UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GIOVANNA BUENO MARCONDES

THREE DIMENSIONAL RECONSTRUCTION METHODS OF VEHICLE DRIVING SCENARIOS FOR DRIVING ASSISTANCE

PONTA GROSSA

2025

GIOVANNA BUENO MARCONDES

THREE DIMENSIONAL RECONSTRUCTION METHODS OF VEHICLE DRIVING SCENARIOS FOR DRIVING ASSISTANCE

Métodos de Reconstrução Tridimensional de Cenários de Condução Veicular para Assistência ao Condutor

Thesis presented as a requirement to obtain the title of Master in Electrical Engineering at the Universidade Tecnológica Federal do Paraná (UTFPR).

Advisor: Prof. Dr. Max Mauro Dias Santos

PONTA GROSSA
2025

Ministerio da Educa ao
Universidade Tecnologica Federal do Parana
Campus Ponta Grossa

GIOVANNA BUENO MARCONDES

THREE DIMENSIONAL RECONSTRUCTION METHODS OF VEHICLE DRIVING SCENARIOS FOR DRIVING ASSISTANCE

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Controle E Processamento De Energia.

Data de aprovação: 13 de Março de 2025

Dra. Cristhiane Goncalves, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Glaucia Melo Dos Santos, Doutorado - Toronto Metropolitan University

Dr. Leopoldo Rideki Yoshioka, Doutorado - Escola Politécnica da Universidade de São Paulo

Dra. Marcella Scoczynski Ribeiro Martins, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Academico da UTFPR a partir dos dados da Ata de Defesa em 21/08/2025.

I dedicate this work to my husband and
professor, for all the support they offered.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my husband for his unwavering support and encouragement throughout this journey. His patience, understanding, and love have been my constant source of strength.

I am also profoundly thankful to my professor, Dr. Santos, for his invaluable guidance, insightful feedback, and continuous mentorship. His expertise and dedication have greatly enriched my research and academic growth.

ABSTRACT

Three-dimensional (3D) reconstruction plays a crucial role in the development of realistic driving scenarios for autonomous vehicle simulations. This work investigates and evaluates various methodologies for 3D reconstruction, aiming to enhance the fidelity and applicability of virtual environments in driving simulators. A comprehensive analysis of different reconstruction techniques was conducted, considering point cloud generation, mesh reconstruction, and post-processing methods. The integration of reconstructed environments into simulation platforms, particularly CARLA, was explored. The study also presents an in-depth investigation into the challenges of integrating Unreal Engine maps with CARLA, identifying the need for multiple files for successful manual integration. Practical tests and evaluations were carried out to assess the accuracy and computational performance of selected methods through evaluations within Cloud Compare. Furthermore, different driving simulators were analyzed to determine their suitability for integrating reconstructed environments. CARLA was selected as the primary simulation tool due to its open-source nature and strong support for autonomous vehicle research. Other simulators, such as SUMO and LGSVL, were also considered but presented limitations regarding 3D scene reconstruction capabilities. The selection process involved evaluating factors like realism, ease of integration, and extensibility, ensuring an optimal environment for autonomous vehicle testing. To support the selection of methodologies, two systematic reviews were conducted following the PRISMA guidelines, utilizing NVivo for qualitative data analysis. This process allowed for the identification and categorization of relevant reconstruction techniques, simulation tools, and integration strategies. The review ensured a structured and comprehensive understanding of the current state of the art, facilitating an informed decision-making process throughout the research. This research contributes by advancing the understanding of 3D reconstruction techniques for driving simulations and providing a structured methodology for integrating virtual environments into autonomous vehicle testing frameworks. Future work includes further optimization of the integration process and the development of automated tools to streamline environment reconstruction for simulation purposes.

Keywords: computer graphics; visualization; 3d reconstruction; mesh texturing; virtual scenarios.

RESUMO

A reconstrução tridimensional (3D) desempenha um papel crucial no desenvolvimento de cenários realistas para simulações de veículos autônomos. Este trabalho investiga e avalia diversas metodologias de reconstrução 3D, visando aprimorar a fidelidade e aplicabilidade dos ambientes virtuais em simuladores de direção. Foi realizada uma análise abrangente de diferentes técnicas de reconstrução, considerando geração de nuvens de pontos, reconstrução de malhas e métodos de pós-processamento. Além disso, explorou-se a integração dos ambientes reconstruídos em plataformas de simulação, com foco no CARLA. O estudo também apresenta uma investigação detalhada sobre os desafios da integração de mapas do Unreal Engine com o CARLA, identificando a necessidade de múltiplos arquivos para uma integração manual bem-sucedida. Testes práticos e avaliações foram conduzidos para analisar a precisão e o desempenho computacional dos métodos selecionados por meio de avaliações no Cloud Compare. Além disso, diferentes simuladores de direção foram analisados para determinar sua adequação à integração de ambientes reconstruídos. O CARLA foi selecionado como a principal ferramenta de simulação devido à sua natureza de código aberto e forte suporte à pesquisa em veículos autônomos. Outros simuladores, como SUMO e LGSVL, também foram considerados, mas apresentaram limitações na reconstrução de cenas 3D. O processo de seleção envolveu a avaliação de fatores como realismo, facilidade de integração e extensibilidade, garantindo um ambiente ideal para testes de veículos autônomos. Para apoiar a seleção das metodologias, foram conduzidas duas revisões sistemáticas seguindo as diretrizes do PRISMA, utilizando o NVivo para análise qualitativa dos dados. Esse processo permitiu a identificação e categorização das técnicas de reconstrução relevantes, ferramentas de simulação e estratégias de integração. A revisão garantiu uma compreensão estruturada e abrangente do estado da arte, facilitando um processo de tomada de decisão fundamentado ao longo da pesquisa. Este estudo contribui para o avanço do entendimento das técnicas de reconstrução 3D aplicadas a simulações de direção e propõe uma metodologia estruturada para a integração de ambientes virtuais em frameworks de teste de veículos autônomos. Trabalhos futuros incluem a otimização do processo de integração e o desenvolvimento de ferramentas automatizadas para agilizar a reconstrução de ambientes para fins de simulação.

Palavras-chave: computação gráfica; visualização; reconstrução 3d; texturização de malhas; cenários virtuais.

LIST OF ABBREVIATIONS AND ACRONYMS

Pseudo-Acronyms

3DRIMR	3D Reconstruction and Imaging via Millimeter-Wave Radar
3DS MAX	Software de modelagem tridimensional
ABNT	Associação Brasileira de Normas Técnicas
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AI	Artificial Intelligence
APOLLO	Autonomous Vehicles Simulation Platafform
AR	Augmented Reality
AVSimulation	Simulador avançado de veículos
Altizure	Platform for 3D Reconstruction
BA	Bundle Adjustment
BPA	Ball-Pivoting Algorithm
CARLA	Simulador para pesquisa em direção autônoma
CASMVSNet	Rede Neural para Multi-View Stereo
CMP-MVS	CMP Multi-View Stereo
CNN	Convolutional Neural Network
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
Cgal	Computational Geometry Algorithms Library
CityGML	City Geography Markup Language
DEM	Digital Elevation Model
DSM	Digital Surface Model
DiM150	Dynamic Motion Simulator with 150 degrees of freedom
GAN	Generative Adversarial Network

GTA5	Grand Theft Auto V (simulação)
HIL	Hardware-In-the-Loop
IMU	Inertial Measurement Unit
LGSVL	LG Smart Vehicle Simulator
LAS	Log ASCII Standard
LiDAR	Light Detection and Ranging
MATLAB	Matrix Laboratory
MMW	Millimeter-Wave
MVE	Multi-View Environment
MVSM	Multi-View Stereo Matching
MVS	Multi-View Stereo
NASA	National Aeronautics and Space Administration
NGP	Neural Graphics Primitives
NeRF	Neural Radiance Fields
OGC	Open Geospatial Consortium
ORB	Oriented FAST and Rotated BRIEF
PCL	Point Cloud Library
PMVS/CMVS	Patch-based Multi-View Stereo / Clustering for Multi-View Stereo
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RTK	Real-Time Kinematic
RTOS	Real-Time Operating System
SAR	Synthetic Aperture Radar
SDV	Software-Defined Vehicles
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
STD	Standard deviation

SUMO	Simulation of Urban Mobility
SURF	Speeded-Up Robust Features
TLS	Transport Layer Security
TORCS	The Open Racing Car Simulator
UTM	Universal Transverse Mercator
UAV	Unmanned Aerial Vehicle
VAE	Variational Autoencoder
VB3D	Visual-based 3D
ResNet	Residual Network
RNN	Recurrent Neural Network

LIST OF BOARDS

LIST OF FIGURES

LIST OF TABLES

CONTENTS

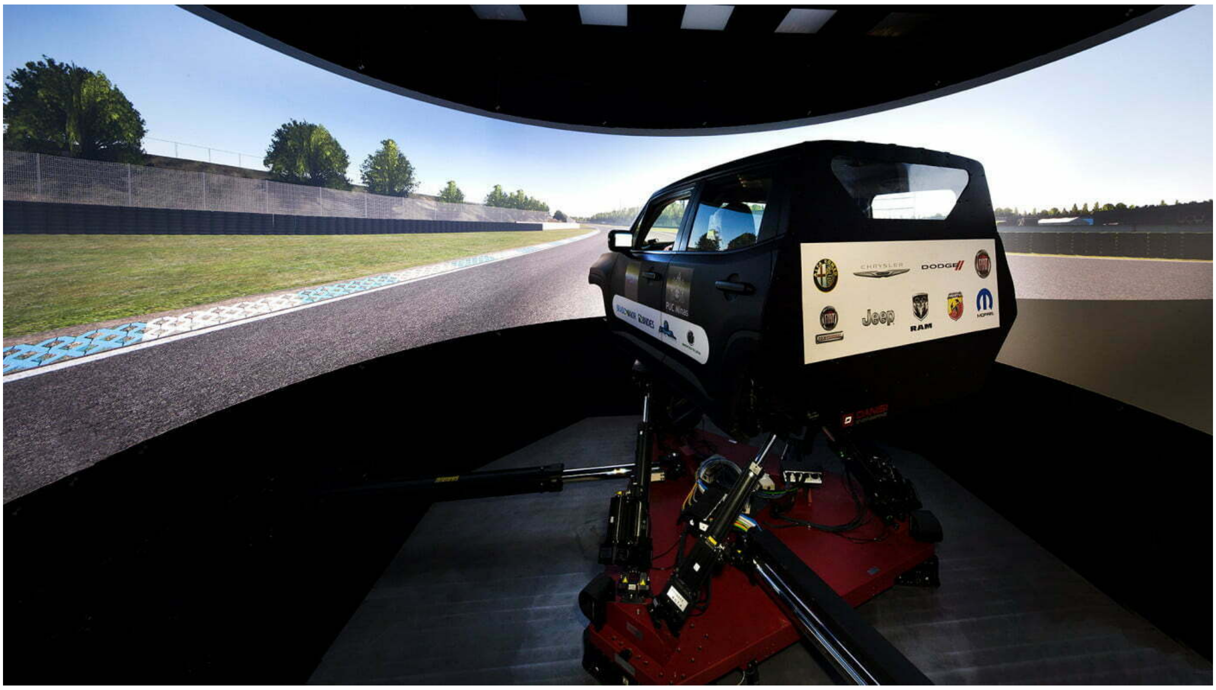
1 INTRODUCTION

Three-dimensional (3D) reconstruction is the process of capturing the shape and appearance of real objects or environments and converting them into a digital 3D model. This technique involves using data from various sensors, such as cameras, Light Detection and Ranging (LiDAR), or structured light, to generate a detailed representation of physical spaces in the digital realm. 3D reconstruction is widely used in fields like virtual reality, autonomous driving, architecture, cultural heritage preservation, and entertainment, as it allows for realistic and immersive interactions with virtual replicas of real-world entities. The benefits of 3D reconstruction include enhanced spatial analysis, as it provides accurate spatial data that aids in decision-making and planning. For autonomous vehicles, for example, it enables the creation of highly realistic simulations of driving environments, where algorithms and ADAS functions can be tested safely. 3D reconstruction can improve visualization and accessibility, allowing users to explore and interact with complex environments remotely, which is invaluable for research, training, and design across numerous fields. The integration of 3D point cloud generation and mesh texturing into vehicle simulation environments has opened new pathways in autonomous vehicle testing and algorithm validation (Tancik *et al.*, 2023).

Vehicle simulators are advanced virtual platforms that replicate real-world driving environments and vehicle behavior, providing a safe and controlled space for testing and development. Used extensively in the automotive industry, these simulators recreate various driving scenarios, weather conditions, and traffic situations, allowing engineers to assess vehicle performance, safety, and reliability without the risks and costs associated with physical testing. In the development of Advanced Driver Assistance Systems (ADAS), vehicle simulators play a crucial role by offering a highly flexible environment where systems like adaptive cruise control, lane-keeping assistance, and collision avoidance can be rigorously tested and fine-tuned. By using simulators, ADAS developers can expose algorithms to challenging and rare events that might be difficult to replicate in real-world tests, thereby improving their robustness and safety. Additionally, simulators provide detailed data and analytics, enabling engineers to monitor and optimize ADAS behavior across a wide range of scenarios before deployment, accelerating development cycles and contributing to safer, more reliable autonomous and semi-autonomous vehicles (CARLA Simulator Team, n.d.).

Static and dynamic simulators are two primary types of vehicle simulators used to test and validate driving systems under different conditions. Static simulators focus on replicating driving scenarios and vehicle behavior without physical movement; they offer a controlled environment where visual and auditory feedback allows engineers to assess driver reactions, ADAS performance, and system reliability in a virtual setup. Dynamic simulators, on the other hand, include motion platforms that physically move to mimic the forces experienced during real driving, such as acceleration, braking, and cornering. This added realism helps simulate real-world driving dynamics more accurately, making dynamic simulators valuable for studying the effects of vehicle handling and driver responses under complex scenarios. An example of an advanced dynamic simulator is the DiM150, located at PUC Minas, Brazil. Figure 1 shows the simulator. The DiM150 is a state-of-the-art motion-based simulator with six degrees of freedom, enabling highly realistic movement to replicate real driving sensations. Equipped with cutting-edge visual and auditory systems, the DiM150 can simulate various driving conditions, providing a comprehensive platform for ADAS and autonomous vehicle development. This simulator's capabilities are crucial for conducting safe, repeatable, and controlled testing, facilitating research and innovation in vehicle dynamics, safety systems, and driver assistance technologies (Rodrigues *et al.*, 2021a).

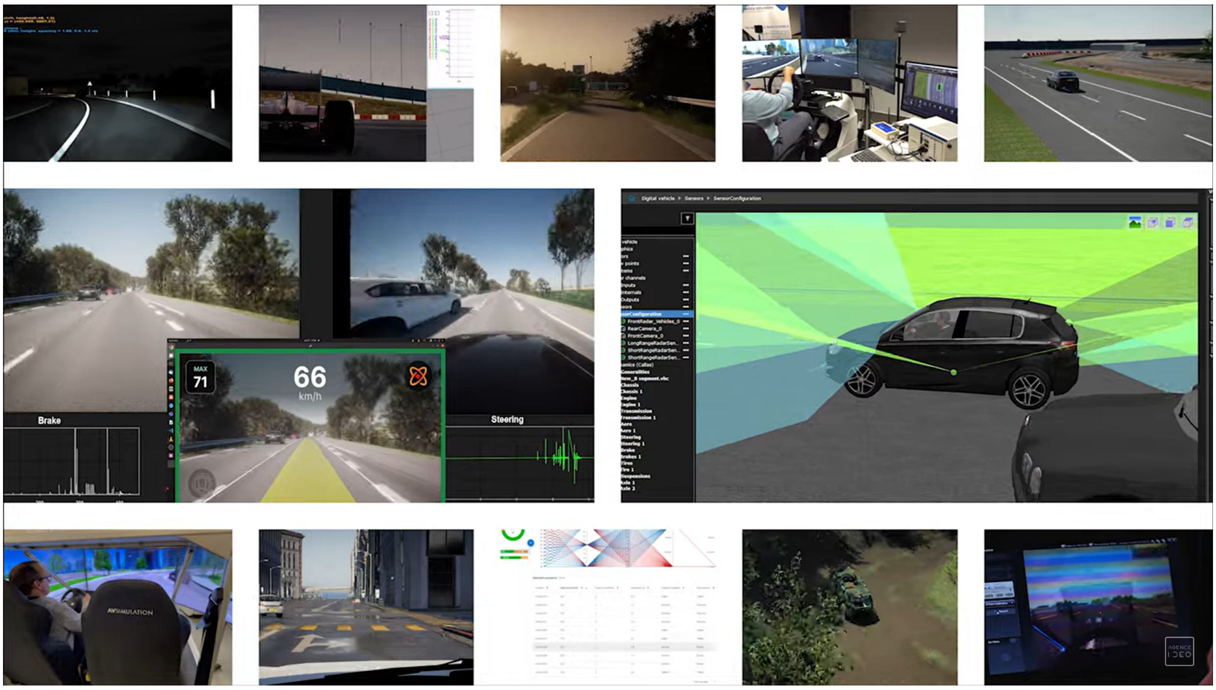
Figure 1 – Dim150 Simulator located at PUC Minas SIMCenter.



Source: Alexandre (2017).

Currently, the dynamic simulator DiM150 at PUC Minas uses SCANeR, an advanced simulation platform developed by AVSimulation (AVSimulation, 2024). SCANeR is shown in figure 2. It provides a comprehensive and highly detailed testing environment, covering a wide range of driving scenarios and simulation parameters, making it ideal for the development and validation of ADAS and autonomous vehicle technologies. However, the software presents some limitations in meeting the specific needs of the Brazilian context. The primary restriction is the lack of detailed modeling of Brazilian roads, which reduces the fidelity of testing for local conditions and limits the applicability of results to the national landscape. Additionally, SCANeR is a paid software, which imposes significant costs for access and maintenance, potentially restricting accessibility and flexibility for researchers. These factors highlight the importance of developing or integrating alternative platforms that allow for greater customization and the inclusion of Brazil-specific data, enabling simulations that are more representative of the national context.

Figure 2 – SCANNeR.



Source: AVSimulation (2024).

ADAS are sophisticated technologies designed to improve vehicle safety and enhance the driving experience by assisting drivers with key tasks and reducing human error. These systems encompass a range of functionalities, including lane departure warnings, adaptive cruise control, blind-spot detection, and emergency braking, all of which aim to prevent accidents and provide safer, more convenient driving. ADAS employs a combination of sensors, such as cameras, radar, LiDAR, and ultrasonic devices, to monitor the vehicle's surroundings and interpret real-time data, enabling quick and accurate responses to changing road conditions and potential hazards. In recent years, ADAS has become a foundational component in the development of autonomous vehicles, as these systems form the building blocks for higher levels of vehicle autonomy. By reducing the cognitive and physical demands on drivers, ADAS contributes significantly to reducing the number and severity of accidents, thereby improving overall road safety and paving the way for a future of fully autonomous transportation.

These techniques serve as a foundational element in various applications, including virtual reality, scene reconstruction, and high-precision mapping (Muñoz-silva *et al.*, 2021; Fruh; Zakhor, 2001; Tancik *et al.*, 2023; Müller *et al.*, 2022). In the realm of autonomous driving, they enable the creation of detailed, realistic virtual models that mirror complex roadway environments. This dissertation focuses on developing a real-time simulation platform to design and validate algorithms for autonomous and automated vehicles. The project also emphasizes generating and integrating 3D models from Brazilian scenarios to increase the simulation's fidelity and real-world applicability.

1.1 Relevance of the Research Topic

The rapid advancements in autonomous vehicle technology necessitate reliable and accurate testing environments to evaluate and validate driving algorithms before deployment in real-world scenarios. Autonomous systems require constant refinement through testing in diverse, often unpredictable, environments, which can be costly, time-consuming, and limited by safety constraints. Simulation platforms present a safer, scalable, and cost-effective alternative, allowing developers to test and enhance the performance of vehicle control systems, sensor integration, and decision-making algorithms in a controlled yet realistic setting (Slavcheva; Baust; Ilic, 2018; Ibrahim; Nagy; Benedek, 2022; Buyukdemircioglu; Kocaman, 2020).

The development of a simulation framework with high-fidelity 3D reconstructions of local scenarios infrastructure is particularly relevant within the Brazilian context, where unique characteristics and infrastructure differences can influence autonomous vehicle performance. By incorporating 3D reconstructions of Brazilian scenarios into the simulation environment, this project aims to enhance simulation realism, thereby providing more accurate representations for validation and testing of ADAS and autonomous vehicle controls.

1.2 Challenges

Developing a integration that incorporates 3D reconstructed environments into a simulation platform presents several technical and operational challenges:

- Data Acquisition and Quality: Capturing high-resolution and accurate data from real-world roadways requires advanced sensor setups, such as LiDAR, high-definition cameras, and sensor fusion techniques. Data quality is critical to ensure that 3D reconstructions faithfully represent real-world environments and can be seamlessly integrated into simulation platforms.

- Selection of an Optimal Vehicle Simulator: Identifying a simulator that aligns with the specific requirements of this project involves analyzing various platforms to assess their compatibility, performance, and adaptability for high-fidelity 3D environments. This challenge requires evaluating simulators based on criteria such as real-time rendering capability, adaptability to custom environments, and support for complex ADAS and autonomous driving algorithms.

- High-Quality 3D Reconstruction of Highway Environments: Generating realistic, high-resolution 3D models of BrazilianScenarios demands advanced techniques for point cloud generation and textured meshing. This process must not only capture the geometric and textural intricacies of actual roadways but also achieve a balance between detail and computational efficiency to enable real-time simulation. The challenge lies in producing reconstructions that are both visually accurate and optimized for seamless integration into a simulation environment.

- Integration of 3D Reconstructions with the Selected Simulator: Successfully incorporating the generated 3D environments into the chosen simulator requires overcoming compatibility and configuration challenges. Ensuring that the reconstructed highway environments load efficiently and operate smoothly within the simulator is crucial. This involves addressing potential issues related to file formats, scene rendering, model scaling, and maintaining simulation fidelity and stability under varying test conditions.

Each of these challenges underscores the complexity of creating a robust, realistic integration capable of supporting autonomous vehicle testing and development within a high-fidelity virtual environment. In the context of this research, high fidelity refers to the creation of 3D reconstructed environments and simulation assets that closely replicate real-world physical and visual characteristics. This includes the accurate representation of geometric details, material properties, textures, lighting conditions, and environmental elements such as road surfaces, signage, and urban structures. High-fidelity models ensure that the virtual environment used for autonomous vehicle simulations provides realistic scenarios that are crucial for validating perception systems and ADAS functionalities. If high level of visual and physical accuracy is maintained, the simulations can better reflect real-world conditions, thus improving the reliability of sensor responses, object detection, and decision-making processes in autonomous driving applications.

1.3 Proposition

This dissertation proposes the development of a high-fidelity, real-time simulation pipeline that incorporates detailed 3D reconstructions of Brazilian scenarios to support the testing and validation of autonomous vehicle algorithms. Using an simulation environment, the project will generate realistic 3D models through advanced point cloud generation and mesh texturing techniques. These models will be integrated to allow for comprehensive testing of ADAS and other control algorithms under simulated conditions that closely mimic real-world Brazilian highway environments.

The proposed platform will serve as a versatile testbed for multiple functions, such as developing and validating ADAS functionalities, ensuring safety and operational standards compliance, creating a framework for real-time simulation of automated driving strategies that can account for local infrastructure and traffic dynamics. This integration of 3D reconstruction with real-time simulation not only advances testing capabilities but also contributes to the development of next-generation vehicle control systems designed for safety and resilience.

1.4 Objectives

The primary objective of this research is to develop a structured workflow for integrating high-quality 3D reconstructed highway environments into a vehicle simulation platform. This includes selecting suitable reconstruction techniques, generating detailed 3D models, and successfully implementing these environments. The project aims to enhance the realism and applicability of simulation platforms, facilitating the design and validation of autonomous vehicle algorithms in realistic driving scenarios. Specific objectives include:

- Identify and analyze the primary vehicle simulators currently in use to select the most suitable tool for this project.
- Identify and analyze the primary reconstruction techniques currently in use to select the most suitable tool for this project.
- Develop High-Quality 3D Reconstructions: Generate detailed point clouds and textured meshes of Brazilian Scenarios.
- Understand and implement the integration of 3D reconstructions: Study the integration process and successfully implement the reconstructed highway environment within Unreal Engine.

1.5 Contributions

This work presents several contributions to the field of three-dimensional reconstruction for autonomous vehicle simulation. The main achievements of this research can be summarized as:

- **Investigation of 3D Reconstruction Techniques:** A detailed study of various 3D reconstruction methodologies was conducted, covering techniques for point cloud generation, mesh reconstruction, and post-processing. This analysis provided a comprehensive understanding of the advantages and limitations of different approaches, helping to establish a solid foundation for integrating reconstructed environments into driving simulations.
- **Systematic Review of Reconstruction and Simulation Methods:** A systematic review was carried out following the PRISMA methodology, using NVivo for qualitative data analysis. This review enabled the identification and classification of relevant reconstruction techniques, simulation tools, and integration strategies, contributing to a structured and comprehensive understanding of the state of the art.

- **Creation of a Custom Dataset from UTFPR:** A dataset comprising 50 images of the UTFPR campus was captured and structured to serve as a basis for 3D reconstruction experiments. This dataset enables controlled evaluations of reconstruction techniques within an academic environment.
- **3D Reconstruction of UTFPR:** The captured dataset was used to perform a 3D reconstruction of the UTFPR campus, demonstrating the applicability of the selected methodologies. The reconstructed model serves as a test case for validating integration with simulation platforms and provides a practical example of environment digitization for autonomous vehicle research.
- **Integration of Unreal Engine Maps with CARLA:** A practical investigation was conducted to integrate maps from Unreal Engine into the CARLA simulator. Through this process, it was discovered that a successful manual integration requires two specific files, which are essential for correctly importing and rendering the reconstructed environments. This finding provides valuable insights for future work aiming to streamline the integration process.
- **Selection of CARLA as the Primary Simulation Platform:** After evaluating multiple driving simulation platforms, CARLA was selected as the primary tool due to its open-source nature, flexibility, and strong support for autonomous vehicle research. Other simulators, such as SUMO and LGSVL, were analyzed but presented limitations regarding 3D scene reconstruction capabilities.
- **Quantitative and Qualitative Evaluation of Reconstruction Methods:** The performance of selected 3D reconstruction techniques, including COLMAP, NeRF, 3D Gaussian Splatting, and MicMac, was assessed using Cloud Compare. The analysis included a comparison of accuracy, computational efficiency, and overall feasibility for integration into simulation platforms.
- **Proposal for Future Automation of the Integration Process:** Based on the findings of this research, a future direction has been outlined for automating the integration of reconstructed environments into driving simulators. The development of tools to generate the necessary files for CARLA automatically could significantly improve the efficiency and accessibility of the simulation setup.

These contributions provide valuable insights for applications, supporting the advancement of realistic virtual environments and facilitating the testing and validation of self-driving technologies.

1.6 Dissertation Structure

This dissertation is structured as follows:

- Chapter 2: Reviews existing research on 3D reconstruction methodologies and simulation platforms for autonomous vehicle testing, with a focus on the integration of real-world data in virtual environments.
- Chapter 3: Presents the methodological framework, including data acquisition techniques, 3D model generation, and the technical process for integrating these models within the chosen simulator.
- Chapter 4: Details how the 3D recontruction methodology was carried out.
- Chapter 5: Details the performance analysis of the proposed system, comparing different 3D reconstruction techniques and evaluating their impact on simulation accuracy and performance.
- Chapter 6: Summarizes the key findings of the study, discusses limitations, and suggests directions for future research on simulation platforms for autonomous vehicles.

2 LITERATURE REVIEW

This chapter provides a comprehensive overview of the key topics relevant to the study, focusing on the evolution, types, functionalities, and future trends of automotive simulation and scene reconstruction. The literature review is structured to cover historical development, scientific comparisons, significant innovations, and the implications of these technologies in the automotive industry.

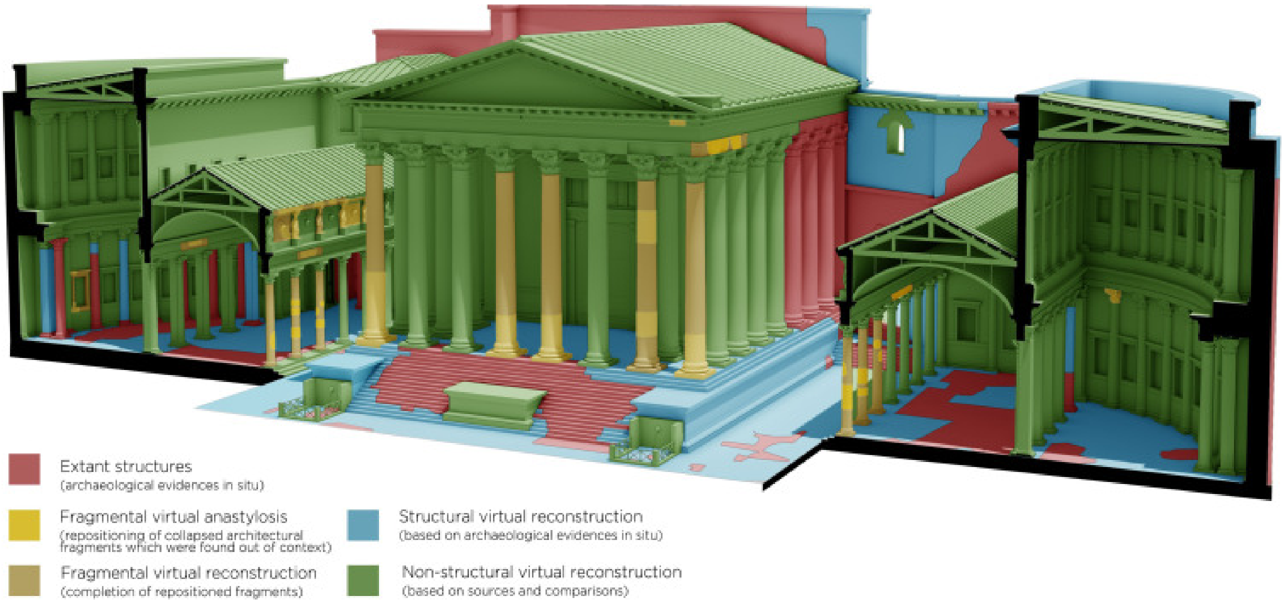
2.1 Background

3D Reconstruction refers to the process of creating a digital three-dimensional representation of an object, scene, or environment using computational techniques. It involves capturing data about the geometry, texture, and spatial relationships of objects, typically through inputs such as photographs, videos, laser scans, or depth sensors. The captured data is then processed using algorithms to reconstruct a 3D model that can be visualized, manipulated, and analyzed on a computer. Applications of 3D reconstruction span various fields, including medical imaging (e.g., for organ modeling), archaeology (e.g., for site preservation), virtual reality, gaming, and autonomous vehicles, where accurate spatial understanding is essential. The resulting models can range from simple shapes to highly detailed and textured replicas, depending on the fidelity and amount of input data available (Prabhasavat; Homgade, 2008; Cheriet *et al.*, 2007).

The concept of 3D reconstruction has its roots in the early developments of computer vision and photogrammetry in the mid-20th century. Photogrammetry, a technique used to measure distances and create maps from photographs, laid the foundation for extracting 3D information from 2D images (Slama, 1968). In the 1970s and 1980s, advancements in computational geometry and computer graphics allowed for the development of algorithms to reconstruct 3D structures from multiple photographic perspectives (Marr, 1982). Early breakthroughs included techniques like stereo vision, where two or more images are used to calculate depth information, and structure-from-motion (SfM), which estimates 3D shapes from a series of 2D images captured from different viewpoints (Longuet-higgins, 1981b). The increasing availability of computational power and improved imaging devices in the 1990s and 2000s further accelerated the adoption of 3D reconstruction in fields such as archaeology, robotics, and entertainment (Szeliski, 2010). Today, 3D reconstruction is a multidisciplinary endeavor, combining advancements in artificial intelligence, sensor technology, and photogrammetric methods to produce highly accurate models of real-world objects and environments.

A 3D reconstruction has evolved significantly over the decades, becoming a vast and multifaceted field. Numerous new methodologies have emerged, driven by advancements in computer vision, artificial intelligence, and sensor technology (Szeliski, 2010; Hirschmüller, 2008). Techniques such as photogrammetry, structure-from-motion (SfM), stereo vision, and the use of LiDAR sensors have revolutionized the accuracy and scalability of 3D modeling processes (David; Ma, 2020; Furukawa; Ponce, 2010). Today, 3D reconstruction is extensively explored across industries, with applications ranging from entertainment to autonomous vehicles (Seitz *et al.*, 2006; Besl; Mckay, 1992). Companies like Google, Meta, and Autodesk, as well as startups focused on augmented reality (AR) and virtual reality (VR), heavily invest in research to refine and innovate 3D reconstruction technologies. The integration of real-time processing and machine learning has further expanded its potential, making it a cornerstone for emerging fields such as digital twins and smart cities (Wu; Zhang, 2020a). The widespread adoption underscores how 3D reconstruction has transformed from a niche research area into a critical commercial and scientific domain.

Figure 3 – Forum of Augustus in VR



Source: Ferdani *et al.* (2020).

(Ferdani *et al.*, 2020) explores the integration of 3D reconstruction techniques with immersive virtual reality (VR) for cultural heritage applications, focusing on the Forum of Augustus in Rome. As can be seen in figure 3 It examines how VR technologies, particularly those used in "serious games" and virtual museums, can enhance the dissemination of historical knowledge through interactive and engaging "edutainment" experiences. The authors emphasize the importance of historical accuracy and validation in creating 3D reconstructions for VR applications. They propose a detailed workflow that combines virtual archaeology and advanced 3D modeling techniques to produce historically faithful assets for immersive experiences. This process involves collaboration across disciplines, requiring significant time and expertise to ensure the models align with historical evidence. The study builds on the authors' prior work and leverages modern VR tools, such as PlayStation VR, to assess the effectiveness and challenges of deploying these applications. This case study demonstrates the practical application of this workflow. The authors analyze how immersive VR technologies can bridge the gap between historical research and public engagement, offering a new medium for exploring and understanding the past. The paper highlights the technical and interdisciplinary complexities involved in creating accurate, interactive VR experiences, positioning 3D reconstruction and virtual archaeology as pivotal tools for the future of cultural heritage preservation and education.

(Bécue *et al.*, 2020) explores an innovative approach to using digital twin technology in the context of modern manufacturing. The authors propose a conceptual framework that extends traditional digital twin functionalities to enhance both optimization and resilience within factories. By integrating real-time data with advanced simulation models, this approach enables continuous monitoring, predictive analysis, and decision-making support. The work highlights how these capabilities can improve operational efficiency, reduce downtime, and adapt to unexpected disruptions, making factories more robust and responsive. This study is particularly relevant as industries transition toward smart manufacturing paradigms, emphasizing digital transformation and Industry 4.0 technologies. The paper provides valuable insights into how digital twins can contribute to the sustainable and adaptive growth of future manufacturing ecosystems.

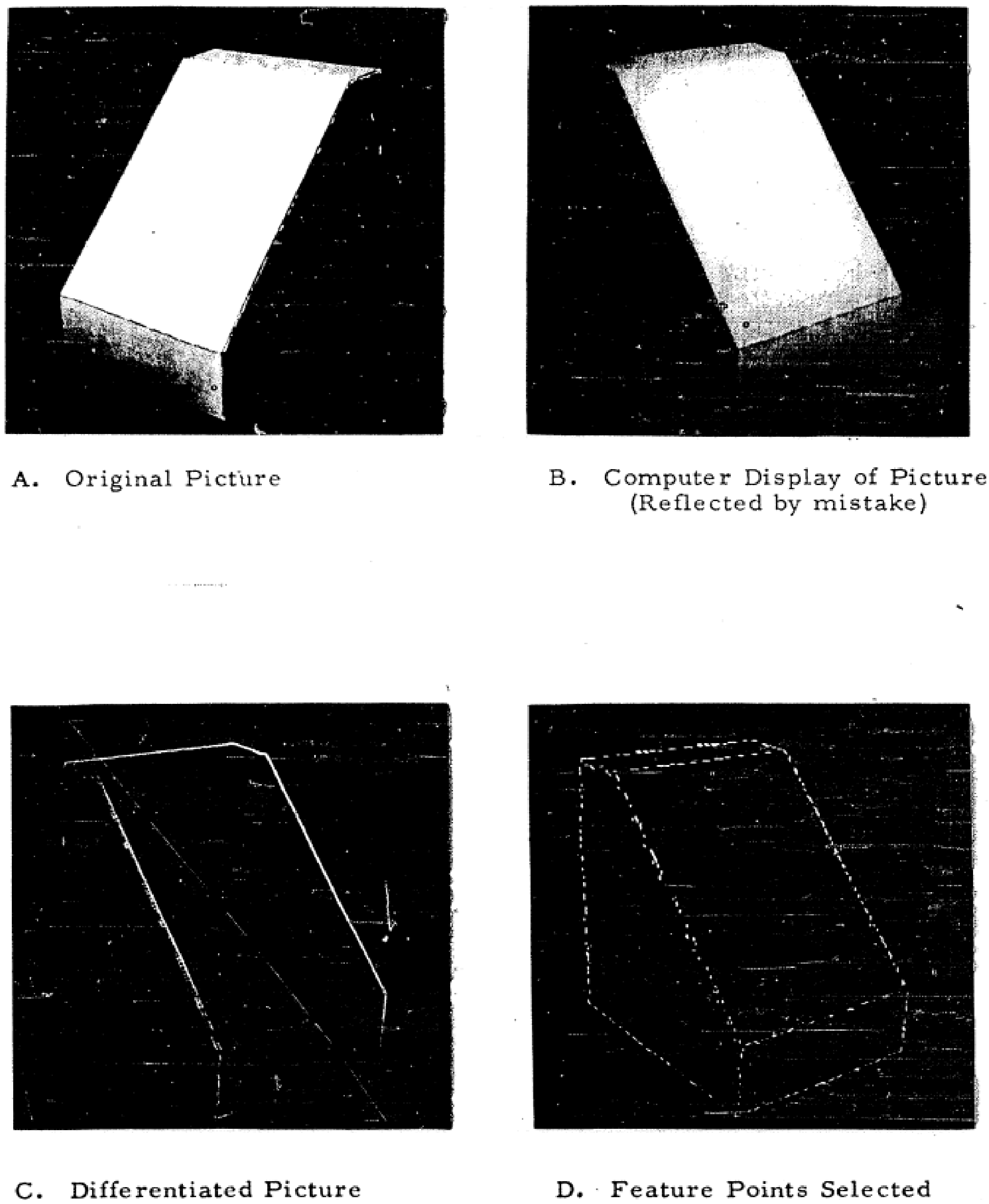
Studying 3D reconstruction is of paramount importance due to its transformative impact on various scientific, industrial, and creative fields. This technology enables the accurate digital replication of physical objects and environments, fostering advancements in fields like medical imaging, where precise anatomical models aid in diagnostics and surgical planning, and archaeology, where it helps preserve and analyze historical artifacts. Moreover, 3D reconstruction drives innovation in industries such as autonomous vehicles, robotics, and virtual reality, where spatial understanding is critical. By developing robust algorithms and techniques for 3D modeling, researchers can address challenges like scalability, real-time processing, and accuracy, unlocking new possibilities for smart manufacturing, digital twins, and immersive entertainment. As we move toward increasingly digital and interconnected systems, understanding and advancing 3D reconstruction methodologies remain vital for progress across disciplines (Poullis; You, 2011).

This chapter provides a comprehensive examination of 3D reconstruction, tracing their historical development, categorizing the various types, exploring their evolution over the past decade, and detailing their primary functions and features. It also discusses emerging trends and future directions in technologies, highlighting their critical role in the automotive industry and their potential for future innovation.

2.1.1 History

The first 3D reconstruction efforts can be traced back to the 19th century with the advent of photogrammetry, a technique for creating 3D-like models and maps using overlapping photographs. Aimé Laussedat, often referred to as the "father of photogrammetry," began using this method in the 1850s, marking the earliest attempts to extract three-dimensional information from two-dimensional images (Laussedat, 1850). However, these analog methods relied on manual geometric calculations. The first computational 3D reconstruction emerged in the 20th century. Larry Roberts' 1963 Ph.D. thesis at MIT is a landmark work, introducing a computational framework to reconstruct 3D objects from 2D line drawings (Roberts, 1963).

Figure 4 – Larry Roberts' 3D reconstruction



Source: Roberts (1963).

The reconstruction of a planar-surfaced object from robert's work can be found on figure 4. This marked the transition from manual to algorithmic approaches in 3D modeling. Later, in 1981, Hugh C. Longuet-Higgins proposed a mathematical model for recovering 3D structure from moving objects using stereo vision (Longuet-higgins, 1981a). This work laid the foundation for structure-from-motion (SfM), a method that became central to modern 3D reconstruction.

Thus, while Laussedat's photogrammetric methods represented the first manual 3D reconstructions, computational approaches were pioneered by Roberts and further advanced by Longuet-Higgins.

During the 1980s and 1990s, significant advancements were made in 3D reconstruction with the development of techniques such as stereo vision and structure-from-motion (SfM). Stereo vision uses pairs of images taken from slightly different viewpoints to calculate depth information, enabling the creation of a 3D representation of a scene (Longuet-higgins, 1981a). Similarly, SfM leverages sequential images captured from different angles to estimate the structure of objects or environments by tracking and matching feature points across frames (Szeliski, 2010). These methods marked a shift toward more automated and precise 3D modeling processes. In the 2000s, the field advanced further with the emergence of powerful graphics processing hardware and the early adoption of machine learning algorithms. These technologies significantly increased the speed and accuracy of 3D reconstructions, allowing for real-time applications. This period saw 3D reconstruction become a cornerstone in fields like augmented reality (e.g., virtual overlays on real-world environments), robotics (e.g., environment mapping for navigation), and medical imaging (e.g., detailed models of organs for surgery planning) (Furukawa; Ponce, 2010; Hirschmüller, 2008).

Stereo vision has its roots in the early exploration of depth perception, inspired by the human visual system. It works by capturing two or more images from slightly different viewpoints and analyzing the disparities between corresponding points to calculate depth. This technique gained prominence in the 1970s and 1980s with the development of computer algorithms capable of automating point matching between images. Early breakthroughs included the use of epipolar geometry to simplify the search for corresponding points, making the process computationally feasible (Longuet-higgins, 1981a). Stereo vision became a cornerstone for 3D reconstruction, enabling applications such as robotic navigation and topographic mapping. Over the decades, improvements in image processing and optimization algorithms, like the introduction of Semi-Global Matching (SGM) (Hirschmüller, 2008), further enhanced its precision and efficiency, paving the way for real-time applications in fields like autonomous vehicles and augmented reality.

Structure-from-Motion (SfM) evolved as an extension of stereo vision, leveraging sequences of images captured from different perspectives to reconstruct 3D geometry. The technique was first formalized in the 1980s and gained traction in the 1990s as computational resources became more accessible. SfM uses feature tracking across multiple frames to estimate both the camera's motion and the 3D structure of the scene simultaneously (Szeliski, 2010). Unlike stereo vision, which requires multiple synchronized cameras, SfM works with unordered image sets, making it versatile and widely applicable. Its ability to reconstruct large-scale and complex environments with minimal equipment has made it a critical tool in fields such as archaeology, urban planning, and filmmaking (Furukawa; Ponce, 2010). Recent advancements in machine learning and computational geometry have further refined SfM, enabling high-quality reconstructions even from sparse or noisy data.

Today, 3D reconstruction continues to evolve, driven by innovations in deep learning and LiDAR (Light Detection and Ranging) technology. Deep learning algorithms allow for robust and efficient reconstruction from complex or incomplete datasets (Wu; Zhang, 2020b), while LiDAR provides precise depth measurements, even in challenging conditions like low light or large-scale environments (Besl; Mckay, 1992). These advancements have expanded the scope of 3D reconstruction to a variety of industries, including entertainment (creating virtual worlds for gaming and movies), autonomous systems (self-driving cars requiring accurate maps), and smart cities (urban planning with 3D modeling of infrastructure). These developments underline the importance of 3D reconstruction as a transformative technology across numerous domains.

2.1.2 Areas of research

3D reconstruction is a dynamic field with applications spanning a wide range of disciplines, each leveraging its capabilities to address unique challenges. In medical imaging, enables the creation of detailed anatomical models, revolutionizing diagnosis, treatment planning, and surgical procedures. In archaeology, it plays a crucial role in preserving and analyzing cultural heritage, from artifacts to entire historical sites. The field also finds extensive application in city planning, where it aids in visualizing infrastructure projects, optimizing urban design, and managing disaster response scenarios. Furthermore, emerging areas such as the digital twin concept extend the scope, creating real-time synchronized virtual models of physical systems for continuous monitoring and optimization. The versatility of techniques underscores their importance in addressing complex problems across scientific, industrial, and societal domains (Poullis; You, 2011).

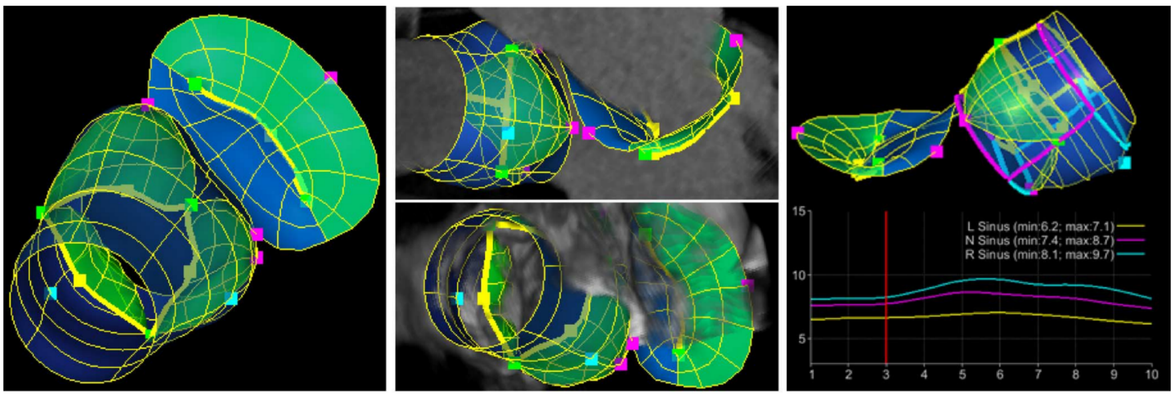
2.1.2.1 Medical imaging

In medical applications, it is utilized to build detailed 3D body images out of CT scans, MRIs, and ultrasounds. These models enable medical professionals to visualize anatomical structures in 3D, aiding in diagnosis, treatment planning, and surgical simulation. For example, they may also be used in neurosurgery to create models of the brain, helping surgeons plan intricate procedures and visualize the spatial relationship between tumors and surrounding structures. Similarly, in orthopedics, reconstructions of bones and joints assist in preoperative planning and guide surgeons during complex procedures, such as joint replacement surgery (Prabhasavat; Homgade, 2008; Cheriet *et al.*, 2007).

One notable application in medical imaging is the creation of patient-specific anatomical models for preoperative planning and surgical simulation. For instance, in neurosurgery, 3D reconstructions are generated from MRI and CT scan data to produce detailed models of the brain, capturing critical structures such as blood vessels, tumors, and functional areas. These models allow surgeons to visualize complex spatial relationships, plan precise surgical approaches, and predict potential complications. A significant example is (Cheriet *et al.*, 2007), which introduced a method for reconstructing 3D models of spinal deformities, particularly in patients with scoliosis. This approach enabled precise assessment and improved surgical outcomes.

Another significant study in medical imaging focuses on cardiovascular applications. (Ionasec *et al.*, 2010) introduced a method for reconstructing 4D patient-specific heart models from CT imaging data. This approach enables dynamic visualization of the heart's anatomy and function throughout the cardiac cycle. By integrating advanced segmentation and motion tracking algorithms, the study allowed for the accurate reconstruction of cardiac chambers, valves, and vessels. These models are valuable for diagnosing heart conditions, planning minimally invasive procedures, and simulating interventions such as valve repair or replacement. The study highlights the transformative potential in understanding complex cardiovascular structures, improving the precision of treatments, and enhancing patient outcomes in cardiology. Figure 5 provides an overview of the patient-specific cardiac model developed by Ionasec et al. (Ionasec *et al.*, 2010). Subfigure (a) depicts the physiological model of the aortic-mitral coupling, highlighting the structural and functional interdependence of these components. Subfigure (b) demonstrates how the model is fitted to patient-specific data derived from CT (top) and TEE (bottom) imaging, ensuring both anatomical accuracy and clinical relevance. Finally, subfigure (c) showcases an example of model-driven quantification, illustrating the dynamic changes in the volumes of the aortic valve sinuses over the cardiac cycle.

Figure 5 – (a) Physiological model of the aortic-mitral coupling. (b) Patient-specific model fitted to CT (top) and TEE (bottom) data. (c) Example of model-driven quantification volumes of the aortic valve sinuses over the cardiac cycle.



(a) (b) (c)

Source: Ionasec *et al.* (2010).

By providing detailed, patient-specific anatomical models that enhance diagnosis, treatment planning, and surgical precision, 3D reconstruction has revolutionized medical imaging. Applications range from neurosurgery and orthopedics to cardiovascular imaging, where 3D models improve understanding of complex structures and facilitate minimally invasive procedures. As computational power and imaging technologies continue to evolve, the integration of

3D reconstruction with advanced techniques such as machine learning and real-time imaging promises to further expand its impact, driving innovation and improving patient outcomes across a wide array of medical disciplines.

2.1.2.2 Archeology

Archaeology uses these techniques to create digital models of artifacts, archaeological sites, and even entire landscapes. These models aid in the preservation, analysis, and dissemination of cultural heritage. For instance, 3D models have been employed to create detailed models of ancient artifacts, enabling researchers to study them remotely and preserve them digitally for future generations. Furthermore, 3D reconstructions of archaeological sites provide valuable insights into past civilizations, allowing researchers to analyze the spatial organization of ancient cities, study architectural features, and simulate historical scenarios (Patay-horváth, 2014; Lercari, 2017).

A pioneering study in the field of virtual archaeology was presented by (Patay-horváth, 2014), focusing on the 3D reconstruction of the east pediment of the Temple of Zeus at Olympia. This research addresses one of the long-standing challenges in classical archaeology: interpreting and reconstructing the fragmented sculptures of this iconic monument. The reconstructed plaster mode can be seen in figure 6.

Figure 6 – Reconstructed plaster mode of the east front of the temple of Zeus at Olympia.

Source: Patay-Horváth (2014).

Leveraging advanced 3D modeling technologies, the study integrates archaeological data, historical records, and modern visualization tools to create a comprehensive virtual reconstruction of the pediment. This approach not only provides a new perspective on the original design and layout of the sculptures but also facilitates further analysis and public dissemination. By solving a historical puzzle using state-of-the-art technology, this work demonstrates the potential of 3D reconstruction to enhance our understanding of ancient cultural heritage, bridging the gap between traditional archaeology and digital innovation.

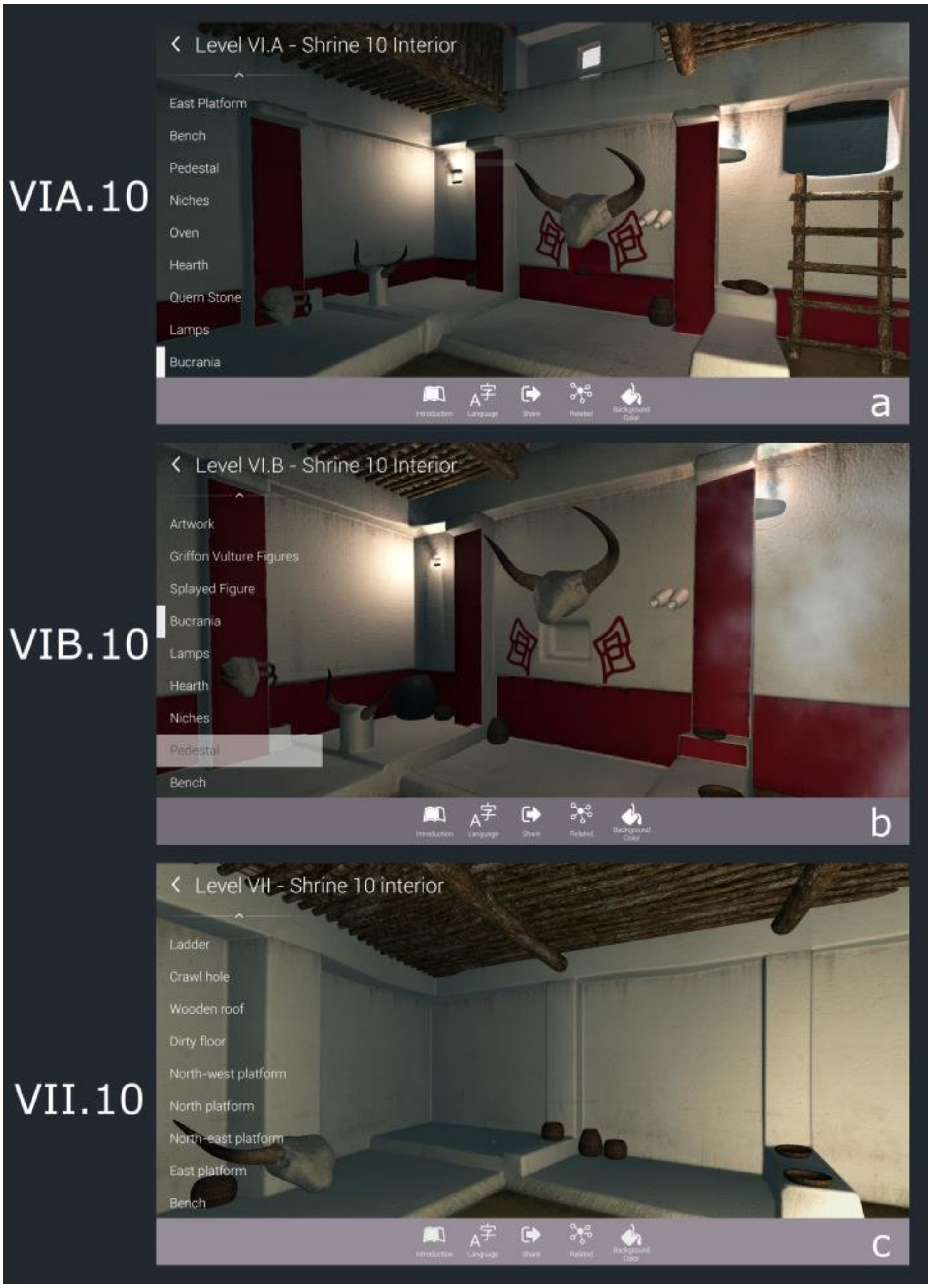
In his study, Lercari (Lercari, 2017) explores the use of 3D visualization techniques in reflexive archaeology through the virtual reconstruction of the history houses at Çatalhöyük, a Neolithic settlement in present-day Turkey. The research focuses on reconstructing the architectural and cultural context of these ancient dwellings, which are renowned for their complex stratigraphy and symbolic wall paintings. Figure 7 shows the virtual representation of the shrine. Using advanced 3D modeling and immersive visualization technologies, the study provides a detailed and interactive representation of the site, allowing archaeologists and the public to virtually explore the spatial organization and material culture of Çatalhöyük. Lercari emphasizes the reflexive aspect of this approach, where the reconstruction process itself becomes a tool for interpreting archaeological data and testing hypotheses. This work highlights the transformative potential of 3D reconstruction in enhancing archaeological research and education, bridging the gap between excavation findings and their broader cultural significance.

2.1.2.3 City Planning

Similarly, these techniques are employed in city planning, to create detailed urban models for applications, such as traffic planning, navigation, and disaster management. These models enable city planners to visualize and simulate various urban scenarios, aiding decision-making processes. For example, 3D city models have been used to simulate the impact of new infrastructure projects, such as roads or buildings, on the surrounding environment and infrastructure. Furthermore, 3D reconstructions of cities are valuable tools for navigation and tourism, allowing users to explore them virtually and plan routes efficiently. Additionally, 3D city models are essential for disaster management, enabling emergency responders to plan and coordinate rescue operations effectively (Huang *et al.*, 2022; Kumar *et al.*, 2021).

Presenting a fully automatic approach for reconstructing compact 3D building models from large-scale airborne LiDAR point clouds, Huang et al (Huang *et al.*, 2022) addresses the frequent absence of vertical wall data in such datasets. The proposed method infers vertical walls by leveraging the structural observation that urban buildings typically consist of planar roofs connected to vertical walls extending to the ground. This process involves hypothesizing building surface faces using planar segments of both roofs and walls, followed by a hypothesis-and-selection-based polygonal surface reconstruction framework enhanced with novel energy terms and hard constraints to ensure accurate topology and detail recovery. The method's robust-

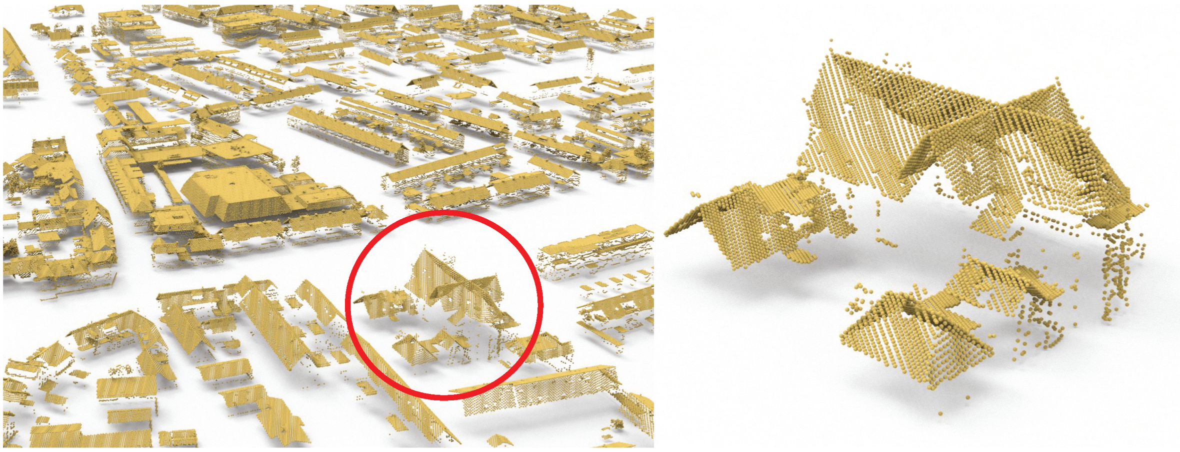
Figure 7 – Overlaying view of 3D reconstructions of (a) 'Shrine' VIA.10, (a) 'Shrine' VIB.10, and (c) 'Shrine' VII.10 in Lifeliqe.



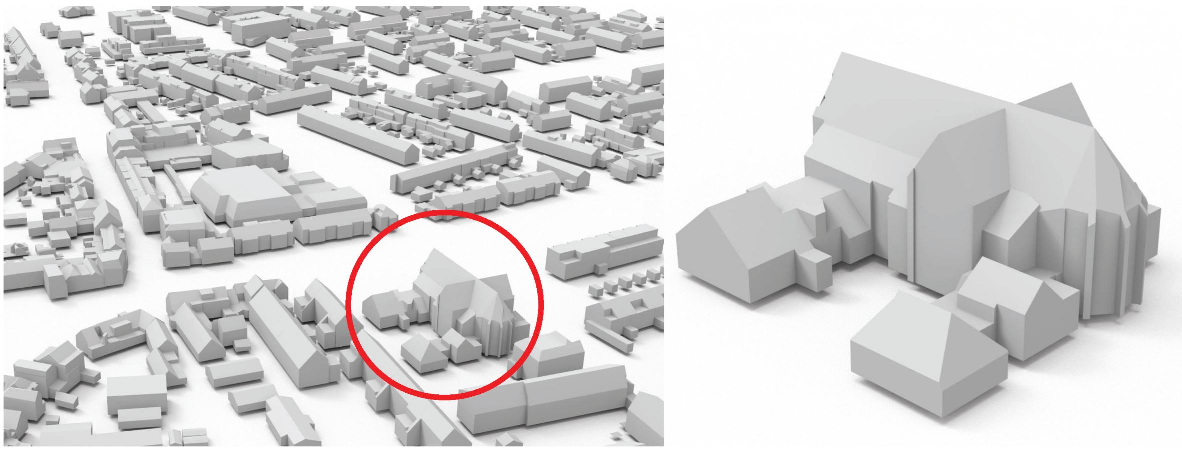
Source: Lercari (2017).

ness and precision were validated through experiments on various large-scale airborne LiDAR datasets, outperforming state-of-the-art approaches. Additionally, the authors generated a new dataset containing point clouds and 3D models of 20,000 real-world buildings, aiming to foster further research in urban reconstruction and the application of 3D city models in urban development and planning. This work significantly advances the field of urban modeling by providing a reliable and automated solution for large-scale building reconstruction.

Figure 8 – Input airbone Lidar Point cloud and reconstruction.



(**a**) Input airborne LiDAR point cloud.



(**b**) Our reconstruction result

Source: Huang *et al.* (2022).

The study by Kumar et al. (Kumar *et al.*, 2021) introduces a novel approach for citywide reconstruction of cross-sectional traffic flow using videos captured by moving cameras. The proposed method leverages computer vision techniques to extract and analyze vehicle trajectories from video feeds, enabling the reconstruction of detailed traffic flow patterns across urban areas. Unlike traditional stationary camera setups, this approach utilizes data from moving platforms, such as vehicles or drones, to provide a more flexible and scalable solution for traffic analysis. By integrating trajectory data with spatial and temporal context, the method generates dynamic 3D reconstructions of traffic flow, offering valuable insights into congestion hotspots and urban mobility patterns. This study demonstrates the potential of leveraging moving camera systems for large-scale, cost-effective traffic monitoring, with applications in urban planning, traffic management, and smart city development.

2.1.2.4 Digital Twin

The concept of the digital twin represents a cutting-edge application of 3D reconstruction, enabling the creation of virtual replicas of physical systems. These digital counterparts are synchronized in real-time with their physical counterparts, allowing continuous monitoring, analysis, and optimization. In industrial settings, digital twins are extensively utilized to simulate and optimize factory workflows, enhance predictive maintenance, and ensure operational efficiency. For instance, a digital twin of a manufacturing line can help identify bottlenecks, test new configurations, and predict equipment failures before they occur. Beyond industry, digital twins are increasingly applied in urban environments to create virtual cities, which aid in managing infrastructure, optimizing resource usage, and simulating disaster scenarios. As the integration of IoT and AI technologies advances, digital twins are becoming indispensable tools across domains, bridging the gap between physical and digital worlds while fostering innovation and efficiency (Bécue *et al.*, 2020; Grieves; Vickers, 2014).

Bécue et al. (Bécue *et al.*, 2020) propose an innovative framework for implementing digital twins in the context of next-generation factories, emphasizing their role in enhancing optimization and resilience. The study introduces a conceptual model that integrates real-time data from physical systems with advanced simulations to create dynamic, virtual counterparts of factory environments. These digital twins are designed to support decision-making processes by enabling predictive analysis, anomaly detection, and adaptive response to disruptions. The authors highlight the potential of this approach to optimize workflows, reduce downtime, and enhance operational flexibility. Additionally, the framework incorporates resilience-focused features, allowing factories to adapt to unexpected changes or disruptions in production. This work demonstrates the transformative potential of digital twins as a cornerstone technology for Industry 4.0, bridging the gap between physical and virtual domains to create more efficient and adaptable manufacturing systems.

Grieves and Vickers (Grieves; Vickers, 2014) trace the origins and foundational principles of the digital twin concept, establishing it as a transformative framework in modern engineering and manufacturing. The study defines a digital twin as a virtual representation of a physical system, synchronized through real-time data exchange, enabling simulation, monitoring, and optimization across a system's lifecycle. The authors discuss the conceptual evolution of digital twins from early computer-aided design (CAD) systems to their integration with emerging technologies such as the Internet of Things (IoT) and predictive analytics. They highlight how digital twins can provide unprecedented insights into product performance, maintenance needs, and design optimization, facilitating more efficient and adaptive manufacturing processes. This foundational work underscores the digital twin's role as a core component of Industry 4.0, bridging physical and digital domains to improve decision-making, reduce costs, and enhance system resilience.

The Siemens Amberg Electronics Plant serves as a benchmark example of digital twin implementation in manufacturing, showcasing the transformative potential of Industry 4.0 technologies. At this facility, a digital twin replicates the entire production environment in real-time, integrating data from IoT sensors embedded in machinery. This continuously updated virtual model allows for workflow simulation, enabling the identification of bottlenecks and testing of new configurations without disrupting physical operations. Predictive maintenance is another key application, where the digital twin analyzes equipment performance data to forecast failures, reducing unplanned downtime and extending machine lifecycles. Additionally, adaptive scheduling, driven by real-time insights, optimizes production efficiency and reduces resource waste. This case study underscores how digital twins bridge the gap between physical and digital domains, enhancing manufacturing resilience and adaptability (Ag, 2019).

2.2 3D Models in Realistic Traffic Simulation Environments

Three-dimensional reconstruction plays a fundamental role in various domains, including computer vision, geospatial analysis, and urban modeling. However, its application in realistic vehicle traffic simulations, including interactions with cyclists, pedestrians, and other road agents, has been less explicitly emphasized in general discussions on 3D modeling research. One of the key challenges in autonomous driving research is the ability to simulate complex traffic scenarios that accurately reflect real-world conditions. Realistic virtual environments, built from detailed 3D models, provide an essential foundation for testing perception, planning, and control algorithms in autonomous vehicles. The integration of 3D models into simulation frameworks enables dynamic and interactive environments where vehicles, pedestrians, and cyclists can be modeled with high precision, improving the reliability and robustness of testing methodologies.

The study by (Azfar; Smith; Khan, 2024) explores a traffic co-simulation framework that integrates infrastructure camera sensing with reinforcement learning-based vehicle control. The research utilizes CARLA and SUMO to simulate real-world driving scenarios while incorporating high-fidelity 3D models for accurate scene representation. Their approach enables dynamic traffic adaptation, where vehicles respond realistically to changes in road conditions and traffic signals. This work relates to ours by demonstrating the importance of combining 3D modeling with driving simulation to enhance the realism and accuracy of traffic environments. Our research extends this concept by focusing on the direct integration of 3D reconstructed scenes into driving simulation platforms.

(Chen; Zhang; Liu, 2025) present a virtual-real-fusion framework for intelligent 3D traffic accident reconstruction. The authors propose using LiDAR-based point cloud reconstruction and deep learning techniques to create highly accurate 3D traffic environments for post-accident analysis. By integrating these models into traffic simulators, they aim to enhance forensic investigations and traffic safety studies. This study aligns with our research by emphasizing the use of detailed 3D reconstruction techniques to improve simulation accuracy. However, our work differs in its focus on integrating these models into real-time traffic scenarios for autonomous vehicle training, rather than forensic reconstruction.

The study on Rapidex by (Nayab; Rafiq; Malik, 2024) introduces a novel approach to integrating 3D models into transportation digital twins. Their research focuses on creating realistic urban traffic simulations by combining AI-generated 3D road scenes with dynamic simulation data. The authors discuss the challenges of aligning real-world geospatial data with digital simulations and propose an automated mapping technique to improve scene accuracy. This research closely parallels our work in its goal of developing a seamless integration pipeline between 3D models and driving simulators. While their approach focuses on large-scale urban environments, our work is more centered on the manual and semi-automated integration of high-quality 3D reconstructions into simulation platforms like CARLA.

(Crampen; Zhao; Lin, 2024) explore the development of 3D urban digital twins (UDTs) within Unreal Engine 5 for real-time traffic monitoring. Their approach integrates live data from real-world traffic systems into an immersive 3D environment, allowing for dynamic traffic visualization. Unlike conventional static 3D models, their framework supports interactive real-time updates, enabling better adaptability for smart city planning. This study highlights the importance of 3D scene reconstruction in transportation applications, complementing our work by showcasing the use of Unreal Engine for advanced visualization. Our research further builds upon this idea by examining the manual integration challenges of 3D maps into CARLA, expanding the possibilities for real-time simulation-based traffic studies.

(Salehi; Jafari; Kamal, 2025) present an application of systems engineering principles in constructing 3D environments for autonomous vehicle simulations. Their study focuses on the structured methodology required to integrate 3D environments with traffic systems, ensuring compatibility between real-world datasets and driving simulation software. They emphasize the need for precise model validation to maintain accuracy in AI-based vehicle training. This research aligns with our work by addressing the complexities of environment integration, a key challenge that was encountered while working. Our contribution extends this knowledge by identifying the specific requirements for manual file integration and proposing improvements to streamline the process.

Together, these studies reinforce the significance of integrating 3D models into autonomous driving simulations, each tackling different aspects of the challenge. Our work adds a unique perspective by investigating the detailed process of transferring high-fidelity 3D environments into a simulation plataform and evaluating the practical challenges associated with manual file preparation. This contribution is essential for advancing the realism of autonomous vehicle simulations, paving the way for more immersive and accurate driving scenarios. Furthermore, the use of 3D models in traffic simulation extends beyond autonomous driving, supporting research in traffic safety, pedestrian behavior analysis, and urban mobility planning. Accurately reconstructing road environments, crosswalks, urban landscapes, and traffic infrastructures, simulations can contribute to the development of intelligent transportation systems and smart city applications. Although the relevance of 3D reconstruction in realistic simulations is explored later in this dissertation, it is important to highlight early on that this research aligns with key areas of investigation in autonomous systems, computer vision, and transportation engineering. This contextualization ensures that the study's contributions are well integrated into the broader academic and technological landscape, reinforcing its importance in advancing simulation-driven research for vehicle automation and urban mobility.

2.2.1 Sensors and data acquisition process

Data acquisition is crucial for 3D reconstruction as the quality of point clouds and meshes relates to the ability to process optimal inputs. These inputs may be exclusive to a given source and sensor, such as color information from cameras, or available from different sensors, such as depth from direct conversions of radar and LiDAR measurements or indirectly estimated from cameras. Here, capturing rich color and texture information can support the learning process for spatial mapping with a heavy reliance on ambient light, which may decrease its reliability. In such a scenario, accurate depth measurements through active illumination, from LiDAR sensors, can offer a more reliable solution and contribute to improve camera-based models. These contributions guide the selection and development of reconstruction methods, ensuring optimal data processing. Considering the importance of sensors for data acquisition and effective reconstruction, some of the most commonly used sensors are presented on this subsection.

2.2.1.1 LiDAR (Light Detection and Ranging)

LiDAR (Light Detection and Ranging) is a technology that uses laser pulses to measure distances by calculating the time it takes for the emitted light to reflect back from a surface. This method enables the creation of precise, high-resolution 3D models of environments. LiDAR is widely employed in various applications, including autonomous vehicles, where it helps in obstacle detection and navigation, and topographical mapping, where it generates detailed representations of terrain and infrastructure. Its ability to capture intricate spatial details, even in low-light conditions, makes LiDAR an indispensable tool in fields such as urban planning, archaeology, and forestry (Fruh; Zakhor, 2001; Milella; Reina, 2014).

An example of LiDAR application in 3D reconstruction is the work by Hyyppä et al. (**??**), which utilized a RIEGL LMS-Q560 LiDAR sensor to reconstruct urban environments. The study focused on generating high-resolution 3D models of building facades and surrounding infrastructure. The RIEGL LMS-Q560, known for its high accuracy ranging capabilities and wide field of view, was mounted on a terrestrial platform to collect dense point cloud data. The researchers processed the data to create precise geometrical representations of urban features, demonstrating the LiDAR's ability to capture intricate details such as window frames, balconies, and surface textures. This work highlighted the potential of LiDAR in urban planning, architectural modeling, and heritage preservation, showcasing the versatility and precision of the RIEGL LMS-Q560 for detailed 3D reconstructions.

Abebe (Abebe, 2024) provides a comprehensive review of LiDAR technology and its applications in 3D city modeling. The study highlights LiDAR's versatility in urban planning, environmental monitoring, and decision-making processes. Key applications discussed include building reconstruction, solar potential assessment, urban vegetation analysis, and flood modeling. By leveraging LiDAR-generated 3D point clouds, the paper emphasizes the technology's ability to deliver high-resolution, accurate representations of urban environments. Furthermore, it explores advanced feature extraction, segmentation techniques, and deep learning algorithms for processing LiDAR data. The review underscores the transformative impact of LiDAR in addressing complex urban challenges, particularly in developing detailed 3D city models that support smarter, more sustainable urban development practices.

Figure 9 – Velodyne HDL-64E

Source: Huang *et al.* (2022).

Lidar HDL-64E 9 is one of the first models developed by Velodyne and was widely used in early autonomous vehicle projects, such as the DARPA Challenge (2005) (Thrun *et al.*, 2006). Its introduction represented a major advancement in 3D environmental perception, providing higher precision and accuracy in mapping surrounding objects. Given its historical relevance, it is important to highlight its pioneering role in the evolution of LiDAR technology. Although the HDL-64E was a milestone in the introduction of three-dimensional sensors for autonomous vehicles, LiDAR technology has evolved significantly since its inception. More recent LiDAR sensors, such as the Velodyne Alpha Prime, Ouster OS2, and Luminar Hydra, offer higher angular resolution, longer detection range, and improved capability to operate under various weather conditions. These advancements enable the acquisition of more precise and detailed three-dimensional representations of the environment, which is essential for advanced perception and navigation applications (Hall, 2011).

An example of LiDAR application in 3D reconstruction is the work by Kühner and Kümmerle (Kühner; Kümmerle, 2024). In their study, they present a framework for large-scale volumetric 3D scene reconstruction using LiDAR sensors, specifically addressing applications in autonomous driving and robotics. The method utilizes volumetric depth fusion combined with a cylindrical projection model to create detailed meshed representations of urban environments. The authors utilized Velodyne HDL-64E LiDAR sensors for their experiments and evaluations. This model can be seen in figure 9 The system is designed to handle loop closures and incorporates advanced algorithms to manage sensor-specific challenges, such as rolling shutter effects and non-single-viewpoint issues. Evaluations on real-world datasets, including the KITTI odometry benchmark, demonstrate the framework's capability to produce high-quality reconstructions with minimal user intervention. The approach achieves a high level of detail over extensive areas, such as a 3.7 km route, processed efficiently using consumer-grade graphics hardware. This work highlights the potential of LiDAR-based volumetric reconstruction for enhancing localization, mapping, and simulation in autonomous systems.

2.2.1.2 Laser Scanners

Laser scanners are devices that use laser beams to capture high-precision spatial data, typically of specific objects or environments. They are particularly effective for close-range and indoor applications, where capturing fine details is essential. Common use cases include cultural heritage preservation, where detailed 3D models of artifacts or structures are created, as well as industrial applications like equipment inspection and architectural modeling (Esrafilian; Gesbert, 2017; Fruh; Zakhor, 2001). Laser scanners are often stationary or handheld, offering millimeter-level accuracy for confined or small-scale areas.

Although LiDAR (Light Detection and Ranging) and laser scanning share the fundamental principle of using laser technology to measure distances, they differ in scale, application, and deployment. LiDAR is primarily designed for large-scale, long-range data collection, often used in outdoor environments such as topographical mapping, forest analysis, or autonomous vehicle navigation. LiDAR systems are typically mounted on platforms like drones, airplanes, or vehicles and integrate with GPS and inertial measurement units (IMUs) to generate large-scale, georeferenced 3D point clouds.

In contrast, laser scanners are more versatile for detailed, small-scale data acquisition, often used for close-range applications. They are stationary or handheld devices suited for capturing intricate details of objects, surfaces, or indoor environments. While LiDAR excels in covering expansive areas with moderate precision, laser scanning is ideal for projects requiring extremely high precision and resolution at shorter distances. The choice between the two depends on the scale, resolution, and specific needs of the application.

Figure 10 – Terrestrial laser scanner in operation at the field survey at Dunakapu Square, Győr.

Source: Fehér (2013).

In his study, Fehér (Fehér, 2013) explores the application of 3D laser scanners in archaeology, focusing on their effectiveness in documenting cultural heritage sites and artifacts. The research highlights the use of terrestrial laser scanners, capable of capturing up to one million points per second, to generate high-resolution 3D models of archaeological features, ruins, and artifacts. One significant implementation involved scanning the ruins of a medieval church in Hungary, using a scanner with a range of 0.3 m to 187 m to produce a point cloud of 225 million points. These scans enable non-contact, highly detailed documentation, preserving irregular surfaces and intricate structures with millimeter-level precision. The study demonstrates how 3D scanning technology can revolutionize the preservation and analysis of cultural heritage by creating scalable, accurate virtual representations that support further research, education, and public engagement. Figure 10 represents the terrestial laser in operation.

2.2.1.3 Cameras (RGB, Infrared, and Video)

Cameras are important tools in 3D reconstruction, each offering unique capabilities that enhance model generation. Standard RGB cameras capture detailed color images, providing the photometric information necessary for texture mapping and ensuring visually accurate 3D models. Infrared cameras, on the other hand, detect thermal variations, making them ideal for applications requiring temperature-sensitive data, such as in structural analysis or medical imaging. Video cameras capture continuous sequences of images, enabling the reconstruction of dynamic scenes and objects in motion. By combining data from these different types of cameras, 3D reconstruction workflows achieve higher accuracy and richer visual fidelity, playing a critical role in applications ranging from cultural heritage preservation to robotics and autonomous navigation (Poullis; You, 2011; Muñoz-silva *et al.*, 2021).

Poullis and You (Poullis; You, 2011) present a novel framework for the 3D reconstruction of urban areas, focusing on creating large-scale, photorealistic models from a combination of aerial and ground-level data. The method integrates data from various sources, including standard RGB cameras and LiDAR, to generate high-resolution 3D models of urban environments. The proposed pipeline emphasizes efficiency, using advanced feature extraction and surface reconstruction techniques to handle the vast and heterogeneous data typical of urban settings. The reconstructed models are optimized for visualization and analysis, providing detailed representations of buildings, roads, and other urban features. This work highlights significant advancements in urban modeling, offering potential applications in city planning, navigation systems, and virtual reality environments.

In their study, they utilized a combination of standard RGB cameras and LiDAR sensors to achieve high-resolution 3D reconstructions of urban areas. The RGB cameras were employed to capture detailed photometric data, providing rich color information for texture mapping, which is essential for creating photorealistic urban models. These cameras were mounted on both aerial platforms and ground-level vehicles to capture a diverse range of perspectives, ensuring comprehensive coverage of urban environments. The integration of RGB camera data with LiDAR point clouds allowed for precise alignment of geometric and visual information, resulting in accurate and visually detailed 3D models. The use of RGB cameras, combined with their seamless integration into the reconstruction pipeline, underscores their importance in enhancing the fidelity and usability of urban 3D models.

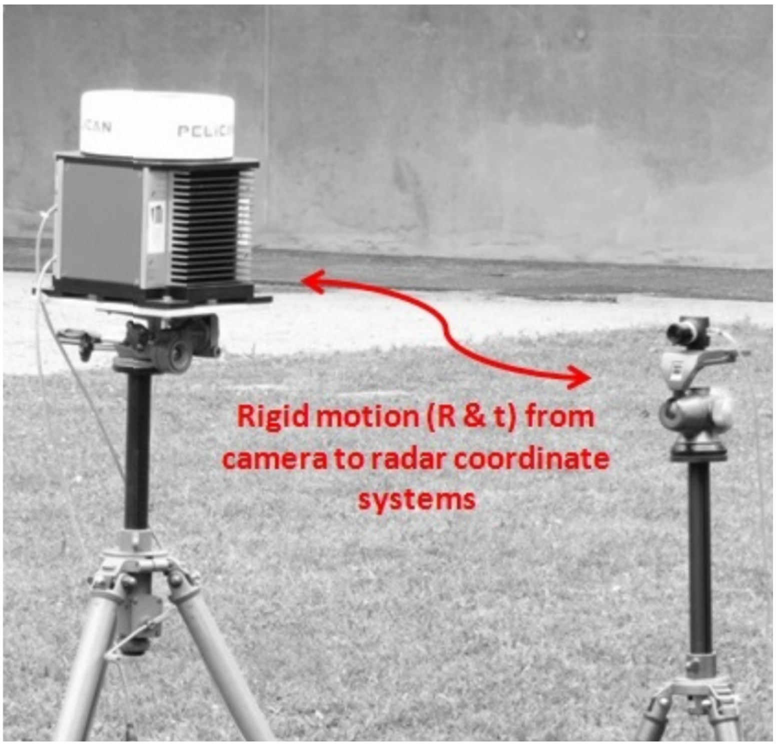
Another interesting study is provided by Muñoz-Silva et al. (Muñoz-silva *et al.*, 2021). The authors provide a comprehensive survey on point cloud generation techniques for 3D scene reconstruction, exploring various methodologies and their applications. The study categorizes existing methods based on the type of input data, such as RGB images, depth maps, and LiDAR scans, and examines their strengths and limitations in different scenarios. Special emphasis is placed on the integration of multiple data sources to enhance the accuracy and detail of 3D reconstructions. The authors discuss key challenges in point cloud processing, including noise reduction, alignment, and computational efficiency, and highlight the emerging role of machine learning in addressing these issues. By synthesizing advances in point cloud generation, the paper serves as a valuable resource for researchers and practitioners working on 3D reconstruction for applications in robotics, virtual reality, and autonomous systems.

2.2.1.4 Radar Sensors

Radar sensors are effective in environments with low visibility or challenging weather conditions, as they can penetrate rain, fog, and dust. This makes them invaluable for applications such as autonomous vehicles and robotics, where reliable depth estimation and navigation are critical. Unlike optical sensors, radar is less affected by lighting conditions, providing robust performance in a wide range of scenarios. Often used in combination with other sensors, such as LiDAR and cameras, radar enhances the accuracy and reliability of sensor fusion systems by contributing complementary data. This integration is particularly beneficial in detecting obstacles, estimating velocities, and ensuring safe navigation in complex environments (Esrafilian; Gesbert, 2017; Untzelmann *et al.*, 2013; Zhang; Jiang; Ai, 2015).

The study by Sun et al. (Sun *et al.*, 2021) introduces the 3DRIMR system, a deep learning-based architecture designed for 3D reconstruction and imaging using mmWave radar. This system addresses the challenges of radar data sparsity, noise, and low resolution by employing a two-stage conditional GAN framework. In the first stage, the model generates 2D depth images from raw radar intensity data using a 3D convolutional neural network. In the second stage, it constructs dense and smooth 3D point clouds from the 2D depth images. The study utilizes the Texas Instruments IWR6843ISK radar sensor, known for its high resolution along the range direction, even without the need for time-consuming synthetic aperture radar (SAR) processes. Experiments demonstrated the system's effectiveness in reconstructing 3D objects with high geometric detail, showcasing its potential for applications in autonomous vehicles, robotics, and low-visibility environments. The 3DRIMR architecture combines the efficiency of point cloud representations with the robustness of radar sensing, setting a benchmark for radar-based 3D reconstruction technologies.

Figure 11 – Radar and camera system



Source: Natour *et al.* (2015).

(Natour *et al.*, 2015) introduces an innovative approach for outdoor 3D reconstruction using a combination of a panoramic millimeter-wave (MMW) radar and a standard camera. Radar is used for its robust depth measurement capabilities, especially under challenging environmental conditions such as rain, fog, and dust. It provides precise distance and azimuth information, forming the foundation for sparse 3D mapping. However, radar alone cannot capture elevation details or textures. To overcome this limitation, the authors integrate data from the radar with images captured by the camera, which offers high spatial resolution and texture information. The study also introduces a novel geometric calibration method to align the radar and camera data accurately, enabling the generation of textured elevation maps. This combination of radar and vision data allows for efficient and accurate 3D reconstruction of large-scale outdoor environments, demonstrating the complementary strengths of these two sensor types. The combinantion is shown in figure 11.

2.2.1.5 Sensor fusion

As described, each sensor offers unique perspectives and qualities on a given subject and, when leveraged effectively, can enhance the reconstruction process. Integrating data from multiple sensors using sensor fusion techniques can further improve the overall quality of the inputs. Data acquisition methods are selected based on specific project requirements, with some relying on repeated measurements from a single sensor type and others leveraging multi-sensor combinations. For example, instrumented vehicles equipped with LiDAR and multiple cameras can capture high-resolution images and precise depth information of their surroundings, enabling detailed urban modeling (Fruh; Zakhor, 2001; Milella; Reina, 2014). Drones, or Unmanned Aerial Vehicles (UAVs), equipped with cameras are widely used for capturing large-scale topographic data from various angles, creating accurate 3D models of extended areas (Esrafilian; Gesbert, 2017; Fruh; Zakhor, 2001).

In close-range applications, handheld cameras allow researchers to gather overlapping images from different perspectives, ideal for detailed reconstructions in confined spaces (Poullis; You, 2011; Muñoz-silva *et al.*, 2021). Satellite imagery, such as that from Google Earth, provides extensive overhead views, enabling the 3D reconstruction of vast areas (Esrafilian; Gesbert, 2017; Untzelmann *et al.*, 2013; Zhang; Jiang; Ai, 2015). Additionally, pre-existing datasets like the KITTI dataset (Lee; Song; Jo, 2016), Geoportal (Kulawiak, 2022), City Intrinsic Images (Xie; Li; Qi, 2019), and OpenStreetMap (Untzelmann *et al.*, 2013) are invaluable for supplementing 3D reconstruction projects with high-quality data collected from various sensors.Combining these acquisition methods enhances data completeness and accuracy, making them applicable to a broad range of 3D reconstruction tasks, from cultural heritage preservation and urban planning to autonomous vehicle navigation.

These sensors and acquisition methods are essential for capturing data needed for 3D reconstruction. They can be used individually or in combination, depending on the project's specific requirements. Acquisition methods include using ground-based LiDAR scanners to record complex structures, rapid data acquisition with 2D laser scanners and cameras mounted on vehicles for city-scale modeling, and hierarchical contour methods for automatic 3D city reconstruction from LiDAR data (Li *et al.*, 2012). Techniques such as using panorama images and LiDAR scans at street level and employing LiDAR technology for quick, accurate 3D surface information acquisition contribute to the diversity of approaches in 3D reconstruction (Pylvänäinen *et al.*, 2012). These methods serve applications like augmented reality, cultural heritage preservation, urban planning, and autonomous navigation, underscoring the versatility and importance of sensor technology in 3D reconstruction (Poullis; You, 2011; Fruh; Zakhor, 2001; Lee; Song; Jo, 2016).

Fusing data from Inertial Measurement Units (IMUs) and other sensors plays a crucial role in enhancing accuracy and robustness. IMUs are composed of gyroscopes, accelerometers, and sometimes magnetometers, which provide orientation, velocity, and acceleration measurements. When integrated with data from other sensors, such as cameras, LiDAR, or GPS, IMUs contribute valuable information for motion estimation, pose estimation, and registration of sensor data. The resulting reconstruction can achieve higher precision and reliability by combining the complementary strengths of different sensors through fusion techniques, such as Kalman filtering or sensor fusion algorithms (Nakao *et al.*, 2004). IMU data helps compensate for motion-related artifacts, such as camera shake or vehicle movement, enabling more accurate alignment of 3D data points. Additionally, sensor fusion facilitates the creation of comprehensive 3D models by incorporating data from multiple sources, leading to a more complete and detailed reconstruction of the environment. Overall, IMU and sensor fusion techniques are critical in improving the quality and reliability of 3D reconstruction systems, making them indispensable components in several applications such as robotics, augmented reality, and autonomous navigation (Pylvänäinen *et al.*, 2012; Lee; Song; Jo, 2016).

2.3 Driver Scenarios for Vehicle Simulator

ADAS rely on digital cameras, radar, and other sensors to enhance driving safety and automation. Simulators—both static and dynamic—play a crucial role in testing and validating these systems by replicating real-world driving conditions in a controlled environment. These simulations integrate sensor fusion techniques, combining data from multiple sources to improve system accuracy and robustness.

Static simulators provide a controlled driving environment where the vehicle remains stationary while the driver interacts with a simulated scene displayed on screens or through virtual reality headsets. These simulators are particularly useful for evaluating ADAS in perception-based scenarios. They allow researchers to test system responses to various conditions without real-world risks. One of their key applications includes assessing ADAS perception systems under different environmental conditions, ensuring their robustness in detecting obstacles, road signs, and lane markings. Additionally, they play a crucial role in studying human-machine interaction (HMI), analyzing how drivers respond to visual and auditory warnings provided by ADAS.

Behavioral studies further benefit from static simulators, as they help evaluate driver reactions to automated interventions in a safe and repeatable manner.

In contrast, dynamic simulators incorporate motion feedback mechanisms that provide real-world forces to the driver using actuators. These simulators enhance realism by simulating acceleration, braking, and cornering forces, making them particularly valuable for evaluating vehicle dynamics and driver behavior under emergency conditions. Dynamic simulators facilitate real-time testing of ADAS functionalities, such as lane-keeping assistance, emergency braking, and adaptive cruise control (ACC). By replicating real-world driving scenarios, they offer an immersive experience that aids in understanding driver fatigue and distraction under prolonged driving conditions. Moreover, they are essential in testing ADAS interventions in complex traffic situations, helping validate the effectiveness of autonomous driving systems.

Static simulators are well-suited for ADAS perception tests that rely on cameras and radar for scene understanding. They are widely used in traffic sign recognition (TSR) evaluations, where digital road signs are displayed under different lighting and weather conditions to analyze system accuracy. Lane departure warning (LDW) systems are also tested using static simulators, ensuring that vehicles can reliably detect lane markings and provide timely alerts for unintended lane drifts. Forward collision warning (FCW) simulations introduce scenarios where a leading vehicle suddenly decelerates, assessing how well the system detects the hazard and issues alerts. Similarly, pedestrian detection tests simulate unexpected pedestrian crossings, helping to refine ADAS algorithms for improved safety in urban environments.

Dynamic simulators, on the other hand, excel in testing real-time motion-based ADAS interventions. Adaptive cruise control (ACC) simulations allow researchers to assess how well a vehicle adjusts its speed in response to surrounding traffic, including stop-and-go conditions. Emergency braking systems (EBS) are rigorously evaluated using sudden stop events, determining the effectiveness of braking responses in collision avoidance. Blind spot detection (BSD) tests involve dynamic vehicle interactions, simulating situations where cars approach from blind spots to validate alert mechanisms. Additionally, dynamic simulators enable the testing of autonomous overtaking maneuvers, analyzing how ADAS responds to slower-moving traffic by executing safe lane changes.

Sensor fusion plays a critical role in enhancing ADAS capabilities by combining data from multiple sources, including cameras, radar, LiDAR, and ultrasonic sensors. This integration improves object detection, classification, and tracking accuracy, making it an essential component in simulation environments. By validating multi-sensor perception reliability under diverse weather and lighting conditions, simulations help refine ADAS decision-making processes. Redundancy and fail-safe mechanisms are also tested, ensuring system functionality even when one sensor type fails—for example, a camera struggling in foggy conditions while radar maintains operational accuracy.

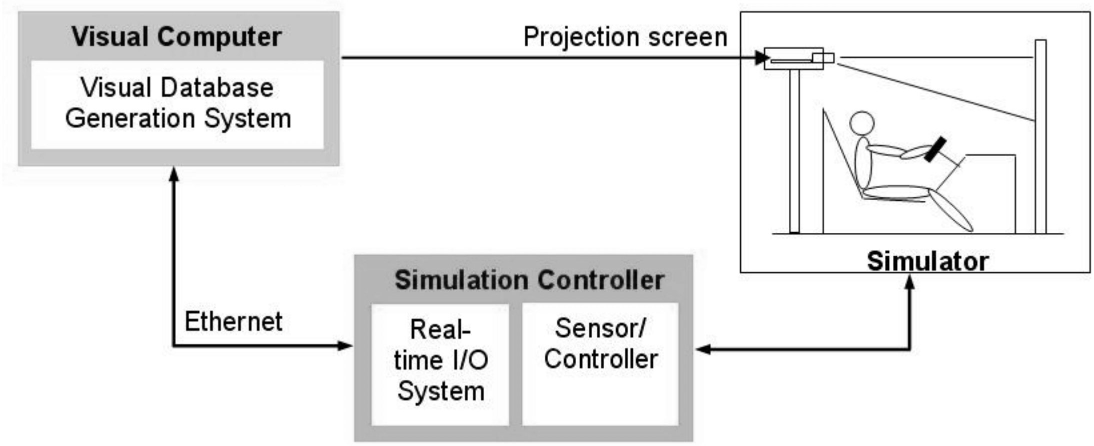
Sensor fusion techniques in ADAS simulations include data-level fusion, which merges raw sensor inputs for a more comprehensive environmental model. Feature-level fusion integrates extracted features from various sensors to enhance situational awareness and improve ADAS performance. Decision-level fusion combines independent sensor decisions to determine the final ADAS action, ensuring that the system responds accurately to real-world driving conditions. These approaches collectively contribute to the advancement of intelligent vehicle safety systems, making them more reliable and adaptive to varying traffic scenarios.

Vehicle simulators play a important role in the development and validation of ADAS and autonomous driving technologies. They provide controlled environments where vehicle behavior, sensor interactions, and system performance can be evaluated under a variety of conditions. Simulators can be broadly classified into two main categories: static and dynamic simulators, each serving distinct purposes in the validation pipeline. This section explores their functionalities, applications, and the importance of their integration into the ADAS development cycle (Dosovitskiy *et al.*, 2017a).

2.3.1 System Architecture

The architecture of a vehicle simulator is designed to replicate real-world driving conditions as closely as possible, enabling engineers and researchers to conduct tests in a safe and repeatable manner. A typical simulation environment consists of three main components: a simulation engine, responsible for physics calculations and virtual environment rendering; sensor models, which replicate real-world perception systems such as LiDAR, radar, and cameras; and control algorithms, which process sensor data and govern vehicle behavior (Behrisch *et al.*, 2011). These components work together to form a comprehensive testbed for evaluating different driving scenarios, ranging from basic lane-following tasks to complex urban interactions involving pedestrians and other vehicles.

Figure 12 – Radar and camera system



Source: Natour *et al.* (2015).

VI-Grade is a leading provider of static and dynamic vehicle simulators, designed for automotive system development, ADAS validation, and autonomous vehicle testing. Their simulators provide real-time driver-in-the-loop (DIL) experiences by integrating multi-sensor data, vehicle dynamics models, and environmental simulations to test ADAS and autonomous driving algorithms. VI-Grade offers two primary types of simulators:

2.3.1.1 Static and Dynamic Vehicle Simulators

Static vehicle simulators provide a controlled driving experience where drivers interact with a virtual environment while remaining stationary. These simulators are widely used in research and development for testing and validating Advanced Driver Assistance Systems (ADAS), studying Human-Machine Interface (HMI) interactions, analyzing driver behavior, and conducting Software-in-the-Loop (SIL) testing (Bengler *et al.*, 2014). Their fixed-platform nature makes them ideal for experiments that do not require physical motion feedback, allowing for a repeatable and controlled testing environment. Researchers use static simulators to evaluate how ADAS features, such as lane departure warning and collision avoidance systems, respond under different conditions without introducing the complexities of motion dynamics (Law, 2014).

In contrast, dynamic vehicle simulators incorporate motion platforms to simulate real-world vehicle forces, providing a more immersive and realistic driving experience. These simulators are essential for vehicle dynamics testing, Hardware-in-the-Loop (HIL) simulations, autonomous vehicle development, and real-time ADAS testing with sensor fusion. By replicating real-world driving conditions, dynamic simulators enable engineers to assess vehicle handling, stability, and performance under various road and environmental scenarios. This makes them particularly valuable for refining driver assistance technologies, such as adaptive cruise control and emergency braking, in a safe and controlled setting before real-world deployment (Peters *et al.*, 2018).

The architecture of VI-Grade simulators consists of several key subsystems that contribute to their realistic simulation capabilities. At the core of these simulators is the computing and simulation system, which serves as the computational backbone, running real-time vehicle models, physics simulations, and ADAS algorithms. The real-time vehicle dynamics model simulates vehicle responses based on driver inputs and external conditions, ensuring accurate behavioral replication. The ADAS and sensor fusion module integrates data from cameras, radar, LiDAR, and ultrasonic sensors, allowing for advanced perception and decision-making. A physics engine models acceleration, braking, tire-road interactions, and environmental effects, while the control system interface connects with embedded control units (ECUs) to enable real-time HIL testing.

To support these simulations, various software solutions are employed, including VI-CarRealTime for vehicle dynamics simulation, MATLAB/Simulink for control system modeling, Carla and CarSim or IPG CarMaker for environmental simulation. These tools enable researchers to develop and validate vehicle control strategies before real-world implementation, reducing the need for costly physical testing. The integration of these software solutions ensures that the simulated environment closely mirrors real-world driving conditions, enhancing the effectiveness of ADAS validation and autonomous vehicle research (Dosovitskiy *et al.*, 2017a; Rajamani, 2011).

Human-Machine Interface (HMI) and driver feedback systems play a crucial role in improving simulation realism. Steering wheel force feedback is implemented to replicate road feel, oversteer, and understeer, providing a tactile connection between the driver and the virtual vehicle. Pedal force replication adjusts braking and acceleration resistance based on vehicle dynamics, ensuring an accurate representation of driver input. Visual display systems include immersive 360-degree projections or virtual reality integration, creating a realistic driving experience. Additionally, driver monitoring systems track eye movement, fatigue levels, and reaction times to assess driver behavior and engagement with ADAS features.

Dynamic simulators incorporate advanced motion and actuator systems to provide a full range of motion feedback. These systems use some degrees of freedom kind of motion platforms that replicate real-world vehicle movements, enhancing the simulation experience. Linear actuators simulate pitch, roll, and yaw, replicating the effects of cornering, braking, and acceleration. Hydraulic or electric motion systems provide high-frequency feedback for analyzing vehicle handling characteristics. Seat and belt tensioner systems enhance the sensation of acceleration, braking, and lateral forces, improving the overall realism of the simulation.

One of the advanced dynamic vehicle simulators available today is the DiM150 (Driver-in-Motion 150) by VI-grade. The DiM150 employs a nine-degrees-of-freedom (9-DoF) motion platform, enabling researchers to replicate real-world vehicle behavior accurately. This simulator is used extensively for ADAS validation, driver training, and virtual prototyping, providing realistic motion feedback that enhances the fidelity of the simulation. By integrating hardware-in-the-loop (HIL) and software-in-the-loop (SIL) capabilities, the DiM150 bridges the gap between simulation-based testing and real-world validation, significantly improving the safety and efficiency of autonomous driving systems (Rodrigues *et al.*, 2021b).

Various motion system configurations exist, ranging from compact simulators with low degrees of freedom (DOF) and a small footprint to full-motion simulators with high DOF for large-scale, highly immersive setups. Compact simulators are suitable for early-stage development and limited-space environments, whereas full-motion simulators provide an unparalleled level of realism for comprehensive vehicle testing.

Sensor integration and environmental simulation modules further enhance the realism of driving scenarios, allowing for extensive ADAS and autonomous vehicle testing. These modules generate synthetic camera and LiDAR data for perception system validation, simulate radar signals for adaptive cruise control and collision avoidance testing, and create dynamic traffic and pedestrian interactions to assess real-world driving scenarios. Weather simulation capabilities adjust lighting, rain, fog, and road conditions dynamically, ensuring that ADAS functions are tested across diverse environmental conditions.

Key technologies used in these simulations include sensor fusion modules, which combine multiple sensor inputs to enhance ADAS validation, driving scenario libraries containing pre-built city, highway, and off-road environments, and traffic flow simulation models that generate realistic vehicle interactions in complex environments. These components collectively create a robust testing platform, enabling researchers and engineers to refine ADAS and autonomous vehicle technologies in a safe, controlled, and highly detailed simulation environment.

2.3.2 Test and Validation of Driving Assistance Features

Simulators play a crucial role in the development, testing, and validation of ADAS and autonomous vehicle technologies. They enable safe and controlled experimentation, reducing the risks associated with real-world testing while accelerating the development process. Some physical simulators can be categorized into static and dynamic simulators, each serving different purposes and requirements (Rajamani, 2011). Given the need for real-world validation of autonomous driving technologies, simulators contribute significantly to the iterative development cycle.

Static models provide an initial framework for evaluating individual system components, such as camera-based lane detection algorithms. Meanwhile, dynamic simulators facilitate comprehensive scenario testing, including the integration of 3D reconstructed environments for more immersive and realistic experiences (Dosovitskiy *et al.*, 2017a). These advanced simulations allow for extensive validation of ADAS functionalities like lane departure warning, adaptive cruise control, and pedestrian detection before real-world deployment.

2.3.3 Component and System Testing

Beyond ADAS validation, vehicle simulators are instrumental in testing individual components and entire vehicular systems. Sensor models, such as LiDAR, radar, and stereo cameras, can be accurately replicated in simulation environments, allowing engineers to fine-tune their performance under various conditions (Geiger; Lenz; Stiller, 2013). Real-time simulation of dynamic vehicle behavior is essential for validating autonomous decision-making algorithms. For instance, an autonomous emergency braking (AEB) system can be tested under simulated high-speed conditions to ensure fast and reliable braking responses without risking actual vehicle damage. Simulators also provide valuable insights into human-in-the-loop (HIL) interactions, helping researchers analyze driver reactions when using semi-autonomous systems (Dosovitskiy *et al.*, 2017a).

2.4 Conclusion

Static and dynamic simulation models play a important role in modern ADAS and autonomous vehicle research. While static simulators offer valuable insights into specific decision-making processes and system functionalities in a controlled, non-moving 3D environment, dynamic simulators enable real-time testing by simulating vehicle behavior under various road and traffic conditions with motion feedback, making them essential for realistic validation scenarios (Law, 2014; Peters *et al.*, 2018). The development of high-fidelity simulation platforms, such as DiM150 (Vi-grade, 2020), has empowered researchers to create realistic and scalable testing environments for autonomous vehicle validation. By integrating sensor models, reconstructed 3D environments, and advanced physics engines, these simulators provide a safe and efficient alternative to real-world testing, allowing for detailed evaluation and optimization of vehicle control algorithms, ADAS features, and human-machine interactions (Goodall, 2014; Rajamani, 2011). This research underscores the importance of simulation in advancing ADAS technologies, demonstrating how virtual environments contribute to the iterative development and refinement of autonomous driving systems. By leveraging both static and dynamic simulation models, researchers and automotive manufacturers can enhance safety, reliability, and performance, paving the way for more robust and adaptable autonomous vehicle technologies in the future.

3 METHODOLOGY

This chapter outlines the research methods and procedures employed to develop guidelines for the framework specifically tailored for 3D reconstruction and environment simulation using cameras. The chapter describes the qualitative approach adopted, detailing the research strategy concerning the literature review and case studies that supported this work. It covers the framework's development process, including data preprocessing, point cloud generation, mesh texturing, and post-processing evaluation for both reconstruction and simulation purposes. Finally, it concludes with the data collection process and the limitations encountered during the research. This comprehensive methodology ensures that the proposed framework is both theoretically grounded and practically applicable for real-world scenarios in 3D reconstruction and environment simulation.

3.1 Research Approach

To fulfill the objectives of this study, the workflow presented in table 1 was designed to provide a structured and systematic approach. The process begins with understanding the principles of data acquisition, followed by the selection of suitable sensors and the collection of required data. Subsequently, the focus shifts to exploring vehicle-specific simulators, leading to the selection of an appropriate simulation platform. Parallelly, the workflow includes studying methodologies for 3D reconstruction and selecting the most suitable approach. This enables the creation of a reconstructed dataset, which is then integrated with the chosen simulation software. This designed table in 1 ensures a comprehensive and methodical progression toward the research objectives.

The process begins with Understanding Data Acquisition (Step 01), which involves researching various data collection techniques and defining the necessary requirements for the project. This is followed by the Selection of Sensors (Step 02), where appropriate sensors are evaluated and chosen based on their suitability for capturing data relevant to the simulation objectives. Once the sensors are selected, the Data Acquisition phase (Step 03) is carried out, where the necessary data is collected to support the 3D reconstruction and simulation tasks. Next, the focus shifts to the simulation environment. In Understanding Simulators in the Vehicle Domain (Step 04), different simulators designed for vehicular applications are explored and analyzed to determine their capabilities. Following this analysis, Selection of Simulator (Step 05) is performed, where the most suitable simulation platform is chosen based on factors such as realism, integration capabilities, and support for autonomous driving research.

The next phase involves the study and implementation of 3D reconstruction methodologies. In Understanding 3D Reconstruction (Step 06), various methodologies, tools, and workflows are examined to assess their effectiveness in generating realistic 3D representations. Based on this analysis, the Selection of Reconstruction Methodology (Step 07) is made, en-

Frame 1 – Steps for Data Acquisition, Simulation, and 3D Reconstruction

Step	Title	Description
01	Understanding Data Acquisition	Research and analyze data acquisition techniques and requirements.
02	Selection of Sensors	Evaluate and choose appropriate sensors for the project.
03	Data Acquisition	Collect the required data using the selected sensors.
04	Understanding Simulators in the Vehicle Domain	Explore and study simulators specifically designed for vehicular applications.
05	Selection of Simulator	Choose the most suitable simulator for the study's objectives.
06	Understanding 3D Reconstruction	Investigate 3D reconstruction methodologies, tools, and workflows.
07	Selection of Reconstruction Methodology	Decide on the methodology to be used for 3D reconstruction.
08	Dataset Reconstruction	Apply the chosen methodology to reconstruct the dataset.
09	Integration of the Reconstruction with the Selected Simulation Software	Combine the reconstructed dataset with the selected simulation software.

Source: Own Work (2025).

suring that the most appropriate technique is chosen to create accurate and high-quality reconstructions. With the selected methodology, the Dataset Reconstruction phase (Step 08) is executed, applying the chosen reconstruction approach to generate the 3D models. In Integration of the Reconstruction with the Selected Simulation Software (Step 09), the incorporation of the reconstructed dataset into the selected simulator is studied, aiming to enable its use in virtual driving scenarios.

3.2 Systematic Literature Review

Systematic literature review (SLR) is a critical research methodology that ensures a comprehensive, unbiased, and transparent synthesis of existing knowledge on a specific topic. Unlike traditional literature reviews, an SLR follows a structured process, including clearly defined research questions, rigorous inclusion and exclusion criteria, and detailed documentation of the search and selection processes. This systematic approach minimizes the risk of bias and ensures that all relevant studies are considered, leading to reliable and replicable results. By identifying gaps in the existing literature, it provides a solid foundation for future research, fosters evidence-based decision-making, and supports the development of new theories and methodologies. It is an invaluable tool for consolidating knowledge in rapidly evolving fields, enabling researchers and practitioners to understand current trends, challenges, and best practices effectively. The meticulous nature of an SLR makes it a cornerstone for advancing academic rigor and practical applications in any discipline.

In this study, the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) (Page *et al.*, 2021) methodology was used. PRISMA is an evidence-based approach that provides a structured framework for identifying, selecting, evaluating, and synthesizing relevant research studies in a transparent and reproducible manner. It ensures that systematic reviews are comprehensive and unbiased by following a clear process that includes defining inclusion and exclusion criteria, systematically searching databases, screening articles, assessing their quality, and analyzing the extracted data. By adopting PRISMA, this research guarantees a rigorous and well-documented review of existing methodologies, tools, and techniques related to 3D reconstruction for autonomous driving simulations, ensuring the validity and reliability of the findings. PRISMA methodology and subdivided into four key phases:

1. **Identification:** This phase involved conducting a comprehensive search in selected databases, guided by research axes and predefined search terms outlined in the research protocol. This step ensured the retrieval of a broad range of potentially relevant studies.

2. **Selection:** Articles were filtered based on the PRISMA criteria, which included a review of titles and abstracts. This step assessed the alignment of each study with the research topic, ensuring only relevant studies proceeded to the next phase.

3. **Eligibility:** In this phase, a full-text review of the articles shortlisted during the selection stage was performed. Studies incompatible with the research focus were excluded, resulting in the design of a refined research portfolio containing only relevant and high-quality articles.

4. **Inclusion:** This final phase involved incorporating additional studies through qualitative analysis, further enriching the final research portfolio. These studies were critically analyzed to ensure they contributed meaningfully to the research objectives.

In this research, two systematic reviews were conducted to address the double requirement of the study: understanding the 3D reconstruction process and exploring the simulation process. The first review aimed to comprehensively analyze the methodologies, tools, and technologies employed in 3D reconstruction, including data acquisition, processing techniques, and rendering workflows. This review provided insights into the state-of-the-art practices and challenges in reconstructing accurate and detailed 3D environments. The second review focused on the simulation process, examining frameworks, algorithms, and applications used for simulating realistic environments and interactions. By exploring the key developments and limitations in simulation technologies, this review facilitated a deeper understanding of how simulated environments can complement 3D reconstructions. Together, these reviews established a foundational understanding of the two interconnected domains, enabling the integration of 3D reconstruction and simulation for more comprehensive and effective applications.

3.2.1 3D Reconstruction Approach

To conduct a comprehensive survey of techniques for 3D point cloud generation and mesh texturing, a SLR was carried out following the PRISMA methodology (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) (Page *et al.*, 2021). This methodology allowed to identify studies congruent with the topic analyzed through a systematic search in pre-selected databases following some relevance criteria. First, it was necessary to determine the keywords related to the topic and define the search criteria in the selected databases. Table I exemplifies the information regarding the research protocol used, such as the indication of its central axis of interest, the systematic literature review method used, the nature of the researched studies, possible language restriction, and the operated database.

Table 1 – Search Axes for 3D Reconstruction.

Search Axes	Temporal Cut	IEEE Xplore	ACM	MDPI	ArXiv	Total
(“3D City”) AND (“Reconstruction from images” OR “Reconstruction from video” OR “Video-Based Reconstruction” OR “Image reconstruction”) NOT (“Microscopy” OR “Biology” OR “X-ray imaging” OR “medical image processing” OR “Indoor” OR “single area” OR “bridge”)	2010 - 2024	528	01	67	23	619
(“3D City” OR “Urban 3D Modeling” OR “Cityscape 3D Generation”) AND (“Ground Video”) OR (“Reconstruction from images” OR “Reconstruction from video” OR “Video-Based Reconstruction” OR “2D images” OR “Image reconstruction”) AND (“Autonomous Vehicle*”) NOT (“aerial images” OR “Drone”) NOT (“Lidar”) NOT (“Sonar” OR “Underwater”)	-	536	10	0	0	546

Source: Own work (2025).

The frame 1 presents the search axes used for identifying relevant studies in the context of 3D city reconstruction, detailing the query structure, temporal constraints, and results obtained from various databases. Two distinct search axes were employed. The first axis focuses on reconstructing 3D cities from images or videos, excluding topics unrelated to the study, such as microscopy, medical imaging, and specific domains like indoor or bridge reconstruction. This search was limited to the period from 2010 to 2023 and yielded a total of 619 results, with contributions from IEEE Xplore (528), ACM (1), MDPI (67), and ArXiv (23). The second axis broadens the scope to include urban 3D modeling and cityscape generation, emphasizing ground-based video and images in the context of autonomous vehicles, while excluding aerial images, drone-based reconstruction, and sensor technologies like LiDAR and sonar. This query was not temporally restricted and resulted in 546 studies, primarily from IEEE Xplore (536) and ACM (10). Together, these search axes comprehensively capture literature on 3D reconstruction methods relevant to urban modeling and autonomous systems.

Frame 2 – Search Protocol for 3D Reconstruction.

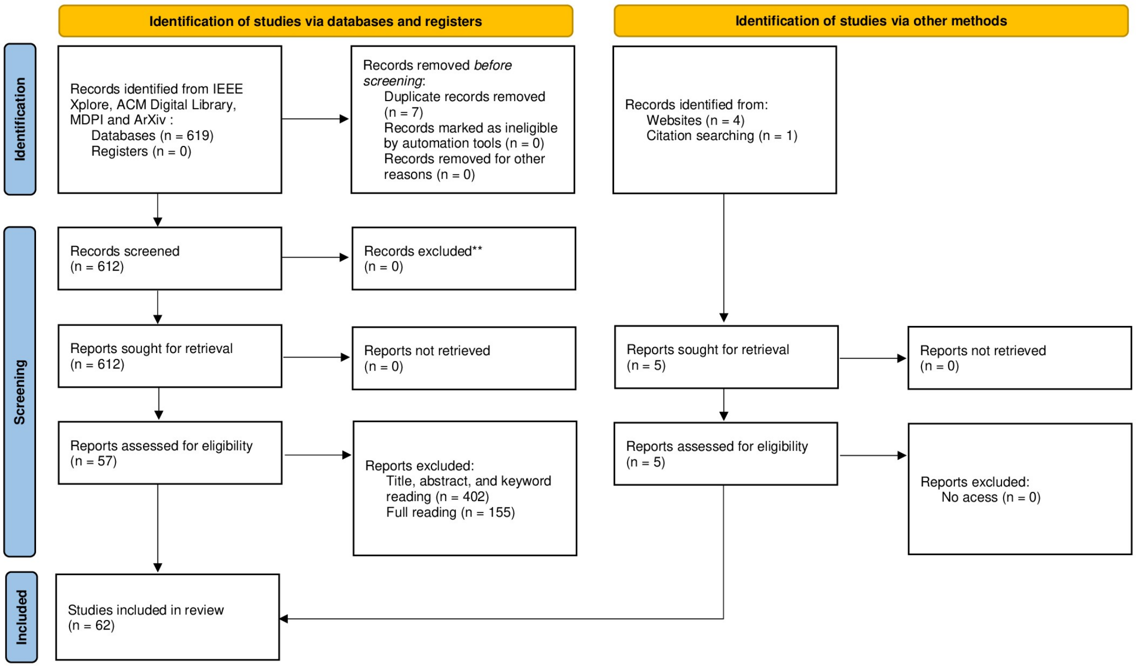
Direction	**Protocol**
Search Aim	(“3D City”) AND (“Reconstruction from images” OR “Reconstruction from video” OR “Video-Based Reconstruction” OR “Image reconstruction”) NOT (“Microscopy” OR “Biology” OR “X-ray imaging” OR “Medical image processing” OR “Indoor” OR “Single area” OR “Bridge”)
Search Strategy	Utilization of logical operators (AND, OR) to connect search terms
Databases	IEEE Xplore, ACM Digital Library, MDPI, and ArXiv
Publication Types	Articles, Review Articles, and Conference Papers
Language	English
Search Period	2010 - 2024

Source: Own Work (2025).

Several databases were tested to find relevant studies for this research. The selection process prioritized databases that returned the highest number of results related to the study’s objectives. Initially, Web of Science was considered; however, when applying the search criteria related to the research focus, only a single document was found. As a result, alternative databases were explored to ensure a broader and more comprehensive collection of sources. One of the main challenges encountered during the literature review was the difficulty in finding studies that fully integrated all aspects of the research. Most of the retrieved papers focused on specific subtopics within the broader domain rather than addressing the complete scope of 3D reconstruction, simulation, and integration in a unified framework. This required a careful selection and combination of different sources to build a more comprehensive understanding of the field.

The search terms were defined according to the different synonyms of the concept of 3D point cloud generation and mesh texturing observed in the literature, and their variations were included in the searches using the Boolean operator “OR”. The search terms were used in English to involve a broader range of studies. This systematic literature review applied a temporal cut from 2010 to 2024. The reason behind this cut-off was the unavailability or obsolescence of many of the methods found before 2010, rendering them unsuitable for the present study. Therefore, the selected timeframe ensured that the most relevant and up-to-date methodologies were included in the analysis, optimizing the effectiveness and applicability of the study. In addition to the selected search axes, attempts were made to include additional keywords and search criteria to ensure comprehensive coverage of relevant literature. Despite multiple attempts with alternative axes, the results did not significantly contribute to the systematic literature review. When subjected to PRISMA's second phase, only a few articles remained. Therefore, the axe mentioned in Table 2 was considered the most suitable for this study, providing a focused and relevant dataset for the analysis. Subsequently, the present research conducted a systematic review of studies identified through the PRISMA methodology, following its structured information flow.

Figure 13 – Development of PRISMA methodology steps



Source: Own Work (2025).

All stages of the systematic literature review were supported by the Mendeley software, characterized as a computational tool for managing references. Therefore, 39 duplicates were identified and discarded from the analysis. Figure 13 represents all the steps needed to develop the methodology applied to the articles obtained, from its initial search in the databases to the selection of the final portfolio. Once all the steps described in the method were carried out, a final portfolio was obtained consisting of 11 articles, which were submitted for further analysis, thus being able to contribute to the objective of the research on the study of the topic. Figure 13 illustrates the PRISMA flow diagram used in this research, detailing the different phases involved in the identification, screening, eligibility assessment, and final inclusion of studies.

The identification phase involved searching for records in multiple databases, including IEEE Xplore, ACM Digital Library, MDPI, and ArXiv, which initially yielded 619 results. After removing duplicate entries (n=7), 612 records were screened based on their title, abstract, and keywords to assess their relevance to the study. At this stage, 402 studies were excluded for not aligning with the research objectives. The screening phase consisted of retrieving the full text of the remaining 210 articles. However, after a complete reading, 155 studies were deemed irrelevant and excluded. Additional sources were identified through website searches and citation tracking, contributing 5 extra articles, all of which passed the eligibility criteria. 62 studies were included in the final literature review. These articles formed the basis for analyzing 3D reconstruction methodologies, simulation techniques, and integration approaches used in autonomous vehicle research.

3.2.2 Simulation Approach

Aiming to explore the methodologies and tools related to the simulation of autonomous and connected vehicles, this section was developed following a methodology similar to that used in the Reconstruction Approach section, employing systematic searches across multiple databases to ensure comprehensive coverage of relevant studies. The approach focuses on identifying simulation frameworks, driving simulators, and testing environments that support the integration of 3D reconstructed datasets into virtual scenarios. By leveraging a structured and methodical process, this section aims to establish a robust foundation for the selection and implementation of simulation tools, aligning with the overall objectives of the study.

Table 2 presents the search axes used to identify relevant studies related to vehicle simulation, following a methodology similar to that described in the Reconstruction section. The table outlines the queries, temporal constraints, and results obtained from multiple databases, emphasizing their alignment with the study's focus on autonomous and connected vehicles. The first axis focuses on simulation approaches for connected and autonomous vehicles, particularly those involving ADAS, yielding a total of 205 results from IEEE Xplore, ACM, and MDPI. The second axis refines the scope by targeting ADAS-specific simulation studies for autonomous vehicles, resulting in 68 relevant studies. The third axis broadens the query to include simulator testing and comparisons for automated driving, generating the most extensive results with 549 studies. The fourth axis emphasizes virtual environments and testing platforms for autonomous vehicles, contributing 116 studies. This systematic search strategy ensures comprehensive coverage of simulation methodologies, aligning with the research objectives and providing a solid foundation for the simulation framework.

Frame 3 outlines the search protocol established for identifying studies related to simulation in the context of autonomous and self-driving vehicles. The table provides a detailed overview of the systematic approach employed to ensure comprehensive and relevant results.

Table 2 – Search Axes for Simulation.

Search Axes	Temporal Cut	IEEE Xplore	ACM	MDPI	Total
("Connected Vehicles" OR "Autonomous Vehicles" OR "Connected Autonomous Vehicles" OR "Self-Driving cars") AND ("Simulation" OR "Driving simulator" OR "Vehicle simulation" OR "Driving simulation environment" OR "Simulated driving") AND ("ADAS" OR "Advanced Driver Assistance Systems")	2020-2024	81	124	0	205
("Autonomous Vehicles" OR "Self-Driving cars") AND ("Simulation" OR "Driving simulator" OR "Vehicle simulation" OR "Driving simulation environment" OR "Simulated driving") AND ("ADAS" OR "Advanced Driver Assistance Systems")	2020-2024	68	0	0	68
("Autonomous Vehicles" OR "Self-Driving Cars" OR "Automated Driving") AND ("Driving Simulator" OR "Vehicle Simulation" OR "Driving Simulation Environment" OR "Simulated Driving" OR "Simulator Comparison" OR "Open Source Simulator" OR "Simulation testing")	2020-2024	239	307	4	549
("Autonomous Vehicles" OR "Self-Driving Cars" OR "Automated Driving") AND ("Driving simulator" OR "Vehicle simulation" OR "Driving simulation environment" OR "Simulated Driving" OR "Simulator Comparison" OR "Open Source Simulator" OR "Simulation testing") AND ("Virtual environment" OR "Simulated Environments" OR "Testing Environments")	2020-2024	12	104	0	116

Source: Own work (2025).

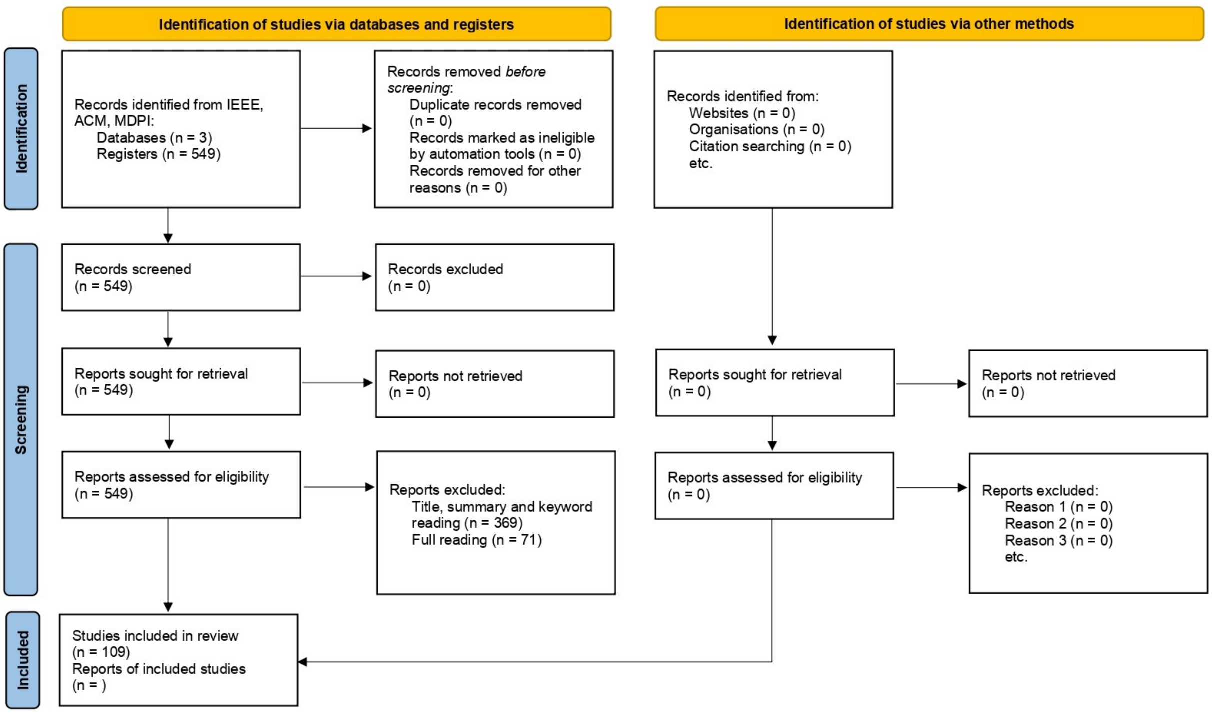
The PRISMA methodology, as previously explained, was applied to the simulation approach to systematically identify and select relevant studies. Using this structured process, studies were screened, evaluated, and included based on predefined criteria, focusing on simulation methodologies and tools related to autonomous vehicles. As a result of this application, the workflow illustrated in Figure 14 was generated, detailing the steps and outcomes of the systematic review in the context of the simulation approach.

Frame 3 – Search Protocol for Simulation.

Direction	**Protocol**
Search Aim	("Autonomous Vehicles" OR "Self-Driving Cars" OR "Automated Driving") AND ("Driving Simulator" OR "Vehicle Simulation" OR "Driving Simulation Environment" OR "Simulated Driving" OR "Simulator Comparison" OR "Open Source Simulator" OR "Simulation testing")
Search Strategy	Utilization of logical operators (AND, OR) to connect search terms
Databases	IEEE Xplore, ACM Digital Library and MDPI
Publication Types	Articles, Review Articles, and Conference Papers
Language	English
Search Period	2020 - 2024

Source: Own work (2025).

Figure 14 – Development of PRISMA methodology steps



Source: Own work (2025).

3.3 Content Analysis

In addition to the systematic literature review, content analysis was conducted using NVIVO software. This allowed identify several themes related to the 3D point cloud generation techniques and mesh texturing. Manual nodes were generated with NVIVO version 10. This method facilitated a comprehensive exploration of the subject. This allowed identify several themes related to the techniques, enabling a deeper understanding of the topic. By utilizing NVivo, a comprehensive exploration of the subject was facilitated. The images presented in Frames 4, 5, 6 and 7 were obtained from the nodes in NVivo, visually representing the relationship between the techniques mentioned.

Frames 4, 5, 6 provides an organized summary of datasets, equations, and introductory nodes used in the study, categorized into three main sections: Datasets, Equations, and Introduction. Each category highlights key resources, methods, and technologies essential to the research, along with their sources and references. In the Datasets category, datasets like KITTI, HoliCity, and NYUv2 are listed for their contribution to 3D reconstruction and simulation, focusing on urban modeling and object detection. The "Sources" column indicates how many studies utilized each dataset, while the "Refs" column shows the number of references. For example, the KITTI dataset has three sources and four references, emphasizing its significance.

Frame 4 – Datasets

Category	**Node • Subnode**	**Sources**	**Refs**
Datasets	City Intrinsic Images dataset	1	2
	DDAD	1	1
	DTU	1	1
	Geoportal data	1	1
	HoliCity	1	4
	KITTI	3	4
	MegaDepth	1	1
	scenes	1	1
	NYUv2	1	1
	Open streets map	1	1
	RobotCar	1	1
	ScanNet	2	2
	Shapenet	1	2
	Stanford-2D-3D	1	1
	SUNCG	1	1
	SYNTHIA	1	2
	WikiScenes	1	1

Source: Own Work (2025).

The Equations category includes mathematical models and algorithms critical for processing spatial and structural data. Techniques such as Chamfer Distance, Earth Mover's Distance, and Epipolar Constraint are essential for aligning datasets and optimizing the reconstruction process. Each equation is accompanied by its corresponding sources and references, in-

Frame 5 – Equations

Category	Node • Subnode	Sources	Refs
Equations	Bruijn sequence	1	1
	Chamfer Distance (CD)	1	1
	Cluster	1	4
	Earth Mover's Distance (EMD)	1	1
	Epipolar constraint	1	2
	Gabor filter	1	2
	Gaussian filter	1	1
	Integer least square	1	1
	Integer Linear Programming (ILP)	1	1
	MRF (Markov Random Field)	1	1
	Projector	1	2
	Triangulation	1	1
	TSDF	2	2

Source: Own Work (2025).

Frame 6 – Sensors

Category	Node • Subnode	Sources	Refs
Sensors	2D laser sensor	1	2
	Aerial Laser Scanning (ALS)	1	2
	Digital camera	8	11
	Google Earth	2	4
	Hand Held camera	2	2
	Laser	1	2
	Laser scanners	1	3
	Lidar	7	13
	Mobile Laser Scanning (MLS)	1	1
	Satellite	4	5
	Sensor fusion	2	2
	Truck	1	2

Source: Own Work (2025).

dicating its application within the study. The Sensors Nodes section details key technologies and tools like Lidar, digital cameras, and laser scanners, which are vital for data acquisition. Other nodes, such as Google Earth and Mobile Laser Scanning, illustrate the variety of methods employed for gathering and integrating data in the study. The "Lidar" node, for instance, is referenced seven times, highlighting its central role.

Frame 7 – Methods subnodes.

Category	Node • Subnode	Sources	Refs
Geometry extraction	Geometry extraction	3	4
	Agisoft Photoscan	1	1
	ArcGIS Pro	1	2
	Coordinate transformation	1	1
	LAStools	1	1
	MicMac	1	1
	Multi-view stereo	1	1
	PolyFit	1	1
	VB3D aerial multiview stereo	1	1
Mesh reconstruction	Mesh Reconstruction	0	0
	Ball-pivoting	2	10
	Bundle adjustment	6	14
	Deep learning	8	12
	Digital Surface Map (DSM)	2	11
	Gaussian	8	23
	Greedy triangulation	1	6
	Multi-View Stereo	12	15
	NeRF (Neural Radiance Fields)	2	5
	Other algorithms	12	27
	Poisson Reconstruction	8	14
	Surface reconstruction	2	8
Other	Other	27	66
Point cloud	Point cloud	3	4
	Multi-view Geometry	2	2
	Neural Network	2	2
	Point Cloud Library (PCL)	1	1
	Render-and-compare	1	8
	Structure from motion	18	33
Post processing	Post-processing	13	19

Source: Own Work (2025).

Frame 7 provides an overview of methods and subnodes categorized into key areas relevant to the study: Geometry Extraction, Mesh Reconstruction, Point Cloud, and Post-Processing. Each category highlights specific techniques, tools, and algorithms employed in 3D reconstruction processes, along with their associated sources and references. The Geometry Extraction category focuses on methods for extracting spatial and structural information from data. Notable techniques include Agisoft Photoscan, ArcGIS Pro, and Multi-View Stereo, each contributing to the generation of accurate geometric representations. For instance, Geometry Extraction has three sources and four references, reflecting its foundational role in the process.

The Mesh Reconstruction category outlines methods for generating 3D surface models. Techniques such as Ball-pivoting, Bundle Adjustment, and Poisson Reconstruction are central to this category, with high citation frequencies (e.g., Bundle Adjustment has six sources and 14 references). Additionally, innovative approaches like Deep Learning and Neural Radiance Fields (NeRF) highlight advancements in reconstruction algorithms, showcasing their growing significance in the field. In the Point Cloud category, methods like Structure from Motion (with 18 sources and 33 references) and the Point Cloud Library (PCL) are integral to generating and processing 3D point data. These approaches are pivotal for accurately capturing and representing complex spatial environments, bridging the gap between raw data and final models. The Post-Processing category focuses on refining and optimizing 3D models. Post-processing, with 13 sources and 19 references, ensures that reconstructed models meet accuracy and quality standards required for further applications.

Table 3 – Simulators and their corresponding sources and references.

Simulator	Sources	Refs
CARLA	4	6
SUMO	3	5
Custom-built	4	4
TORCS	1	1
Matlab	1	1
Carsim	1	1
SCANNER	1	1
LGSVL	2	2
Autoware	1	1
CarMaker	1	1
GTA5	2	2
APOLLO	2	2

Source: Own Work (2025).

Table 3 presents an overview of the simulators used in the study, along with the number of sources and references associated with each. The table highlights the diversity of simulation tools utilized in research related to autonomous and connected vehicles. CARLA emerges as the most frequently mentioned simulator, with 4 sources and 6 references, showcasing its popularity and robustness in the field. SUMO and a custom-built simulator also appear frequently, each with multiple sources and references, emphasizing their significance in specific simulation contexts.

Other simulators, such as TORCS, MATLAB, Carsim, and SCANNER, have limited mentions, indicating niche applications or specialized use cases. Meanwhile, LGSVL, Autoware, CarMaker, GTA5, and APOLLO also contribute to the study, each having 1-2 sources and references, demonstrating the variety of tools available for different simulation needs. This table underscores the breadth of simulation technologies employed, reflecting their adaptability to various research objectives and scenarios. The inclusion of both well-established and less common simulators highlights the comprehensive approach taken to ensure a diverse and well-rounded analysis in the study.

3.4 Data Collection

This section outlines the processes and considerations involved in acquiring the data necessary for this study. It begins with the Sensor Evaluation and Selection subsection, which details the criteria and methodology used to assess and choose the most suitable sensors for capturing the required data, ensuring compatibility with the objectives and the simulation workflow. The subsequent subsection, Dataset "Road", describes the collection and characteristics of a dataset focused on road environments, emphasizing the data's structure, resolution, and relevance to 3D reconstruction and simulation. Finally, the Dataset "UTFPR" subsection highlights a second dataset acquired from the UTFPR campus, providing complementary data to enhance the study's scope and robustness.

3.4.1 Sensor Evaluation and Selection

Based on the results presented in Frame 6, the methodology selected for this study was "Hand-held camera" due to the ease of obtaining the dataset. This approach allowed for a more accessible and straightforward data acquisition process, ensuring the project's feasibility within the given constraints. However, for future work, the use of sensor fusion is proposed to enhance the quality and accuracy of the datasets. Sensor fusion, combining data from multiple sensors, would provide more comprehensive and robust results, enabling more advanced 3D reconstruction and simulation capabilities.

The use of a hand-held camera in this reconstruction work offers several benefits, making it a practical and efficient choice for data acquisition. Cameras are highly accessible and cost-effective, eliminating the need for specialized or expensive equipment. They provide flexibility during data collection, allowing operators to capture images from diverse angles and perspectives, ensuring comprehensive coverage of the environment. The portability of hand-held cameras makes them ideal for use in various settings, including indoor and outdoor environments, where other equipment might face logistical challenges.

The straightforward operation of these cameras reduces the complexity of the data acquisition process, enabling rapid deployment and adaptation to different scenarios. Furthermore, modern cameras often include high-resolution imaging capabilities, ensuring sufficient detail for accurate 3D reconstruction. These advantages make this strategy a versatile and practical choice for projects focused on efficient and scalable 3D reconstruction.

3.4.2 Dataset "UTFPR"

The dataset for this study, referred to as the UTFPR dataset, was generated using the selected methodology of a hand-held camera. This approach allowed for the capture of 177 high-resolution images of the UTFPR campus, ensuring comprehensive coverage of the target environment. The hand-held camera's flexibility enabled the collection of data from various angles and perspectives, which is critical for accurate 3D reconstruction. The dataset focuses on key architectural and spatial features of the campus, providing a robust foundation for testing and validating the reconstruction and simulation workflows. This collection represents an accessible and practical starting point, paving the way for further enhancements in future studies, such as the integration of additional data sources through sensor fusion. Figure 15 presents a selection of images captured for the dataset, showcasing the diversity and quality of the visual data used in the study.

Figure 15 – Dataset UTFPR.

Source: Own Work (2025).

Subsequently, a second partial dataset was generated from the original UTFPR dataset to facilitate quick testing and iterative evaluations. This partial dataset consists of 45 images carefully selected from the initial collection, ensuring sufficient coverage of key areas while reducing processing time. The smaller dataset was specifically designed for rapid testing of the reconstruction and simulation processes, enabling efficient validation of methodologies and workflows without the need to process the entire dataset. This approach ensures a balance between accuracy and computational efficiency during preliminary tests.

3.5 Integration of Proposed Methods

The methodological structure described in this chapter is built upon a solid foundation of systematic research, ensuring that each phase aligns with state-of-the-art techniques in 3D reconstruction and simulation. As introduced earlier, the PRISMA methodology was used to conduct a rigorous systematic literature review, guiding the selection of studies that contributed to defining the best practices and methodologies adopted in this work. Figure 13 and 14 illustrates the PRISMA flowchart, outlining the steps taken to refine the literature search and select relevant studies.

This process was essential for ensuring that the workflow proposed in Table 1 is based on validated methodologies and proven techniques. The identification, screening, and eligibility phases allowed for a careful selection of high-impact research papers, ensuring that the integration of methods presented in the research is supported by well-established knowledge. By following this systematic approach, the methodology ensures that the integration between data acquisition, 3D reconstruction, and simulation is well-grounded in previous research. The insights gained from the literature review were crucial in determining the most suitable sensors, reconstruction techniques, and simulation platforms, reinforcing the structured and coherent workflow proposed in this study.

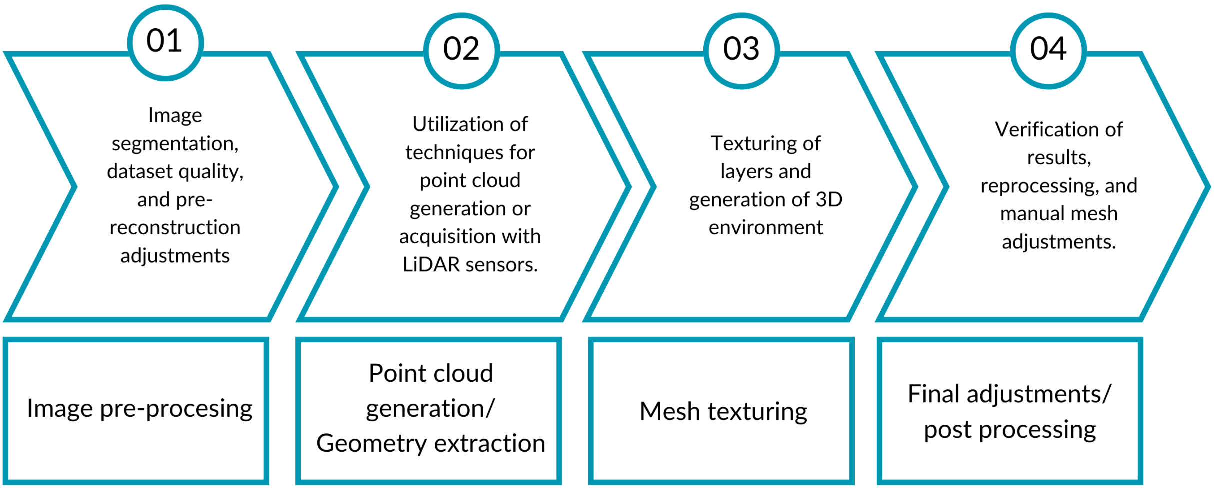
3.5.1 3D Reconstruction process

The 3D reconstruction process is a multi-stage workflow that ensures the creation of accurate and detailed models. It begins with image pre-processing to prepare the input data, followed by point cloud generation to extract spatial geometry. Next, mesh texturing adds realistic surface details, enhancing the model's visual fidelity. The final stage involves adjustments and post-processing to refine the reconstruction for practical applications. Figure 16 illustrates this process in a clear and systematic flowchart.

The first step involves image segmentation, dataset quality evaluation, and pre-reconstruction adjustments. This step ensures that the input data is clean, accurate, and suitable for further processing. Image segmentation separates the relevant features from the background, making it easier to extract meaningful geometry. Quality checks ensure that datasets have minimal noise and artifacts, reducing potential errors during reconstruction. Additionally, pre-reconstruction adjustments such as alignment and normalization of images prepare the data for optimal performance in subsequent stages. This may include noise reduction, image alignment, and feature extraction to ensure accurate reconstruction (Ballabeni *et al.*, 2015; Calantropio *et al.*, 2020).

In the second step, techniques for generating or acquiring point clouds are applied. For projects utilizing LiDAR sensors, the data acquisition process involves scanning the environment to produce dense 3D point clouds. These point clouds represent the spatial geometry of objects

Figure 16 – Flowchart of the 3D Reconstruction Process.



Source: Own Work (2025).

and environments with high precision. Alternatively, geometry extraction can be performed using image-based techniques, where photogrammetry or stereo vision methods are employed to create 3D structures from 2D images. This stage lays the groundwork for building a detailed and accurate 3D model. Techniques, such as stereo-matching (Hu *et al.*, 2022), structure from motion (Pylvänäinen *et al.*, 2012), and LiDAR scanning (Ledoux *et al.*, 2021), are commonly used to derive these point clouds from the input data.

Once the point cloud is generated, the mesh texturing phase adds surface texture information to the point cloud, creating a visually realistic representation of the scene (Hu *et al.*, 2022; Han; Shen, 2019; Müller *et al.*, 2022; Tancik *et al.*, 2023). Texturing involves overlaying photometric data, such as color or surface details, onto the geometric structure to produce a visually realistic model. This step enhances the model's fidelity by adding surface properties that accurately reflect the real-world appearance. The generation of the 3D environment involves assembling layers and integrating the textured model into a virtual or augmented reality context, creating a complete scene for visualization or analysis.

The final step includes verification of results, reprocessing, and manual adjustments to refine the 3D model. Quality checks ensure that the reconstructed model is free of errors, with all features accurately represented. If discrepancies are identified, reprocessing techniques are applied to address them. Manual adjustments, such as mesh smoothing, hole filling, or edge refinements, ensure that the model meets the desired standards of accuracy and visual quality. This stage is critical for preparing the model for practical applications, such as simulations, analyses, or presentations. Adjustments are made to refine the reconstructed model, ensuring accuracy and consistency (Xu; Wang; An, 2014; Buyukdemircioglu; Kocaman, 2020). This may involve geometric corrections, texture blending, or fine-tuning of parameters to improve the overall quality of the reconstruction. (Müller *et al.*, 2022; Buyukdemircioglu; Kocaman, 2020).

4 FRAMEWORK DEVELOPMENT

This chapter outlines the process of framework development by detailing the study of reconstruction and simulation methodologies. Starting with the selected methodology, a systematic approach was employed to analyze various techniques individually. Each methodology was evaluated based on key criteria, including publication dates, recent updates, alignment with state-of-the-art practices, and overall suitability for the study's objectives. This thorough examination ensured that the selected methods were not only current but also aligned with best practices in the field. The chapter concludes with the definition of the final methodologies for both reconstruction and simulation, providing a solid foundation for the integration of these processes into the overall framework. This structured and detailed approach guarantees the reliability and relevance of the developed framework.

After conducting a content analysis of the NVIVO nodes, all identified 3D reconstruction technologies and applications were categorized and organized into three distinct tables. These tables provide a comprehensive overview of the methodologies and tools used in different stages of the reconstruction process. The first table focuses on methodologies and applications specifically related to point cloud generation, detailing tools and techniques employed for extracting spatial data. The second table addresses mesh generation, listing tools used to create 3D surface models from point cloud data. Finally, the third table, categorized as Other, includes applications for post-processing reconstructions, as well as tools for evaluating and analyzing the results of 3D reconstruction processes. This organization ensures a clear and systematic presentation of the technologies, facilitating their understanding and comparison.

4.1 Point Cloud Generation

Frames 8 and 9 provides an extensive overview of various tools and techniques used for 3D reconstruction, detailing their characteristics, methodologies, and updates. Each row represents a specific tool or framework, highlighting its contributions to the field and the approaches it employs for generating 3D models. It includes key information for each tool, such as its name, the number of references citing it, the year of creation, the last update, the techniques used, and a brief description of its methodology. For instance, Meshroom, created in 2018 and updated in 2024, leverages Structure-from-Motion (SfM), SIFT, Multi-View Stereo (MVS), and PMVS/CMVS techniques for reconstruction, providing both a user-friendly interface and advanced customization options through its integration with AliceVision. Similarly, ColMap, a widely cited tool, combines SfM and MVS to deliver precise and detailed reconstructions.

The table also includes advanced methodologies such as NeRF (Neural Radiance Fields) used in Nerfstudio, which employs neural networks to synthesize novel views, showcasing the integration of artificial intelligence in modern reconstruction processes. Tools like MVS-Net and 3D-ReConstNet further emphasize the role of deep learning, employing neural architectures

Frame 8 – Tools and Techniques for 3D Reconstruction

Name	Refs	Year	Last Update	Techniques Used	Technique Description
Meshroom	1	2018	2024	SFM, SIFT, MVS, PMVS/CMVS	Based on the AliceVision pipeline, Meshroom offers an easy-to-use graphical interface for 3D reconstruction. For more advanced customization, AliceVision can be used directly. Both tools can be combined for specific tasks or advanced features.
ColMap	8	2016	2024	SFM, MVS	Combines Structure-from-Motion (SfM) and Multi-View Stereo (MVS) for accurate 3D reconstruction.
MicMac Photogrammetry Software	1	2003	2024	SFM, MVSM	Implements Structure-from-Motion (SfM) and Multi-View Stereo Image Matching (MVSM) for photogrammetry.
VisualSFM	1	2010	?	SFM, SIFT, PMVS/CMVS	A GUI-based application for 3D reconstruction using Structure-from-Motion (SfM). It integrates SIFT on GPU, Multicore Bundle Adjustment, and incremental SfM techniques. It supports dense reconstruction using PMVS/CMVS. Outputs are compatible with tools like CMP-MVS, MVE, and MeshRecon.
Nerfstudio	4	2023	2024	NeRF	Neural Radiance Fields (NeRF) focuses on 3D reconstruction using neural networks to synthesize novel views.
Point Cloud Library (PCL)	27	2011	2024	Various methods	Offers an extensive set of algorithms and tools for point cloud processing, covering segmentation, filtering, and visualization tasks.

Source: Own Work (2025).

for depth inference and single-view reconstruction. Other entries, such as Pix4Dmapper and

Frame 9 – Tools and Techniques for 3D Reconstruction 2

Name	Refs	Year	Last Update	Techniques Used	Technique Description
OpenMVG/ OpenMVS	2	2012	2023	SfM, MVG	OpenMVG (Open Multiple View Geometry) specializes in Multi-View Geometry and Structure-from-Motion (SfM). OpenMVS is tailored for dense 3D reconstruction.
AliceVision	1	2018	2024	Various techniques	Features cutting-edge algorithms for photogrammetry, feature matching, dense 3D reconstruction, camera tracking, and Structure-from-Motion (SfM).
PMVS/CMVS	3	2011	2019	PMVS, CMVS, MVS	PMVS (Patch-based Multi-View Stereo) and CMVS (Clustering for Multi-View Stereo) are used for dense 3D reconstruction in photogrammetry and computer vision pipelines.
Gipuma	1	2015	2022	MVS, MVG, Photogrammetry	Implements Multi-View Stereo (MVS), Multi-View Geometry (MVG), and Photogrammetry for accurate 3D reconstruction.
MVS-Net	3	2019	2020	Deep Learning, MVS	Uses neural networks for depth map inference from unstructured multi-view images. Includes the extended R-MVSNet method.
Pix4Dmapper	2	2011	2023	Various techniques	Provides professional photogrammetry software for generating 3D models and maps from images.
Agisoft Photoscan (Metashape)	1	2010	2024	Various techniques	A professional tool for photogrammetry and 3D reconstruction from images.

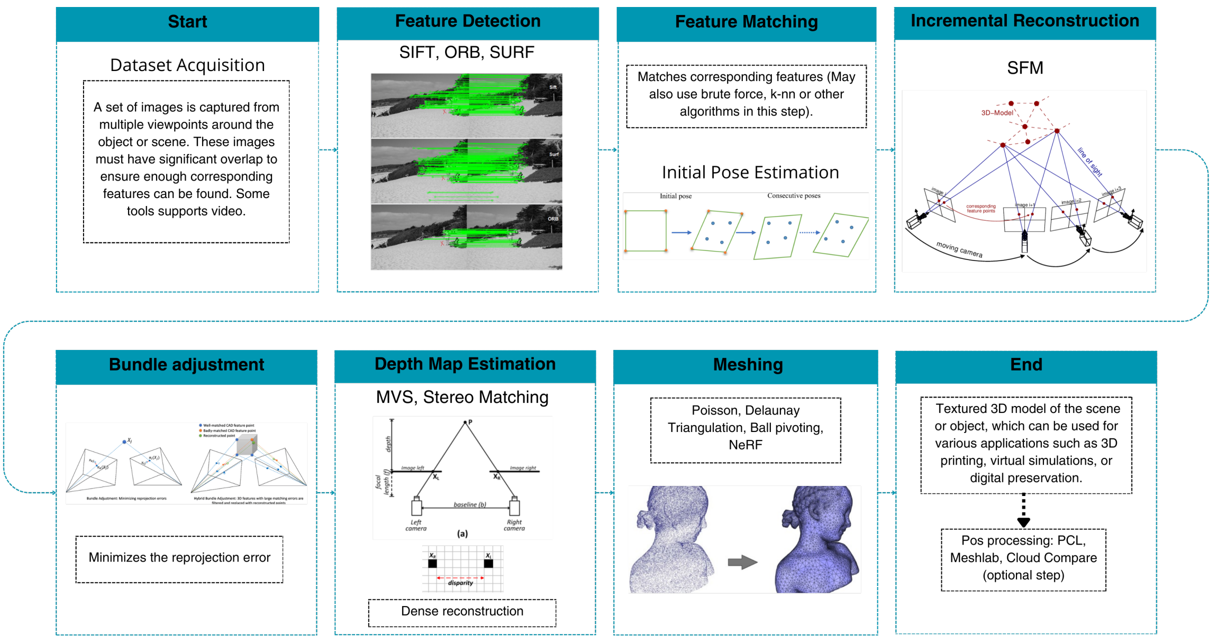
Source: Own Work (2025).

RealityCapture, highlight professional photogrammetry software tailored for industry use, offering high-quality results for mapping and modeling applications. Additionally, versatile libraries like the Point Cloud Library (PCL) stand out for their wide range of algorithms supporting point cloud processing tasks, including segmentation and visualization.

4.1.1 Structure from motion

Structure-from-motion (SfM) is a pivotal technique in computer vision for reconstructing 3D scenes from 2D images. As proposed by (Untzelmann *et al.*, 2013) and (Cheng *et al.*, 2016), SfM algorithms leverage multiple photos of a scene taken from different viewpoints to estimate the 3D structure and camera poses. It involves critical steps, such as feature extraction, matching, and triangulation, as outlined in (Muñoz-silva *et al.*, 2021). Notably, recent advancements in SfM have enabled its application to large datasets containing millions of images (Untzelmann *et al.*, 2013). They have paved the way for efficient reconstruction of entire cities, overcoming challenges related to data acquisition and processing (Zhang; Jiang; Ai, 2015). SfM techniques have been integrated with deep learning methods, as demonstrated in (Ma *et al.*, 2022), to enhance feature detection and reconstruction accuracy, particularly in hyperspectral imaging scenarios.

Figure 17 – Main steps in Sfm digitalization process.



Source: Own Work (2025).

The pipeline comprises several vital steps that collectively contribute to accurately reconstructing 3D scenes from 2D images. Initially, feature extraction is performed to identify distinctive points within each image, utilizing some algorithms, such as SIFT or SURF (Cheng *et al.*, 2016; Muñoz-silva *et al.*, 2021; Zhang; Jiang; Ai, 2015). Then, feature matching is conducted to establish correspondences between features across multiple images. These correspondences are crucial for the subsequent step of 3D reconstruction, where the initial 3D structure and camera positions are estimated using techniques like triangulation. To refine these estimations, bundle adjustment is employed, optimizing both the 3D points and camera parameters to minimize the reprojection errors. Optionally, the process may include meshing and texturing from the point cloud, which involves generating a surface mesh and applying textures to create a realistic and visually compelling 3D model. An explanatory flowchart of the SfM process is provided in Figure 17, illustrating the main steps involved in this robust 3D reconstruction technique. This figure is based on the work of (Bouain *et al.*, 2018; Riel, 2016; Yu *et al.*, 2022; Al-zoube, 2022; Youssef; Shehaby; Fayed, 2020; Cgal. . . ,).

Structure-from-motion (SfM) algorithms enable the simultaneous computation of camera projection matrices and 3D points by utilizing corresponding points in multiple views. Formally, given n projected points u_{ij} in m images, where $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, the objective is to determine both the projection matrices $P_1, \ldots, P_m$ and a consistent 3D structure $X_1, \ldots, X_n$. The typical steps involved in reconstructing a 3D scene using SfM are as follows:

1. **Feature Extraction**: Detecting distinctive points or features in the images.
2. **Feature Matching**: Identifying corresponding features across different images.
3. **3D Reconstruction**: Estimating the initial 3D structure and camera positions.
4. **Bundle Adjustment**: Refining the 3D structure and camera parameters to minimize reprojection errors.
5. **Meshing and Texturing from the Point Cloud (optional)**: Generating a surface mesh and applying textures for a realistic 3D model.

4.1.2 Multi-view Geometry

Multi-view geometry is a fundamental concept in computer vision that deals with the geometric relationships between multiple views of a scene or object. It encompasses techniques and algorithms for understanding and exploiting the information captured from different viewpoints to reconstruct the three-dimensional structure of the scene. MVG is a mature and complete open-source project targeting a Structure-from-Motion pipeline, which recovers camera poses, and a sparse 3D point cloud from an input set of images, playing a crucial role in several applications, including 3D reconstruction, stereo vision, object tracking, and camera calibration (Moulon *et al.*, 2016). By analyzing the correspondences between points in different views and leveraging geometric constraints, such as epipolar geometry and triangulation, it enables the estimation of camera poses, depth information, and scene geometry. Techniques, such as multi-view stereo (MVS), leverage multi-view geometry principles to reconstruct detailed 3D models from images or video sequences. It provides a pipeline using MVG and MVS (Xie; Li; Qi, 2019). Additionally, multi-view geometry forms the basis for advanced applications, such as augmented reality, where accurate alignment of virtual content with the real-world scene requires precise estimation of camera poses and scene geometry from multiple viewpoints. Multi-view geometry provides the theoretical foundation and practical tools for extracting rich three-dimensional information from various views, enabling a wide range of computer vision tasks and applications.

4.1.3 Neural Network

Neural networks have emerged as powerful tools in 3D reconstruction, offering innovative solutions to complex problems. Leveraging deep learning capabilities, neural networks can effectively process large volumes of data and extract meaningful features for reconstructing 3D scenes. These networks can be trained to directly associate sensor inputs and patterns with desired spatial maps or learn from geometric and density estimations to enhance 3D representations. This spatial understanding can then be improved by leveraging diverse data sources. Several architectures, such as convolutional neural networks (CNNs) (Li *et al.*, 2020) and recurrent neural networks (RNNs) (Kundu; Li; Rehg, 2018), have been adapted and tailored to address specific challenges in 3D reconstruction, including point cloud generation and mesh reconstruction. Through the iterative refinement of network architectures and optimization techniques, neural networks have been extensively applied to 3D reconstruction.

4.2 Mesh Generation

Frames 10 and 11provides a detailed comparison of various tools and techniques used for mesh reconstruction in 3D modeling. The table includes information about each tool's name, year of creation, most recent updates, pricing models, applicability to 3D reconstruction tasks, and their primary purposes. It highlights the diversity of approaches available for mesh generation, ranging from open-source solutions to high-cost commercial software. Open-source tools, such as *Point2Mesh*, *Ball-Pivoting Algorithm*, *CesiumJS*, *MeshLab*, and *Altizure Online Platform*, are prominently featured for their accessibility and effectiveness in tasks like converting point clouds into meshes and creating geospatial maps. Deep learning-based tools, such as *Point2Mesh* and *CasMVSNet*, leverage neural networks to enhance the accuracy of mesh reconstruction, showcasing the growing role of AI in improving 3D modeling processes.

Commercial software like *ConTextCapture*, *3DS MAX*, and *Geomagic* provide advanced capabilities for realistic mesh generation and detailed 3D modeling, often requiring licenses or subscriptions due to their specialized functionalities. Tools such as *OGC CityGML* focus on standardizing 3D city and landscape models, supporting applications like urban planning and simulation, while platforms like *Nostalgin Engine* are tailored for niche use cases, such as reconstructing ancient cities from low-quality images. Versatile platforms like *MeshLab* and *CloudCompare* support a wide range of tasks, including editing, processing, and analyzing point clouds and meshes. Table **??** demonstrates the breadth of tools available for mesh reconstruction, offering researchers and practitioners a wide selection to address diverse project requirements.

Frame 10 – Mesh Reconstruction Tools and Techniques 1

Name	Year	Last Update	Price	Applicable?	Purpose
Point2Mesh	2020	2020	Open source	Yes, it is designed to reconstruct a 3D surface mesh from an input point cloud.	A deep learning-based algorithm for generating 3D meshes, converting point clouds from sensor or image data into textured 3D meshes.
Ball-Pivoting Algorithm	2021	2021	Open source	Maybe, it is specifically designed for reconstructing 3D surfaces from point clouds, such as those obtained by 3D scanners.	Generates 3D meshes from point cloud data acquired through sensors or scanning techniques.
CesiumJS	2015	2024	Open source	Yes, it can create 3D maps from images, especially when elevation data is included.	Provides a platform for developing applications with geospatial data, such as 3D maps, terrains, satellite imagery, and geospatial analysis.
MeshLab	2020	2023	Open source	Yes, it is a 3D mesh processing software designed for managing and editing large, unstructured meshes.	Offers advanced tools for visualization, editing, and analyzing 3D meshes.
Altizure Online Platform	2017	2021	Open source	Yes, it generates textured 3D meshes from input images. The software runs the full 3D reconstruction pipeline, outputting textured meshes and camera positions.	An online platform for reconstructing 3D textured meshes from image datasets using computer vision techniques.
CloudCompare	2020	2020	Open source	Yes, it is a software for point cloud processing.	Processes and analyzes point cloud data.

Source: Own Work (2025).

Frame 11 – Mesh Reconstruction Tools and Techniques 2

Name	Year	Last Update	Price	Applicable?	Purpose
CasMVSNet	2021	2022	Open source	Maybe, used for reconstructing buildings.	A convolutional neural network for 3D reconstruction from multi-view stereo (MVS).
Monomer Model				No, it represents the independent 3D entity of a single object, such as a building or vehicle.	
Nostalgin Engine	2019	2019	Open source	Maybe, used to reconstruct ancient cities from low-quality images.	
OGC CityGML	2008	2021	Open source	No, it is a standardized data model and exchange format for storing and sharing 3D city and landscape models.	Stores and describes 3D city objects like buildings, roads, rivers, bridges, and vegetation, along with their relationships. It includes standardized levels of detail (LoDs) for various applications like simulations and urban management.
Digital Surface Map	2003	2003		Maybe, uses aerial and terrestrial images.	
ConTextCapture	2015	2022	R$10,731 per year	Yes, allows the creation of realistic 3D meshes of any scale using photographs or LiDAR point clouds.	Produces realistic 3D meshes from image or LiDAR data.
3DS MAX	1996	2024	R$853 per year, with a free trial	No, it is a modeling software.	

Source: Own Work (2025).

4.2.1 Ball-Pivoting Algorithm

The Ball-Pivoting Algorithm (BPA) is a surface reconstruction method that effectively creates a mesh from a set of unorganized 3D points, typically obtained from a scanning process.

The BPA leverages a ball of a fixed radius to "pivot" around the edges of an existing triangle, forming new triangles and progressively building the mesh. It is available in several software packages, such as MeshLab (Cignoni *et al.*, 2008), and has been utilized by many researchers (Hall *et al.*, 2022). The primary steps are described as follows (Bernardini *et al.*, 1999):

1. **Seed Triangle Formation**: Identify an initial triangle $\triangle = (p_i, p_j, p_k)$ with a ball of radius r.
2. **Pivoting**: For each edge (p_a, p_b) of the current mesh, find a point p_c such that the ball touches p_a, p_b, and p_c.
3. **New Triangle Formation**: Form a new triangle $\triangle = (p_a, p_b, p_c)$.
4. **Iterate**: Repeat the pivoting process for new edges until all points are processed.
5. **Multi-Scale Approach**: Optionally repeat with different ball radii to refine the mesh.

The BPA thus provides a robust method for surface reconstruction from point clouds, handling both sparse and dense datasets effectively. Its operation is described in the following sections.

4.2.1.1 Initialization

The algorithm begins by selecting an initial seed triangle. Let $P = \{p_1, p_2, \ldots, p_n\}$ be the set of 3D points. The first triangle $\triangle = (p_i, p_j, p_k)$ is formed such that the ball of radius r can touch $p_i, p_j,$ and p_k simultaneously without including any other points inside it. This can be formulated by solving for the center c of the ball:

$$||c - p_i|| = ||c - p_j|| = ||c - p_k|| = r$$

where $|| \cdot ||$ denotes the Euclidean distance. The center c must also satisfy:

$$||c - p_m|| > r \quad \forall p_m \in P \setminus \{p_i, p_j, p_k\}$$

4.2.1.2 Pivoting

Once the initial triangle is established, the algorithm pivots the ball around the edges of the triangle to form new triangles. Given an edge (p_a, p_b) and the current ball center c, the algorithm finds a new point p_c such that the ball, pivoted around (p_a, p_b), touches p_c and forms a new triangle $\triangle = (p_a, p_b, p_c)$. The center of the new ball position c' can be found by:

$$||c' - p_a|| = ||c' - p_b|| = ||c' - p_c|| = r$$

The pivoting operation involves solving for c' while maintaining contact with p_a and p_b:

$$c' = c + r \cdot \left(\frac{(p_c - p_a) \times (p_b - p_a)}{||(p_c - p_a) \times (p_b - p_a)||} \right)$$

4.2.1.3 Expansion

The algorithm pivots and forms new triangles around the mesh boundary until all reachable points are considered. The process iterates over the boundary edges of the growing mesh. For each boundary edge (p_i, p_j), a search for a new point p_k that satisfies the ball-pivoting criteria is performed.

4.2.1.4 Handling Variations

The BPA can be repeated with different ball radii to manage areas with varying sampling densities and handle noise in the point cloud. This multi-scale approach helps to ensure that the algorithm robustly reconstructs the surface.

4.2.2 Bundle Adjustment

Bundle Adjustment (BA) is a crucial optimization technique in SfM that simultaneously refines the 3D structure and camera parameters. The primary objective of BA is to minimize the reprojection error, which is the difference between the observed image points and the projected 3D points (Schönberger; Frahm, 2016). The reprojection error is defined as in equation 1.

$$e_{ij} = \|\mathbf{x}_{ij} - \pi(\mathbf{P}_i, \mathbf{X}_j)\|^2, \tag{1}$$

$\mathbf{x}_{ij}$ represents the observed image point of the j-th 3D point $\mathbf{X}_j$ in the i-th image, $\mathbf{P}_i$ is the camera projection matrix, and $\pi(\mathbf{P}_i, \mathbf{X}_j)$ is the projection function.

The optimization problem is formulated as in equation 2.

$$\min_{\{\mathbf{P}_i, \mathbf{X}_j\}} \sum_{i,j} \rho(e_{ij}), \tag{2}$$

ρ is a robust loss function for outliers. The main steps in bundle adjustment are described in the subsequent sections.

4.2.2.1 Parameterization

For robust parameter estimation, the Cauchy robust loss function is often employed, as shown in equation 3

$$\rho(e) = c^2 \log\left(1 + \left(\frac{e}{c}\right)^2\right), \tag{3}$$

c is a tuning parameter.

4.2.2.2 Optimization

The optimization is typically performed using the Levenberg-Marquardt algorithm, which iteratively updates the parameters to minimize the reprojection error. The Jacobian matrix J of the reprojection error concerning the camera parameters $\mathbf{P}_i$ and 3D points $\mathbf{X}_j$ is crucial for the optimization process.

4.2.2.3 Local and Global Bundle Adjustment

In practice, local BA is performed frequently on the most connected images after each image registration to mitigate local errors, while global BA is conducted less regularly to refine the entire model. The local BA focuses on a subset of the images and points to ensure efficiency, whereas the global BA encompasses all registered images and points for comprehensive optimization.

4.2.2.4 Filtering and Re-Triangulation

After each BA step, points with significant reprojection errors are filtered out, and re-triangulation is performed to improve the completeness of the reconstruction. This iterative process of BA, filtering, and re-triangulation continues until convergence, significantly enhancing the accuracy and completeness of the 3D reconstruction. The entire BA process is illustrated in the schematic diagram of the SfM pipeline in figure 17. The iterative nature of BA, combined with robust parameter estimation and efficient optimization techniques, makes it a cornerstone of modern 3D reconstruction methods (Schönberger; Frahm, 2016).

4.2.3 Greedy Triangulation

Greedy triangulation (GT) is a surface reconstruction algorithm to generate a triangular mesh from a set of 3D points. This method is beneficial in real-time applications due to its speed and robustness to noise. Its core idea is to iteratively add the shortest possible edge to form triangles without any intersections (Davis *et al.*, 2021). The main steps and the main models used are provided next. They were based on Dickson's work and Point Cloud Library's Greedy Projection Triangulation class Template Reference (Dickerson *et al.*, 1994; Rusu; Cousins, 2011).

1. **Neighbor Selection**: For each point p in the point cloud, the algorithm identifies k neighboring points within a predefined radius. The choice of k and the radius is influenced by the local density of points.

2. **Surface Normal Estimation**: A plane is estimated at each point p using a weighted least squares method. This plane approximates the local surface normal at p. Let $\mathbf{p}_i$ be the i-th neighbor of p, then the plane's normal $\mathbf{n}$ can be computed by minimizing equation 4.

$$\sum_{i=1}^{k} w_i \|(\mathbf{p}_i - p) \cdot \mathbf{n}\|^2 \tag{4}$$

   where w_i are the weights based on the distance of $\mathbf{p}_i$ from p.

3. **Visibility and Connectivity Pruning**: Points that are not visible from p are pruned. This is determined based on the angle between the surface normal at p and the vector $\mathbf{p}_i - p$. If the angle exceeds a threshold, the point is not considered visible.

4. **Triangle Formation**: The algorithm then iteratively forms triangles by connecting p with its visible neighbors. The goal is to form triangles that satisfy certain geometric constraints:

   - **Maximum Edge Length**: Any triangle edge length must not exceed a specified maximum.
   - **Angle Criteria**: The internal angles of the formed triangles must lie within a specified range to avoid skinny triangles.

5. **Triangle Addition**: Triangles are added to the mesh if they do not intersect with any existing triangles. This ensures that the mesh remains manifold.

6. **Surface Smoothing**: Although the basic GT algorithm does not include smoothing, post-processing steps may be applied to improve the mesh quality by reducing noise and eliminating small artifacts.

The primary models and methodologies underlying this technique are detailed below.

- **Distance-Based Weighting**: The weight w_i used in the least squares estimation of the surface normal can be defined as in equation 5.

$$w_i = \frac{1}{\|\mathbf{p}_i - p\|} \tag{5}$$

  This ensures that closer points have a higher influence on the average estimation.

- **Angle Threshold for Visibility**: The angle θ between the normal $\mathbf{n}$ and the vector $\mathbf{p}_i - p$ is given by 6.

$$\cos\theta = \frac{(\mathbf{p}_i - p) \cdot \mathbf{n}}{\|\mathbf{p}_i - p\| \|\mathbf{n}\|} \tag{6}$$

  A point $\mathbf{p}_i$ is considered visible if θ is below a certain threshold θ_{max}.

- **Triangle Quality Constraints**: The quality of the triangles is maintained by enforcing constraints on the edge lengths and internal angles. If $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ are the vertices of a triangle, the edge lengths $\|\mathbf{t}_i - \mathbf{t}_j\|$ must be less than a maximum length l_{max}. The internal angles α, β, γ must satisfy equation 7.

$$\alpha_{min} \leq \alpha, \beta, \gamma \leq \alpha_{max} \tag{7}$$

GT is widely used in 3D reconstruction due to its simplicity and efficiency, being recommended when dealing with large point clouds requiring real-time processing. By incrementally adding edges and forming triangles, the algorithm can quickly generate a surface mesh that approximates the underlying geometry of the object or scene. It is suitable for robotics, virtual reality, and geographic information systems (GIS) applications. Implementing GT in libraries like the Point Cloud Library (PCL) allows for extensive customization and optimization, making it a powerful tool for researchers in the field of 3D reconstruction (Rusu; Cousins, 2011; Davis *et al.*, 2021).

4.2.4 Gaussian Splatting

Gaussian Splatting is a technique used in 3D reconstruction and rendering that represents surfaces or point clouds using Gaussian functions rather than discrete points, unlike traditional point clouds, where each point represents a fixed position in space, Gaussian splatting models each point as a Gaussian distribution, which provides a smoother and more continuous representation of the underlying surface. This approach helps address issues of sparsity and noise in point clouds, producing softer, higher-quality reconstructions (Bagdasarian *et al.*, 2024).

The core idea of Gaussian splatting is to project Gaussian kernels into 3D space to represent the scene more densely. Instead of rendering a surface by connecting points with triangles, Gaussian splatting uses overlapping Gaussian functions that "splatter" across space to represent both surfaces and volumes. This method is recommended when dealing with noisy data or sparse point clouds, as the Gaussian distributions smooth out irregularities in the data.

Each point in the 3D space is modeled as a multivariate Gaussian function $G(x, y, z)$, typically defined as:

$$G(x, y, z) = \exp\left(-\frac{(x - \mu_x)^2}{2\sigma_x^2} - \frac{(y - \mu_y)^2}{2\sigma_y^2} - \frac{(z - \mu_z)^2}{2\sigma_z^2}\right)$$

where: - (x, y, z) are the coordinates of a point in 3D space, - (μ_x, μ_y, μ_z) are the mean values representing the center of the Gaussian in each dimension, - $\sigma_x, \sigma_y, \sigma_z$ are the standard deviations, controlling the spread of the Gaussian in each direction.

By summing or blending these Gaussian functions for a set of points, the resulting surface appears smooth, continuous, and less noisy than traditional point cloud representations (Bagdasarian *et al.*, 2024).

This method is used in several applications, including 3D rendering, point cloud processing, and neural rendering. In 3D rendering, the technique generates smoother and more visually appealing 3D models from sparse or noisy point clouds. Point cloud processing enhances point clouds by smoothing surfaces and reducing artifacts caused by sparsity or noise in the data. When used in neural rendering, Gaussian splatting can provide efficient scene rendering in real-time applications, effectively handling noisy and incomplete data. Among its advantages, it reduces noise inherently by smoothing the data, allowing for more accurate and continuous surfaces. Additionally, it provides control over the density of the reconstruction by adjusting the Gaussian spread, making it flexible for representing fine details or broader structures. Moreover, it handles sparse data effectively by spreading the influence of each point, filling gaps that traditional surface reconstruction methods might struggle with (Bagdasarian *et al.*, 2024).

Despite its advantages, there exist some challenges. Handling large numbers of overlapping Gaussians can be computationally expensive, especially for dense point clouds. Furthermore, choosing the proper spread parameters ($\sigma_x, \sigma_y, \sigma_z$) is critical to success, as improper values can lead to overly sharp or oversmoothed reconstructions. Gaussian splatting is an advanced technique that addresses some limitations of traditional point cloud rendering by offering smoother, noise-resistant surfaces. Its flexibility in dealing with sparse or noisy data, combined with its applications in neural rendering and real-time 3D reconstruction, makes it a valuable method in modern computer vision and graphics.

4.2.5 Delaunay Triangulation

Delaunay triangulation is a fundamental algorithm in computational geometry used to create a mesh of triangles from a given set of points in 2D or tetrahedra in 3D. It is widely used in 3D reconstruction, surface reconstruction, and mesh generation due to its properties that maximize the minimum angle in each triangle, which helps avoid narrow, elongated triangles that can lead to unstable geometry (Lee; Schachter, 1980).

The primary property is the empty circumcircle property. For any triangle in the Delaunay triangulation, the circumcircle that passes through its three vertices contains no other points from the dataset inside it. This property ensures that the triangulation is optimal in terms of geometric quality, as it avoids sliver triangles, which are undesirable in mesh generation. Formally, for a set of points $P = \{p_1, p_2, ..., p_n\}$ in the plane, a Delaunay triangulation $DT(P)$ is a triangulation such that no point $p \in P$ is inside the circumcircle of any triangle in $DT(P)$. This can be extended to higher dimensions, where the equivalent structure is called a Delaunay tetrahedralization in 3D (Lee; Schachter, 1980).

It can be constructed using several algorithms, including:

- Incremental Insertion: Points are added incrementally to the triangulation, and local adjustments are made to ensure that the Delaunay condition is maintained.
- Divide and Conquer: The point set is recursively divided into smaller sets, each of which is triangulated separately, and the triangulations are merged.
- Bowyer-Watson Algorithm: A popular algorithm for Delaunay triangulation that starts with a super-triangle containing all the points, then iteratively inserts points and retriangulates the affected region to preserve the Delaunay property.
- Flip Algorithm (Edge Flipping): In this algorithm, the triangulation is initially generated without considering the Delaunay condition, and non-Delaunay triangles are iteratively "flipped" (i.e., edges are swapped) to satisfy the empty circumcircle property.

There are several important properties:

- Maximizing the Minimum Angle: maximizes the minimum angle of all the angles of the triangles in the triangulation, which improves the geometric quality of the mesh.
- Uniqueness: If no four points are co-circular (in the case of 2D), the Delaunay triangulation is unique.
- Duality with Voronoi Diagrams: dual graph of the Voronoi diagram. This relationship is often leveraged in computational geometry to construct both structures efficiently.

In 3D reconstruction, it plays a critical role in surface meshing from point clouds, where it is used to connect points and generate a surface mesh that represents the underlying geometry of the object. By constructing triangles (or tetrahedra in 3D), the algorithm helps convert a set of discrete points into a continuous surface. This methodology is beneficial in applications like terrain modeling, medical imaging, and finite element analysis, where maintaining high-quality, well-shaped triangles or tetrahedra is crucial for the stability and accuracy of simulations or visualizations.

Despite its advantages, it has some limitations. For example, it does not handle non-convex point sets well, and the algorithm can be computationally expensive for large datasets. However, its robustness and ability to produce high-quality meshes make it an essential technique in 3D reconstruction and surface generation workflows. Delaunay triangulation is a powerful and widely used method in 3D reconstruction for converting point clouds into meshes.

Its geometric properties ensure the creation of well-shaped triangles, and its applications span across several fields, making it an essential tool for mesh generation.

4.2.6 Multi-View Stereo

Multi-view Stereo (MVS) is a critical technique in computer vision for reconstructing detailed 3D models from multiple images taken from different viewpoints. It aims to generate dense 3D reconstructions by leveraging the parallax observed in images captured from various angles. (Cernea, 2020) provides the main steps involved in MVS.

- **Image Acquisition and Preprocessing**: Multiple images of the scene are captured from different viewpoints. Preprocessing steps such as calibration, undistortion, and normalization are applied to ensure consistency in the dataset.
- **Feature Detection and Matching**: Keypoints or features are detected in each image using techniques like Scale-Invariant Feature Transform (SIFT) or Oriented FAST and Rotated BRIEF (ORB). Corresponding features across images are then matched to establish point correspondences.
- **Initial 3D Reconstruction**: Using the matched features, an initial sparse 3D reconstruction is created through triangulation. This step estimates the 3D coordinates of the matched features and the camera parameters (intrinsic and extrinsic).
- **Depth Map Estimation**: For each image, a depth map is computed to represent the distance of each pixel from the camera. This involves solving a depth optimization problem using stereo-matching techniques, such as plane-sweeping or graph cuts, which minimize a cost function defined over pixel disparities.

  Mathematically, the depth d of a pixel p in image I can be estimated by minimizing the cost function in equation 8.

  $$\sum_{q\in\mathcal{N}(p)} \phi\left(I(p) - I'(p')\right) + \lambda \sum_{q\in\mathcal{N}(p)} \psi(d - d(q)), \tag{8}$$

  where $\mathcal{N}(p)$ is the neighborhood of pixel p, ϕ and ψ are cost functions for data and smoothness terms, and I' is the corresponding image in the stereo pair.
- **Depth Map Fusion**:

  The depth maps from multiple images are fused to generate a consistent and dense 3D point cloud. Techniques, such as volumetric integration or depth map merging, combine the information from different viewpoints, reducing noise and filling gaps.

- **Surface Reconstruction**:

  The dense point cloud is processed to create a continuous surface mesh. Algorithms like Poisson Surface Reconstruction or Delaunay triangulation are employed to generate a mesh that accurately represents the underlying geometry of the scene.

  The Poisson Surface Reconstruction formulates the problem as a Poisson equation 9

  $$\nabla \cdot \mathbf{v} = \mathbf{f}, \tag{9}$$

  where $\mathbf{v}$ is the vector field representing the oriented point cloud, and $\mathbf{f}$ is the scalar field representing the indicator function of the surface.

- **Texture Mapping**:

  Textures from the original images are mapped onto the reconstructed surface to create a realistic appearance. This step involves projecting the images onto the 3D mesh and blending them to ensure seamless transitions between different textures.

MVS can produce high-quality 3D reconstructions from widely available photographic data, allowing a versatile and powerful tool in computer vision. Some articles provide comprehensive insights into advanced MVS techniques and their mathematical foundations, offering valuable contributions to the ongoing development of this field, such as (Hu *et al.*, 2022; Han; Shen, 2019; Zolanvari *et al.*, 2019; Orsingher *et al.*, 2022; Cernea, 2020).

4.2.7 Neural Radiance Fields

Neural Radiance Fields (NeRFs) are a cutting-edge approach for 3D reconstruction that models a scene's geometry and appearance using a neural network. Its core idea is to represent the volume density and color of a scene as a continuous function using a neural network, which takes in a 3D coordinate and a viewing direction as inputs and outputs the corresponding color and volume density at that point (Müller *et al.*, 2022; Tancik *et al.*, 2023). The primary steps in NeRF 3D reconstruction include:

1. **Positional Encoding**: The input 3D coordinates $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \phi)$ are first encoded using high-frequency functions, often trigonometric functions, to enable the network to capture high-frequency details. The encoded inputs are fed into the neural network.

2. **Neural Network**: The core of the NeRF model is a multi-layer perceptron (MLP) that estimates the RGB color and volume density σ at each spatial location. The MLP is designed to output a four-dimensional vector, where the first component represents the density σ, and the remaining three components represent the RGB color $\mathbf{c}$.

3. **Volume Rendering**: Given a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ where $\mathbf{o}$ is the camera origin and $\mathbf{d}$ is the direction, the color of the pixel is computed by integrating the colors along the ray, weighted by the volume density. This process can be mathematically expressed as in equation 10.

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt, \tag{10}$$

   where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds\right)$ accounts for the accumulated transmittance.

4. **Optimization**: The neural network parameters are optimized by minimizing the difference between the rendered images and the ground truth images. This is typically done using mean squared error (MSE) between the predicted and actual pixel colors.

In recent years, NeRFs have seen the development of numerous models and variations, each targeting specific challenges and enhancements in 3D scene representation and rendering. The original NeRF model, proposed by Mildenhall *et al.*, set the foundation by introducing a fully connected neural network to encode volumetric scene functions. It was followed by advancements such as Mip-NeRF and Mip-NeRF 360, which addressed issues like anti-aliasing and extended the model's capabilities to handle unbounded scenes more efficiently (Tancik *et al.*, 2023). Other important contributions include FastNeRF, which focuses on high-fidelity rendering at real-time speeds, and Plenoxels, which deviates from neural networks, opting instead for a sparse voxel representation that offers faster rendering times (Tancik *et al.*, 2023).

Models like NeRF-W and RawNeRF were developed to handle real-world complexities, such as varying lighting conditions and noisy data. Ref-NeRF, conversely, aims to refine the quality of rendered images by incorporating reflection effects (Tancik *et al.*, 2023). Each of these models contributes uniquely to the evolution of NeRFs, pushing the boundaries of what is possible in neural rendering and real-time 3D reconstruction. NeRF has shown remarkable success in rendering high-quality novel views from a set of 2D images, enabling the synthesis of photo-realistic views from arbitrary viewpoints. It effectively combines machine learning and traditional computer graphics techniques to achieve state-of-the-art results in 3D reconstruction.

4.2.8 Poisson Reconstruction

Poisson Reconstruction is a widely used method for surface reconstruction from point clouds, particularly when a smooth and watertight surface is desired. This technique relies on the mathematical foundation of solving the Poisson equation, which reconstructs a function (the surface) from its gradient field (the normals of the points). The main steps in the Poisson Reconstruction process are outlined as follows (Kazhdan; Bolitho; Hoppe, 2006).

1. **Point Cloud Normal Estimation**: The first step involves estimating normals for the points in the point cloud. Normals are crucial as they define the orientation of the surface at each point. These are typically computed using nearest neighbors.

2. **Divergence of Normal Field**: The next step is to compute the divergence of the standard field. The divergence operator applied to the normals gives a scalar field representing the difference between the actual and estimated normals, which is expressed as in equation 11

$$\nabla \cdot \mathbf{n} = \sum_i \frac{\partial n_i}{\partial x_i}, \tag{11}$$

   where $\mathbf{n}$ represents the normal vector, and x_i are the coordinates.

3. **Solving the Poisson Equation**: The core of the Poisson Reconstruction is solving the Poisson equation 12.

$$\Delta \phi = \nabla \cdot \mathbf{n}, \tag{12}$$

   where Δ is the Laplace operator, ϕ is the implicit function that defines the surface, and $\nabla \cdot \mathbf{n}$ is the divergence computed in the previous step. The solution ϕ defines the reconstructed surface, where the zero-level set represents the final surface.

4. **Surface Extraction**: The final surface is extracted from the implicit function ϕ using methods like the Marching Cubes algorithm. This step involves creating a mesh approximating the zero-level set of ϕ.

5. **Post-Processing**: The resulting mesh may undergo additional processing to refine the surface, remove noise, and close any holes that may have formed during reconstruction.

Poisson Reconstruction is particularly effective in generating smooth surfaces even from noisy data. It performs well in city-scale reconstructions, producing watertight meshes crucial for applications like mixed reality visualizations (Han; Shen, 2019; Davis *et al.*, 2021; Hall *et al.*, 2022). However, the method requires careful tuning of parameters such as octree depth and point weight to achieve the desired balance between fitting the input data and maintaining a smooth surface (Han; Shen, 2019; Davis *et al.*, 2021; Hall *et al.*, 2022).

4.2.9 Other Reconstruction Algorithms

In addition to usual methodologies for 3D reconstruction, other algorithms provide unique approaches to generating and refining 3D models. These methods often complement the primary techniques and can be particularly useful in specific contexts or for handling unique data types.

These methodologies, among others, provide essential tools and techniques for advancing the field of 3D reconstruction. They offer flexibility and precision, allowing practitioners to tailor their approaches to specific datasets and application requirements.

4.2.9.1 3D Reconstruction via Height Field Estimation

One effective method for reconstructing 3D models from 2D data is through height field estimation. This technique involves using elevation or height data to extrapolate a 3D surface from a 2D base, such as a GIS map. The process typically involves interpolating between known height points and generating a continuous surface, often represented as a triangulated mesh. This method is especially useful in urban modeling and topographic mapping, where elevation data can be easily obtained from digital elevation models (DEMs) or LiDAR data. The resulting 3D models are precious for urban planning and environmental studies applications.

4.2.9.2 Point Cloud Filtering and Classification

Point cloud filtering and classification are critical steps in processing raw point cloud data, mainly when dealing with large datasets, such as those obtained from LiDAR scans. Filtering involves removing noise and irrelevant points, often achieved through statistical analysis and clustering techniques. Classification then segments the point cloud into different categories, such as ground, vegetation, and man-made structures. Machine learning algorithms, including supervised classifiers like random forests or neural networks, are increasingly employed to enhance the accuracy and automation of this process. The result is a cleaner, segmented point cloud that can be further processed into a 3D model (Vosselman; Maas, 2010).

4.2.9.3 Surface Reconstruction via Energy Minimization

Surface reconstruction from point clouds often involves minimizing an energy function that represents the difference between the point cloud and the reconstructed surface. One common approach is to use variational methods, where the energy function is defined to penalize deviations from the input data and enforce smoothness constraints on the surface. The solution to this optimization problem yields a smooth, continuous surface that best fits the input points. Techniques like Poisson surface reconstruction extend this concept by solving a spatial Poisson equation, which provides a global approach to surface fitting that is robust to noise and outliers (Kazhdan; Bolitho; Hoppe, 2006).

4.2.9.4 Implicit Surface Modeling

Implicit surface modeling is an advanced technique that defines surfaces implicitly as the zero-level set of a scalar field. Unlike explicit methods that directly represent the surface geometry, implicit methods allow for smooth and continuous surface representations, which are particularly useful for capturing complex shapes and details. The scalar field can be represented by mathematical functions or neural networks trained to approximate the surface. This approach is powerful for applications requiring smooth surfaces, such as medical imaging and computer graphics. The marching cubes algorithm is often used to extract the final mesh from the implicit representation, providing a versatile framework for surface reconstruction (Carr; Fright; Beatson, 1997).

4.2.9.5 Mesh Refinement and Smoothing

Mesh refinement and smoothing are critical post-processing steps in 3D reconstruction. Refinement involves increasing the resolution of a mesh by adding more vertices and faces, often guided by the curvature of the surface. This can be achieved through techniques like subdivision surfaces or adaptive mesh refinement. Smoothing, on the other hand, aims to reduce noise and irregularities in the mesh, typically using algorithms like Laplacian smoothing or Taubin smoothing. These techniques are essential for preparing 3D models for visualization, simulation, and analysis, ensuring that the surfaces are both aesthetically pleasing and accurate (Taubin, 1995).

4.3 Other

Frames 12, 13 and 14 provides a comprehensive overview of related software, focusing on their year of creation, last update, pricing models, applicability to the study, and specific purposes. The tools are categorized based on their functionalities, ranging from volumetric approaches and depth simulation to geospatial analysis and procedural modeling. Several tools, such as *Depth Buffer*, *NASA World Wind*, and *Unity Game Engine*, are noted for their potential application in enhancing 3D reconstruction and simulations, particularly for interactive and geospatial data visualization. For example, *Depth Buffer* aids in simulating depth perception during rendering, while *NASA World Wind* provides open-source terrestrial data visualization capabilities.

It also includes specialized tools like *LAStools* and *OGC CityGML*, which are focused on LiDAR data processing and standardized city model storage, respectively. These tools are invaluable for urban simulations, though not directly applicable to core 3D reconstruction tasks. Software like *Esri CityEngine*, *Rhino*, and *SketchUp* are tailored for procedural modeling, industrial design, and architecture, with limited applicability to 3D reconstruction in the current study. Similarly, tools like *Structure from Motion* and *ScanComplete* cater to specific tasks, such as reconstructing sparse or incomplete 3D data.

Upon further review, some tools such as *Surface Net* and *Structure from Motion*, were found to serve specialized roles in enhancing existing 3D models or handling sparse data. These tools were subsequently categorized under post-processing methodologies or excluded from this

Frame 12 – Related Software 1

Name	Year	Last Update	Price	Applicable?	Purpose
Depth Buffer			Open source	Potentially useful, it is designed to simulate depth perception in 3D environments during rendering.	Determines object distance relative to the camera in a scene.
Autodesk 3ds Max				No, it is modeling software.	
Blender	1994	2024	Open source	No, it is modeling software.	
Bundler				No, it is a container technology software.	
City GML				No, it is a 3D City Database.	
OGC CityGML	2008	2021	Open source	No, it is a standardized data format for storing and sharing 3D city models.	Used for urban simulations, urban data mining, and management.
Pix3D	2018	2021	Open source	No, it reconstructs objects from a single image.	
QGIS	2002	2023	Open source	No, primarily designed for geospatial analysis, map creation, and working with vector and raster data in 2D.	

Source: Own Work (2025).

study due to their limited relevance to the primary objectives. Overall, it highlights the wide range of tools available for 3D reconstruction, visualization, and processing, allowing researchers to evaluate and select the most appropriate solutions for specific use cases.

4.4 Notable Applications

Numerous software tools have been developed to facilitate the generation of point clouds and meshes from several data sources (Ledoux *et al.*, 2021; Schönberger; Frahm, 2016). These tools leverage advanced algorithms and computational techniques to transform raw data into detailed 3D models, making them invaluable in fields such as computer vision, archaeology, and

Frame 13 – Related Software 2

Name	Year	Last Update	Price	Applicable?	Purpose
ScanComplete	2017	2020	Open source	No, it reconstructs incomplete 3D files.	Completes partially scanned 3D scenes and predicts full 3D models with semantic voxel labels.
SRLP				Unlikely, its main purpose is to enhance computational efficiency for reconstructing sparse images in LASAR environments.	
SSCNet	2016	2017	Open source	No, it reconstructs a single 3D image.	
Surface Net	2017	2020	Open source	Maybe, a volumetric approach to deal with "incompleteness" and "imprecision" in sparse MVS configurations.	Enhances 3D meshes with superior accuracy and recall, even in sparse setups.
Coordinate Transformation					
LAStools	2007	2023	Open source	No, designed for efficient processing and analysis of LiDAR data.	Performs operations such as filtering, classification, statistics, and visualization on large LiDAR datasets.
Structure from Motion	2010	2024	Open source	No, it is a computer vision and photogrammetry technique.	Reconstructs 3D structure from 2D images taken from multiple viewpoints.

Source: Own Work (2025).

virtual reality. This section provides an overview of some of the most prominent and widely used software tools created by researchers and developers worldwide. By examining their capabilities, strengths, and unique features, a comprehensive understanding of the current landscape of 3D reconstruction technologies and their practical applications can be achieved.

Frame 14 – Related Software 3

Name	**Year**	**Last Update**	**Price**	**Applicable?**	**Purpose**
NASA World Wind	2004	2024	Open source	Potentially, it can serve as a source of terrestrial data.	Open-source software for interactive 3D visualization of geospatial data and Earth models.
Unity Game Engine	2004	2024	Free to 28,611 per year	Maybe, it can import 3D models to create interactive experiences, simulations, or games.	A game development platform also used for 3D simulations and virtual training.
Rhino	1998	2022	$ 995	No, it is not specialized for creating 3D maps from images but is used for industrial design, architecture, and engineering.	3D modeling software.
SketchUp			Free and paid versions available	Depends, it is manual modeling software with the ability to import images for reference.	Allows for manual 3D modeling.
Esri CityEngine	2008	2023	R$1,412 per year to R$52,406 per year	No, focused on procedural 3D modeling and urban environment visualization.	Advanced 3D modeling software for urban planning, architecture, and geosciences.
ArcGIS Pro	2015	2023	R$9,000 to R$50,000 per year, depending on the package	Maybe for final visualization. It focuses on geospatial analysis and data visualization, with limited 3D modeling capabilities.	Spatial data analysis software.

Source: Own Work (2025).

4.4.1 OpenMVG

OpenMVG, or Open Multiple View Geometry, is a robust and versatile open-source library designed for 3D reconstruction from multiple images. Developed by (Moulon *et al.*, 2016) to support academic research and industrial applications, OpenMVG focuses on providing a complete pipeline for structure-from-motion (SfM) tasks. The library includes a range of tools for feature detection, matching, camera pose estimation, and triangulation, contributing to the creation of sparse and dense point cloud generation. One of its key strengths lies in its modular design, allowing users to customize and extend the pipeline according to their specific needs. By leveraging state-of-the-art algorithms, OpenMVG ensures high accuracy and reliability in reconstructing 3D models from photographic datasets. Additionally, its comprehensive documentation and active community support make it accessible to both beginners and experienced practitioners in the field of computer vision and 3D reconstruction, being employed by other researchers (Xie; Li; Qi, 2019) in the development of reconstruction pipelines.

4.4.2 3D-ReConstnet

In single-view 3D object reconstruction, the challenge lies in inferring the self occluded portions of objects, rendering the task inherently ambiguous. Addressing this issue, a novel neural network called 3D-ReConstnet has been proposed by (Li *et al.*, 2020). This network, designed as an end-to-end reconstruction framework, leverages a residual network to extract features from a 2D input image, culminating in a feature vector. To contend with the uncertainty surrounding the self-occluded segments of objects, 3D-ReConstnet employs a Gaussian probability distribution learned from the feature vector to predict the point cloud. Remarkably, this approach enables the generation of determinate 3D outputs for images with adequate information, while also facilitating the generation of semantically distinct 3D reconstructions for self-occluded or ambiguous object segments. Evaluation on ShapeNet and Pix3D datasets has demonstrated promising enhancements in the reconstruction results.

4.4.3 3D-RCNN

3D RCNN, or 3D Region-based Convolu-tional Neural Network, is a novel approach for instance-levelc3D object reconstruction from 2D images proposed by (Kundu; Li; Rehg, 2018). Unlike traditional methods, which rely on manual segmenta-tion or bounding box annotations, 3D RCNN leverages deep learning techniques to directly infer the 3D shapes and poses of object instances within an image. The network architecture is designed to disentangle the complex attributes of a scene, such as lighting, shape, and surface properties, by factoring the scene into object instances with associated shape and pose. This disentangled representation allows for more accu-rate scene understanding and enables downstream tasks like path planning and future object location prediction. The key innovation of 3D RCNN lies in its differentiable Render-and-Compare loss, which enables the network to obtain supervision from 2D annotations and bootstrap with direct 3D supervi-sion when available. By exploiting rich shape priors learned from CAD model collections, 3D RCNN achieves state-of-the-art performance in complex real-world datasets, making it a promising solution for various applications, including autonomous driving and robotics.

4.4.4 Point Cloud Library

The Point Cloud Library (PCL) is an open-source, BSD-licensed software library designed for n-D point cloud and 3D geometry processing. Developed with efficiency and performance in mind, it is a fully templated C++ library that integrates seamlessly with the Robot Operating System (ROS). It provides a comprehensive range of algorithms and data structures for 3D perception, making it an invaluable tool in fields such as robotics, computer vision, and more (Rusu; Cousins, 2011). PCL supports a wide variety of algorithms for processing point clouds, including filtering, feature estimation, surface reconstruction, registration, model fitting, and segmentation. The library's modular design allows for easy integration of its components into several applications, which is facilitated by its extensive use of template programming, SSE optimizations, and multi-core parallelization support through OpenMP and Intel's Threading Building Blocks (TBB) library (Rusu; Cousins, 2011). The architecture is built around several key components, each providing a specific set of functionalities. For example, libpcl_filters is responsible for data filtering tasks, such as downsampling and outlier removal, while libpcl_segmentation handles cluster extraction and model fitting. PCL also includes specialized modules like libpcl_io for input/output operations and libpcl_visualization for rendering point cloud data (Rusu; Cousins, 2011). In the literature, it has been widely adopted for several applications, including autonomous navigation, object recognition, and manipulation tasks in robotics. Due to its ability to handle large datasets efficiently and its comprehensive set of tools, it is convenient for applications with 3D point cloud data (Davis *et al.*, 2021).

4.4.5 Colmap

Colmap is a state-of-the-art Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline that provides a comprehensive solution for 3D reconstruction from unordered image collections. Developed by (Schönberger; Frahm, 2016; Schönberger *et al.*, 2016), it features an incremental SfM approach, enabling it to iteratively refine a 3D model by adding new images and points. The pipeline begins with feature extraction and matching, followed by geometric verification to establish reliable correspondences between images. It then proceeds with incremental reconstruction, where the model is incrementally built and refined using techniques such as bundle adjustment and outlier filtering. This approach is particularly effective for handling large and complex datasets (Schönberger; Frahm, 2016). Colmap's utility extends across several research areas, including cultural heritage preservation, urban mapping, and autonomous driving. It has been employed in numerous studies for tasks such as the detailed reconstruction of historical sites, the creation of 3D maps for urban planning, and the generation of training data for machine learning models in autonomous systems. The flexibility and robustness make it a preferred choice for researchers and practitioners in computer vision and related fields (Schönberger *et al.*, 2016).

4.4.6 Metashape

Metashape is a commercial photogrammetric software tool developed by Agisoft that processes digital images, including aerial and close-range photography, to generate 3D spatial data. It supports a wide range of input data, such as images from RGB, thermal, and multispectral cameras, as well as laser scans, allowing for the creation of point clouds, textured polygonal models, georeferenced orthomosaics, and digital surface models (DSM) or digital terrain models (DTM) (Agisoft, 2023). It is capable of processing large datasets, with the ability to handle over 50,000 photos through local cluster or cloud-based distributed processing, which is particularly suitable for applications in GIS, cultural heritage documentation, visual effects production, and several engineering fields. Metashape's robust toolset includes capabilities for eliminating shadows and texture artifacts, calculating vegetation indices, and extracting data for agricultural equipment maps. It also features advanced options like stereoscopic mode and precise control over accuracy, catering to both beginners and experts in photogrammetry (Agisoft, 2023). Researchers and professionals use it to create highly detailed 3D reconstructions of urban environments, archaeological sites, and natural landscapes.

4.4.7 MicMac

MicMac is a comprehensive, free, and open-source photogrammetry software suite designed to handle several aspects of 3D reconstruction from images. Developed by (Rupnik; Daakir; Deseilligny, 2017), MicMac offers robust tools for processing large datasets, making it suitable for applications ranging from small-scale projects to extensive urban modeling. As with other leading photogrammetry solutions, such as Pix4D and ContextCapture, MicMac excels in feature extraction, image matching, and dense 3D reconstruction. The versatility of this software tool is further enhanced by its ability to generate high-quality, large-scale meshes and realistic textures, similar to commercial counterparts like COLMAP and OpenMVS. Moreover, MicMac's open-source nature promotes continuous improvement and customization by the user community, ensuring it remains a cutting-edge tool in the rapidly evolving field of photogrammetry (Rupnik; Daakir; Deseilligny, 2017). The workflow in MicMac typically involves key steps such as data acquisition, feature extraction, bundle adjustment, and optional mesh generation and texturing, aligning with the standard practices observed in UAV-based photogrammetry for 3D reconstruction (Han *et al.*, 2021).

4.4.8 Pix 4d

Pix4D is a commercial photogrammetry software suite designed for creating 3D maps and models from images captured by drones, smartphones, or other devices. It is widely used in construction, agriculture, surveying, and real estate sectors, for generating high-quality geospatial data. It employs advanced algorithms for image processing, point cloud generation, and 3D reconstruction, offering a range of products like Pix4Dmapper, Pix4Dfields, and Pix4Dcloud, each tailored to specific cases. One of the key features is its ability to perform detailed and accurate mapping through photogrammetry, making it a valuable tool in applications requiring precision and reliability. It allows for the processing of images into dense point clouds, mesh models, and textured surfaces, enabling users to extract valuable spatial information from imagery. In the literature, Pix4D has been utilized in several applications (Han *et al.*, 2021; Li *et al.*, 2023). These examples illustrate the versatility and effectiveness of this software tool in generating accurate and detailed spatial data across several fields.

subsectionVisual SFM

VisualSFM (Wu, 2011) is a powerful and user-friendly software tool designed for 3D reconstruction using structure-from-motion (SfM) techniques. As described by the author, VisualSFM enables the automatic computation of camera positions and 3D points from a series of images, making it a versatile tool for several photogrammetry applications. The system integrates with multi-view stereo (MVS) algorithms to enhance the reconstruction process, offering high-quality dense point clouds and 3D models. VisualSFM's efficiency and ease of use make it a popular choice among researchers and professionals in the field. The typical workflow in VisualSFM involves feature extraction, feature matching, 3D reconstruction, and bundle adjustment, similar to other photogrammetry tools like COLMAP and MicMac. Additionally, VisualSFM's ability to handle large datasets and produce detailed 3D models with realistic textures, making it suitable for applications ranging from small-scale projects to extensive urban reconstructions.

4.4.9 Gipuma

Gipuma is a method for dense 3D model reconstruction in digital cities that utilizes computationally efficient multi-view stereo networks. The method aims to reconstruct accurate 3D points while minimizing the generation of non-existent false points (Galliani; Lasinger; Schindler, 2015). In a quantitative evaluation on the DTU dataset proposed by (Hu *et al.*, 2022), Gipuma achieved competitive results in terms of accuracy and computational complexity compared to other methods such as SurfaceNet, MVSNet, and CasMVSNet. However, it was observed that Gipuma's higher accuracy comes at the cost of increased computational complexity, making it less practical for reconstructing large-scale scenes or handling higher resolution images. Despite this limitation, Gipuma represents a promising approach for dense 3D model reconstruction, particularly in scenarios where accuracy is paramount and computational resources allow for its use.

4.4.10 MeshLab

MeshLab is a comprehensive open-source tool designed for processing and editing 3D triangular meshes. Developed by (Cignoni *et al.*, 2008), it provides a variety of functions for the inspection, cleaning, repairing, rendering, and conversion of large unstructured 3D models obtained from several sources, including 3D scanning, photogrammetry, and other modeling techniques, having a large set of features that support a wide range of mesh processing tasks. These include surface reconstruction, which helps in creating a complete surface from a point cloud, mesh simplification, which reduces the number of polygons while preserving the overall shape and appearance, and mesh optimization, which improves the quality and usability of the 3D models by addressing issues such as noise, holes, and non-manifold elements (Cignoni *et al.*, 2008). It supports several file formats, enabling users to import and export models in formats like PLY, STL, OBJ, 3DS, and COLLADA. It has been widely adopted for numerous applications. For instance, it is frequently used in the digitization of cultural heritage artifacts, allowing researchers to create detailed 3D models of historical objects for preservation, study, and virtual display. The software's capabilities in mesh cleaning and optimization are particularly beneficial in this context, as they ensure the accuracy and quality of the digital representations (Davis *et al.*, 2021). It is also used in the field of biomedical imaging to create and refine 3D models of anatomical structures. Its powerful mesh processing tools enable medical researchers to analyze complex geometries and prepare models for simulation and visualization purposes. In computer graphics, is employed to prepare models for animation, gaming, and other visual effects, taking advantage of its robust feature set to optimize and refine 3D assets. MeshLab stands out as a versatile and powerful tool for 3D mesh processing, offering a rich array of functionalities that cater to the needs of both novice and advanced users. Its widespread use in several research and practical applications underscores its importance and utility in the field of 3D modeling and reconstruction.

4.4.11 OpenMVS

OpenMVS, or Open Multi-View Stereo, is a specialized library designed for computer-vision scientists and the Multi-View Stereo reconstruction community. It addresses a critical gap in the photogrammetry workflow by providing a complete set of algorithms to recover the full surface of a scene from input data, which typically includes camera poses and a sparse point-cloud generated by other Structure-from-Motion (SfM) tools such as OpenMVG (Cernea, 2020). The output from OpenMVS is a detailed textured mesh, including several key capabilities that make it effective in the 3D reconstruction pipeline. The dense point-cloud reconstruction process generates a comprehensive and highly accurate point cloud from the initial sparse input, serving as the foundation for subsequent steps. The mesh reconstruction step estimates a mesh surface that best fits the dense point cloud, creating a coherent and continuous 3D model. Mesh refinement then focuses on recovering fine details in the mesh, enhancing the overall quality and precision of the 3D model (Cernea, 2020). Finally, mesh texturing computes a sharp and accurate texture to color the mesh, resulting in a realistic and visually appealing 3D representation. It has been utilized in a variety of research and practical applications due to its robust functionality. It integrates well with other SfM tools by taking their output and advancing the reconstruction process to produce fully textured 3D models. In the literature, OpenMVS is often cited for its ability to produce high-quality 3D models with minimal user intervention. Researchers have employed it to enhance the accuracy of reconstructions and to streamline the workflow from image capture to final 3D model generation (Xie; Li; Qi, 2019; Han *et al.*, 2021). The comprehensive documentation and support for multiple input formats, including OpenMVG, COLMAP, Metashape, iTwin Capture Modeler, and Polycam, further contribute to its widespread adoption in the academic and industrial sectors. OpenMVS continues to be an essential tool in the field of 3D reconstruction, offering advanced capabilities that significantly enhance the quality and usability of reconstructed models from multi-view stereo data (Cernea, 2020).

4.4.12 Nerfstudio

Nerfstudio is a modular framework designed for the development of NeRF. Developed by (Tancik *et al.*, 2023), it aims to streamline the development and deployment of NeRF-based methods by offering a flexible, modular, and comprehensive framework. The primary goal is to simplify the integration of NeRF into several projects, providing a range of plug-and-play components that can be easily customized and extended. Nerfstudio supports multiple input data pipelines and is built around core modular components, including encoders, samplers, fields, and renderers. Its modular design facilitates the implementation of custom NeRF methods and allows for extensive real-time visualization tools, streamlined data processing, and various export modalities such as video, point cloud, and mesh representations. The framework's real-time web viewer enables interactive visualization of NeRF scenes during both training and testing, making it accessible without requiring a local GPU machine. In the literature, Nerfstudio has been utilized to consolidate several NeRF techniques, enabling researchers and practitioners to experiment with combining components from multiple methods. For instance, the development of Nerfacto, a method that balances speed and quality by integrating features from recent NeRF papers, demonstrates its flexibility (Tancik *et al.*, 2023). The framework's modular nature supports real-world data processing, making it suitable for applications in computer vision, graphics, robotics, and more. Nerfstudio has been cited for its contributions to simplifying NeRF development and enabling more efficient and effective experimentation. Its ability to integrate with various data input formats, including mobile capture applications and popular photogrammetry tool, further enhances its versatility. Researchers have leveraged Nerfstudio to develop methods that achieve high-quality 3D reconstructions, benefiting from its comprehensive suite of tools and real-time visualization capabilities (Tancik *et al.*, 2023). The framework's open-source nature and community-driven development model have led to significant contributions from several academic, corporate, and independent collaborators. This has resulted in continuous improvements and the addition of new features, making Nerfstudio a valuable resource for the NeRF research community (Tancik *et al.*, 2023). It represents a significant advancement in the development of Neural Radiance Fields, providing a robust and flexible platform that accelerates research and application in the field of neural rendering.

Instant Neural Graphics Primitives (Instant-NGP) is a framework developed by NVIDIA researchers (Müller *et al.*, 2022). It leverages a novel multiresolution hash encoding to enable rapid training and rendering of neural graphics primitives across several tasks, including image synthesis, signed distance functions (SDF), neural radiance caching (NRC), and Neural Radiance Fields (NeRF). This approach allows for the use of smaller neural networks without compromising quality, significantly reducing the computational cost associated with floating point and memory access operations. The core innovation of Instant-NGP lies in its multiresolution hash encoding, which organizes trainable feature vectors into a hash table optimized through stochastic gradient descent. This structure disambiguates hash collisions and facilitates parallelization on modern GPUs using fully fused CUDA kernels, resulting in a speedup of several orders of magnitude. Training high-quality neural graphics primitives can be achieved in seconds, and rendering occurs in milliseconds at resolutions up to 1920×1080 (Müller *et al.*, 2022). Instant-NGP has been utilized for several applications due to its efficiency and high-quality results. For example, it has been incorporated into methods like Nerfacto, which combines ideas from multiple research papers to enhance performance and reconstruction quality. Nerfacto uses Instant-NGP's hash encoding to generate efficient and accurate scene density functions, contributing to improved sampling processes and overall NeRF performance (Tancik *et al.*, 2023; Müller *et al.*, 2022). Its effectiveness has also been demonstrated in tasks such as gigapixel image representation, where a neural network maps 2D coordinates to RGB colors, and in SDF learning, where a network maps 3D coordinates to distances from a surface. These applications highlight the versatility and adaptability of encoding across different types of neural graphics primitives. Instant-NGP represents a significant advancement in neural graphics primitives, offering a scalable and efficient solution that enhances the speed and quality of 3D reconstructions and other related tasks.

4.5 Framework Implementation

The proposed framework serves as an integrated pipeline designed to enable the future development of realistic ADAS simulations by bridging the gap between 3D reconstruction, Unreal Engine, and a selected simulation tool. By structuring a workflow that incorporates real-world 3D environments into driving simulators, this framework enhances the realism, accuracy, and scalability of autonomous vehicle simulations. The framework is structured into three main phases. The first phase involves the 3D reconstruction of real-world environments using a methodology, such as COLMAP, MicMac, NeRF-based approaches, or 3D Gaussian Splatting

(3DGS). These methods allow for the creation of detailed and high-fidelity digital representations of roads, intersections, and urban environments. The reconstructed data is then processed and converted into a format compatible with game engines and simulation platforms. The second phase consists of integrating the reconstructed 3D model into Unreal Engine. Unreal Engine is used as an intermediary platform where mesh optimization, material adjustments, and lighting corrections are performed to enhance the quality and usability of the reconstructed environment. The meshes are imported into Unreal Engine using tools such as Volinga Creator and Volinga Import, particularly for 3DGS-based reconstructions. Unreal Engine provides real-time rendering capabilities, allowing visualization and necessary modifications before the final deployment into CARLA. The third phase involves exporting the refined environment for simulation purposes.

One of the primary contributions of this framework is its ability to integrate photorealistic 3D reconstructions into a simulation environment, providing high-detail and realistic testing environments for autonomous vehicle research. The use of cutting-edge 3D reconstruction methodologies and high-fidelity game engine rendering significantly enhances the adaptability of ADAS simulations. By combining these elements, the framework supports the development of a structured methodology for integrating real-world 3D environments into autonomous vehicle simulators. Future work in this framework may involve the automation of the integration pipeline to facilitate the importation of reconstructed scenes. Further enhancements may include real-time dynamic scene generation, allowing for more adaptive and interactive simulation environments. Improvements in AI-driven perception models and multi-sensor fusion, including LiDAR, stereo vision, and radar, could contribute to more comprehensive ADAS testing.

5 RESULTS AND DISCUSSION

This chapter presents and analyzes the findings of the study, offering a detailed discussion on the outcomes of the methodologies applied for 3D reconstruction and simulation. The results are structured to address the main research objectives, including the effectiveness of the selected reconstruction tools, the integration of datasets into simulation platforms, and the evaluation of the overall framework. Each section focuses on interpreting the results in the context of the existing literature, highlighting contributions to the field, and identifying areas for improvement. Furthermore, the discussion explores the implications of the findings for practical applications, focusing on how the proposed framework can enhance workflows in urban planning, autonomous vehicle simulation, and related fields. This chapter concludes with a critical reflection on the limitations of the study and potential directions for future research.

5.1 Performance Evaluation

This subsection delves into the evaluation of the performance of the proposed methodologies and tools employed throughout this work. By assessing key metrics such as accuracy, computational efficiency, and robustness, the analysis aims to determine the effectiveness of the approaches in meeting the objectives outlined in previous chapters. The evaluation process includes comparisons with state-of-the-art methods, where applicable, to highlight strengths and identify areas for potential improvement. The results of these assessments provide a comprehensive understanding of the practical viability and scalability of the developed techniques, offering insights that are crucial for guiding future applications and research in the domain.

5.1.1 Mesh methodologies

Table 8 presents a detailed comparison of various tools and techniques for point clouds in 3D modeling. However, after further analysis, it was discovered that some methodologies listed in the table, such as *NeRF*, are primarily suited for mesh reconstruction rather than their original intended category. Subsequently, these methodologies were transferred to be studied in more appropriate sections of the investigation, ensuring a clear and accurate categorization of tools. Furthermore, certain methodologies listed in the table were determined to be incompatible with the present case, either due to technical limitations or lack of relevance to the study's specific objectives. This refinement process highlights the dynamic nature of tool selection in research and ensures that only the most suitable methodologies are explored in detail.

The same is applicable for frames 10 and 11. It provides an overview of various tools and techniques associated with mesh reconstruction, detailing their features such as the year of creation, last updates, pricing models, applicability, and purpose. Upon further analysis, it was discovered that some of the listed methodologies, such as *MeshLab* and *Geomagic*, primarily serve as post-processing tools rather than direct mesh reconstruction techniques. These tools were subsequently transferred to be studied within a separate section dedicated to post-processing methodologies. Additionally, certain tools, such as *Monomer Model*, were determined to be incompatible with the present study's objectives due to their specific or unrelated applications. This reclassification ensures a clearer distinction between methodologies and aligns the tools with their appropriate roles in the overall research framework.

3D reconstruction and point cloud processing are critical components in several fields such as computer vision, robotics, and virtual reality. The following analysis compares some of the leading tools and frameworks in this domain. These tools vary in their approach, performance, and application areas, making them suitable for different use cases.

Table 4 – Cloud to Mesh Comparison.

Name	**Mean**	**Std.**
Colmap	0.007060	0.1823
MicMac	10.4512	1.2348
Nerfstudio: Nerfacto	3.9745	1.3280
Nerfstudio: Instant-NGP	-5.1412	4.4478
Nerfstudio: 3D Gaussian Splatting	3.3637	1.1816

Source: Own Work (2025).

Table 4 presents a comparative analysis of different 3D reconstruction methods based on their Cloud-to-Mesh accuracy. The evaluation was conducted using Cloud Compare, a widely used software for point cloud processing and comparison. The table reports the Mean and Standard Deviation (Std.) values for each method, which were obtained by measuring the distances between the reconstructed point cloud and the generated mesh.

The Mean value represents the average deviation of the reconstructed model from the reference, indicating how closely the reconstruction aligns with the ground truth. A lower mean value suggests a more accurate reconstruction. The Standard Deviation (Std.) quantifies the dispersion of errors, reflecting the consistency of the reconstruction method—a lower standard deviation indicates fewer variations in accuracy.

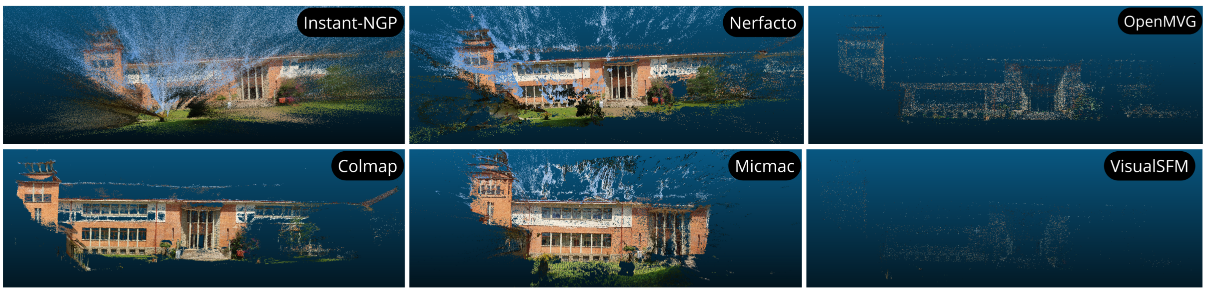
Colmap achieved a very low mean error (0.007060), indicating high accuracy in aligning the reconstructed mesh to the original cloud. Its standard deviation (0.1823) suggests minimal variability across different reconstructions. MicMac showed a significantly higher mean error (10.4512), implying a larger discrepancy between the mesh and the cloud. The lower standard deviation (1.2348) indicates consistent but less accurate results. Nerfstudio: Nerfacto yielded a moderate mean error (3.9745) and a standard deviation of 1.3280, reflecting reasonable accuracy and moderate variability in performance. Nerfstudio: Instant-NGP produced a negative mean error (-5.1412), which could indicate a systematic bias or deviation in the reconstruction process. The high standard deviation (4.4478) suggests substantial variability and less reliable results.

Nerfstudio: 3D Gaussian Splatting achieved a mean error of 3.3637 with a relatively low standard deviation (1.1816), demonstrating a balance between accuracy and consistency. It highlights the trade-offs between accuracy and consistency for each method. While Colmap stands out for its exceptional accuracy and low variability, other methods such as MicMac and Nerfstudio: Nerfacto offer alternative approaches with varying degrees of precision and consistency. These results provide a basis for selecting the most suitable reconstruction method based on specific project requirements and constraints.

5.1.2 Point cloud methodologies

Figure 18 and 19 presents the point clouds generated by different software tools and methodologies. Each software tool processes the input data differently, leading to variations in the quality, density, and completeness of the point clouds. As shown in the figure, Instant-NGP and Nerfacto offer high-speed reconstruction but may result in noisier outputs in comparison to tools like Colmap and Micmac, which provide denser and more detailed reconstructions at the cost of higher computational resources. OpenMVG and VisualSFM, while producing sparser point clouds, are still effective for specific applications, particularly those focused on camera pose estimation rather than dense surface generation. The comparison highlights the strengths and trade-offs of each tool in terms of processing time, reconstruction quality, and hardware requirements.

Figure 18 – Point clouds generated by different tools.

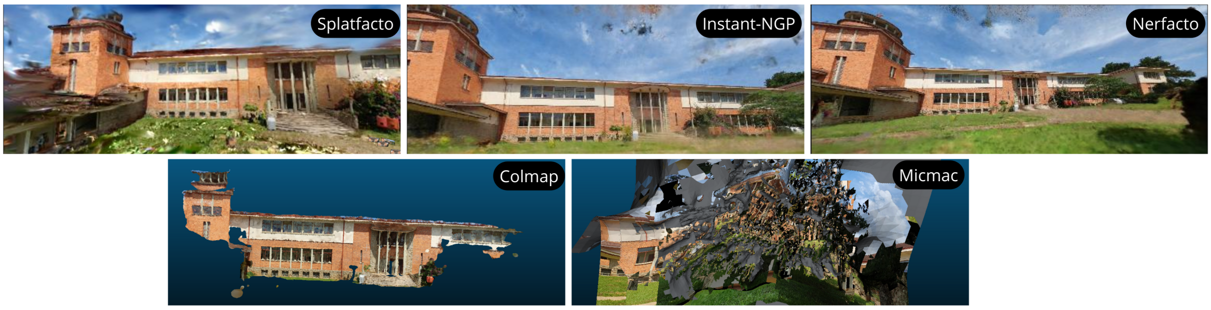


Source: Own Work (2025).

Table 5 presents the cloud-to-cloud distance computations, focusing on the mean distances and standard deviations between several 3D reconstruction methods: OpenMVG, Colmap, MicMac, Instant-NGP, Nerfacto, and VisualSFM. These values are essential for evaluating the accuracy of point cloud alignments generated by different tools.

OpenMVG demonstrates a notably high mean distance when compared with Colmap and shows higher variability in terms of standard deviation. This suggests that OpenMVG's reconstructions, when compared to Colmap, tend to deviate significantly in terms of point alignment, possibly due to differing underlying algorithms in feature detection and matching. However, in comparisons with MicMac, Instant-NGP, Nerfacto, and VisualSFM, OpenMVG shows relatively

Figure 19 – Textured meshes generated by different 3D reconstruction tools.



Source: Own Work (2025).

smaller mean distances, with values ranging from 0.66 to 1.71, indicating better alignment in these cases. Colmap, on the other hand, exhibits very high accuracy when compared with other methods such as Instant-NGP and Nerfacto. For instance, the mean distance between Colmap and Instant-NGP is 1.74 with a small standard deviation of 0.66, indicating consistent results across the point cloud. However, the comparisons with MicMac show relatively higher mean distances and deviations, indicating that the algorithms might handle specific features or areas differently, affecting the overall alignment. MicMac maintains good alignment with OpenMVG (0.61 mean distance) and Instant-NGP, indicating close consistency in the point cloud generation. However, a higher mean distance and standard deviation in comparison with Nerfacto show that these methods produce more divergent point clouds, likely due to differences in point cloud density or scene complexity.

Instant-NGP exhibits relatively consistent alignment with OpenMVG and Colmap, while showing greater divergence when compared with MicMac. The standard deviation in its comparison with Nerfacto suggests variability in the point clouds produced by these methods, potentially due to differences in how these tools handle scene reconstruction and surface texturing. Nerfacto shows the smallest mean distance when compared with Instant-NGP, indicating excellent alignment between these two methods, likely because of shared underlying techniques. However, the comparison with other methods, particularly MicMac and VisualSFM, reveals larger mean distances and higher standard deviations, suggesting discrepancies in how Nerfacto reconstructs scenes compared to these other tools. Finally, Visual SFM exhibits relatively stable performance with other methods, as shown by low standard deviations and mean distances compared with Colmap, Instant-NGP, and OpenMVG. However, like other methods, it shows greater divergence when compared with MicMac.

Nerfacto and Instant-NGP utilize an existing Neural Radiance Fields (NeRF) model to generate point clouds by leveraging neural representations to encode 3D scenes. NeRF models are designed to map 2D images to a volumetric representation by optimizing the rendering of each pixel based on camera poses. Instant-NGP speeds up this process through a multiresolution hash grid, allowing for faster training and rendering, while Nerfacto combines elements from various NeRF techniques to strike a balance between speed and reconstruction quality. By optimizing the NeRF model, they can effectively generate dense and accurate point clouds directly from neural scene representations, offering a more efficient approach to 3D reconstruction compared to traditional photogrammetry methods. This neural-based approach allows for faster processing times while maintaining a high level of detail in the reconstructed scene.

No single tool consistently outperforms the others across all metrics. Colmap and Instant-NGP exhibit strong accuracy and low variability when compared to many of the other tools, making them reliable for generating high-quality point clouds. OpenMVG performs well in certain scenarios but struggles with consistency when compared to Colmap. Nerfacto, while having close alignment with Instant-NGP, shows higher variability with other tools, possibly indicating its specialized applicability. These results suggest that the choice of software should depend on the specific requirements of the project, such as point cloud accuracy, consistency, and computational complexity. For tasks requiring high precision and consistent results, Colmap and Instant-NGP are likely the best choices. On the other hand, OpenMVG and VisualSFM are better suited for scenarios where speed or lower computational overhead is required, despite some variability in results.

Figure 19 demonstrates the textured meshes produced by different software tools, namely Splatfacto, Instant-NGP, Nerfacto, Colmap, and Micmac. Each tool employs unique methodologies for generating and applying textures to the 3D mesh. For example, neural rendering approaches like Splatfacto and Instant-NGP focus on generating photorealistic textures directly from 2D images by leveraging neural networks, resulting in highly realistic textures with smooth transitions. However, as seen with Nerfacto, while the textures are visually appealing, there are some artifacts that may appear in certain regions of the mesh.

Traditional methods, such as Colmap and Micmac, on the other hand, generate more geometrically accurate meshes but may have challenges with texture mapping, leading to less photorealistic outputs compared to neural methods. The differences between these approaches highlight the trade-offs between photorealism and geometric accuracy in mesh texturing. Table 5 presents the mean and standard deviation (Std) values for cloud-to-cloud distance computations, comparing the Colmap point cloud with other 3D reconstruction methods, including MicMac and several Nerfstudio-based approaches: Nerfacto, Instant-NGP, and 3D Gaussian Splatting.

Colmap, when compared to itself, naturally results in near-zero mean distances (0.007) and a very small standard deviation (0.182), indicating internal consistency within its own point cloud. This serves as a baseline for understanding deviations with other methods. The comparison between Colmap and MicMac reveals a significantly larger mean distance (10.45) with a moderate standard deviation (1.23). This indicates that MicMac's reconstruction deviates more substantially from Colmap's, likely due to differences in their underlying algorithms and how they handle feature detection and matching during the reconstruction process. Despite the higher mean, the moderate standard deviation suggests that while the two point clouds differ, the variation is relatively consistent across the cloud.

Table 5 – Distance Computation (Cloud-to-Cloud Distance)

Metric	Name	OpenMVG	Colmap	MicMac	Instant-NGP	Nerfacto	Visual SFM
Mean Dist.	**OpenMVG**	-	128.8810	**0.6091**	1.6108	4.4834	1.7084
	Colmap	6.5834	-	6.3426	7.1493	**1.8212**	**1.1484**
	MicMac	**0.6091**	9.2355	-	9.2674	8.5326	1.8027
	Instant-NGP	1.6108	**1.7416**	1.6873	-	2.6043	2.4443
	Nerfacto	4.4834	1.8212	8.5326	2.6043	-	2.3806
	Visual SFM	1.7084	4.3091	9.3365	**1.4993**	4.1649	-
Std Dev.	**OpenMVG**	-	481.8060	0.1102	0.6300	10.1742	0.7694
	Colmap	1.1834	-	1.4531	1.0414	**0.8379**	**0.4012**
	MicMac	**0.1102**	2.3317	-	1.9322	33.6128	0.5149
	Instant-NGP	0.6253	**0.6608**	**0.3901**	-	9.6101	0.6403
	Nerfacto	10.1742	0.8379	33.6128	2.6043	-	0.7091
	Visual SFM	0.7694	0.7640	1.9163	**0.6872**	9.8412	-

Source: Own Work (2025).

Nerfstudio's Nerfacto model shows a mean distance of 3.97 and a standard deviation of 1.32. This result suggests a closer alignment between Nerfacto and Colmap compared to MicMac, although the deviation is still notable. Nerfacto's combination of neural techniques appears to produce reasonably accurate reconstructions but with slightly more noise, as indicated by the higher standard deviation compared to Colmap's own values. On the other hand, Instant-NGP, another Nerfstudio approach, demonstrates a negative mean distance (-5.14) with a high standard deviation (4.45). The negative mean suggests a systematic offset between the two point clouds, and the large standard deviation reflects considerable variability in alignment. This result suggests that while Instant-NGP can generate fast reconstructions, its output may suffer from inconsistencies when compared to Colmap, which could impact its use in tasks requiring high accuracy.

The 3D Gaussian Splatting technique from Nerfstudio exhibits a relatively low mean distance (3.36) and a standard deviation of 1.18, indicating better overall alignment with Colmap compared to Instant-NGP and Nerfacto. This suggests that 3D Gaussian Splatting produces more accurate reconstructions with respect to Colmap, and it may provide a more consistent output in terms of surface alignment, offering a good balance between speed and accuracy. Colmap remains the most consistent and geometrically accurate method, serving as a strong baseline for comparison. MicMac, while differing significantly, remains a viable alternative with consistent variation in alignment. Nerfstudio's approaches, particularly 3D Gaussian Splatting, demonstrate promise in terms of accuracy but show more variability compared to traditional photogrammetry tools. Instant-NGP, though efficient, shows greater inconsistency, which may limit its applicability in precise tasks. These findings suggest that the choice of reconstruction tool should depend on the specific requirements of the task, with Colmap being favored for accuracy and consistency, and Nerfstudio's models offering faster but slightly less precise alternatives.

5.2 Simulation Tool Selection and Implementations

The selection of an appropriate simulation tool is a critical step in ensuring the success and reliability of this study. This subsection outlines the criteria and process for identifying a simulation platform that aligns with the requirements of 3D reconstruction and urban modeling. Factors such as ease of integration, computational efficiency, extensibility, and support for realistic physics and rendering were considered. By analyzing a variety of tools against these benchmarks, the aim was to select a platform that not only meets the immediate project needs but also provides flexibility for future expansions and experiments.

Based on the data presented in Table 3, the simulators were further investigated in depth, and some practical tests were conducted to evaluate their capabilities. Following this detailed analysis and hands-on experimentation, the CARLA simulator was selected as the most suitable option for this study, owing to its robust features and compatibility with the project's requirements.

CARLA is an open-source simulator designed for autonomous driving research, providing a versatile platform for testing and development (Dosovitskiy *et al.*, 2017b). Built on Unreal Engine, it offers a high-fidelity virtual environment that supports various urban settings, weather conditions, and dynamic scenarios. CARLA provides detailed 3D maps, sensor simulation (such as LiDAR, radar, and cameras), and an array of pre-designed vehicle models to mimic real-world behavior. The simulator enables integration with machine learning frameworks, making it a powerful tool for training and validating autonomous systems. Its modular design allows researchers to customize scenarios and parameters, ensuring comprehensive evaluation of algorithms under controlled and diverse conditions. By fostering a collaborative ecosystem, CARLA has become a cornerstone for academia and industry in advancing autonomous vehicle technologies.

Unreal Engine provides advanced rendering capabilities, scene adjustments, and debugging tools that allow for fine-tuning the 3D model before it is imported into CARLA. Integrating the 3D reconstruction into Unreal Engine is a crucial step to ensure that the reconstructed environment is properly formatted, optimized, and visually accurate before being used in autonomous driving simulations. By first integrating the reconstruction in Unreal Engine, it is possible to check for geometry inconsistencies, material and texture issues, and proper scaling of the environment, ensuring that the model accurately represents real-world conditions. Unreal Engine's real-time visualization and interactive scene-editing tools facilitate collision detection, lighting adjustments, and environmental optimizations, which are essential for creating a realistic driving simulation. Once the environment is successfully set up in Unreal, it can then be exported in a format compatible with CARLA, including the necessary .xodr (OpenDRIVE) files for road definition and navigation data. This workflow streamlines the integration process, reducing errors and ensuring that the final CARLA simulation runs smoothly, with a well-structured and high-fidelity driving environment.

5.2.1 Importing Reconstructed Mesh into Unreal Engine using 3D Gaussian Splatting

This subsection describes the process of importing a reconstructed 3D mesh into the Unreal Engine environment using 3D Gaussian Splatting (3DGS) and Volinga's plugin. This approach facilitates high-quality scene reconstruction by leveraging recent advancements in Neural Radiance Fields (NeRF) and Gaussian Splatting techniques.

5.2.1.1 Overview of Volinga Plugin

The integration of 3DGS models into Unreal Engine is facilitated by Volinga's tools, which consist of two main components. Each tool plays a critical role in the workflow, ensuring a smooth transition from raw reconstruction data to a Unreal Engine environment. Volinga Creator is a tool designed to generate 3DGS models from images or video input. The process consists of the following steps:

1. **Input Data Collection:** The tool processes multi-view images or videos of a real-world scene.
2. **3D Gaussian Reconstruction:** A neural network-based approach is used to generate a 3D scene composed of thousands of Gaussian splats.
3. **Optimization and Compression:** The model undergoes refinements to enhance visual fidelity while maintaining efficient rendering performance.
4. **Export as a 3DGS Model:** The resulting scene is stored in a format that retains the volumetric nature of Gaussian splats, ensuring high-quality rendering with minimal computational cost.

Unlike traditional 3D reconstruction methods that generate polygonal meshes (e.g., .FBX, .OBJ), Volinga Creator focuses on a volumetric representation that excels in real-time rendering. Once the reconstruction is complete, it might be imported into Unreal Engine. However, Unreal Engine does not natively support Gaussian Splatting models. This is where Volinga Import plays a crucial role. Volinga Import is a tool that converts the 3DGS or NeRF reconstructions into an .nvol file, which is compatible with Unreal Engine's rendering pipeline. The conversion process follows these steps:

1. **Load the 3DGS Model into Volinga Import:** The Gaussian Splatting reconstruction generated by Volinga Creator is imported.
2. **Generate an .nvol File:** The tool converts the 3DGS data into the **.nvol format**, which is optimized for Unreal Engine's visualization engine.

3. **Import the .nvol File into Unreal Engine:** Using the Volinga Plugin for Unreal, the converted scene is loaded into the engine.

4. **Adjust Scene Properties:** Scaling, lighting, and rendering settings are adjusted within Unreal Engine to ensure proper integration.

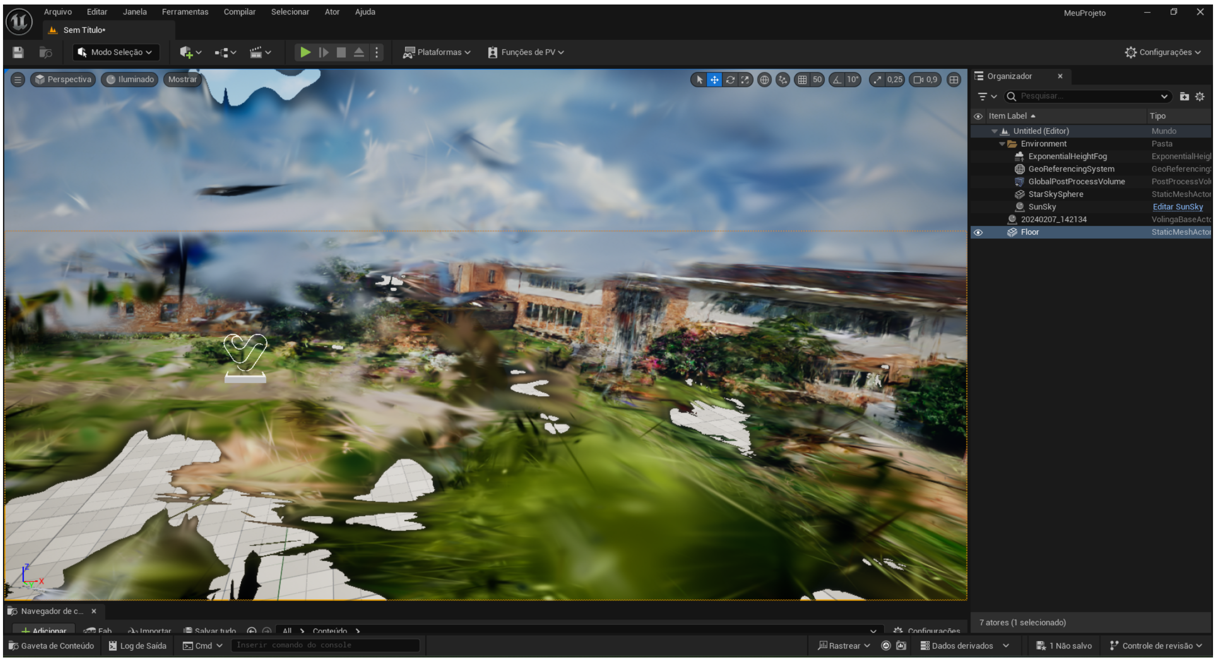
A test was conducted using a NeRF-based mesh generated with Nerfstudio to evaluate its feasibility for integration within Unreal Engine. However, it was found that Volinga no longer provides support for NeRF models, limiting the direct use of these reconstructions in the workflow. While NeRF remains a powerful approach for scene reconstruction, its lack of compatibility with Volinga's import pipeline presented a challenge for integration. As a result, alternative methods, such as 3D Gaussian Splatting (3DGS), were explored, offering a more suitable approach for importing and rendering reconstructed environments within the simulation framework.

Figure 20 illustrates the successful integration of the UTFPR 3DGS mesh into Unreal Engine, showcasing the reconstructed environment within the simulation platform. The 3D mesh was generated using Volinga Creator, which processes image-based 3D reconstructions into 3D Gaussian Splatting (3DGS) meshes. The generated mesh was then imported into Unreal Engine through Volinga Import, which converts the reconstruction into an .nvol file, allowing for seamless visualization and interaction within the engine. This step was crucial for evaluating the geometry, texture accuracy, and overall fidelity of the reconstruction before integrating it into CARLA for autonomous driving simulations. Unreal Engine's real-time rendering capabilities enabled further optimizations, such as collision adjustments, material refinements, and lighting enhancements, ensuring that the environment is suitable for realistic driving simulations. The integration of the UTFPR 3DGS mesh in Unreal serves as an essential intermediary step, allowing for detailed scene validation and refinement before exporting it for use in CARLA.

For the initial proof of concept, a low-quality 3DGS mesh was used to validate the integration process within Unreal Engine before proceeding with higher-fidelity reconstructions. This preliminary mesh served as a functional test to ensure compatibility between the Volinga-generated 3D Gaussian Splatting (3DGS) model and the simulation pipeline. Despite its reduced level of detail and potential artifacts, it provided valuable insights into the workflow, importation process, and rendering capabilities within Unreal. The use of a simplified mesh allowed for faster processing and testing, confirming that the methodology was feasible before committing to more computationally expensive high-resolution models for full-scale simulation.

With the reconstructed environment successfully imported into Unreal Engine, the next step is to integrate it with CARLA. This process involves. Converting the 3DGS model into CARLA-compatible assets while preserving key features, generating OpenDRIVE (.xodr) files to define road structures and navigation paths and setting up collision detection and physics interactions for autonomous vehicle simulations. By leveraging Volinga's tools and Unreal Engine, this workflow enables realistic scenarios that can be used for autonomous vehicle research,

Figure 20 – UTFPR 3DGS Mesh into Unreal Engine.



Source: Own Work (2025).

sensor simulation, and AI training. The steps followed for importing the reconstructed mesh into Unreal Engine are as follows:

1. **Generating the 3D Reconstruction:** The environment was reconstructed using a 3DGS-based approach, ensuring high visual fidelity and efficient rendering performance.

2. **Exporting the 3DGS Model via Volinga:** The reconstructed Gaussian Splatting model was exported using Volinga's plugin, which converts the model into an Unreal Engine-compatible format.

3. **Importing into Unreal Engine:** Using the Volinga plugin, the exported model was loaded into Unreal Engine. This process ensures that the spatial consistency and visual details of the original 3D reconstruction are maintained.

4. **Aligning the Imported Mesh:** After import, the reconstructed scene was centered at coordinates (0,0,0) to align correctly within the Unreal Engine world space.

5.3 Pipelines for 3D Map Generation

The generation of 3D maps for virtual environments and driving simulators relies on structured pipelines that integrate several stages, ranging from data acquisition to final simulation deployment. These pipelines aim to create high-fidelity digital replicas of real-world environments, ensuring accurate representation of roads, buildings, vegetation, and other key features required for realistic autonomous vehicle (AV) testing. Each step is critical to ensuring geometric accuracy, texture realism, and compatibility with simulation platforms, such as Unreal Engine and CARLA. Table 15 summarizes the proposed pipeline used in this work, demonstrating how datasets acquired through image capture are processed into point clouds, reconstructed into textured meshes, and finally integrated into simulation environments for ADAS validation and testing.

Frame 15 – Steps for 3D Environment and Scenario Creation in SCANeR Studio

Step	Title	Description
1	Requirements Gathering	Collection of customer requirements regarding the virtual environment, including road types, urban/rural settings, objects, vehicles, pedestrians, and signage.
2	Project Proposal & Planning	Definition of project stages, timeline, partial and final deliverables, and formal approval of the project scope by the customer.
3	3D Environment Modeling	Creation of 3D elements such as: road infrastructure, buildings, urban elements, environmental conditions (weather, lighting), and landscape elements.
4	Scenario Configuration	Setup of specific scenarios, including traffic flow, dynamic vehicles, pedestrian paths, event triggers (accidents, detours, adverse weather conditions).
5	Integration into SCANeR Studio	Importation of the 3D environment and configured scenarios into SCANeR Studio. Association with vehicle models, ADAS features, and simulation parameters.
6	Testing & Validation	Execution of validation tests within SCANeR Studio to check consistency, positioning accuracy, interaction between vehicles, pedestrians, and infrastructure.
7	Customer Review & Delivery	Presentation to the customer, incorporation of feedback, and final delivery of the fully integrated and functional environment and scenarios.

Source: Adapted from AVSimulation Documentation (2025).

Table 16 presents the complete pipeline designed for generating and integrating 3D environments into the CARLA simulator, ensuring compatibility with autonomous vehicle testing scenarios. The process begins with the acquisition of image datasets from real-world locations, captured using a handheld camera to facilitate flexible and cost-effective data collection. These images are processed using COLMAP, which generates dense point clouds, replacing the need for more expensive LiDAR systems while preserving geometric accuracy. The point clouds are subsequently transformed into textured meshes, capturing both geometry and surface appearance. These meshes are imported into Unreal Engine for initial spatial adjustments and visual inspections. A critical step in the process is the creation of a material and texture classification system, which automatically identifies key surface types — such as roads, buildings, sidewalks, and vegetation — ensuring proper semantic segmentation required for CARLA's sensors. Additionally, two essential files are generated: a .fbx file containing the environment's 3D geometry, and a .xodr file containing the road network definition in the OpenDRIVE format, allowing CARLA to simulate traffic behavior and vehicle dynamics accurately. Finally, the reconstructed and classified environment is integrated into CARLA, where ADAS validation logic is implemented, enabling comprehensive testing of autonomous driving functionalities in a virtual replica of real-world locations.

Frame 16 – 3D Environment Creation and Integration Process with CARLA

Step	Title	Description
1	Dataset Acquisition	Capture images of the target environment using a handheld camera, ensuring coverage from multiple angles for later reconstruction.
2	Point Cloud Generation with COLMAP	Process images in COLMAP to generate a 3D point cloud, serving as an alternative to traditional LiDAR scanning.
3	Textured Mesh Creation	Convert the point cloud into a fully textured 3D mesh, refining geometry and applying realistic textures based on the captured images.
4	Import into Unreal Engine	Import the textured 3D mesh into Unreal Engine for visualization, spatial adjustments, and preliminary compatibility checks.
5	Material and Texture Classification System	Develop a classification system to automatically identify and label materials and textures, distinguishing roads, buildings, vegetation, sidewalks, and other elements critical for simulation realism.
6	Generation of CARLA Files (.fbx and .xodr)	Create the two required files for CARLA integration: a .fbx file containing the 3D geometry and a .xodr file with OpenDRIVE road network data, ensuring CARLA's traffic logic compatibility.
7	Unreal to CARLA Integration	Integrate the Unreal environment into CARLA, ensuring correct alignment, scale, collision settings, and semantic segmentation compatibility within the simulator.
8	ADAS Validation Logic Development	Create and implement logic for ADAS validation within CARLA, allowing the evaluation of autonomous systems' responses to the reconstructed environment's features, traffic, and events.

Source: Own Work (2025).

The pipeline developed for the creation and integration of 3D environments into CARLA represents a significant contribution to research and testing in the field of autonomous vehicles. By replacing traditional data capture methods based on LiDAR sensors with image-based reconstruction techniques, the proposed methodology reduces costs and increases accessibility for creating realistic urban environments for simulation purposes. The integration with Unreal

Engine allows for visual adjustments and mesh validation before exporting to CARLA, ensuring that the virtual environment preserves not only the geometry but also essential texture features for perception simulations. Furthermore, the automatic classification of materials and the generation of the required CARLA-compatible files (.fbx and .xodr) offer a structured workflow that facilitates the creation of dynamic, semantically enriched scenarios, crucial for the development and validation of ADAS systems and autonomous driving algorithms. By consolidating all these steps into a cohesive process, the created pipeline optimizes development time, improves consistency across different test scenarios, and enables the reproduction of real-world environments in controlled simulations, bringing virtual assessments closer to real-world conditions.

5.4 Conclusion of Results

The results obtained in this study provide a comprehensive evaluation of the methodologies and technologies employed for 3D reconstruction and integration into simulation environments. By systematically analyzing different reconstruction techniques, evaluating their accuracy, efficiency, and suitability for simulation, and testing their integration within Unreal Engine and CARLA, this research has yielded valuable insights into the feasibility and practical challenges of incorporating reconstructed environments into autonomous vehicle simulations.

The comparative analysis of 3D reconstruction methods, including COLMAP, MicMac, NeRF-based techniques, and 3D Gaussian Splatting (3DGS), demonstrated significant variations in reconstruction accuracy and computational requirements. The Cloud Compare evaluation highlighted how point cloud generation and mesh reconstruction approaches differ in terms of precision and standard deviation, providing quantitative data that supports the selection of appropriate techniques based on specific application needs. While some methods delivered highly detailed reconstructions, their computational cost and integration complexity posed challenges for real-time simulation applications.

The integration of reconstructed environments into simulation platforms proved to be a key aspect of this research. The workflow of importing reconstructed 3DGS meshes into Unreal Engine and subsequently preparing them for CARLA was extensively tested. A significant finding was that CARLA requires additional file formats for seamless integration, reinforcing the necessity of pre-processing and optimizing reconstructed meshes within Unreal Engine before final importation. Furthermore, the Volinga plugin was tested for NeRF-based reconstructions, but its lack of continued support for NeRF models limited its usability for this research.

Additionally, the study revealed the importance of selecting an appropriate simulator for the project. Among the tested platforms, CARLA was chosen due to its open-source nature, strong support for autonomous driving research, and high compatibility with 3D models. However, other simulators, such as SUMO and LGSVL, were evaluated and found to be less suited for integrating high-quality 3D reconstructions. The findings from these experiments highlight the necessity of developing a streamlined and automated pipeline for importing reconstructed models into simulation environments, a key area for future improvements.

In summary, the results validate the feasibility of integrating 3D reconstructed environments into simulation frameworks while emphasizing the challenges that remain, particularly in mesh optimization, real-time rendering, and automated integration pipelines. The insights gained from this research not only contribute to the advancement of simulation-based autonomous vehicle testing but also establish a foundation for further improvements in 3D reconstruction methodologies, simulator compatibility, and performance optimization. Future research will focus on enhancing reconstruction quality, improving the automation of the integration process, and expanding simulation capabilities to include dynamic environmental factors for more realistic and scalable autonomous vehicle simulations.

6 CONCLUSION

This dissertation presented an extensive study on 3D reconstruction and simulation methodologies, aiming to advance the integration of realistic virtual environments into autonomous vehicle simulations. The research covered a broad spectrum of topics, from theoretical concepts in reconstruction techniques to practical implementations and evaluations using state-of-the-art tools and frameworks. By addressing key challenges in the field, this work provides valuable insights that contribute both to academic research and industrial applications. A major focus of this research was the development and assessment of datasets tailored for 3D reconstruction. A structured approach to data collection was followed, including sensor selection, data acquisition, and preprocessing, which led to the creation of a robust dataset. This included a full dataset as well as a smaller subset optimized for rapid testing. The use of real-world data from UTFPR further enhanced the practical applicability of the study, ensuring that the findings are relevant to real-world scenarios.

A thorough analysis of various reconstruction techniques was provided, including point cloud generation, mesh reconstruction, and post-processing methodologies. Tools such as COLMAP, NeRF, 3D Gaussian Splatting, and MicMac were evaluated in detail, allowing for a comprehensive understanding of their strengths and weaknesses. The comparison between these methods, conducted using Cloud Compare, highlighted their accuracy, computational efficiency, and feasibility for simulation integration. This analysis contributed to identifying the most suitable techniques for creating high-fidelity 3D environments. One of the central challenges addressed in this research was the integration of 3D reconstructed models into simulation platforms, particularly Unreal Engine and CARLA. The study initially focused on importing the reconstructed models into Unreal Engine as an intermediary step before integrating them into CARLA, due to the additional file requirements and compatibility constraints of the CARLA simulator. A key finding was that manual integration requires specific file formats (.fbx and .xodr), which necessitates a two-step process: preparing the reconstructed mesh in Unreal Engine before importing it into CARLA. This insight provides a practical guideline for future researchers working on similar integrations.

Furthermore, simulation tools were explored to assess their suitability for working with reconstructed environments. CARLA was ultimately selected as the primary simulation platform due to its open-source nature, realistic physics, and strong support for autonomous vehicle research. Other simulators, such as SUMO and LGSVL, were considered but exhibited limitations in their ability to handle detailed 3D environments. The research also involved practical testing with Volinga, a tool that facilitates the importation of 3D Gaussian Splatting (3DGS) meshes into Unreal Engine. However, experiments with NeRF-based meshes revealed that Volinga no longer supports NeRF, limiting its use for specific reconstruction approaches. This dissertation also introduced a systematic literature review (SLR), conducted using the PRISMA methodology, to identify and categorize the most relevant research on 3D reconstruction and simulation techniques. The NVivo software was utilized to manage and analyze qualitative data from the literature review, allowing for a structured synthesis of existing methodologies and technologies. The dual focus of the SLR—on both reconstruction and simulation approaches—ensured a comprehensive overview of the field, facilitating a well-informed selection of tools and techniques.

Despite the progress made in this research, several areas remain open for further exploration and improvement. A crucial future direction involves enhancing the quality of reconstructed environments by incorporating higher-resolution datasets and leveraging more advanced reconstruction techniques, such as neural radiance fields (NeRF) and deep learning-based point cloud processing. Additionally, automating the integration pipeline between reconstruction tools and simulation platforms like CARLA remains a key challenge. Developing a standardized workflow or software tool that streamlines this process would significantly improve efficiency and accessibility. Another promising avenue is the validation of reconstructed environments by comparing them against real-world sensor data, ensuring that virtual simulations accurately represent physical environments. Expanding the scope of this research to include dynamic elements, such as real-time traffic interactions and environmental conditions, would further improve the realism and applicability of the simulation framework. Integrating sensor fusion techniques—combining LiDAR, RGB cameras, and depth sensors—could enhance reconstruction accuracy and overcome limitations associated with individual sensor modalities.

This dissertation has contributed a validated methodology for 3D reconstruction and simulation, bridging the gap between real-world data acquisition and virtual environment simulation. By addressing fundamental challenges in this domain and proposing practical solutions, this work provides a strong foundation for future advancements. The findings not only enhance our understanding of current reconstruction and simulation technologies but also pave the way for further innovation, particularly in the integration of automated reconstruction-to-simulation pipelines. The impact of 3D reconstruction technologies extends beyond autonomous vehicle research. The methodologies developed and tested in this study can be applied to various domains, including urban planning, digital twins, cultural heritage preservation, and virtual reality applications. As technology advances, interdisciplinary collaboration between computer vision, machine learning, and simulation engineering will play a key role in refining and expanding the capabilities of 3D reconstruction for real-world applications. Ultimately, this work underscores the transformative potential of 3D reconstruction technologies in shaping the future of simulation, autonomous systems, and digital modeling. With continued research and technological development, the integration of highly realistic reconstructed environments into real-time simulations will become a crucial element in the evolution of autonomous systems and smart city technologies.

BIBLIOGRAPHY

ABEBE, G. Lidar systems and its applications for 3d city modeling: A review, . v. 7„ p. 2023, 09 2024.

AG, S. Industrie 4.0: Siemens demonstrates digital twin in actual operation. **Siemens Industry Journal**, „, 2019. Disponível em: https://press.siemens.com/global/en/pressrelease/industrie-40-siemens-demonstrates-digital-twin-actual-operation.

Agisoft. **Metashape Professional Edition User Manual, Version 2.1**. [*S.l.*], 2023. Accessed: 2024-08-01. Disponível em: https://www.agisoft.com.

AL-ZOUBE, M. Efficient vision-based multi-target augmented reality in the browser. **Multimedia Tools and Applications**, v. 81„ p. 1–18, 04 2022.

ALEXANDRE, R. **Primeiro simulador veicular da América Latina ajuda a desenvolver soluções para o carro e seu ecossistema**. [*S.l.*]: , 2017. Acessado em: 12 nov. 2024. Disponível em: https://www.projetodraft.com.

AVSimulation. **SCANeR: Driving Simulation Software**. [*S.l.*]: , 2024. Acessado em: 12 nov. 2024. Disponível em: https://www.avsimulation.com/en/scaner/.

AZFAR, M.; SMITH, J.; KHAN, A. Traffic co-simulation framework: Integrating infrastructure camera sensing with reinforcement learning-based vehicle control. **IEEE Transactions on Intelligent Transportation Systems**, v. 25, n. 3, p. 1234–1245, 2024.

BAGDASARIAN, M. T. *et al.* 3dgs. zip: A survey on 3d gaussian splatting compression methods. **arXiv preprint arXiv:2407.09510**, „, 2024.

BALLABENI, A. *et al.* Advances in image pre-processing to improve automated 3d reconstruction. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Copernicus GmbH v. 40„ p. 315–323, 2015.

BÉCUE, A. *et al.* A new concept of digital twin supporting optimization and resilience of factories of the future. **Applied Sciences**, MDPI v. 10, n. 13, p. 4482, 2020.

BEHRISCH, M. *et al.* Sumo - simulation of urban mobility: An overview. *In*: PROCEEDINGS OF SIMUL 2011, THE THIRD INTERNATIONAL CONFERENCE ON ADVANCES IN SYSTEM SIMULATION. 2011. **Anais [...]** [*s.n.*], 2011. p. 55–60. Disponível em: https://www.eclipse.org/sumo/.

BENGLER, K. *et al.* Three decades of driver assistance systems: Review and future perspectives. **IEEE Intelligent Transportation Systems Magazine**, IEEE v. 6, n. 4, p. 6–22, 2014.

BERNARDINI, F. *et al.* The ball-pivoting algorithm for surface reconstruction. **IEEE transactions on visualization and computer graphics**, IEEE v. 5, n. 4, p. 349–359, 1999.

BESL, P. J.; MCKAY, N. D. A method for registration of 3-d shapes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE v. 14, n. 2, p. 239–256, 1992.

BOUAIN, M. *et al.* An embedded multi-sensor data fusion design for vehicle perception tasks. **Journal of Communications**, v. 13„ p. 18–14, 01 2018.

BUYUKDEMIRCIOGLU, M.; KOCAMAN, S. Reconstruction and efficient visualization of heterogeneous 3d city models. **Remote Sensing**, v. 12, n. 13, p. 2128, 2020. Disponível em: https://www.mdpi.com/2072-4292/12/13/2128.

CALANTROPIO, A. *et al.* Image pre-processing strategies for enhancing photogrammetric 3d reconstruction of underwater shipwreck datasets. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Copernicus GmbH v. 43„ p. 941–948, 2020.

CARLA Simulator Team. **CARLA Simulator Documentation: Getting Started Guide (Version 0.9.5)**. [*S.l.*], n.d. Retrieved from https://carla.readthedocs.io/en/0.9.5/getting_started/.

CARR, J. C.; FRIGHT, W. R.; BEATSON, R. K. Surface interpolation with radial basis functions for medical imaging. **IEEE transactions on medical imaging**, IEEE v. 16, n. 1, p. 96–107, 1997.

CERNEA, D. **OpenMVS: Multi-View Stereo Reconstruction Library**. [*S.l.*]: , 2020. Disponível em: https://cdcseacave.github.io/openMVS.

CHEN, W.; ZHANG, M.; LIU, K. Virtual-real-fusion for intelligent 3d traffic accident reconstruction. **Journal of Autonomous Systems and Simulations**, v. 10, n. 1, p. 98–112, 2025.

CHENG, W. *et al.* A Data-Driven Point Cloud Simplification Framework for City-Scale Image-Based Localization. **IEEE Transactions on Image Processing**, v. 26, n. 1, p. 262–275, 2016.

CHERIET, F. *et al.* A novel system for the 3-d reconstruction of the human spine and rib cage from biplanar x-ray images. **IEEE transactions on biomedical engineering**, IEEE v. 54, n. 7, p. 1356–1358, 2007.

CIGNONI, P. *et al.* MeshLab: an Open-Source Mesh Processing Tool. *In*: SCARANO, V.; CHIARA, R. D.; ERRA, U. (Ed.). EUROGRAPHICS ITALIAN CHAPTER CONFERENCE. 2008. **Anais [...]** [*S.l.*]: The Eurographics Association, 2008. ISBN 978-3-905673-68-5.

CRAMPEN, R.; ZHAO, M.; LIN, W. 3d urban digital twins in unreal engine 5 for real-time traffic monitoring. **Smart Urban Systems Journal**, v. 18, n. 4, p. 355–370, 2024.

DAVID, C.; MA, Y. Lidar-based 3d reconstruction for autonomous vehicles: State-of-the-art and challenges. **IEEE Transactions on Intelligent Transportation Systems**, v. 21, n. 4, p. 1468–1485, 2020.

DAVIS, C. *et al.* 3D Modeling of Cities for Virtual Environments. *In*: 2021 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA). 2021. **Anais [...]** [*S.l.*: *s.n.*], 2021. p. 5587–5596.

DICKERSON, M. T. *et al.* Fast greedy triangulation algorithms. *In*: PROCEEDINGS OF THE TENTH ANNUAL SYMPOSIUM ON COMPUTATIONAL GEOMETRY. 1994. **Anais [...]** [*S.l.*: *s.n.*], 1994. p. 211–220.

DOSOVITSKIY, A. *et al.* Carla: An open urban driving simulator. **arXiv preprint arXiv:1711.03938**, „, 2017.

DOSOVITSKIY, A. *et al.* CARLA: An open urban driving simulator. *In*: PROCEEDINGS OF THE 1ST ANNUAL CONFERENCE ON ROBOT LEARNING. 2017. **Anais [...]** [*S.l.*: *s.n.*], 2017. p. 1–16.

ESRAFILIAN, O.; GESBERT, D. 3d city map reconstruction from uav-based radio measurements. *In*: GLOBECOM 2017 - 2017 IEEE GLOBAL COMMUNICATIONS CONFERENCE. 2017. **Anais [...]** [*S.l.*: *s.n.*], 2017. p. 1–6.

FEHéR, A. Using 3d scanners in archaeology. **Hungarian Archaeology E-Journal**, v. 2013, n. Summer, p. 1–5, 2013. Disponível em: https://www.researchgate.net/publication/315783869.

FERDANI, D. *et al.* 3d reconstruction and validation of historical background for immersive vr applications and games: The case study of the forum of augustus in rome. **Journal of Cultural Heritage**, v. 43,, p. 129–143, 2020. ISSN 1296-2074. Disponível em: https://www.sciencedirect.com/science/article/pii/S1296207419307964.

FRUH, C.; ZAKHOR, A. 3D model generation for cities using aerial photographs and ground level laser scans. *In*: PROCEEDINGS OF THE 2001 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. CVPR 2001. 2., 2001. **Anais [...]** [*S.l.*: *s.n.*], 2001. p. II—-II.

FURUKAWA, Y.; PONCE, J. Towards internet-scale multi-view stereo. **Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)**, „ p. 1434–1441, 2010.

GALLIANI, S.; LASINGER, K.; SCHINDLER, K. Massively parallel multiview stereopsis by surface normal diffusion. *In*: 2015 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV). 2015. **Anais [...]** [*S.l.*: *s.n.*], 2015.

GEIGER, A.; LENZ, P.; STILLER, C. Vision meets robotics: The kitti dataset. **The International Journal of Robotics Research**, v. 32, n. 11, p. 1231–1237, 2013. Disponível em: https://journals.sagepub.com/doi/10.1177/0278364913491297.

GOODALL, N. J. Machine learning and intelligent transportation systems. **Transportation Research Part C: Emerging Technologies**, Elsevier v. 41,, p. 256–264, 2014.

GRIEVES, M.; VICKERS, J. Origins of the digital twin concept. **Research-Technology Management**, v. 57, n. 3, p. 10–15, 2014.

HALL, B. *et al.* Networked and Multimodal 3D Modeling of Cities for Collaborative Virtual Environments. *In*: 2022 IEEE/ACM INTERNATIONAL CONFERENCE ON BIG DATA COMPUTING, APPLICATIONS AND TECHNOLOGIES (BDCAT). 2022. **Anais [...]** [*S.l.*: *s.n.*], 2022. p. 204–212.

HALL, D. Lidar-based advanced driver assistance. **IEEE Spectrum**, v. 48, n. 5, p. 24–29, 2011.

HAN, J.; SHEN, S. Distributed Surface Reconstruction from Point Cloud for City-Scale Scenes. *In*: 2019 INTERNATIONAL CONFERENCE ON 3D VISION (3DV). 2019. **Anais [...]** [*S.l.*: *s.n.*], 2019. p. 338–347.

HAN, J. *et al.* Urban scene lod vectorized modeling from photogrammetry meshes. **IEEE Transactions on Image Processing**, IEEE v. 30,, p. 7458–7471, 2021.

HIRSCHMüLLER, H. Stereo processing by semiglobal matching and mutual information. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE v. 30, n. 2, p. 328–341, 2008.

HU, Y. *et al.* Dense 3D Model Reconstruction for Digital City Using Computationally Efficient Multi-View Stereo Networks. *In*: IGARSS 2022 - 2022 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM. 2022. **Anais [...]** [*S.l.*: *s.n.*], 2022. p. 959–962.

HUANG, J. *et al.* City3D: Large-Scale Building Reconstruction from Airborne LiDAR Point Clouds. **Remote Sensing**, v. 14, n. 9,, 2022. ISSN 2072-4292. Disponível em: https://www.mdpi.com/2072-4292/14/9/2254.

IBRAHIM, Y.; NAGY, B.; BENEDEK, C. Multi-view based 3d point cloud completion algorithm for vehicles. *In*: 2022 26TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR). 2022. **Anais [...]** [*S.l.*: *s.n.*], 2022. p. 2121–2127.

IONASEC, R. I. *et al.* Patient-specific modeling and quantification of the aortic and mitral valves from 4-d cardiac ct and tee. **IEEE Transactions on Medical Imaging**, IEEE v. 29, n. 9, p. 1636–1651, 2010.

KAZHDAN, M.; BOLITHO, M.; HOPPE, H. Poisson surface reconstruction. *In*: PROCEEDINGS OF THE FOURTH EUROGRAPHICS SYMPOSIUM ON GEOMETRY PROCESSING. 7 n. 4., 2006. **Anais [...]** [*S.l.*: *s.n.*], 2006.

KULAWIAK, M. A cost-effective method for reconstructing city-building 3d models from sparse lidar point clouds. **Remote Sensing**, v. 14, n. 5, p. 1278, 2022. Disponível em: https://www.mdpi.com/2072-4292/14/5/1278.

KUMAR, A. *et al.* Citywide reconstruction of cross-sectional traffic flow from moving camera videos. *In*: 2021 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA). 2021. **Anais [...]** [*S.l.*: *s.n.*], 2021. p. 1670–1678.

KUNDU, A.; LI, Y.; REHG, J. M. 3D-RCNN: Instance-Level 3D Object Reconstruction via Render-and-Compare. *In*: 2018 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. 2018. **Anais [...]** [*S.l.*: *s.n.*], 2018. p. 3559–3568.

KüHNER, T.; KüMMERLE, J. Large-scale volumetric scene reconstruction using lidar. **IEEE International Conference on Robotics and Automation (ICRA)**, IEEE v. 31,, p. 6261–6267, 2024.

LAUSSEDAT, A. **La Métrophotographie: ou, Application de la Photographie à l'arpentage**. [*S.l.*]: Paris, 1850.

LAW, A. M. **Simulation modeling and analysis**. [*S.l.*]: McGraw-Hill, 2014.

LEDOUX, H. *et al.* 3dfier: automatic reconstruction of 3d city models. **Journal of Open Source Software**, v. 6, n. 57, p. 2866, 2021.

LEE, D.-T.; SCHACHTER, B. J. Two algorithms for constructing a delaunay triangulation. **International Journal of Computer & Information Sciences**, Springer v. 9, n. 3, p. 219–242, 1980.

LEE, H.; SONG, S.; JO, S. 3D Reconstruction using a sparse laser scanner and a single camera for outdoor autonomous vehicle. *In*: 2016 IEEE 19TH INTERNATIONAL CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS (ITSC). 2016. **Anais [...]** [*S.l.*: *s.n.*], 2016. p. 629–634.

LERCARI, N. 3d visualization and reflexive archaeology: A virtual reconstruction of çatalhöyük history houses. **Digital Applications in Archaeology and Cultural Heritage**, Elsevier v. 6,, p. 10–17, 2017.

LI, B. *et al.* 3D-ReConstnet: A Single-View 3D-Object Point Cloud Reconstruction Network. **IEEE Access**, v. 8,, p. 83782–83790, 2020.

LI, H.-y. *et al.* A hierarchical contour method for automatic 3D city reconstruction from LiDAR data. *In*: 2012 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM. 2012. **Anais [...]** [*S.l.*: *s.n.*], 2012. p. 463–466.

LI, Q. *et al.* Optimized views photogrammetry: Precision analysis and a large-scale case study in qingdao. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE v. 16,, p. 1144–1159, 2023.

LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. **Nature**, Springer Nature v. 293, n. 5828, p. 133–135, 1981.

LONGUET-HIGGINS, H. C. Computer analysis of visual motion. **Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences**, The Royal Society v. 293, n. 1063, p. 385–397, 1981.

MA, T. *et al.* A Deep Learning-Based Hyperspectral Keypoint Representation Method and Its Application for 3D Reconstruction. **IEEE Access**, v. 10,, p. 85266–85277, 2022.

MARR, D. **Vision: A Computational Investigation into the Human Representation and Processing of Visual Information**. [*S.l.*]: W. H. Freeman and Company, 1982.

MILELLA, A.; REINA, G. 3d reconstruction and classification of natural environments by an autonomous vehicle using multi-baseline stereo. **Intelligent Service Robotics**, Springer v. 7,, p. 79–92, 2014.

MOULON, P. *et al.* OpenMVG: Open multiple view geometry. *In*: SPRINGER. INTERNATIONAL WORKSHOP ON REPRODUCIBLE RESEARCH IN PATTERN RECOGNITION. 2016. **Anais [...]** [*S.l.*], 2016. p. 60–74.

MÜLLER, T. *et al.* Instant neural graphics primitives with a multiresolution hash encoding. **ACM Trans. Graph.**, ACM New York, NY, USA v. 41, n. 4, p. 102:1–102:15, jul. 2022. Disponível em: https://doi.org/10.1145/3528223.3530127.

MUÑOZ-SILVA, E. M. *et al.* A Survey on Point Cloud Generation for 3D Scene Reconstruction. *In*: 2021 INTERNATIONAL CONFERENCE ON MECHATRONICS, ELECTRONICS AND AUTOMOTIVE ENGINEERING (ICMEAE). 2021. **Anais [...]** [*S.l.*: *s.n.*], 2021. p. 82–87.

NAKAO, K. *et al.* 3d shape reconstruction using extended kalman filter with an active camera. *In*: IEEE INTERNATIONAL SYMPOSIUM ON COMMUNICATIONS AND INFORMATION TECHNOLOGY, 2004. ISCIT 2004. 2., 2004. **Anais [...]** [*S.l.*: *s.n.*], 2004. p. 857–860 vol.2.

NATOUR, G. E. *et al.* Radar and vision sensors calibration for outdoor 3d reconstruction. *In*: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). 2015. **Anais [...]** [*S.l.*: *s.n.*], 2015. p. 2084–2089.

NAYAB, A.; RAFIQ, U.; MALIK, H. Rapidex: Ai-based 3d model integration for transportation digital twins. *In*: PROCEEDINGS OF THE 2024 IEEE INTERNATIONAL CONFERENCE ON SMART CITIES. 2024. **Anais [...]** [*S.l.*: *s.n.*], 2024. p. 200–210.

ORSINGHER, M. *et al.* Efficient view clustering and selection for city-scale 3d reconstruction. *In*: SPRINGER. INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING. 2022. **Anais [...]** [*S.l.*], 2022. p. 114–124.

PAGE, M. J. *et al.* Prisma 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. **bmj**, British Medical Journal Publishing Group v. 372,,, 2021.

PATAY-HORVÁTH, A. The virtual 3d reconstruction of the east pediment of the temple of zeus at olympia an old puzzle of classical archaeology in the light of recent technologies. **Digital Applications in Archaeology and Cultural Heritage**, Elsevier v. 1, n. 1, p. 12–22, 2014.

PETERS, J. *et al.* A review of dynamic simulators for autonomous vehicle development. **IEEE Transactions on Intelligent Vehicles**, IEEE v. 3, n. 1, p. 66–77, 2018.

POULLIS, C.; YOU, S. 3d reconstruction of urban areas. *In*: 2011 INTERNATIONAL CONFERENCE ON 3D IMAGING, MODELING, PROCESSING, VISUALIZATION AND TRANSMISSION. 2011. **Anais [...]** [*S.l.*: *s.n.*], 2011. p. 33–40.

PRABHASAVAT, K.; HOMGADE, C. Variation of hepatic artery by 3-d reconstruction mdct scan of liver in siriraj hospital. **Medical journal of the Medical Association of Thailand**, v. 91, n. 11, p. 1748, 2008.

PYLVÄNÄINEN, T. *et al.* 3D City Modeling from Street-Level Data for Augmented Reality Applications. *In*: 2012 SECOND INTERNATIONAL CONFERENCE ON 3D IMAGING, MODELING, PROCESSING, VISUALIZATION & TRANSMISSION. 2012. **Anais [...]** [*S.l.*: *s.n.*], 2012. p. 238–245.

RAJAMANI, R. **Vehicle dynamics and control**. [*S.l.*]: Springer Science & Business Media, 2011.

RIEL, S. V. **Exploring the use of 3D GIS as an analytical tool in archaeological excavation practice**. [*S.l.*]: , 2016. Disponível em: http://rgdoi.net/10.13140/RG.2.1.4738.2643.

ROBERTS, L. G. **Machine perception of three-dimensional solids**. 1963. Tese (Ph.D. thesis) — Massachusetts Institute of Technology (MIT) 1963.

RODRIGUES, I. P. *et al.* Driveability evaluation using a dynamic driving simulator. *In*: SAE TECHNICAL PAPERS. 2021. **Anais [...]** SAE International, 2021. Disponível em: https://www.sae.org/publications/technical-papers/content/2020-36-0199/.

RODRIGUES, I. P. *et al.* Driveability evaluation using a dynamic driving simulator. *In*: SAE TECHNICAL PAPERS. 2021. **Anais [...]** SAE International, 2021. Disponível em: https://doi.org/10.4271/2021-36-0012.

RUPNIK, E.; DAAKIR, M.; DESEILLIGNY, M. P. Micmac–a free, open-source solution for photogrammetry. **Open geospatial data, software and standards**, Springer v. 2,, p. 1–9, 2017.

RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). *In*: IEEE International Conference on Robotics and Automation (ICRA). 2011, Shanghai, China. **Anais [...]** Shanghai, China: [*s.n.*], 2011.

SALEHI, F.; JAFARI, H.; KAMAL, R. Systems engineering for 3d environment construction in autonomous vehicle simulations. *In*: 2025 INTERNATIONAL CONFERENCE ON AUTONOMOUS VEHICLES AND SIMULATIONS. 2025. **Anais [...]** [*S.l.*: *s.n.*], 2025. p. 125–135.

SCHöNBERGER, J. L.; FRAHM, J.-M. Structure-from-motion revisited. *In*: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). 2016. **Anais [...]** [*S.l.*: *s.n.*], 2016.

SCHöNBERGER, J. L. *et al.* Pixelwise view selection for unstructured multi-view stereo. *In*: EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV). 2016. **Anais [...]** [*S.l.*: *s.n.*], 2016.

SEITZ, S. M. *et al.* A comparison and evaluation of multi-view stereo reconstruction algorithms. *In*: PROCEEDINGS OF THE IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). 2006. **Anais [...]** [*S.l.*: *s.n.*], 2006. p. 519–528.

SLAMA, C. W. **Photogrammetry: Basic Principles and General Survey**. [*S.l.*]: American Society of Photogrammetry, 1968.

SLAVCHEVA, M.; BAUST, M.; ILIC, S. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. *In*: 2018 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. 2018. **Anais [...]** [*S.l.*: *s.n.*], 2018. p. 2646–2655.

SUN, Y. *et al.* 3drimr: 3d reconstruction and imaging via mmwave radar based on deep learning. *In*: IEEE INTERNATIONAL PERFORMANCE, COMPUTING, AND COMMUNICATIONS CONFERENCE (IPCCC). 2021. **Anais [...]** [*S.l.*]: IEEE, 2021. p. 20–28.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. [*S.l.*]: Springer, 2010.

TANCIK, M. *et al.* Nerfstudio: A modular framework for neural radiance field development. *In*: ACM SIGGRAPH 2023 CONFERENCE PROCEEDINGS. 2023. **Anais [...]** [*S.l.*: *s.n.*], 2023.

TAUBIN, G. A signal processing approach to fair surface design. *In*: PROCEEDINGS OF THE 22ND ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES. 1995. **Anais [...]** [*S.l.*: *s.n.*], 1995. p. 351–358.

Cgal, Computational Geometry Algorithms Library. [*S.l.*]: , . Http://www.cgal.org.

THRUN, S. *et al.* Stanley: The robot that won the darpa grand challenge. **Journal of Field Robotics**, Wiley Online Library v. 23, n. 9, p. 661–692, 2006.

UNTZELMANN, O. *et al.* A Scalable Collaborative Online System for City Reconstruction. *In*: 2013 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS. 2013. **Anais [...]** [*S.l.*: *s.n.*], 2013. p. 644–651.

VI-GRADE. **DiM150: Dynamic Driving Simulator for Automotive RD**. [*S.l.*]: , 2020. Disponível em: https://www.vi-grade.com.

VOSSELMAN, G.; MAAS, H.-G. **Airborne and terrestrial laser scanning**. [*S.l.*]: CRC Press (Taylor & Francis), 2010.

WU, C. Visualsfm: A visual structure from motion system. **http://www. cs. washington. edu/homes/ccwu/vsfm**, „, 2011.

WU, J.; ZHANG, Z. Deep 3d reconstruction with generative adversarial networks. *In*: NEURAL INFORMATION PROCESSING SYSTEMS (NEURIPS). 2020. **Anais [...]** [*S.l.*: *s.n.*], 2020. p. 7893–7904.

WU, J.; ZHANG, Z. Deep 3d reconstruction with generative adversarial networks. *In*: NEURAL INFORMATION PROCESSING SYSTEMS (NEURIPS). 2020. **Anais [...]** [*S.l.*: *s.n.*], 2020. p. 7893–7904.

XIE, Y.; LI, Y.; QI, Y. A Data-Driven Method for Intrinsic Decomposition of 3D City Reconstruction Scene. *In*: 2019 INTERNATIONAL CONFERENCE ON VIRTUAL REALITY AND VISUALIZATION (ICVRV). 2019. **Anais [...]** [*S.l.*: *s.n.*], 2019. p. 87–92.

XU, Q.; WANG, J.; AN, X. A pipeline for surface reconstruction of 3-dimentional point cloud. *In*: 2014 INTERNATIONAL CONFERENCE ON AUDIO, LANGUAGE AND IMAGE PROCESSING. 2014. **Anais [...]** [*S.l.*: *s.n.*], 2014. p. 822–826.

YOUSSEF, S.; SHEHABY, M. E.; FAYED, S. A smart multi-view panoramic imaging integrating stitching with geometric matrix relations among surveillance cameras (smpi). **Multimedia Tools and Applications**, v. 79,,, 11 2020.

YU, Y. *et al.* Multi-view 2d–3d alignment with hybrid bundle adjustment for visual metrology. **The Visual Computer**, v. 38,,, 04 2022.

ZHANG, X.; JIANG, G.; AI, D. A new method for 3D city map generation. *In*: 2015 IEEE WORKSHOP ON SIGNAL PROCESSING SYSTEMS (SIPS). 2015. **Anais [...]** [*S.l.*: *s.n.*], 2015. p. 1–5.

ZOLANVARI, S. *et al.* Dublincity: Annotated lidar point cloud and its applications. **arXiv preprint arXiv:1909.03613**, ,,, 2019.