



SCHOOL OF AEROSPACE AND GEODESY  
— GEODESY AND GEOINFORMATION

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Master of Science

**3D reconstruction of street view scenarios  
using a single monocular camera**

Yue Tan



# 3D reconstruction of street view scenarios using a single monocular camera

Author: Yue Tan  
Supervisor: Olaf Wysocki, M. Sc. ,  
Dr.-Ing Yan Xia,  
Nick Wandelburg, M. Sc. ,  
Oussema Dhaouadi, M. Sc. ,  
Prof. Dr.-Ing. Christoph Holst,  
Prof. Dr. Daniel Cremers

Submission Date: 10.06.2024

Collaboration: DeepScenario 

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 10.06.2024

Yue Tan

Yue TAN

## Acknowledgments

I would like to express my heartfelt gratitude to my supervisors, Olaf Wysocki, Dr.-Ing Yan Xia, Nick Wandelburg, and Oussema Dhaouadi, for their invaluable support and guidance throughout my research. Our weekly meetings were immensely helpful, providing timely and effective advice that significantly improved my thesis. I am grateful for the suggestions on experimental improvements and the detailed feedback on my writing. Specifically, their encouragement has helped me to overcome challenges and stay focused on my goals. I would also like to extend my sincere gratitude to Prof. Dr.-Ing. Christoph and Prof. Dr. Daniel Cremers for their provision of essential resources and support throughout my thesis.

I would also like to thank my friends Zhuoying, Zhangdesai, Echo and Xiaoman. Their constant presence, understanding, and motivational words have been incredibly important to me during this journey. I am grateful for your patient companionship and emotional support.

Finally, my deepest gratitude is towards my mother, Xiaoying, who's always been a role model for her positive attitude and determination. I won't make it without you.

Having the support and guidance of these wonderful people has been an incredible stroke of luck. I am truly grateful for this good fortune and the role they have played in my journey.

# Abstract

In recent years, with the development of autonomous driving technology, 3D reconstruction from street-view perspectives has become a focal point for many researchers. However, due to the characteristics of street-view perspectives, this task has consistently faced numerous challenges. Additionally, in practical scenarios, camera calibration accompanied by Global Navigation Satellite System (GNSS) / Inertial measurement unit (IMU) systems will incur higher equipment costs than camera-only systems. Meanwhile, using only one monocular commercial camera for 3D reconstruction can also reduce the equipment cost of this task. On the other hand, the emergence of Neural Radiance Fields (NeRF) technology has introduced new approaches to the task of 3D reconstruction, leading to the proliferation of numerous algorithms based on NeRF of reconstructing 3D scenes [28, 66]. However, most existing research focuses on standardized and calibrated datasets or is limited by the need for multi-view inputs or the inclusion of LiDAR data to deal with street-view reconstruction [35, 15, 56].

In this work, a comprehensive workflow from street-view video to dense 3D reconstruction with one monocular camera is developed. Using a commercial camera, this workflow aligns video capturing, camera calibration, pose estimation, and 3D reconstruction with evaluation for practical autonomous driving applications. To deal with the characteristics of street-view scenes, image segmentation and image inpainting are also involved in data preprocessing steps in this pipeline. Meanwhile, two different methods, Structure-from-Motion (SfM) and Direct Sparsity Odometry (DSO), are tested and compared for the camera calibration and pose estimation tasks. To address the problem of street-view reconstruction with calibration information, streetsurf [15] is applied for our monocular dataset. The performance of such a GoPro dataset has achieved an average Peak Signal-to-Noise Ratio (PSNR) of 30.50, Structural Similarity Index (SSIM) of 0.934, and Root Mean Square Error (RMSE) of 1.30m.

In summary, this thesis sets out a successful workflow for monocular vision-based 3D reconstruction in street-view scenes. This research provides valuable insights and directions for using uncalibrated commercial cameras in autonomous driving applications, addressing the practical challenges of real-world scenarios.

**Keywords:** uncalibrated camera, street-view 3D reconstruction, monocular camera.

# Contents

<b>Acknowledgments</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Theoretical Background . . . . .	2
1.2.1 Camera model . . . . .	2
1.2.2 Street-view scene . . . . .	5
1.2.3 Camera calibration . . . . .	7
1.2.4 Pose estimation . . . . .	7
1.2.5 3D Representation and reconstruction . . . . .	9
1.2.6 Priors for scene representation . . . . .	9
1.2.7 Neural Radiance Fields (NeRF) . . . . .	10
1.3 Related work . . . . .	12
1.3.1 Street-view scene editing . . . . .	12
1.3.2 Uncalibrated camera . . . . .	14
1.3.3 Monocular 3D reconstruction . . . . .	15
1.4 Structure and content . . . . .	17
<b>2 Methodology</b>	<b>19</b>
2.1 Overview . . . . .	19
2.2 Video capturing and scene editing . . . . .	19
2.2.1 Video capturing . . . . .	20
2.2.2 Sky mask . . . . .	20
2.2.3 Dynamic object . . . . .	22
2.3 Camera calibration and pose extraction . . . . .	23
2.3.1 COLMAP . . . . .	24
2.3.2 Direct Sparse Odometry (DSO) . . . . .	24
2.3.3 Comparison on COLMAP and Direct Sparsity Odometry (DSO)	25
2.3.4 Pose transformation . . . . .	26
2.4 3D reconstruction . . . . .	27
2.4.1 Supervision for normal and depth . . . . .	27

*Contents*

---

2.4.2	Streetsurf . . . . .	29
2.4.3	Multi-View Stereo (MVS) . . . . .	29
2.5	Evaluation . . . . .	30
<b>3</b>	<b>Experiments</b>	<b>31</b>
3.1	Datasets . . . . .	31
3.1.1	GoPro Dataset . . . . .	31
3.1.2	KITTI Dataset . . . . .	33
3.2	Baseline . . . . .	33
3.3	Validation . . . . .	34
3.3.1	Quantitative metrics . . . . .	34
3.3.2	Qualitative methods . . . . .	36
3.3.3	Results . . . . .	36
<b>4</b>	<b>Discussion</b>	<b>47</b>
4.1	Calibration and reconstruction . . . . .	47
4.2	Photometric difference and reconstruction . . . . .	49
4.3	Overlapping rate and reconstruction . . . . .	50
4.4	Geometric cues from Multi-View Stereo (MVS) and Streetsurf . . . . .	50
4.5	Dynamic objects . . . . .	52
<b>5</b>	<b>Conclusion</b>	<b>53</b>
5.1	Summary . . . . .	53
5.2	Limitations . . . . .	54
5.3	Future work . . . . .	55
	<b>Abbreviations</b>	<b>56</b>
	<b>List of Figures</b>	<b>58</b>
	<b>List of Tables</b>	<b>60</b>
	<b>Bibliography</b>	<b>61</b>

# 1 Introduction

## 1.1 Motivation

3D reconstruction for street views plays an important role in autonomous driving now [56]. Nevertheless, in typical scenarios, the street-view data accessible to moving vehicles is inherently unbounded, sparsely overlapping, and contains highly dynamic objects [15]. Additionally, the pose information pertaining to the commonly available camera models remains uncalibrated in authentic application situations. Hence, the process of 3D reconstruction attained through street-view video capturing constitutes a challenging undertaking. However, the existing camera calibration methods, for example [3, 12, 39], along with street-view reconstruction algorithm [15, 56], provide us the possibility to such treatment of extracting 3D dense meshes for self-driving cars. Our study aims to perform 3D reconstruction for street views using a single monocular camera, with the expectation of establishing a complete and efficient process from street-view data acquisition to 3D depth information.

In recent years, NeRF [28] has achieved remarkably notable results in the field of scene reconstruction and rendering. It can provide impressive novel view synthesis with implicit neural representation using a collection of posed images. However, the original NeRF requires object-centric camera views with high overlap and bounded scenes. Closely following there is a series of various studies based on NeRF. Some of them focus on reconstruction for static large-scale outdoor scenes with detailed structures [25], or address the presentation of street view with LiDAR as additional supervision [35]. Besides, some studies also extend the neural surface reconstruction with images captured on moving cars using the implicit method [15, 56]. Moreover, most recent studies focus on 3D reconstruction for self-driving cars employing the standard and calibrated datasets [24, 31], and there are also studies demonstrating NeRF using uncalibrated images with COLMAP [39], but limited to indoor and object-centric scenes [19].

In this research context, this thesis aims to address the following research questions:

- Can a comprehensive workflow be built up from street-view video capturing to dense 3D reconstruction with only one monocular commercial camera?
- Can video capturing, camera calibration, and pose estimation with the procedure



of 3D reconstruction and evaluation for reconstruction be aligned in the practical application of autonomous driving?

- To what extent can the issues encountered in real-world scenarios, distinct from standardized data, be identified, and solutions to these challenges be raised?

This thesis outlines a basic workflow for obtaining and evaluating reconstruction accuracy from videos. In the video acquisition stage, an uncalibrated monocular commercial camera is employed to capture information on the street scene. Subsequently, these videos undergo preprocessing with semantic segmentation and image inpainting aimed at eliminating interference during camera calibration and 3D reconstruction processes. Following this, COLMAP [39] is utilized for camera calibration and pose estimation. With the obtained pose information and preprocessed images, Streetsurf [15] is applied for 3D reconstruction. Finally, control points serving as ground truth are utilized for accuracy assessment. The implementation of this work is available in the GitHub repository <sup>1</sup>.

## 1.2 Theoretical Background

This chapter provides an introduction to the theoretical background of this work. First, the camera model and corresponding parameters are introduced. Then, the characteristics of street-view scenes are analyzed, and the methods for scene editing are also presented. Next, the principles and classifications of camera calibration and pose estimation are also analyzed. After that, the methods for 3D scene representation and the corresponding algorithmic principles are introduced. Additionally, the role of priors in scene reconstruction is explained as well. Finally, it elaborates on the important concept of NeRF discussed in this paper.

### 1.2.1 Camera model

Camera models have been playing a vital role in modern 3D reconstruction, being applied across diverse fields, including autonomous driving and robotics. Both traditional 3D reconstruction algorithms such as MVS [40] and state-of-the-art methods based on deep learning networks such as NeRF [28] and 3D Gaussian splatting [20] require basic parameters of the camera model as input. Therefore, Understanding the camera model and its imaging process is important for achieving the goal of 3D reconstruction.

---

<sup>1</sup>[https://github.com/yue-t99/MT\\_3Dreconstruction](https://github.com/yue-t99/MT_3Dreconstruction)

Generally, the camera model refers to the projection from world coordinates to pixel coordinates in images, which can be represented as follows [1].

$$\mathbf{x} \approx \mathbf{P}\mathbf{X} \quad (1.1)$$

where  $\mathbf{x} = [x \ y \ 1]^T$  refer to the homogeneous image coordinates,  $\mathbf{X} = [X_w \ Y_w \ Z_w \ 1]^T$  are the homogeneous world coordinates, and  $P$  represents the perspective projection matrix between them. The parameters of the  $\mathbf{P}$  matrix introduce the information from focal length and principle points. In this thesis, as the video data used in this experiment has been processed by the built-in algorithm of the GoPro camera to eliminate lens distortion, using a pinhole camera model is sufficient. Therefore, this chapter will focus on discussing the parameter settings of this model.

For the pinhole model, Eq. 1.1 can be described as Eq. 1.2 [2], and the transfer matrix can be divided into 2 parts: intrinsic parameters and extrinsic parameters. Fig. 1.1 shows how a real-world point is connected with a pixel in an image in a linear model.

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1.2)$$

### Intrinsic parameters

Intrinsic parameters are parameters related to the intrinsic characteristics of the camera itself, and they include the transformation from image coordinate system  $P(x, y)$  to camera coordinate system  $P_c(X_c, Y_c, Z_c)$ . In Eq 1.2, the intrinsic parameters refer to  $f_x, f_y, c_x, c_y$ . The distance between the origin point  $O_c$  and the image plane is  $(f_x, f_y)$ , and the image coordinate of the projected original point  $O$  is  $(c_x, c_y)$ . As shown in Fig 1.1, based on the principle of perspective projection, each pixel in the image should obey the rule:

$$x = f_x \cdot \frac{x_c}{z_c}, \quad y = f_y \cdot \frac{y_c}{z_c} \quad (1.3)$$

According to Eq 1.3, the transformation from image coordinate system  $P(x, y)$  to camera coordinate system  $P_c(X_c, Y_c, Z_c)$  can be calculated.

### Extrinsic parameters

Extrinsic parameters are parameters related to the pose information of the camera in the world coordinate system, and they include the transformation from camera

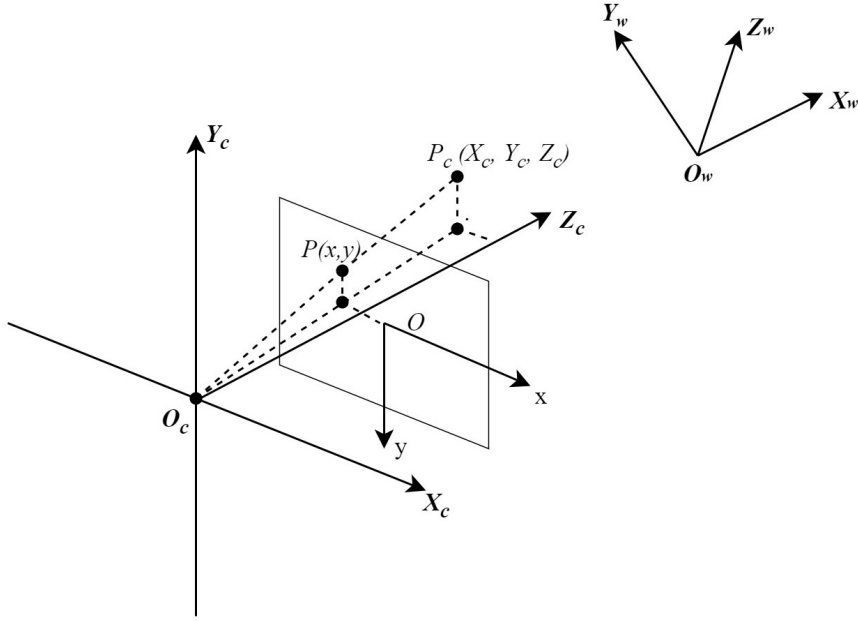


Figure 1.1: Imaging process of pinhole model

coordinate system  $P_c(X_c, Y_c, Z_c)$  to world coordinate system  $P_w(X_w, Y_w, Z_w)$ . This kind of transformation is rigid transformation. In Eq 1.4, the extrinsic parameters refer to the affine transformation matrix  $1/s \cdot [\mathbf{R}|\mathbf{T}]$ , where  $\mathbf{R}$  refers to the rotation matrix and  $\mathbf{T}$  refers to the translation from camera coordinate system to world coordinate system. As shown in Fig 1.1, based on the principle of rigid transformation, the origin point  $O_c$  from the camera coordinate system to the world coordinate system  $O_w$  is:

$$s \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1.4)$$

Where  $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$  and  $\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ .

Combining Eq 1.3 and Eq 1.4, the final equation Eq 1.2 can be extracted, and the projection procedure can be described as Fig 1.2.

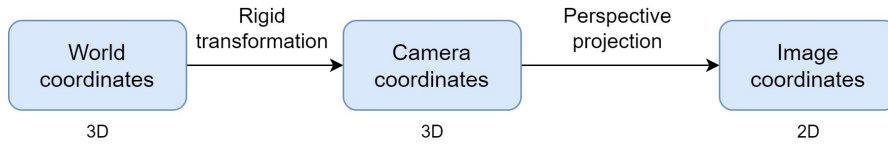


Figure 1.2: Transformation in between image coordinate, camera coordinate and world coordinate with intrinsics and extrinsics

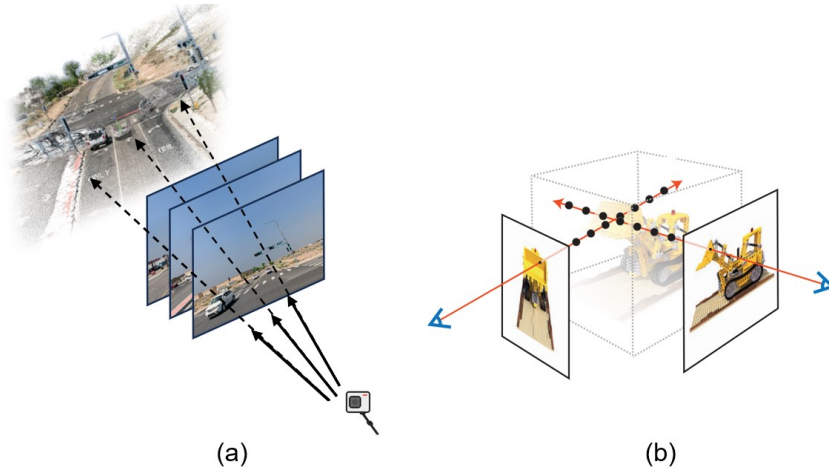


Figure 1.3: Difference between street-view scene and typical Nerf scene: (a) street-view scene, (b) typical Nerf scene, adapted from [28]

## 1.2.2 Street-view scene

### Characteristics of street-view scene

The street-view scene has many characteristics that are distinct from other scenes or officially released datasets. On one hand, as the vehicle moves forward, the camera’s perspective remains relatively constant. According to the principles of projection, areas in the infinite-far part would remain unchanged. On the other hand, unlike classical NeRF scenes, the image set from the vehicle’s perspective is not object-centroid or bounded [15]. Instead, the image rays can extend continuously along with the viewpoint (see Fig 1.3). Additionally, since the video data is real-captured on the road, there is a great deal of dynamic objects within the scene, making it more challenging to represent the static part.

The characteristics of street-view scenes applied for autonomous driving in this experiment bring a great challenge to pose estimation and 3D reconstruction. For pose

estimation, the parallax between point pairs in the images represents the motion of the camera [16]. Due to the camera's ability to capture the infinite distance in the street-view scene, the pixel coordinates in this area remain constant regardless of the camera's position changes. This kind of characteristic results in very small parallax, leading to the failure of camera pose estimation. Meanwhile, the volume rendering of NeRF-based method relies on the idea of ray tracing [28], and several current work [49, 58] introduced the idea of Signed distance function (SDF) for surface reconstruction in it. However, as shown in Fig 1.3, in the street-view scene, the perspective along ray tracing is always towards the infinite, not the target object. And the object-centroid designed SDF is not suitable and applicable for the street-view scene. Last but not least, in this thesis, mesh reconstruction aims to capture the information of the static part, such as the road. The temporal variations of dynamic objects are not considered, yet they will disturb the rendering procedure of each frame.

### **Image semantic segmentation**

In order to solve the mentioned problems in street-view scene, two important concepts are applied in this thesis: image segmentation and video inpainting, both of which are deep learning-based methods used for editing the street-view scene.

Image segmentation refers to the process of assigning a semantic label to each pixel in an image or partitioning the image into individual objects [29]. In this thesis, image semantic segmentation is included to address the challenge of extracting the infinite-far regions or dynamic objects in the autonomous-driving scene, and these regions usually have typical semantic characteristics, including sky or car. Therefore, the selected model should have achieved good accuracy on autonomous driving datasets such as Cityscapes [6] and be able to efficiently generate segmentation results with a zero-shot solution under low-computation cost conditions.

### **Video inpainting**

Video inpainting is also a challenging task in the field of computer vision. It refers to naturally and realistically completing missing parts of a video [33]. Video inpainting shares a very similar concept with image inpainting but deals with a sequence of images with temporal information. During the development of video and image inpainting, most of the traditional methods usually struggled to restore semantic texture information [67].

To edit the autonomous scene, image semantic segmentation can provide the mask of a target with a specific semantic label in the scene, with such a mask, inpainting models can be used for filling in the desired content. In this way, target objects can be

erased [63]. Considering the characteristics of the street-view scene and the size of the dataset, inpainting models are ideally expected to ensure consistency between frames to prevent semantic errors. Additionally, the selected model should not be sensitive to camera motion and have manageable computational requirements.

### 1.2.3 Camera calibration

The process of obtaining the geometric imaging model parameters of a camera through experimental and computational methods is called camera calibration [32]. These geometric imaging model parameters include both the linear transformation parameters of the pinhole model and the nonlinear distortion parameters corresponding to camera distortions. Therefore, based on this, traditional camera calibration methods categorize past research on this issue into "Direct Linear Transformation Method," "Nonlinear Optimization Method," and "Two-Step Method." [32] However, these calibration methods rely on reference control points in world coordinates or external parameters of the camera. Apart from that, Dr. Zhang Zhengyou proposed a camera calibration method based on a calibration board [68], which also requires a chess board to serve as external additional information. When there is no strict requirement for the precision of the intrinsic matrix, an alternative choice is to estimate an initial value for it and update the intrinsic parameters during the incremental reconstruction process. This is the method used by COLMAP for three-dimensional reconstruction when there are no intrinsic parameters available [39].

### 1.2.4 Pose estimation

As presented in section 1.2.1, extrinsic parameters are related to the camera's pose information. Therefore, the concept of pose estimation is to determine the rigid transformation parameters in Eq. 1.4. In general, the process of pose estimation is always combined with sparse 3D reconstruction, and such algorithms can be divided into three categories: visual Simultaneous Localization and Mapping (SLAM), visual odometry, and Structure-from-Motion (SfM) [43], see Fig 1.4.

These methods are interconnected, and each has its own limitations. Traditional SfM methods rely on the point correspondences to calculate relative camera poses [39]. However, due to the principles of Scale Invariant Feature Transform (SIFT) and Bundle Adjustment (BA) on which SfM is based, it is limited by computational complexity [38]. As a result, most existing SfM-based systems are offline, such as COLMAP [39] and VisualSFM [53]. Visual odometry can be viewed as a special case of SfM. Unlike SfM, Visual odometry does not consider global optimization, but only several frames and pixel changes in a slight window [60]. The difference between visual odometry

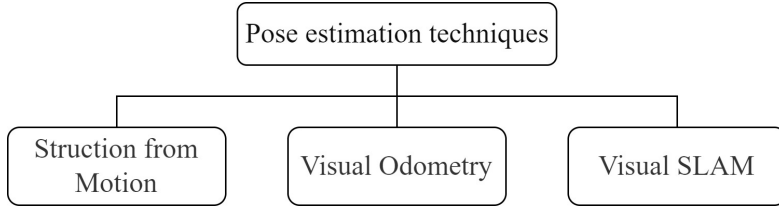


Figure 1.4: Taxonomy of pose estimation techniques

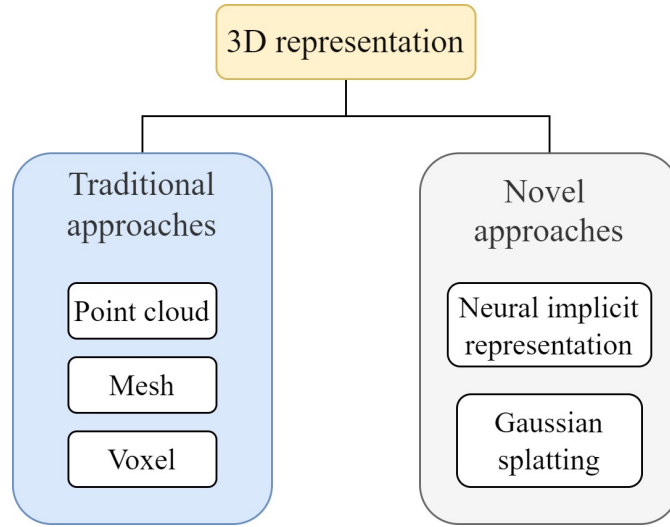


Figure 1.5: 3D representation approaches

and visual SLAM is that visual SLAM takes the global map optimization into consideration [43]. However, when faced with dynamic objects in the scene, many SLAM methods are based on the idea of segmenting the scene into dynamic objects and static scenes. This approach is effective only when the number of dynamic objects is small, and they are not dominant in all frames [38].

While minimizing computational complexity is also one of the goals of this work, computational cost is not the primary concern. Instead, the accuracy of camera poses directly impacts the reconstruction results. Therefore, addressing the complexities of street-view perspectives and providing accurate pose information is the core of the camera pose estimation section in this thesis.

### 1.2.5 3D Representation and reconstruction

In decades, image-based 3D reconstruction has become a popular topic in computer vision tasks. The concept of image-based 3D reconstruction refers to recovering the 3D representation of the original scene from available 2D image information in one or multiple images [37]. There has been a significant evolution in the representation methods for 3D reconstruction in recent years. Based on how the 3D scenes are represented, 3D reconstruction algorithms can be categorized into traditional representation-based 3D reconstruction and novel representation-based 3D reconstruction [7], as shown in Fig 1.5. The graphical representation of 3D space can be divided into two parts: appearance and rendering. Traditional 3D representation methods for appearance are typically rigid, based on Euclidean space or geometric shapes to represent 3D objects. In contrast, novel 3D spatial representation methods render appearance as continuous, non-rigid implicit representations, not capturing precise geometric shapes of objects but focusing on view synthesis of objects.

As for traditional 3D scene representations such as point clouds and voxels, several 3D reconstruction algorithms already provide robust solutions, such as MVS [40]. On the other hand, with the development of computer graphics and advancements in deep learning models, several transformative 3D representation methods have garnered significant attention from researchers. Examples include NeRF [28] and 3D Gaussian Splatting (3DGS) [21].

### 1.2.6 Priors for scene representation

Many reconstruction methods based on deep learning models have limitations and are inadequate for addressing various reconstruction problems in different scenes. Therefore, many researchers incorporate prior constraints into their algorithms to improve their effectiveness.

As outlined in Section 1.2.5, the problem of 3D reconstruction from 2D images can be understood as Eq. 1.5, where  $I = \{I_1, I_2, I_3, \dots\}$ , refers to a set of 2D images, and  $\mathbb{R}^3$  is the representation of 3D scene. With this principle, deep learning models train and optimize 3D spatial representations by designing appropriate loss functions, as shown in Eq. 1.6. In deep learning tasks, the loss function plays a crucial role. It is primarily used to measure the difference between the model's predicted results and the actual targets, thereby guiding the optimization of model parameters.

$$f : I \rightarrow S = \{p | p \text{ in } \mathbb{R}^3\} \quad (1.5)$$

$$L = L\{\hat{S} = f(I); S\} \rightarrow \min \quad (1.6)$$



In certain tasks, prior knowledge can be introduced by modifying the loss function. Assuming we have prior knowledge  $A$  with additional information, we can introduce a regularization term based on  $A$  into the loss function, as shown in Eq. 1.7, where  $g$  refers to the presentation function of prior  $A$  in the 3D scene.

$$L_{\text{prior}}(A) = \sum_{i=1}^n (g(\hat{S}) - A_i)^2 \quad (1.7)$$

$$L = L(\hat{S}; S) + \lambda L_{\text{prior}}(A)$$

This approach effectively incorporates prior knowledge into deep learning models, enhancing their generalization ability and robustness. Many studies have attempted similar approaches, especially when the geometry accuracy generated by NeRF-based methods is not satisfactory, for instance, DietNeRF [18] utilizes a pre-trained visual encoder to extract semantic information as prior knowledge to aid rendering, NeuRIS [47] uses pre-trained normals to enhance the precision of geometry, similarly, in 3D Gaussian splatting [20], sparse point clouds generated by SfM are applied for initialization. Going through these methods, it's evident that most of the additional prior information is sourced either from pre-trained models in other algorithms or from sparse point clouds obtained through traditional methods, and these priors either enhance semantics or improve geometric accuracy. Although many new 3D reconstruction approaches offer novel rendering and viewpoint synthesis methods, this study focuses more on the accuracy of the reconstructed geometry. Therefore, this thesis mainly concentrates on how to leverage externally acquired geometrical cues to improve the reconstruction accuracy.

### 1.2.7 NeRF

NeRF is a 3D reconstruction method first introduced by Ben Mildenhall et al. in their paper [28]. It has become a focal point of research in the field of 3D reconstruction in recent years. Its core idea is to learn the radiance or color value of each spatial point in the scene and use this information to render an image of the scene.

Unlike traditional 3D representation methods, NeRF sets the problem of 3D scene view synthesis into a function of its radiance  $C(x)$ . This function is typically modeled by a Multilayer perceptron (MLP). As shown in Fig 1.6, at any given point, the radiance  $c$  is determined solely by the position information represented by  $(x, y, z)$  and the viewing direction information represented by  $\sigma$  and  $\theta$ . This process can be represented by Equation 1.8. NeRF also learns the density  $\sigma(x)$  (or opacity factor) of each point in

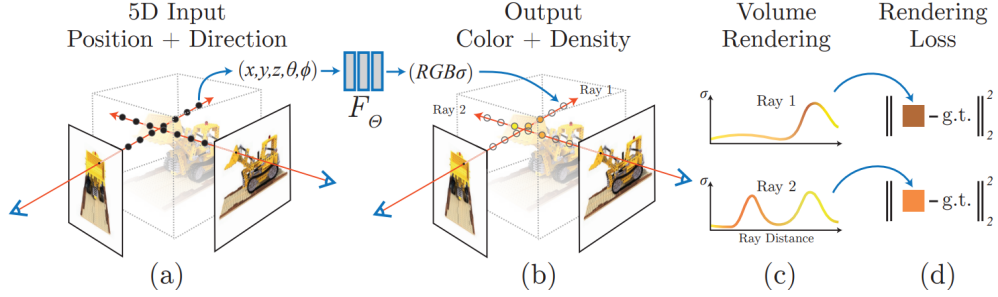


Figure 1.6: Principle of NeRF, adapted from [28]

the scene to better handle occlusion relationships.

$$f : (x, y, z, \sigma, \theta) \xrightarrow{\text{MLP}} C(x), \text{ which refers to RGB} \quad (1.8)$$

The positional information of this 3D spatial point is determined using the principles of ray tracing. As shown in Fig 1.6, for each pixel in the generated image, a ray is cast through the scene. The starting point of this ray is the camera origin, and the endpoint is the coordinate of the pixel in 3D space. In this way, the ray is defined by the camera position and the pixel position. The ray tracing algorithm samples the scene along each ray to determine the intersection points with objects in the scene. With the designed MLP, information about density and color can be gathered.

Once the ray tracing algorithm gathers the sampled points in the scene, NeRF employs the volume rendering equation to compute the color of each pixel, represented in Eq. 1.9. This equation integrates the density, color, and transparency between adjacent sample points along the ray to determine the final color value of the pixel from a given viewpoint.

$$L(\mathbf{p}_{\text{pixel}}) = \int_{t_{\min}}^{t_{\max}} \sigma(\mathbf{x}(t))C(\mathbf{x}(t))e^{-\int_{t_{\min}}^t \sigma(\mathbf{x}(s))ds} dt \quad (1.9)$$

here the density  $\sigma(\mathbf{x}(t))$  determines how much each point contributes to the final color, while the color  $C(\mathbf{x}(t))$  specifies what that contribution is. The exponential term  $e^{-\int_{t_{\min}}^t \sigma(\mathbf{x}(s))ds}$  accounts for how much light is attenuated by passing through the scene, reducing the influence of points that are deeper in the scene and have more density in front of them.

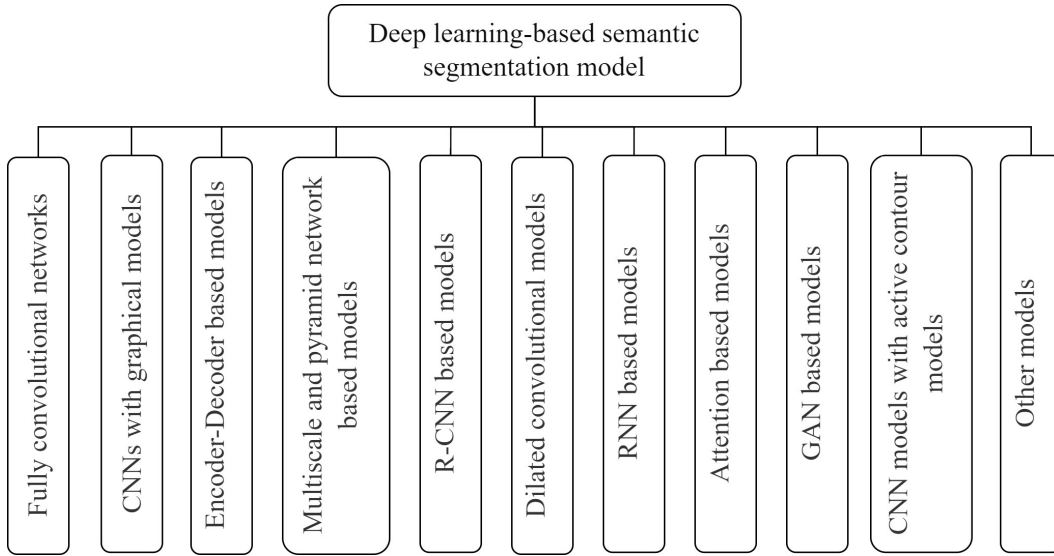


Figure 1.7: Taxonomy of image segmentation models, adapted from [29]

## 1.3 Related work

### 1.3.1 Street-view scene editing

Scene editing refers to the process of adding, removing, or modifying elements in a scene. Editing operations can help to improve the quality of an image, remove unwanted elements, or add new content to meet the needs of a specific application [67]. A number of algorithms have been investigated to edit the scene from different levels, and these editing tools can be basically categorized into image-based editing, video-based editing and NeRF-based editing.

Effective street scene editing requires an in-depth understanding of the semantic content of a scene. This includes recognizing and understanding the different objects in the scene, their interrelationships, and the structural and semantic information of the overall scene. This process is challenging due to the complexity of street-view data, such as dynamic elements and perspective changes. Besides, as discussed in section 1.2.2, to deal with the problem of street-view editing, a zero-shot solution that is invariant with size should be applied. Meanwhile, the computational costs should also be considered.

### Image-based and video-based editing

Generally, image and video editing can be divided into two parts: semantic segmentation and inpainting with semantics. Nowadays, numerous scholars have conducted extensive research on these two tasks.

For image segmentation, deep-learning-based methods have become mainstream for the task of semantic segmentation, achieving very robust results, such as Fully convolutional networks (FCNs) [26] and Transformer-based models [34, 69, 50]. In general, these deep learning-based semantic segmentation methods can be classified into eleven categories, see Fig 1.7. Each of them has its own strengths and weaknesses, suitable for different scenarios and tasks. For example, Vision Transformer (ViT) [34] has shown an impressive performance tested on ImageNet [9], but it cannot deal with multi-scale images and requires large computation costs [55].

In these semantic segmentation models, many are already tested with autonomous driving benchmarks, such as KITTI [14] and Cityscape [6], providing pre-trained models that can be directly used for street-view editing. As one of the earliest fully convolutional network methods, the convolutional network-based model FCNs [26] has been tested on several autonomous driving scene datasets. Compared to the origin FCNs, Deeplab-v3 [5] can provide a pre-trained model with better accuracy performance on datasets with different scales. Meanwhile, transformer-based models such as Segformer [55], and the Segment Anything Model (SAM) [22] using Convolutional neural network (CNN)/ViT as the backbone have also shown impressive performance with autonomous datasets. Segformer [55] stands out as it can rapidly generate test results while controlling model parameters and computational complexity. Meanwhile, SAM [22] can generate different prompts and provide corresponding solutions.

With the rapid advancement of deep learning models, inpainting algorithms based on deep learning models are continuously emerging. For the task of image inpainting, CNN and Generative Adversarial Network (GAN) have been two major mainstream algorithms. Typical examples include DeepFill-v1 [61], Realfill [44], LaMa [42], ect. With the popularity of the attention mechanism, many researchers have also attempted to integrate transformers into the task of inpainting, such as [4].

Meanwhile, there are also inpainting algorithms that consider time-series video data. These deep-learning models for video inpainting can be generally categorized into four types, as shown in Fig 1.8. Similarly, each of these models has its own strengths and weaknesses. In most cases, 3D-CNN based approaches adjusted from 2D-CNN network, for example [17, 62], require high computation cost; For shift based approaches such as [71], they still remain the problem of misalignment within a limited temporal window; Flow guided approaches [57, 13] are sensitive to complex motion; And attention based approaches [65, 23] are facing the problem of inpainting resolution [67].

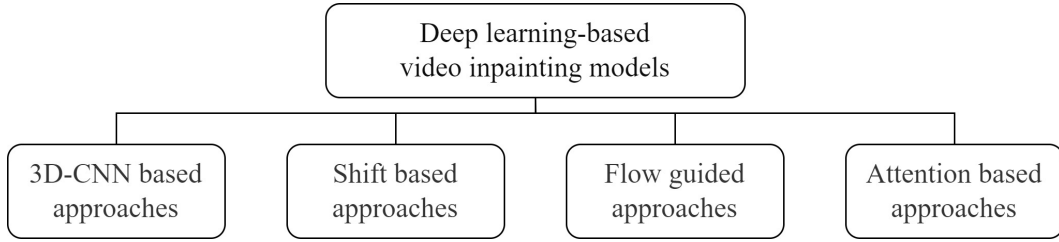


Figure 1.8: Taxonomy of video inpainting models, adapted from [67]

### NeRF-based editing

As inpainting algorithms have advanced, researchers have also begun to focus on techniques for editing 3D scenes. In the study Inpainting Anything [63], an inpainting interface that addresses different dimensions was published, including image-based, video-based, and NeRF-based inpainting. For NeRF-based inpainting, they directly edit the original image data, which is then used as input for NeRF to generate the 3D scene. Another example is Silvan Weder’s research [52], which involves performing inpainting on input RGB-D images and then incorporating a filter in the model computation to eliminate inconsistencies in the frames after inpainting. Similarly, some models [30, 51] consider integrating common inpainting methods, such as Diffusion, into NeRF inpainting to address noise issues in the original NeRF model.

Despite the significant advancements in image inpainting and NeRF, the specific area of NeRF-based inpainting for street-view scenes remains relatively underexplored.

### 1.3.2 Uncalibrated camera

In most existing datasets such as KITTI [14] and Waymo [41], cameras are often equipped with various well-calibrated information to enhance data quality. Such well-organized data usually include multi-view camera setups, strictly-calibrated intrinsic camera parameters, pose information from Global Navigation Satellite System (GNSS) and Inertial measurement unit (IMU), or LiDAR point cloud. These additional sources of information provide rich context and enable robust performance in many computer vision tasks. However, in this thesis, a single and uncalibrated monocular GoPro camera is utilized, which is quite different from the standard datasets. This setup poses unique challenges. To address these problems, existing algorithms can generally be divided into two categories: using pose estimation methods such as visual odometry, or integrating pose information with the 3D reconstruction process.

### Calibration and localization

Many camera calibration methods have achieved robust results across various scenarios. As described in Section 1.2.3, classical camera calibration methods include SfM-based, visual odometry, and visual SLAM approaches. The classical SfM [39] algorithm utilizes SIFT to extract feature points and perform feature matching. It employs the principle of bundle adjustment for 3D reconstruction, providing high-precision results even without intrinsic parameter information. Another example to solve this task is DSO [12], which is a visual odometry-based pose estimation method. In DSO, the algorithm leverages photometric error of pixel values and computes keyframe poses within a sliding window, enabling real-time pose estimation. Another noteworthy work is ORB-SLAM2 [3], which is based on the idea of SLAM and utilizes RGB-D data as input. Based on a framework of "tracking," "local mapping," and "loop closure detection," ORB-SLAM2 can achieve accurate pose estimation and map construction.

### Joint optimization with NeRF

In fact, many SLAM-based pose estimation methods already integrate dense reconstruction for joint optimization of pose and 3D models [3, 45]. However, NeRF-based 3D reconstruction methods for this purpose are still limited. Nevertheless, some researchers have proposed solutions to address this issue. For example, NICER-SLAM [70] is raised based on its existing research, solving the restrictions of RGB-D input. It combines the pose estimation process with neural field reconstruction, achieving promising results in indoor scene tests. Besides, Meuleman et al. [27] proposed a method that jointly optimization the poses and radiance fields for outdoor scenes that are unbounded, further extending the joint optimization algorithm with robustness and scalability for large-scale environments.

### 1.3.3 Monocular 3D reconstruction

Monocular vision-based 3D reconstruction faces numerous challenges. Firstly, in terms of input data, compared to official datasets that rely on LiDAR, multi-view images, and high-precision pose data, monocular videos provide limited viewpoints and lack precise pose information, making the reconstruction process more complex and unstable. Secondly, regarding reconstruction methods, although NeRF-based approaches have shown excellent reconstruction performance in specific scenarios, these methods usually depend on accurate camera pose estimation and extensive viewpoint coverage, which are difficult to achieve with monocular videos. Additionally, the complexity of the reconstruction scene further adds to the task's difficulty. Large-scale, unbounded open scenes such as street views typically contain intricate geometries and dynamic

elements, complicating the reconstruction process. Therefore, monocular vision-based 3D reconstruction must overcome challenges related to data limitations, methodological dependencies, and scene complexity.

As discussed in section 1.2.5, the reconstruction method can be categorized by the 3D scene representation method. 3D reconstruction algorithms based on traditional 3D representations have undergone a development from sparse to dense and from offline to real-time. Early classical 3D reconstruction methods such as SfM [39] utilize SIFT to extract matching key points and reconstruct sparse point clouds based on disparities. With given camera poses, MVS [40] can also acquire dense point clouds based on the principle of epipolar geometry. With the advancement in robotics, reconstruction algorithms based on SLAM [3, 8] have emerged extensively, overcoming the computational limitations of traditional SfM methods and enabling real-time acquisition of sparse point clouds. The rise of deep learning has also led to the development of many reconstruction methods based on traditional 3D representations [48], especially those utilizing MLP networks based on SDF functions, paving the way for new approaches to 3D representation.

Compared to traditional 3D reconstruction methods, recent implicit neural representation algorithms not only achieve highly realistic rendering effects but also generate high-precision, rigid geometric shapes using SDF [15]. The subsequent emergence of Gaussian splatting [21] has also left a profound impression due to its low computational cost and excellent rendering capabilities.

## **MVS**

One typical method for extracting traditional 3D scene representation is MVS [40]. It can extract dense point clouds based on epipolar theory with given intrinsic and extrinsic. However, searching for corresponding points for each pixel in every image is undoubtedly computationally expensive, however, due to epipolar geometry constraints, every pixel corresponds to a line in the image, which passes through the camera's optical center. With these epipolar lines, the matching problem between images can be transformed into a problem of matching points along these lines, thereby reducing the search space and improving the accuracy and efficiency of matching. After obtaining a large number of corresponding point pairs using epipolar line constraints, their depth values can be calculated. Combined with the camera's intrinsic parameters, three-dimensional coordinates can be obtained. Subsequently, dense reconstruction point cloud results can be obtained by performing point cloud fusion in different frames.

## NeRF-based Street-view reconstruction

Many researchers have focused on the problem of 3D reconstruction for outdoor street view scenes. SUDS [46] employs a hash table to encode the dynamic and static content, which is designed for handling a large number of objects and long-time sequences. However, it relies on sparse LiDAR data. S-NeRF [56] also focuses on large-scale backgrounds and moving objects in street scenes, utilizing sparse LiDAR data to enhance geometric structures. This approach is suitable for applications requiring high-precision reconstruction and moving object handling. However, it still has some dependency on LiDAR data and requires cameras mounted on the vehicle at different angles to ensure sufficient overlap. Streetsurf [15] proposed by Guo et al., on the other hand, does not require LiDAR data. It can leverage geometric priors and multi-view surface reconstruction techniques, making it possible for street-view scenes to extract geometry without LiDAR data. However, it relies on geometric priors from monocular models, which may face challenges in complex scenes. Moreover, it is limited by the presence of dynamic objects in the scene.

## 1.4 Structure and content

This thesis is organized into five main chapters, offering a comprehensive understanding of the research conducted and the findings obtained.

The first chapter **Introduction** sets the stage for the entire thesis, beginning with the motivation behind the research and providing the necessary theoretical background. It covers various fundamental concepts, including the camera model, street-view scene characteristics, camera calibration, pose estimation, 3D representation and reconstruction, and the NeRF framework. This chapter also includes a detailed related work section that reviews prior studies on street-view scene editing, uncalibrated camera usage, and monocular 3D reconstruction, highlighting the challenges and gaps in existing research.

The second chapter **Methodology** outlines the proposed methodology, starting with an overview and moving on to dataset preparation, including video capturing, sky mask creation, and handling dynamic objects. It then details the camera calibration and pose extraction processes, comparing COLMAP and DSO methods. The chapter concludes with the 3D reconstruction approach, supervision for normal and depth, the StreetSurf method, and MVS.

The third chapter **Experiments** provides a general look at the datasets used, including the GoPro and KITTI datasets, and the control points for validation. It presents the comparison methods and discusses the proposed approach's validation.



This chapter **Discussion** interprets the experimental results, discussing their implications, strengths, and limitations. It provides a critical analysis of the mesh as well as rendering results.

The final chapter **Conclusion** summarizes the key findings of the research, addresses its limitations, and suggests directions for future work.

## 2 Methodology

### 2.1 Overview

Fig 2.1 shows the general workflow of 3D reconstruction method presented in this thesis. In the first part, video capturing and scene editing, a GoPro camera is employed to capture street-level perspectives. Followed is the second step, where COLMAP [39] is selected to determine the pose information for each frame. In the third part, focusing on 3D reconstruction, the pre-trained models from Omnidata [11] are applied to obtain supervision for normal and depth and adjust into the Nerf-based Streetsurf [15] algorithm for reconstruction. Finally, accuracy assessment is conducted using Ground Control Points (GCPs) obtained through GNSS.

### 2.2 Video capturing and scene editing

As shown in fig 2.2, the pipeline of this section involves two main components: video capture and scene editing. And the scene editing step consists of two main parts: sky mask acquisition and removal using SegFormer [55] and dynamic object removal using Segment Anything [22] combined with the idea of Inpainting [63].

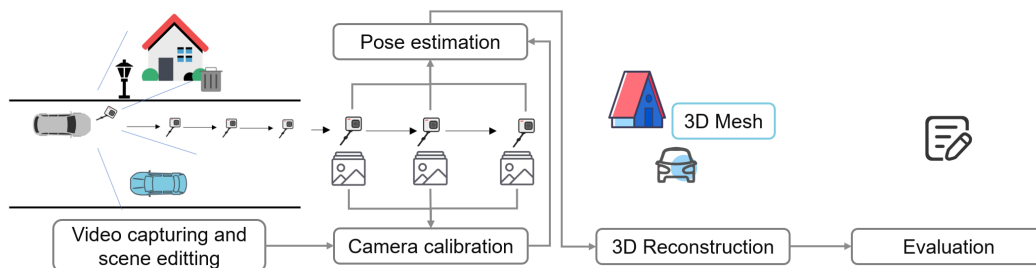


Figure 2.1: Overview of the method presented in this thesis

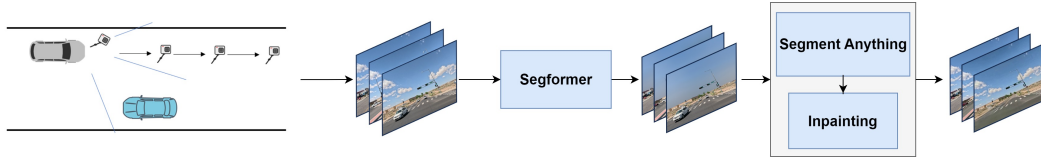


Figure 2.2: Pipeline of Video capturing and scene editing

### 2.2.1 Video capturing

This study aims to explore the challenges common cameras face in 3D reconstruction. We selected the commonly available GoPro camera for video acquisition to facilitate this investigation. During the video capture, a monocular GoPro camera is positioned at the windshield of a vehicle. As the vehicle proceeded normally, the GoPro captured the road conditions from this perspective. After capturing the video, frames are further processed by cropping them to retain only the relevant frames for analysis. Furthermore, scaling and cropping operations are conducted on the original videos to align the input images with pre-trained models applied in the work and reduce computational complexity during the 3D reconstruction process. As a result, the processed images are resized to 1024\*512.

### 2.2.2 Sky mask

The removal of the sky region from the videos aims to eliminate the influence of key points in these areas while estimating poses. The accuracy of pose estimation heavily relies on the quality of key points. However, as shown in fig 2.3, throughout the video sequence, many of the key points are distributed in the cloud part in the sky, while they remain in the same position in each frame. Suppose the raw, unprocessed images are directly used for pose estimation. In that case, both SfM and DSO algorithms assume that the camera position remains constant in each frame, leading to failure in pose estimation and sparse reconstruction. Therefore, finding a method to mitigate the impact of these key points located in the sky region on pose estimation is essential.

Although COLMAP supports filtering out keypoints in certain areas of the image by setting masks, manually setting mask ranges for each frame or generating masks for each sequence is time-consuming and labor-intensive. Additionally, compared to publicly available datasets including KITTI and Waymo, it's worth noting that in these proprietary datasets, the sky background has already been removed or blurred. Inspired by this, this thesis proposes an approach using semantic segmentation. Initially, the scene is segmented into different parts, then focusing on regions semantically labeled as "sky," a stitching process is applied. This process involves combining the Gaussian-

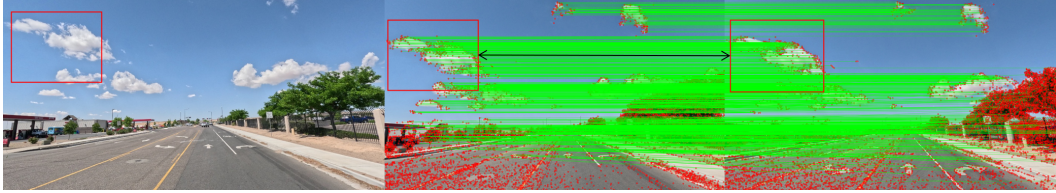


Figure 2.3: Matches for original image pair



Figure 2.4: Semantic segmentation results, (a) SegFormer, (b)Segment Anything

blurred image with the original image’s sky regions to effectively remove the influence of key points in those areas on pose estimation.

Deep learning methods have demonstrated robust performance in the field of semantic segmentation, considering the characteristics of different Deep learning based method discussed in section 1.2.2, Segformer [55] and Segment Anything [22] are tested for removing sky regions, both of which offer promptable pretrained models extracted from large sementation datasets. These pre-trained models can provide a zero-shot solution of semantic segmentation results on automatic driving scenes. Similar to ViT [10], the working principle of Segformer involves partitioning the image into a series of patches and then processing these patches using Transformer layers to capture the global semantic information of the image. While for SAM, it consists of an ViT-based image encoder and a fast prompt encoder/mask decoder, allowing reuse of image embeddings with different prompts, predicting masks from prompts in about 50ms in a web browser. Fig 2.4 shows the result of semantic segmentation on gopro frames. Comparing the performance of SegFormer and Segment Anything, Segformer exhibits faster speed. Additionally, its pre-trained models include specialized semantic masks for the sky. Therefore, Segformer is selected for the cloud-removing semantic segmentation task in this work. Fig 2.5 shows the image pair matching result after removing the cloud using a blurred sky mask. Evidently, the disturbance in cloud regions is not detected as a key point.

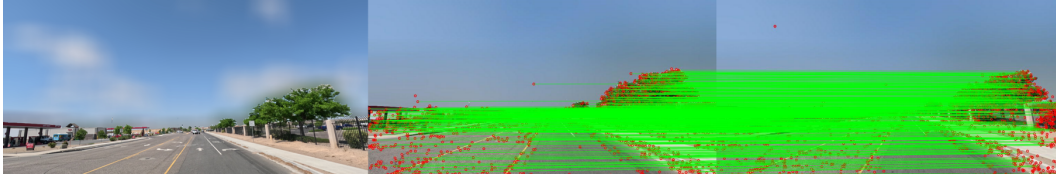


Figure 2.5: Matches for segmented image pair

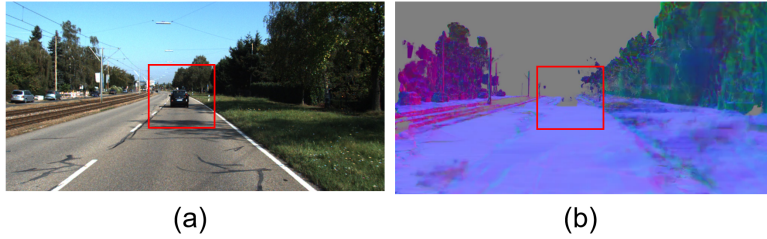


Figure 2.6: Mesh result of Streetsurf with a dynamic object in almost synchronized motion. (a) Original images, (b) Mesh result

### 2.2.3 Dynamic object

The purpose of removing dynamic objects within the scene is to minimize their impact on the Nerf-based 3D reconstruction method. In this work, Streetsurf [15] was chosen as the reconstruction method, which is significantly affected by dynamic objects in the scene. On one hand, if an object’s position varies in each frame, aligning differences in between frames becomes challenging. On the other hand, as illustrated in the figure, ray sampling direction aligns with the direction of the car’s movement. Consequently, if there are objects in the scene that move almost synchronously with the camera, determining their position through photometric loss becomes difficult. Fig 2.6 shows the mesh result with a car in almost synchronized motion with an ego car. When an object moves almost synchronously with the camera in the scene, the reconstruction results may exhibit significant errors, manifesting as large artifacts or gaps.

Directly editing the original video sequence could be a solution, utilizing semantic segmentation combined with the concept of inpainting, aiming to remove dynamic objects from the video. Inpainting anything [63] is employed in this step. The inpainting process involves utilizing SAM to obtain masks for dynamic objects in the video, followed by employing the Spatial-Temporal Transformer Network (STTN) [65] model to inpaint the masked regions. Here, STTN is proposed for high-quality video inpainting. Instead of using attention models independently for each frame, STTN learns a joint spatial-temporal transformer network. It can simultaneously fill missing



Figure 2.7: Before and after processed frames using Inpainting Anything. (a) Original frame, (b) Inpainted frame

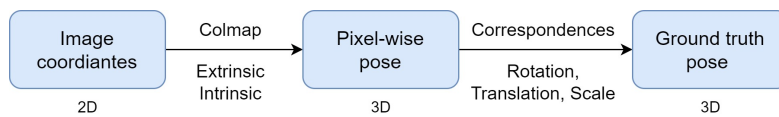


Figure 2.8: Pipeline of camera calibration and pose extraction

regions in all input frames using self-attention mechanisms. Firstly, frames containing dynamic objects are extracted from the sequence. Subsequently, this work utilizes the OpenCV library to obtain image coordinates via `setMouseCallback` clicks and set them as inputs for the SAM’s promote encode. After obtaining the processed video, the frames are finally collected from the video in chronological order. Fig 2.6 shows an example frame before and after processing using Inpainting Anything.

### 2.3 Camera calibration and pose extraction

In this part, two prominent methods for pose extraction are explored: COLMAP [39] and DSO [12]. In this section, our input consists solely of preprocessed image sequences. Due to the absence of a calibration board and corresponding intrinsic parameter information, camera calibration becomes a challenging problem. Without a calibration board, no known reference points or features exist in the images, making it difficult to establish correspondences and estimate camera parameters. Meanwhile, the GoPro camera has wide-angle lenses, which can introduce significant radial and tangential distortion. Calibrating such distortion accurately is crucial for geometrically correct image processing. Furthermore, only a monocular camera is utilized in this scenario, resulting in a single viewpoint. Additionally, the presence of numerous dynamic objects such as vehicles in the scene further complicates the problem of pose estimation. In such a context, DSO and SfM applied in COLMAP offer distinct advantages, the sparse point cloud produced and the quality of poses lead us to adopt COLMAP due to its

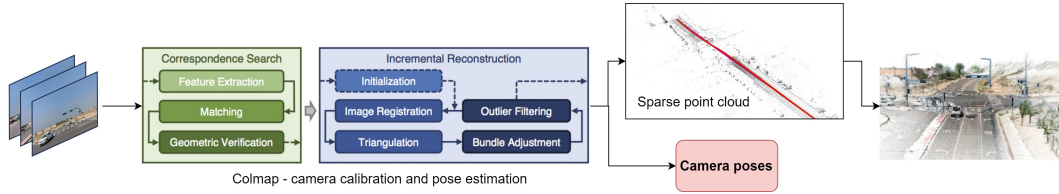


Figure 2.9: Pipeline of pose extraction using COLMAP, adapted from [39]

robustness and efficiency. Additionally, the pose scale ambiguity is fixed by applying control points that align the obtained camera matrices with world coordinates. Fig 2.8 shows us the pipeline of pose extraction in this section.

### 2.3.1 COLMAP

Fig 2.9 shows the pipeline of pose extraction using COLMAP. COLMAP is a general-purpose SfM [39] and MVS [40] pipeline interface, with an input data of a series of images only. Firstly, it is necessary to add masks to filter out these feature points. Due to the similarity in scenes across certain sequences, the feature points near the vanishing point remain relatively unchanged. This phenomenon can lead to reconstruction failure. In this part of the work, the mask range encompasses areas near the vanishing point of the images that remain unchanged. After that, feature extraction and feature matching are conducted. Except for setting the camera mask, all the parameters are set as default. Since the images obtained have already been corrected internally by the GoPro software system, the camera model is set as pinhole or simple pinhole. All images are set to share the same intrinsic parameters to obtain the corresponding geometric features. Furthermore, considering that the obtained images are acquired and arranged in chronological order, sequential matching is used in the feature-matching process. Following that, bundle-adjustment-based reconstruction is initiated by clicking "start reconstruction" in the window. After this step, a sparse point cloud, as well as the extrinsic and intrinsic parameters of all images, are extracted. With these results, MVS can be utilized to obtain dense reconstruction geometry.

### 2.3.2 Direct Sparse Odometry (DSO)

DSO is a sparse visual odometry method based on direct methods, it requires an input consisting of a series of images, information for camera mode and intrinsic parameters. However, the camera used in this work is a common commercial camera GoPro, and no strict calibration with a calibration board is performed. Therefore, its focal length is unknown. In the experiments with DSO, the intrinsic parameters are derived from

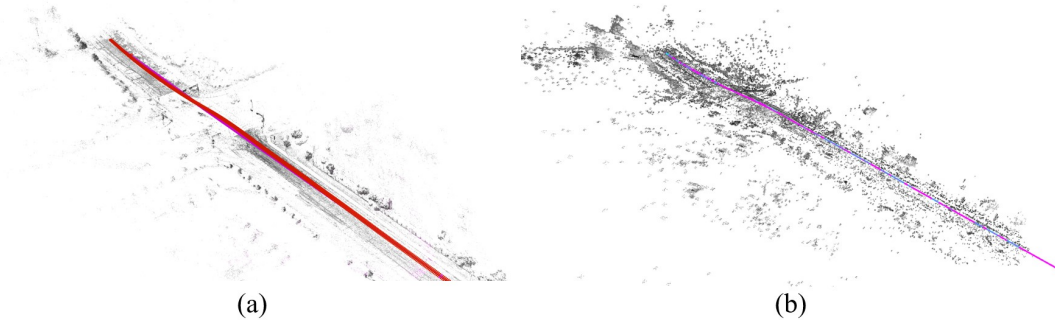


Figure 2.10: Cameras track and sparse point cloud extracted from (a) COLMAP, (b) DSO

the results of COLMAP after SfM reconstruction. Additionally, all image sequences are assumed as no photometric distortion. The output of DSO consists of pose information for selected keyframes.

### 2.3.3 Comparison on COLMAP and DSO

Table 2.1: Advantage and disadvantage of COLMAP and DSO

perspective	COLMAP	DSO
Noise in Sparse point cloud	Low	High
Time consumption	More than 1 hour	Real-time
Disturbance of Dynamic object	Low	High
Poses of frames	All	Only keyframes

Fig 2.10 shows the cameras' track information and sparse point cloud extracted from COLMAP and DSO. Considering the quality of the generated sparse point cloud, the point cloud quality from COLMAP reconstruction is superior, while the point cloud from DSO exhibits greater noise and difficulty in discerning geometric features in the scene. From the perspective of input data requirements, COLMAP is independent of any external camera calibration information, while DSO relies on provided intrinsic parameter information. However, from the standpoint of computational complexity and time consumption, COLMAP is time-consuming. In a CUDA environment with an RTX 3090 GPU, reconstructing and obtaining pose information for over 700 frames takes at least 1 hour. On the other hand, DSO can achieve real-time pose estimation. Considering the impact of scene complexity on reconstruction effectiveness, COLMAP's



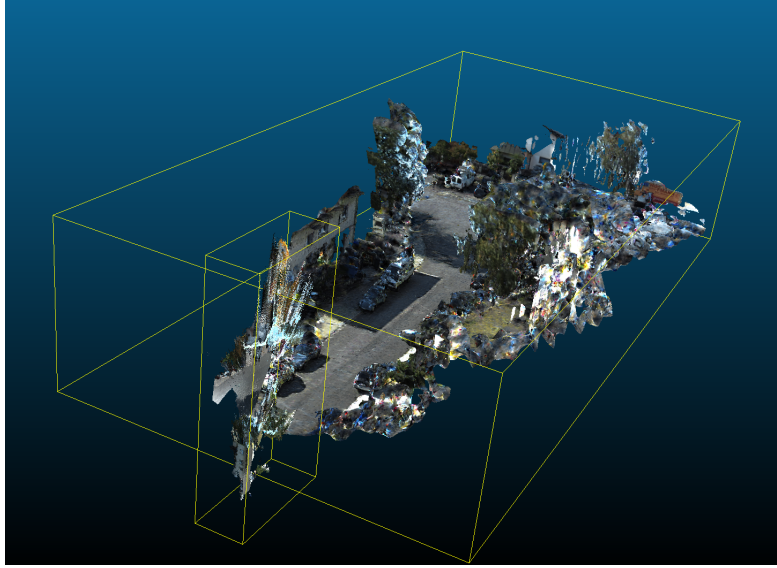


Figure 2.11: Difference in between point cloud extracted from pixel-wise pose and ground truth pose

reconstruction results and pose information are less susceptible to the influence of dynamic objects. However, DSO's quality of pose information is significantly affected by dynamic objects in the scene. In terms of the quality of pose information, COLMAP can obtain both intrinsic and extrinsic parameters for all frames, whereas, under real-time conditions, DSO can only retrieve pose information for keyframes. Considering the advantages and disadvantages outlined above, COLMAP is chosen as the main method of pose extraction.

### 2.3.4 Pose transformation

In the results obtained from COLMAP, due to the lack of correspondence between the camera and real-world pose information, the pose information derived is pixel-wise. This implies that there is an ambiguity in the transformation parameters for the obtained poses in the above steps, involving seven degrees of freedom: translation, scale and rotation. Fig 2.11 shows such a difference in between mesh extracted ground truth pose and colmap pose.

This transformation is essentially a rigid transformation from the pixel-wise world coordinate system to the ground truth world coordinate system, encompassing an affine relationship between 3D coordinates, as shown in Eq. 2.1. To solve this problem, a set of control points derived from ground truth is applied to calculate such a rigid

transformation. For these control points, corresponding coordinates on the dense point cloud obtained through MVS are manually selected. Thus, a set of control point pairs corresponding to the point cloud computed by COLMAP is obtained.

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \text{scale} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \\ \mathbf{z}_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (2.1)$$

Based on Eq. 2.1, firstly, the affine transformation between control points pairs is computed using the function *estimateAffine3D* in the OpenCV library. Utilizing this affine transformation, unit vectors along the x, y, and z axes are set to calculate the scale on different coordinate axes. Using these scale factors coordinates from COLMAP are rescaled to match those of the ground truth. Then, both the COLMAP and ground truth coordinates are subtracted with their centroid coordinates to align their origins. As a result, only a rotation matrix remains for the transformation between the two sets of coordinates. Through the theory of Singular Value Decomposition (SVD), this rotation matrix can be deposed, and with the rotation matrix R, the scale factors, and the translation T can also be extracted from Eq. 2.1.

## 2.4 3D reconstruction

The main method applied in this study is Streetsurf, while a traditional method for 3D reconstruction MVS is also implemented as a comparative reference in this work. In this section, the input data for 3D reconstruction consists of each frame of the images and their intrinsic and extrinsic parameters calibrated through COLMAP. Meanwhile, Streetsurf, as a 3D reconstruction algorithm based on Nerf, relies on depth maps and normal maps to constrain geometric loss when no Lidar data is available as additional input. Therefore, this chapter mainly consists of three parts: the first part involves generating depth maps and normal maps for each frame using pre-trained models released by Omnidata, the second part focuses on reconstructing the monocular visual image sequence using the Streetsurf method, and the third part serves as a reference, utilizing traditional MVS methods for 3D reconstruction. Fig 2.12 shows the basic pipeline of 3D reconstruction with Streetsurf.

### 2.4.1 Supervision for normal and depth

In this section, the depth map and normal map are designed to calculate the geometric errors in the selected 3D reconstruction algorithm Streetsurf. Streetsurf employs two strategies for calculating geometric loss arising from textureless regions and insufficient

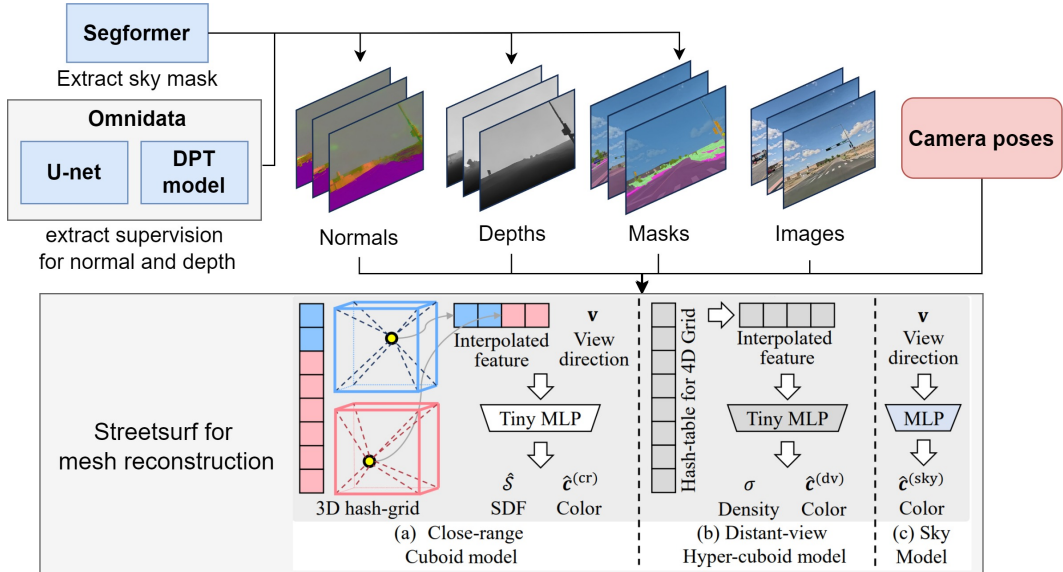


Figure 2.12: Pipeline of 3D reconstruction using Streetsurf, adapted from [15]

viewing angles. One strategy is designed for cases where Lidar data is available, and the other for cases where Lidar data is unavailable [15]. The idea of adapting mono normals and mono depths is inspired by MonoSDF [64], which is as shown in Eq. 2.2.

$$L_{\text{geometry}} = L_{\text{mono\_normal}} + \lambda L_{\text{mono\_depth}} \quad (2.2)$$

Here  $L_{\text{geometry}}$  is the geometric loss,  $L_{\text{mono\_normal}}$  refers to consistency on the volume-rendered normal and the 'ground truth' normal, and  $L_{\text{mono\_depth}}$  represents the consistency in between rendered expected depth and 'ground truth' depth [64].

In both the MonoSDF [64] and Streetsurf [15] methods, pre-trained models from Omnidata [11] are utilized to generate corresponding depth maps and normal maps for use as ground truth values. Similarly, this approach to obtain supervision for normals and depths for each frame is adopted in this study. For the estimation of normal, a standard U-Net [36] model was trained with a generated starter dataset for in-the-wild mode surface normal estimation. For the estimation of depth, a DPT-based [34] pre-trained model was extracted with a large start dataset for the human-level zero-shot solution. With the mentioned pre-trained model in Omnidata, a vision-based dataset with normals and depths is prepared for Streetsurf reconstruction.

### 2.4.2 Streetsurf

Streetsurf [15] serves as the primary 3D reconstruction method used in this thesis. It is specifically designed for autonomous driving and street-view scenes, employing a NeRF-based approach. While most NeRF-based algorithms are designed for rendering appearance using ray tracing and intended for centroid objects, the single viewpoint and unbounded scene pose significant challenges for reconstruction with such an image dataset extracted from street-view driving cars. As shown in Fig 2.12, to address the challenges arising from the characteristics of driving scenes image data, streetsurf introduces a model that divides the scene into three parts: the close-range cuboid model, distant-range hyper-cuboid model, and sky model. The core of NeRF-based algorithms lies in the implicit representation of three-dimensional scenes, relying on the design of the photometric loss function. For the close-range cuboid model, a NeuS [49] model is adjusted to describe the photometric loss; for the distance-range hyper-cuboid mode, a NeRF++ [66] is modified to calculate the photometric loss of this part; for the sky mode which is semantically segmented from Segformer [55], a directional MLP is set to calculate this photometric loss. By superimposing the mentioned photometric loss for different parts, an L1 photometric loss is employed to limit the difference between rendered pixel color and ground truth color [15].

Along with the geometric loss calculated in section 2.4.1, a sky mask loss, an entropy regularization loss to improve the model’s generalization capability and stability, an eikonal regularization loss for the geometric consistency and sparsity regularization loss aimed at encouraging sparsity in the model parameters are collected to describe this model, as shown in Eq. 2.3 [15]

$$L_{\text{all}} = L_{\text{photometric}} + \lambda_1 L_{\text{geometry}} + \lambda_2 L_{\text{mask}} + \lambda L_{\text{eikonal}} + \lambda L_{\text{sparsity}} + \lambda L_{\text{entropy}} \quad (2.3)$$

### 2.4.3 Multi-View Stereo (MVS)

For the 3D reconstruction task of street-view scenes, in addition to utilizing the state-of-the-art method Streetsurf, the traditional 3D reconstruction method MVS is also employed for comparison on the gopro dataset. In section 2.3.1, a sparse point cloud based on key points can be obtained using the calculated intrinsic and extrinsic parameters in SfM. However, as discussed in section 1.3.3, with the camera model information, MVS can obtain a point cloud with not only key points but all pixels.

In COLMAP, a dense reconstruction pipeline is set with the theory of MVS. Firstly, distortion correction is applied to the images. Then, depth and normal maps are computed for the undistorted images with the parameters from the camera model and epipolar theory. These depth and normal maps obtained for each frame are fused through point cloud fusion. Finally, the surface is estimated by meshing the point cloud.

Utilizing such a built-in module in COLMAP, MVS can provide a dense reconstruction result.

## 2.5 Evaluation

For the results of this work, Peak Signal-to-Noise Ratio (PSNR) and Root Mean Square Error (RMSE) of control points in the generated mesh or point cloud from TUM-FACADE [54] are used for evaluation. Some of the control point data originates from GNSS measurements and aerial image reconstruction results, while another set of the control point data comes from Mobile Laser Scanning (MLS) point cloud scans. Evaluation for the rigid geometry can be done by comparing the Euclidean distances between manually selected key points in the mesh results and the ground truth feature points. Additionally, PSNR and Structural Similarity Index (SSIM) can be used to assess the rendering quality.

# 3 Experiments

## 3.1 Datasets

In this thesis, experiments are conducted using street-view video datasets that comprise recordings captured from a mono-view camera mounted on the car or by hand, primarily focusing on urban road environments. There are two main data sources: monocular, street view perspective video acquisition using GoPro, and the KITTI dataset released by KIT [14]. Table 3.1 provides an overview of all the data used and their basic information. Besides, Fig 3.1 shows an overview of the video dataset perspective.

### 3.1.1 GoPro Dataset

This dataset was captured using a GoPro Hero11 monocular camera and is designed for the task of street-view reconstruction. The GoPro datasets were captured in two separate locations, consisting of five videos that recorded street-view traffic conditions from different roads. Four of these sequences were collected by mounting the GoPro on a vehicle, while one sequence was captured with a handheld GoPro.

one of them was captured on highway roads, located on *Latitude: 35.08539, Longitude: -106.73099 (WGS84)*, due to their complexity in traffic dynamics, lack of building features, and high speed, aiming to simulate real-world driving scenarios. Four of them were captured in the TUM Campus area, located on *Latitude: 48.14806, Longitude: 11.56583 (WGS84)*, with dense traffic, various buildings, and pedestrians.

Different from other officially released autonomous driving datasets, the GoPro datasets used in this experiment all come from GoPro Hero11 cameras, which possess the characteristics of simplicity and accessibility. Additionally, the precisely calibrated internal parameters of the camera are not provided in this dataset. In addition to the image sequences from different scenes, the varying capture modes help us better understand the challenges faced by an onboard camera perspective in a moving vehicle. Detailed information about the different GoPro sequences can be found in the table 3.1.

### 3 Experiments





DS-Scene	<p data-bbox="858 533 976 564">DS-Scene1</p> 
Campus-Scene	<p data-bbox="635 795 801 826">Campus-Scene1</p>  <p data-bbox="1050 795 1216 826">Campus-Scene2</p>  <p data-bbox="635 1025 801 1057">Campus-Scene3</p>  <p data-bbox="1050 1025 1216 1057">Campus-Scene4</p> 
KITTI	<p data-bbox="635 1281 801 1312">KITTI-road-0015</p>  <p data-bbox="1018 1281 1264 1312">KITTI-Residential-0035</p> 

Figure 3.1: Overview of the datasets in this experiments

### 3.1.2 KITTI Dataset

The KITTI dataset was set as a comparison group for the experiment. The KITTI dataset is a high-quality dataset widely used in computer vision and autonomous driving research. Two types of monocular images are used in this experiment: residential and road. The category “residential” contains image data collected in urban residential areas, recording vehicles traveling on residential streets. The data include a variety of typical streetscape elements such as residential buildings, pedestrians, parked vehicles, and greenbelts. And the category “road” contains image data collected on different types of roads, including urban roads and highways. In addition to one monocular camera, to increase the versatility and usefulness of the data, the KITTI dataset is equipped with GNSS and IMU, providing precise position information and inertial measurement data for path tracking and localization. Therefore, for the KITTI dataset, this experiment utilizes the provided precisely calibrated pose information for 3D reconstruction. The details of the KITTI dataset used are also summarized in the table 3.1

Table 3.1: Description for GoPro dataset and KITTI dataset

Index	Position	Dur(s) <sup>*</sup>	Frames <sup>*</sup>	Fps	Resolution	Mode
DS-Scene1	Highway	31	70	24	5312*2988	On car
Campus-Scene1	Campus	19	450	24	3840*2160	Handheld
Campus-Scene2	Campus	28	450	24	3840*2160	On car
Campus-Scene3	Campus	12	250	24	3840*2160	On car
Campus-Scene4	Campus	32	600	24	3840*2160	On car
KITTI-Residential-0035	City	13	50	10	1242*375	On car
KITTI-road-0015	City	30	70	10	1242*375	On car

<sup>\*</sup> *Dur* refers to the duration of the original captured video

<sup>\*</sup> *Frames* refers to the number of selected frames put in this experiment.

## 3.2 Baseline

Except for the experiment with streetsurf using the GoPro dataset as a baseline, the performance of the officially-released KITTI dataset [14] with the same reconstruction algorithms was also evaluated. Additionally, the reconstruction results with traditional dense reconstruction methods MVS [40] is set as a comparison. Meanwhile, the difference between sequences that have undergone dynamic object removal by the



inpainting method and those that have not are also compared using the two quantitative metrics PSNR and SSIM.

### 3.3 Validation

#### 3.3.1 Quantitative metrics

##### Peak Signal-to-Noise Ratio(PSNR)

PSNR is a commonly used evaluation metric in NeRF-based algorithms. It can be employed as an evaluation metric to quantify the discrepancy between the images generated by the model and the ground truth images. The images regenerated by the model are obtained by projecting the 3D points in the scene onto camera viewpoints. These reprojected images can then be compared against the real captured images.

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (3.1)$$

PSNR is a metric used to compare such a difference between the original images and the projected images, which can provide a numerical value that represents the quality of the reconstruction. In the context of NeRF, PSNR is used to evaluate how accurately the model can recreate the original scene. The PSNR is calculated based on Mean Squared Error (MSE) of pixel differences between the rendered image and ground truth image. Eq. 3.1 shows how MSE is calculated with pixels, where  $I(i, j)$  refers to the original pixel value in  $(i, j)$ ,  $K(i, j)$  refers to the pixel value of the rendered image. With Eq. 3.1, PSNR can be calculated as Eq. 3.2. Higher PSNR values indicate better rendering quality.

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) \quad (3.2)$$

##### Structural Similarity Index(SSIM)

SSIM is a perceptual metric that measures the similarity between two images. Unlike PSNR, which is based on pixel-to-pixel differences, SSIM considers changes in structural information, luminance, and contrast. The comparison with structure can be calculated via Eq. 3.3, the luminance difference can be calculated via Eq. 3.4, and the contrast difference can be calculated via Eq. 3.5, where  $\sigma_x$  refers to the mean values of image  $x$ ,  $\mu_x$  refers to the standard deviation of image  $x$ , and  $\sigma_x$  refers to the covariance of image  $x$ . Combining these equation together, the SSIM can be calculated as Eq. 3.6,

where usually  $\alpha$ ,  $\beta$ , and  $\gamma$  are equal to 1.

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \quad (3.3)$$

$$l(x, y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (3.4)$$

$$c(x, y) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (3.5)$$

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (3.6)$$

The SSIM can provide a better approximation of human visual perception. In NeRF, SSIM is used to assess the similarity of the rendered images to the ground truth, focusing on the preservation of structural details and overall image quality. Higher SSIM values indicate a higher degree of similarity and better visual quality.

#### Root Mean Square Error(RMSE)

RMSE is a standardized measure of a model's error in predicting quantitative data. It is calculated as the square root of the mean of the squared differences between the predicted and observed values, and Eq. 3.7 shows how the metrics of RMSE are calculated, where  $\hat{y} = (x, y, z)$ , refers to the 3D coordinates of control points.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2} \quad (3.7)$$

The RMSE is a single measure of the accuracy of the model, with smaller values indicating a better fit. In 3D geometry mesh evaluation, RMSE is used to quantify the difference between the mesh extracted from NeRF and the ground truth control points.

By computing the PSNR and SSIM between them, the metrics scores can be used to indicate the quality of rendering and mesh reconstruction. Meanwhile, the RMSE calculates the geometry difference between the coordinates on control points and the reconstructed 3D coordinates. The geometric accuracy of mesh reconstruction results was validated using the ground truth LiDAR point clouds or mesh results extracted from aerial images.

### 3.3.2 Qualitative methods

Four aspects were considered in this experiment to qualitatively evaluate the quality of the reconstructed mesh: surface smoothness, visual correctness, consistency of the depth map, and quality of the rendered output.

The smoothness of the mesh is assessed by examining the continuity and uniformity of the surfaces. A high-quality reconstruction should exhibit smooth transitions without noticeable artifacts or irregularities, which indicates a well-formed surface representation. Meanwhile, visual correctness is evaluated by checking the positional correctness of vertices and the overall geometry. Accurate reconstructions should closely match the original shapes and structures. Besides, depth map consistency is evaluated by analyzing the coherence of depth values across different views. A consistent depth map ensures that the depth information is reliable and aligns well across multiple perspectives, which is crucial for accurate 3D reconstruction. The visual quality of the rendered images is determined based on their realism and perceptual fidelity in novel view synthesis. High-quality renders should look visually convincing, with proper lighting, shading, and texture details that closely resemble real-world scenes.

### 3.3.3 Results

For the GoPro datasets, after acquiring the original video data, SegFormer [55] is used to blur the sky background. Subsequently, the processed image frames are used as input for COLMAP [39], which provides the corresponding intrinsic parameters and quaternions representing the extrinsic parameters. COLMAP can also generate a dense point cloud based on the MVS [40] principle. By comparing the coordinates of control points in this dense point cloud, the transformation matrix between the quaternions and the real world can be calculated. Using these parameters, we can construct the camera pose description file `scenario.pt` required by Streetsurf [15]. Additionally, for frames containing dynamic objects, these frames are segmented and the Inpainting Anything [63] is applied to remove the dynamic objects. Then, Omnidata [11]’s pre-trained models help to generate depth maps and normal maps, serving as a geometric prior in Streetsurf [15] under no-LiDAR conditions. Finally, this information is used as input for Streetsurf [15], which differs from the multi-view condition in the original literature by using image data captured by a single monocular camera for reconstruction.

For the KITTI dataset, since the sky background has already been removed and the dataset already includes camera intrinsics as well as IMU/GPS information, the intrinsic and extrinsic matrix can be directly calculated. These matrices can also be used to construct the camera pose description file `scenario.pt` required by Streetsurf. Dynamic objects in the scenes are not processed. However, it is noteworthy that two sets

of sequences were used in the experiments: one set contains dynamic objects (road-15) and the other does not (residential-35). The following processing is similar to that for the GoPro scenes. Omnidata is utilized to generate depth and normal maps, and the single monocular image data is used as input for Streetsurf for the experimental results.

### Evaluation metrics

Table 3.2 and Table 3.3 show the quantitative evaluation results of all the tested methods in our work. In this table, the tested data can be generally divided into three categories: Highway Scenes (DS-Scene1): These sequences represent complex highway environments with a large number of dynamic objects such as cars. Campus Scenes (Campus-Scene1 to Campus-Scene4): These sequences represent a city-view scene around the university campus with full-textured architecture. KITTI Dataset (KITTI-Residential-0035, KITTI-road-0015): These are specific scenes from the KITTI dataset focusing on residential and road environments.

Table 3.2: Quantitative evaluation of rendering in the 3D reconstruction result

Dataset	PSNR(db) $\uparrow$	SSIM $\uparrow$
DS-Scene1	33.26	0.975
Campus-Scene1	30.32	0.923
Campus-Scene2	28.22	0.895
Campus-Scene3	29.72	0.931
Campus-Scene4	30.98	0.946
<b>GoPro-average</b>	30.50	0.934
KITTI-Residential-0035	31.30	0.922
KITTI-road-0015	30.57	0.951
<b>KITTI-average</b>	30.94	0.937

Table 3.3: Quantitative evaluation of geometry accuracy in the 3D reconstruction result

GoPro-Dataset	Highway		Campus			Average
	Scene1	Scene1	Scene2	Scene3	Scene4	
RMSE(m) $\downarrow$	0.76	1.15	2.76	1.03	0.40	1.30

In Table 3.2, the PSNR and SSIM metrics in different scenes are represented. Among them, DS-Scene1 (highway scene) shows higher PSNR and SSIM values of 33.26 and

0.975 respectively. The results for the campus scenes (Campus-Scene1 to Campus-Scene4) represent slight fluctuations, but the overall performance is also good, especially with Campus-Scene4 achieving an SSIM of 0.946. In the two scenes from the KITTI dataset, the PSNR and SSIM values are relatively stable, and their average values are comparable to those of the GoPro scenes.

Table 3.3 presents the RMSE evaluation results in terms of geometric accuracy. The highway scene (Highway Scene1) represents a low RMSE of 0.76 meters. In the campus scenes, RMSE varies significantly, ranging from 0.40 meters to 2.76 meters. Overall, the average RMSE is 1.30 meters. Due to the lack of corresponding control points, RMSE for the KITTI dataset is not provided, but qualitative analysis can still be conducted through visualization of the mesh results lately.

In summary, the highway scene demonstrates excellent performance in both visual quality and geometric accuracy, while the campus scenes show greater variability in geometric accuracy. In the section 4, the underlying reasons for these values will be further explored.

### Highway mesh result

Fig. 3.2 shows the mesh result produced by streetsurf [15] for the scene DS-Scene1, which is captured in a highway. Fig 3.1 contains a screenshot of the captured video. In the highway scene, the condition of road took the major part of the scene, and it was captured at a crossing. Meanwhile, there were three cars running in the original video, as presented in Fig 2.7, and all of them are removed with our method. Since the input consists of only 70 frames, the final frame is approximately at the position of the zebra crossing facing the car. Due to the Streetsurf [15] model dividing the scene into "close view," "distant view," and "sky view," objects beyond a certain distance are considered infinitely far and are no longer used for texture information calculation.

It can be observed that the ground reconstruction accuracy is very high, and the continuity between frames is well maintained. Although there are some deviations when projecting colors onto the corresponding vertices, the overall reconstruction result is quite good. However, other objects outside the highway tend to appear fragmented. These features are usually isolated and difficult to accurately reconstruct in 3D with a single viewpoint.

Additionally, it is noteworthy that, despite using pre-trained inpainting models to cover areas occluded by dynamic objects, the STTN model does not alter the structural, semantic information in the video and cannot guarantee high-precision inpainting results, leading to some degree of blurring. This causes Streetsurf to fail to comprehend these minor photometric disturbances, resulting in geometric errors in the reconstructed mesh at the locations of the original dynamic objects, which is shown in Fig. 3.3.

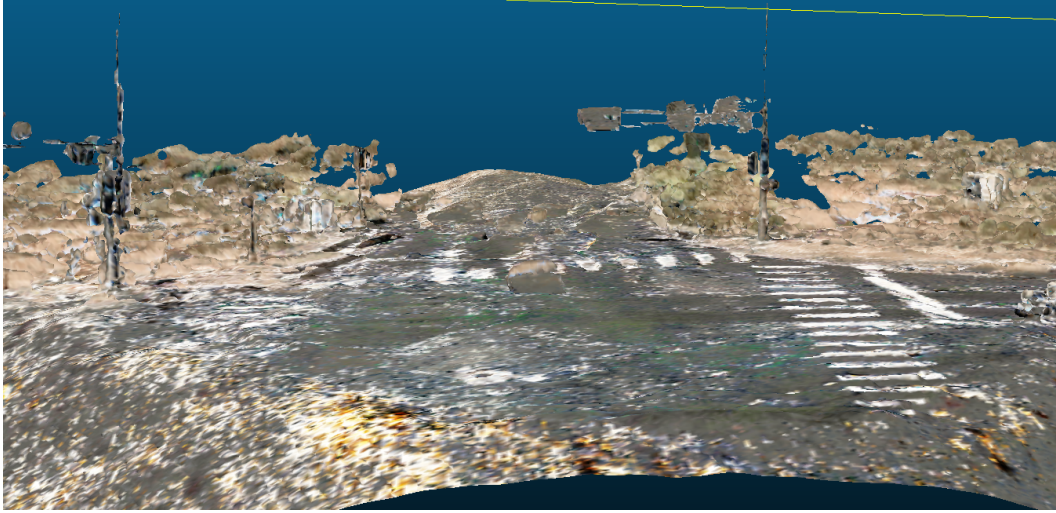


Figure 3.2: Mesh result produced with Streetsurf [15] using monocular camera, **Highway scene**

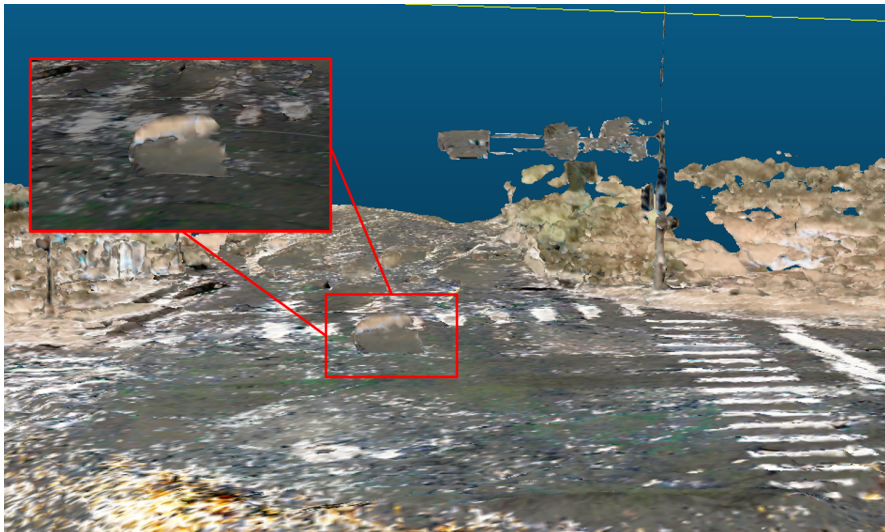


Figure 3.3: Mesh details in **Highway scene**, where boxed area is the geometric error caused by blurred mask from inpainting model

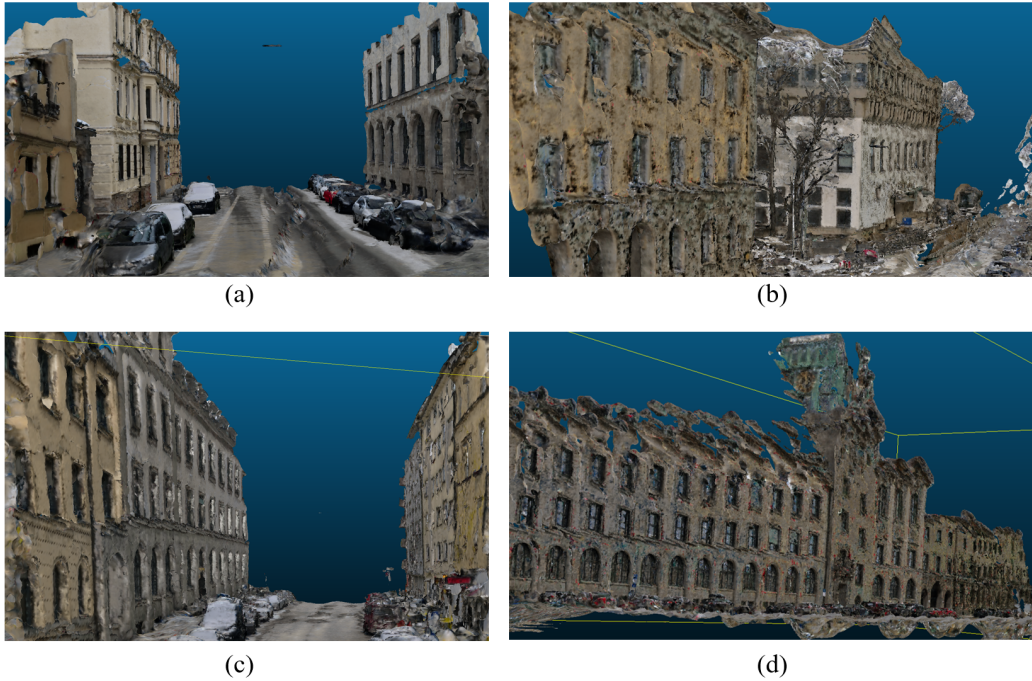


Figure 3.4: Mesh result produced with Streetsurf [15] using a monocular camera, **Campus scene**, in which (a) is captured via handheld mode, guarantee a higher overlapping rate between frames, and (b), (c), (d) are captured in the front of a car. The (b) and (d) scenes were cropped for better perspective.

### Campus mesh result

Four sequences were captured and reconstructed in the campus area, and the mesh result reconstructed with streetsurf [15] is shown in Fig 3.4. In the campus area, the major content of the scenes is antique buildings with complex facade structures, as represented in Fig 3.1. Meanwhile, there are also parked vehicles in the scene, with the ground surface occupying only a small portion. In Scene 2, there are also some difficult-to-replicate trees and many moving vehicles. For Scenes 1, 3, and 4, frames were selected to avoid moving vehicles, while for Scene 2, an inpainting model was employed to remove the moving vehicles, as shown in Fig 3.5. The first sequence was captured by handholding camera, while the others were captured with a camera mounted in the front of a car. In general, the campus scenes include richer and more complicated texture information.

Comparing the reconstruction results of these four scenes, Scene 1 has the best reconstruction mesh appearance, followed by Scene 4, while Scene 3's reconstructed



Figure 3.5: Comparison of images in scene2 before and after inpainting preprocessing

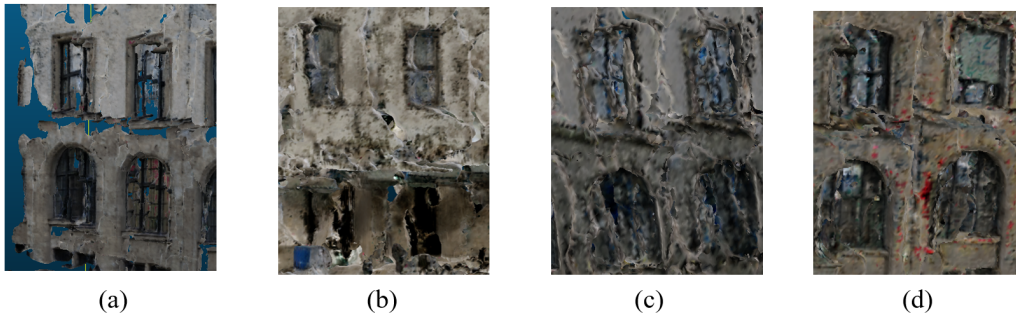


Figure 3.6: Details of mesh result extracted in campus, screenshot respectively from Fig 3.4, showing the discontinuity in facades

mesh was not quite good from the visualization. The reasons for these differences will be discussed in detail in Section 4. As shown in Fig 3.5, Scene 2 faces the same issues as the highway scene. Although pre-trained models are used for dynamic object segmentation, tracking, and inpainting, and the first two achieved good masks, the inpainting model struggles to accurately fill pixels that align with the complex building background. This results in the areas from which dynamic objects have been removed being too blurred, even affecting the generating of depth and normal map, as well as the reconstruction of the building facades. Additionally, for these scenes containing complex building facades, almost every mesh reconstruction result shows discontinuity issues in the exterior walls of the buildings, as illustrated in Fig 3.6. Besides, the mesh results also reveal the impact of a single viewpoint on ray tracing in Streetsurf. From the car’s perspective, the mesh appears continuous. However, when switching viewpoints, the discontinuities become very apparent.





Figure 3.7: Mesh result reconstructed from KITTI dataset with no dynamic removal preprocessing applied in these two sequences. (a), KITTI-residential-0035; (b) KITTI-road-0015.

### KITTI mesh result

Two sequences from the KITTI dataset were conducted with streetsurf. As presented in Fig 3.7, the residential scene is from an urban neighborhood, including simple buildings (without complex facade texture), vegetation, and stationary cars. In contrast, the road scene is from a highway with vegetation on both sides and a car moving almost synchronously with the ego car in front. Therefore, the main difference between the two scenes is that the sequence from the road includes a dynamically moving car. Apart from the scene context, the processing workflows for both sequences are identical. Since the camera model information used for both comes from the well-calibrated GNSS and IMU, the correctness and accuracy can be ensured. As presented in Fig 3.7 and Fig 3.1, in scene road-0015 containing dynamic objects, there are many incorrect-reconstructed vertices, mainly located in the perspective of the front car. In contrast, the mesh appearance of the residential-0035 scene, which includes only static elements, is quite good, both in the road and building parts.

### Highway rendering result

Fig 3.8 shows one screenshot from the rendering result produced by streetsurf [15] with highway scene, including the ground truth image, the rendered image, the rendered depth map as well as the rendered normal map. This result is consistent with the previous quantitative metrics and also corresponds to the previous discussion on the mesh result. On the one hand, Tabel 3.2 indicates that the rendering synthesis of the

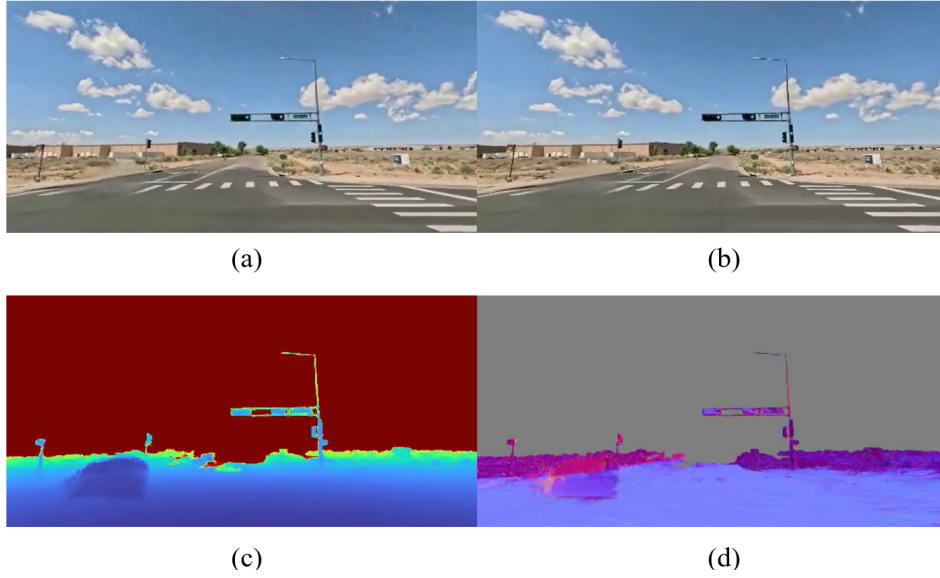


Figure 3.8: Rendering result from **Highway scene**, (a) ground truth image, or original image; (b) rendered image; (c) rendered depth map; (d) rendered normal map

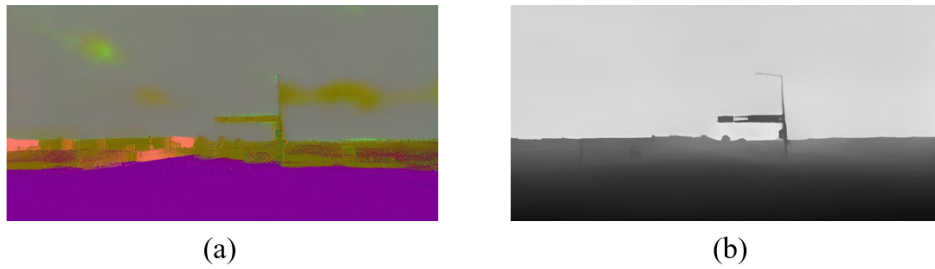


Figure 3.9: Original depth map and normal map generated from Omnidata [11] in **Highway scene**, (a) normal map; (b) depth map

### 3 Experiments

---

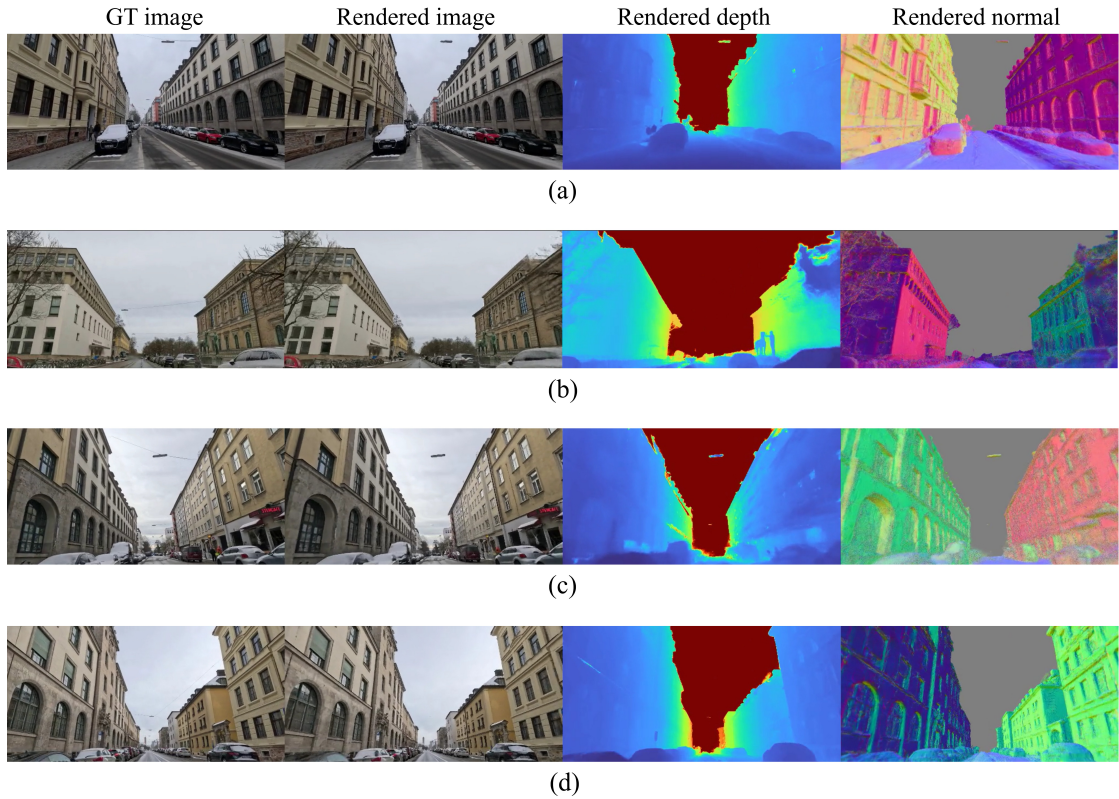


Figure 3.10: Rendering result from **Campus scene**, (a) campus-scene1; (b) campus-scene2; (c) campus-scene3; (d) campus-scene4

highway sequence can be considered "realistic," which is also evident from the visual perspective. On the other hand, the discussion of the mesh results highlighted the limitations of the inpainting model. While as shown in Fig 3.9, the geometric prior maps used as references for the rendered depth and rendered normal are not sensitive to the pixel blurring effects caused by the inpainting model, the rendered depth and rendered normal maps do display the masks of dynamic objects, adversely affecting the final reconstruction outcome.

#### Campus rendering result

Fig 3.10 summarizes all the rendering results in the campus scene, with a screenshot also including the ground truth image, the rendered image, the rendered depth map as well as the rendered normal map for each scene. Overall, the rendering results are consistent with the previous mesh analysis. From the perspective of novel view

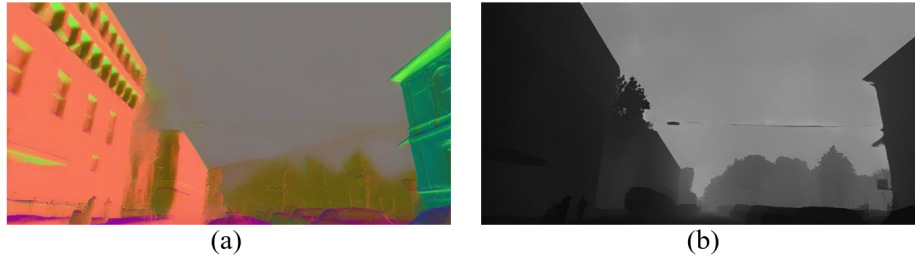


Figure 3.11: Original depth map and normal map generated from Omnidata [11] in **Campus-scene2**, (a) normal map; (b) depth map

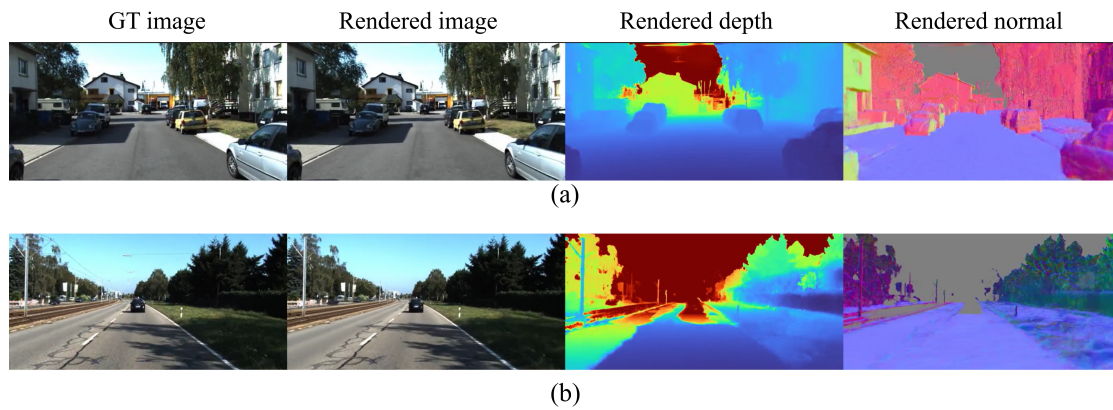


Figure 3.12: Rendering result from **KITTI dataset**, (a) Residential-0035; (b) Road-0015

synthesis, the rendered images for each scene are very similar to the original ground truth images. However, when considering the rendered depth maps and rendered normals, scenes 1 and 4 perform well, while scenes 3 and 2 exhibit noise to some extent. Particularly for scene 2, the large number of dynamic objects and the complex background buildings affect the depth maps and normal maps used as geometric priors after inpainting. As shown in Fig 3.11, in the normal map, pixels originally belonging to the ground are mixed with those of the buildings. Additionally, the depth map shows foreground masks that originally belonged to dynamic objects. Incorrect geometric priors and confused photometric information would result in unsatisfying mesh outcomes.

### KITTI rendering result

Fig 3.12 shows the rendering result from KITTI dataset, the information included in this figure is as that of the GoPro scene. Similar to the GoPro scenes, the KITTI dataset

also performs well in the task of novel view synthesis. However, when it comes to geometry-related depth maps and normal maps, the performance of dynamic scenes differs significantly from that of static scenes. In the dynamic scene road-0015, the normal map represents erroneous gaps on the road at the positions corresponding to the moving vehicle, caused by occlusions during its movement, while in the static scenes, the reconstruction of the road surface is very accurate.

## 4 Discussion

### 4.1 Calibration and reconstruction

In the process of 3D reconstruction, camera calibration plays a very important role. Accurate camera calibration directly affects the precision of the final reconstruction results. Here, camera calibration involves the calibration of both intrinsic and extrinsic parameters. As discussed in section 1.2.1, intrinsic parameters affect the projection relationship from image space to camera space, thereby influencing the scale of the reconstructed scene. Extrinsic parameters affect the projection relationship between the camera space and the world coordinate system, impacting the alignment between frames and the camera's pose in world coordinates. Since the GoPro dataset lacks accurately calibrated camera intrinsic and extrinsic parameters, a quantitative comparison and analysis focusing on the relationship between the 3D reconstruction quality and camera calibration accuracy is performed in this section.

In the experiments involving Streetsurf [15], the experiments from Campus scenes can represent the impact of calibration on the final reconstruction results. Under similar conditions regarding scene characteristics and cameras used, the correctness of pose parameters and camera intrinsics can be observed and compared using the sparse point cloud obtained from SfM. As shown in Fig 4.1, Scene1, Scene2, and Scene4 can all ensure the logical and topological correctness of the sparse point cloud. However, it is evident that buildings in Scene3 exhibit noticeable tilting, indicating some errors in the camera calibration results for Scene3. This also explains the larger RMSE error in Table 3.3 and the unsatisfactory geometric appearance of the mesh. In addition, compared to the campus scenes, under the condition that the scene does not contain dynamic objects, the KITTI dataset with strictly calibrated camera model parameters obtains more accurate mesh results in a static scene, which also supports the impact of the calibration accuracy on the geometry of the reconstruction results.

Besides, in the calculation of aligning pixel-wise extrinsic with the real-world coordinate system, the scale parameters between the pixel-wise camera coordinate system and the real world play an important role. If different scale parameters are considered for each coordinate axis, the reconstructed model will be stretched.

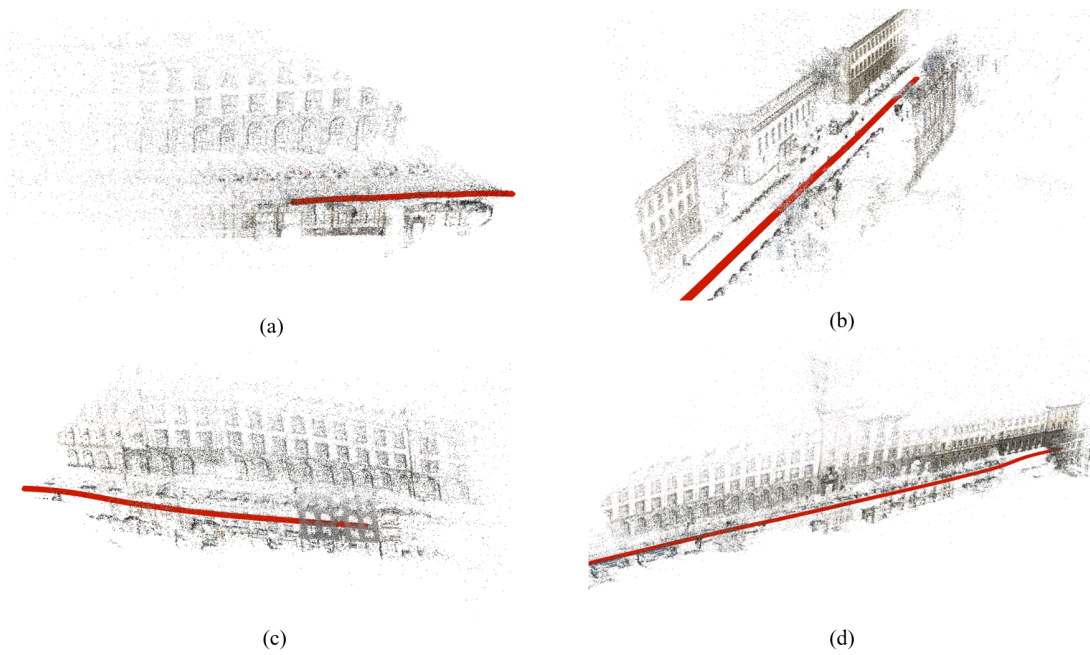


Figure 4.1: Sparse point cloud from SfM in **Campus scene**, (a) campus-scene1; (b) campus-scene2; (c) campus-scene3; (d) campus-scene4

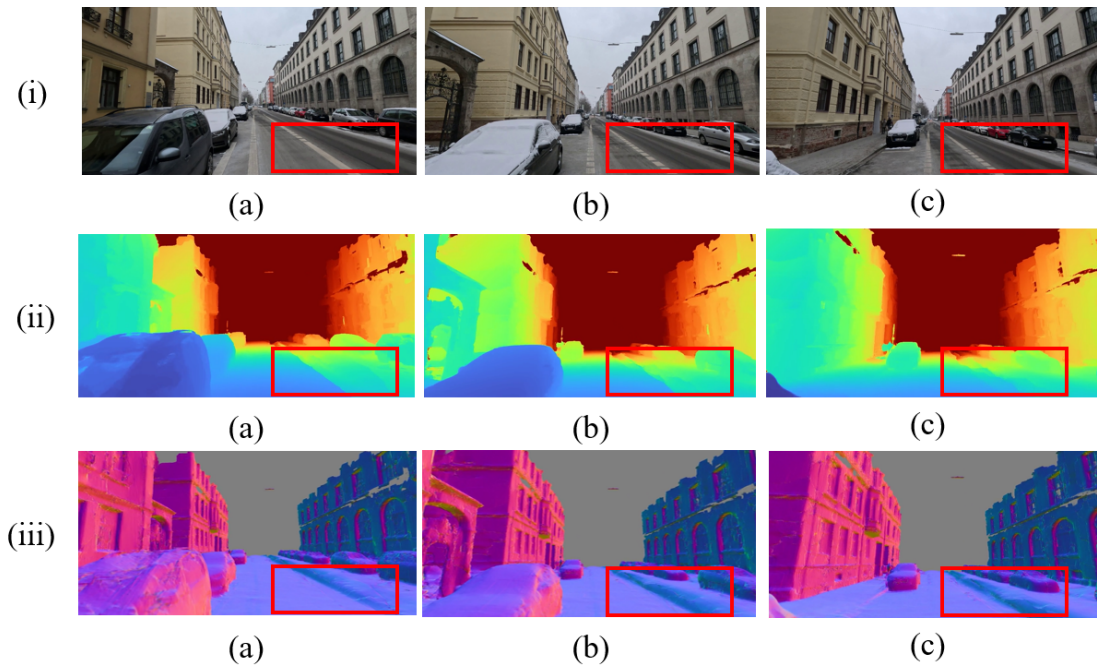


Figure 4.2: screenshot of rendering result from **Campus scene1** , (a) Frame 50; (b) Frame 120; (c) Frame 190; (i) original image; (ii) Rendered depth; (iii) Rendered normal. The boxed area represents the pixels that remain unchanged during forwarding.

## 4.2 Photometric difference and reconstruction

The core of NeRF-based reconstruction algorithms lies in the design of the photometric loss. As shown in Fig 1.3 and Fig 2.12, in the Streetsurf [15] model, which is specifically designed for street-view camera perspectives, the direction of ray tracing aligns with the viewpoint as the car moves forward. This means that the camera perspective is very singular, making it nearly impossible to change angles to observe objects in the scene. However, if the pixel values at the same location in each frame remain unchanged, the model will be unable to generate the correct SDF and color through the MLP. As a result, these pixels will be considered far away from their original geometric texture in the rendered depth map and rendered normal map.

Fig 4.2 is a screenshot with different frames in Campus scene1. As presented in this figure, the pixel values of the tire tracks on the right side remain almost unchanged with the change in perspective. This means that despite the change in camera pose and the direction of ray tracing, the RGB values used as supervision do not change.



Consequently, the MLP will struggle to accurately reconstruct the true SDF information.

### 4.3 Overlapping rate and reconstruction

Overlapping rate refers to the occupancy rate of the overlapping area between frames to the entire image as the camera viewpoint changes. The original NeRF model requires a high overlapping rate. However, due to the moving car's speed, it's usually difficult to provide high-overlap videos. In this work, in addition to video sequences obtained from a moving car, another set of videos was captured using a handheld camera to simulate the viewpoint of a moving car, offering a chance to analyze the impact of overlapping rates on reconstructions.

From the perspective of meshes' appearance, among the campus scenes, Scene 1, which has a higher overlapping rate between frames, produces a more realistic mesh. However, its RMSE accuracy is not the highest. As mentioned before, RMSE is also influenced by camera model parameters. If the distribution of control points affects the scale calculation, it can lead to increasing in RMSE. In terms of rendering synthesis, the PSNR and SSIM metrics for Scene 1 with high overlap and Scene 4 (mounted on a car) with lower overlap are quite similar.

Additionally, as a comparable dataset, the KITTI dataset has a frame rate of 10 fps, lower than the 24 fps of the GoPro dataset, suggesting its overlapping rate should be lower. Nevertheless, the static scene Residential-0035 still achieved good rendering results and mesh appearance.

### 4.4 Geometric cues from MVS and Streetsurf

The geometric cues generated by MVS [40] and Streetsurf [15], including depth maps and normal maps, will be discussed in this chapter. As a traditional 3D reconstruction method, MVS has achieved satisfying performance. The depth map and normal map can usually be considered as a basis of 3D reconstruction results. Utilizing depth maps, 3D coordinate information can be recovered from two-dimensional images, while normal maps can provide rich texture information, aiding in the reconstruction of 3D surfaces.

Fig 4.3 and Fig show the differences between the depth maps and normal maps generated by MVS and the rendered depth maps and rendered normal maps generated by Streetsurf in the campus scenes.

From the representation of depth maps and normal maps generated by MVS, although MVS can obtain relatively accurate 3D point coordinates, the reconstructed scene is significantly affected by noise. Despite the preprocessing for MVS including

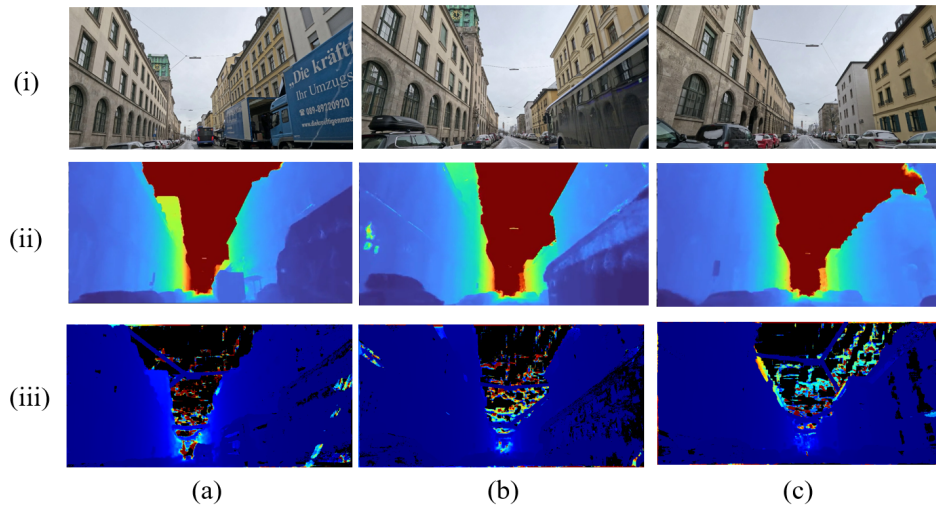


Figure 4.3: Screenshot of depth map from **Campus scene4** , (a) Frame 100; (b) Frame 300; (c) Frame 500. (i) original image; (ii) Rendered depth from Streetsurf; (iii) Normal map from MVS.

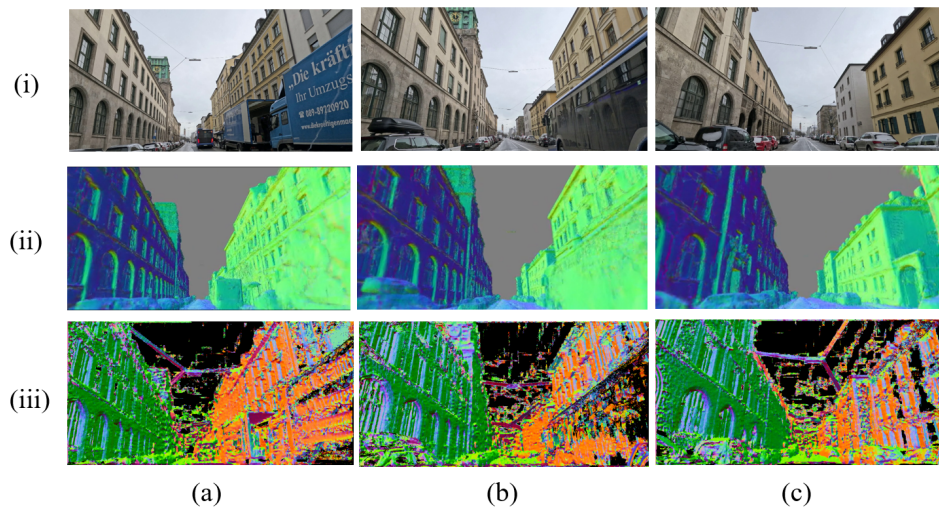


Figure 4.4: Screenshot of normal map from **Campus scene4** , (a) Frame 100; (b) Frame 300; (c) Frame 500. (i) original image; (ii) Rendered normal from Streetsurf; (iii) Normal map from MVS.



Figure 4.5: Screenshot of rendering result from **Campus scene2** , (a) original image; (b) rendered image; (c) rendered depth. (d) rendered normal;

blurring the sky background, there are still many noisy points present in the sky part, and some gaps can be observed in the reconstructed building facades. In contrast, the depth maps and normal maps obtained by Streetsurf are smoother. Although they are limited by viewpoint and contain some areas with incorrect scene representation (such as the bus parked by the roadside), overall, they show fewer topological errors.

## 4.5 Dynamic objects

For the original video sequence without scene editing, dynamic objects are one of the most critical factors affecting the reconstruction geometry. This is related to the characteristics of NeRF-based algorithms. On one hand, both the original version of NeRF [28] and Streetsurf [15] assume that the scene is static, which means that the temporal dimension is not considered in the model’s inputs and outputs. On the other hand, dynamic objects can continuously occlude certain parts of the scene. As discussed in the section on the impact of photometric differences on reconstruction quality, if the pixels in a specific area remain unchanged as the camera viewpoint changes, the geometric texture of that area will be difficult to reconstruct accurately.

The two sets of mesh reconstruction results from the KITTI dataset explained the impact of dynamic objects on reconstruction outcomes. Given a calibrating matrix with nearly the same accuracy and similar buildings in the scenes, when a vehicle moves almost simultaneously with the ego car on the road, it severely occludes the scene information directly in front.

Additionally, although using inpainting models can mitigate the impact of dynamic objects to some extent, this approach may disrupt the original scene’s semantic information, increasing the likelihood of errors in the reconstructed scene, as shown in Fig 4.5.

# 5 Conclusion

## 5.1 Summary

This thesis aims to establish a comprehensive workflow from street view video capture to dense 3D reconstruction, supporting practical applications in autonomous driving. Although traditional 3D reconstruction methods perform well on static and standardized datasets, they frequently encounter various challenges in real street view environments, such as numerous dynamic objects and complex building structures, posing significant difficulties for 3D reconstruction. Our work establishes an efficient and complete process for acquiring 3D depth information from street view data using only a single monocular GoPro camera. With the assistance of a minimal number of control points, a single-shot solution for monocular vision-based 3D reconstruction of street view scenes is proposed in this thesis.

Such a workflow contains mainly four parts. The work begins with data acquisition and preprocessing, followed by preprocessing steps, including semantic segmentation with pre-trained models from Segformer to remove sky backgrounds and video inpainting using SAM and STTN pre-trained models to remove dynamic objects. The second part is camera calibration and pose estimation. The sky-removed images are then processed with COLMAP and DSO to determine the camera model and pose information. After comparing the performance and computational requirements of both methods, COLMAP is chosen for camera calibration. Consequently, camera parameters calculated from SfM are transformed into the world coordinate system using the P3P principle and serve as input for 3D reconstruction. The NeRF-based method Streetsurf is used for reconstruction in the third part. Due to the absence of point cloud data, Streetsurf employs pre-trained models from Omnidata for geometric cue extraction, which act as priors in supervising model training. Finally, the reconstruction results are evaluated using control points from point clouds or precisely calibrated mesh models to assess geometry accuracy.

Overall, the GoPro dataset achieved excellent novel view synthesis results, with an average PSNR of 30.50 and an SSIM of up to 0.93. However, the RMSE, reflecting the accuracy of the reconstructed mesh, varied from 0.40m to 2.52m across different scenes. It performed well in highway scenes but showed significant fluctuations in the campus scenes, with more complex building facades.

Additionally, in this work, both MVS and the KITTI dataset were used as comparisons for an inside analysis of the workflow. From a qualitative perspective, compared to the method used in this study, the depth maps and normal maps generated by MVS showed higher noise levels. Moreover, the KITTI dataset also highlighted the impact of pose accuracy and dynamic objects on reconstruction accuracy with its well-calibrated camera parameters.

## 5.2 Limitations

This study presents a comprehensive workflow for 3D reconstruction from street view video capture to NeRF-based reconstruction and accuracy evaluation, aiming to support practical applications in autonomous driving. However, several limitations still need to be addressed to provide a comprehensive understanding and guide future research.

- Each step of the workflow is independently separated, such as camera calibration and 3D reconstruction being two non-interfering parts. While this design facilitates checking and analyzing each stage in practical operations, some NeRF-based methods [27] have integrated these two parts into a combined training process. This approach not only reduces redundant computation but also may provide better solutions through combined optimization with higher accuracy.
- For dynamic objects in the scene, video inpainting methods still have limitations. Moreover, it may even interrupt the mesh's topological structure or produce errors in semantic information. Addressing the modeling problem of scenes with dynamic objects from such a singular perspective could be one of the key focuses for future work.
- Although only a single monocular camera is used in this scene, the accuracy assessment of the mesh results still relies on a small number of real 3D spatial points to determine the scale.
- The final 3D results are limited by photometric ambiguity caused by the singular viewpoint, leading to discrepancies in both the mesh and the depth and normal maps, such as trenches that do not align with the original scene's semantics.
- Since all GoPro data needs to be collected from a moving vehicle, and considering the cost of collection and the difficulty of obtaining control points, the videos used in this work are limited to only six sequences.

### 5.3 Future work

Based on the mentioned limitations in section 5.2, our future work aims to further enhance the accuracy and applicability of 3D reconstruction from street-view videos. The primary goals are to improve the current methodology, expand the dataset, and integrate additional technologies.

Firstly, it is necessary to expand the experimental video data collection based on the existing dataset since only five sequences of GoPro videos are applied in the work due to a lack of control points. Repeated recording of a scene that includes control point data can be conducted. Additionally, the MLS system could be utilized for extracting control points. This will enhance the accuracy and reliability of the GoPro dataset, leading to more precise results and deeper insights.

Secondly, consider integrating the calibration task with the Streetsurf task, incorporating the accuracy of pose information into the optimization process to achieve higher precision in pose estimation and mesh results. This combination can reduce redundant calculations and simplify the entire workflow, potentially yielding more accurate results. By streamlining the calibration and 3D reconstruction processes, it's possible to improve the efficiency and accuracy of the designed workflow.

Additionally, it is worth considering incorporating pre-trained models that can provide zero-shot solutions into this work, making the reconstructed results that correspond to the real world no longer limited by the availability of control points. A feasible solution is to use zero-shot depth estimation models such as Metric3D [59] to fix the calibration's scale ambiguity.

# Abbreviations

**3DGS** 3D Gaussian Splatting

**BA** Bundle Adjustment

**CNN** Convolutional neural network

**DSO** Direct Sparsity Odometry

**FCNs** Fully convolutional networks

**GAN** Generative Adversarial Network

**GCPs** Ground Control Points

**GNSS** Global Navigation Satellite System

**IMU** Inertial measurement unit

**MLP** Multilayer perceptron

**MLS** Mobile Laser Scanning

**MSE** Mean Squared Error

**MVS** Multi-View Stereo

**NeRF** Neural Radiance Fields

**PSNR** Peak Signal-to-Noise Ratio

**RMSE** Root Mean Square Error

**SAM** Segment Anything Model

**SDF** Signed Distance Field

**SDF** Signed distance function

**SIFT** Scale Invariant Feature Transform

**SLAM** Simultaneous Localization and Mapping

**SSIM** Structural Similarity Index

**STTN** Spatial-Temporal Transformer Network

**SVD** Singular Value Decomposition

**SfM** Structure-from-Motion

**ViT** Vision Transformer



# List of Figures

1.1	Imaging process of pinhole model . . . . .	4
1.2	Transformation in between image coordinate, camera coordinate and world coordinate with intrinsics and extrinsics . . . . .	5
1.3	Difference between street-view scene and typical Nerf scene: (a) street-view scene, (b) typical Nerf scene, adapted from [28] . . . . .	5
1.4	Taxonomy of pose estimation techniques . . . . .	8
1.5	3D representation approaches . . . . .	8
1.6	Principle of NeRF, adapted from [28] . . . . .	11
1.7	Taxonomy of image segmentation models, adapted from [29] . . . . .	12
1.8	Taxonomy of video inpainting models, adapted from [67] . . . . .	14
2.1	Overview of the method presented in this thesis . . . . .	19
2.2	Pipeline of Video capturing and scene editing . . . . .	20
2.3	Matches for original image pair . . . . .	21
2.4	Semantinc segmentation results, (a) SegFormer, (b)Segment Anything .	21
2.5	Matches for segmented image pair . . . . .	22
2.6	Mesh result of Streetsurf with a dynamic object in almost synchronized motion. (a) Original images, (b) Mesh result . . . . .	22
2.7	Before and after processed frames using Inpainting Anything. (a) Original frame, (b)Inpainted frame . . . . .	23
2.8	Pipline of camera calibration and pose extraction . . . . .	23
2.9	Pipeline of pose extraction using COLMAP, adapted from [39] . . . . .	24
2.10	Cameras track and sparse point cloud extracted from (a) COLMAP, (b) DSO . . . . .	25
2.11	Difference in between point cloud extracted from pixel-wise pose and ground truth pose . . . . .	26
2.12	Pipeline of 3D reconstruction using Streetsurf, adapted from [15] . . . . .	28
3.1	Overview of the datasets in this experiments . . . . .	32
3.2	Mesh result produced with Streetsurf [15] using monocular camera, <b>Highway scene</b> . . . . .	39
3.3	Mesh details in <b>Highway scene</b> , where boxed area is the geometric error caused by blurred mask from inpainting model . . . . .	39

3.4	Mesh result produced with Streetsurf [15] using a monocular camera, <b>Campus scene</b> , in which (a) is captured via handheld mode, guarantee a higher overlapping rate between frames, and (b), (c), (d) are captured in the front of a car. The (b) and (d) scenes were cropped for better perspective. . . . .	40
3.5	Comparison of images in scene2 before and after inpainting preprocessing	41
3.6	Details of mesh result extracted in campus, screenshot respectively from Fig 3.4, showing the discontinuity in facades . . . . .	41
3.7	Mesh result reconstructed from KITTI dataset with no dynamic removal preprocessing applied in these two sequences. (a), KITTI-residential-0035; (b) KITTI-road-0015. . . . .	42
3.8	Rendering result from <b>Highway scene</b> , (a) ground truth image, or original image; (b) rendered image; (c) rendered depth map; (d) rendered normal map . . . . .	43
3.9	Original depth map and normal map generated from Omnidata [11] in <b>Highway scene</b> , (a) normal map; (b) depth map . . . . .	43
3.10	Rendering result from <b>Campus scene</b> , (a) campus-scene1; (b) campus-scene2; (c) campus-scene3; (d) campus-scene4 . . . . .	44
3.11	Original depth map and normal map generated from Omnidata [11] in <b>Campus-scene2</b> , (a) normal map; (b) depth map . . . . .	45
3.12	Rendering result from <b>KITTI dataset</b> , (a) Residential-0035; (b) Road-0015	45
4.1	Sparse point cloud from SfM in <b>Campus scene</b> , (a) campus-scene1; (b) campus-scene2; (c) campus-scene3; (d) campus-scene4 . . . . .	48
4.2	screenshot of rendering result from <b>Campus scene1</b> , (a) Frame 50; (b) Frame 120; (c) Frame 190; (i) original image; (ii) Rendered depth; (iii) Rendered normal. The boxed area represents the pixels that remain unchanged during forwarding. . . . .	49
4.3	Screenshot of depth map from <b>Campus scene4</b> , (a) Frame 100; (b) Frame 300; (c) Frame 500. (i) original image; (ii) Rendered depth from Streetsurf; (iii) Normal map from MVS. . . . .	51
4.4	Screenshot of normal map from <b>Campus scene4</b> , (a) Frame 100; (b) Frame 300; (c) Frame 500. (i) original image; (ii) Rendered normal from Streetsurf; (iii) Normal map from MVS. . . . .	51
4.5	Screenshot of rendering result from <b>Campus scene2</b> , (a) original image; (b) rendered image; (c) rendered depth. (d) rendered normal; . . . . .	52

## List of Tables

2.1	Advantage and disadvantage of COLMAP and DSO . . . . .	25
3.1	Description for dataset . . . . .	33
3.2	Quantitative evaluation of the rendering . . . . .	37
3.3	Quantitative evaluation of geometry . . . . .	37

# Bibliography

- [1] de Agapito L, Hayman E, Reid I (1998) Self-calibration of a rotating camera with varying intrinsic parameters. In: BMVC, pp 1–10
- [2] Bradski G (2000) The OpenCV Library. Dr. Dobb's Journal of Software Tools
- [3] Campos C, Elvira R, Rodríguez JJG, Montiel JM, Tardós JD (2021) Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Transactions on Robotics* 37(6):1874–1890
- [4] Cao C, Dong Q, Fu Y (2022) Learning prior feature and attention enhanced image inpainting. In: European conference on computer vision, Springer, pp 306–322
- [5] Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 801–818
- [6] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
- [7] Dalal A, Hagen D, Robbersmyr KG, Knausgård KM (2024) Gaussian splatting: 3d reconstruction and novel view synthesis, a review. 2405.03417
- [8] Davison AJ, Reid ID, Molton ND, Stasse O (2007) Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence* 29(6):1052–1067
- [9] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09
- [10] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: Transformers for image recognition at scale. 2010.11929

- [11] Eftekhari A, Sax A, Malik J, Zamir A (2021) Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10,786–10,796
- [12] Engel J, Koltun V, Cremers D (2017) Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence* 40(3):611–625
- [13] Gao C, Saraf A, Huang JB, Kopf J (2020) Flow-edge guided video completion. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, Springer, pp 713–729
- [14] Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*
- [15] Guo J, Deng N, Li X, Bai Y, Shi B, Wang C, Ding C, Wang D, Li Y (2023) Streetsurf: Extending multi-view implicit surface reconstruction to street views. arXiv preprint arXiv:2306.04988
- [16] Hartley R, Zisserman A (2003) *Multiple view geometry in computer vision*. Cambridge university press
- [17] Iizuka S, Simo-Serra E, Ishikawa H (2017) Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36(4):1–14
- [18] Jain A, Tancik M, Abbeel P (2021) Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp 5885–5894
- [19] Jeong Y, Ahn S, Choy C, Anandkumar A, Cho M, Park J (2021) Self-calibrating neural radiance fields. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp 5846–5854
- [20] Kerbl B, Kopanas G, Leimkühler T, Drettakis G (2023) 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42(4), URL <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- [21] Kerbl B, Kopanas G, Leimkühler T, Drettakis G (2023) 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42(4):1–14
- [22] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, Dollár P, Girshick R (2023) Segment anything. arXiv:2304.02643

- [23] Lee S, Oh SW, Won D, Kim SJ (2019) Copy-and-paste networks for deep video inpainting. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 4413–4421
- [24] Li Z, Li L, Zhu J (2023) Read: Large-scale neural scene rendering for autonomous driving. In: Proceedings of the AAAI Conference on Artificial Intelligence, 37(2):1522–1529
- [25] Li Z, Müller T, Evans A, Taylor RH, Unberath M, Liu MY, Lin CH (2023) Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 8456–8465
- [26] Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
- [27] Meuleman A, Liu YL, Gao C, Huang JB, Kim C, Kim MH, Kopf J (2023) Progressively optimized local radiance fields for robust view synthesis. In: CVPR
- [28] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R (2021) Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 65(1):99–106
- [29] Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N, Terzopoulos D (2021) Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence 44(7):3523–3542
- [30] Müller N, Siddiqui Y, Porzi L, Bulo SR, Kotschieder P, Nießner M (2023) Diffrf: Rendering-guided 3d radiance field diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4328–4338
- [31] Ost J, Mannan F, Thuerey N, Knodt J, Heide F (2021) Neural scene graphs for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 2856–2865
- [32] Qi W, Li F, Zhenzhong L (2010) Review on camera calibration. In: 2010 Chinese Control and Decision Conference, pp 3354–3358, DOI 10.1109/CCDC.2010.5498574
- [33] Quan W, Chen J, Liu Y, Yan DM, Wonka P (2024) Deep learning-based image and video inpainting: A survey. International Journal of Computer Vision pp 1–34
- [34] Ranftl R, Bochkovskiy A, Koltun V (2021) Vision transformers for dense prediction. 2103.13413

- [35] Rematas K, Liu A, Srinivasan PP, Barron JT, Tagliasacchi A, Funkhouser T, Ferrari V (2022) Urban radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 12,932–12,942
- [36] Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. 1505.04597
- [37] Samavati T, Soryani M (2023) Deep learning-based 3d reconstruction: a survey. *Artificial Intelligence Review* 56(9):9175–9219
- [38] Saputra MRU, Markham A, Trigoni N (2018) Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)* 51(2):1–36
- [39] Schönberger JL, Frahm JM (2016) Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR)
- [40] Schönberger JL, Zheng E, Pollefeys M, Frahm JM (2016) Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV)
- [41] Sun P, Kretschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, Vasudevan V, Han W, Ngiam J, Zhao H, Timofeev A, Ettinger S, Krivokon M, Gao A, Joshi A, Zhang Y, Shlens J, Chen Z, Anguelov D (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
- [42] Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, Kong N, Goka H, Park K, Lempitsky V (2021) Resolution-robust large mask inpainting with fourier convolutions. arXiv preprint arXiv:2109.07161
- [43] Taketomi T, Uchiyama H, Ikeda S (2017) Visual slam algorithms: A survey from 2010 to 2016. *IPSJ transactions on computer vision and applications* 9:1–11
- [44] Tang L, Ruiz N, Qinghao C, Li Y, Holynski A, Jacobs DE, Hariharan B, Pritch Y, Wadhwa N, Aberman K, Rubinstein M (2023) Realfill: Reference-driven generation for authentic image completion. arXiv preprint arXiv:2309.16668
- [45] Teed Z, Deng J (2021) DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*
- [46] Turki H, Zhang JY, Ferroni F, Ramanan D (2023) Suds: Scalable urban dynamic scenes. In: *Computer Vision and Pattern Recognition (CVPR)*

- [47] Wang J, Wang P, Long X, Theobalt C, Komura T, Liu L, Wang W (2022) Neuris: Neural reconstruction of indoor scenes using normal priors. In: European Conference on Computer Vision, Springer, pp 139–155
- [48] Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang YG (2018) Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European conference on computer vision (ECCV), pp 52–67
- [49] Wang P, Liu L, Liu Y, Theobalt C, Komura T, Wang W (2023) Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. 2106.10689
- [50] Wang W, Xie E, Li X, Fan DP, Song K, Liang D, Lu T, Luo P, Shao L (2021) Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 568–578
- [51] Warburg F, Weber E, Tancik M, Holynski A, Kanazawa A (2023) Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 18,120–18,130
- [52] Weder S, Garcia-Hernando G, Monzpart A, Pollefeys M, Brostow GJ, Firman M, Vicente S (2023) Removing objects from neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 16,528–16,538
- [53] Wu C, et al (2011) Visualsfm: A visual structure from motion system
- [54] Wysocki O, Zhang J, Stilla U (2021) Tum-faÇade. DOI 10.14459/2021mp1636761.001, URL <https://mediatum.ub.tum.de/1636761>
- [55] Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P (2021) Segformer: Simple and efficient design for semantic segmentation with transformers. In: Neural Information Processing Systems (NeurIPS)
- [56] Xie Z, Zhang J, Li W, Zhang F, Zhang L (2023) S-nerf: Neural radiance fields for street views. In: ICLR 2023
- [57] Xu R, Li X, Zhou B, Loy CC (2019) Deep flow-guided video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3723–3732
- [58] Yariv L, Kasten Y, Moran D, Galun M, Atzmon M, Basri R, Lipman Y (2020) Multiview neural surface reconstruction by disentangling geometry and appearance. 2003.09852



- [59] Yin W, Zhang C, Chen H, Cai Z, Yu G, Wang K, Chen X, Shen C (2023) Metric3d: Towards zero-shot metric 3d prediction from a single image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 9043–9053
- [60] Yousif K, Bab-Hadiashar A, Hoseinnezhad R (2015) An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems* 1(4):289–311
- [61] Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Free-form image inpainting with gated convolution. arXiv preprint arXiv:1806.03589
- [62] Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2019) Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)
- [63] Yu T, Feng R, Feng R, Liu J, Jin X, Zeng W, Chen Z (2023) Inpaint anything: Segment anything meets image inpainting. arXiv preprint arXiv:2304.06790
- [64] Yu Z, Peng S, Niemeyer M, Sattler T, Geiger A (2022) Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. 2206.00665
- [65] Zeng Y, Fu J, Chao H (2020) Learning joint spatial-temporal transformations for video inpainting. In: The Proceedings of the European Conference on Computer Vision (ECCV)
- [66] Zhang K, Riegler G, Snavely N, Koltun V (2020) Nerf++: Analyzing and improving neural radiance fields. 2010.07492
- [67] Zhang X, Zhai D, Li T, Zhou Y, Lin Y (2023) Image inpainting based on deep learning: A review. *Information Fusion* 90:74–94, DOI <https://doi.org/10.1016/j.inffus.2022.08.033>, URL <https://www.sciencedirect.com/science/article/pii/S1566253522001324>
- [68] Zhang Z (2000) A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence* 22(11):1330–1334
- [69] Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, Fu Y, Feng J, Xiang T, Torr PH, Zhang L (2021) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 6881–6890
- [70] Zhu Z, Peng S, Larsson V, Cui Z, Oswald MR, Geiger A, Pollefeys M (2024) Nicerslam: Neural implicit scene encoding for rgb slam. In: International Conference on 3D Vision (3DV)

- [71] Zou X, Yang L, Liu D, Lee YJ (2021) Progressive temporal feature alignment network for video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 16,448–16,457