have picked 4 video sequences from CO3D dataset and created aliases to refer to them in our work:

- **106_12650_23736** - outdoor camera sequence around car, we use alias **car**;
- **106_12648** - outdoor camera sequence around fire hydrant, we use alias **hydrant**;
- **188_20295_35748** - indoor camera sequence around plant, we use alias **plant**;
- **34_1404_4419** - indoor camera sequence around teddy bear, we use alias **teddybear**;

4.2 Synthetic Data

For the purposes of testing pipeline and to closely simulate conditions similar to those encountered in real datasets like CO3D, we have created a custom video sequence specifically adopted for our experimental needs. This dataset consists of sequence comprising 200 frames, which has a close structure to those sequence represented in CO3D[Reizenstein et al., 2021] dataset.

To ensure comprehensive testing, every eighth frame in sequence is designated as a testing frame. This systematic approach allows us to evaluate the performance of our reconstruction method at regular intervals, providing consistent checkpoints at which we can assess progress and fidelity. The selection of every eighth frame as a testing point helps maintain a realistic testing regimen that mirrors the periodic evaluation often required in practical applications.

In our experiments, we employ two separate configurations to handle depth estimation, which is crucial for the effectiveness of the 3D reconstruction process:

- in the first configuration, we use depth information that is directly rendered from a rasterizer. This ground truth data serves as a benchmark for maximum possible accuracy in depth estimation, providing a control setup against which we can measure the performance of other depth estimation methods;
- the second configuration utilizes predicted depth obtained from a depth estimation model, referred to as a Depth-Anything metric estimator[Yang et al.,

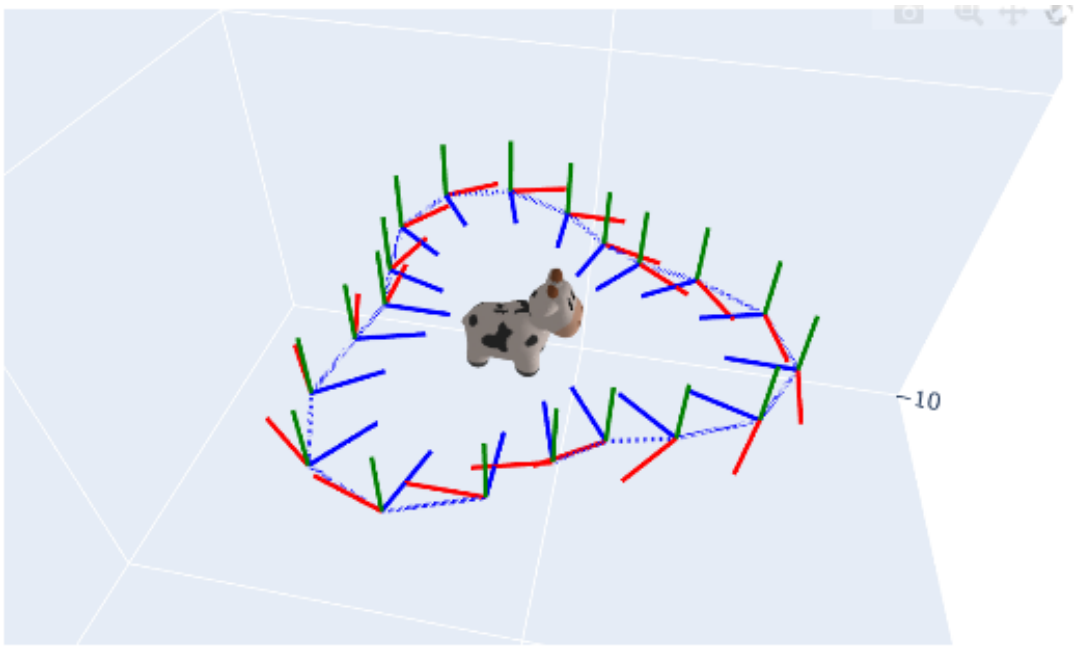

FIGURE 4.3: Example of custom synthetic dataset with moving trajectorie around the cow object. RGB axices represent camera's orientation, red is right vector, blue is forward and green is up.

2024]. This model is designed to predict depth maps from input images, simulating a more challenging and realistic scenario where exact depth information is not known a priori.

The dual configuration approach allows us to comprehensively evaluate our 3D reconstruction method under both ideal and realistic conditions. By comparing the outcomes from both ground truth and predicted depth data, we can better understand the strengths and limitations of our approach.

Chapter 5

Experiments

5.1 Implementation Details

Our solution is built using PyTorch[Paszke et al., 2017], a widely adopted framework for deep learning applications, which facilitates robust handling of tensor operations and gradient propagation essential for our optimization algorithms. To handle synthetic dataset generation, camera management, and the calculation of transformations between predicted and ground truth trajectories, we integrate the PyTorch3D[Ravi et al., 2020] library. This library provides specialized 3D functionalities that enhance our ability to manipulate and render complex 3D geometries efficiently.

For the implementation of 3D Gaussian Splattings, we utilize [Bernhard Kerbl, 2023], a Python repository designed for training and rendering 3D Gaussian Splattings, which has been extended to include differentiable Gaussian rasterizer[Bernhard Kerbl, 2023] written in CUDA[NVIDIA, Vingelmann, and Fitzek, 2020]. This rasterizer is crucial for achieving the high-performance computations required for real-time 3D rendering and adjustment of Gaussian parameters.

The pipeline for processing video data into 3D reconstructions involves several key steps:

1. **frame preparation:** videos are first split into individual frames, which are then resized to a uniform resolution using utilities from the OpenCV[Itseez, 2015] library.

2. **depth estimation:** each frame undergoes a process to estimate metrical depth[Yang et al., 2024], providing the data needed for generating point clouds and initializing the local 3D Gaussian models.

3. **local 3DGS initialization:** using the color and depth information from each frame, we construct a point cloud and initialize local 3D Gaussian Splattings. This step forms the basis for the detailed scene reconstructions that follow.

4. **pose adjustment and optimization:** modifications to our algorithm are applied to determine and refine the relative pose of each frame. To facilitate conversions between rotation matrices and quaternions—and ensure these transformations are differentiable—we employ the ROMA[Brégier, 2021] library, which is built on PyTorch and supports gradient-friendly operations.

Our algorithms are executed on an L4 Google Cloud machine, which provides the necessary computational power to handle of 3D scene reconstruction. Depending on the complexity of the scene and the resolution of the images, the GPU memory usage ranges from 1 to 3 GB per scene. Due to the computational limitations of our custom differentiable Gaussian rasterizer, the batch size during rendering is set to 1.

5.1.1 Data Preprocessing

CO3D

For our experimental validation, we utilize the CO3D dataset, which is known for its challenging camera pose changes, typically featuring orbital trajectories around objects in both indoor and outdoor settings. The dataset is ideal for testing our algorithm due to the dynamic nature of the camera movements and the variety of environments it includes. Each video sequence within the dataset comprises approximately 200-220 frames, from which every 8th frame is designated as test interpolation data for novel view synthesis.

To thoroughly assess different configurations of our algorithm and identify potential issues, we selectively choose three video sequences from the dataset, prioritizing those with the highest quality metrics as provided by the dataset authors. These sequences represent diverse settings and challenges, offering a comprehensive basis for evaluation.

Each selected video sequence comes with precalculated COLMAP[Schönberger and Frahm, 2016] trajectory cues, they are consentrated around world space origin and are normalized. The video data is processed in a sequence of 200 consecutive frames. For each frame, we apply a monocular depth estimator to generate depth maps. Additionally, we calculate masks using the depth information and RGB colors to isolate the foreground from the background, focusing our analysis on the primary objects of interest.

The dataset is structured in a format similar to that used in Neural Radiance Fields (NeRF)[Mildenhall et al., 2020], with each frame accompanied by its corresponding extrinsic and intrinsic camera parameters. To optimize processing efficiency and accommodate the lower resolution of depth maps, we resize each rectangular image to a resolution of 256x128 (width and height). This resizing not only aligns with the native resolution of the depth maps but also significantly reduces the computational load during the rendering process, making our experimental evaluations more efficient and manageable.

Cow Synthetic Sequences

We utilized the GT depth component of this dataset to verify the fundamental operations of our pipeline, including camera functionality, depth accuracy, and the process of unprojecting points from screen space to camera space. This testing is crucial for ensuring that our system is accurately interpreting and handling spatial data as intended. Additionally, the comparison with predicted depth data helps us evaluate the degradation of our method under conditions of noisy depth, providing insights into the robustness and limitations of our approach.

The training and testing splits for this artificial dataset are maintained consistent with those used for the CO3D dataset to facilitate a direct comparison and ensure uniformity in our evaluation approach. All images in this dataset are rendered at a resolution of 512x512. We chose square images because they simplify many aspects of the pipeline testing.

5.1.2 Training Pipeline and Parameters

Training Local 3DGS

The training process for local 3D Gaussian Splattings (3DGS) begins with initializing the camera in camera space with zero translation and identity rotation. Using this

initial camera setup, we unproject approximately 33,000 points from a 256x128 pixel grid into 3D space, forming the base for our optimization of the local 3DGSs.

FIGURE 5.1: Unprojected pcd in camera space for I_0 frame **car** sequence. Here axises represent camera orientation with blue being forward, red is right and green is up vectors.

For each set of local Gaussians, we conduct an optimization over 1,000 iterations using the standard 3DGS optimization pipeline. A key modification in our approach involves the timing of the densification process; we start to densify the Gaussian representations from the 100th iteration and perform additional densification steps at 600th iteration. We also do not reset opacity during optimization of local 3DGSs as it can harm next step of pose recovery with the gaussians that didn't catch up. This step in the pipeline is notably efficient, typically requiring only about 3-6 seconds of processing time on an L4 GPU.

FIGURE 5.2: Before and after fitting local 3DGS for I_0 frame **car** sequence.

Relative Pose Estimation with Local 3DGS

Following the optimization of local 3D Gaussian Splattings (3DGS), the subsequent step focuses on recovering the relative pose for the next frame, I_{t+1}. In this phase, we specifically target the optimization of the rotation quaternion $\mathbf{q}_t$ and the translation vector $\mathbf{t}_t$ over 500 iterations, employing a learning rate of 1×10^{-3}. During this step, we completely freeze the local Gaussians to eliminate any ambiguity that might arise in the pose prediction process, leading to more precise and reliable estimation of the relative transformations.

Using a higher learning rate in this context is advantageous for swiftly converging towards the true pose. A lower learning rate, while potentially offering finer adjustments, tends to result in the optimization process lingering near the initial guess, particularly if that starting point is far from the true pose.

Training Global 3DGS

The global scene representation is initially formed using the optimized local 3D Gaussian Splattings (3DGS) from the I_0 frame. This foundational set serves as the starting point for further refinements as more frames are processed. Upon successful recovery of the relative pose using the first two input frames, these frames are then incorporated into the ongoing optimization of the global representation.

The training regimen for the global scene is structured to grow in complexity and duration as additional frames are integrated into the model. Specifically, we start

with 1,000 iterations for the first two frames. As subsequent frames are added to the dataset, we increment the number of iterations by 10 for each new frame. This scaling approach ensures that as the global representation becomes more comprehensive with the inclusion of more frames, it receives proportionately more computational attention, allowing finer adjustments. By the time the 200th frame is processed, the global representation undergoes 3,000 iterations of optimization.

To further refine the trajectory and fit towards the end of the sequence, an additional intensive optimization phase is applied, involving 30,000 iterations using all available camera trajectories. This extensive training phase is crucial for aligning the global model closely with the complete set of observational data, enhancing the overall accuracy and stability of the reconstructed scene.

Camera Pose Adjustment with Global 3DGS
Once the global scene representation is updated with the I_t frame, the next crucial step involves refining the relative pose estimates initially predicted using local Gaussian Splattings. This refinement step is pivotal for enhancing the robustness of the pose adjustments within the reconstruction process. We implement two distinct settings for this pose refinement:

Single Pose Adjustment: In this setting, only the most recently predicted pose is adjusted, and then the process moves forward. This approach focuses on incrementally refining each new pose based on the latest global scene context, maintaining a continuous and immediate adjustment strategy.

Simultaneous Pose Adjustments: Alternatively, we adjust all previously predicted poses along with the Gaussian splattings. This comprehensive adjustment ensures consistency and coherence across all frames, aligning the entire sequence more accurately with the global model.

For the single pose adjustment, we optimize the pose for 300 iterations with a learning rate of 5×10^{-5} for both rotation and translation parameters. The lower learning rate is deliberately chosen to allow for finer convergence, leveraging the stability provided by the newly updated global scene to achieve a more precise alignment.

In the case of simultaneous adjustments across all poses, the optimization extends over 1,000 iterations, with the number of iterations increasing by 10 for each new frame added, mirroring the approach used for fitting the global scene. This extended and escalating iteration count allows the adjustments to become increasingly refined as more frame data and context accumulate within the global model.

Both adjustment strategies are designed to optimize the alignment of the predicted poses with the actual camera movements recorded in the sequences, thereby enhancing the overall accuracy and temporal consistency of the 3D reconstruction.

5.1.3 Evaluation Metrics

To accurately assess the effectiveness of our pose recovery techniques in 3D scene reconstruction, we rely on two principal metrics: Relative Pose Error and Relative Pose Error. These metrics are crucial for evaluating the precision of the pose estimations provided by our algorithms and are defined as follows:

$\mathbf{RPE_{rotation}}$: This metric measures the angular difference in rotation between the estimated pose and the ground truth. It is calculated using the formula:

$$\mathbf{RPE_{rotation}} = \arccos\left(\frac{\text{trace}(R_{\text{est}}^{-1} R_{\text{gt}}) - 1}{2}\right) \tag{5.1}$$

where R_{est} is the estimated rotation matrix, R_{gt} is the ground truth rotation matrix, and trace is the trace of a matrix. $\mathbf{RPE_{rotation}}$ condenses the angular discrepancies across all three dimensions of rotation into a single number by measuring the angle needed to rotate one orientation to align perfectly with the other. This single value summarizes the overall rotational error without needing to break it down by individual axes.

$\mathbf{RPE_{translation}}$: This metric quantifies the Euclidean distance between the estimated translation and the ground truth translation, represented by the formula:

$$\mathbf{RPE_{translation}} = \|\mathbf{t}_{\text{est}} - \mathbf{t}_{\text{gt}}\| \tag{5.2}$$

where $\mathbf{t}_{\text{est}}$ and $\mathbf{t}_{\text{gt}}$ are the estimated and ground truth translation vectors, respectively.

These metrics are particularly informative in the context of pose recovery as they provide a detailed insight into the accuracy of both rotational and translational components of the camera's movement, which are critical for aligning and integrating sequences in 3D scene reconstructions.

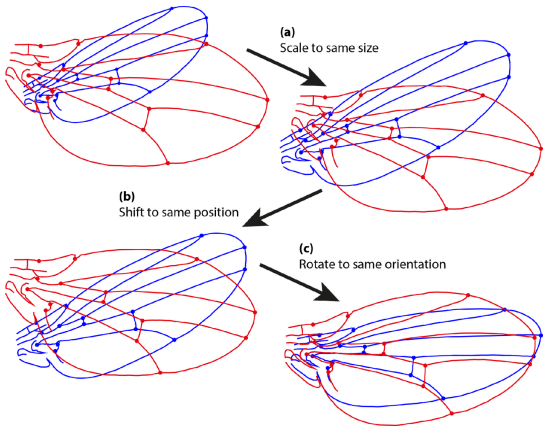

FIGURE 5.3: Procrustes analysis applied to 2 unaligned set of points. a) scales 2 sets to same size; b) shifting to same position; c) aligning orientation. Image provided by [*Procrustes analysis* n.d.]

To compute these metrics accurately, especially when predicted depth is involved, we employ Procrustes analysis. This statistical analysis method is used to determine the optimal alignment of two sets of points (in this case, the estimated poses versus the ground truth) by allowing only for rotation, translation, and scaling transformations. The Procrustes analysis is essential because it removes any discrepancies that arise due to the coordinate system differences or scaling issues between the predicted and actual camera poses. This alignment is particularly necessary when dealing with predicted depth data, which can introduce systematic biases or scale mismatches in the estimated poses relative to the ground truth.

By applying Procrustes analysis, we ensure that our evaluations of $\mathbf{RPE_{rotation}}$ and $\mathbf{RPE_{translation}}$ are not adversely affected by external factors unrelated to the actual performance of our pose estimation algorithms, thus providing a fair and consistent basis for assessing the accuracy of our 3D reconstruction approach.

To evaluate the quality of images generated through novel-view synthesis, we utilize two standard metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Both metrics provide insights into the fidelity of the synthesized images compared to the original images.

PSNR: This metric measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The formula for PSNR is:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) \tag{5.3}$$

Here, MAX_I is the maximum possible pixel value of the image (e.g., 255 for 8-bit images), and MSE is the mean squared error between the original and synthesized images. PSNR is expressed in decibels (dB), with higher values indicating better image quality.

SSIM: The Structural Similarity Index measures the perceived quality of an image by comparing its structural information, luminance, and contrast with those of a reference image. The formula for SSIM is:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5.4}$$

In this equation, x and y are the original and synthesized images respectively, μ_x and μ_y are their average pixel values, σ_x^2 and σ_y^2 are their variances, σ_{xy} is the covariance, and c_1 and c_2 are constants used to stabilize the division with weak denominators. SSIM values range from -1 to 1, with higher values indicating greater similarity and therefore higher image quality.

Both PSNR and SSIM are crucial for assessing the visual and structural integrity of synthesized views, helping to ensure that the generated images are both accurate and visually pleasing when compared to their real-world counterparts.

5.2 Comparing Pose Recovery Strategies

5.2.1 Local 3DGS

Local Gaussians play a critical role in the initial stages of 3D scene reconstruction, offering distinct advantages and facing particular challenges in their implementation.

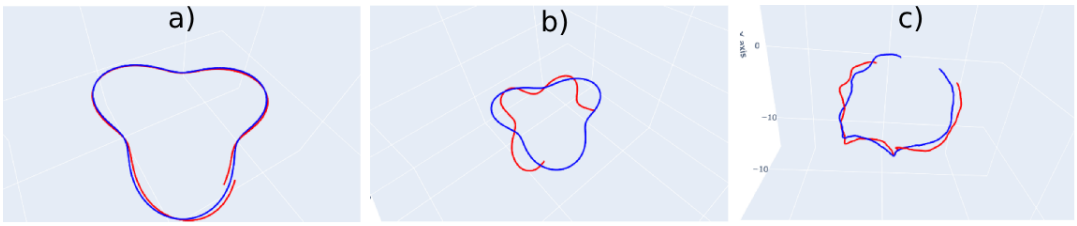

FIGURE 5.4: Recovered trajectories for video sequences using only local 3DGS; a) cow synthethic with ground truth depth; b) cow synthethic using estimated depth; c) teddybear_co3d sequence; blue line is ground truth trajectory and red is recovered;

One of the key advantages of using local Gaussians is their ability to provide a robust initial estimation for each frame independently. This approach ensures that each estimation starts with a fresh representation, which is not influenced by potential errors from previous frames. This independence is particularly beneficial in scenarios where the dataset may contain anomalies or inconsistencies, as it allows each frame to be processed based on its own merits without carrying forward errors.

Despite these advantages, local Gaussians are not without their drawbacks. The primary issue they face is drift over time, which occurs because the estimations for each frame are made independently without adjusting for the overall sequence. This lack of sequence-wide adjustment can lead to cumulative errors, especially when a single frame's relative pose is estimated inaccurately due to depth map inconsistencies or other issues. Such errors are not self-correcting and can propagate through subsequent frames, negatively impacting the fidelity and accuracy of the end reconstruction.

Furthermore, while local Gaussians are beneficial for avoiding the propagation of errors, this same feature can be a disadvantage as it prevents the system from learning from past errors and improving incrementally across the sequence. Each new set of local Gaussians is determined without the propagation from previous frames, which might otherwise help in adjusting and refining the approach based on past outcomes.

5.2.2 Local and Global 3DGS

To address the limitations of local Gaussians and enhance the robustness of relative pose estimation, we incorporate a concept of global Gaussians into our workflow. This approach provides two potential estimations for each new frame: one derived from local Gaussians and another from the global Gaussians. This dual estimation strategy aims to leverage the strengths of both local and global approaches to improve overall accuracy and stability.

Despite their potential benefits, global Gaussians also present challenges that can affect the reconstruction process. Similar to the drift seen with local Gaussians, global Gaussians can accumulate errors, particularly if the initial frames are not aligned accurately. Since the global model builds upon each addition, early mistakes can propagate and magnify, affecting the entire sequence. In scenarios where the scene contains high levels of detail, global Gaussians may sometimes underfit to new frames. This underfitting occurs because the global model, while comprehensive, may not adjust quickly enough to accommodate highly detailed or rapidly changing elements within the scene. This lag can result in a less accurate representation of newer frames, especially if the scene complexity increases.

5.2.3 Opacity Filtering

To enhance the efficiency of global 3D Gaussian Splattings (3DGS) fitting, we experimented with integrating new Gaussian splattings from the local 3DGS of the forthcoming frame directly into the global scene. This method aimed to accelerate convergence for upcoming frames by leveraging the immediate updates provided by the local Gaussians.

Introducing local Gaussians from the next frame into the global model before the comprehensive global fitting process provides a significant speed boost in convergence. This preemptive update allows the global model to incorporate new data points earlier in the sequence, enabling quicker adjustments to changes and new information presented by subsequent frames. However, this approach introduces certain challenges, primarily the potential bias from the local 3DGS. Local representation is optimized for specific frames without the broader context of the sequence, which might not always align with the global scene's cumulative data. In some scenarios, this can be beneficial, providing fresh insights or corrections to the global

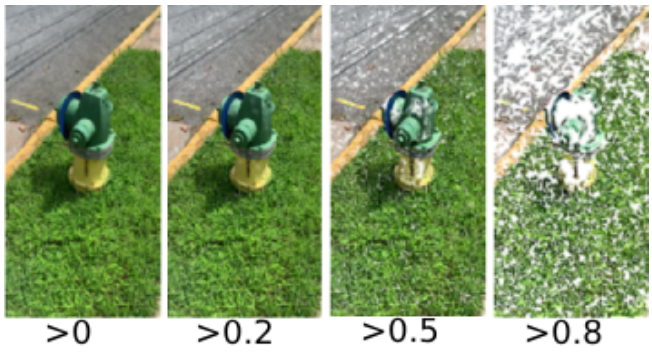

FIGURE 5.5: Filtering 3DGS with different opacity treshholds

model. In others, it may lead to discrepancies, where the local optimization priorities conflict with global accuracy and consistency.

To mitigate the risk of overfitting the global model to a particular frame and potentially compromising the fidelity of previous frames, we implemented a selective filtering strategy. Instead of integrating all new Gaussians from the local 3DGS, we selectively added gaussian splattings with opacity > 0.8 which approximately 5% of the local 3DGS, with a preference for those positioned at the edges of objects. Edge gaussian splattings typically contain critical boundary information that is vital for accurate scene reconstruction and help maintain the structural integrity of the model across transitions between frames

5.2.4 Simultaneous Optimization of Camera Poses and Global 3DGS

In our efforts to enhance the accuracy and cohesion of the trajectory in 3D scene reconstruction, we experimented with a strategy that involves simultaneously optimizing the global 3DGSs and camera pose parameters within the same processing loop. This approach aims to create a more integrated and dynamically updated model by adjusting both the camera poses and the shape parameters of the 3DGSs concurrently.

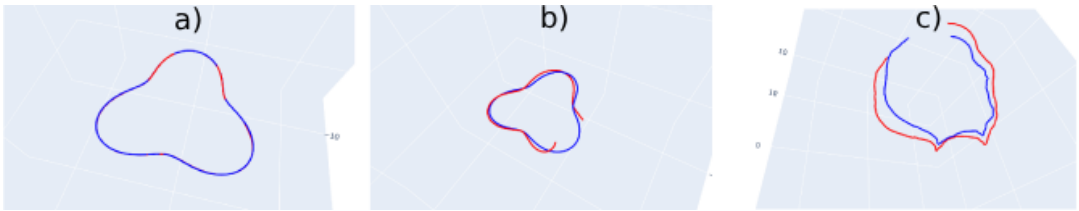

FIGURE 5.6: Recovered trajectories for video sequences using local and global 3DGS with simultaneous optimization; a) cow synthethic with ground truth depth; b) cow synthethic using estimated depth; c) teddybear_co3d sequence; blue line is ground truth trajectory and red is recovered;

The primary advantage of this method is the continuous adjustment of all elements in the model, which can lead to a more cohesive and accurate trajectory over the entire sequence. By simultaneously refining the camera poses and the 3DGSs, the model can better accommodate shifts and changes in the scene, potentially leading to more reliable reconstructions. This integration ensures that updates to the camera poses are immediately reflected in the Gaussian adjustments, fostering a more synchronized evolution of the model.

However, this approach also presents significant challenges, primarily due to the sensitivity of the model to the optimization parameters. One of the main drawbacks is that the continuous adjustment of camera poses can prevent the 3DGSs from settling into a stable configuration that accurately reflects the scene. If the camera parameters are adjusted too aggressively, it may lead to a scenario where the 3DGSs do not have sufficient time to accurately fit the data from the estimated frames, resulting in a less accurate reconstruction.

Moreover, this method is highly sensitive to the learning rates applied to the camera poses. To mitigate excessive movement and instability in the camera adjustments, we set the learning rates for rotations and translations to a relatively low value of 5×10^{-5}. This cautious approach helps minimize rapid shifts in camera positioning, allowing for more gradual and considered integration of frame data into the global model.

5.2.5 Results

In this section, we provide a results of the performance metrics for pose recovery and novel-view synthesis across different strategies and datasets. This evaluation helps in understanding the effectiveness of each approach in accurately estimating camera poses and generating high-quality synthesized views. By comparing the results from various strategies and datasets, we can identify the strengths and weaknesses of our methods.

TABLE 5.1: Comparison of $\mathbf{RPE_{rot}}$ metrics for different strategies across video sequences. $\mathbf{RPE_{rot}}$ units of measure are degrees.

Metric	Setting	cow+gt	cow+pred.	car	hydrant	plant	teddybear
$\mathbf{RPE_{rot}} \downarrow$	local only	0.074	0.637	0.677	**0.217**	**0.218**	**0.341**
	global	0.065	0.501	**0.647**	0.380	0.407	0.366
	global+all	0.106	**0.348**	0.731	0.296	0.468	0.332
	global+add	**0.054**	1.041	0.652	0.352	0.363	0.370
	global+add+all	0.108	0.427	0.657	0.319	0.478	0.367

TABLE 5.2: Comparison of $\mathbf{RPE_{trans}}$ metrics for different strategies across video sequences

Metric	Setting	cow+gt	cow+pred.	car	hydrant	plant	teddybear
$\mathbf{RPE_{trans}} \downarrow$	local only	0.020	0.169	0.561	**0.443**	**0.526**	**0.426**
	global	0.007	0.107	0.571	0.826	2.252	0.666
	global+all	0.005	**0.092**	0.592	0.536	1.182	0.743
	global+add	0.005	0.336	0.581	0.862	2.837	0.552
	global+add+all	**0.005**	0.118	**0.519**	0.457	1.162	0.623

Here is some information about experiment naming and what they mean:

local: utilizes only the local 3DGS for each frame without integrating information across the sequence. This approach starts fresh with each new frame and does not accumulate any knowledge from previous frames.

global: begins with local 3DGS for initial estimates and then incorporates these estimates into a global model for further adjustment.

TABLE 5.3: Comparison of **SSIM** metrics for different strategies across video sequences

Metric	Setting	cow+gt	cow+pred.	car	hydrant	plant	teddybear
$\mathbf{SSIM_{train}} \uparrow$	local only	0.980	0.937	0.626	0.761	**0.840**	0.893
	global	0.991	**0.966**	0.606	0.755	0.796	0.889
	global+all	0.988	0.964	0.575	0.759	0.829	**0.899**
	global+add	**0.993**	0.951	**0.743**	0.766	0.782	0.888
	global+add+all	0.988	0.965	0.706	**0.782**	0.835	0.894
$\mathbf{SSIM_{test}} \uparrow$	local only	0.955	0.909	**0.226**	**0.239**	**0.415**	0.517
	global	0.984	0.909	0.217	0.204	0.327	**0.590**
	global+all	**0.990**	**0.930**	0.223	0.207	0.326	0.512
	global+add	0.988	0.878	0.183	0.211	0.209	0.558
	global+add+all	0.990	0.916	0.197	0.209	0.324	0.531

TABLE 5.4: Comparison of **PSNR** metrics for different strategies across video sequences

Metric	Setting	cow+gt	cow+pred.	car	hydrant	plant	teddybear
$\mathbf{PSNR_{train}} \uparrow$	local only	28.705	21.074	20.685	23.301	25.386	28.339
	global	34.029	26.207	20.601	23.281	24.461	28.205
	global+all	32.777	**26.954**	19.973	23.358	25.317	**28.656**
	global+add	**35.002**	24.033	**21.914**	**23.452**	24.046	28.277
	global+add+all	32.874	26.471	21.398	23.396	**25.791**	28.653
$\mathbf{PSNR_{test}} \uparrow$	local only	24.919	15.334	10.379	**15.857**	**15.809**	14.639
	global	31.278	17.876	9.869	13.427	12.191	**16.742**
	global+all	33.787	**21.058**	10.110	14.192	12.445	14.087
	global+add	32.387	13.818	10.283	13.726	14.000	16.087
	global+add+all	**33.919**	19.115	**10.663**	14	12.500	15.005

global + all: extends the global strategy by adjusting not only the most recent frame's pose but also retrospectively refining all previously estimated poses based on the latest global model insights.

global + add: after obtaining the initial relative pose using local 3DGS, this method involves fitting local 3DGS for the subsequent frame and adding them to the global model. This is done selectively with an opacity threshold of 0.8, focusing on integrating primarily those 3DGS that likely represent critical structural details.

global + add + all: combines the strategies of adding 3DGS selectively and adjusting all frame poses within the global model.

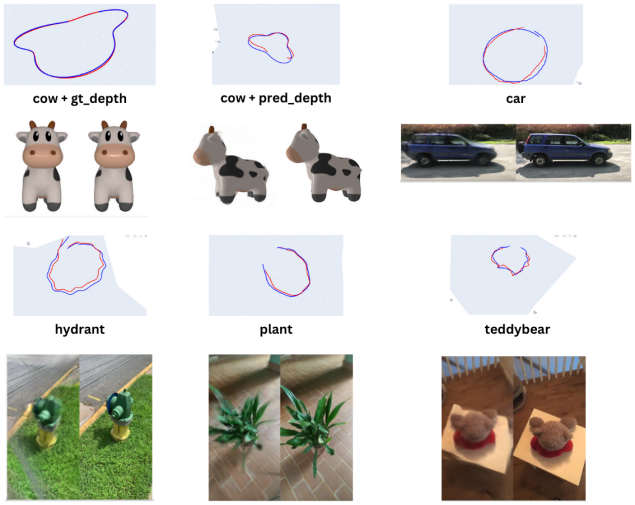

FIGURE 5.7: Recovered trajectories for each video sequence and novel-view synthesis. Blue trajectories are ground truth and red ones are estimated. On left novel-view images generated from model and on right side are ground truth images.

Chapter 6

Conclusions

6.1 Experiments Summary

Our comprehensive evaluation of different 3D reconstruction strategies using both synthetic data with known ground truth depth and real-world datasets provides insights into the effectiveness and challenges of each approach.

On synthetic datasets, the strategies that incorporated more complex logic, such as the integration of global 3DGS and adjustments to previous camera poses, generally showed an improvement in quality metrics like PSNR and Reletive Pose Error. These results suggest that these more sophisticated strategies can effectively leverage the precise and consistent depth maps available in synthetic data to enhance the reconstruction quality.

However, the performance on real-world data presented a contrasting scenario. In almost all real-world examples, strategies that relied primarily on local 3DGS demonstrated better results in both pose recovery and novel-view synthesis. This indicates a significant challenge in maintaining a global 3DGS representation in environments where data may not be as consistent or accurate as synthetic setups.

The global 3DGS approach, while theoretically advantageous for integrating information across frames, proves to be tricky in practice. It tends to accumulate errors rapidly, leading to misaligned trajectories from the onset of the reconstruction process. This accumulation highlights the delicate balance required in managing global information, where early errors can disproportionately skew later results.

Strategies that involve adding local 3DGS to the global model or simultaneously estimating camera poses for previous frames are particularly sensitive to several factors. The learning rates for rotation, translation, and even the Gaussian parameters themselves must be finely tuned to match the specific characteristics of the scene. For instance, in scenes where depth information for distant objects is noisy and tends to converge towards similar values, the appropriate learning rates can vary significantly. This variability can make a set of parameters suitable for one scene but entirely unsuitable for another.

These findings underscore the complexities involved in extending local reconstruction strategies into more global contexts. The effectiveness of sophisticated strategies that rely on the integration of historical data depends heavily on the accuracy and consistency of the input data, as well as on precise parameter tuning.

Despite its simplicity and the inherent noise challenges, the local 3D 3DGS approach continues to demonstrate decent accuracy in both pose recovery and novel-view synthesis. This method's resilience and effectiveness underscore its utility as a straightforward yet potent solution for 3D scene reconstruction. Local 3DGS operates independently frame by frame, which not only mitigates the risk of error accumulation seen in more complex, global strategies but also ensures that each frame is processed in isolation.

6.2 Limitations

One of the most significant challenges in reconstructing 3D scenes using 3DGS is the recovery of accurate camera poses. The effectiveness of the strategies we have explored is highly contingent upon the availability of precise and temporally consistent depth maps. The accuracy of the depth maps plays a critical role, as all our metrics and the subsequent quality of the reconstruction heavily depend on their reliability.

Our strategies presume that the camera movement between consecutive frames is minimal, allowing for substantial overlap and shared information, which is crucial for effective pose estimation and novel-view synthesis. This assumption holds well in controlled environments, such as those using synthetic data where the overall camera trajectory is complex yet the relative movement between frames is kept small. However, this assumption becomes problematic in real-world datasets like CO3D, which feature video sequences with rapid and varied camera movements against fast-changing backgrounds. The divergence from the assumption of minimal movement introduces significant challenges in maintaining the accuracy and effectiveness of these reconstruction strategies.

From a computational perspective, the simplicity of local 3DGS does offer some advantages. For instance, the local 3DGS strategy requires about 30 seconds on an L4 GPU to recover the pose for each frame, which is relatively efficient. However, more complex strategies, which might provide better integration and potentially more accurate reconstructions, require significantly more processing time. Depending on the complexity of the scene and the strategy employed, pose recovery can take between one to 1 and 1.5 minutes per frame. Consequently, for long video sequences, these methods could take hours to complete, posing a substantial limitation in terms of time efficiency and practical applicability in real-time or near-real-time scenarios.

6.3 Contribution

This thesis intoduces a potential baseline for reconstructing scenes using 3DGS without reliance on pose priors. We have systematically tested various strategies across different scenarios to evaluate their reliability and to identify possible improvements over simpler methods. This exploration has led to a deeper understanding of the capabilities and limitations of 3DGS in different operational contexts.

By implementing different reconstruction strategies, we assessed how changes in the complexity of the approach affect the overall accuracy and efficiency of the reconstruction process. Each strategy was applied under varied conditions to measure how well they perform in terms of pose recovery and novel-view synthesis, especially in scenarios where no prior information about the camera's movement is available. This thorough testing helps in point out the most effective techniques for different types of data and movement patterns, providing a nuanced view of how 3DGS can be optimized for real-world applications.

From a temporal perspective, even the worst-case scenarios showed promising results: for a video sequence of 200 frames, the scene reconstruction process took about 2-3 hours to complete using our most complex strategies. This duration is considerably favorable when compared to classical methods, which could take between 8-10 hours just for the pose recovery phase. Such efficiency not only demonstrates the practical viability of using 3DGS for scene reconstruction but also highlights

its potential to significantly reduce the processing time without compromising the quality of the output.

6.4 Future Work

A promising direction for future work involves the experimentation with adaptive and scheduled learning rates to enhance the convergence of pose estimations in 3D scene reconstruction. The potential for improved convergence through dynamically adjusted learning rates could significantly enhance the accuracy of pose recovery, particularly in challenging scenarios where the camera dynamics are complex. Furthermore, developing a methodology to map the quality or characteristics of depth maps directly to an initial learning rate setting could streamline the optimization process. This approach would adjust the learning rate based on the expected difficulty or error characteristics of the input depth maps, potentially leading to faster and more stable convergence across diverse datasets.

Another area for future exploration is the development of advanced depth map preprocessing strategies and the integration of depth map consistency losses. These methods would aim to enhance the reliability and accuracy of depth information, particularly for distant objects in the scene that tend to have higher measurement errors. By implementing preprocessing techniques that refine depth data or by incorporating consistency losses that enforce logical depth relationships across frames, it may be possible to reduce the impact of erroneous depth readings on the training process. Such improvements could lead to more accurate global models and reduced cumulative error in long sequence reconstructions.

An important step for future research would be to conduct a comprehensive comparison of the current methods with existing state-of-the-art (SOTA) techniques for 3D scene reconstruction. This comparative analysis would provide valuable insights into how the proposed strategies using 3D Gaussian Splattings stack up against other leading approaches in terms of accuracy, efficiency, and robustness. Understanding the strengths and weaknesses of our methods relative to the broader field can help identify specific areas for improvement and potential integration of other successful techniques.

Finally, improving the training pipeline for global 3DGS represents a critical area for development. While global 3DGS theoretically offer superior estimates due to their comprehensive scene integration, practical implementations often struggle with error accumulation and model drift. Developing a more robust training pipeline that can effectively manage and mitigate these issues would be invaluable. This could involve new strategies for error correction, enhanced integration techniques for incoming frame data, or more sophisticated models that better capture the temporal dynamics of the scene.

Bibliography

Alex Yu Sara Fridovich-Keil, Matthew Tancik Qinhong Chen Benjamin Recht Angjoo Kanazawa (2021). "Plenoxels: Radiance Fields without Neural Networks". In: *ArXiv* abs/2112.05131.

Barron, Jonathan T et al. (2021). "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864.

Bernhard Kerbl Georgios Kopanas, Thomas Leimkühler George Drettakis (2023). "3D Gaussian Splatting for Real-Time Radiance Field Rendering". In: *ArXiv* abs/2308.04079.

Brégier, Romain (2021). "Deep Regression on Manifolds: a 3D Rotation Case Study". In.

Charatan, David et al. (2023). "pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction". In: *arXiv*.

Cheng, Zezhou et al. (2023). *LU-NeRF: Scene and Pose Estimation by Synchronizing Local Unposed NeRFs*. arXiv: 2306.05410 [cs.CV].

Chung, Jaeyoung, Jeongtaek Oh, and Kyoung Mu Lee (2023). "Depth-regularized optimization for 3d gaussian splatting in few-shot images". In: *arXiv preprint arXiv:2311.13398*.

Franke, Linus et al. (2024). "TRIPS: Trilinear Point Splatting for Real-Time Radiance Field Rendering". In: *Computer Graphics Forum*. Wiley Online Library, e15012.

Fu, Yang et al. (2023). "COLMAP-Free 3D Gaussian Splatting". In: arXiv: 2312.07504.

Galliani, Silvano, Katrin Lasinger, and Konrad Schindler (2015). "Massively parallel multiview stereopsis by surface normal diffusion". In: *Proceedings of the IEEE international conference on computer vision*, pp. 873–881.

Gaurav Chaurasia Sylvain Duchene, Olga Sorkine-Hornung and George Drettakis (2013). "Depth synthesis and local warps for plausible image-based navigation". In: *ACM Transactions on Graphics* TOG.

Georgios Kopanas Julien Philip, Thomas Leimkühler and George Drettakis (2021). "PointBased Neural Rendering with Per-View Optimization". In: *Computer Graphics Forum 40*.

Itseez (2015). *Open Source Computer Vision Library*. https://github.com/itseez/opencv.

Jeong, Yoonwoo et al. (2021). "Self-calibrating neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5846–5854.

Keetha, Nikhil et al. (2024). "SplaTAM: Splat, Track Map 3D Gaussians for Dense RGB-D SLAM". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kopanas, Georgios et al. (2021). "Point-Based Neural Rendering with Per-View Optimization". In: *Computer Graphics Forum*. Vol. 40. 4. Wiley Online Library, pp. 29–43.

Lin, Chen-Hsuan et al. (2021). "Barf: Bundle-adjusting neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5741–5751.

Mildenhall, Ben et al. (2020). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*.

Müller, Thomas et al. (2022). "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM transactions on graphics (TOG)* 41.4, pp. 1–15.

NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek (2020). *CUDA, release: 10.2.89*. URL: https://developer.nvidia.com/cuda-toolkit.

Paliwal, Avinash et al. (2024). "CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians". In: *arXiv preprint arXiv:2403.19495*.

Paszke, Adam et al. (2017). "Automatic differentiation in PyTorch". In.

Procrustes analysis (n.d.). https://en.wikipedia.org/wiki/Procrustes_analysis. 12 May 2024.

Ravi, Nikhila et al. (2020). "Accelerating 3D Deep Learning with PyTorch3D". In: *arXiv:2007.08501*.

Reizenstein, Jeremy et al. (2021). "Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction". In.

Schönberger, Johannes Lutz and Jan-Michael Frahm (2016). "Structure-from-Motion Revisited". In.

Schönberger, Johannes Lutz et al. (2016a). "A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval". In: *Asian Conference on Computer Vision (ACCV)*.

Schönberger, Johannes Lutz et al. (2016b). "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*.

Snavely, Noah, Steven M Seitz, and Richard Szeliski (2006). "Photo tourism: exploring photo collections in 3D". In: *ACM siggraph 2006 papers*, pp. 835–846.

Sweeney, Chris (n.d.). "Theia Multiview Geometry Library: Tutorial & Reference". In: ().

Tewari, Ayush et al. (2022). "Advances in neural rendering". In: *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library, pp. 703–735.

Truong, Prune et al. (2023). "Sparf: Neural radiance fields from sparse and noisy poses". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4190–4200.

Wang, Zirui et al. (2022). *NeRF–: Neural Radiance Fields Without Known Camera Parameters*. arXiv: 2102.07064 [cs.CV].

Wenjing Bian Zirui Wang, Kejie Li-Jia-Wang Bian Victor Adrian Prisacariu (2022). "NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior". In: *ArXiv* abs/2212.07388.

Wu, Guanjun et al. (2023). "4d gaussian splatting for real-time dynamic scene rendering". In: *arXiv preprint arXiv:2310.08528*.

Xu, Qingshan and Wenbing Tao (2019). "Multi-scale geometric consistency guided multi-view stereo". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5483–5492.

Yang, Lihe et al. (2024). "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data". In.

Yugay, Vladimir et al. (2023). *Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting*. arXiv: 2312.10070 [cs.CV].

Yurkova, Kate (2023). "A Comprehensive Overview of Gaussian Splatting". In.

Zhang, Kai et al. (2020). "Nerf++: Analyzing and improving neural radiance fields". In: *arXiv preprint arXiv:2010.07492*.

Zhang, Yanshu et al. (2024). "Papr: Proximity attention point rendering". In: *Advances in Neural Information Processing Systems* 36.

Zhiwen Fan Panwang Pan, Peihao Wang-Yifan Jiang Hanwen Jiang Dejia Xu Zehao Zhu Dilin Wang Zhangyang Wang (2023). "Pose-Free Generalizable Rendering Transformer". In: *https://arxiv.org/pdf/2311.13398*.