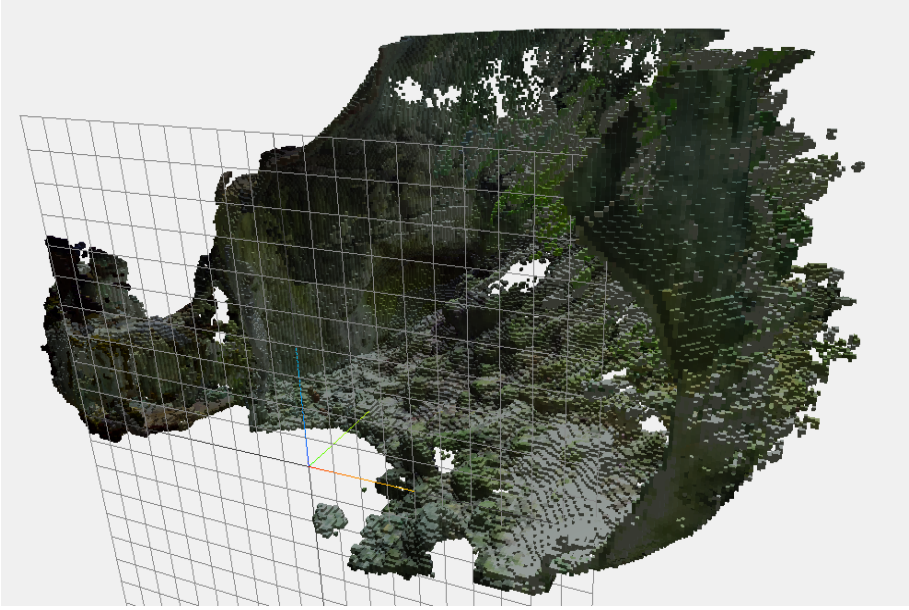
ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA



Application of Stereoscopy in Speleological Surveying

MARTIM AUGUSTO TEIXEIRA GONÇALVES

(Licenciado em Engenharia Informática e Multimédia)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e Multimédia

Orientadores: Doutor Pedro Fazenda

Doutor Pedro Mendes Jorge

Júri:

Presidente:	Doutor Diogo Nuno Crespo Ribeiro Cabral
Vogais:	Doutor Rui Pedro Da Silva Nóbrega
	Doutor Pedro Viçoso Fazenda

Novembro 2025

Application of Stereoscopy in Speleological Surveying

MARTIM AUGUSTO TEIXEIRA GONÇALVES

(Licenciado em Engenharia Informática e Multimédia)

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e Multimédia

Orientadores: Doutor Pedro Fazenda, DEI/ISEL
Doutor Pedro Mendes Jorge, DEI/ISEL

Júri:

Presidente:	Doutor Diogo Nuno Crespo Ribeiro Cabral, DEI/ISEL
Vogais:	Doutor Rui Pedro Da Silva Nóbrega, DI/FCT/UNL
	Doutor Pedro Viçoso Fazenda, DEI/ISEL

Novembro 2025

To all those who were there to listen, even when it did not make sense.

Acknowledgements

I would like to express my deepest gratitude to my advisers for their guidance, encouragement, and invaluable feedback throughout the development of this work.

I am also grateful to ISEL (Instituto Superior de Engenharia de Lisboa) and NOVA LINCS for their partial financial support, which made this research possible. I would like to extend special thanks to José Ribeiro from Centro de Estudos e Actividades Especiais da Liga para a Protecção da Natureza (CEAE-LPN) and his son, Miguel Ribeiro, for allowing me to assist them during the traditional survey of the cave.

Finally, I wish to thank my colleagues, friends, and family for their patience, motivation, and support during this journey.

Statement of integrity

I declare that this dissertation is the result of my personal and independent research. Its content is original, and all sources listed in the bibliographic references were consulted and are duly mentioned in the text. I further declare that all scientific and technical references relevant to the development of the work are duly cited and included in the bibliographic references.

The author

Lisbon, 4, December

Abstract

Cave topography plays a fundamental role in supporting various fields that require the exploration of underground environments by specialists. However, speleologists rely on traditional techniques which remain labor-intensive and prone to errors. While LiDAR and photogrammetry boast advanced mapping accuracy, high costs, bulk, laborous preparation and operational complexity limit widespread adoption. This study aims to investigate the use of a portable, cost-effective alternative surveying method by leveraging stereoscopy and established tools from the robotics ecosystem consisting on the ZED 2 3D camera and OctoMap framework. Field experiments compared stereoscopic mapping against a traditional compass-and-laser workflow, evaluating accuracy, efficiency and usability. Results demonstrate that stereo-SLAM pipelines produce metrically accurate 3D models in real time, offering an interesting development path to bridge the gap between manual surveys and high-end LiDAR scans. Limitations in portability, environmental conditions and robustness were identified along with future directions to address them. Despite these limitations in the method, it shows promising results by reducing cost and effort in obtaining structured and machine-readable cave representations with applications beyond speleology. These findings may support the development of tools to assist adjacent fields such as archaeology, geology, biology and environmental monitoring. The work establishes a foundation field-ready stereoscopic systems supporting semantic mapping, advanced spatial analysis, and integration with robotic exploration.

Keywords: speleology, cave mapping, stereoscopy, stereo-SLAM, topographic survey

Resumo

A topografia de cavernas desempenha um papel fundamental no apoio a diversas áreas que exigem a exploração de ambientes subterrâneos por especialistas. Em espeleologia, continua a recorrer-se a técnicas tradicionais de levantamento, caracterizadas por serem morosas e suscetíveis a erro humano e imprecisões. Apesar de tecnologias como o LiDAR e a fotogrametria oferecerem elevada precisão de mapeamento, o seu custo elevado, a dimensão e peso do equipamento, a preparação exigente e a complexidade operacional revelam-se um entrave à sua adoção generalizada. O presente estudo tem como objetivo investigar a utilização de um método alternativo para o levantamento da topografia, portátil e economicamente viável, através da aplicação de estereoscopia e ferramentas estabelecidas no ecossistema da robótica, nomeadamente a câmara ZED 2 e a *framework* OctoMap. Para tal efeito, foram realizados ensaios de campo que compararam o mapeamento estereoscópico com o fluxo de trabalho tradicional baseado em bússola e laser, avaliando-se a precisão, a eficiência e a usabilidade. Os resultados obtidos demonstram que *pipelines* de stereo-SLAM permitem a geração de modelos tridimensionais metricamente precisos em tempo real, constituindo uma via promissora para preencher a lacuna existente entre levantamentos manuais e varrimentos LiDAR de elevada resolução. Identificaram-se, contudo, limitações relacionadas com a portabilidade, as condições ambientais e a robustez do sistema, apontando-se igualmente direções futuras para melhorar o seu desempenho e os seus resultados. Apesar destas limitações, a abordagem proposta evidencia resultados encorajadores ao reduzir custos e esforços na obtenção de representações estruturadas e legíveis de ambientes subterrâneos, com potencial de aplicação para além da espeleologia. As conclusões alcançadas poderão contribuir para o desenvolvimento de ferramentas de apoio a áreas adjacentes, como a arqueologia, a geologia, a biologia e a monitorização ambiental. Este trabalho estabelece ainda as bases para a evolução de sistemas estereoscópicos operacionais em campo, capazes de suportar mapeamento semântico, análises espaciais avançadas e integração com plataformas de exploração robótica.

Palavras-chave: espeleologia, mapeamento de grutas, estereoscopia, stereo-SLAM, levantamento topográfico

Contents

List of Figures

List of Tables

Glossary

alcove	A small chamber or indentation in a cave wall or ceiling, sometimes forming blind pockets. 31
blind tunnel	A cave passage that terminates abruptly without connecting to another passage. A dead end tunnel. 28, 31, 33
cave surveying	The process of measuring the distances, bearings, and inclinations of cave structures, passages, and chambers to produce a technical map of the cave. 1
cliff-side	A steep rock face, often forming part of cave entrances or collapses. 33
clinometer	An instrument used for measuring angles of slope, elevation, or depression of an object with respect to gravity. 1, 5
collapse feature	A chamber or opening in a cave formed when the ceiling or roof of a conduit falls in. 27
conduit	A natural tunnel or channel within rock through which water once flowed or still flows. 27
CUDA	Compute Unified Device Architecture. A parallel computing platform developed by NVIDIA that allows GPUs to accelerate computational tasks like depth estimation 16, 17, 44
doline	A natural depression or sinkhole formed by the collapse or dissolution of limestone in karst areas. 27
flowstone	A sheet-like mineral deposit formed by water flowing over cave walls or floors, creating smooth surfaces. 2
ghosting	Erroneous or duplicate features in a map caused by poor depth estimation, often creating visual artifacts. 21
karst	A landscape formed by the dissolution of soluble rocks (commonly limestone, dolomite, or gypsum), characterized by caves, sinkholes, and underground drainage systems. 27

limestone pavement — A flat, exposed surface of limestone that has been weathered into blocks separated by deep fissures. 27

loop closure — A SLAM technique where the system recognizes a previously visited location, correcting accumulated mapping errors. 13

Maciço Calcário Estremenho — A major limestone massif in Portugal, known for its extensive karst systems, including the cave tested in this study. 27

Monocular SLAM — A form of Simultaneous Localization and Mapping (SLAM) that uses a single camera to estimate motion and build a map of the environment. 7

OctoMap — A software framework that stores 3D environments in an octree data structure, representing occupied, free, and unknown spaces efficiently. 2, 8, 12, 13, 17, 21, 33, 40, 42, 43, 45

Octree — A hierarchical 3D grid data structure that recursively divides space into eight smaller cubes, used for efficient mapping and storage. 1, 2, 13

overhang — A rock formation where part of a wall or ceiling extends outward, creating a ledge or shelter. 33

photogrammetry — A technique that reconstructs 3D structures from overlapping 2D photographs, often using Structure-from-Motion algorithms. 1, 43

point cloud — A collection of data points in 3D space representing the external surfaces of objects or environments. 2

rockfall — Blocks of rock that have broken off from cave walls or ceilings, often creating obstacles inside caves. 28

semantic mapping — A mapping approach that augments geometric or topographic data with contextual information (e.g., object categories, materials, or functions), enabling higher-level reasoning and interpretation of environments. 2

speleology — The scientific study of caves, their formation, geology, ecology, and exploration. 1

stalactite — A mineral deposit hanging from the ceiling of a cave, formed by dripping water rich in dissolved minerals. 2

Stereolabs — Company which produces the ZED cameras. 10

stereoscopic — Adjective for a system which uses two cameras (like human eyes) to capture slightly different viewpoints and calculate depth through triangulation. 1

traverse surveying — A traditional cave mapping method using compasses, clinometers, and tape measures to record angles and distances between stations. 5

vertical shaft (pit) A steep, often cylindrical vertical cave passage that can connect surface and underground systems. 27

ZED 2 A commercially available stereoscopic camera used for depth perception and SLAM in robotics and mapping. 9, 10, 12, 13, 15, 16, 17, 18, 24, 43, 44

Acronyms

AR	Augmented Reality 31
BT	Binary OcTree 12
CAD	Computer-Aided Design 45
CEAE-LPN	Centro de Estudos e Actividades Especiais da Liga para a Protecção da Natureza vii
DOE	Diffractive Optical Element 10
DOEs	Diffractive Optical Elements 10
FOV	Field of View 24
GIS	Geographic Information System 11, 13
GNSS	Global Navigation Satellite System 14, 16
GPS	Global Positioning Systems 11, 16
IMU	Inertial Measurement Unit 7, 13
LAS	LASer file format 11, 12
LAZ	Compressed LASer file format 11, 12
LiDAR	Light Detection and Ranging 1, 2, 6, 7, 9, 10, 11, 13, 14, 28, 43, 45
ML	Machine Learning 2, 14, 50
OT	OcTree 12, 23, 33, 40, 61
PCD	Point Cloud Data 11, 12, 33, 40
PCL	Point Cloud Library 11, 19, 40, 57
PLY	Polygon File Format 11, 12, 61
PNSAC	Parque Natural das Serras de Aire e Candeeiros 27
ROI	Region of Interest 24

ROS	Robot Operating System 11, 17, 18, 20
RTAB-Map	Real-Time Appearance-Based Mapping 7, 8, 14, 22, 43
SDK	Software Development Kit 13, 16, 17
SfM	Structure from Motion 6, 9, 12, 45
SLAM	Simultaneous Localization and Mapping 6, 10, 12, 13, 14, 43, 45
SPA	Single Page Application 61
TLS	Terrestrial Laser Scanner 2, 6, 9, 13, 45
TSDF	Truncated Signed Distance Field 14
TSDFs	Truncated Signed Distance Fields 8
TSV	Tab-Separated Values 20, 24, 41, 57
VIO	Visual-Inertial Odometry 7, 13, 14, 17, 43
VO	Visual Odometry 7
VR	Virtual Reality 31
vSLAM	Visual Simultaneous Localization and Mapping 7, 14, 43, 45

1 Introduction

1.1 The Role of Cave Topography

Cave mapping has long been a crucial aspect of speleology, providing valuable insights into underground landscapes, geological formations, and environmental conditions. Traditionally, cave surveying has relied on manual methods such as compass and clinometer measurements, tape surveys, and hand-drawn sketches. Although these techniques have enabled the documentation of countless cave systems, they are time consuming, physically demanding, and prone to human error, particularly in complex or hazardous environments. The introduction of digital mapping tools, such as laser scanners and photogrammetry, has significantly improved the accuracy and efficiency of cave surveying. However, many of these advanced techniques have limitations in cost, portability, and adaptability to confined or obstructed spaces.

1.2 Bridging the Technological Gap

In recent decades, the continuous development and integration of stereoscopic vision systems in robotics has emerged as a promising solution for mapping in many environments. Stereoscopic cameras, which mimic human binocular vision to reconstruct 3D depth information, offer a lightweight, low budget (cost-effective) alternative to solutions based on Light Detection and Ranging (LiDAR) (Fig. 1.2). Digital 3D maps may contain massive structural and color information derived from thousands to several millions of points. While they store a data rich representation of an environment, it is important to consider their purpose. Interpretation of convoluted 3D maps or still pictures of them can be hard for a human, losing depth perception and getting lost among trivial details. To be legible by a human, these maps require interactive tools with the capability of selecting regions, filtering irrelevant details and projecting views from different perspectives. Stereoscopic cameras can be combined with data structures such as Octrees to create detailed and structured cave representations while optimizing storage and computational requirements [1]. The same way infrared night vision goggles allow humans to enhance their vision in low-light environments by converting the light from the infrared to the visible spectrum, subsets of

information contained in map data can be highlighted to produce views focusing on what matters. Unlike traditional surveying tools, these systems can facilitate on-site or even remote operation (e.g., using drones [2]), reducing the reliance on manual measurement and improving safety by allowing remote data collection in dangerous or inaccessible cave sections.

1.3 Proposed Solution and Contribution

This study explores the application of the ZED 2 stereoscopic camera (Fig. 1.1) together with the OctoMap library for cave mapping, assessing its effectiveness in comparison to both traditional and modern surveying methods. By leveraging stereoscopic vision and structured data storage, this method aims to bridge the gap between manual cave surveying and high-end LiDAR solutions, offering a balance between cost, portability, and accuracy. The findings of this research have implications not only for speleology but also for related fields such as archaeology, geology, biology, and environmental science, where detailed underground mapping is essential.

To achieve this goal, it is crucial to monitor the quality and cost-effectiveness of stereoscopic mapping relative to both traditional and LiDAR-based methods. Quality is defined in terms of spatial accuracy, resolution of morphological features, and completeness of coverage, while cost-effectiveness encompasses hardware expenses, field time, and post-processing requirements. For instance, a stereoscopic survey of a 100 m passage may achieve ± 5 cm accuracy at a fraction of the cost of a high-end Terrestrial Laser Scanner (TLS) system (e.g., Leica BLK360, Ouster OS0), while capturing sufficient morphological detail for speleological analysis.

This work lays the foundations for a centralized repository of stereoscopic cave maps and associated metadata. By enforcing maps to be represented in an extensible data structure, such a repository facilitates advanced applications, including automatic segmentation of cave features, detection of morphological changes over time, and comparative studies across karst systems – all crucial to develop semantic mapping capabilites. Structured data in formats such as Octrees not only preserve metric accuracy but also enable the direct application of algorithms established for common data structures, such as trees, for further processing, including semantic mapping and Machine Learning (ML) [3–5]. For example, annotated datasets could be used for training models to identify stalactites, flowstone, or collapse features directly from 3D point clouds, or to extract 2D topographic plans programmatically from volumetric data.

The main contributions of this work are:

- Promoting a foundation for the introduction of a scientific, data-driven methodology for speleology;
- Combining cost-efficient mapping with reproducibility;
- Standardized data structures and long-term reusability of results.

This document is organized in five chapters:

1. **Introduction:** Presents the motivation and main contributions of this study.
2. **State of the Art:** Describes the traditional surveying method, sensors employed in modern 3D mapping, common data structures, compares them and highlights how they may be integrated into a novel approach.
3. **Proposed Method:** Defines a hardware configuration and the software architecture required to bring the mapping system to fruition, as well as the workflow employed for topographic acquisition.
4. **Experimental Results:** Introduces the results of mapping experiments, providing a qualitative and a quantitative analysis concerning map quality and system performance.
5. **Conclusion:** Finishing thoughts on design choices, workflow and results, identifying the key limitations of the proposed method and suggesting future work to address them.

Figure 1.1: *Stereoscopic ZED series cameras and ZED Box with a NVIDIA Jetson board.*

Figure 1.2: *Leica's BLK2GO (left, usable on the move) and BLK360 (right) LiDAR devices.*

2 State of the Art

The mapping of subterranean environments is a field that, despite its long history, continues to evolve with the advent of new technologies. The methods employed range from centuries-old manual techniques to modern digital approaches, each with its own set of advantages and limitations. A proper understanding of this spectrum is essential to contextualize the contribution of new methodologies, such as the one proposed in this dissertation.

2.1 Traditional Surveying

The foundational methods of cave surveying have, for centuries, relied on a set of simple, robust tools and the meticulous work of speleologists. The basic equipment consists of a compass for measuring bearings, a clinometer for vertical angles, and a measuring tape for distances [6]. This process, often called traverse surveying, involves establishing a series of stations throughout the cave. At each station, measurements are taken to subsequent stations and to the surrounding cave walls, ceiling, and floor. These measurements are recorded by hand in a logbook, often accompanied by hand-drawn sketches that capture the morphological details of the passages [7].

This traditional approach, though labor-intensive and susceptible to cumulative human error, has been the standard for generations and is responsible for the vast majority of existing cave maps. In recent years, this process has been augmented by digital tools that increase efficiency and reduce transcription errors. Devices like the DistoX [7–9] (Fig. 2.1), a handheld instrument that combines a laser distance meter, a compass, and a clinometer, can send measurements via Bluetooth to software like TopoDroid running on a mobile device [10]. This "paperless" workflow digitizes data at the point of capture, but it is important to note that it assists rather than replaces the traditional methodology. The surveyor's interpretation and the fundamental principles of station-based measurement remain central to the process [7].

Figure 2.1: *A series of Leica Disto laser distance meters.*

2.2 Digital 3D Data Acquisition Methods

The complete digitization of cave environments has been made possible by a range of advanced technologies that capture the three-dimensional structure of the space directly.

LiDAR is often considered the gold standard for high-accuracy 3D mapping. TLS emit laser pulses and measure the time-of-flight to determine the precise distance to surrounding surfaces, generating a dense and accurate point cloud of the environment. In speleology, LiDAR provides unparalleled detail for geomorphological and structural analysis [11] – the best to obtain rigorous geometry for topographical features. However, its application is limited by significant practical constraints: the equipment is typically expensive, heavy, and sensitive, making it difficult to transport and operate in the challenging conditions of a cave, which often involve tight passages, mud, and water [2]. Most importantly, the exclusive employment of laser sensors discards any additional information, such as color data. Whichever the format (e.g., RGB, HSV, etc.), preserving the color of the cave's features aids in improving the visuals of 3D topography, as well as its reconstruction and virtual representation of the caves. Different types of sensors may be fused, having their data synchronized to register distance, color, humidity, temperature and magnetic fields, among others. The richer the information associated with the topography, the wider the applications where it can be of use.

As a more accessible alternative, photogrammetry, particularly using Structure from Motion (SfM) algorithms, has gained popularity. This technique uses a series of overlapping photographs taken from different viewpoints to reconstruct a 3D model. SfM is highly cost-effective, as it only requires a standard digital camera. It has been successfully used for the virtualization of underground heritage sites, including semi-submerged caves [12]. The main drawbacks are its sensitivity to lighting conditions and the need for significant texture on surfaces for feature matching, which can be a problem in homogeneous or poorly lit cave sections [13].

Simultaneous Localization and Mapping (SLAM) represents a paradigm shift from static data acquisition to dynamic mapping. A sensor-equipped device, often a camera, is moved through the environment, and SLAM algorithms simultaneously build a map of the space while tracking the device's location within it. When the primary sensor is a camera, it is

known as Visual Simultaneous Localization and Mapping (vSLAM). The core of vSLAM is Visual Odometry (VO), which estimates the camera's motion between consecutive frames. This can be enhanced with data from an Inertial Measurement Unit (IMU) in what is called Visual-Inertial Odometry (VIO), which helps to mitigate issues like motion blur or tracking failure in textureless areas. A critical distinction in vSLAM is between monocular (single camera) and stereoscopic (dual camera) systems. Monocular SLAM systems can estimate 3D structure but suffer from scale ambiguity – they cannot determine the absolute size of the environment without external information. Stereoscopic systems, like the one used in this work, resolve this ambiguity by using the fixed baseline between the two cameras to calculate depth directly through triangulation, providing a true-to-scale map of the environment [14].

2.3 Core Technologies and Trade-offs

The choice of a mapping system for cave environments involves a series of trade-offs between different technologies, from the sensors that perceive the world to the data structures that represent it.

The primary trade-off in cave mapping sensors is between LiDAR and stereo cameras. LiDAR is an active sensor, meaning it provides its own illumination and can operate in complete darkness, delivering high-density, high-accuracy data over long ranges. Its main disadvantages are cost, portability and no color data. Passive stereo cameras, on the other hand, are lightweight, affordable, and computationally less demanding. However, their performance is critically dependent on ambient lighting and the presence of visual features on surfaces for stereo matching. In dark or morphologically uniform sections of caves, this can be a significant challenge, requiring artificial illumination [15]. Active stereo cameras integrate laser (like LiDAR) or structured light patterns to overcome some of these limitations. The trade-offs between passive and active stereo will be addressed later.

Modern SLAM systems are predominantly built on graph-based optimization (Fig. 2.2 shows a general overview of the data flow for such systems). Each node in the graph corresponds to a camera pose (i.e., position and orientation of a sensor at a given viewpoint) and the edges encode spatial constraints between poses derived from odometry. Over time, small odometry errors accumulate, an effect known as drift. To correct this, the system applies loop closure: when a previously visited location is recognized, a new constraint is added to the graph [16, 17]. An optimization algorithm, such as g2o, then adjusts all the poses in the graph to minimize the overall error, resulting in a globally consistent map [18]. Frameworks like Real-Time Appearance-Based Mapping (RTAB-Map) [19] are specifically designed with robust, appearance-based loop closure detection. RTAB-Map typically operates by first constructing and optimizing the pose graph online, while the generation of a dense 3D map from this graph is often performed as a subsequent, separate step. This makes it highly suitable for large-scale and long-term operation, but can be less interactive for on-the-fly reconstruction. This workflow can also lead to operational "hiccups", where the process halts if a new keyframe cannot be confidently matched, pausing

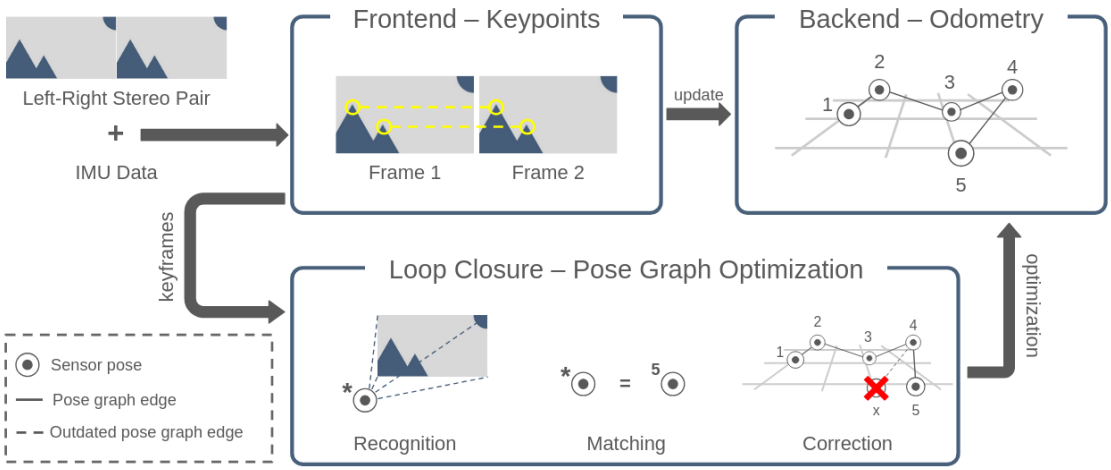


Figure 2.2: *Stereo-SLAM with VIO overview (general data flow).*

all mapping until the system can re-localize. The halting behavior in RTAB-Map stems from its reliance on a graph-based SLAM approach. When the system processes a new frame from the sensor, it must first localize itself within the existing graph. It does this by trying to match visual features in the current view to features stored in previous keyframes. If a confident match is found, the system knows its position and can add a new keyframe and associated edges to the graph, extending the map. However, if the system enters a visually ambiguous area (like a uniform hallway) or moves too erratically, it may fail to find a reliable match. In this state, the system is considered "lost" as it cannot determine its pose relative to the map. To prevent adding erroneous data and corrupting the graph with an incorrect position, RTAB-Map's default strategy is to pause mapping operations. It stops integrating new information and waits until it can successfully re-localize by recognizing a known keyframe. This safety mechanism is what causes the operational halt. This topic will be addressed later, especially concerning its implications when traversing a cave.

The raw output of most 3D sensors is a point cloud, a simple but unstructured representation. For applications requiring real-time updates, analysis, or navigation, structured formats offer advantages in update and manipulation operations. Volumetric representations, which break space down into a grid of 3D cells (voxels), are common. A key example is the OctoMap, which uses an octree data structure [1]. Octrees recursively subdivide space into eight cubic child nodes, allowing for a multi-resolution representation that is memory-efficient, as it does not need to store empty space. OctoMap uses a probabilistic model to update the occupancy of each voxel, making it robust to sensor noise and capable of distinguishing between free, occupied, and unknown space. This is a significant advantage over simpler point clouds or surface-based models like surfels [20] or Truncated Signed Distance Fields (TSDFs) [21], which are excellent for surface reconstruction but do not explicitly model free or unknown space, a critical requirement for navigation and planning. The ability of OctoMap to efficiently handle real-time updates and remove "ghost" artifacts from moving objects or mapping errors makes it a powerful choice for dynamic mapping scenarios. Further research has built upon this to improve real-time performance, with frameworks like MAP-ADAPT proposing quality-adaptive techniques

Table 2.1: 3D Mapping Sensors: Capabilities

Sensor	**Technology**	**Range (Depth)**	**Outdoor Use**
Monocular Cam.	Vision (SfM)	Ambiguous Scale	Excellent
ZED 2	Passive Stereo	0.2 m–20 m	Excellent
RealSense D435i	Active Stereo (IR)	0.2 m–10 m	Poor
Azure Kinect DK	Time-of-Flight	0.5 m–4 m	Poor
Ouster OS0-128[1]	LiDAR	Up to 50 m	Excellent
Leica BLK360 G1[1]	LiDAR (TLS)	0.6 m–60 m	Excellent
Leica BLK360 G2[1]	LiDAR (TLS)	0.5 m–45 m	Excellent

Table 2.2: 3D Mapping Sensors: Costs and Key Trade-offs

Sensor	**Approx. Cost (USD)**	**Key Trade-off**
Monocular Cam.	100	Lacks metric accuracy.
ZED 2	500	Balanced cost-effectiveness; robust.
RealSense D435i	300	Fails in sunlight; limited range.
Azure Kinect DK	400	High power use; very limited range.
Ouster OS0-128[1]	15,000	High cost and power; professional grade.
Leica BLK360 G1[1]	10,000	Portable for field use (1 kg); slower scan speed.
Leica BLK360 G2[1]	25,000	Lighter, faster, VIS; somewhat reduced max range.

to manage computational load in semantic 3D mapping [22].

2.3.1 Sensor Comparison

The selection of a primary 3D sensor for a portable mapping system, particularly in environments such as caves, needs a careful balancing of trade-offs between performance, cost, and environmental resilience. The comparative data presented in Tables 2.1 and 2.2 motivated the choice of the ZED 2 passive stereoscopic camera for this work.

At the highest end of the spectrum of best accurate sensors, LiDAR systems such as the Ouster OS0 [23] and the Leica BLK360 [24] series offer unparalleled accuracy, range, and environmental robustness. They are the standard for professional surveying, capable of generating dense, metrically precise point clouds in any lighting condition. However, their significant drawbacks are prohibitive for this research context: high cost (ranging from thousands to tens of thousands of dollars), substantial power consumption, and greater weight and bulk. The static, stop-and-scan nature of a TLS like the BLK360 is also ill-suited for the dynamic, on-the-move mapping paradigm central to this project.

At the other extreme, a simple monocular camera represents the most affordable and lightweight option. However, its fundamental limitation is the inability to perceive absolute scale from a single viewpoint. While 3D reconstruction is possible through SfM or monocular SLAM, the resulting map is dimensionless until scaled by other means, rendering it unsuitable for direct topographic measurement.

This leaves a middle ground occupied by consumer-grade depth cameras. Active-light

[1]Retail prices for certain sensors are hard to find as they are determined on a client-by-client basis upon contacting sales teams. Prices listed for these items are collected from second hand sellers and may not reflect their true price.

sensors, such as the Intel RealSense D435i (active stereo) and the Azure Kinect DK (Time-of-Flight), are popular due to their low cost and ability to function in complete darkness by projecting their own infrared light patterns. Classified as structured light sensors, they work by using Diffractive Optical Elements (DOEs), projecting a known shape or pattern onto a surface and measure its deformation to determine depth values. Diffractive Optical Element (DOE)s deflect infrared light into multiple orders at precise angles. Periodicity and their spatial frequencies, rather than the surface topography profile, determine the quality of the measurements. This very feature, however, becomes a critical vulnerability in outdoor or mixed-lighting conditions. The infrared radiation from sunlight overwhelms the sensor, severely degrading or completely disabling its depth perception capabilities. Most importantly, their short range is highly restrictive (1-4m), making them not suitable for cave mapping. These limitations make them unreliable and not suitable for cave mapping.

In this context, the ZED 2 passive stereo camera emerges as the most pragmatic and well-rounded solution. As a passive sensor, it functions like a pair of human eyes, relying on ambient light to perceive depth through triangulation. This makes it immune to solar interference, ensuring robust performance when transitioning from outdoors to indoors. While it requires sufficient ambient light and surface texture to perform stereo matching – a potential challenge inside a dark cave – this is readily mitigated with portable artificial lighting, a standard component of any speleological expedition. Stereolabs states that ZED cameras, at 1 meter, "can reach a precision of few centimeters (<5%)", while "at 20 meters the distance precision is 10% therefore around 2 meters (19m - 21m)" [25]. It is expected for depth-sensing cameras and LiDAR systems to exhibit a progressive loss of precision as the measurement range increases, resulting in higher uncertainty and reduced reliability of spatial data at greater distances. For cave mapping applications where accurate reconstruction of irregular geometries is critical, it is therefore necessary to find a compromise between range and precision. This can be achieved via programmatic constraints within the system configuration to limit data acquisition to ranges to attain the desired precision. By restricting measurements to validated distance thresholds, the mapping process can maintain consistent accuracy while avoiding the propagation of range-induced errors into the final 3D model. Conversely, limited ranges may not be practical when attempting to map caves with irregular terrain and cavernous, expansive chambers dozens of meters tall and wide. In such cases, sacrificing accuracy and precision for reach may produce adequate results.

Furthermore, the ZED 2 provides a true-to-scale metric map, resolving the ambiguity of monocular systems. Its rolling shutter, however, is a disadvantage. In comparison to cameras with a global shutter, the ZED 2 is more prone to distortion in movement. Some SLAM algorithms, such as ORB-SLAM3 [26], are designed to primarily support global shutter cameras and may not work properly with rolling shutters. Nevertheless, it achieves balanced functionality at a fraction of the cost of LiDAR, while being lightweight and power-efficient enough for a portable, human-operated system. The ZED 2, therefore, occupies a critical niche: it makes metric 3D mapping accessible by providing a cost-effective, versatile, and environmentally resilient sensor that is "good enough" for a wide

Table 2.3: Common 3D Mapping File Formats: Basic Properties

Format	Type	Encoding
PCD	Point cloud	ASCII/Binary
PLY	Point cloud/mesh	ASCII/Binary
LAS / LAZ	Point cloud	Binary/Compressed
Graph	Pose graph	ASCII
OT	Octree (full)	Binary
BT	Octree (leaves)	Binary

Table 2.4: Common 3D Mapping File Formats: Strengths and Limitations

Format	Strengths	Limitations
PCD	Native to PCL; flexible fields	Poor scaling; large files
PLY	Widely supported; simple	Limited compression
LAS / LAZ	Rich geospatial metadata	Heavy GIS orientation
Graph	Lightweight; reproducible	No spatial geometry
OT	Occupied/free/unknown; reproducible	Larger overhead in clutter
BT	Compact transmission	No intermediate nodes

range of field applications, including the topographic survey methodology proposed in this dissertation.

2.3.2 Data Structures and Their Properties

Beyond sensor and algorithmic considerations, the choice of file format has a direct impact on storage efficiency, data accessibility, and usability. Several formats are commonly employed in 3D mapping, each with specific properties.

The Point Cloud Data (PCD) (`.pcd`) format, native to the Point Cloud Library (PCL), stores unstructured 3D point data together with optional attributes (e.g., point's color, intensity, normals). It supports both ASCII and binary encodings. ASCII files are human-readable but incur a high storage overhead, while binary files are compact and faster to parse. The "binary compressed" data format for PCD files applies further compression upon its already compact by nature binary representation. The Polygon File Format (PLY) (`.ply`) is widely used for geometry exchange and visualization. Like PCD, it supports ASCII and binary representations. It excels in interoperability with modeling tools, but lacks efficient compression mechanisms and is less suitable for large-scale, real-time datasets. The trade-offs to consider here are that PCD lends itself best as an extensible format for the point clouds themselves. At the same time, PLY excels at containing meshes, which is particularly useful when exporting them to more common 3D editing software, such as Blender.

The LASer file format (LAS) (`.las`) is a standard in geospatial applications, storing LiDAR point clouds with rich metadata (classification, Global Positioning Systems (GPS) time, scan angle). Its compressed counterpart, Compressed LASer file format (LAZ) (`.laz`), achieves substantial reductions in storage footprint without loss of information. These file formats are widely supported in Geographic Information System (GIS) software but are less integrated into robotics frameworks such as Robot Operating System (ROS).

Graph-based SLAM systems often export results as `.graph` files, which encode the pose graph (sensor poses and spatial constraints). These files are lightweight and essential for loop-closure optimization and reproducibility, but they do not directly store 3D spatial data.

The OctoMap framework employs octree-based representations. OcTree (OT) (`.ot`) files store a full probabilistic occupancy tree, including free, occupied, and unknown voxels, in a binary serialization. Binary OcTree (BT) (`.bt`) files provide a more compact, binary-only format storing only leaf nodes, suitable for memory-efficient transmission. The octree's hierarchical structure enables compression by pruning large homogeneous regions into parent nodes, leading to superior scaling in spacious environments. However, in cluttered or noisy environments with irregular surfaces, the subdivision into many small voxels diminishes compression efficiency.

Formats supporting binary encoding (i.e., PCD, PLY, OT, BT and LAS/LAZ) are substantially more compact than their ASCII equivalents.

In this study, PCD and OT were selected as the primary storage formats. There are two reasons for this: (i) PCD preserves sensor measurements in a straightforward, per-point structure, ensuring transparency and compatibility with common processing pipelines – a simple format for visualization; and (ii) OT provides a structured, probabilistic volumetric representation suitable for advanced analysis. While OT files exhibit greater complexity, they retain critical information on free and unknown space, which is valuable for reproducibility, later reprocessing, and potential applications in navigation or semantic mapping.

2.4 Theoretical Introduction

Mapping subterranean environments with low-cost, portable sensors (such as stereo cameras) combines three main theoretical strands: (i) geometric sensing and metric reconstruction; (ii) probabilistic volumetric representation and occupancy estimation; and (iii) graph-based SLAM (estimation / optimization) with loop closure and robustness considerations. This section provides a theoretical background to contextualise, in light of the previous points, the choice made in this work: a pipeline based on the ZED 2 camera and OctoMap.

Passive stereoscopy computes depth by triangulating corresponding image features across a fixed baseline. In contrast to monocular systems, a calibrated stereo rig resolves scale from the first frame, producing metric depth estimates (subject to measurement noise and rolling shutter effects). This metric observability implies that the reconstruction error has two major sources: stereo correspondence error (depends on the texture, illumination and baseline) and geometric calibration / timing error (rolling shutter, IMU time sync). Photogrammetric SfM remains widely used for high-resolution heritage and cave virtualization. Nocerino *et al.* (2019) demonstrate SfM approaches on a semi-submerged cave, highlighting excellent geometric detail but also the method's sensitivity to lighting and the need for carefully planned capture and heavy offline processing [12]. SfM remains preferable for preservation tasks requiring dense textured meshes; but its offline nature

contrasts with the real-time goals of this stereoscopic SLAM approach. High-end TLS and SLAM-based mobile LiDAR provide superior geometric fidelity and long range, and recent works show practical workflows for rapid cave scanning and GIS-compatible basemaps. These approaches are resource-intensive (cost, weight, power) and often need static setups, which justifies the investigation of lower-cost real-time systems where the application favors mobility and immediate feedback. Stereolabs' ZED 2 (and ZED 2i) are commonly used in robotic and mapping projects because they provide a stereo baseline, on-board IMU, and Software Development Kit (SDK) support for real-time depth/VIO. The manufacturer data and third-party datasheets quantify ranges and precision (noting degradation with range), and many field studies corroborate the ZED 2's practical "good-enough" trade-offs for medium-range mapping.

The built-in fusion of stereo vision and IMU data significantly improves the robustness of SLAM, particularly in VIO. The IMU provides high-frequency accelerometer and gyroscope data that helps estimate motion during fast movements or when visual features are scarce, such as the aforementioned low light conditions, motion blur, or textureless environments. This tight coupling between visual and inertial measurements reduces drift and improves short-term pose estimation accuracy. It does not, however, make the system impervious to it. The same IMU data can also introduce bias and noise that accumulate over time if not properly calibrated or fused with anomalous visual data.

Volumetric occupancy maps model space as a partitioned grid in which each cell (voxel) stores an occupancy probability. Octrees provide a memory-efficient multiresolution representation (refining only occupied/interesting regions) and explicit modelling of free/unknown space — properties that are crucial for navigation, planning and later semantic processing. OctoMap explicitly propagates sensor uncertainty into occupancy updates and supports raycasting for carving free space, which helps to reduce reconstruction noise, uncertainty and other phenomena. These formal properties justify the choice of OctoMap for use as opposed to raw point clouds alone.

Modern SLAM is commonly cast as a factor graph: nodes represent sensor poses and possibly landmarks; factors encode odometry, loop closure, and other constraints. Optimization minimizes a global cost and distributes correction across the graph. The theoretical guarantees (local minima, convergence) hinge on the quality of constraints and the correctness of loop-closure hypotheses; in texture-poor cave areas, incorrect or missing correspondences lead to large residuals or catastrophic miscorrections.

From a theoretical perspective, robustness in subterranean mapping depends on the sensor model (active vs passive), the degree of prior knowledge (e.g., known scale, known rough geometry), and the availability of complementary modalities (IMU aiding VIO, or LiDAR for structure in textureless regions). The literature on SLAM for extreme underground environments highlights adversarial conditions (low light, lack of texture, dust, water) and recommends multi-modal fusion and specialized exploration strategies to achieve resilient mapping. These concerns must be tackled by ensuring the system is protected against damage and has enough battery life for extended mapping sessions (i.e., 1 hour and above).

Representations that preserve free, occupied and unknown semantics (octrees, Truncated Signed Distance Field (TSDF)s with occupancy extraction) make it feasible to define pipelines for automatic 2D plan extraction and semantic segmentation. Semantic mapping introduces supervised learning and domain-shift concerns: pre-trained models may fail in subterranean imagery because of unusual textures and lighting; datasets collected specifically for cave environments are needed to bound generalization error.

RTAB-Map (appearance-based loop closure) and other graph-based frameworks (e.g., g2o optimization, ORB-SLAM variants) are standard building blocks for vSLAM. RTAB-Map is often chosen for long-term online operation and multi-sensor setups but its online keyframe matching strategy can cause operational halts in ambiguous visual environments. The literature therefore recommends hybrid strategies: robust online mapping for coverage and offline reprocessing (or multi-sensor fusion) to produce high-quality reconstructions when needed. Recent surveys on SLAM in extreme underground environments (e.g., underground/planetary cave SLAM reviews) stress a set of remote operation constraints: lack of Global Navigation Satellite System (GNSS), lighting scarcity, particulate matter, and feature-poor surfaces [27]. They recommend robust initialization, active lighting strategies, sensor fusion (VIO, LiDAR), and algorithmic fallbacks for re-localization. These recommendations justify future work suggestions: rugged hardware, dedicated embedded compute, power redesign, and targeted data collection for machine learning applications.

There is an emerging interest in applying ML to cave datasets to produce semantic labels (e.g., CaveSeg dataset and projects) and to perform object/formation segmentation for scientific or robotic use [28, 29]. These works make it clear that curated datasets from the target environment are essential, transfer from standard datasets is limited due to unusual textures and lighting, and voxel or octree representations are a good backbone for combining geometry and semantics. This dissertation proposes data collection and storage purposefully with the intent to build a baseline repository to support future projects and applications.

3 Proposed Method

3.1 System Design and Methodology

A key consideration in the design of field-deployable systems is environmental robustness. Speleological expeditions frequently involve traversing mud, flooded passages, and narrow conduits, while also exposing equipment to rain, wind, and direct sunlight. Mechanical impacts, falls, and surface abrasion are likewise expected. Consequently, the final implementation of the cave mapping system must be sufficiently resilient to withstand these operational stressors.

3.1.1 Hardware Configuration

The physical configuration presented in Fig. 3.2 illustrates the system's components. The prototype version consisted solely of a ZED 2 camera coupled to a laptop computer – adequate for testing in mild environments. This arrangement presented several limitations compared to the intended final configuration, in which a compact computation unit (e.g., an NVIDIA Jetson module) and an extended-duration power source would be integrated into a protective, watertight enclosure (such as a PVC housing). Further discussion of this design is provided in Subsection 5.0.1, noting that resource constraints made it unavailable for the final field tests.

The subterranean and isolated nature of cave systems imposes additional constraints on

Figure 3.1: *NVIDIA Jetson board beside enclosure (left) and laptop (right).*

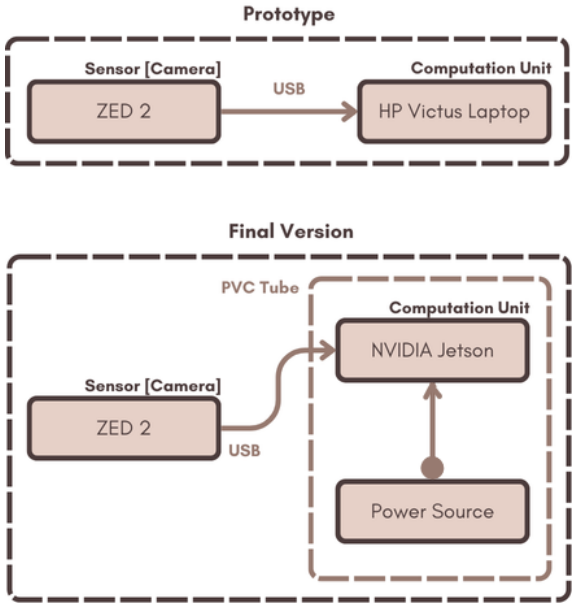


Figure 3.2: *Hardware configurations for the system's experimental prototype and intended final version.*

system architecture. The absence of Wi-Fi and GPS coverage renders GNSS-based localization and cloud-based processing impossible. Although this limitation primarily affects the software framework rather than the hardware platform, it informs component selection. Specifically, the inclusion of radio, Wi-Fi, or GPS modules is unnecessary, as these capabilities cannot be utilized reliably in the target operational environment.

The chosen primary sensor was the ZED 2, a passive stereoscopic camera chosen for its balance of cost and performance. Its suitability for environments with natural light made it a versatile choice over active-light sensors, which are susceptible to solar interference.

The computation unit for the experimental prototype was an HP Victus 16-s0xxx laptop. Its key hardware component for this work was an NVIDIA GeForce RTX 4070 Mobile GPU, which is necessary for the GPU accelerated processing in the ZED SDK which depends on CUDA. The system was also equipped with an AMD Ryzen 7 7840HS CPU, 32 GB of RAM, and a 512 GB SSD.

Considering the devices in Fig. 3.1, transitioning from a laptop to a NVIDIA Jetson board would improve handling of the equipment in the terrain. The smaller form factor of the board allows it to fit snuggly into a speleologist's bag (Fig. 3.3) and different types of enclosures to secure it where a flimsy laptop might break.

Figure 3.3: *Example of a speleologist's bag.*

3.1.2 Software Architecture

The system's software architecture is built upon a stack of open-source tools, chosen to balance performance, flexibility, and rapid development. The base image is derived from the official `stereolabs/zed-ros2-wrapper` for ZED SDK v4.2, which includes Ubuntu 22.04, ROS 2 Humble, and the necessary CUDA Toolkit (v12.6). ROS 2 Humble was chosen as the middleware, serving as the communication backbone that connects disparate software components (nodes). Its publish-subscribe message passing, third-party integrations, and debugging utilities were deemed essential for managing the system's complexity, despite known overheads. The ZED 2 camera is supported by its official SDK and a ROS 2 wrapper, which provides access to rectified stereo images, depth maps, and on-board VIO. For 3D representation, the OctoMap library was selected for its memory-efficient and probabilistic octree data structure, which is well-suited for real-time mapping in unknown environments.

To ensure reproducibility and simplify deployment, the entire software environment was containerized using Docker, as illustrated in Fig. 3.4 and Fig. 3.5. This approach encapsulates all dependencies, from the CUDA Toolkit required by the ZED SDK to the specific ROS 2 and library versions, into a single portable image. This is particularly advantageous for future deployment on embedded systems like NVIDIA Jetson boards. GPU passthrough is configured to allow the containerized application to access the host's NVIDIA GPU, a requirement for the CUDA-accelerated processing in the ZED SDK.

The tooling dependencies for the system on a Linux (Fig. 3.4) or Windows (Fig. 3.5) host machine are split across three layers. Solid arrows indicate that a layer supports another.

Dashed arrows indicate the flow of access to the GPU and graphics. Component bricks depend on others below them. Side by side component bricks symbolize collaboration without dependency. A brief description of the three layers:

1. **Hardware Layer (Bottom):** Physical layer holding hardware components such as GPUs and sensors;
2. **Host Machine Layer (Middle):** Operating system layer of the computation device from which the development environment is configurated and launched;
3. **Container Layer (Top):** Virtual layer upon which the core application runs and in which libraries and other dependencies are installed.

The core of the system is a data processing pipeline implemented as a series of custom ROS 2 nodes. Data flows from the ZED 2 camera, through a filtering and accumulation node, and finally into a mapping node that constructs the 3D environment model. Two primary custom nodes were developed: the `PointCloudAccumulatorNode` for data refinement and the `HybridMapperNode` for map construction.

The `PointCloudAccumulatorNode` is a ROS 2 node implemented using C++ and designed for the aggregation and refinement of 3D point cloud data. It subscribes to `/zed/zed_node/depth/depth_registered` messages published by the ZED wrapper, containing colored point cloud data. Upon reception, each point cloud is transformed into a global map frame using the `tf2` library to ensure a consistent coordinate system. The transformed cloud then passes through a configurable, multi-stage filtering pipeline, implemented using a Chain of Responsibility pattern for modularity (see Fig. 3.7). The default pipeline includes:

- **TF Outlier Filter:** Rejects frames where sensor motion between timestamps exceeds translational or rotational thresholds, mitigating the impact of erroneous localization data.

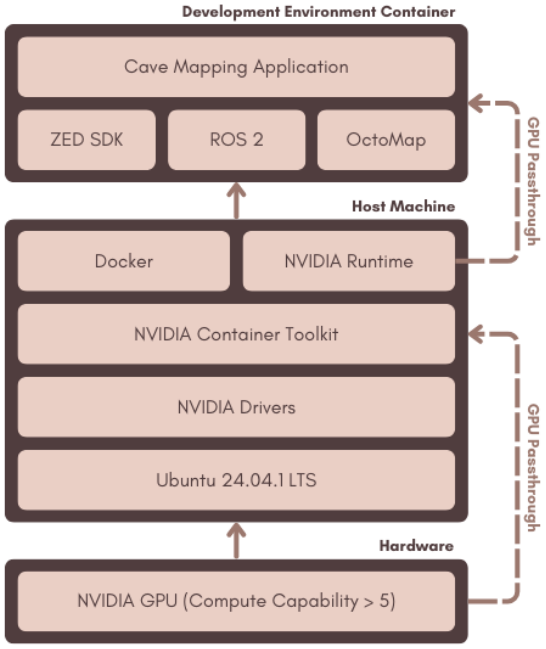


Figure 3.4: *Tooling layers required to support the cave mapping application on a Linux host.*

- **NaN Filter:** Cleans data by removing points with depth values designated "Not a Number" (NaN), placeholders for missing data.
- **Spatial Filter:** A density-based filter that removes sparse noise by eliminating points with few neighbors within a given radius. This filter was omitted from the final tests as it heavily impacted performance. On average, it took around one second to put a point cloud through this filter alone and, visually, it was not impactful enough to justify its inclusion.
- **Statistical Outlier Removal:** A standard PCL filter that removes points whose mean distance to neighbors is outside a statistical threshold.
- **Temporal Filter:** Retains points that demonstrate temporal consistency by appearing in similar locations across multiple frames. As with the spatial filter, this was not included in the final tests due to poor performance and minimal results.

Filtered point clouds are aggregated into a global cloud. To manage memory and computational load, this accumulated cloud is continuously downsampled using a `pcl::VoxelGrid` filter. A key feature is the adaptive voxel scaling mechanism, where the leaf size of voxels in the grid is dynamically adjusted based on the total number of points using a parametrizable curve. This ensures high resolution in the initial sparse map while gracefully managing data density as the mapped area grows. The node publishes both the intermediate filtered cloud and the final accumulated global cloud. For robustness, the accumulated cloud is periodically saved to disk, and processing metrics are logged for offline analysis. An overview of the modular structure of the package can be observed in Fig. 3.8. The package was designed with a modular architecture which leverages a `Curve` interface and a `Filter`

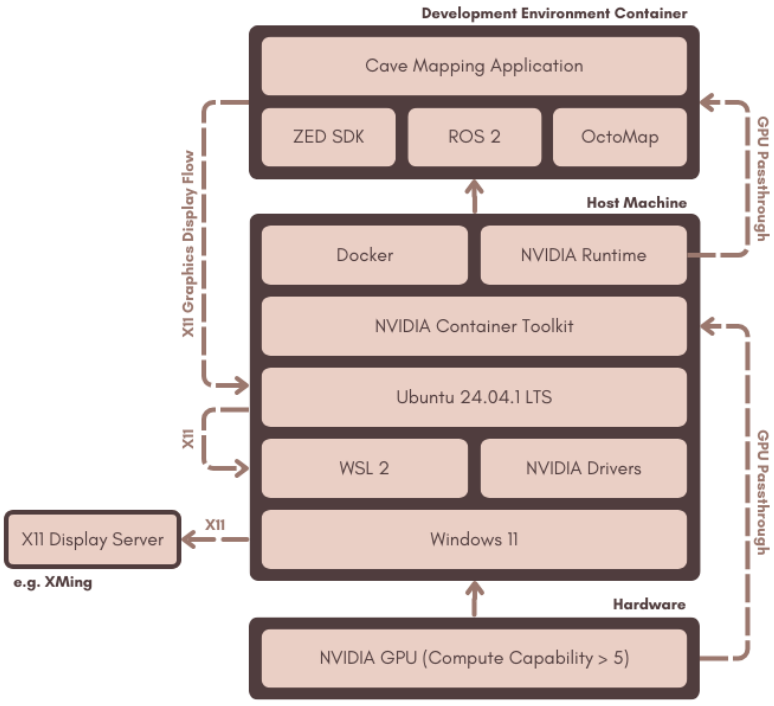


Figure 3.5: *Tooling layers required to support the cave mapping application on a Windows host.*

abstract class to allow the seamless swap of different curve and filter implementations. Method chaining, the Chain of Responsibility and the Template Method patterns were used to make the definition of pipelines ergonomic, with interchangeable filters and open the door to future modification (e.g., introduction of the command pattern to cancel or undo filtration steps and observe the outputs at each step).

Both the results of filtration and accumulation are published to separate ROS 2 topics, making the data available for further processing. Furthermore, these are written to disk at an interval set by the user. This makes the system robust to unforeseen crashes, accidents or battery outages by ensuring at least a partial result of a mapping session is registered. Metrics individual to each filter are logged as records to files in Tab-Separated Values (TSV) format. For each execution of the node, a unique run folder is created to isolate its output files (i.e., `.pcd` for periodic accumulated/fused point clouds and `.tab` for TSV format logs). This prevents outputs from different runs from overwriting each other and helps organize experimental data. They are stored to the `savefolder` directory passed as a parameter to the node. Likewise, the `savefile` parameter is prefixed with the timestamp (using an underscore as a separator) following a `YYYYMMDD-HHMMSS` format. Additional information about the contents of the produced artifact for each written file is appended with further underscores as separators. All accumulated point clouds saved during this specific run will be placed inside this directory along with logs and additional files.

The `HybridMapperNode`, the second custom node, is responsible for constructing the final 3D map. It ingests the filtered point clouds from the accumulator and inserts them into an

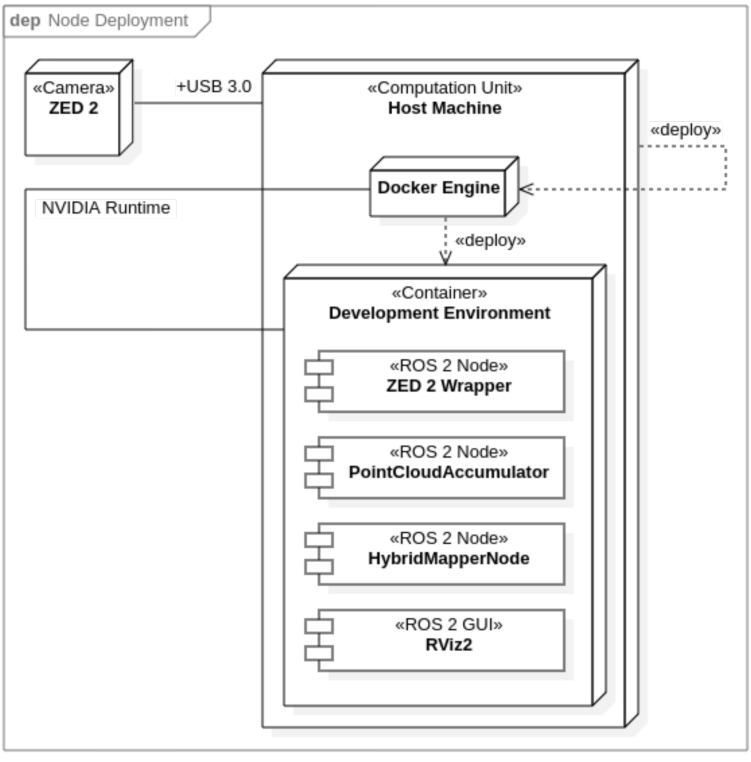


Figure 3.6: *Deployment diagram of the ROS 2 nodes.*

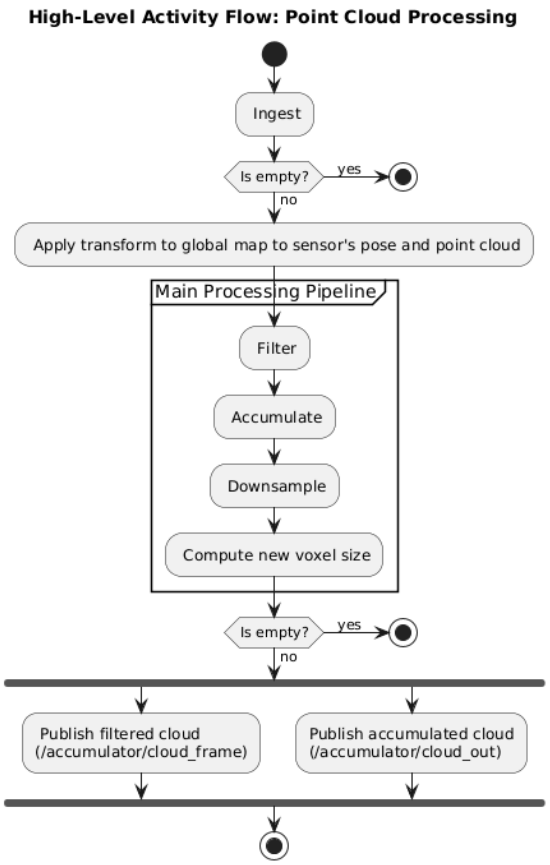


Figure 3.7: *Activity diagram of point cloud processing performed by the PointCloudAccumulatorNode.*

OctoMap. A significant challenge in stereoscopic mapping is the phenomenon of "ghosting", where erroneous depth measurements, often along object edges or in poorly textured areas, create persistent artifacts in the map. This is visible in Fig. 3.9, where occluded areas behind tables are incorrectly marked as occupied space. The green line shows the odometry – the progress of the camera's position from right to left. By capturing images from the starting point on the right side, the areas occluded by the tables were marked as occupied. The point clouds published by the ZED wrapper node were polluted due to misregistration of depth values along the edges of the tables. Despite depth confidence thresholds and special parameters to mitigate the impact of the other aforementioned factors, the polution was severe enough that even the filtration pipeline in the `PointCloudAccumulatorNode` can't remove the noise. As these points are inserted into the octree, they are persisted. This means that even as the camera captures the surfaces previously occluded by the tables the noise won't be removed.

To mitigate this, the `HybridMapperNode` leverages the raycasting capabilities of the OctoMap library. When a new point cloud is inserted, rays are cast from the current sensor position to each point in the cloud. Voxels along this ray that were previously marked as occupied are updated with a lower occupancy probability. This process effectively "clears out" the free space between the sensor and the newly observed surfaces, removing ghost

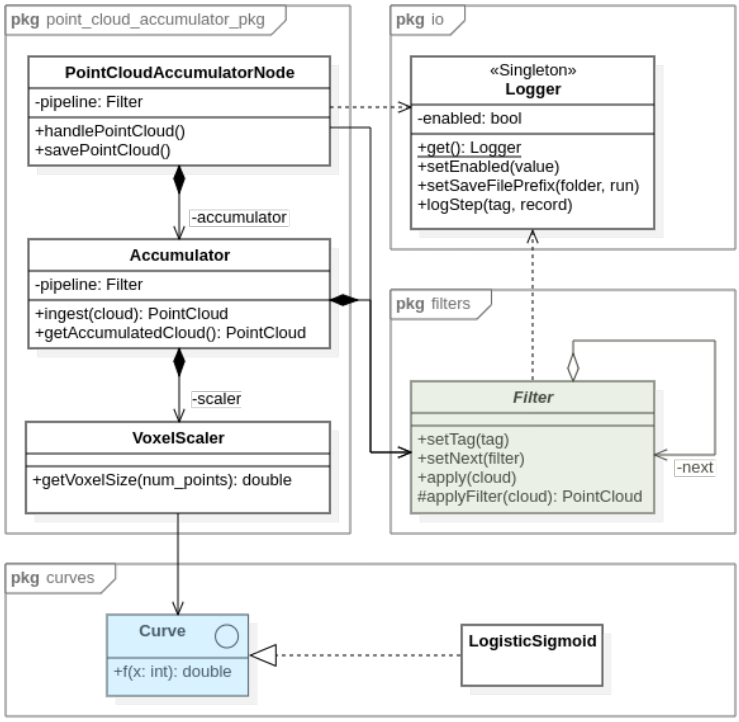


Figure 3.8: *Class diagram of the Point Cloud Accumulator package.*

artifacts. The result, as shown in Fig. 3.10, is a much cleaner and more accurate representation of the environment, where previously occluded areas are correctly modeled as free space once they become visible. This real-time correction mechanism is a key advantage over offline reconstruction methods like RTAB-Map, which was found to produce similarly poor reconstructions which suffered from ghosting. Logging is handled the same way as for

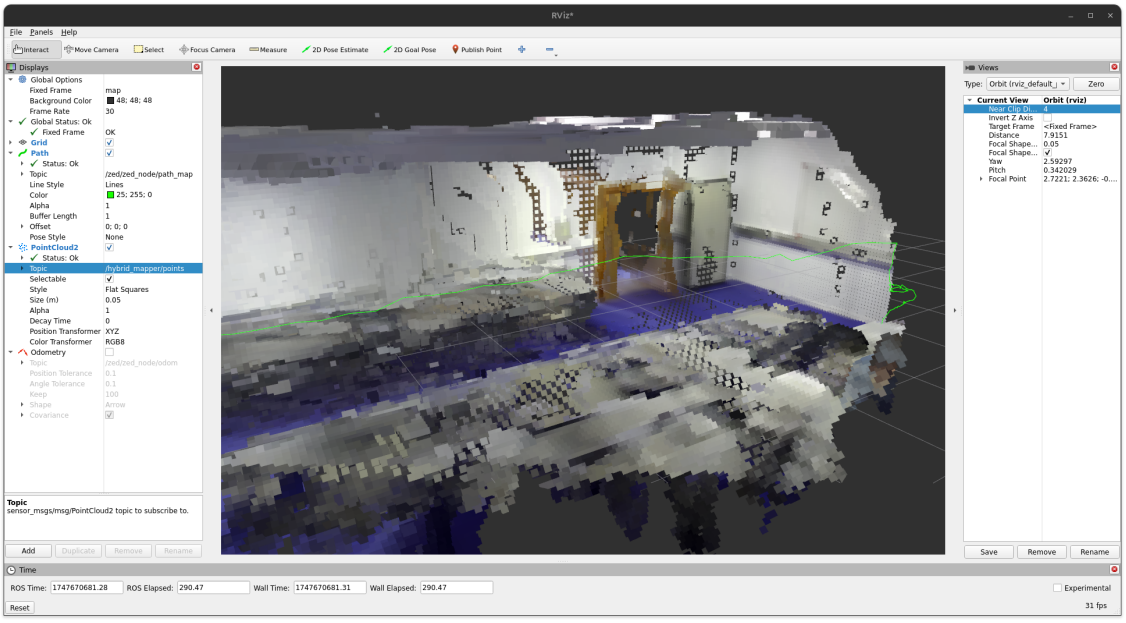


Figure 3.9: *Partial OctoMap of one of ISEL's laboratories without using raycasting to free unoccupied nodes.*

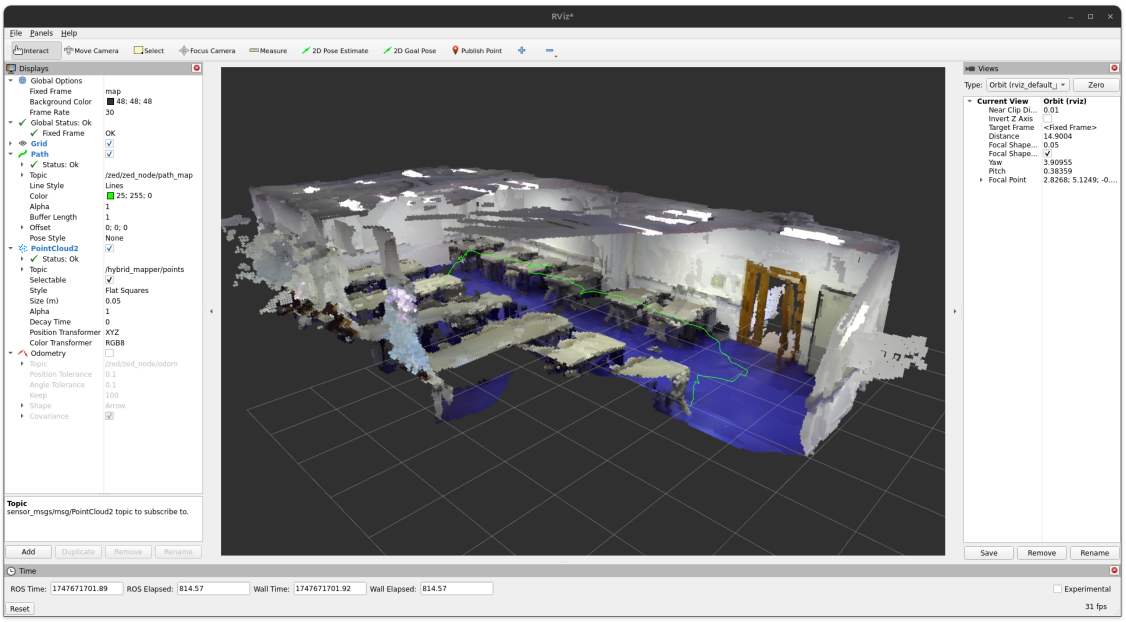


Figure 3.10: *Partial OctoMap of one of ISEL's laboratories using raycasting to free unoccupied nodes.*

the `PointCloudAccumulatorNode`, except the main artifacts saved to files are octomaps in OT format instead of point clouds.

3.1.3 Workflow for Topographic Acquisition

The workflow used for 3D mapping and, as such, topographic acquisition, consisted on following a series of steps to ensure the best results. Firstly, the development environment container was restarted. As a rule of thumb, a minimum of one Bash terminal per node to be launched and an additional two (one for RViz2, another free for use) were attached to it, allowing an operator to track logs for each one in real time. The ZED wrapper node was launched first, along with RViz2. The camera's odometry and registered point clouds were observed through Path, Odometry and PointCloud2 displays (as demonstrated in Figs. 3.9 and 3.10).

The starting position for the camera was always as horizontal to the ground as possible, facing directly towards a place with rich visual features and abundant detail. Fig. 3.11 is an example of a good initialization scene. There, paintings and plates on the walls break up the monotonous, textureless wall surface with contrasting, well-defined geometric shapes. Furniture with clutter on top provides additional detail from which to extract keypoints to situate the sensor between frames in the SLAM algorithm. The camera was moved around the space being mapped out in an effort to optimize the pose graph while monitoring the displays and thus improve the results produced by the subsequent mapping. Once instability in the odometry (i.e., sudden corrections to sensor pose or jumps in the detection of surfaces) was no longer detected, the camera was brought back to the initial position and the rest of the nodes and RViz2 were launched by the order seen in Fig. 3.6 (descending).

Figure 3.11: *Panoramic shot of a wall with rich visual details for SLAM algorithm initialization.*

From that moment onward, the messages published by the ZED wrapper node to topics relevant to the function of the system and debugging were recorded to a `rosbag` for analysis *a posteriori* or to be replayed – a way to reproduce the real-time mapping off-site by substituting the ZED wrapper node with the replay. The map was tracked in real-time by switching the topic represented by the PointCloud2 display on RViz2. To save resources, both point clouds and octomaps can be represented using flat squares accepting that knowing where the center of each node in an octree is sufficient for visualization. Other options available for representation as shown in Fig. 3.12, where boxes were selected instead of the economic flat squares.

The point clouds (`.pcd`) and octomaps (`.ot`) saved to disk were then visualized using the `open3d` library for Python and a custom, browser based map viewer, respectively. The logs, kept in TSV format (`.tab`), were used to perform an analysis using a Jupyter Notebook to gain insight into a variety of statistics such as processing time per filter, filtered points per cloud at each stage of the pipeline, track frames identified as outliers due to misregistration, among others.

The wrapper node for the ZED 2 camera was configured to limit depth detection to the range between 0.3m and 15.0m. A Region of Interest (ROI) was defined – a rectangle between the coordinates (0.2, 0.2) and (0.8, 0.8) of the captured images – to ignore points on the border of the camera's Field of View (FOV). Additionally, the `image_height_ratio_cutoff` was kept with its default value, considering only their lower half. These parameters were set to

provide reliable mapping and accommodate for a human operator manipulating the camera. By having a cut-off above the horizontal line the camera is pointed at and discarding the periphery, a speleologist making use of the system can reliably know what they are pointing at, capture it intentionally and easily avoid other areas.

For the `PointCloudAccumulatorNode`, the curve was parametrized by setting maximum and minimum thresholds at 1,000 and 1,000,000 points, respectively. Voxel size was set to vary from 0.01m to 0.1m according to them. Values were bounded to a logarithmic curve following the function described by Eq. 3.1, using a steepness of 5.0. Effectively, this means that the curve maps how fine or coarse the resolution of 3D maps for a space are. For small spaces, a finer resolution with smaller voxels is targeted. The inverse is true for large spaces. Constraints for filtering transform outliers were set, discarding those with translations greater than 0.3m or rotations greater than 15^{o} around any given axis between consecutive frames. The save interval was set to 90 seconds, ensuring results were saved to disk regularly. The final results were saved on shutdown.

$$f(x) = y_{\min} + \frac{y_{\max} - y_{\min}}{1 + \exp\left(-\frac{\text{steepness}}{\text{thr2}-\text{thr1}} \cdot \left(x - \frac{\text{thr1}+\text{thr2}}{2}\right)\right)} \tag{3.1}$$

The `HybridMapperNode` was configured with a 0.05m size per voxel and regular saves every 90 seconds.

Figure 3.12: *OctoMap of a living room, represented using boxes in RViz2.*

4 Experimental Results

4.1 Experimental Evaluation

4.1.1 Test Site and Conditions

The Lapa dos Pocilgões is located in São Bento (Porto de Mós, Leiria), within the Parque Natural das Serras de Aire e Candeeiros (PNSAC), and is accessible via the PR3 hiking trail [30]. It constitutes a karstic cavity developed in Jurassic limestones of the Maciço Calcário Estremenho [31]. The cave has two possible entrances: (i) one corresponds to a former subterranean conduit subsequently exposed through erosion and a collapsed roof (collapse feature accessible via abseiling), (ii) the other corresponds to a descending tunnel connected to a trail (accessible on foot). The surrounding area exhibits characteristic karst landforms, including limestone pavement, dolines, and the vertical shaft (pit), and is associated with an extensive subterranean aquifer of high hydrogeological relevance [31].

The vegetation is dominated by Mediterranean scrub, holm oak (*Quercus ilex*) stands, and patches of Portuguese oak (*Quercus faginea*), complemented by rosemary (*Rosmarinus officinalis*), seasonal orchids, and traditional agricultural landscapes such as olive groves and pastures [32]. The climate represents a transition between Atlantic and Mediterranean regimes, with annual precipitation ranging from 900–1300 mm. Summers are typically hot and dry, while winters are mild; however, rainfall infiltrates rapidly through the limestone, resulting in limited surface water availability [31]. The site also preserves elements of cultural heritage (e.g., dry-stone walls, prehistoric remains) and constitutes habitat for cave-dwelling fauna, notably several bat species.

The test site was approached during daytime in two periods: one in the morning and another in the afternoon. The traditional method of cave surveying was carried out in the late morning. Overall, it was a hot and dry summer day with no clouds in sight, providing strong illumination from sunlight alone. Tests of the system discussed in this dissertation were conducted during the early afternoon period. Access to the site and traversal while inside were hindered by dense vegetation, loose eucalyptus leaves, and clay soil, combined with steep inclines just beyond the cave entrance.

Within the cave, a collapsed ceiling allows natural light to illuminate the interior. Vegetation including eucalyptus, other trees, and bushes thrives among the rockfall. A notable morphological feature is a narrow recess at the rear of the cave, forming a hook-shaped, dead-end tunnel just large enough to accommodate a standing person. This alcove, reminiscent of a "horn" in plan view, represents a blind passage that terminates abruptly, and it is typical of minor, constricted chambers frequently encountered in karstic environments (hence known as a "blind tunnel"). The large eucalyptus tree at the center constitutes a visual obstacle with interesting implications in testing laser sensors against cameras. The latter, along with top of the line LiDAR, won't be able to ignore the noise presented by the foliage, while manual laser Time-of-Flight (ToF) devices can be carefully handled manually to register measure between branches through the treetop (Fig. 4.1).

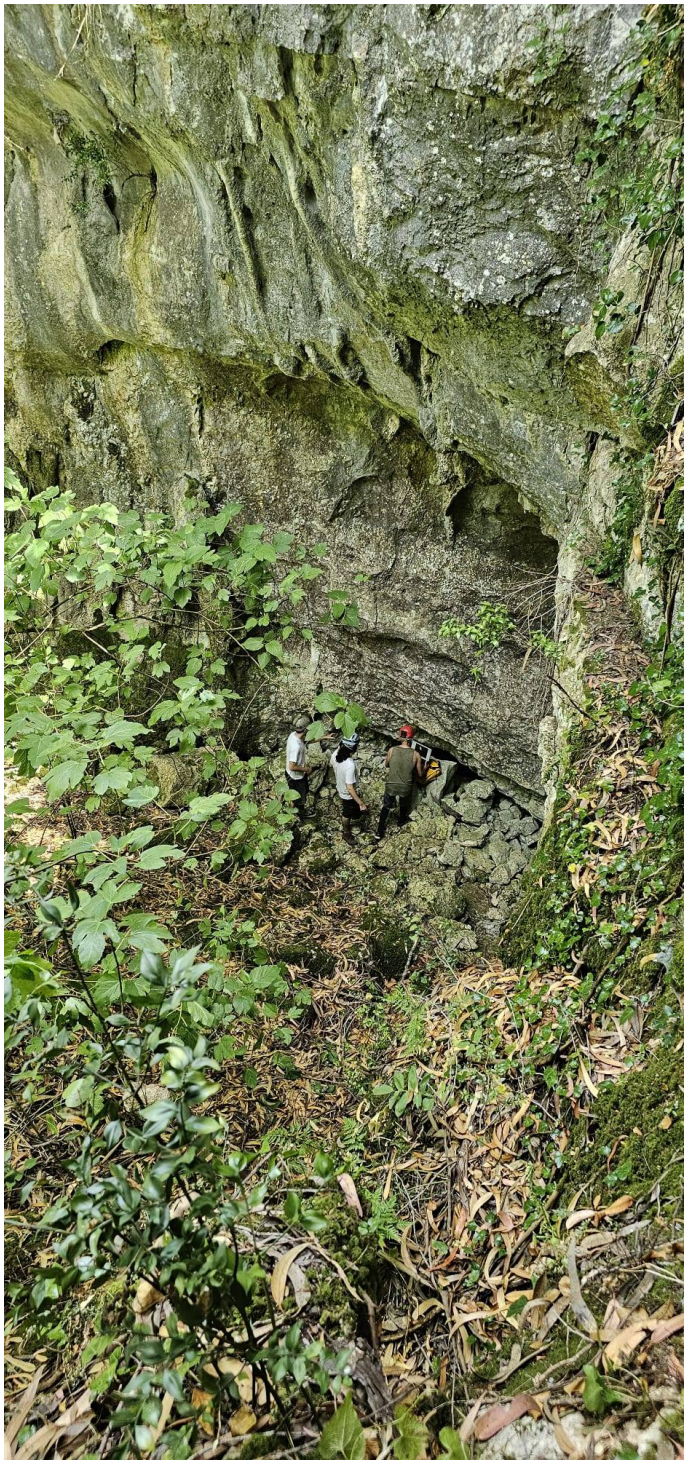


Figure 4.1: *Bottom of the south-east cave cliff, displaying foliage and rock debris.*

4.1.2 Comparative Methodologies

The field tests to evaluate the mapping system were conducted, as described in Subsection 4.1.1, in two phases: traditional surveying and surveying with stereoscopy. The traditional survey served as the control, both in procedure and in the resulting maps (Figs. 4.2 and 4.3). Comparison was carried out at two levels: (i) the cartographic products, assessed in terms of legibility, clarity, and representational accuracy; and (ii) the survey processes,

assessed in terms of effort, precision, and workflow requirements. This design provides a consistent basis for evaluating the efficacy, efficiency, and usability of stereoscopy in topographic mapping.

The first phase, employing traditional surveying techniques, was conducted with a Leica DistoX310 fitted with a magnetic compass. Readings were recorded using the TopoDroid application on an Android device paired to the instrument via Bluetooth. Although methodological variations exist, this workflow is representative of tool-assisted surveying widely regarded as "traditional" within the speleology community at the time of writing. In this approach, survey stations were established as reference points, with successive stations chained along the cave traversal. Each station was selected based on its suitability for capturing geometrically informative measurements of the cave's topography. At each step, a set of distance, inclination, and azimuth readings was collected using the laser pointer, thereby parameterizing the vector between the station and the target point. These data supported the construction of maps from which multiple projections of the cave could be derived. The accuracy of this method depends strongly on reliable compass readings, which can be compromised in environments with magnetic interference, such as the presence of ferromagnetic rock. The resulting maps thus served as the control products for comparison with stereoscopic surveying (see Subsection 4.1.3).

The second phase employed stereoscopy, using the software stack described in Section 3.1. Because vertical traversal while carrying the equipment was impractical, data collection was divided into two runs: one beginning at the base of the cave near the entrance to the blind tunnel, and the other at a large rock near the main entrance (see Fig. 4.2). These positions, chosen for their visibility and feature richness, served as functional analogues to survey stations, though the camera was moved continuously during acquisition. Effective operation required abundant visual references, including textured walls, distinctive rock formations, and structural features such as the cave entrance arches and the blind tunnel opening, combined with sufficient illumination for robust SLAM depth registration. Prior to mapping, the ZED SLAM node was initialized to stabilize the pose graph, after which the full mapping pipeline was launched. The camera was moved freely within a radius of approximately two meters around the starting points, oriented toward salient topographical features from multiple angles to trigger loop closure and enable pose graph optimization. This initialization stage, lasting one to two minutes, substantially reduced errors associated with odometry drift and minimized surface duplication due to misaligned registration.

4.1.3 Results and Comparison

The topographic maps produced using the traditional method present the relevant characteristics of Lapa dos Pocilgões with adequate precision, without overloading a speleologist with unnecessary information. The symbology adopted by the speleology community allows for the succinct identification of terrain features, even in a 2D projection. In Figs. 4.2 and 4.3, the usage of this symbology can be observed in detail. The entry point is indicated by a black arrow and the gradient (indicative of the slope and its direction) by curves and small arrows. The distribution of more relevant rocks and clay soil or mud

is represented with appropriate markers (small scattered dots, in the case of clay/mud), helping to exercise caution and plan the route carefully. Steps or ledges are indicated by lines perpendicular to lines that might otherwise represent walls. Thus, one becomes aware of where there is a risk of falling from a ledge or precipice. Lastly, symbols in the shape of the letter "T" indicate the formation of a ceiling (head outwards, stem inwards). Finally, triangles demarcate the stations taken during the topographic survey. Through this set of symbols, speleologists efficiently and effectively convey the information that is useful to them for exploring caves. As mentioned in Subsection 4.1.1, the DistoX310 sensor allowed for capturing the topography even through the treetops.

The perspective presented in Fig. 4.3 highlights the progression of the blind tunnel inside the wall over other recesses, such as the alcove behind it.

The 3D maps produced using stereoscopy represent the cave as it is seen. A resolution of 5cm (length of the edges of each voxel) was chosen for the capture to keep the occupied RAM and latency (processing time per point cloud) within limits considered stable for the laptop used on site. Like several other parameters of the nodes implemented in this context, the resolution can be adjusted to obtain results closer to a reconstruction of reality (higher resolution, smaller edges) or a rough sketch of the space. With the resolution set for the experimental evaluation, detail was lost. It may be difficult to discern what is a rock and what is soil. Nonetheless, the general features of the terrain were captured accurately. Foliage proved itself a real obstacle to visual depth registration. The camera was unable to accurately estimate the depth of surfaces behind the treetops. Furthermore, branches are not static objects. Any movement provoked by wind contributes to an increase in the noise caused by them.

Although the traditional methodology requires greater involvement, it enables flexible decisions about which details should be represented with greater accuracy. Some are moved out of focus or not even presented at all depending on the intended use for the map. This is where 3D maps aiming to represent caves with different techniques come in. Depending on the use case, the data stored in them may be used to extract valuable information to aid in decision making. Provided a capable computer is used, the method discussed in this dissertation can be improved to produce results similar to the reconstructions achieved using photogrammetry, allowing people to visit caves using Virtual Reality (VR) or projecting them using Augmented Reality (AR). Furthermore, keeping caves registered in a file storing a hierarchical, extensible data structure with a supporting library makes them immediately more workable with. With the right 3D editing and viewing tools one can slice caves up in different levels, use algorithms to project 2D views from different perspectives, make spatial annotations about features of interest and other operations, the rich 3D representation offers data with value beyond producing 2D maps. In their raw form, the results obtained using stereoscopy are visually messy. Further processing would allow us to glean different representations and key insights from the structured data.

The views shown in Figs. 4.4-4.7 display different perspectives of the bottom of the cave. In Fig.4.4, the blind tunnel can be seen on the left, while foliage made registration of walls on the top right impractical. This figure showcases the gradient descent from right to left

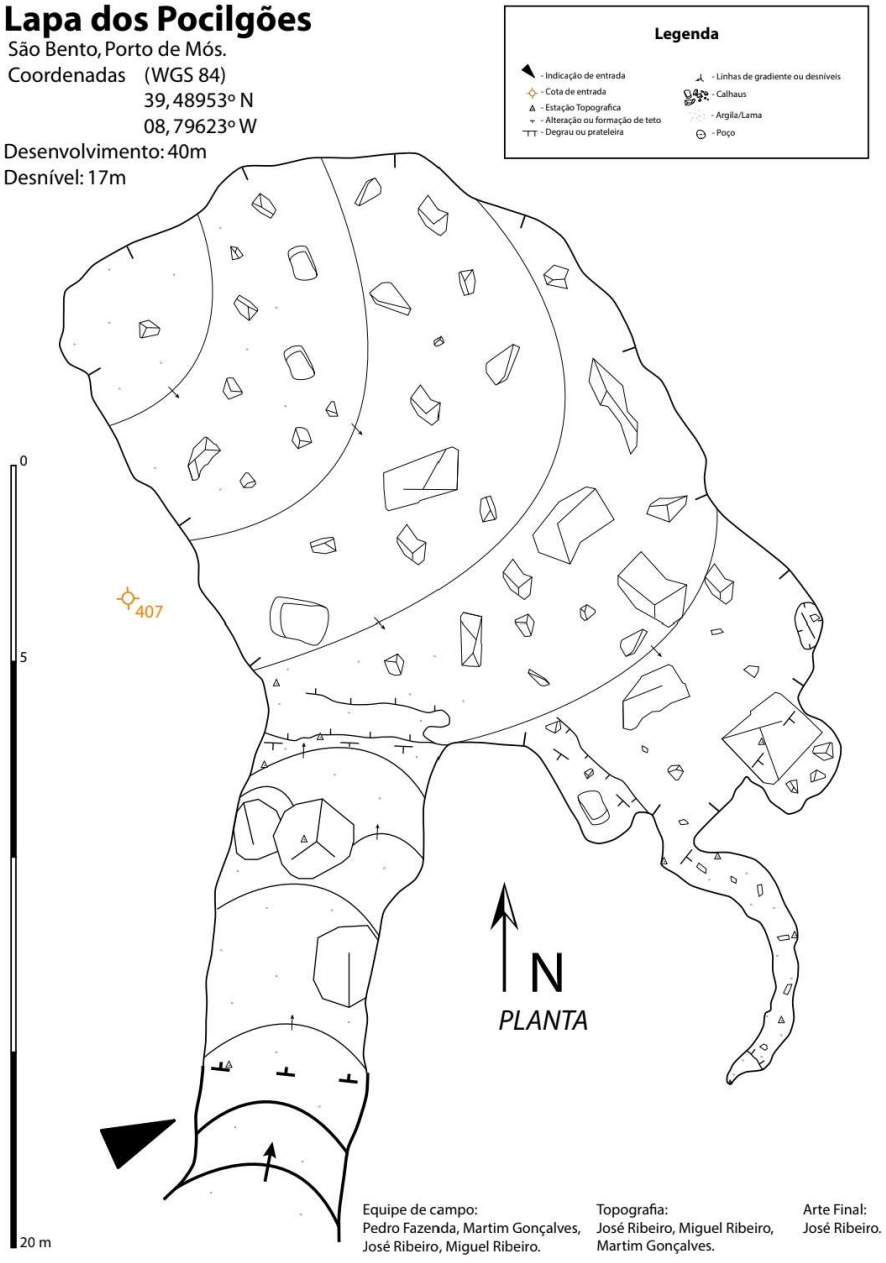


Figure 4.2: *Plant of the Lapa dos Pocilgões.*

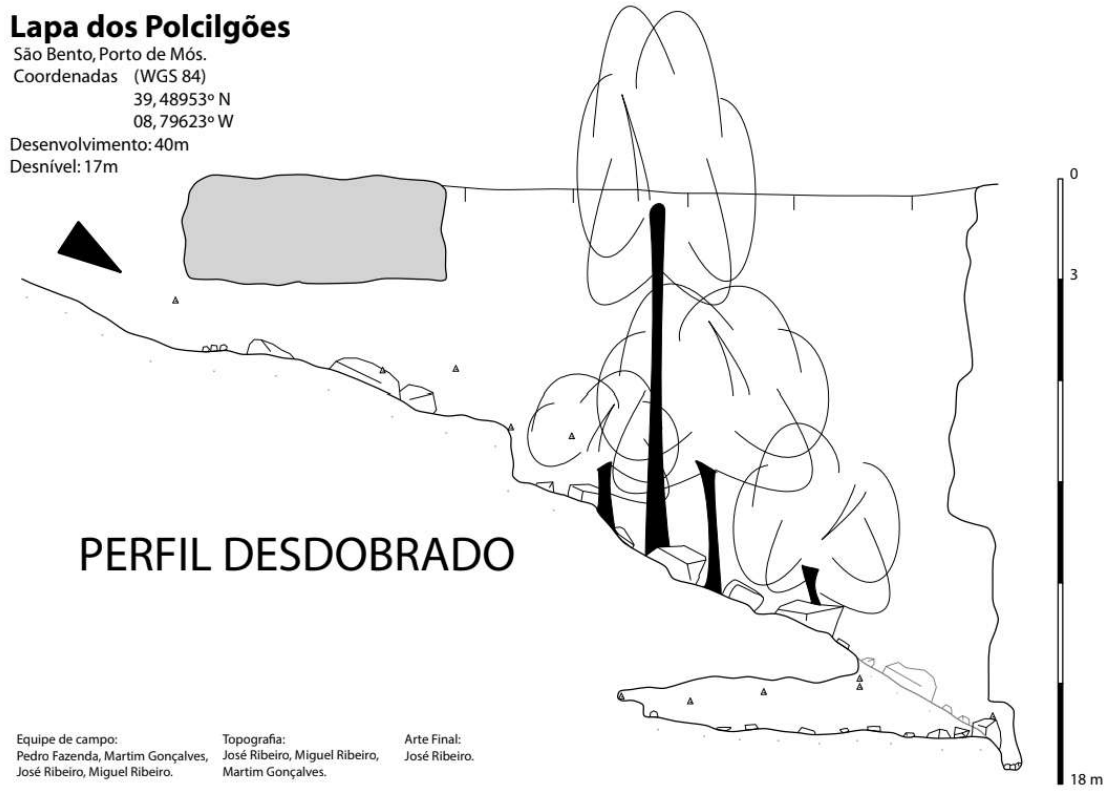


Figure 4.3: *Unfolded profile of the Lapa dos Pocilgões topography.*

and the rocky terrain. An overhead view matching the orientation of the plant of the cave can be seen in Fig. 4.6. Fig. 4.7 highlights the cliff-side overhang above the opening in the wall (bottom center), leading into the blind tunnel. The photograph in Fig. 4.5 shows the same opening on its right side.

Comparing the cross-sections of the cave entrance (Figs. 4.8 and 4.9), the maps produced using stereoscopy are less visually clear in comparison to those drawn by hand (Figs. 4.2 and 4.3). Areas covered in foliage become extremely noisy (left-most and right-most extremes of the cross-sections). Depth perception is lost to the user, especially when presented as images in a non-interactive visualizer.

In Fig. 4.10, a section of the OctoMap produced as a result of run 6 presents the cave's entrance with the ceiling removed. This serves as an exemplary occurrence of the aforementioned problem common with pose graph optimization during real-time mapping. The walls of the entrance are duplicated, facing different directions as a result of a recalibration of the camera's pose. The annotated arrows in red and green highlight the flow of the two sets of walls into the cave.

An analysis of the storage characteristics of two primary file formats employed in 3D mapping, PCD and OT, was conducted. Understanding their storage efficiency and scaling properties is crucial for optimizing data pipelines and making informed decisions about data storage strategies for further research. Evaluation of the experimental data was conducted by extracting relevant statistics about its speed and memory consumption. Tables 4.1 and 4.2 (corresponding to analyses of the `PointCloudAccumulatorNode` and `HybridMapperNode`,

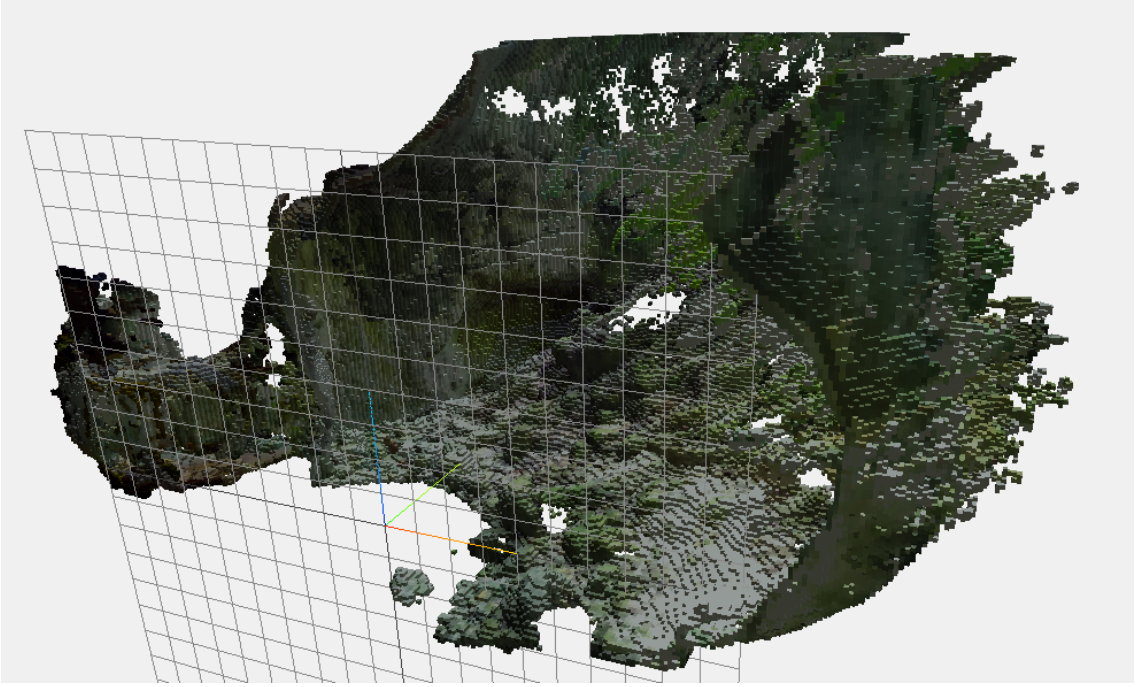


Figure 4.4: *View of the cave bottom (facing southwest).*

Figure 4.5: *Photograph showcasing the bottom of the cave.*

Figure 4.6: *Overhead view of the cave bottom (facing north).*

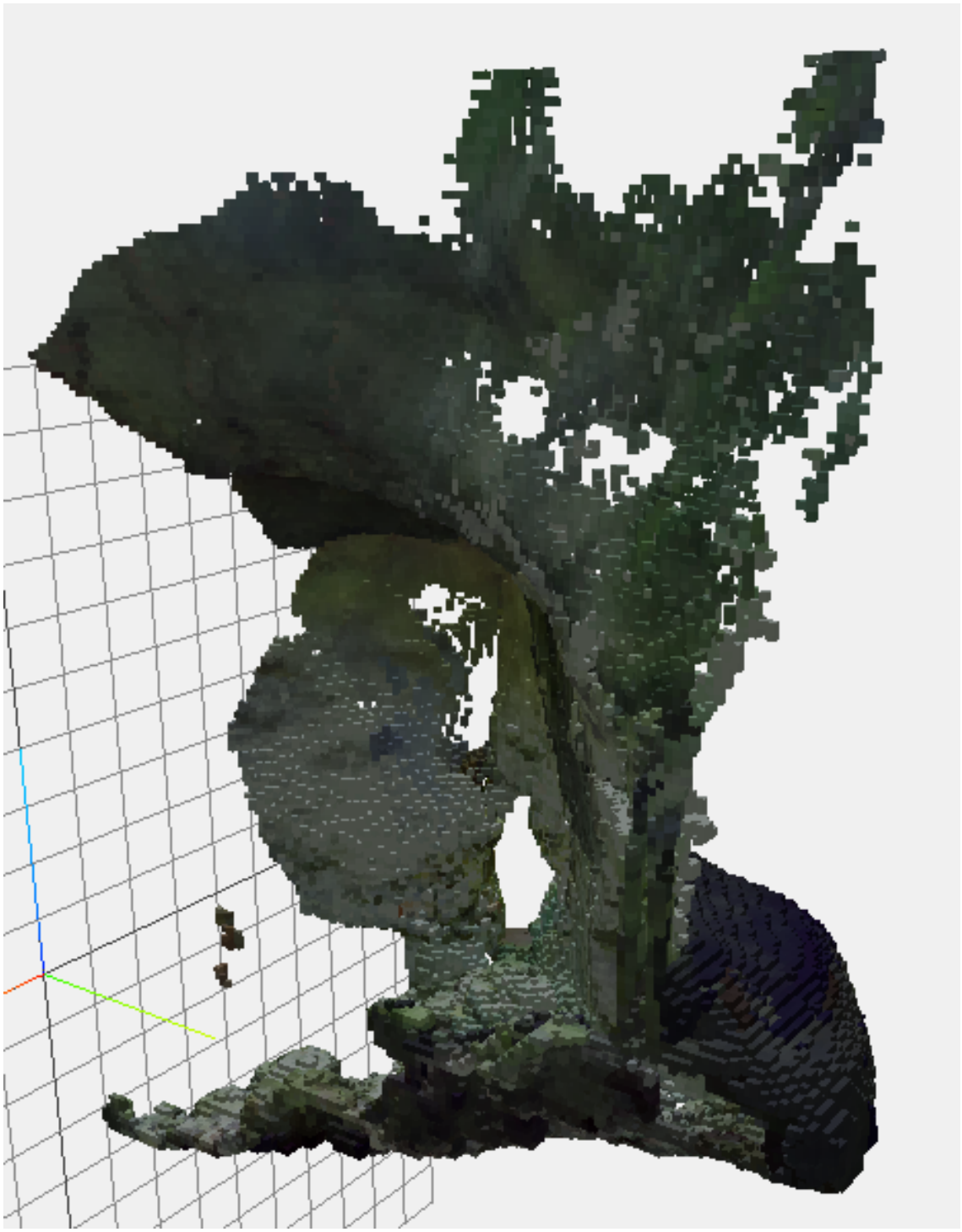


Figure 4.7: *Frontal view of the cliff overhanging the blind tunnel entrance (facing southeast).*

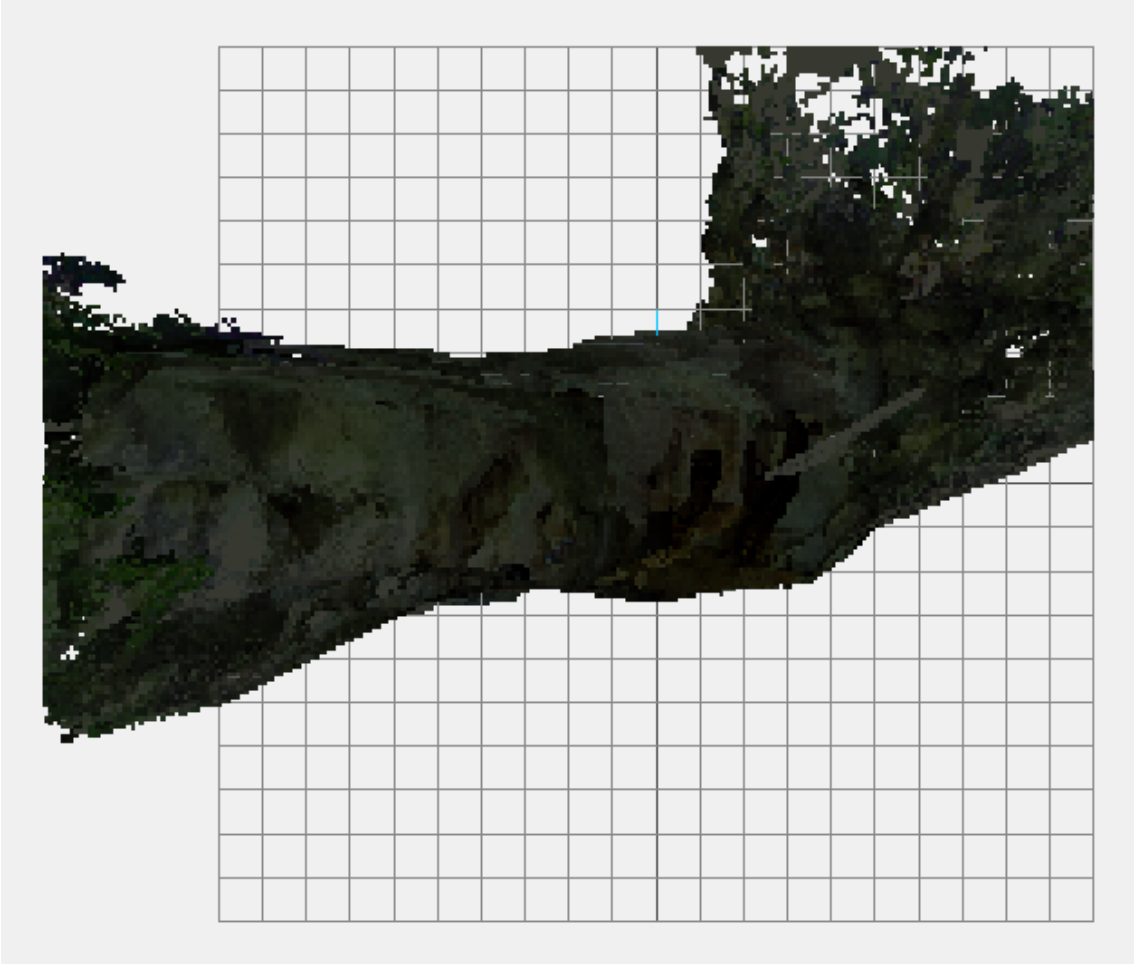


Figure 4.8: *Cross-section of the cave entrance (facing east).*

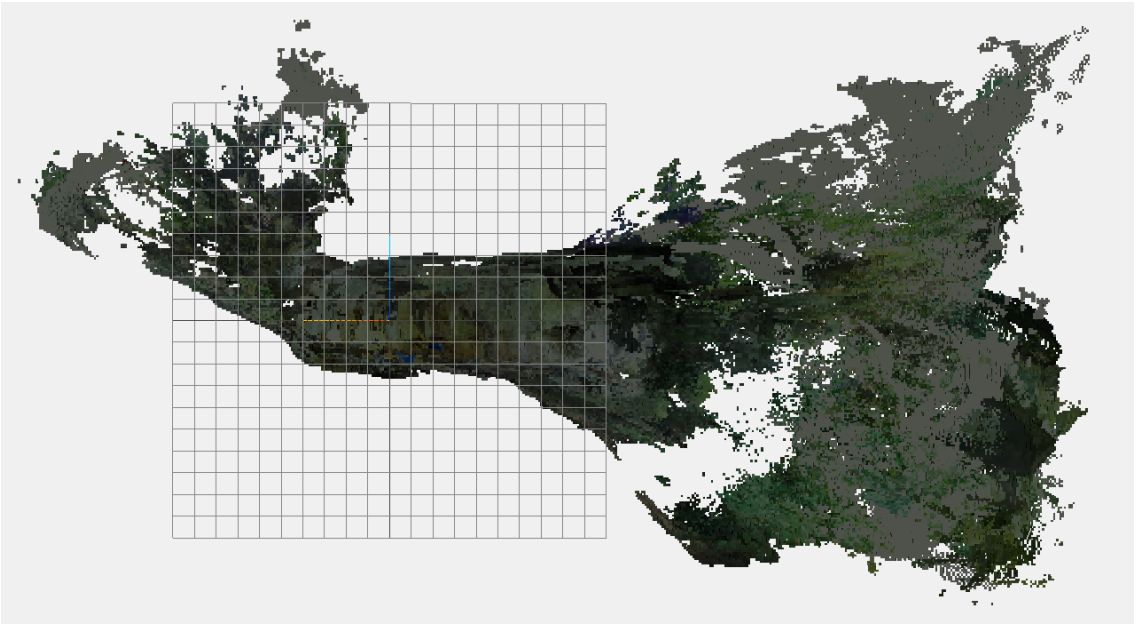


Figure 4.9: *Cross-section of the cave entrance (facing west).*

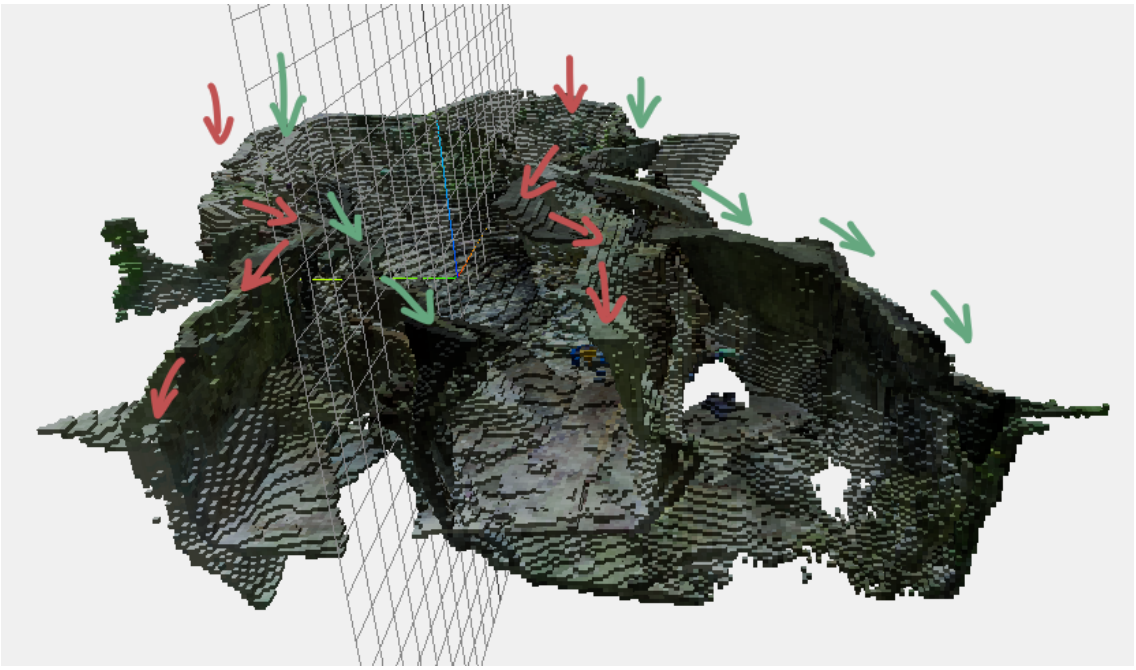


Figure 4.10: *Annotated view of duplicated, cross-secting walls at the cave entrance (facing south).*

respectively) display four metrics of high importance to monitor the system's performance. The artifacts produced by the last six mapping runs were collected for this purpose. The first three runs were carried out in an apartment scenario, while the last three correspond to those carried out in the cave. Run number four served to map out the bottom of the cave. Run number six is a reattempt of run number five. These served to map out the top of the cave, near the entrance, but as will be discussed, the fifth run presented here demonstrated anomalous performance and was as such considered an outlier.

The `PointCloudAccumulatorNode` (Table 4.1) was shown to process each registered point cloud put through the filtration pipeline in a little under a second, averaging 0.82 seconds if disregarding the outlier run. File size on disk (Fig. 4.11) was kept generally low even at high point counts. The adaptive resolution mechanism further contributed to this as the larger the accumulated cloud, the coarser the resolution, attenuating memory and disk space consumption both during execution and in storage. The case is not the same for the octrees produced by the `HybridMapperNode` (Table 4.2), storing the octomaps. Due to having a fixed and finer grained resolution in comparison to the point clouds, as well as explicitly modeling unoccupied space, octomaps representing the same area as a point cloud (i.e., artifacts produced by the nodes for the same run) hold more detail. As such, they occupy more disk space, demonstrating size increases between 9 and 30 times the size of their respective point clouds. Larger octomaps showed a severe latency in the insertion of new point clouds. For small maps, such as those in runs number one and three, multiple point clouds were inserted into the octree per second, on average. For the larger maps, insertion took several seconds. This could be attributed not only to the overhead associated with updating the occupancy likelihoods of a larger data structure, but also the necessity for raycasting to perform more collision checks in the search for freed nodes. These findings

Table 4.1: Point Cloud Processing and Resulting PCD File Statistics

#	Freq. (Hz)	Size (MB)	Num. Points	Res. (mm)
1	1.37	0.9	58,530	64
2	1.35	1.5	93,303	90
3	1.04	0.8	52,677	57
4	1.19	1.1	72,651	77
5	0.34	1.4	85,309	86
6	1.23	5.8	367,121	93

Table 4.2: OcTree Insertion and Resulting OT Statistics

#	Freq. (Hz)	Size (MB)	Num. Voxels	Res. (mm)
1	2.13	7.9	993,273	50
2	0.86	25.5	3,191,854	50
3	1.79	9.9	1,239,951	50
4	0.66	30.6	3,823,457	50
5	0.20	42.6	5,330,965	50
6	0.57	60.3	7,535,709	50

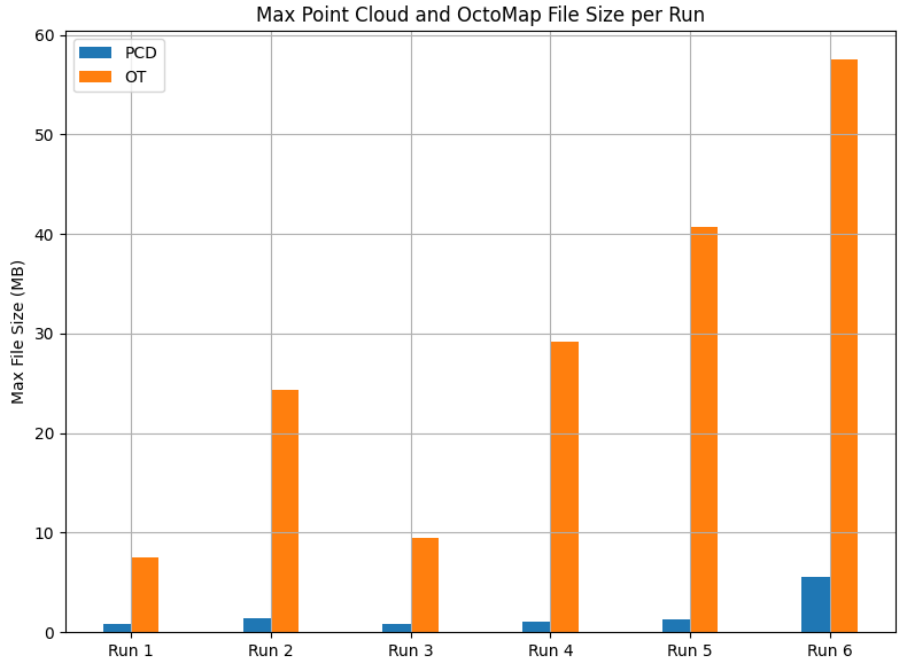


Figure 4.11: *Comparison of file size occupied by the largest artifacts (.pcd and .ot) produced for each mapping run.*

imply that additional testing must be done when mapping large environments, as the insertion of new points in already heavy octomaps may severely slow down and make the process unfeasible. Bottlenecks and possible optimizations would need to be studied to address this problem.

The boxplots in Fig. 4.12 show the processing times for each filter applied to ingested point clouds in the `PointCloudAccumulatorNode`'s pipeline during run number two. Time elapsed during the operations for discarding point clouds with transforms considered invalid for the sensor's pose ("tfoutlier") and cleaning out points with invalid depths ("nan") is

measured in the order of microseconds. The Statistic Outlier Removal ("sor") filter takes the brunt of the processing time. The total time elapsed while processing point clouds matches the observations in Table 4.1. Previous tests following the same type of analysis proved other filters developed and tested in the scope of this study to be resource intensive, doubling or more the time taken to process data and consuming increased memory, leading to stalling in some cases. That motivated the selection of a minimal filtration pipeline consisting only on the aforementioned filters to ensure better stability.

The results presented on the lineplot in Fig. 4.13 indicate that file sizes for both the PCD and OT file formats scale linearly with the number of points or voxels in them contained. The line for the PCD format demonstrates a steep slope. In contrast, the OT format exhibits a less steep lope. This means that for an equal number of points or voxels, an octree always occupies less disk space than point cloud data – the average cost per voxel is lower. It is important to note the PCD files are in binary compressed format and the OT files were written in binary format. Overall, this suggests that as the mapped environment grows, the OT format becomes progressively more space-efficient compared to PCD. It not only is more scalable, but provides structured data too, which may be valuable for the development of algorithms and systems that work with map data.

Across all statistics, run number five demonstrated the worst performance in terms of processing speed. The results were skewed enough to consider this run an outlier. The processing times for each run, in Fig. 4.14, are split into PCL and OT, representing the `PointCloudAccumulatorNode` and `HybridMapperNode`, respectively. The discrepancy between run number five and the rest, in both nodes, is of hundreds of milliseconds, but impacting the OctoMap building node (OT) much more severely. Not only is the mean

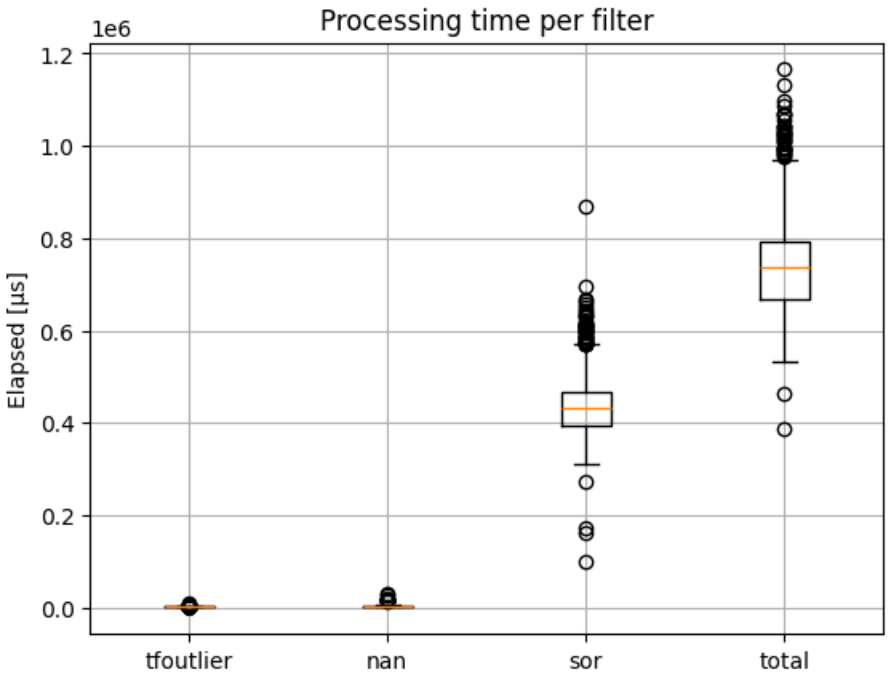


Figure 4.12: *Elapsed processing time per each filter in the* `PointCloudAccumulatorNode`*'s pipeline.*

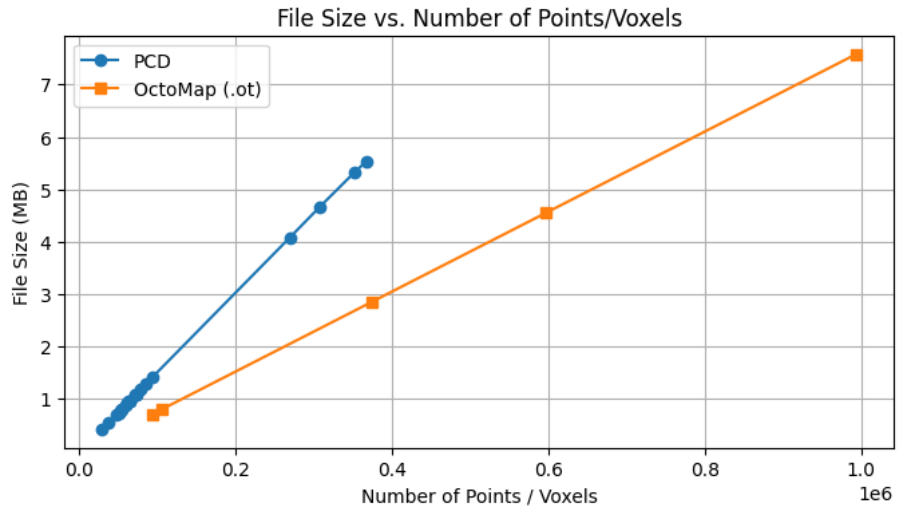


Figure 4.13: *Scalability of file sizes storing Point Cloud Data (.pcd) and OcTree (.ot) file formats.*

processing time for it just below five seconds, but it demonstrates a very high standard deviation, hitting an extreme of ten seconds per point cloud insertion. The elapsed times logged to the TSV file (Table 4.3) for the run indicate that insertion was already taking longer than excepted on launch. While no concrete evidence was found pointing towards the cause for this incident, it is feasible that the summer heat and the previous run mapping the bottom of the cave contributed to the computation unit overheating. After halting the process and restarting the laptop, run number six proceeded normally.

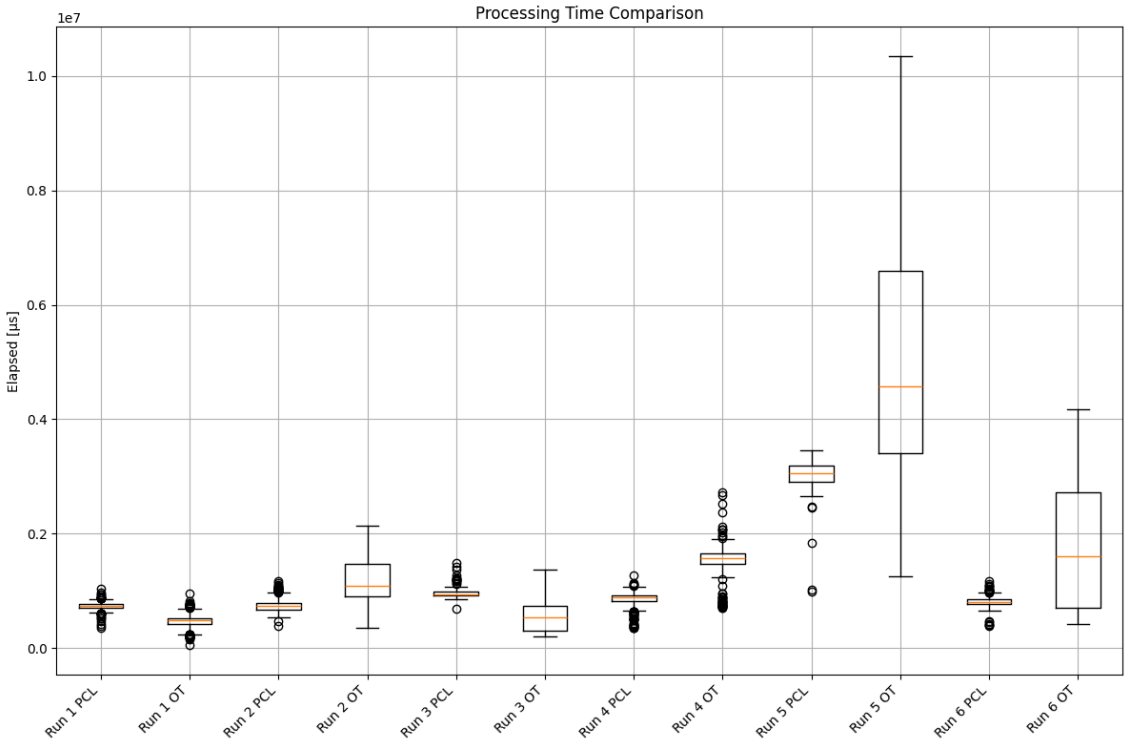


Figure 4.14: *Processing times, per run, for both the Point Cloud Accumulator (PCL) and Hybrid Mapper (OT) nodes.*

Table 4.3: `HybridMapperNode` Processing Logs Excerpt

Timestamp	Elapsed (μs)	Num. Voxels
20250628-155833	1464623	127744
20250628-155837	1446880	130840
20250628-155840	1410673	131249
...	...	...
20250628-160300	10344430	4131158

The efficiency of the OctoMap format is fundamentally linked to the geometric complexity of the environment. The octree data structure achieves compression by representing large, uniform volumes of space (either occupied or free) with a small number of parent nodes. The statistics indicate that octrees are highly susceptible to noisy environments where clutter or jagged surfaces and tight spaces do not allow large expanses of unoccupied space to be pruned. It would be highly beneficial, for non-robotics applications, to remove the nodes representing the free space. For experimentation and record keeping, this data may be useful for further processing or simulations in projects with goals beyond the scope of this study. Storing data-rich representations of the mapped space instead of removing what may be redundant data allows future work to decide how to preprocess the data into a format tailored for its requirements.

From the timestamps in the generated logs, runs number four and six took around seven and eight minutes, respectively. Including two to five minutes for each run's pose graph stabilization phase and cave traversal time between stations, mapping using stereoscopy took a total of thirty to forty minutes.

4.2 Discussion

4.2.1 Analysis of Stereoscopic Approach

The experimental evaluation reveals a series of trade-offs inherent to the stereoscopic SLAM approach, highlighting both its practical advantages for rapid surveying and its significant operational sensitivities.

A primary strength of the method is the speed and density of data acquisition. Compared to the meticulous, station-based traditional survey, the stereoscopic system captures a complete 3D model of the environment in a fraction of the time. Furthermore, the real-time map construction, which leverages OctoMap's raycasting to actively carve out free space, proved highly effective at mitigating the "ghosting" artifacts that commonly plague 3D reconstruction (as shown in Figs. 3.9 and 3.10). This provides a more fluid and "hands-free" mapping experience, which is a considerable advantage in difficult terrain. Traversal in rugged cave passages benefits greatly from this workflow, as it avoids the operational pauses found in frameworks like RTAB-Map, where the entire process can halt if the system fails to localize or match consecutive keyframes [19]. This aligns with the intended design goal: allowing a speleologist to traverse a cave naturally without being forced to cater to the limitations of the system. In practice, the prototype requires only minimal user

attention—mainly to monitor mapping progress or cover missed areas due to field of view or camera frame rate.

However, the approach is critically dependent on the quality of its initial state and the surrounding environment. The ZED 2's VIO relies on tracking visual features between frames to estimate its 6-DOF pose. This process is highly sensitive to the initial conditions of the mapping session. As demonstrated in the field test workflow, a successful initialization requires a feature-rich scene with good, consistent lighting and varied geometry (Fig. 3.11). Starting in darkness or facing a homogeneous, textureless surface provides insufficient visual information for the stereo matching and feature tracking algorithms, leading to a failure to establish a stable initial pose.

An ill-defined pose can degrade the entire mapping session. The accumulation of small errors in the VIO leads to drift, and while the system's loop closure mechanism is designed to correct this, a poor initial map can cause catastrophic corrections later on, where previously mapped surfaces are incorrectly shifted or rotated. This sensitivity is a well-documented challenge in vSLAM, where environmental factors like poor illumination, the lack of texture, and even excessive camera motion can lead to tracking degradation or failure [13]. During field tests, it was observed that dark, feature-poor areas of the cave led not only to inconsistent depth registration, but also to a tendency for the system to estimate depths closer to its maximum configured range. Depth registration itself was constrained between approximately 0.3 m and 15 m, with surfaces closer than 0.3 m not being measured correctly. This means that crevices found inside the blind tunnel were not always properly registered.

The choice of data structure also presents a trade-off. An OctoMap is an excellent representation for robotics tasks such as navigation, semantic mapping, and path planning, as it explicitly models free, occupied, and unknown space [1]. However, for the sole purpose of high-fidelity digital preservation, other representations like dense point clouds from high-end LiDAR or textured meshes from photogrammetry may be more suitable, albeit at a higher cost in terms of equipment, time, or computational post-processing. RTAB-Map, in contrast, was observed to produce poor dense point clouds as reconstructions of mapped areas, often suffering from ghosting effects similar to those in Fig. 3.9. A potential compromise for users who prefer offline workflows would be to register a graph of the captured point clouds along with the corresponding sensor poses. These could then be replayed and ingested into OctoMap, applying the same principles of the real-time approach but allowing higher-quality offline reconstruction of the 3D map. The experimental data in Subsection 4.1.3 indicates the software system supporting 3D mapping using stereoscopy has plenty of room for improvement, especially in regards to optimization, flexibility (i.e., architectural changes to accommodate tests for other cameras and SLAM algorithms) and data storage.

The proposed system, therefore, finds its niche not as a replacement for high-accuracy digitization methods, but as a rapid, cost-effective tool for generating metrically-correct 3D models for real-time analysis and robotic applications.

4.2.2 Real-World Viability

Placing the system in a real-world scenario revealed the constraints and challenges that often go unremarked in a laboratory setting. The field test, and the feedback from the speleologists involved, served to identify and confirm key flaws in the prototype across three main areas: portability and ergonomics, environmental resilience, and system endurance.

First, the prototype severely lacks portability. The system was originally designed for deployment on a compact NVIDIA Jetson board. However, due to resource constraints which made the intended hardware unavailable for the final field tests, a consumer laptop was substituted for the experimental validation. This decision, while necessary, proved to be the greatest operational bottleneck. Carrying a heavy, bulky laptop connected by a USB cable to the camera required two operators (one for the camera, one for the computer) and constant coordination. This setup restricted the lead operator's freedom of movement, making it impossible to climb or navigate narrow passages that required the use of both hands. The original plan to use a compact NVIDIA Jetson board would have mitigated this, but the failure of the only available unit forced the use of this non-optimal configuration.

Second, the system has almost no environmental resilience. The field test in the Lapa dos Pocilgões, a relatively accessible cave, highlighted the prototype's vulnerability. The interior of the blind tunnel was damp and tight; while the ZED 2 camera is robustly built, the plastic chassis of the laptop is susceptible to scrapes, impacts, and, critically, water damage from dripping ceilings or humid air. This fragility makes the prototype entirely unsuitable for the more demanding conditions of karst environments, which can involve crawling through mud or traversing inundated chambers.

Lastly, the system's endurance is a critical limiting factor. Processing the high-volume data stream from the ZED 2 is computationally demanding, capturing point clouds with thousands of points at over fifteen frames per second. This requires heavy and sustained usage of both the CPU and the GPU (for CUDA-accelerated depth perception), leading to significant power consumption that quickly drains the laptop's battery. For any serious speleological expedition, where missions can last for many hours far from any power source, such limited operational time renders the prototype impractical.

Despite these physical limitations, the value of the real-time software feedback was an unforeseen advantage. The ability for the operator to see the map being built on-screen was initially for debugging, but it proved invaluable for tracking progress and ensuring complete coverage on-the-fly. This is a key software feature that should be preserved in any future iteration.

Concluding on the viability of the stereoscopic mapping approach, while its methodology is promising, the current prototype is not viable for real-world cave surveying. Its limitations in portability, resilience, and endurance are prohibitive. The future of this approach is therefore contingent on the development of a purpose-built hardware solution. A rugged, waterproof enclosure containing a power-efficient embedded system (such as an NVIDIA Jetson Orin Nano) and a dedicated battery supply would address all the major flaws identified. Such a self-contained unit could be carried in a bag or mounted on a helmet,

finally providing the speleologist with a truly portable and robust tool for rapid 3D mapping.

4.2.3 Positioning Against State of the Art

The proposed stereoscopic mapping system occupies a distinct and pragmatic position within the spectrum of cave surveying technologies, offering a novel balance of trade-offs when compared to traditional, LiDAR-based, and other camera-based methods.

Relative to traditional surveying, the system represents a paradigm shift. Instead of producing a sparse set of manually measured points and interpretive 2D sketches [7], it generates a dense, comprehensive 3D model of the environment. This leap in information density comes at the cost of increased equipment complexity and a different set of environmental sensitivities, as discussed previously. However, the speed of capture and the richness of the 3D data offer possibilities for analysis that are simply unattainable with traditional methods.

When compared to LiDAR, the current gold standard for accuracy, the stereoscopic approach is not a direct competitor in terms of geometric precision. High-end TLS deliver unparalleled accuracy [11], which remains essential for rigorous geomorphological studies. However, this performance comes with prohibitive costs, weight, and a static, stop-and-scan workflow that is ill-suited to rapid traversal [2]. The proposed system positions itself as a cost-effective alternative for applications where sub-centimeter accuracy is not the primary requirement. It provides a metrically accurate model with adequate results that is vastly more informative than a traditional survey, without the logistical and financial burden of LiDAR. As a downside, the abundance of information may make it difficult to consult, making the use of interactive tools for its visualization

Against photogrammetry and SfM, the key advantage of our system is its real-time nature. While SfM can produce visually impressive 3D models from a simple camera, the process typically requires extensive offline computation and a carefully executed pattern of overlapping photographs [12]. The SLAM-based approach, by contrast, provides immediate feedback and a usable map as the cave is being explored. This is a critical advantage for in-field decision-making, verification, and for any robotics applications that require an live model of the world.

Perhaps the most important distinction is against monocular vSLAM. While monocular systems are even cheaper, they suffer from inherent scale ambiguity: they can reconstruct the shape of an environment, but not its true size [14]. For topography, where absolute measurement is fundamental, this makes them unsuitable. By using a stereo camera with a fixed baseline, our system resolves this ambiguity, generating a true-to-scale map from the very first frame.

Finally, the choice of OctoMap as the data structure is a significant departure from the file formats typically used in speleology. A traditional survey might result in a PDF or a Computer-Aided Design (CAD) file; a LiDAR scan produces a dense point cloud. Storing the resulting map in an octree makes it immediately useful for further processing and data extraction, possibly even useful for robotics tasks like path planning and autonomous

exploration in the future. It opens the door to advanced spatial analysis one step beyond what is possible with 2D maps or unstructured 3D points. The system, therefore, does not just digitize the cave walls; it creates a structured, machine-readable understanding of the 3D space itself.

5 Conclusion

This work set out to study and evaluate the viability of a portable, cost-effective stereoscopic SLAM system for the specific application of cave topography. The research successfully developed and tested a complete hardware and software prototype, yielding critical insights into both the promise of the technology and its significant real-world limitations.

The primary achievement of this work is the successful demonstration of a real-time 3D mapping pipeline that bridges the gap between traditional and cutting-edge methods with stereoscopy. It represents a balanced midpoint between the widely adopted, labor-intensive survey methods and expensive, high-end LiDAR systems. We established that a consumer-grade passive stereo camera, when coupled with a robust software stack, can generate dense, metrically accurate 3D models of a complex cave environment. A key finding was the effectiveness of our custom mapping node, which leverages the raycasting capabilities of the OctoMap library. This approach proved highly successful in mitigating the persistent "ghosting" artifacts common in stereo-based reconstruction, producing a cleaner and more accurate map in real-time than was achieved with the offline dense reconstruction features of established frameworks like RTAB-Map. This real-time feedback loop, allowing the operator to see the map build and make live corrections, demonstrating a powerful advantage for ensuring full coverage during a survey.

The project also illuminated limitations which are not always obvious, perhaps the most significant being the physical prototype itself. The laptop-based system, showed to be not the most suited to the realities of caving. Its lack of portability, poor ergonomics, and vulnerability to the harsh, damp environment highlighted the pain points to keep in mind when designing a more refined setup. For any new technology to be adopted by the speleology community, it must match the ruggedness and unobtrusiveness of their existing tools. Furthermore, our field tests underscored the inherent sensitivity of visual-inertial SLAM to its environment. The system's performance depended critically on consistent lighting and the presence of visual features, a significant constraint in dark and morphologically uniform passages of the cave. This operational fragility requires a level of user accommodation that is, currently, a disadvantage compared to the simplicity of traditional tools.

The contribution of this research to the speleology community is therefore twofold. First,

it provides a successful methodology for a cost-effective 3D mapping system, which is an order of magnitude cheaper than LiDAR, making dense digital modeling of underground spaces more accessible. Secondly it also offers a clear-eyed assessment of the practical obstacles that still remain. By highlighting the poor usability and user experience of the prototype, we steer the conversation towards the user-centric engineering required for a truly field-ready device. The introduction of data structures like OctoMap also opens the door for the speleology community to move beyond simple visualization towards advanced spatial analysis, volume calculation, and even future robotic exploration, using maps that are machine-readable from their inception. The stereoscopy and OctoMap pipeline provides a principled, metric, and compact representation making it attractive for real-time field use and robotic applications. However, the vulnerability on initialization, texture or lighting scarcity, and rolling shutter along with compute constraints explains the empirical limitations observed in field tests and motivates careful initialization protocols, constrained acquisition ranges, optional sensor fusion (IMU/LiDAR), and offline reprocessing to improve fidelity.

5.0.1 Future Work

The insights gained from this study lay a clear foundation for future work, which can be broadly categorized into hardware engineering, software optimization, and advanced data processing.

First and foremost, the most critical next step is to address the prototype's primary shortcomings – fragility, bulk and low autonomy. The development of a rugged, self-contained hardware platform offers the most return on investment, enabling collection of more and better experimental data from mapping runs. The future system should be a waterproof, impact-resistant unit, likely housed in a PVC or custom-molded enclosure, containing a power-efficient embedded computer (e.g., an NVIDIA Jetson Orin Nano) and a substantial lithium-ion battery pack. This would create a truly portable, single-unit device that a speleologist can carry effortlessly in their bag (Fig. 3.3), freeing their hands and enabling access to more challenging cave environments. This would solve the issues of portability, resilience, and endurance simultaneously.

On the software front, there is some room for optimization. The current ROS 2 based pipeline, while flexible within the ecosystem, suffers from locking the system into depending on its topics for communication and third party nodes with unreliable versioning or possibly discontinued support. A future iteration could be built on a more lightweight framework or as a monolithic C++ application to maximize performance. A key bottleneck identified was the point cloud filtering process, which consumes a significant portion of the processing time for each frame. The modular filter chain in the `PointCloudAccumulatorNode` was designed with future optimization in mind; these filters could be re-implemented to run in parallel on the GPU, drastically reducing latency and allowing the system to process more points or run at a higher frame rate. While this could enhance the real-time responsiveness and detail of the map, it is important to consider that not all filter operations are parallelizable and the overhead of scheduling jobs for the GPU may limit any speed-ups.

Implementing a custom mapping framework with its own SLAM pipeline (Fig. 5.1) would grant the precise control over mechanisms such as loop closure, abstracted behind the ZED wrapper node. This would grant control over how features, keypoints or residuals are extracted from images at the SLAM frontend and how pose graph optimization is handled at the backend. Previously mapped zones could be adjusted using the available information to circumvent the pose graph calibration phase and prevent glitches as demonstrated in Fig. 4.10. Developing this system in a modular fashion, leveraging interfaces and polymorphism, different SLAM libraries and algorithms could be experimented with. The "Mapper" and "Viewer" components would, as an example, take the logic present in the custom ROS 2 nodes developed for this study and replace RViz2, respectively. Different types of viewers could be developed, with features tailored to correspond to the needs of future experiments (e.g., exporting maps to varied formats, toggling between different map views and projections such as 2D height maps and real-time dashboards for each run).

Fig. 5.1 encapsulates the "Data Provider" as an abstract representation of a source of sensor data. It would house code specific to each sensor. To avoid bottlenecking acquisition, the system would be made non-blocking by running it on a separate thread to the core "SLAM" mechanism. "Frames" containing sensor data (e.g., images and associated IMU data) would be enqueued by the provider and dequeued by the core, running on its own thread – the queue, a shared resource. The "SLAM" component coordinates the algorithm, calling upon multiple SLAM "Frontend" workers to extract information from the frames and enqueue them. From it, keyframes are identified and contribute to building the pose graph in the SLAM "Backend". Keyframes are enqueued as loop closure candidates. As the process carried out by the "Loop Closure" worker may take some time, it runs on another thread as well. When new loops are formed, they get added to the "Backend" and pose graph optimization is triggered.

In an effort to take a scientific approach to the extraction of information from 3D mapping, contributing to the body of knowledge of the field of speleology, it is necessary to properly organize experimental data. Designing and deploying a centralized repository of mapping run results obeying to a set structural standard. Such constraints would be conducive to implementing automated batch data processing to extract key insights into performance and computational resource consumption. At a more advanced stage, that data could feed the training of machine learning (ML) models to identify geological formations (provided with proper manual labeling first). These would then be retroactively integrated into the pipeline to enhance the extraction of information. This methodic approach to data collection and usage would serve to inform and guide data-driven decision for future directions. Along with the envisioned mapping framework, this would allow configurations, algorithms and libraries, as well as their versions, to be registered along with the results. Ensuring technical reports cover the process from beginning to end makes results trackable and reproducible.

Finally, the ultimate goal is to make the rich 3D data more accessible and useful. The most important future work, from the point of view of a speleologist, would be the development of algorithms for the automatic extraction of 2D topographic plans from the generated 3D models. This would involve programmatically slicing the OctoMap to generate plan

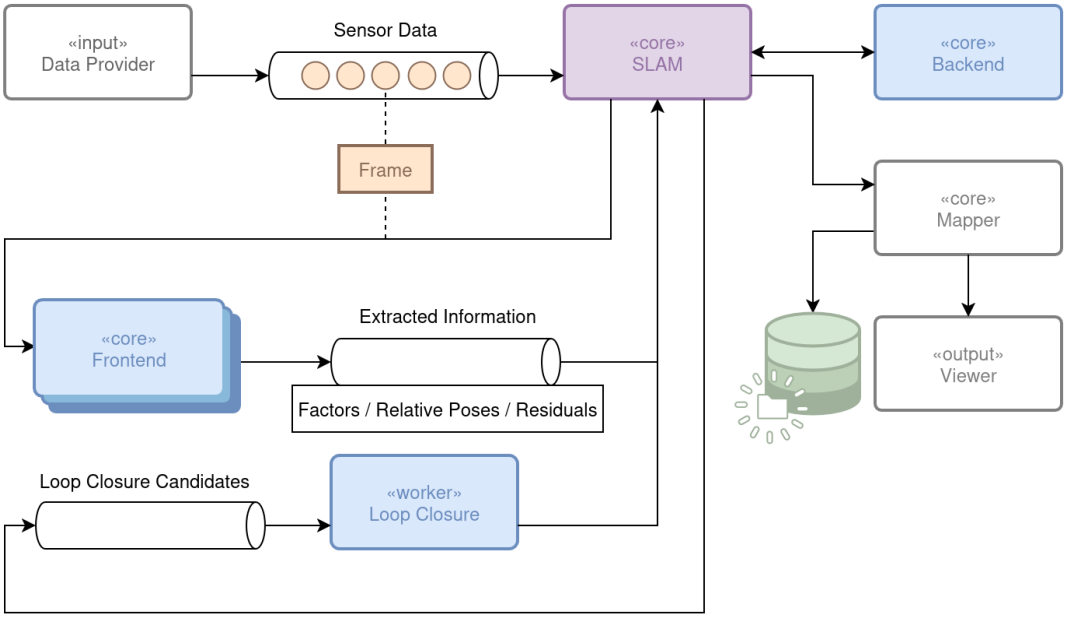


Figure 5.1: *High level diagram showcasing a proposed architecture for a modular, trackable and versionable 3D mapping (SLAM) experimentation framework.*

and profile views, and potentially using computer vision techniques to identify and render standard speleological symbols. Such a feature would bridge the gap between the robotics-focused 3D model and the familiar, standardized 2D maps that are common and readily understood by the caving community. Building on this, semantic mapping offers another exciting frontier. By employing the ML models to identify geomorphological formations, geometric maps could enhance the topography with a rich, multi-layered geological and environmental model. A specialized viewer could allow users to quickly alternate between layers to omit or highlight the information most relevant to them (e.g., hide vegetation, hide geological details and display only depth and fossilized remains from a top-down perspective) transforming it from a simple surveying tool into a powerful scientific instrument.

Bibliography

[1] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* 34.3 (2013), pp. 189–206 (cit. on pp. 1, 8, 43).

[2] M.-A. Leclerc, J. Bass, M. Labbé, D. Dozois, J. Delisle, D. Rancourt, and A. Lussier Desbiens. "NetherDrone: A tethered and ducted propulsion multirotor drone for complex underground mining stope inspections". In: *Drone Systems and Applications* 11 (2023), pp. 1–17 (cit. on pp. 2, 6, 45).

[3] X. Han, S. Li, X. Wang, and W. Zhou. "Semantic Mapping for Mobile Robots in Indoor Scenes: A Survey". In: *Information* 12 (Feb. 2021), p. 92. DOI: 10.3390/info12020092 (cit. on p. 2).

[4] X. Ruan, P. Guo, and J. Huang. "A Semantic OctoMap Mapping Method Based on CBAM-PSPNet". In: *Journal of Web Engineering* 21.3 (2022), pp. 879–910. DOI: 10.13052/jwe1540-9589.21315 (cit. on p. 2).

[5] S. Raychaudhuri and A. X. Chang. *Semantic Mapping in Indoor Embodied AI – A Survey on Advances, Challenges, and Future Directions*. 2025. arXiv: 2501.05750 [cs.RO]. URL: https://arxiv.org/abs/2501.05750 (cit. on p. 2).

[6] J. Mattes. "Underground fieldwork—A cultural and social history of cave cartography and surveying instruments in the 19th and at the beginning of the 20th century". In: *International Journal of Speleology* 44.3 (2015), pp. 251–266 (cit. on p. 5).

[7] K. P. Trimmis. "Paperless mapping and cave archaeology: A review on the application of DistoX survey method in archaeological cave sites". In: *Journal of Archaeological Science: Reports* 18 (2018), pp. 399–407 (cit. on pp. 5, 45).

[8] B. (Paperless). *Paperless*. Online. Available: https://paperless.bheeb.ch/, Accessed: 5 September 2025. n.d. (Cit. on p. 5).

[9] D. Bristol. *Electronic Cave Survey*. Online. Available: https://www.derekbristol.com/electronic-survey, Accessed: 5 September 2025. n.d. (Cit. on p. 5).

[10] M. Corvi. *TopoDroid*. 2017. URL: https://raw.githubusercontent.com/marcocorvi/topodroid/master/manual.pdf (visited on 09/25/2025) (cit. on p. 5).

[11] T. Slavova. "Modern methods and devices for mapping underground galleries and natural caves". In: *Modern Technologies, Education and Professional Practice in Geodesy and Related Fields*. 2013 (cit. on pp. 6, 45).

[12] E. Nocerino, F. Menna, E. Farella, and F. Remondino. "3D virtualization of an underground semi-submerged cave system". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W15 (2019), pp. 857–864 (cit. on pp. 6, 12, 45).

[13] M. R. Saputra, P. P. de Gusmão, L. Sentis, and P. Stone. "Visual SLAM and Structure from Motion in Dynamic Environments: A Survey". In: *ACM Computing Surveys (CSUR)* 51.2 (2018). Article 37, pp. 1–36 (cit. on pp. 6, 43).

[14] L. Somlyai and Z. Vámossy. "Improved RGB-D Camera-based SLAM System for Mobil Robots". In: *Acta Polytechnica Hungarica* 21.8 (2024), pp. 107–124 (cit. on pp. 7, 45).

[15] J. Behley and C. Stachniss. "Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments". In: *Robotics: Science and Systems (RSS)*. 2018 (cit. on p. 7).

[16] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. "A tutorial on graph-based SLAM". In: *IEEE Transactions on Intelligent Transportation Systems Magazine* 2 (Dec. 2010), pp. 31–43. DOI: 10.1109/MITS.2010.939925 (cit. on p. 7).

[17] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 (Dec. 2016), 1309–1332. ISSN: 1941-0468. DOI: 10.1109/tro.2016.2624754. URL: http://dx.doi.org/10.1109/TRO.2016.2624754 (cit. on p. 7).

[18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. "g2o: A general framework for graph optimization". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 3607–3613 (cit. on p. 7).

[19] M. Labbé and F. Michaud. "RTAB-Map as an open-source lidar and visual slam library for large-scale and long-term online operation". In: *Journal of Field Robotics* 36.2 (2019), pp. 416–446. URL: https://onlinelibrary.wiley.com/doi/10.1002/rob.21831 (visited on 09/05/2025) (cit. on pp. 7, 42).

[20] J. Stückler and S. Behnke. "Multi-resolution surfel maps for efficient dense 3D modeling and tracking". In: *Journal of Visual Communication and Image Representation* 25.1 (2014), pp. 137–147 (cit. on p. 8).

[21] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1366–1373 (cit. on p. 8).

[22] L. Schmid, J. Dubach, C. Stachniss, and C. Cadena. "MAP-ADAPT: Real-Time Quality-Adaptive Semantic 3D Maps". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024 (cit. on p. 9).

[23] Ouster. *Ouster OS0 Ultra-Wide View LiDAR Sensor*. 2024. URL: https://ouster.com/products/hardware/os0-lidar-sensor (visited on 09/25/2025) (cit. on p. 9).

[24] Leica Geosystems. *Leica BLK360*. 2024. URL: https://leica-geosystems.com/products/laser-scanners/scanners/leica-blk360 (visited on 09/25/2025) (cit. on p. 9).

[25] Stereolabs. *ZED 2 – AI Stereo Camera for 3D Depth Sensing and Motion Tracking*. 2024. URL: https://www.stereolabs.com/zed-2/ (visited on 09/25/2025) (cit. on p. 10).

[26] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM". In: *IEEE Transactions on Robotics* 37.6 (Dec. 2021), 1874–1890. ISSN: 1941-0468. DOI: 10.1109/tro.2021.3075644. URL: http://dx.doi.org/10.1109/TRO.2021.3075644 (cit. on p. 10).

[27] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petrlík, A. Reinke, V. Krátký, S. Zhao, A.-a. Agha-mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau, M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone. "Present and Future of SLAM in Extreme Environments: The DARPA SubT Challenge". In: *IEEE Transactions on Robotics* 40 (2024), pp. 936–959. DOI: 10.1109/TRO.2023.3323938 (cit. on p. 14).

[28] A. Abdullah, T. Barua, R. Tibbetts, Z. Chen, M. J. Islam, and I. Rekleitis. "CaveSeg: Deep Semantic Segmentation and Scene Parsing for Autonomous Underwater Cave Exploration". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Dataset and code available at robopi.ece.ufl.edu/caveseg. 2024 (cit. on p. 14).

[29] T. Racine et al. "KarstConduitCatalogue: a dataset of LiDAR derived point clouds for the analysis of karstic conduit geometry and morphology". In: *Earth System Science Data* 17 (2025), pp. 4671–4690. DOI: 10.5194/essd-17-4671-2025 (cit. on p. 14).

[30] AllTrails. *Trilho da Lapa dos Pocilgões (PR3 PMS)*. URL: https://www.alltrails.com/pt-pt/trilho/portugal/leiria/pr3-lapa-dos-pocilgoes (visited on 09/25/2025) (cit. on p. 27).

[31] Instituto da Conservação da Natureza e das Florestas (ICNF). *Parque Natural das Serras de Aire e Candeeiros*. 2024. URL: https://www.icnf.pt/conservacao/rnapareasprotegidas/parquesnaturais/pnserrasdeaireecandeeiros (visited on 09/25/2025) (cit. on p. 27).

[32] A. Flor, A. R. Pina, C. Mendes, et al. *A Vegetação de Portugal*. 1st ed. Lisboa: Imprensa Nacional-Casa da Moeda, May 2021. URL: https://imprensanacional.pt/wp-content/uploads/2022/03/A-Vegetacao-em-Portugal.pdf (visited on 09/25/2025) (cit. on p. 27).

I Annex 1 Development Environment

The `tfm17-dev-environment-v1` project provides a portable, Docker-based development environment created for the study on 3D cave mapping. Its primary purpose is to facilitate research and development of prototypes which integrate 3D mapping tools with a focus on the stack described in this dissertation. The architecture is centered around a containerized ROS 2 (Humble) workspace, ensuring a reproducible and encapsulated setup. The environment is defined and managed using Docker Compose. A custom Dockerfile extends the official Stereolabs ZED SDK image with ROS 2, CUDA, and essential mapping libraries such as OctoMap. RTAB-Map and ORB-SLAM3 are included as Git submodules, but only tested in the second version. The configuration handles necessary hardware integrations, including GPU acceleration via the NVIDIA runtime and USB device passthrough for the camera. Functionality is managed through a set of Bash scripts that automate dependency checking, image building, and container life-cycle operations. The environment is suitable for developing and running custom ROS 2 nodes, enabling fast configuration of modular processing pipelines with sensor integration. Python is incorporated, via Jupyter Notebooks, for data analysis and visualization. They are primarily assisted by the `numpy`, `pandas`, `matplotlib` and `open3d` libraries to process performance logs and render 3D point cloud data. This combination of technologies creates a self-contained and powerful platform for rapid prototyping and experiments.

The second version, `tfm17-dev-env-v2`, extends the first to test mapping algorithms that were not fully configured in the base image. The `README.md` file serves as a detailed log of the steps required to install and build key libraries from source within the running container. This includes the Pangolin library, the ORB-SLAM3 visual SLAM system, and its corresponding ROS 2 wrapper. It also covers the setup for RTAB-Map and experimental integration of OctoMap-RT, a GPU-accelerated version of OctoMap using NVIDIA OptiX. Crucially, it establishes a robust workflow for recording and replaying sensor data using `rosbag2`, enabling consistent and repeatable testing. Comparative analysis is performed in Jupyter Notebooks designed to automatically recover and process data from multiple experimental runs to generate comparative plots and extract performance metrics.

Repositories:

- Version 1: https://github.com/martim-goncalves/tfm17-dev-environment-v1
- Version 2: https://github.com/martim-goncalves/tfm17-dev-env-v2

II Annex 2 Point Cloud Accumulator Package

The `point_cloud_accumulator_pkg` is a self-contained ROS 2 package implemented in C++ that functions as a point cloud processing and aggregation node. Its primary role within a 3D mapping pipeline is to ingest a stream of point cloud frames from a sensor, apply a series of data enhancement filters, and fuse them into a single, persistent point cloud map. It publishes both the filtered cloud and the accumulated cloud to separate topics. Its architecture was designed to allow fast implementation and testing of point cloud filter implementations and perform corrections to SLAM. Another significant feature is the dynamic downsampling mechanism.

The node leverages the PCL for its core processing tasks. It is highly configurable via a YAML file and a ROS 2 launch file, exposing parameters for ease of testing. For diagnostics and analysis, the package includes a custom logging utility that records detailed performance metrics for each filter into TSV files.

Repository: `https://github.com/martim-goncalves/point_cloud_accumulator_pkg`

Annex 3 Hybrid Mapper Package

The `hybird_mapper_pkg` is a ROS 2 package that implements a hybrid mapping strategy, designed to create a colored, volumetric 3D map from the output of an external SLAM system. It subscribes to a topic holding point cloud messages.

The architectural choice of decoupling point cloud processing from OctoMap building was made to separate the complex task of SLAM from the process of building a specific type of map representation. By relying on an external service for accurate localization and the creation of a static map, this node can focus on translating that output into a volumetric OctoMap. It improves modularity, allowing future different SLAM providers and map builders to be interchangeable in the pipeline. Another advantage is that separating concerns in individual, smaller packages reduces complexity and makes it easier to apply changes exclusively where necessary.

Implemented in C++17, the package uses the OctoMap library to build a map. It is configurable via a YAML file and a ROS 2 launch file, with parameters for map resolution and the input topic. The package also includes utilities for performance logging and for saving the final map to a file, either periodically or upon shutdown.

Repository: https://github.com/martim-goncalves/hybrid_mapper_pkg

Annex 4 Custom Map Viewer

A full-stack web application to render and interact with 3D maps from binary OT files was developed as an experiment during this dissertation. While not the focus of the study, it emerged from the future work identified as necessary to view and work with 3D mapping experiment results. It provides a viewer where users to upload a map file, visualize it in 3D, and export it in PLY format.

The application employs a client-server architecture. The server is built with Python using the FastAPI framework, while the client is a Single Page Application (SPA) developed with Angular. The server exposes a REST API to handle core operations. Its primary endpoint receives an uploaded OctoMap file and executes a custom C++ script to serialize octree data into a JSON format understood by the client. The client provides an interactive user interface for the 3D map. The visualization is powered by Three.js, which renders the voxel data within an HTML5 canvas. For performance, it leverages an instanced mesh to efficiently draw a large number of voxels. The UI allows users to upload map files and includes controls for toggling rendering options like shading, switching between perspective and orthographic camera projections, and defining a bounding box to select and view specific regions of the map.

The project is designed for production deployment, with documentation detailing how to configure an Apache web server as a reverse proxy. This setup serves the static files of the built Angular application while forwarding API requests to the backend, which is run as a persistent `systemd` service using `gunicorn`. Deployment has been tested and found to be functional, but the domain referenced below may be suspended at the time of reading.

Repository: `https://github.com/martim-goncalves/map-viewer`

Domain: `http://speleolabs.com/`

Annex 5 Stereolabs' SVO File Format Test

A very small project was built to test the ZED 2 camera and Stereolabs' proprietary SVO format. It details how to create a Docker container running the ZED Explorer software, using diagnostics tools to search for compatible official images. A Python script is available to convert a recorded stereo video stream stored in a SVO file to PLY. Tools such as CloudCompare or `open3d` can then be used to view the results.

Repository: https://github.com/martim-goncalves/ZED_Explorer