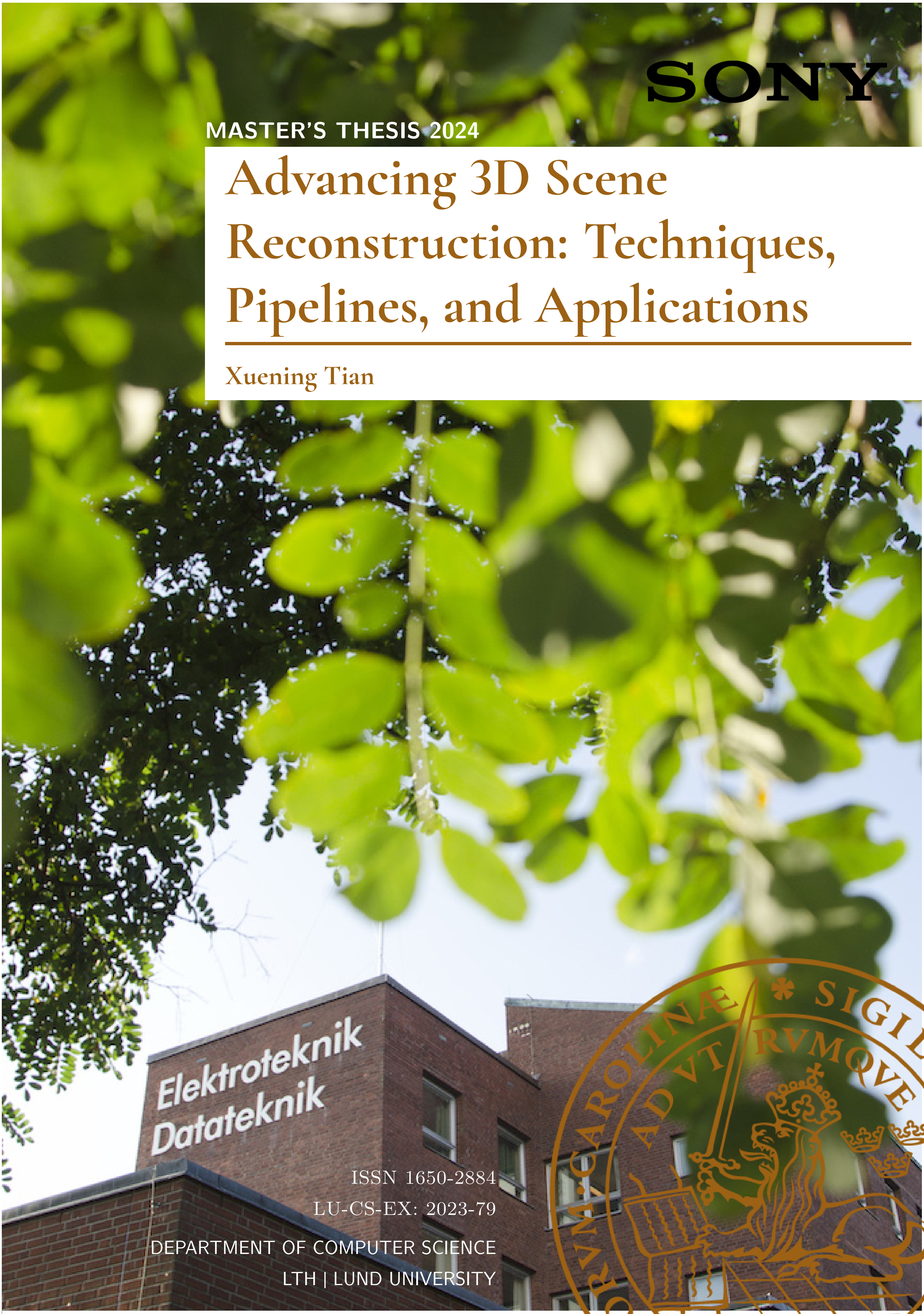


MASTER'S THESIS 2024

Advancing 3D Scene Reconstruction: Techniques, Pipelines, and Applications

Xuening Tian

ISSN 1650-2884
LU-CS-EX: 2023-79
DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2023-79

Advancing 3D Scene Reconstruction: Techniques, Pipelines, and Applications

Avancera 3D-scenrekonstruktion: tekniker, rörledningar och tillämpningar

Xuening Tian

Advancing 3D Scene Reconstruction: Techniques, Pipelines, and Applications

Xuening Tian
xu6035ti-s@student.lu.se

June 12, 2024

Master's thesis work carried out at Sony Nordic.

Supervisors: Shangxia Huang, sangxia.huang@sony.com
Michael Doggett, michael.doggett@cs.lth.se

Examiner: Günter Alce, gunter.alce@design.lth.se

Abstract

This master thesis is carried out at Sony Nordic which aims to investigate state-of-the-art methods on 3D Scene reconstruction and explores the potential utilization in the industry. The project addresses challenges in reconstructing complex scenes using both static camera setups which consider scenarios with freely moving rigid objects and freely moving cameras. Throughout the research, several key questions were answered, resulting in a robust pipeline including image capture, camera calibration, foreground segmentation, camera estimation, and model training. The reconstruction utilizes technologies such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting. The result of this work demonstrates the feasibility and highlights the potential challenges of 3D reconstruction under various camera settings. Additionally, we propose several applications that could benefit from these advancements, depending on the scenarios.

Keywords: 3D reconstruction, Computer Vision, Neural Radiance Fields, Gaussian Splatting, Rendering, Deep Learning

Acknowledgements

I am here to express my deepest appreciation to all those who helped me to complete this thesis work. A special gratitude is given to my supervisor Sangxia and his colleague Roberto, who contributed to the comprehensive guidance and necessary help in development with their extraordinary experience and insights. I am also thankful to my supervisor Michael who helped me to coordinate my thesis structure and writing with enough encouragement during the four months of thesis work.

I am also thankful to the Vision and AI Systems Department at Sony Nordic which has offered me an opportunity to conduct both my internship and master thesis project during the year. Same for the fellow colleagues that I have worked with, I have learned so many things from all of you, not only on work skills but also the suggestions for my future career path.

In the end, I need to give a big thanks to both my family and friends who encouraged me and helped me get through the frustrating time when I was finishing my thesis alone. Especially the friends I met from ESN Lund. I feel so lucky to be a part of this amazing organization. You enriched my student life a lot in the last period of my study here.

Contents

Glossary

NeRF	Neural Radiance Fields 3DGS
3D Gaussian Splatting AR	Augmented Reality
VR	Virtual Reality
XR	Mixed Reality
SOTA	State-of-the-Art
AI	Artificial Intelligence
SfM	Structure-from-Motion
MLP	Multi-Layer Perceptron
CUDA	Compute Unified Device Architecture
SDF	Signed Distance Function
NeuS	Neural Surface Reconstruction
OpenCV	Open Computer Vision Libary
COLMAP	the pipeline of Structure-From-Motion and Multi-View Stereo
Instant-NGP	Instant Neural Graphic Primitives
RMSE	Root Mean Square Error
MSE	Mean Square Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
SIFT	Scale-Invariant Feature Transform
SAM	Segment Anything
GUI	Graphical User Interface
IoU	Intersection over Union
SuGaR	Surface-Aligned Gaussian Splatting
FoV	Field of View
DoF	Degree of Freedom

Chapter 1
Introduction

This master thesis is carried out at Sony Nordic which aims to investigate state-of-the-art (SOTA) methods on 3D Scene reconstruction and explore the potential utilization in the industry. The project addresses challenges in reconstructing complex scenes using both static camera setups which consider scenarios with freely moving rigid objects and freely moving cameras. During this process, several key research questions were answered and a robust pipeline from image capture to final result was built. The findings of this work aim to enhance the understanding and application of 3D scene reconstruction in different scenarios.

1.1 Background

In recent years, the field of computer vision has made significant strides in understanding and interpreting the rich visual information present in images [30, 7] and videos [15, 22]. One of the fundamental goals of computer vision is to endow machines with the ability to perceive and comprehend implicit and explicit information from complex scenes in the way of human vision. 3D Scene Reconstruction is one of the exciting and challenging research areas. Compared to the discrete grids on 2D images, 3D representations of objects such as points cloud or mesh are able to provide more spatial and semantic information that infers the 3D structure, depth, and relationships within a scene.

At the same time, the high demand of Extended Reality (XR), encompassing Virtual Reality (VR) and Augmented Reality (AR), even makes this area catch more attention. It has seen rapid growth in XR applications spanning from entertainment to education and industry, which raises a need for highly accurate 3D environment understanding. The high demand for robust VR/AR applications such as XR training systems, for example, holds immense promise for enhancing learning and skill development in various domains. Central to the effectiveness of these systems is the ability to reconstruct complex 3D scenes and accurately track objects with seamless interaction.

1.2 Motivation

AI technology has made significant progress in robotics, auto-driving, and decision-making. Each one of these areas requires AI with an extraordinary understanding of the surroundings. Therefore, accurately reconstructing the 3D world from the receiving information becomes a very important standard to assess this capability of AI.

The traditional 3D reconstruction always perform under the ideal assumption that the camera will freely move around a single object[37] or the entire scene[12]. However, due to the complexity and occlusion during the capture, the 3D reconstruction in an industrial application becomes much more challenging. Starting from this point, this research provides us with a better image of which is the most challenging part of the 3D scene reconstruction workflow. During the experiment, we utilized both static and dynamic camera setups which offer distinct advantages that can complement each other in practical applications.

Dynamic cameras, which we define as camera that moves around the object to capture multi-view images, have the capability to capture the environment from multiple angles. This mobility allows for comprehensive scanning of objects, providing detailed and complete 3D models. The ability to capture different perspectives is particularly beneficial in understanding the geometry and spatial relationships within a scene.

Conversely, we define static cameras as those fixed on specific position. This type of camera offers precise tracking and monitoring capabilities. When positioned strategically, they can provide continuous, stable, and high-resolution data from a fixed viewpoint. This stability is advantageous for zero-shot level applications where initial or sporadic tracking without prior information is necessary. Static cameras excel in environments where the focused objects are moving, such as in motion capture for animation, real-time tracking in augmented reality (AR), and surveillance systems. They ensure consistent data capture without the need for recalibration, which is essential for accurate and reliable tracking over time.

The integration of both static and dynamic camera systems can significantly enhance 3D object reconstruction in real-world scenario. For instance, in an XR training system, combining both setups allows for robust and flexible data acquisition. In a training simulation for industrial maintenance, static cameras can track the trainee's movements and tools, while dynamic cameras can capture the environment's details, creating an immersive and accurate virtual model. This dual approach ensures comprehensive coverage and accurate data, enhancing the overall effectiveness and realism of the simulation.

1.3 Aim and Scope

This project will be carried out with the perspective of developing a whole working pipeline from capturing the video from single camera to reconstructing the 3D structure of target objects. It will be assumed there are two different camera setups. In one circumstance, there will be a static camera with known intrinsic parameters and the target object will keep free moving within the field of view to simulate the context under a multi-cam system. In the second setup, like traditional 3D scanning, the objects will be placed on the plane instead and captured by the camera from different angles

By the proposed experiment setup and target application scenario, the following research questions will be answered:

- How can arbitrary rigid objects be reconstructed from static cameras, and what are the primary challenges involved in this process?
- What are the necessary components in the pipeline from captured images to the final reconstructed object?
- Which reconstruction method is considered most effective among existing algorithms?
- What are the potential uses of this technique?

1.4 Contribution to the State of Knowledge

The first question aims to investigate the methodologies and difficulties associated with reconstructing 3D models of arbitrary rigid objects using static camera setups. It will explore challenges such as camera extrinsic estimation, feature detection and matching, and handling occlusions or complex object geometries. The second question seeks to identify and outline the essential stages and components involved in the reconstruction pipeline, starting from capturing images to generating a final 3D model. Key components may include camera calibration, preprocessing, feature extraction, structure-from-motion (SfM) for camera pose estimation, and mesh reconstruction techniques.

The third question focuses on evaluating and comparing different reconstruction algorithms including novel-view-synthetic based methods like NeRF, 3D Gaussian Splatting, and Photogrammetry based methods to determine the most suitable approach for generating accurate and detailed 3D meshes from reconstructed point clouds or from the images. For the fourth question, we will explore the practical uses and benefits of using reconstructed 3D scenes in various areas.

In the last problem, we will focus on developing methodologies and metrics to assess the quality and suitability of reconstructed scenes for XR training. Evaluation criteria may include the fidelity of the synthetic novel view images compared to the real-world objects, time cost, and stability of the reconstruction quality under different capture trajectories.

1.5 Sustainable Development Goals

The Sustainable Development Goals (SDGs), known as the Global Goals, were adopted by all United Nations members in 2015, aiming to create peace and prosperity for people and the planet [28]. The 17 SDG goals cover a wide range from climate change to public health, education, economics, etc. The contributions of our project align with the following goals

Quality Education

XR and AI technologies have shaped education with new digital experiences. Some institutions that adopted these technologies acknowledge that XR and AI applications have significantly boosted the quality of their education [43]. Our project allows for the creation of highly detailed and accurate models of real-world objects, environments, or historical artifacts. In an educational XR setting, students can interact with these models in virtual or augmented reality, exploring details that are either too small, too distant, or too fragile to engage with physically.

Figure 1.1: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all

Decent Work and Economic Growth

A fast-growing economy requires a large number of skilled industrial workers. Our works make highly detailed and interactive models of machinery, equipment, or work environments used in XR training scenarios become possible. This can dramatically enhance the training process for employees, especially in sectors like manufacturing, healthcare, and construction with faster learning curves, reduced training costs, and fewer workplace accidents.

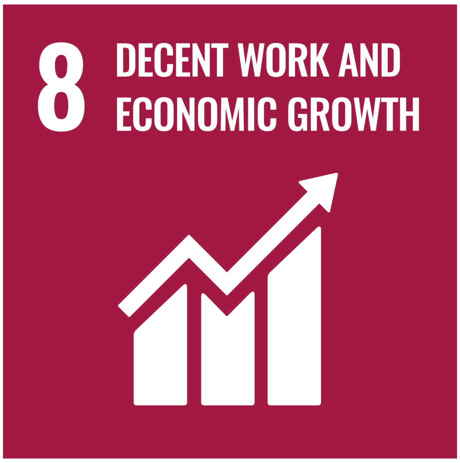


Figure 1.2: Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

1.6 Ethics

The usage of a single-camera or multi-camera system for 3D reconstruction also raises several ethical questions that we should consider.

Privacy and Copyright Concerns

One of the biggest concerns of our work is the privacy issue. As this technology will capture and store large amounts of videos and reconstruct highly accurate models of private properties and public areas, there is a risk of infringing on individuals' privacy. On the other hand, the 3D objects reconstructed from real commercial objects may raise the copyright issue when utilized in the applications.

Impact on Employment

As 3D scene reconstruction technologies become more advanced, there could be implications for employment, particularly in fields like surveying, construction, and architecture. While these technologies can enhance productivity and create new opportunities, they might also lead to job displacement. Addressing these changes responsibly, such as through workforce retraining and education, is important.

Chapter 2
Theory

This chapter serves as an overview of theoretical frameworks that have been applied in this thesis project. Structure from Motion (SfM) is the most critical part of the entire workflow, offering an explicit estimation of the camera position and orientation. A correct SfM result is the prerequisite for the subsequent tasks. Following this, we will explain the basic mechanism of Neural Radiance Field (NeRF) and the rendering technology it relies on. To retrieve an explicit 3D object representation, we will also present a novel surface reconstruction algorithm alongside its associated

2.1 Structure from Motion

2.1.1 Camera Model

The pinhole model represents a fundamental camera model that simplifies the process of light projection from a 3D object onto an image plane. As can be seen from the right side of Figure 2.1, we refer to this system as a camera coordinate system since the origin of this coordinate system is located at the camera center (the pinhole) $C = (0, 0, 0)$. The image plane is typically positioned at $z = 1$, with its normal vector aligned parallel to the z-axis. Consider a point X located at (X_1, X_2, X_3) in 3D space. The corresponding projection point x on the image plane can be determined by inferring that $x = (X_1/X_3, X_2/X_3, 1)$ [16].

2.1.2 Camera Parameter

In general, the camera parameter can be classified into intrinsic parameter and extrinsic parameter [16]. The intrinsic parameters of a camera define its internal characteristics and how it projects a 3D scene onto a 2D image. These parameters are specific to the camera itself and remain constant as long as the camera's internal configuration does not change. For example, maintaining consistent settings on the zoom lens and aperture. Usually, the intrinsic

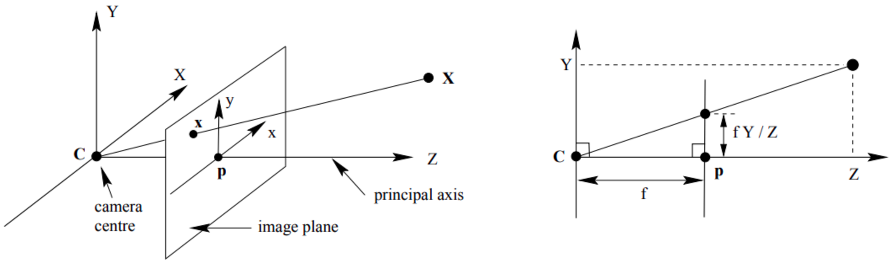


Figure 2.1: Pinhole camera geometry. C is the camera centre and p the principal point[16]

parameter can be represented as a matrix K:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 0 \end{bmatrix} \tag{2.1}$$

where f_x, f_y are the focal lengths, c_x and c_y are refer as the coordinates of the principal point or optical center. s is represented as pixel skew which is used to correct for tiled pixels.

The extrinsic parameter describes the position and orientation of the camera coordinate relative to the world coordinate system. In another word, these parameters will be changed whenever the camera is moved. In general, the extrinsic parameters are made up of a 3×3 rotation matrix R and a 3×1 transition vector t

$$[R \quad t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{2.2}$$

In a pinhole camera model, a 3D point in space will be simply projected onto the image plane via a projective transformation, preserving straight lines in both 3D space and the resulting 2D image [16]. However, modern consumer cameras normally equip multiple lenses to achieve desired photographic effects, leading to inevitable distortion in the captured images. To accurately model a camera in real life, it is vital to consider the distortion parameter.

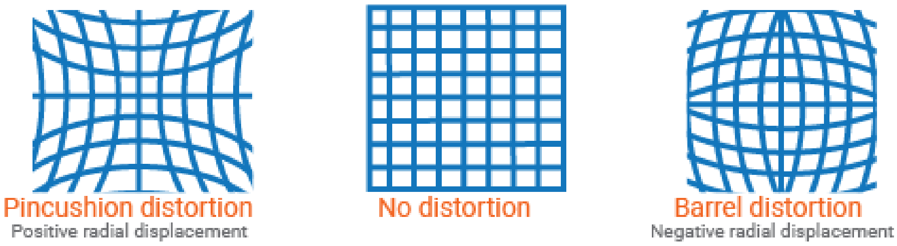


Figure 2.2: Example of different types of radial distortion. Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center [17].

Using radial distortion as an example, which follows a non-linear function whereby straight lines positioned closer to the lens periphery exhibit a greater degree of bending. This results in two distinct types of radial distortion, each associated with specific signs of displacement, as depicted in figure 2.2. The process of undistorting the image can be modeled as:

$$x_u = \begin{pmatrix} d(r_d)x_d \\ d(r_d)y_d \\ 1 \end{pmatrix} \tag{2.3}$$

where $x_d = (x_d, y_d, 1)$ is the distorted point and $r_d = \sqrt{x_d^2 + y_d^2}$ is the distance to the principal point. d is usually a polynomial function where the coefficients $k_1, k_2, k_3....$ are considered as distortion parameter.

2.1.3 Triangulation

Triangulation is one of the most crucial processes that aim to find the unknown 3D point X under the assumption that the correspondence between image pairs and the camera parameter is known. The core of this question is finding the solution of the equation:

$$\lambda_i x_i = P_i X_i, i = 1, ..., n \tag{2.4}$$

where $P = K[R \quad t]$ is the known camera parameter that we explained in the last subsection. λ is the unknown scale or depth. At least two matching image points that are projected from the same 3D point are needed to solve this problem by Direct Linear Transformation (DLT). In most cases, there is no exact solution due to the noise from computation and measurement but searching for the optimal solution with homogeneous least squares. Many optimization strategies can be chosen for this homogeneous least squares question such as Singular Value Decomposition (SVD) and lower–upper (LU) decomposition [33].

2.2 Neural Radiance Fields

Neural Radiance Fields (NeRF) was published in 2020 [26] and has become the game-changer in the novel-view-synthesis area by its outstanding performance in tackling the complex scene and innovative concept of neural radiance field. Compared to the mesh-based representation of the scene [3, 8], NeRF has more flexibility and is even more desirable for rendering specific objects like volumetric cloud. Compared to other volumetric-based methods, NeRF performs impressively with higher quality and less time and space complexity on high-resolution images.

The key contribution of NeRF can be summarized as designing an approach that uses Multi-Layer Perceptron (MLP) networks as a 5D neural radiance field to represent static scenes. The MLP takes 5D coordinate (x, y, z, θ, ϕ) as input where (x, y, z) represent a spatial point in 3D space and (θ, ϕ) represent the viewing direction. The output is an emitted color $c = (r, g, b)$ and volume density σ. The overview of the method can be seen in Figure 2.3. During the rendering for a specific viewpoint, sets of points will be sampled in a march along the ray from each pixel of the image. Those points will be concatenated with their respective 2d view directions and forwarded to the neural network as input. The final pixel color is composed by accumulating the RGB color and densities by volume rendering techniques. The rendering loss for this training is based on the discrepancy between the reconstructed pixel and the ground truth pixel [26].

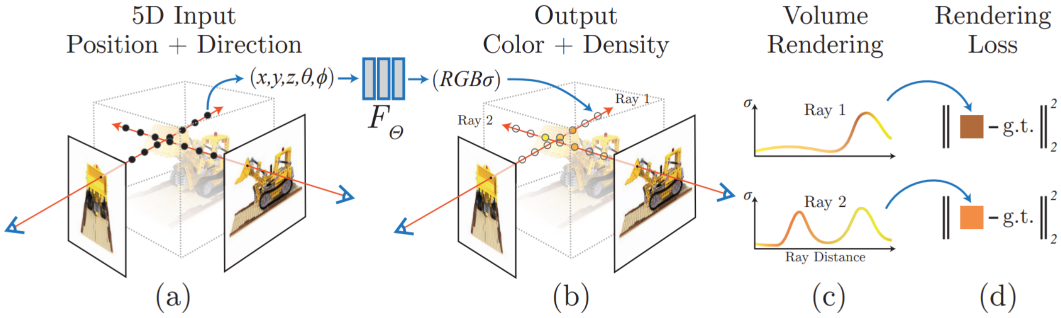


Figure 2.3: Overview of NeRF scene representation and differentiable rendering procedure [26]

2.2.1 Volume Rendering

Volume rendering plays a crucial role in the implementation of NeRF by modeling how rays behave as they traverse through particles and culminate in the final color. Generally, four types of interactions happen between the photons and particles. Photons may either be absorbed by the particle or scattered outwards from the previous path, according to the light scattering theory [1], which leads to a reduction in the radiance intensity. Conversely, the particles themselves may emit light and the in-scattering happens where the photons in other directions may coincide with photons in the current direction, thereby augmenting the radiation intensity along the current light path. The following equation was introduced in the original paper of NeRF [26]:

$$\begin{aligned} C(r) &= \int_{t_f}^{t_n} T(t)\sigma(r(t))c(r(t),d)\,dt, \\ \text{where } T(t) &= \exp\left(-\int_{t_n}^{t} \sigma(r(s))\,ds\right) \end{aligned} \tag{2.5}$$

In this equation, $r(t) = o + td$ denotes the camera ray. The light scattering is all simplified into $\sigma(x)$ and interpreted as the differential probability of a ray terminating at an infinitesimal particle at location x. $T(t)$ denotes the accumulated transmittance along the ray from near bound to position t, which can be interpreted as the probability the ray travels from t_n to t without hitting any other particles.

To efficiently estimate the integral, a stratified sampling from each interval was also applied.

The following equation describes the estimation from N discrete samples:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) c_i,$$
$$\text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \tag{2.6}$$

In the equation, $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples

2.3 Neural Surface Reconstruction

NeRF gave us a novel approach to implicitly representing 3D scenes. Nevertheless, explicit representation is still indispensable for most of the practical application. While some variations [27, 9, 29] of NeRF have made strides in surface reconstruction, it remains challenging due to the absence of surface constraints in the representation. Inspired by both volume rendering and Signed Distance Function (SDF), Neural Surface Reconstruction (NeuS) outperforms the state-of-the-art in high-quality surface reconstruction and receives an impressive result on objects and scenes with complex structures and self-occlusion [38].

Building upon prior research, NeuS2[39] was introduced in 2022, which overcomes the slow training speed of NeuS and brings in support for multi-view dynamic objects. Inspired by Instant-NGP [27], Neu2 implements multi-resolution hash tables to accelerate the training process of the neural surface representation. Additionally, to reduce computational complexity further, a novel second-order derivative is also presented to leverage CUDA parallelism.

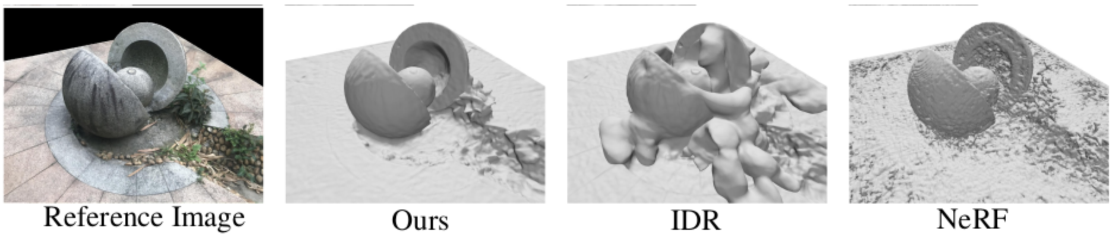


Figure 2.4: Comparison on surface reconstruction [38] among NeuS, IDR [42] and NeRF

2.3.1 Signed Distance Function

Signed Distance Function (SDF) is a function that accepts a position as input and returns the distance to the nearest point on an object's surface. Usually, if the position lies inside the shape, the distance will be positive. If outside, it's negative. SDF has been widely used in computer graphics and geometry processing to represent geometric shapes implicitly. One of the key advantages of it is the ability to describe complex scenes with mathematical expression and it is differentiable. For example, the zero level of SDF $f(x) = 0$ represents the surface of the geometric object. Many surface reconstruction algorithms including NeuS2 are built based on this theory.

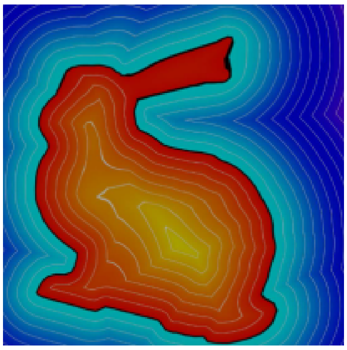


Figure 2.5: The 2d signed distance field to the Stanford bunny[32]

2.4 Related work

COLMAP

COLMAP [34] is a robust end-to-end pipeline implementation for Structure-from-Motion (SfM) and Multi-View Stereo (MVS), which integrates a wide range of control options during the reconstruction based on both GUI and command-line. Meanwhile, the paper behind this work also contributes a novel geometric verification strategy and innovation on the incremental reconstruction process.

Instant-NGP

As one of the most successful variants of NeRF, Instant-NGP proposed a multi-resolution hash encoding that splits one MLPs for representing Neural Graphic primitives into smaller, more efficient MLPs [27]. This multiresolution structure also allows for an excellent parallelism on modern GPUs. By implementing the whole system with the CUDA kernel, Instant-NGP accelerates the reconstruction speed for hours to seconds.

3D Gaussian Splatting

Different from treating the whole scene as a neural radiance field, 3D Gaussian Splatting (3DGS) represents the scene as a set of 3D Gaussians initialized from the points cloud [20]. The Gaussians are then projected onto 2D, which is known as Splatting, and rendered using a fast, differentiable rasterizer. This approach allows for real-time rendering of high-quality novel views of the scene, even on consumer-grade GPUs. The method is evaluated on several established datasets and is shown to achieve state-of-the-art results in terms of both quality and speed. While it excels at rendering, extracting the scene's surface from these Gaussians for further editing and manipulation has been a challenge.

Surface-Aligned Gaussian Splatting

Surface-Aligned Gaussian Splatting (SuGaR) addresses the challenge of surface extraction from 3D Gaussian by introducing a regularization term during optimization [14]. This regularization term encourages the Gaussians to align with the scene's surface, making subsequent mesh extraction easier. The method then efficiently samples points on the visible surface and employs Poisson reconstruction to generate a detailed mesh within minutes on a single GPU. Additionally, SuGaR offers an optional refinement step to bind Gaussians to the mesh, enabling high-quality rendering and editing capabilities.

OpenCV

OpenCV (Open Source Computer Vision Library) is a free, open-source library widely used for computer vision and machine learning tasks. It provides a comprehensive set of tools and algorithms for camera calibration, real-time image and video processing, analysis, and manipulation [2].

Chapter 3

Methods

In this chapter, we will give a brief overview of the structure of our pipeline and the experiment setup. Then, the details of each component will be explained.

3.1 Data Overview

Static Camera with Arbitrary Rigid Objects

At the beginning of this project, the videos for single rigid objects were collected for testing. From easy to hard, three different objects with clear features on their appearance were chosen, as shown in Figure 3.1. The first object we picked is a soccer ball because of the simple geometry, clear appearance such as stripes and conspicuous logo on its surface. The second object is a keyboard which has more complex geometry but still hold the clear features on each keys. The last one we chose is a helmet which has more complexity in geometry shape and features compared to the soccer or the keyboard.

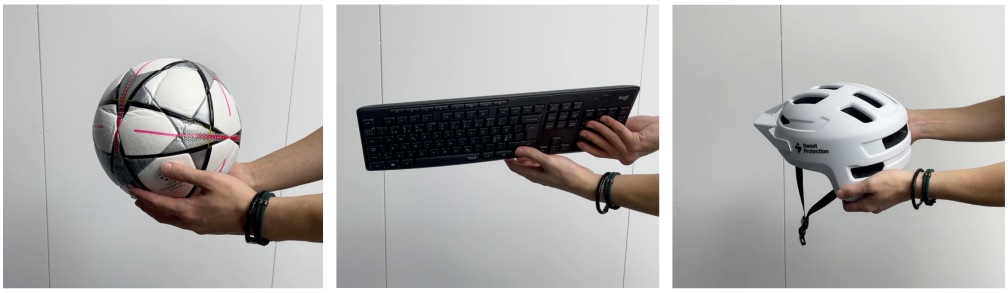


Figure 3.1: Several Objects we selected for reconstruction

All videos were captured by an iPhone 12 mounted on a tripod with the fixed 1× focal length. The assistant flipped the objects in front of the camera to expose as many facets of the object as possible and tried to avoid occlusion caused by the hands.

Static Objects with Moving Camera

Following the aforementioned experiment, we proceeded to capture the videos with objects positioned in the center while the camera moved around them. Since we lack stabilization equipment, in this scenario, the camera should be held at a slow pace to avoid motion blur. Furthermore, we varied the height and angle of movement around the objects to enhance the comprehensiveness of the captured trajectory 3.2.

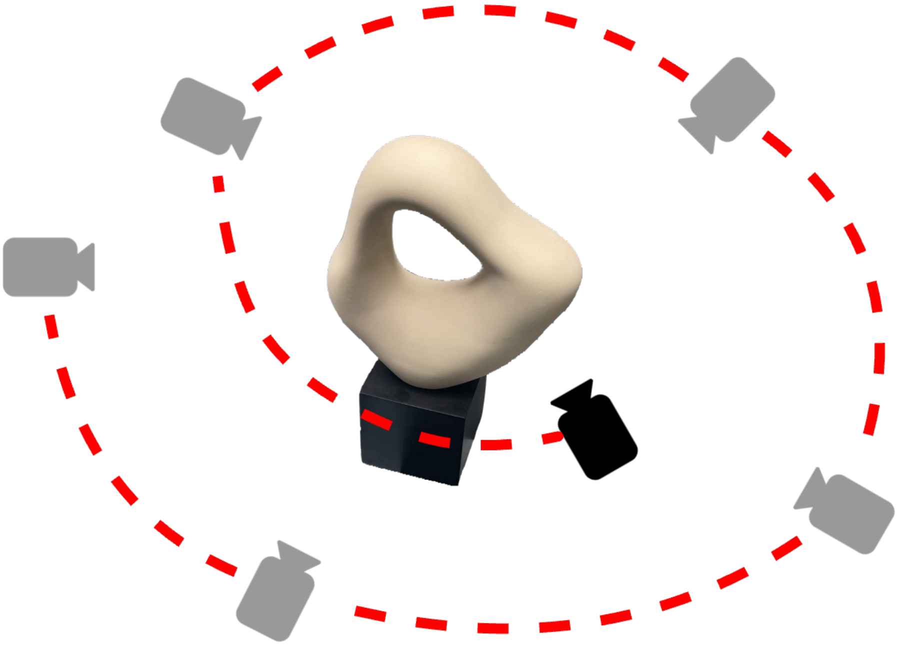


Figure 3.2: The video capture with the moving camera

Static Complex Scene with Moving Camera—Taking Medical Training as an Example

In most industrial scenarios we are faced with the whole scene instead of a single rigid object. To increase the robustness and adaptability of our system, We recreated a medical training scenario to increase the complexity of the reconstruction. In this task, an operation table was placed with different varieties of medical instruments. To mitigate potential reflections, the table is covered by a white tablecloth. For better control of the camera intrinsic parameter, we use a Sony Alpha 7C mark II camera with 24-70mm zooming lens. To keep the intrinsic parameter unchanged, we chose the manual focus and fixed focus length, as depicted in Figure 3.3.

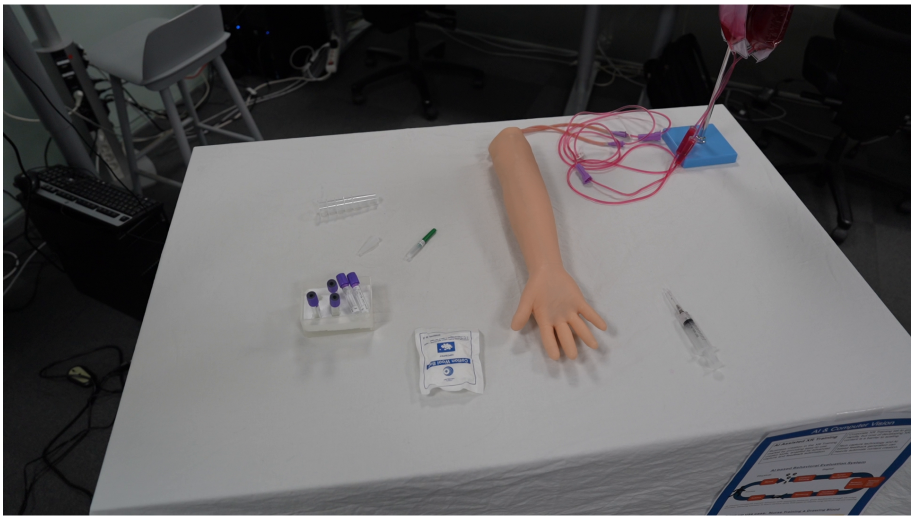


Figure 3.3: The medical operation table we set to simulate the venipuncture

3.2 Pipeline Overview

In this section, we will give an overview of our pipeline developed for 3D scene understanding and reconstruction, which integrates advanced computational techniques at each stage of the data processing and equips the ability to manage both static and dynamic objects under varying camera conditions (see Figure 3.4).

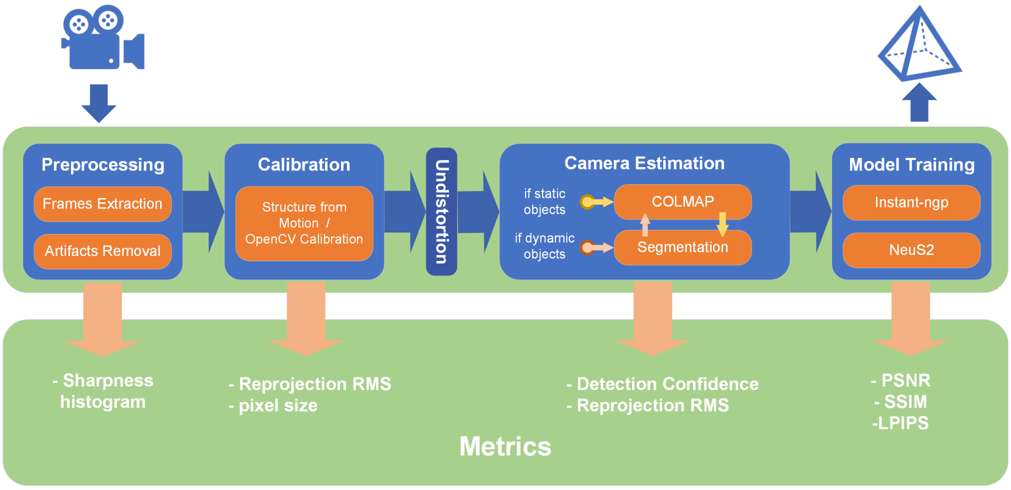


Figure 3.4: The end-to-end pipeline of our works

The first stage of the pipeline involves preprocessing, where relevant frames are selected from video streams or multiple images and cleaned to remove noise and other visual artifacts like motion blur which could impair accuracy for the following task.

Camera calibration is performed to retrieve intrinsic parameters which are essential for correcting the image distortion from the camera model and recovering the real-world dimension.

This involves using either incremental approaches to refine camera parameters over successive iterations or using a chessboard mark to track the feature with OpenCV.

Camera estimation varies depending on whether the scene objects are static or dynamic and whether the camera is moving or static. For static objects with a moving camera, COLMAP is used for photogrammetric reconstructions to capture diverse data points, which aid in accurate structural mapping. When dealing with dynamic objects and a static camera, initial segmentation is employed to effectively isolate and track object movements. Subsequent adjustments in the scene's reconstruction are made based on observed movements.

The model training component utilizes cutting-edge neural network techniques for scene representation and rendering. Instant-NGP employs neural graphics primitives for rapid scene representation, allowing real-time rendering. NeuS2, a surface reconstruction model, enables an effective explicit representation of the scene.

During each stage of the pipeline, validated metrics will be given to assess the quality of the intermediate results. In preprocessing, sharpness histograms evaluate image quality. Calibration accuracy is gauged by reprojection RMS. For camera estimation, detection confidence and reprojection error measure precision. Finally, the fidelity of the rendered results is evaluated using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) between the synthesized novel view and the ground truth.

3.3 Pre-processing

Image pre-processing is a set of operations to enhance the quality of images for the following processing purposes [13]. This typically includes Noise Filtering, Color Balance, and Grayscale Conversion. In our pipeline, the captured video is processed to extract images, eliminating motion blur and removing the repeated frames as part of the pre-processing phase.

3.3.1 Motion Blur Removal

The motion blur is usually caused by abrupt movement or vibration of the camera during the exposure time. In the domain of image processing, the precision of feature extraction techniques, particularly the Scale-Invariant Feature Transform (SIFT)[24], is pivotal for successful calibration and camera pose estimation. Removing motion blur is therefore crucial for the SIFT algorithm to accurately detect and describe local features across varying scales.

In this work, the sharpness of each frame is computed by the variance of the result obtained from Laplacian operator and visualized in the histogram. Laplacian operator is defined as the second derivative of the pixel intensity, which reflects the edges in an image[2]. As expressed in formula 3.1, where $\nabla^2 I$ denotes the Laplacian of the image I, μ is the mean of the Laplacian values, and N is the total number of pixels in the image. This gives an overview of motion blur throughout the whole video clip and a threshold for filtering can be assigned by clicking on the histogram.

$$\mathrm{Var}(\nabla^2 I) = \frac{1}{N}\sum_{i=1}^{N}\left(\nabla^2 I(x_i) - \mu\right)^2, \tag{3.1}$$

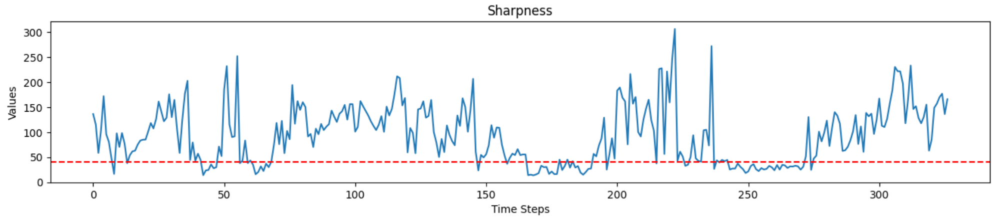


Figure 3.5: Statistic of the sharpness of the recorded helmet

3.4 Camera Calibration

An accurate camera intrinsic estimation is the prerequisite of a correct camera pose estimation. During most of the time of the experiment, we kept using the same camera with the same focal length so that the camera parameters could be utilized repeatedly. Here we present several methods that we used for camera calibration and validate metrics.

3.4.1 Intrinsic Parameter Estimation

During image registration, solving the PnP problemn[10] enables estimation of the camera pose P_c and intrinsic parameters K. However, these initial intrinsic parameters are often imprecise, and the local minimum has not yet been reached. Benefiting from the Bundle Adjustment [36] in COLMAP, as shown in Figure 3.11, parameters can be further refined using additional registered images. In this context, whenever camera recalibration is necessary, a dedicated video with features-rich surroundings is captured for parameter extraction.

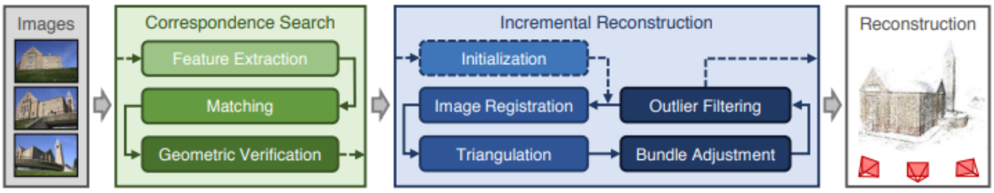


Figure 3.6: Incremental SfM pipeline implemented in COLMAP [34].

3.4.2 Intrinsic Parameter Validations

To assess the accuracy of the intrinsic parameter, we calculated the reprojection with intrinsic parameters from different calibration methods. We proceeded with feature extraction from the validation images utilizing the SIFT algorithm. These extracted features were used to triangulate sparse 3D points, which were then projected onto the image plane using the iteratively refined camera parameters. By comparing the resulting projections with the corresponding 2D feature coordinates, we calculated reprojection errors, including Root Mean Square Error (RMSE) and Mean Square Error (MSE). This approach effectively validates calibration accuracy, particularly in scenarios where ground truth data is unavailable.

3.5 Object Detection and Segmentation

Segment Anything (SAM) is a powerful zero-shot image segmentation model published by Meta. It was trained over 1 billion masks on 11M images [21] with an impressive performance on segmenting any unfamiliar objects. Compared to the traditional segmentation model, no fine-tuning is needed for SAM. Additionally, SAM accepts using bounding boxes or anchor points as prompts for accurate segmentation, which provides more solutions for our task. By using SAM as the segmentation baseline, we designed two strategies to apply accurate single-object segmentation for our task.

3.5.1 Segment Anything and Adjacent Searching

In the first method, we generated masks for the entire image along all the video frames without using any prompt and then started an adjacent searching after annotating one of the frames manually. In figure 3.7, we show the pipeline of this method.
There are three model checkpoints that can be chosen from. Considering the performance cost, we chose the smallest one `vit_b` as the checkpoint. The first 64 objects' masks will be saved into a bitmap by bitwise OR operation on each image. Then, an annotation function will be executed and the annotation window will be rendered by OpenCV GUI library. By collecting the mouse and keyboard events from the callback function, the user can scroll to a desired image with less occlusion and select the target objects. The highlighted outline is done by bitwise XOR between the object mask and the eroded mask, which is represented in Figure 3.8. Once the target object has been selected, the corresponding mask will be saved and passed to the adjacent searching function.

In the adjacent searching function, we use the annotated mask as the starting point M_i and search for the best matching mask from $i - n$ to $i + n$ with the highest Intersection over Union (IoU) and inclusion score. To account for scenarios where a single object might be fragmented into multiple smaller regions, we have developed a multi-mask matching mechanism. This feature facilitates the computation of IoU and inclusion across several bitmap representations. Throughout the process, an index queue and a mask list were maintained. Once the highest IoU and inclusion score have been computed, we add the corresponding frame index to the queue and mask list if it hasn't already been recorded. The detailed algorithm will be explained in Algorithm 1.

The adjacent searching function works based on the assumption that neighboring frames will typically feature masks with similar spatial positions, thereby giving a high IoU value. However, this solution met the problem in the scenario with the dynamic camera. As shown

Algorithm 1 Adjacent Search

```
function AdjacentSearch(queue, masks, mask_bitmaps, imgs, $N_s$, $\theta_{inc}$, $\theta_{iou}$, $\theta_{multi}$)
    curr_round ← 0
    while queue is not empty do
        Sort(queue)
        curr ← queue.pop(0)
        curr_round ← curr_round + 1
        masks[curr]['final'] ← True
        ref_mask ← GetMask(mask_bitmaps[curr])
        img ← ReadImage(imgs[curr])
        result ← CalculateMaskedImage(img, ref_mask)
        for $i \in \{curr - N_s, curr + N_s + 1\}$ do
            if $i$ == curr or $i < 0$ or $i \geq$ mask_bitmaps.shape[0] then
                continue
            end if
            if masks[i] is not None and masks[i]['final'] then
                continue
            end if
            $i_{best}, s_{incl}, s_{iou}$ ← FindBestMatch(mask_bitmaps[i], ref_mask)
            $i_{mbest}, s_{mincl}, s_{miou}$ ← FindMultiMatch(mask_bitmaps[i], ref_mask, $\theta_{multi}$)
            if $s_{mincl} > s_{incl}$ and $s_{miou} > s_{iou}$ then
                $i_{best}, s_{incl}, s_{iou} \leftarrow i_{mbest}, s_{mincl}, s_{miou}$
            end if
            if $i_{best}$ is not None and ($s_{incl} > \theta_{inc}$ or $s_{iou} > \theta_{iou}$) then
                if masks[i] is None then
                    masks[i] ← CreateMaskEntry($i_{best}$, curr_round, $s_{iou}$, $s_{incl}$)
                    queue.append(i)
                else
                    if masks[i]['score'] < $s_{iou}$ then
                        UpdateMaskEntry(masks[i], $i_{best}$, curr_round, $s_{iou}$, $s_{incl}$)
                    end if
                end if
            end if
        end for
    end while
end function
```

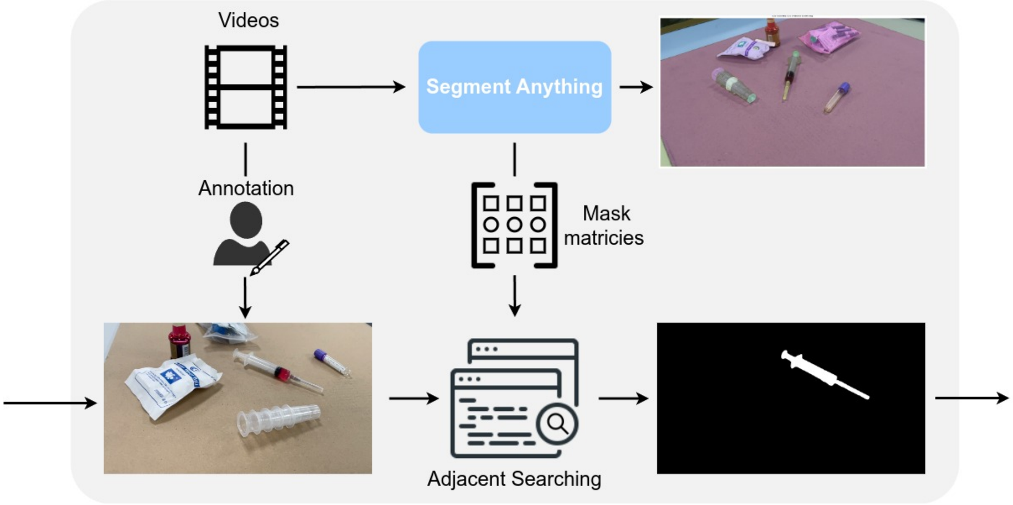


Figure 3.7: Annotation window from the segmentation result

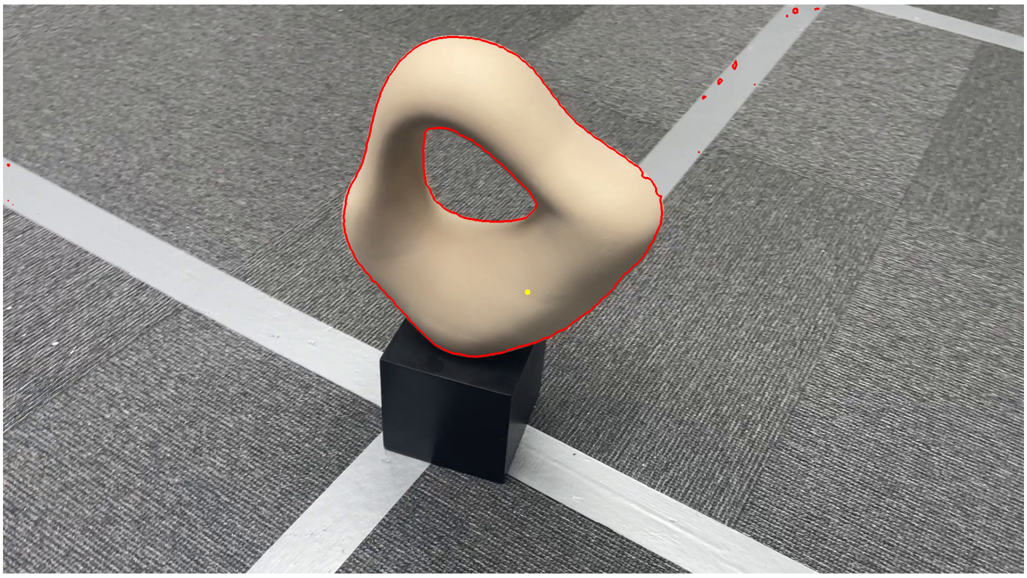


Figure 3.8: Annotation window from the segmentation result

in Figure 3.9, the dramatic camera movement caused by video shooting can cause the object to deviate from the frame's center, which leads to the mismatching between foreground and background.

To address this issue and enhance the robustness and flexibility of the system, we raise another solution: integrating bounding box prompts by YOLO-World [4] into the Segment Anything (SAM).

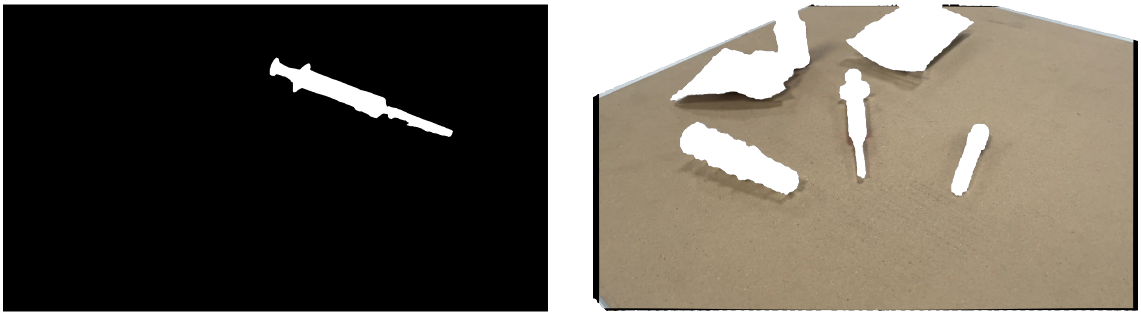


Figure 3.9: Annotation and the wrong matching result

3.5.2 Segment Anything with Annotated Bounding Box Prompt

The YOLO-World model is a YOLOv8-based[31] approach for Open-Vocabulary Detection tasks. It means that the model can even detect objects that have never appeared during the train period. This enables straightforward object detection with simple descriptive texts [5]. With a pre-trained model on larger-scale datasets, this method can detect a wide range of objects which will help us skip the process of manual data labeling.

In our case, "syringe" is set as the text prompt. After inference from YOLO-World model, several bounding boxes will be given. To retrieve the bound box prompt with a clear definition, we added a filter to keep the bounding box with the highest confidence. In most cases, this will accurately annotate the target object, and provide a desirable prompt from SAM.

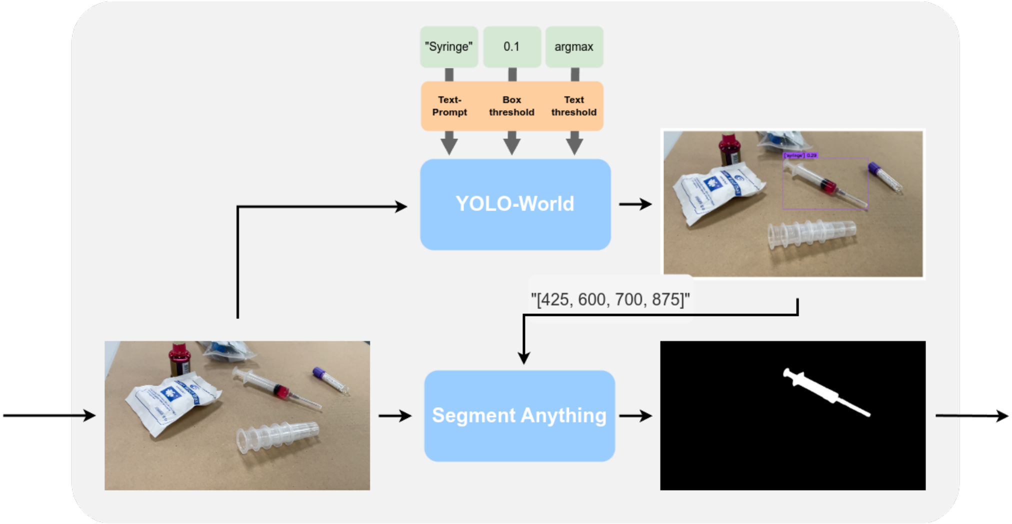


Figure 3.10: Segmentation Result from SAM by YOLO-World generated bounding box

The detail of this segmentation method is given in Figure 3.10. We used the bounding box predicted by YOLO-World as the prompt for the SAM model. In this case, we avoid the time cost of segmenting the unrelated objects. From table 4.2, we can see this strategy results in nearly four times faster than segmenting the whole image and searching the target object

Methods	YOLO-World	SAM	AS	Total Time(s/it)
SAM+AS	···	1.03	0.78	1.81
YOLO-World+SAM	0.02	0.46	···	0.48

Table 3.1: Time cost on each segmentation module

3.6 Incremental SfM Reconstruction

3.6.1 Feature Detection and Extraction

The first step of SfM always starts with feature extraction. Similar to the camera calibration phase, a new image registration database is appointed and the path for the input is assigned as the undistorted images. Given that distortion has been removed, a simple pinhole model, requiring only the focal length and principal point, will be used instead of a complex camera model. The focal length will be the one we got from calibration, and the principal point will be the center coordinate of the sensor. To increase the number of matches detected by SIFT, we also estimate the affine feature shape with option `estimate_affine_shape=true`

3.6.2 Feature matching

As shown in Figure 3.7, the second step of COLMAP's SfM is feature matching and geometric verification. The feature matching matches the image pairs with their appearance description and cannot guarantee a precise correspondence between images to the scene point. Geometric verification verifies this process by estimating a valid transformation mapping between the feature points and scene points[34].

There are several pre-defined matching modes that can be chosen from COLMAP library. During the experiment, we mainly focus on Sequential Matching and Exhaustive Matching. Sequential Matching only matches image pairs along N consecutive frames since the higher possibility of visual overlapping, which is suitable when images are captured in sequential order, for instance, video frames in our case. On the other hand, exhaustive matching, sometimes referred as global matching, involves comparing features across all possible image pairs and not only consecutive ones. This method is more computationally intensive but technically provides more accurate and robust results, especially in a small-scale dataset or dataset where the camera motion is not well-ordered.

3.6.3 Sparse Reconstruction

After producing the scene graph in the previous two steps, the incremental reconstruction process can be started by `colmap mapper`. COLMAP will first seed an initial image pair among all the extracted data from the database. The quality of the initial pair is significant for the whole sparse reconstruction process. Then, the scene is incrementally extended by registering new images and triangulating new points. The result after sparse reconstruction can be visual by COLMAP GUI, as shown in Figure **??**

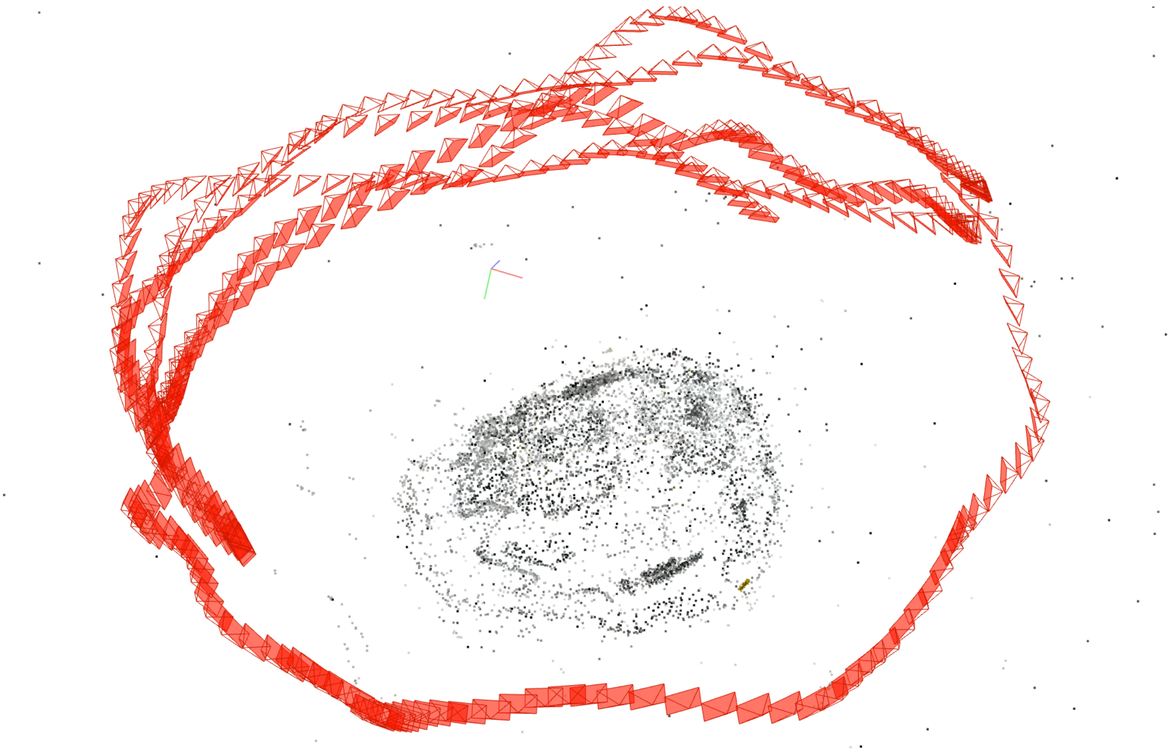


Figure 3.11: SfM Reconstruction Result with a helmet

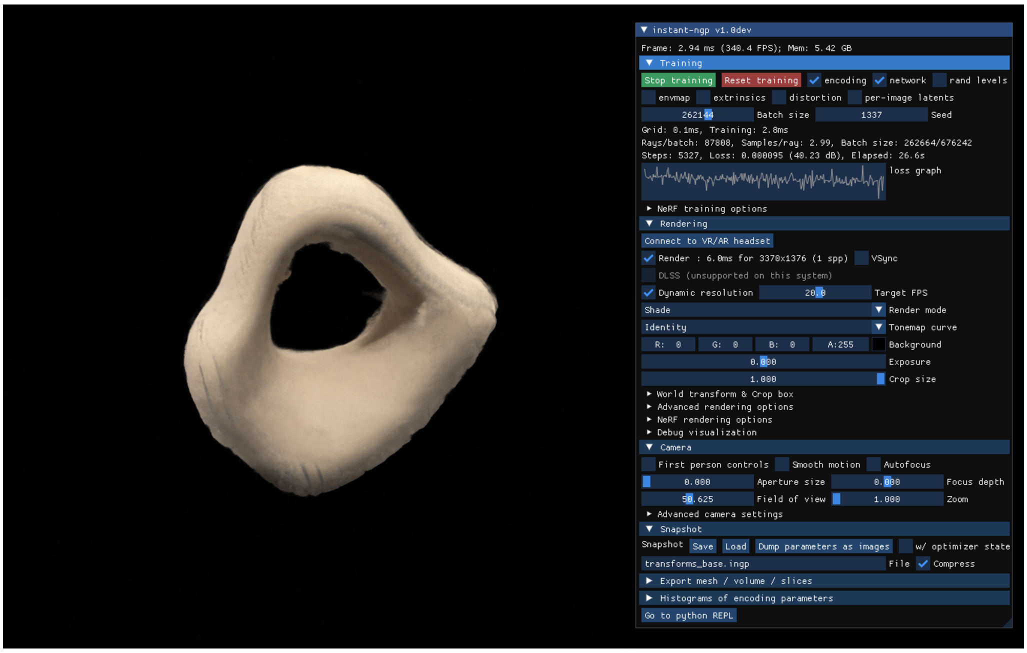
3.7 Objects 3D Reconstruction

In this section, we examine several applications of advanced reconstruction methods — Instant Neural Graphics Primitives (Instant-NGP) [27], 3D Gaussian Splatting [20], Neural Implicit Surfaces (NeuS) [38], Surface-Aligned Gaussian Splatting [14] for Neural Field Rendering and surface reconstruction. variety of objects captured under different conditions using static and dynamic camera setups to evaluate the performance of each technique. Instant-NGP was known for its rapid training and inference capabilities, making it ideal for interactive applications. 3D Gaussian Splatting proved effective in handling sparse datasets with high-qualified novel view synthesis, while NeuS excelled in producing high-quality surface reconstructions with fine geometric details. Through comparative analysis using both subjective and objective metrics, we identified the strengths and limitations of each method in handling complex object geometries and motion dynamics. The findings from these tests lead to practical insights into the selection of appropriate methods for specific applications in 3D reconstruction, providing a foundation for future research.

3.7.1 Novel View Rendering

Instant Neural Graphics Primitives(Instant-NGP)[27] is one of the variations of the NeRF. Compared to NeRF, Instant-NGP still replies on volume rendering for the final output but with a smaller neural network with the help of Multiresolution Hash Encoding. The excellent training pipeline and full implementation on CUDA make this method accelerated by several orders of magnitude.

To start the training with Instant-NGP, an input JSON file needs to be prepared, using the camera parameters obtained from the prior step. The camera angle measures the angle of Field of View (FoV) which is straightforward to calculate along with the image size and focal length. For Instant-NGP, camera poses are required in the form of a transformation matrix. Therefore, the quaternion vectors given by COLMAP need to be converted into a rotation matrix. The training detail is presented in Figure A.2



```
▼ instant-ngp v1.0dev
Frame: 2.94 ms (340.4 FPS); Mem: 5.42 GB
▼ Training
Stop training  Reset training  ☑ encoding  ☑ network  □ rand levels
□ envmap  □ extrinsics  □ distortion  □ per-image latents
262144  Batch size  1337  Seed
Grid: 0.1ms, Training: 2.8ms
Rays/batch: 87808, Samples/ray: 2.99, Batch size: 262664/676242
Steps: 5327, Loss: 0.000095 (40.23 dB), Elapsed: 26.6s
                                          loss graph
▶ NeRF training options
▼ Rendering
Connect to VR/AR headset
☑ Render : 6.0ms for 3370x1376 (1 spp)  □ VSync
□ DLSS (unsupported on this system)
☑ Dynamic resolution  20.0  Target FPS
Shade  ▼ Render mode
Identity  ▼ Tonemap curve
R:  0  G:  0  B:  0  A:255  Background
0.000  Exposure
1.000  Crop size
▶ World transform & Crop box
▶ Advanced rendering options
▶ NeRF rendering options
▶ Debug visualization
▼ Camera
□ First person controls  □ Smooth motion  □ Autofocus
0.000  Aperture size  0.000  Focus depth
50.625  Field of view  1.000  Zoom
▶ Advanced camera settings
▼ Snapshot
Snapshot  Save  Load  Dump parameters as images  □ w/ optimizer state
transforms_base.ingp  File  ☑ Compress
▶ Export mesh / volume / slices
▶ Histograms of encoding parameters
Go to python REPL
```

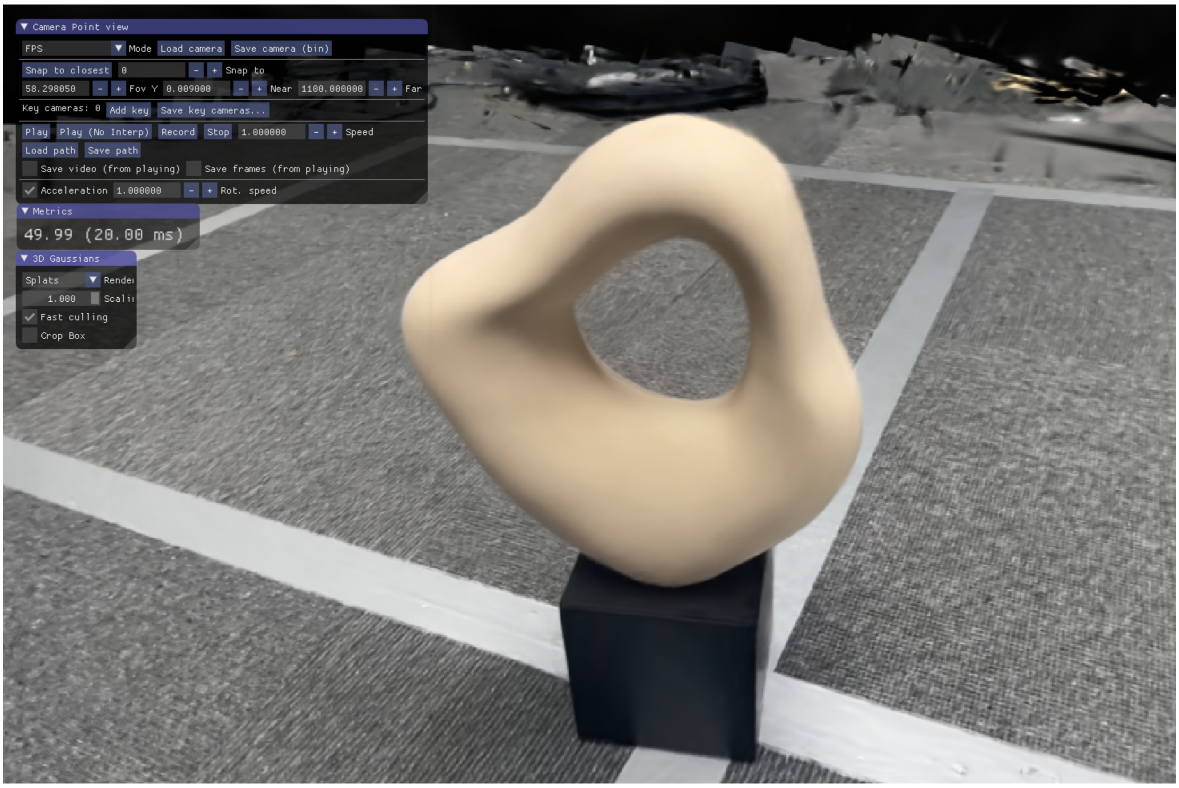
Figure 3.12: Ongoing training in Instant-NGP

3D Gaussian Splatting (3DGS) is a novel real-time radiance field rendering technique. Different from NeRF-based methods which represent the scene with volume density and RGB color generated by Multi-Layer Perceptron (MLP), 3DGS represents the scene with volumetric 3D Gaussian and renders the final images with a differentiable tile rasterizer. This fusion grants 3DGS the advantage from both worlds: competitive training efficiency and state-of-the-art visual quality [20]. As part of this thesis work, we also conducted tests on various objects using 3DGS to facilitate comparison. 3DGS accepts either NeRF synthetic data or COLMAP data as input. We can directly use the previous results without data transformation. The rendering result for a recorded status is shown in Figure 3.13.

3.7.2 Surface Reconstruction

The motivation to use NeuS as well as SuGaR stems from their capacity to represent and reconstruct 3D mesh. This explicit representation is not only more conducive to editing but also enhances performance when using traditional rendering pipelines such as rasterization or ray tracing.

As described in Section 2.3, given the help of Signed Distance Function and multi-resolution



```
▼ Camera Point view
FPS ▼ Mode  Load camera  Save camera (bin)
Snap to closest  0  - + Snap to
58.298050  - + Fov Y  0.009000  - + Near  1100.000000  - + Far
Key cameras: 0  Add key  Save key cameras...
Play  Play (No Interp)  Record  Stop  1.000000  - + Speed
Load path  Save path
□ Save video (from playing)  □ Save frames (from playing)
☑ Acceleration  1.000000  - + Rot. speed
▼ Metrics
49.99 (20.00 ms)
▼ 3D Gaussians
Splats ▼ Rende
1.000  Scali
☑ Fast culling
□ Crop Box
```

Figure 3.13: Training result from 3D Gaussian Splatting

hash tables, NeuS2 equips both efficiency and quality on surface reconstruction even for dynamic scenes. NeuS2 supports the same form of input as Instant-NGP, so there is no need for transform data. The rendered result with 20 000 iterations training is shown in Figure 3.14, more results can be found in Chapter 4.4.

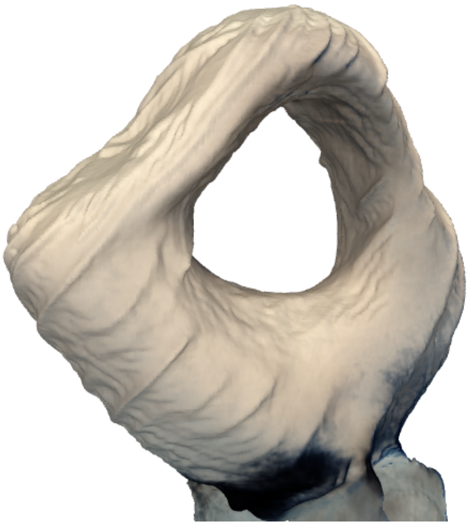


Figure 3.14: Training result from 3D Gaussian Splatting

The result indicates that the surface reconstruction in NeuS2 significantly outperforms texture reconstruction. Although the predicted images appeared reasonable compared to the ground truth, they did not accurately reflect the same on the textures. Additionally, we also noticed that despite the acceptable appearance of the predicted images, the mesh shapes were often severely distorted. Several strategies for enhancing performance were tested including adjusting the number of iterations and modifying the learning rate. However, these did not yield significant improvements.

The previous experiment of rendering novel view from 3DGS has demonstrated its outstanding performance, this makes us wonder how the mesh extracted from 3DGS will look like. Extracting a mesh from millions of 3D Gaussian is a significant challenge. However, Surface-Aligned Gaussian Splatting(SuGaR) gives a solution [14]. In this model, a method that aligns 3D Gaussian to sample points on the real surface of the scene is proposed and Poisson reconstruction is utilized to extract the surface. In Figure 3.15, we demonstrate the result from 3DGS to SuGaR. The holes lying on the objects show that the 3D Gaussian were not completely aligned with the surface. Also some outlines still exist around the object.

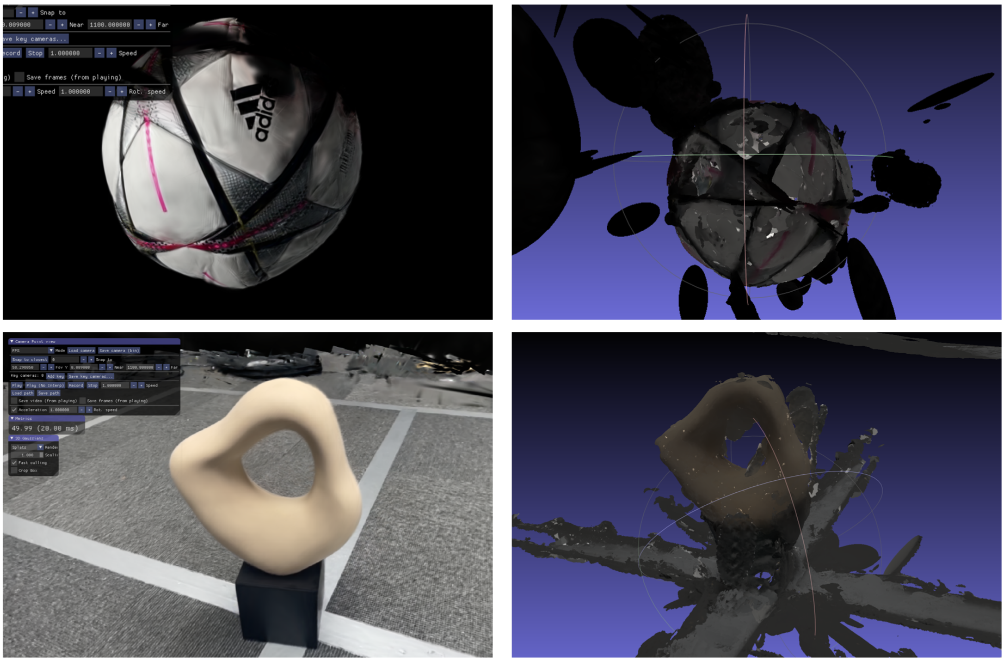


Figure 3.15: Training result from 3D Gaussian Splatting and SuGaR

Chapter 4

Results

In this chapter, we present the results of our 3D reconstruction pipeline, detailing the outcomes and interdependencies of each step, from preprocessing and camera calibration to image segmentation, camera extrinsic estimation, and model training. We evaluate the pipeline's effectiveness through comparative analysis of different experimental methods and captured data, assessed by both visual quality and quantitative metrics. This comprehensive evaluation not only demonstrates the pipeline's overall performance but also identifies areas for future improvement, providing insights into the practical implications of our approach in real-world scenarios.

4.1 Evaluation Metrics

4.1.1 Reprojection Error

The reprojection error is a common measure used in computer vision to quantify the accuracy of a 3D reconstruction or camera calibration. It represents the discrepancy between the projected points, where the points are projected from the 3D object space into the image plane using the camera's parameters, and the actual observed points in the image.

Usually, the formula for reprojection error is given by:

$$e = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \tag{4.1}$$

where (x, y) are the coordinates of the observed point in the image, and $(\hat{x}, \hat{y})$ are the coordinates of the projected point using the estimated camera parameters.

4.1.2 Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR)

To have a quantizable criterion for evaluating the reconstruction result and assess the effectiveness of the different 3D reconstruction algorithms, here we applied two widely used metrics, which are Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). PSNR provides a quantitative measure of the difference between two images based on their pixel-wise intensity differences which makes it particularly sensitive to any small discrepancies between the target and the reconstructed images. A higher value indicates superior performance. SSIM quantifies the discrepancy between the original and reconstructed images[40], considering factors like luminance, contrast, and structure through local patches. Since SSIM was computed by each local patch, it provides a more perceptually relevant measure of image quality. A higher SSIM value signifies better performance.

4.2 Calibration and Distortion Correction Results

Here in Figure 4.2, we present the visual undistortion result got from two different camera calibration strategies. The first setup is calibration by recognizing the patterns on ChArUco board with the help of OpenCV functions. The ChArUco board is shown in Figure 4.1 which is a combination of a chess board and an ArUco board that includes many synthetic square markers composed of a wide black border and an inner binary matrix that determines its identity. The ChArUco board has proven to have strong robustness against occlusions. The other method rely on the sparse reconstruction process in COLMAP, which estimates both the camera transformation matrix and camera intrinsic through solving the PnP problem. Here we tried one situation the ChArUco board was kept in the scene but the other only extracted the features from the surroundings.

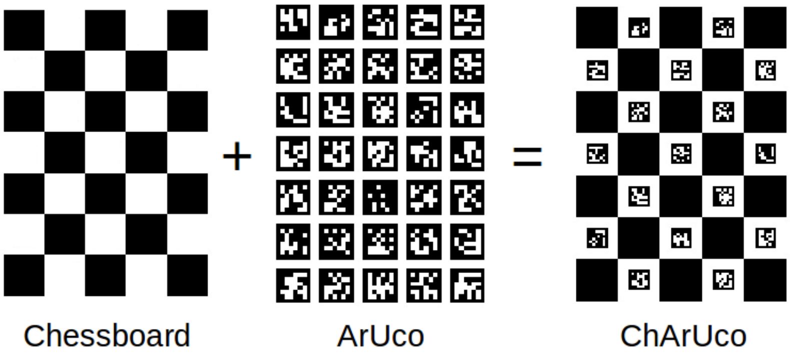


Figure 4.1: Definition of ChArUco board [2]

Since we used the 0.5x zoom lens on an iPhone 12 instead of the fisheye camera, the bending effect is not very clear in our examples also in our undistortion result so we showed the re-projection error in Table 4.1 by applying the undistortion with calibrated parameters on a test scene. As shown in Figure 4.3

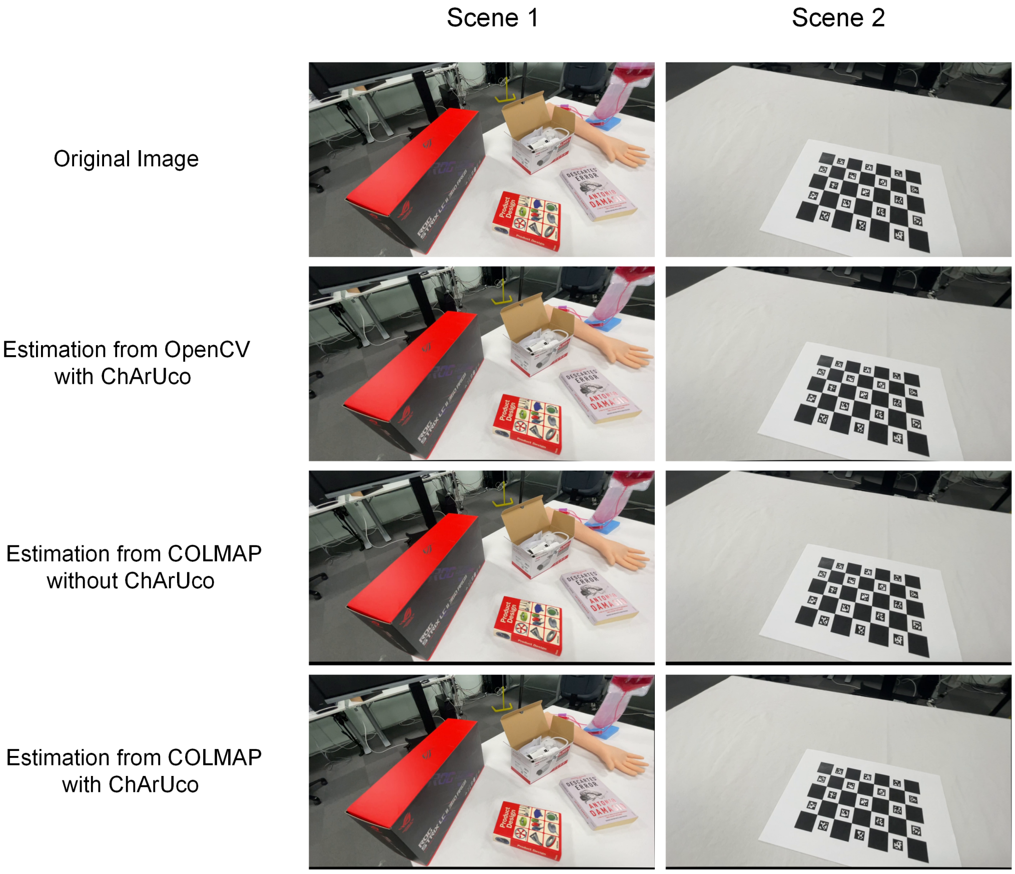

Scene 1
Scene 2
Original Image
Estimation from OpenCV with ChArUco
Estimation from COLMAP without ChArUco
Estimation from COLMAP with ChArUco

Figure 4.2: Undistortion result by using COLMAP

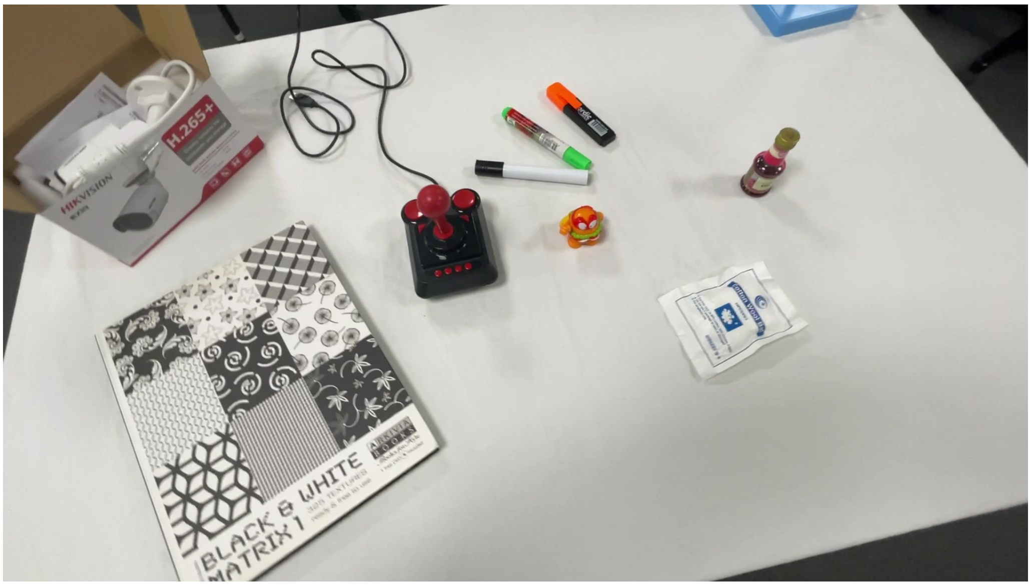


Figure 4.3: Testing scene for computing reprojection error

The results from three different calibrations are really beyond our expectations. It indicates the sparse construction is as good as calibration with marks, or even better. It's also a

Table 4.1: The re-projection error by different calibration solutions on the test scene

Methods	RMS [px]
COLMAP calibration w ChAruCo	0.765476
COLMAP calibration w/o ChAruCo	0.770751
ChAruCo-based calibration (OpenCV)	0.780637

possible reason that, in COLMAP, the camera intrinsics are refined by Bundle Adjustment, where re-projection error has already been used as part of the loss function.

4.3 Camera Extrinics Estimation Results

In this section, we present the results which show the effective extrinsic estimation under a static camera.

Figure 4.4 presents the camera trajectories estimated by COLMAP. The first row shows the objects and the second and third rows show the close shot and long shot view. The result from the keyboard is not satisfactory because the trajectories are fragmented into multiple segments. This issue occurred because the keyboard, when oriented horizontally to the camera, did not present enough distinct features to track effectively. So we here only show the segment that keeps the most of the camera trajectories. In general, it can still prove that estimating camera pose from a fixed view camera is feasible, and reliable.

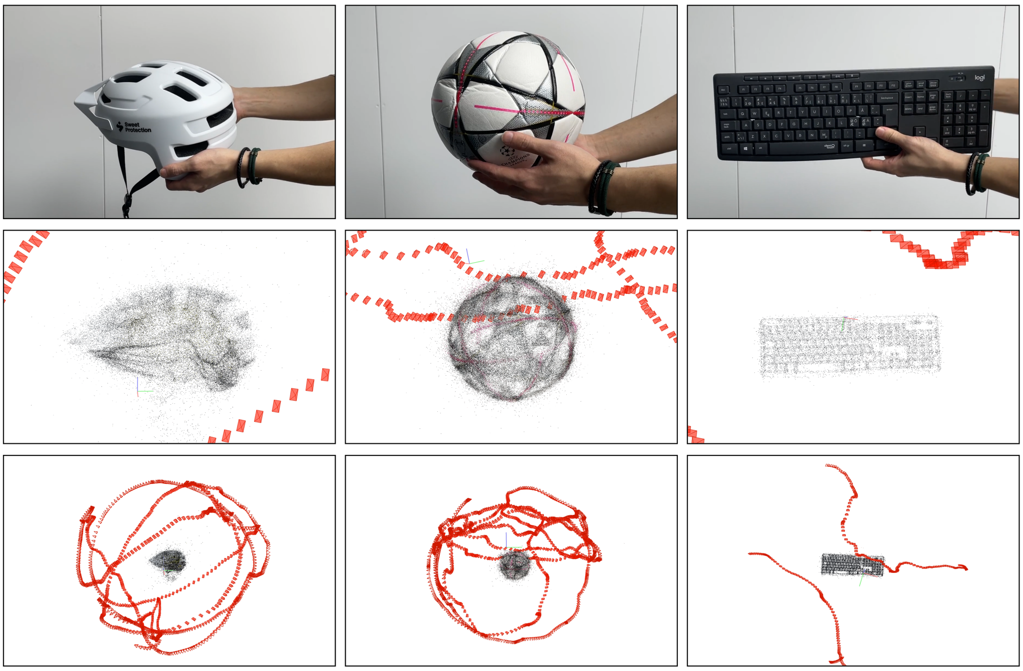


Figure 4.4: Relative camera trajectory for moving rigid objects

Here in Figure 4.6, 4.7, 4.8, we present how different objects perform with two different camera setups. We controlled each object with the same calibration, the same feature matching method, and very close number of frames for training. The SSIM and PSNR are computed between the testing ground truth image and the synthesized image from Instant-NGP.

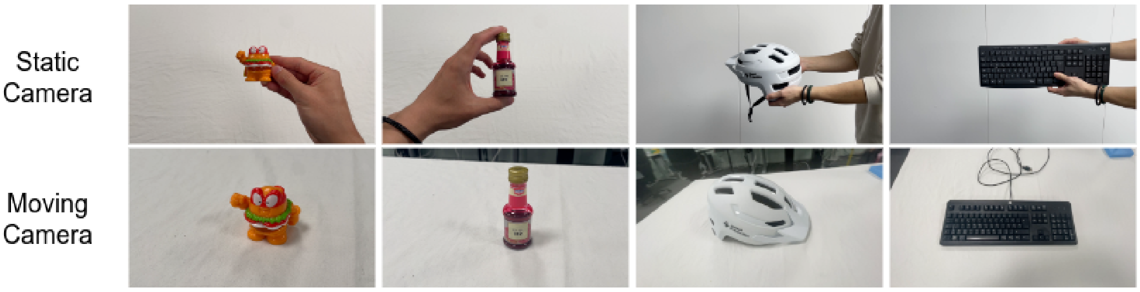


Figure 4.5: The objects that we picked

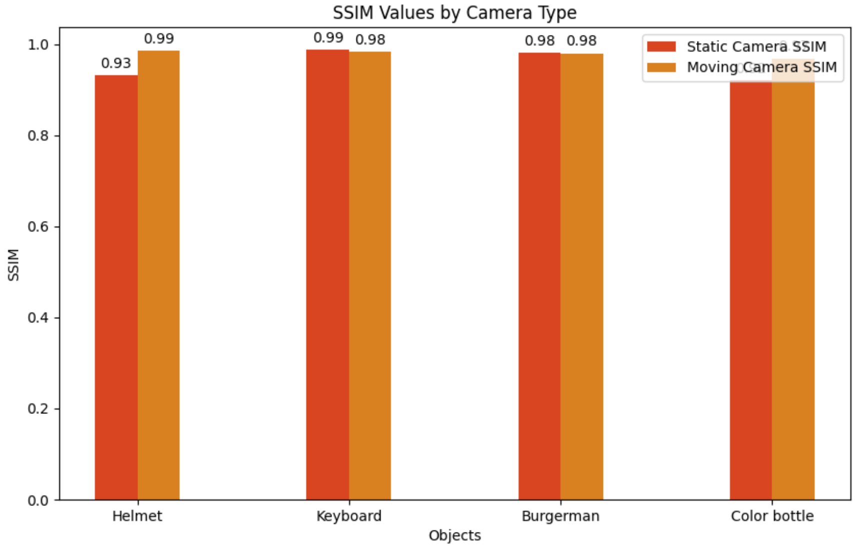


Figure 4.6: Comparison of four different objects on SSIM by camera type

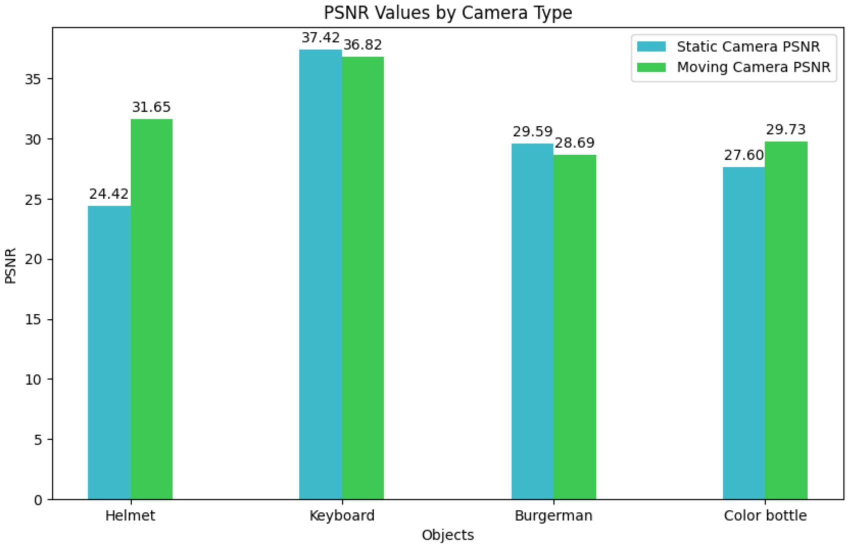


Figure 4.7: Comparison of four different objects on PSNR by camera type

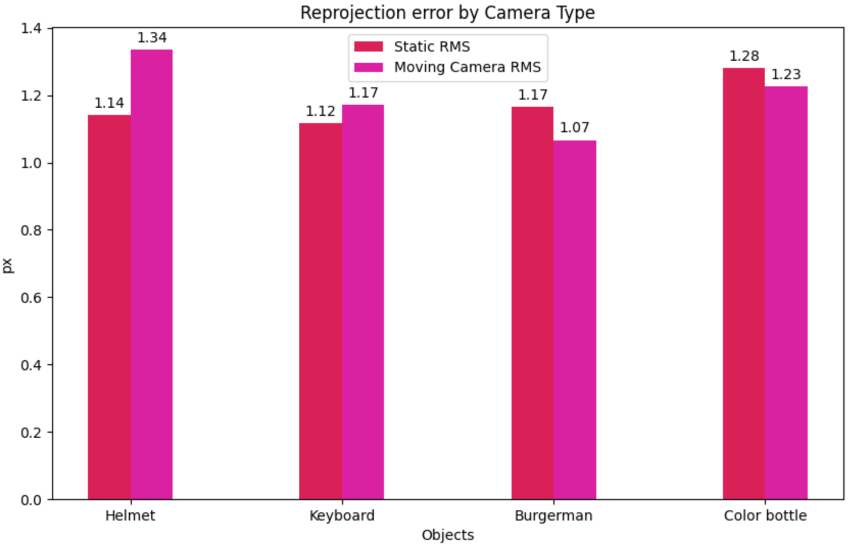


Figure 4.8: Comparison of four different objects on re-projection error by camera type

The results show that with all other conditions the same, the camera pose extracted from the static camera is almost similar to that from the moving camera and is highly competitive.

4.4 Reconstruction results

We have previously discussed the surface reconstruction results by using SuGaR in section 3.7.2. Here in Figure 4.9 we present the mesh generated by Instant-NGP. The second row illustrates the normal map of the surface. It is evident from the mesh that, although the results from this method are reasonably good, noticeable outlines remain around the objects.

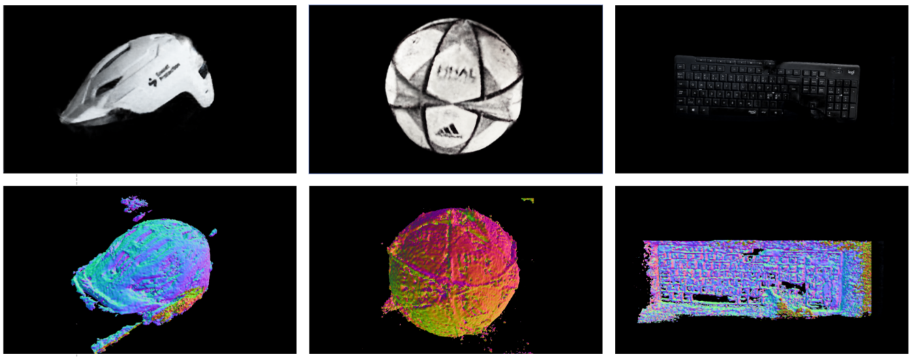


Figure 4.9: Meshed result from Instant-NGP

In Figure 4.10, the ground truth, predicted image and the mesh result are shown. In general, NeuS2 has better surface reconstruction performance. However, the texture reconstruction is not very ideal sometimes. We also found that for transparent objects like syringes, both surface reconstruction and texture reconstruction became more difficult.

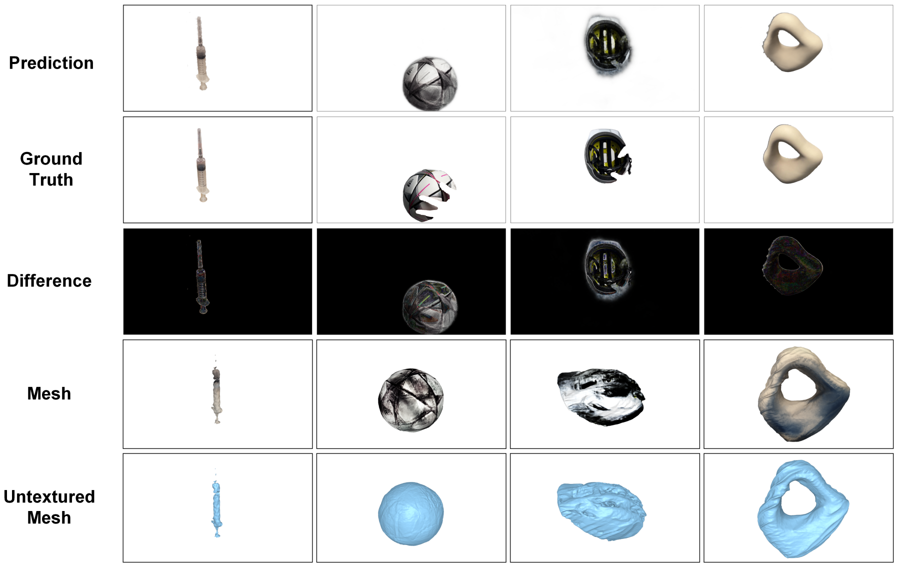


Figure 4.10: Reconstruction and rendering result on four objects. The football and helmet are captured by a static camera. The syringe and the statues are captured by the moving camera

In table 4.2, we show the evaluation metrics we got from reconstruction the video of the statues with four different methods that we tested. The higher SSIM and PSNR indicate better image quality which means better novel-view-synthesis performance. Here we can see Instant-NGP, 3DGS, and NeuS2 have similar results. SuGaR have relatively worse result compare three other methods.

Table 4.2: Evaluation metrics with different methods on the recorded statues, as shown in the last volume of Figure 4.10

Methods	**SSIM↑**	**PSNR↑**
INGP-Base	0.989	33.91
3DGS	0.982	28.61
NeuS2	0.983	31.52
SuGaR	0.975	27.90

Chapter 5

Discussion

In this chapter, we explore the difficulties we faced during our research and discuss possible explanations for these issues. Despite these challenges, the chapter also explores promising potential uses of 3D scene reconstruction, including generating digital assets for entertainment and XR experiences, creating datasets for training machine learning models, and enhancing XR-assisted training with immersive simulations. Finally, we suggest potential directions for future research in this field.

5.1 Challenges on 3D Reconstruction

5.1.1 Trade-off of Static and Dynamic Camera Setups

During the experiment, we found that static cameras heavily rely on extracting high-quality features from the objects to estimate relative camera poses accurately. Objects lacking distinct features often lead to catastrophic failure during the reconstruction. For such objects, capturing images around them, and using background features for assistance, allowed to reach significantly better results. However, static cameras still offer significant advantages, such as consistent lighting conditions and ease of calibration, which contribute to their robustness in controlled environments. More importantly, this setup can be easily extended to multi-camera systems to resolve the depth ambiguity and occlusion in various tracking tasks [42].

On the other hand, dynamic cameras, with their ability to capture objects from multiple angles, provide comprehensive coverage and detailed reconstructions. It has been indicated by our experiment that for certain objects, such as syringes with transparent surfaces, capturing video around is still the only solution. However, we observed that the motion blur becomes more pronounced with a dynamic camera, which can potentially affect the accuracy of the reconstructions.

5.1.2 Impact on Inaccurate Camera Pose and Complex Illumination

Another finding we got from the experiment is that the inaccurate camera extrinsic parameters and lighting conditions are still big challenges for most of the current 3D reconstruction algorithms.

For instance, the Synthetic Lego and DTU dataset[19] are the two most well-known datasets for evaluating surface reconstruction. In NeuS2, they have also been adopted for visual comparisons. However, the Synthetic Logo is generated from Blender with error-free camera poses. DTU dataset, as shown in Figure 5.1, is engaged with a robot arm and controlled light source, which is qualified for a training dataset but not representative enough for the custom data, which leads to difficulties in reproducing reconstruction effects close to those in the papers.

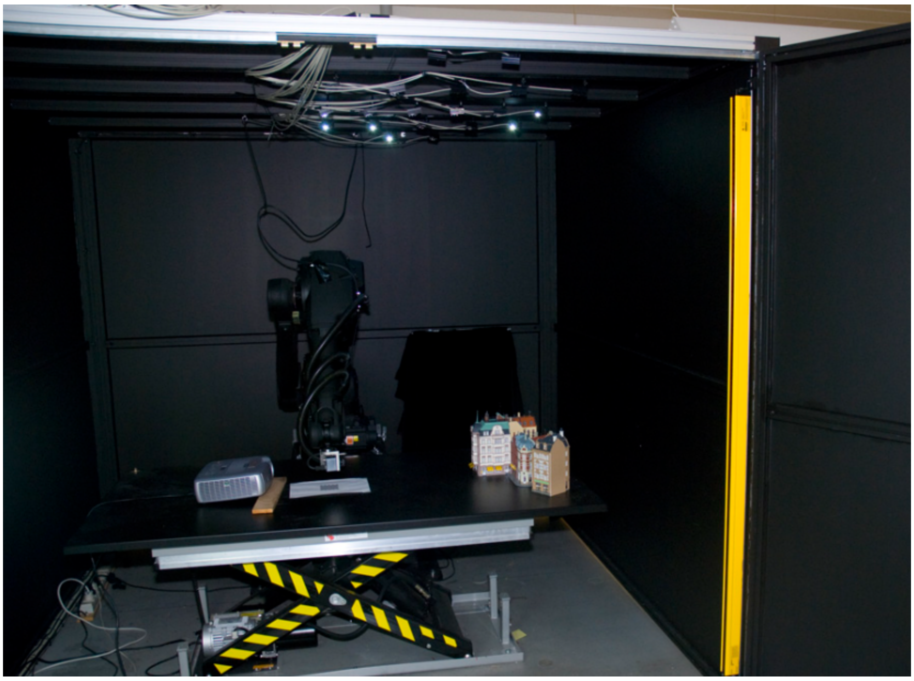


Figure 5.1: The data capture setup for DTU dataset[19]

5.1.3 Limitations on Sequential Matching and Exhausting Matching

As described in Section 3.6.2, we tested both sequential feature matching and exhaustive feature matching. However, we notice some artifacts from the results with dynamic asymmetric objects. As shown in Figure 5.2, some ghosting point clouds are located on the edges of the helmet which are indicated with the red line. We assume this is caused by offset on triangulation with repeat frames, which means the repeated frames will be triangulated individually generated 3D points with accumulated error.

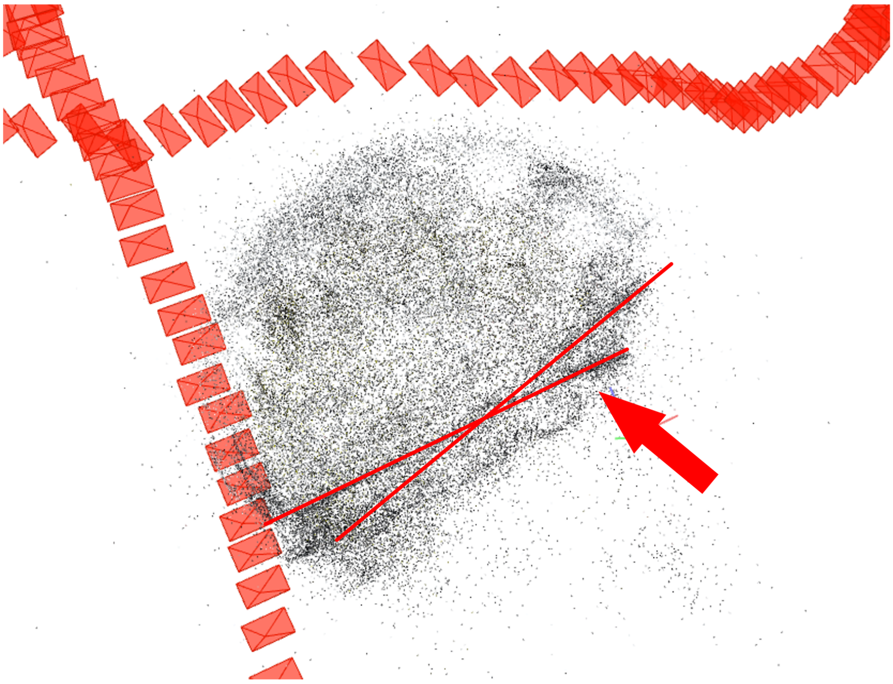


Figure 5.2: Artifact on the points cloud caused by inaccurate camera pose

Controlling the object's motion to ensure that every surface is uniquely captured by the camera is challenging. After attempting exhaustive matching on the same object, the ghosting effect on the edge of the helmet became much better but some new artifacts as the two point clouds of the helmet overlapped in opposite directions. We suspect this issue stems from the incorrect feature correspondences between images, more specifically by the mismatching between frames with symmetry. With the use of sequential matching, even if the repeated frames across a long span will not be matched together, they still eventually result in ghosting because of the accumulated estimation error. Although exhaustive matching addresses the issue of repeated frames, it will still face the challenge of mismatching image pairs.

5.1.4 Shortcomings of 3D Gaussian Splatting (3DGS) in Surface Reconstruction

While the anticipated outcome of the SuGaR was high-quality 3D mesh for the captured objects, the results obtained were significantly lower than expected. This divergence comes with the following possible reasons.

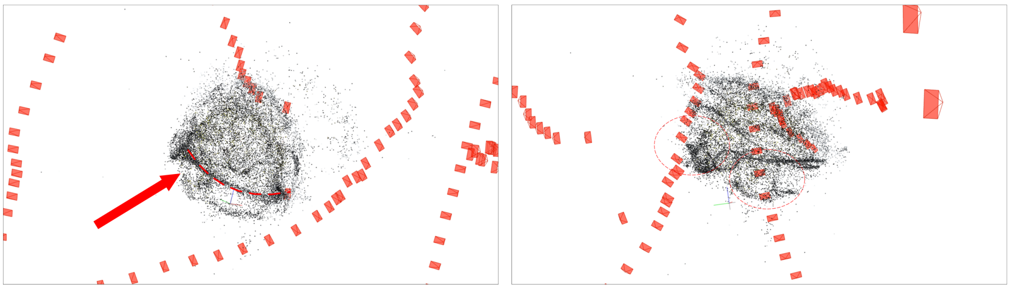


Figure 5.3: Artifact on the points cloud caused by inaccurate camera pose

The inherent limitation from 3DGS

Despite that we have masked the input image with Alpha channel masking, there will still be outlines that are generated in the masked area during SfM. Naturally, the 3D Gaussians are initialized from the existing points cloud even if the masking exists, which will finally produce artifacts on the rendering result.

The lack of masking supportion

The core contribution of SuGaR is to align the 3D Gaussian with the surface of the scene. Essentially, SuGaR serves as a post-processing step that refines the results from 3DGS. We found that due to the fact SuGaR still does not support masking at the moment, the artifacts from 3DGS will continue to affect the surface reconstruction result.

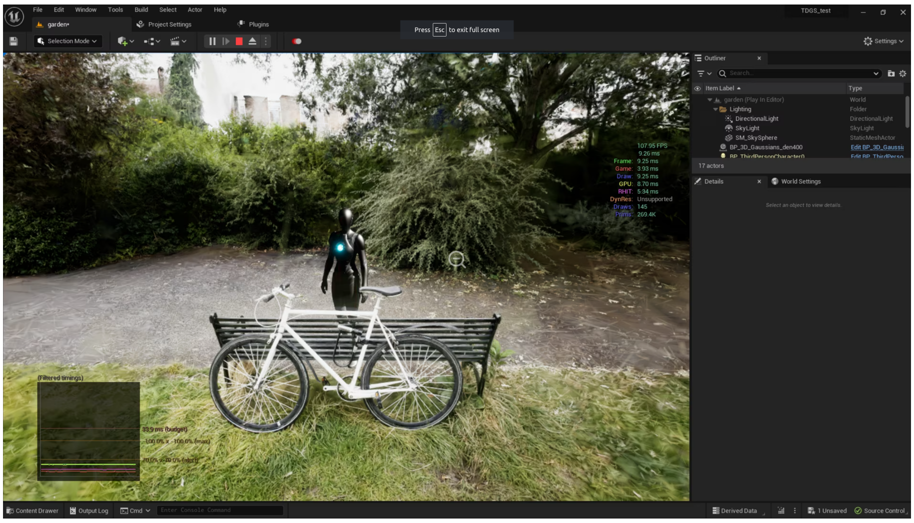
5.2 Potential Usages

5.2.1 Digital Assets Generation

The booming market growth in the entertainment industry has raised a strong need for high-quality 3D models for content creation. The ability to generate realistic 3D scenes from the real world helps developers put more focus on the game logic design without spending too much time on extensive 3D models. In another aspect, the detailed reconstruction of real-world environments facilitates more realistic interactions between the virtual and real elements. Especially in AR and MR, where blending the virtual and real seamlessly is key. The user would use their MR device with multi-cameras to scan the surrounding objects into digital assets and simulate the interaction in real-time. Seeing Figure 5.4 where Unreal 5 has published a powerful plugin [18] that allows users to import 3DGS rendering results into the editor

5.2.2 Dataset Generation

Both 3DGS and instant-ngp ultimately serve as methods for synthesizing the novel view from a given viewing direction. We can naturally expect that this feature can be utilized to generate datasets with ground truth camera pose. Related research has already been conducted.



```
107.95 FPS
  9.26 ms
  Frame: 9.25 ms
   Game: 3.93 ms
   Draw: 9.25 ms
    GPU: 8.70 ms
   RHIT: 5.34 ms
 DynRes: Unsupported
  Draws: 145
  Prims: 269.4K
```

```
(Filtered timings)
33.9 ms (budget)
100.0% x 100.0% (max)
70.0% x 70.0% (alert)
```

Outliner

Item Label	Type
garden (Play In Editor)	World
Lighting	Folder
DirectionalLight	DirectionalLight
SkyLight	SkyLight
SM_SkySphere	StaticMeshActor
BP_3D_Gaussians_den400	Edit BP_3D_Gaussia
BP_ThirdPersonCharacter0	Edit BP_ThirdPerso

17 actors

Details | World Settings

Select an object to view details.

Figure 5.4: Editable 3DGS rendering result in Unreal 5 [18]

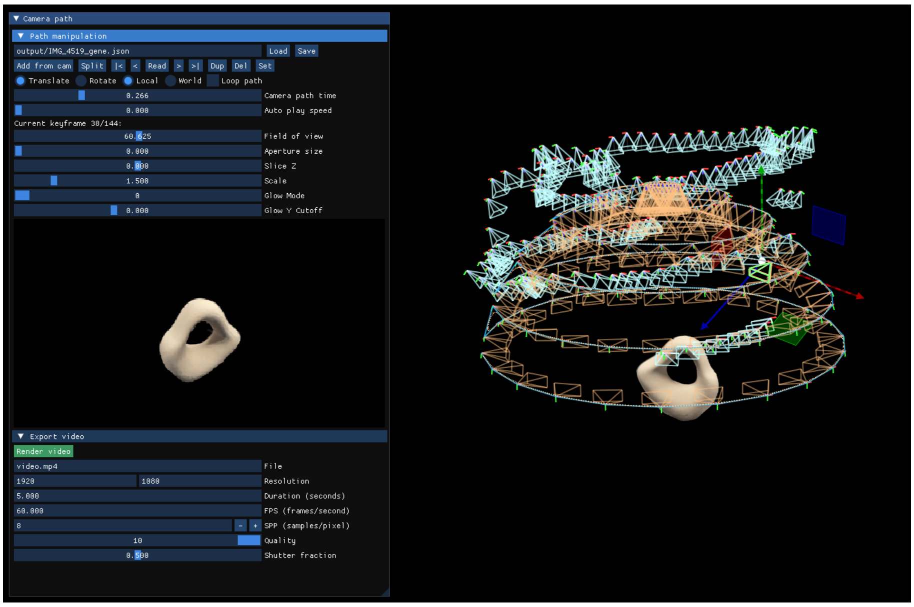
For example, Neural-Sim proposed a differentiable synthetic data pipeline that uses NeRF in a closed-loop and also proved the effectiveness on object detection tasks [11]. Zhong et al.[44] also proposed a tactile sensory data generation strategy with significant improvement in tactile classification task.

As presented in Figure 5.5, we render a result with the generated cameras that are arranged along the hemispherical trajectory over the object. Each camera is also aimed at the center of the object and is able to render an image out of the object. The blue triangles are the camera trajectory from the captured video. This setup allows us to obtain images of the object from any distance and field of view without needing to capture every aspect physically. Such an approach is greatly beneficial for tracking tasks such as 6 DoF pose estimation, where acquiring the ground truth data is usually challenging.

5.2.3 XR Assisted Training

As the XR market has been undergoing rapid growth in recent years, there is a critical need to enable immersive and interactive XR training experiences that simulate real-world scenarios. Traditional professional training methods cannot often provide realistic, hands-on experiences that are crucial for skill acquisition in fields such as industrial maintenance, healthcare, and engineering. By harnessing advancements in 3D scene understanding and tracking technologies, XR training systems can bridge this gap by offering dynamic, personalized learning environments.

Our work has shown the following benefits on XR assisted training scenario:



output/IMG_4519_gene.json
0.266 Camera path time
0.000 Auto play speed
Current keyframe 38/144:
60.625 Field of view
0.000 Aperture size
0.000 Slice Z
1.500 Scale
0 Glow Mode
0.000 Glow Y Cutoff
video.mp4 File
1920 1080 Resolution
5.000 Duration (seconds)
60.000 FPS (frames/second)
8 SPP (samples/pixel)
10 Quality
0.500 Shutter fraction

Figure 5.5: Generated cameras with recording cameras

Seemless object highlighted without animated explicit 3D mesh model

Most industrial XR applications highlighted the interacted object with overlayed pre-build 3D model. However, the latency and occlusions always cause offset which significantly damages the immersive experience. Benefits from the on-the-fly training on 3DGS [35], the rendering result can be directly combined with the received video stream and cast to Head Mounted Display (HMD) without extra graphics rendering.

Accurate tracking with muti-camera collaboration

Typically, the camera system on HMD enables plane detection through the head motion. However, as we mentioned before, this brings extra problems in object tracking. By integrating with a multi-camera system set at a fixed angle, occluded information can be captured, significantly enhancing the system's robustness.
In this thesis project, we conducted general research on 3D scene reconstruction from the respective of both key components in the pipeline and potential usages. We answered key questions by presenting several conclusions we got during the experiment. We have shown that reconstructing objects from a fixed viewpoint is possible even if some challenges need to be solved. We also point out several future works that can significantly benefit the industry by applying this technology.

5.3 Future works

5.3.1 High-fidelity neural surface reconstruction with the help of the points cloud

Nowadays, both NeRF-based surface reconstruction methods like Neuralangelo [23] and SDF-based surface reconstruction methods such as NeuS [38] accept the camera poses as model input. As the by-products during camera pose estimation, the points cloud gives very rich spatial information but not many methods think about utilizing them in a good way. 3DGS is an excellent practice but still faces the challenge we mentioned above. How to use those point clouds as a prompt for a high-fidelity neural surface reconstruction can be good work in the future.

5.3.2 Inverting NeRF for 6-DoF pose estimation

NeRF is a model that takes view direction as input and synthesizes novel view images. But what about doing it oppositely? Previous studies [41] have explored training a Neural Object Field online along with pose graph optimization to robustly accumulate information into a consistent 3D representation. It can handle challenging sequences with large pose changes, occlusions, untextured surfaces, and specular highlights. Following this, PixTrack addresses the limitations of previous methods by using a NeRF as the canonical representation of the object [6], enabling graceful handling of in-plane rotations, producing photo-realistic reference frames, and accurately filtering 3D points for feature-metric alignment.

Inspired by these related studies, one of our future works would be adding a convolution network for estimating pseudo pose as the prompt for a pre-trained NeRF model. By minimizing the residual between the synthesized image from NeRF and the captured frame, this model can be trained with both accuracy and efficiency in 6-DoF pose estimation.

Chapter 6

Conclusion

6.1 Key Findings

The first key finding we have is the adaptation of two different camera setups in different scenarios. We proved the possibility of adding masking to estimate the relative pose on static camera. But at the same time, this is limited by the texture and shape of the objects during to the amount of extracted features. Even so, static cameras still offer significant advantages, such as consistent lighting conditions, ease of calibration, and ease of extend to muti-camera systems, which contribute to the robustness of the system.

Additionally, we found that either the sequential features matching or exhausting features have their limitation. Compared to exhausting matching, sequential matching is more efficient by exploiting the temporal correlation between the neighboring frames of videos, but the offset on triangulation and lack of spatial correlation on the existing points will lead to ghosting points cloud occasionally. Similarly, exhausting matching matches the frame with the closest feature distance but ignores the temporal correlation, which can also cause mismatching on similar semantic features but opposite camera poses. Some other flexible matching strategies have been provided in COLMAP such as Spatial Matching and Custom Matching. This can be a future work for more improvement.

Furthermore, we find that the inaccurate camera pose and complex illumination are still the biggest challenges on custom data in 3D reconstruction. Most of the testing results on the Sota 3D reconstruction method stem from either a synthetic dataset or a dataset with a controlled environment. This makes it difficult to reproduce similar results in the paper for practical applications, but this also inspires us with a new question of how to compensate for the discrepancy between actual and ideal data.

In the end, although 3DGS is a very innovative method that perfectly balances the quality and efficiency of novel view synthesis, the method is not as effective in surface reconstruction as it should be. Triangles are still the most commonly used primitive for graphics rendering due to their mathematical simplicity and stability. Therefore, it's difficult for 3DGS which uses 3D Gaussian as primitive to avoid the holes on the reconstructed surface. Some related studies [14, 25] have been conducted in this aspect but there is still a long way to go.

6.2 Answer the research questions

Research Question 1: How can arbitrary rigid objects be reconstructed from static cameras, and what are the primary challenges involved in this process?

We discovered that the crucial aspect of this question is to estimate the relative camera pose within the local coordinates of objects. From this point, we found that reversing the order of extrinsic estimation and segmentation within our pipeline is the simplest solution. By isolating the background through masking, triangulation is only conducted using paired features derived directly from the objects. This approach allows for precise estimation of the relative extrinsic parameters. Subsequent experiments validated the efficacy of this method. However, this solution is still facing limitation, particularly because it highly relies on the object's appearance. For instance, objects with a single color or made from glass often contain very few features, which can hinder accurate extrinsic estimation.

Research Question 2: What are the necessary components in the pipeline from captured images to the final reconstructed object?

In this study, we developed a comprehensive pipeline comprising several key stages: preprocessing, camera calibration, image distortion correction, segmentation, extrinsic estimation, and reconstruction model training. During the preprocessing phase, we effectively eliminate issues such as motion blur. We evaluated two different camera calibration methods, discovering that Structure-from-Motion (SfM)-based calibration and OpenCV-based calibration exhibit comparable performance, but the former one has more flexibility. Depending on the dynamics of the scene and the movement of the camera, our approach to camera estimation alternates between prioritizing segmentation or using COLMAP initially.

Research Question 3: Which reconstruction method is considered the most effective among existing algorithms?

During our project, we compared two novel view synthesis methods 3D Gaussian Splatting (3DGS) and Instant Neural Graphics Primitives (Instant-NGP). While 3DGS provided superior rendering resolution compared to Instant-NGP, it required a longer training period. Additionally, we evaluated two surface reconstruction methods derived from these algorithms. Based on the visual quality of the reconstructed 3D meshes, we conclude that NeuS2 has better surface reconstruction capabilities but bad at texture reconstruction. SuGaR presents a viable approach to extract mesh from 3D Gaussian but struggles with handling masks effectively.

Research Question 4: What are the potential uses of this technique?

We propose that 3D scene reconstruction can be broadly used in generating digital assets for the entertainment industry. Since both 3DGS and instant-NGP serve as methods for synthesizing the novel view from a given viewing direction, we expect that this feature can be utilized to generate datasets for deep learning model training. Finally, we present a potential usage for object highlighting and accurate tracking under the XR training scenario.

References

[1] *Absorption and Scattering by an Arbitrary Particle*, chapter 3, pages 57–81. John Wiley & Sons, Ltd, 1998.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. *Unstructured Lumigraph Rendering*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023.

[4] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.

[5] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:*, 2024.

[6] Prajwal Chidananda, Saurabh Nair, Douglas Lee, and Adrian Kaehler. Pixtrack: Precise 6dof object pose tracking using nerf templates and feature-metric alignment, 2024.

[7] Brandon Clark, Alec Kerrigan, Parth Parag Kulkarni, Vicente Vivanco Cepeda, and Mubarak Shah. Where we are and what we're looking at: Query based worldwide image geo-localization using hierarchies and scenes, 2023.

[8] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. *Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023.

[9] Jan-Niklas Dihlmann, Andreas Engelhardt, and Hendrik Lensch. Signerf: Scene integrated generation for neural radiance fields, 2024.

[10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.

[11] Yunhao Ge, Harkirat Behl, Jiashu Xu, Suriya Gunasekar, Neel Joshi, Yale Song, Xin Wang, Laurent Itti, and Vibhav Vineet. Neural-sim: Learning to generate training data with nerf. *arXiv preprint arXiv:2207.11368*, 2022.

[12] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

[13] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., USA, 2006.

[14] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv preprint arXiv:2311.12775*, 2023.

[15] Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. In-hand 3d object scanning from an rgb sequence, 2023.

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[17] The MathWorks Inc. statistics and machine learning toolbox documentation, 2022.

[18] Akiya Research Institute. 3d gaussians plugin, 2023.

[19] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), jul 2023.

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[22] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2023.

[23] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[24] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[25] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting, 2024.

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, dec 2021.

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *CoRR*, abs/2201.05989, 2022.

[28] United Nations. The un sustainable development goals, 2015.

[29] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction, 2021.

[30] Dou Quan, Shuang Wang, Yu Gu, Ruiqi Lei, Bowu Yang, Shaowei Wei, Biao Hou, and Licheng Jiao. Deep feature correlation learning for multi-modal remote sensing image registration. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022.

[31] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8, 2023.

[32] Mathieu Sanchez, Oleg Fryazinov, P.A. Fayolle, and Alexander Pasko. Convolution filtering of continuous signed distance fields for polygonal meshes. *Computer Graphics Forum*, 34:277–288, 09 2015.

[33] A. Schwarzenberg-Czerny. On matrix factorization and efficient least squares solution. , 110:405, April 1995.

[34] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[35] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. *arXiv preprint arXiv:2403.01444*, 2024.

[36] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK, 2000. Springer-Verlag.

[37] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z. Jane Wang, and Rabab Ward. Multi-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5722–5731, October 2021.

[38] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction, 2023.

[39] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction, 2023.

[40] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[41] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Muller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. *CVPR*, 2023.

[42] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance, 2020.

[43] Oleg Yavoruk. The study of observation in physics classes through xr technologies. In *Proceedings of the 4th International Conference on Digital Technology in Education*, ICDTE '20, page 58–62, New York, NY, USA, 2020. Association for Computing Machinery.

[44] Shaohong Zhong, Alessandro Albini, Oiwi Parker Jones, Perla Maiolino, and Ingmar Posner. Touching a nerf: Leveraging neural radiance fields for tactile sensory data generation. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1618–1628. PMLR, 14–18 Dec 2023.

Appendices

Appendix A

Reconstruction Results by Each Step

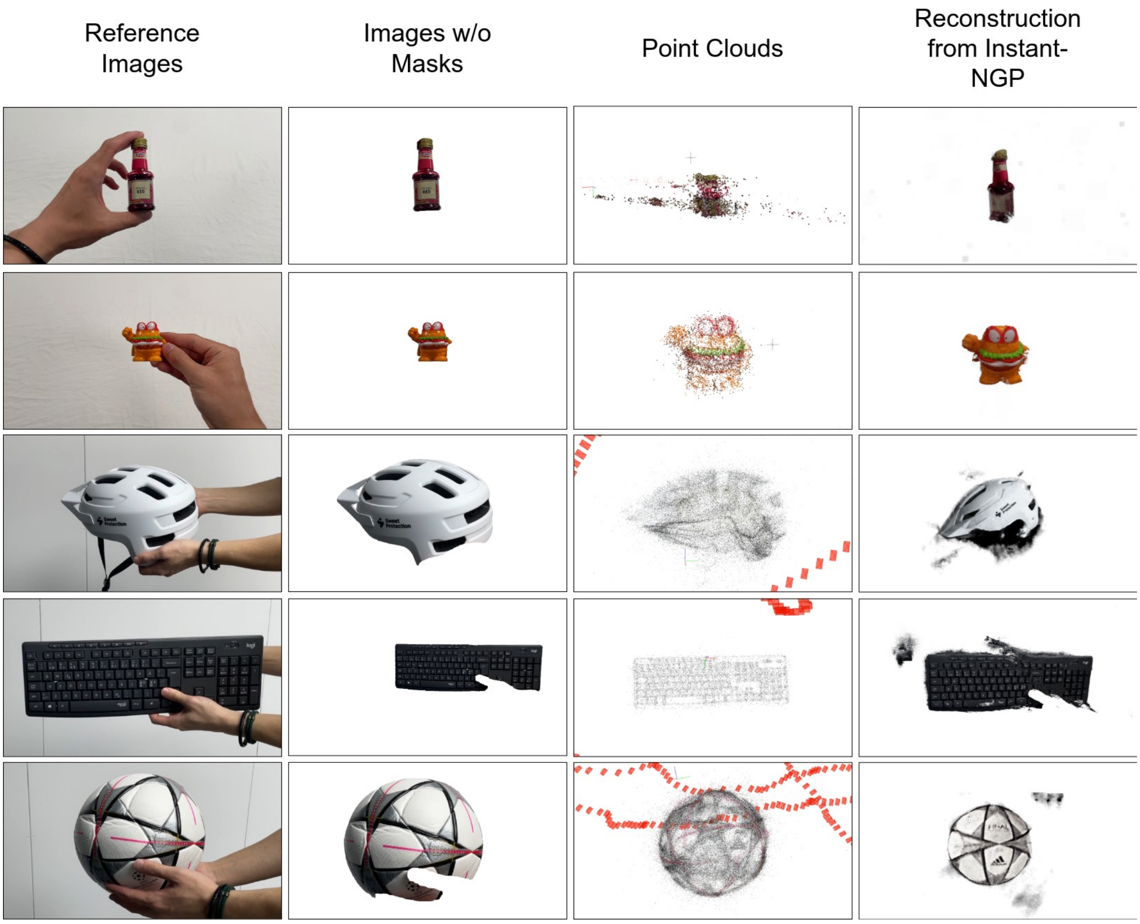


Figure A.1: Results from each steps of the pipeline with static camera

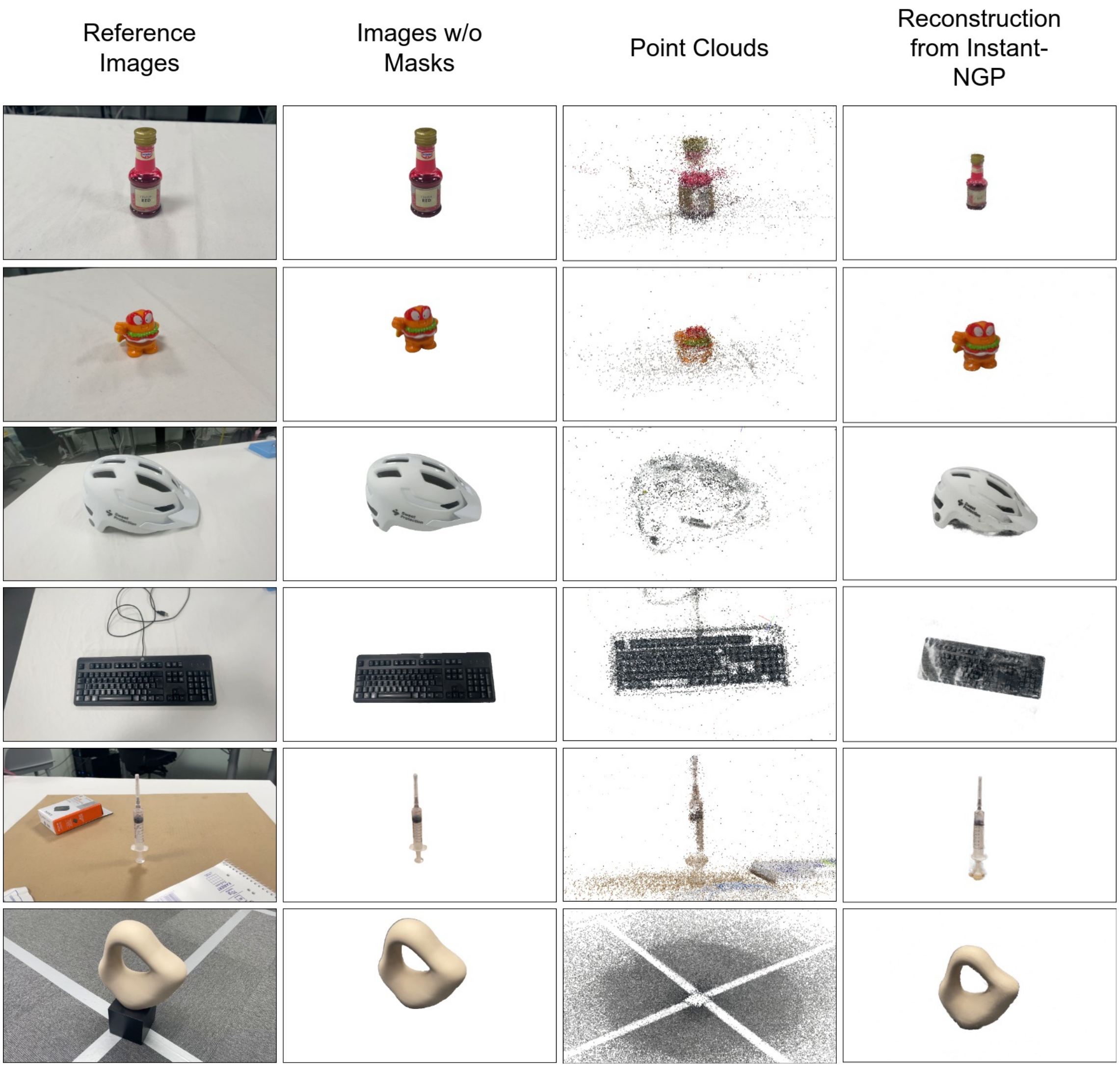


Figure A.2: Results from each steps of the pipeline with freely moving camera